Package: multipathaic (via r-universe)

October 29, 2026

Type Package

Title Multi-Path Stepwise Selection with AIC

Version 0.1.1

Description Implements a multi-path forward selection algorithm using the Akaike Information Criterion (AIC) for both linear (Gaussian) and logistic (binomial) regression models. Unlike traditional stepwise methods that follow a single path, this approach explores multiple competitive model paths simultaneously by retaining near-optimal models at each step. The package includes stability estimation via bootstrap resampling and identifies plausible models based on AIC tolerance and variable stability thresholds.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports stats, utils

Suggests lars, knitr, rmarkdown, shiny, shinydashboard, plotly, DT, ggplot2, readxl, htmltools, kableExtra

VignetteBuilder knitr

URL https://github.com/mobuobi/multipathaic

BugReports https://github.com/mobuobi/multipathaic/issues

Repository https://mobuobi.r-universe.dev

Date/Publication 2026-10-07 14:24:38 UTC

RemoteUrl https://github.com/mobuobi/multipathaic

RemoteRef HEAD

RemoteSha 812c589129dd2d6daf99aa6ce0eb9d96a75ca59f

Contents

build_paths 2
confusion_metrics 4
launch_app 6
multipath_aic 6
plausible_models 8
stability 10

Index **13**

`build_paths` *Multi-Path Forward Selection Using AIC*

Description

Implements a branching forward-selection algorithm using AIC for either linear (‘gaussian‘) or logistic (‘binomial‘) regression models.

At each step, every current parent model attempts to add every unused predictor, producing a set of children models. Children are kept if they: (1) improve AIC relative to the parent by at least ‘eps‘, and (2) lie within ‘delta‘ of the best child AIC for that parent.

Duplicate models are merged, and if more than ‘L‘ children remain, only the ‘L‘ best (lowest AIC) models are kept. The process continues for up to ‘K‘ steps or until no parent improves AIC.

Usage

```
build_paths(
  X,
  y,
  family = c("gaussian", "binomial"),
  K = min(ncol(X), 10),
  eps = 1e-06,
  delta = 2,
  L = 100,
  weights = NULL,
  verbose = TRUE
)
```

Arguments

`X`	A data frame or matrix of predictors. All predictors must be numeric with finite values; encode categorical predictors explicitly.
`y`	A numeric response vector. For ‘family = "binomial"‘, must be 0/1.
`family`	Character string: either ‘"gaussian"‘ (linear regression) or ‘"binomial"‘ (logistic regression).
`K`	Integer. Maximum number of forward-selection steps to explore. Default: `min(ncol(X), 10)`.

eps	Minimum AIC improvement required for a child model to be considered an improvement over its parent. Default: ‘1e-6‘.
delta	AIC tolerance for keeping near-best children at each step. Children with AIC <= (best child AIC + delta) are retained. Default: 2.
L	Maximum number of child models retained at each step after deduplication. Helps limit tree size. Default: 100.
weights	Optional numeric vector of observation weights for the underlying ‘lm()‘ or ‘glm()‘ fits.
verbose	Logical. If TRUE (default), prints detailed information about each selection step, including candidate AICs and parent/child summaries.

Details

This function constructs a *model forest* representing multiple competitive forward-selection paths, rather than a single greedy stepwise sequence.

For each parent model: - All possible one-variable additions are evaluated. - Children that pass AIC filters (‘eps‘ and ‘delta‘) are retained. - Duplicates are merged by retaining the best AIC among identical variable sets.

The algorithm stops early if no parent produces an improved child.

All candidate models use the same rows. Missing or non-finite inputs are rejected. Failed, non-finite, saturated, and nonconverged or boundary logistic fits are excluded. The search is a bounded heuristic, not exhaustive best-subset selection. Models are cached by variable set. The model list is authoritative; keys escape percent and pipe characters.

Value

A list containing:

path_forest List with ‘frontiers‘ component containing models at each step.

aic_by_model Named list of AIC values indexed by model key.

all_models A combined data frame of retained models across all steps, sorted by size and AIC.

meta A list of the parameters used (‘K‘, ‘eps‘, ‘delta‘, ‘L‘, and ‘family‘).

