Package ‘ArvindRF’

August 21, 2026

Type Package

Title Random Forest Regression with Arvind Distribution Error Model

Version 1.0.0

Description Implements Random Forest regression under the Arvind distribution error model. Provides core distribution functions (density, cumulative distribution, quantile, random generation, hazard, survival), parameter estimation via Expectation-Maximization (EM) and Markov Chain Monte Carlo (MCMC), non-parametric bootstrap confidence intervals (at 90%, 95%, and 99% levels), Highest Posterior Density (HPD) intervals, model evaluation metrics (estimated values, bias, mean squared error, risk value), homoscedastic prediction intervals, and goodness-of-fit diagnostic tests (Kolmogorov-Smirnov and Anderson-Darling tests, Akaike Information Criterion, and Bayesian Information Criterion). References: Breiman (2001) <doi:10.1023/A:1010933404324>; Wright and Ziegler (2017) <doi:10.18637/jss.v077.i01>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports ranger, coda, goftest, stats, graphics

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Shikhar Tyagi [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1606-0844>),
Aruna Rajballie [aut],
Vrijesh Tripathi [aut]

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-08-21 13:10:53 UTC

Contents

---

arvind *The Arvind Distribution*

Description

Density, distribution function, quantile function, random generation, hazard function, and survival function for the Arvind distribution with parameter `theta`.

Usage

```
darvind(x, theta, log = FALSE)

parvind(q, theta, lower.tail = TRUE, log.p = FALSE)

qarvind(p, theta, lower.tail = TRUE, log.p = FALSE)

rarvind(n, theta)

harvind(x, theta)

sarvind(x, theta)
```

Arguments

`x, q`	Vector of quantiles.
`theta`	Scale parameter ($\theta > 0$).
`log, log.p`	Logical or character string indicating boolean evaluation (`TRUE` or `"TRUE"` / `FALSE` or `"FALSE"`). If `TRUE`, probabilities/densities are given as `log(p)`.
`lower.tail`	Logical or character string indicating boolean evaluation. If `TRUE` (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
`p`	Vector of probabilities.
`n`	Number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

The Arvind distribution is a continuous distribution on $[0, \infty)$ with scale parameter $\theta > 0$. Its probability density function (PDF) is given by:

$$f(x; \theta) = \frac{\theta(1 + 2x + 2\theta x^2)}{(1 + \theta x)^2} \exp(-\theta x^2), \quad x \geq 0, \theta > 0$$

Its cumulative distribution function (CDF) is:

$$F(x; \theta) = 1 - \frac{\exp(-\theta x^2)}{1 + \theta x}, \quad x \geq 0$$

Its survival function is:

$$S(x; \theta) = \frac{\exp(-\theta x^2)}{1 + \theta x}, \quad x \geq 0$$

Its hazard function is:

$$h(x; \theta) = \frac{\theta(1 + 2x + 2\theta x^2)}{1 + \theta x}, \quad x \geq 0$$

Quantiles are calculated using root-finding via `uniroot`, and random values are generated via inversion sampling.

Value

`darvind` gives the density, `parvind` gives the distribution function, `qarvind` gives the quantile function, `rarvind` generates random deviates, `harvind` gives the hazard function, and `sarvind` gives the survival function.

Examples

```
# Evaluate PDF and CDF at x = 1, theta = 0.5
darvind(1, theta = 0.5)
parvind(1, theta = 0.5)

# Quantile function
qarvind(0.95, theta = 0.5)

# Survival and hazard functions
sarvind(1, theta = 0.5)
harvind(1, theta = 0.5)

# Generate random samples
set.seed(123)
r_samples <- rarvind(10, theta = 0.5)
print(r_samples)
```

diagnostics — *Goodness-of-Fit Diagnostics and S3 Methods for RF-Arvind Models*

Description

Evaluates Kolmogorov-Smirnov (KS) and Anderson-Darling (AD) goodness-of-fit diagnostic tests, information-theoretic criteria (AIC, BIC), and provides `summary`, `print`, and `plot` methods for `rf_arvind` objects.

Usage

```
ks_arvind(r, theta)

ad_arvind(r, theta)

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

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

## S3 method for class 'summary.rf_arvind'
print(x, ...)

## S3 method for class 'rf_arvind'
plot(x, ...)
```

Arguments

`r`	Vector of absolute residuals.
`theta`	Estimated Arvind distribution parameter θ.
`...`	Additional arguments passed to default methods.
`object, x`	Object of class `rf_arvind`.

Details

Goodness-of-fit of absolute residuals $r_i = |y_i - \hat{y}_i|$ to the Arvind distribution is evaluated via:

- **Kolmogorov-Smirnov Test:** Compares empirical CDF $F_n(x)$ against Arvind CDF $F(x; \hat{\theta})$.
- **Anderson-Darling Test:** Evaluates tail-weighted goodness-of-fit using `ad.test`.
- **AIC and BIC:** Calculated as $\mathrm{AIC} = 2k - 2\ell(\hat{\theta})$ and $\mathrm{BIC} = k \log n - 2\ell(\hat{\theta})$ with parameter count $k = 1$.