Examples

```
## Linear regression example
set.seed(1)
n <- 150; p <- 8
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
beta <- c(2, -1.5, 0, 0, 1, rep(0, p-5))
y <- as.numeric(as.matrix(X) %*% beta + rnorm(n, 1))

forest <- build_paths(
  X, y,
  family = "gaussian",
  K = 10, eps = 1e-6, delta = 2, L = 100,
```

```
  verbose = TRUE
)

## Logistic regression example
set.seed(1)
n <- 200; p <- 8
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
eta <- 1.5*X$x1 - 2*X$x2 + 1*X$x5
prob <- 1/(1 + exp(-eta))
y <- rbinom(n, 1, prob)

forest <- build_paths(
  X, y,
  family = "binomial",
  K = 10, eps = 1e-6, delta = 2, L = 100,
  verbose = TRUE
)
```

`confusion_metrics` *Confusion Matrix and Diagnostic Metrics for Selected Logistic Models*

Description

Computes a confusion matrix and several standard diagnostic metrics for a **logistic regression model** selected by the Multi-Path AIC procedure.

This function extracts a model from the set of plausible models produced by [‘multipath_aic()‘], refits the logistic regression on the original data, and evaluates training-data performance at a user-specified cutoff. These metrics are not an estimate of held-out predictive performance.

Usage

```
confusion_metrics(result_object, model_index = 1, cutoff = 0.5, verbose = TRUE)
```

Arguments

`result_object`	The full output list returned by [‘multipath_aic()‘]. Must contain ‘forest‘ and ‘plaus‘.
`model_index`	Integer index specifying which plausible model to evaluate. Default is ‘1‘ (the top plausible model).
`cutoff`	Probability threshold for classification. Default: ‘0.5‘.
`verbose`	Logical; if TRUE, prints the confusion matrix and metrics.

Details

The metrics computed include:

- **Prevalence** — proportion of positive outcomes.
- **Accuracy** — proportion of correct classifications.
- **Sensitivity (Recall)** — \(TP / (TP + FN)\).
- **Specificity** — \(TN / (TN + FP)\).
- **False Discovery Rate (FDR)** — \(FP / (TP + FP)\).
- **Diagnostic Odds Ratio (DOR)** — \((TP/FN) / (FP/TN)\).

Intercept-only models are supported. Undefined ratios are NA. The diagnostic odds ratio is infinite when its denominator is zero and numerator is positive, and NA when both are zero.

Value

A data frame containing:

Prevalence Proportion of observations in the positive class.

Accuracy Correct classification rate.

Sensitivity True positive rate.

Specificity True negative rate.

FDR False discovery rate.

DOR Diagnostic odds ratio.

Examples

```
## Logistic Model Evaluation Example
set.seed(1)
n <- 150; p <- 6
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
eta <- 1.5*X$x1 - 2*X$x2 + 1*X$x5
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, 1, prob)

result <- multipath_aic(
  X, y,
  family = "binomial",
  K = 6,
  B = 20,
  verbose = FALSE
)

# Evaluate first plausible model
confusion_metrics(result, model_index = 1)
```

launch_app *Launch Interactive Shiny App*

Description

Launches an interactive Shiny application to explore the multi-path AIC selection procedure with visualizations.

Usage

```
launch_app()
```

Examples

```
## Not run:
launch_app()

## End(Not run)
```

multipath_aic *Overall Multi-Path AIC Selection Procedure*

Description

This function implements the Multi-Path AIC Search Framework. It combines the three main components:

1. **Multi-Path Forward Selection** (‘build_paths()‘)
2. **Stability Estimation via Resampling** (‘stability()‘)
3. **Plausible Model Selection** (‘plausible_models()‘)

The output contains all intermediate results and the final set of plausible, stable models, along with the retained-model search record and variable stability summaries.

Usage

```
multipath_aic(
  X,
  y,
  family = c("gaussian", "binomial"),
  K = min(ncol(X), 10),
  eps = 1e-06,
  delta = 2,
  L = 100,
  B = 100,
```

```
  resample_fraction = 0.8,
  Delta = 2,
  tau = 0.6,
  verbose = TRUE
)
```

Arguments

X	Predictor matrix or data frame.
y	Response variable (numeric for Gaussian, binary 0/1 for Binomial).
family	Model family: ‘"gaussian"‘ or ‘"binomial"‘.
K	Maximum number of forward selection steps. Default: `min(ncol(X), 10)`.
eps	Minimum AIC improvement required for a model expansion.
delta	AIC "near-tie" threshold for branching.
L	Maximum number of child models retained at each step.
B	Number of resamples in stability estimation.
resample_fraction	Fraction of rows sampled at each stability iteration.
Delta	AIC tolerance for selecting plausible models.
tau	Stability threshold for filtering plausible models.
verbose	Logical; if TRUE, prints progress summaries.

Details

The procedure works as follows:

Step 1: Multi-Path Forward Selection Generates a branching set of candidate models using AIC-based exploration.

Step 2: Stability Estimation Evaluates how frequently each variable is selected across \(B\) resamples.

Step 3: Plausible Model Selection Combines AIC plausibility and stability screening to identify a final set of stable and competitive models.

Step 4: Compile and Return Results Outputs a structured object containing the full path search, stability scores, and the final plausible model set.

Value

A list with the following components:

forest Output of `build_paths()`.

stab Output of `stability()`.

plaus Output of `plausible_models()`.

Examples

```
## Gaussian Model Example
set.seed(123)
n <- 120; p <- 6
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
beta <- c(2, -1, 0, 0, 1, 0)
y <- as.numeric(as.matrix(X) %*% beta + rnorm(n))

result <- multipath_aic(
  X, y,
  family = "gaussian",
  K = 6,
  B = 20,
  Delta = 2,
  tau = 0.6,
  verbose = FALSE
)

## Inspect plausible models
result$plaus$summary

## Logistic Model Example
set.seed(1)
n <- 150; p <- 6
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
eta <- 1.5*X$x1 - 2*X$x2 + 1*X$x5
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, 1, prob)

result <- multipath_aic(
  X, y,
  family = "binomial",
  K = 6,
  B = 20,
  verbose = FALSE
)
```

---

plausible_models *Select Plausible and Stable Models from Multi-Path AIC Search*

Description

This function implements *Algorithm 3* of the multi-path AIC procedure. It filters the full model set created by ‘build_paths()‘ using two criteria:

1. **AIC Plausibility**: Keep all models whose AIC is within Δ of the best AIC.

2. **Stability Threshold** (optional): If variable stability scores are supplied (from ‘stability()‘), then models whose average stability is below τ are removed.

The function also computes variable inclusion probabilities across all remaining plausible models.

Usage

```
plausible_models(forest, pi = NULL, Delta = 2, tau = 0.6, verbose = TRUE)
```

Arguments

`forest`	A result object from ‘build_paths()‘, containing ‘all_models‘.
`pi`	Optional named vector of per-variable stability values (from ‘stability()$pi‘). Default: ‘NULL‘.
`Delta`	AIC tolerance Δ defining the plausible region. Models with AIC less than or equal to (best AIC + Delta) are kept. Default: ‘2‘.
`tau`	Minimum average model stability τ required to retain a model (if stability scores supplied). Default: ‘0.6‘.
`verbose`	Logical. If TRUE, prints detailed progress and summaries. Default: TRUE.

Details

After AIC filtering (Steps 1–3), the function optionally applies a stability filter (Steps 4–5). For each plausible model, the "average stability" is computed as:

$$\text{avg_stability}(M) = \frac{1}{|M|} \sum_{v \in M} \pi_v$$

If ‘pi = NULL‘, the stability-based filtering is skipped.

The function also calculates variable inclusion probabilities across all final plausible models (Step 6). Variables appearing in all models are marked with no indicator; those appearing in only a subset are marked with ‘"(-)"‘.