Value

For `ks_arvind` and `ad_arvind`, a list containing test statistics and p-values. For `summary.rf_arvind`, a summary list object printed cleanly. For `plot.rf_arvind`, produces diagnostic plots.

Examples

```
set.seed(123)
df <- data.frame(
  y = 5 + 2 * rnorm(50),
  x1 = rnorm(50),
  x2 = runif(50)
)
fit <- rf_arvind(y ~ x1 + x2, data = df, num.trees = 50)
summary(fit)
plot(fit)
```

---

mcmc_bayesian — *Bayesian MCMC Estimation and Highest Posterior Density (HPD) Intervals*

Description

Estimates the Arvind distribution parameter `theta` using Markov Chain Monte Carlo (MCMC) sampling, and computes Highest Posterior Density (HPD) intervals at 90

Usage

```
mcmc_arvind(
  r,
  n_iter = 5000,
  burn_in = 1000,
  thin = 2,
  init_theta = NULL,
  seed = NULL
)
```

Arguments

`r`	Vector of absolute residuals.
`n_iter`	Total number of MCMC iterations (default `5000`).
`burn_in`	Number of initial burn-in iterations to discard (default `1000`).
`thin`	Thinning interval for posterior chain (default `2`).
`init_theta`	Initial value for θ. If `NULL`, calculated via `arvind_mle`.
`seed`	Optional random seed.

Details

MCMC sampling is performed using an adaptive Metropolis-Hastings algorithm on the log-scale of θ ($\phi = \log \theta$) to ensure positivity and numerical stability. A weakly informative Gamma(0.001, 0.001) prior is placed on θ. The posterior chain is analyzed using `HPDinterval` to extract HPD intervals at 90

Value

A list containing:

posterior_mean Posterior mean estimate of θ.

posterior_median Posterior median estimate of θ.

hpd_90 Highest Posterior Density interval at 90% level.

hpd_95 Highest Posterior Density interval at 95% level.

hpd_99 Highest Posterior Density interval at 99% level.

acceptance_rate Acceptance rate of the MCMC sampler.

mcmc_chain mcmc object of posterior samples.

Examples

```
set.seed(123)
r_sample <- rarvind(50, theta = 0.5)
mcmc_fit <- mcmc_arvind(r_sample, n_iter = 2000, burn_in = 500)
print(mcmc_fit$posterior_mean)
print(mcmc_fit$hpd_95)
```

---

mle_em *Maximum Likelihood Estimation and Bootstrap Inference for Arvind Distribution*

Description

Functions to compute the Maximum Likelihood Estimate (MLE) / Expectation-Maximization (EM) point estimate for the Arvind distribution parameter theta, and nonparametric bootstrap confidence intervals.

Usage

```
arvind_mle(r, tol = 1e-09, max_iter = 100)

bootstrap_ci(
  y,
  X,
  n_boot = 100,
  conf_levels = c(0.9, 0.95, 0.99),
  num.trees = 100,
  seed = NULL
)
```

Arguments

r	Vector of non-negative absolute residuals.
tol	Numerical tolerance for convergence (default 1e-9).
max_iter	Maximum number of iterations (default 100).
y	Response vector for bootstrap estimation.
X	Predictor matrix or data frame.
n_boot	Number of bootstrap samples (default 100).
conf_levels	Vector of confidence levels (default c(0.90, 0.95, 0.99)).
num.trees	Number of trees for ranger in bootstrap fits (default 100).
seed	Optional random seed.

Details

The score equation for θ given absolute residuals $r_i = |y_i - \hat{y}_i|$ is:

$$\frac{n}{\theta} = \sum_{i=1}^{n} \left[r_i^2 + \frac{2r_i}{1+\theta r_i} - \frac{2r_i^2}{1+2r_i+2\theta r_i^2} \right]$$

This equation is solved iteratively using Newton-Raphson / EM optimization. Under the EM/MLE method, confidence intervals for parameters and performance metrics (bias, MSE, risk) are obtained using non-parametric bootstrap resampling at 90

Value

For arvind_mle, a single scalar estimate of θ. For bootstrap_ci, a list containing point estimates, bootstrap samples, and lower/upper bounds for confidence intervals at the specified confidence levels (90

Examples

```
set.seed(123)
r_sample <- rarvind(50, theta = 0.5)
est_theta <- arvind_mle(r_sample)
print(est_theta)
```

---

predict.rf_arvind *Predict Method for RF-Arvind Models*

Description

Obtains point predictions and homoscedastic prediction intervals (at 90

Usage

```
## S3 method for class 'rf_arvind'
predict(object, newdata = NULL, conf_levels = c(0.9, 0.95, 0.99), ...)
```

Arguments

`object`	Object of class `rf_arvind`.
`newdata`	Data frame of new predictor observations. If omitted, fitted training values are used.
`conf_levels`	Vector of confidence levels for prediction intervals (default `c(0.90, 0.95, 0.99)`).
`...`	Additional arguments passed to `predict.ranger`.

Details

Under the Arvind error distribution, the $(1-\alpha)100\%$ prediction interval for a new response Y^* at predictor $\mathbf{x}^*$ is given by:

$$\mathrm{PI}_{1-\alpha}(\mathbf{x}^*) = [\hat{f}(\mathbf{x}^*) - Q_{1-\alpha}(\hat{\theta}),\ \hat{f}(\mathbf{x}^*) + Q_{1-\alpha}(\hat{\theta})]$$

where $Q_{1-\alpha}(\hat{\theta})$ is the $(1-\alpha)$-th quantile of the Arvind distribution computed via `qarvind`.

Value

A data frame containing:

`fit`	Point predictions $\hat{y}^*$.

Columns for lower and upper prediction interval bounds for each requested level (e.g. `PI_90_Lower`, `PI_90_Upper`, `PI_95_Lower`, `PI_95_Upper`, `PI_99_Lower`, `PI_99_Upper`).

Examples

```
set.seed(123)
df <- data.frame(
  y = 5 + 2 * rnorm(50),
  x1 = rnorm(50),
  x2 = runif(50)
)
fit <- rf_arvind(y ~ x1 + x2, data = df, num.trees = 50)
preds <- predict(fit, newdata = df[1:5, ])
print(preds)
```

rf_arvind *Random Forest Regression with Arvind Distribution Error Model*

Description

Fits a Random Forest regression model combined with an Arvind distribution error model. Supports parameter estimation via EM algorithm (with bootstrap confidence intervals at 90

Usage

```
rf_arvind(
  formula,
  data,
  method = c("em", "mcmc"),
  num.trees = 500,
  n_boot = 100,
  n_iter = 5000,
  burn_in = 1000,
  conf_levels = c(0.9, 0.95, 0.99),
  na.action = stats::na.omit,
  seed = NULL,
  ...
)
```

Arguments

`formula`	Object of class `formula` specifying model formula.
`data`	Data frame containing the variables in the model.
`method`	Estimation method: `"em"` (default) or `"mcmc"`.
`num.trees`	Number of trees in the Random Forest ensemble (default `500`).
`n_boot`	Number of bootstrap iterations for EM confidence intervals (default `100`).
`n_iter`	Number of MCMC iterations for Bayesian estimation (default `5000`).
`burn_in`	Number of burn-in MCMC iterations (default `1000`).
`conf_levels`	Confidence levels for intervals (default `c(0.90, 0.95, 0.99)`).
`na.action`	Function specifying action for missing values (default `na.omit`).
`seed`	Optional integer seed for reproducibility.
`...`	Additional arguments passed to `ranger`.

Details

The RF-Arvind regression model assumes:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \ldots, n$$

where $f(\mathbf{x})$ is estimated by a Random Forest predictor, and absolute errors $|\varepsilon_i|$ follow the Arvind distribution with scale parameter $\theta > 0$.

In Stage 1, Random Forest regression is fitted using `ranger` under the mean squared error criterion to predict point estimates $\hat{y}_i = \hat{f}(\mathbf{x}_i)$. In Stage 2, absolute residuals $r_i = |y_i - \hat{y}_i|$ are modeled using the Arvind error distribution.

Parameter estimation can be performed via:

- **EM Algorithm (`method = "em"`):** Computes maximum likelihood point estimate $\hat{\theta}$ and uses non-parametric bootstrap resampling to evaluate confidence intervals at 90
- **Bayesian MCMC (`method = "mcmc"`):** Draws MCMC posterior samples for θ via adaptive Metropolis-Hastings and computes Highest Posterior Density (HPD) intervals at 90

Missing values are processed according to the specified `na.action`. Logical parameters accept both standard R boolean flags (`TRUE`/`FALSE`) and character representations (`"TRUE"`/`"FALSE"`).

Value

An S3 object of class `"rf_arvind"`, containing:

rf_model The fitted `ranger` Random Forest object.

theta Estimated Arvind scale parameter θ.

method Estimation method used (`"em"` or `"mcmc"`).

fitted_values Fitted point predictions $\hat{y}$.

residuals Raw residuals $y - \hat{y}$.

abs_residuals Absolute residuals $|y - \hat{y}|$.

bias Estimated model bias.

mse Mean squared error.

rmse Root mean squared error.

mae Mean absolute error.

risk Model risk value (empirical squared loss).

intervals Bootstrap confidence intervals (for EM) or HPD intervals (for MCMC) at 90%, 95%, and 99% levels.

bootstrap_results Detailed bootstrap outputs if `method = "em"`.

mcmc_results Detailed MCMC sampler outputs if `method = "mcmc"`.

formula Model formula.

data Processed training data frame.

response_name Name of response variable.

predictor_names Vector of predictor variable names.

Examples

```
# Fit RF-Arvind model on synthetic regression data
set.seed(123)
df <- data.frame(
  y = 5 + 2 * rnorm(50),
  x1 = rnorm(50),
  x2 = runif(50)
)
fit_em <- rf_arvind(y ~ x1 + x2, data = df, method = "em", num.trees = 50, n_boot = 30, seed = 123)
print(fit_em)
```

Index