The threshold applies to the mean variable score in each model, not to every variable individually. Intercept-only models have undefined (NA) average stability and are exempt from this filter. Inclusion values are unweighted fractions of retained models, not posterior probabilities. Delta is a relative AIC screening tolerance, not a test of statistical equivalence.

Value

A list containing:

plausible_models Data frame of plausible (and optionally stable) models.

inclusion Named vector of variable inclusion probabilities.

summary Data frame combining stability and inclusion probabilities.

Delta The AIC tolerance used.

tau The stability threshold used.

best_aic The best (minimum) AIC observed.

Examples

```
## Complete pipeline example
set.seed(123)
n <- 120; p <- 6
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
beta <- c(2, -1, 0, 0, 1, 0)
y <- as.numeric(as.matrix(X) %*% beta + rnorm(n))

# Algorithm 1: multi-path forward selection
forest <- build_paths(X, y, family = "gaussian", K = 6, verbose = FALSE)

# Algorithm 2: variable stability
stab <- stability(X, y, family = "gaussian", B = 20, verbose = FALSE)

# Algorithm 3: plausible models
plaus <- plausible_models(
  forest = forest,
  pi = stab$pi,
  Delta = 2,
  tau = 0.6,
  verbose = TRUE
)
```

`stability` *Stability Estimation via Resampled Multi-Path Forward Selection*

Description

Computes variable-selection stability by repeatedly resampling the data and applying the 'build_paths()' algorithm.

For each bootstrap replicate (sampling rows with replacement), the function runs the multi-path forward-selection search, extracts the variables appearing in the final step of the search path, and computes the proportion of models containing each variable (z_j^(b)). The stability score is the average of these proportions across successful resamples. This is the mean within-frontier inclusion fraction, not the fraction of resamples selecting a variable at least once.

Stability is defined as:

$$\pi_j = \frac{1}{B} \sum_{b=1}^{B} z_j^{(b)}$$

where z_j^(b) is the proportion of models in resample b that contain variable j.

Usage

```
stability(
  X,
  y,
```

```
  family = c("gaussian", "binomial"),
  K = min(ncol(X), 10),
  eps = 1e-06,
  delta = 2,
  L = 100,
  B = 100,
  resample_fraction = 0.8,
  verbose = TRUE
)
```

Arguments

X	A data frame or matrix of predictors.
y	A numeric outcome vector for 'gaussian', or 0/1 for 'binomial'.
family	Model family, either '"gaussian"' (linear) or '"binomial"' (logistic).
K	Maximum number of forward-selection steps in each resample. Default: `min(ncol(X), 10)`.
eps	Minimum AIC improvement required for a parent model to expand. Default: '1e-6'.
delta	AIC tolerance controlling which near-best models are kept. Default: 2.
L	Maximum number of child models kept at each step. Default: 100.
B	Number of resamples for stability estimation. Default: 100.
resample_fraction	Fraction of the dataset used per resample (sampling with replacement). Default: '0.8'.
verbose	Logical. If TRUE, prints progress for each resample. Default: TRUE.

Details

Each resampled dataset is fit using the same multi-path forward-selection procedure as in Algorithm 1. For each resample, the proportion of models containing each variable is computed (z_j^(b)), and these proportions are averaged across successful resamples to obtain the final stability scores (pi_j).

Failed resamples are excluded from the denominator and reported with a warning. All-failed runs raise an error. Valid intercept-only runs contribute zero scores.

Only the deepest retained frontier contributes in each successful replicate; earlier terminal branches are not included. B_requested records attempted draws and failures records error messages. Increasing B reduces Monte Carlo noise but does not establish variable importance or false-discovery control.

Value

A list with:

pi Named vector of stability scores, sorted decreasing.

pi_sum Running sum used to compute pi (for diagnostic purposes).

B Number of successful resamples used as the denominator. See also B_requested and failures.

call The matched function call.

Examples

```
## Linear regression example
set.seed(123)
n <- 150; p <- 10
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
beta <- c(2, -1.5, 0, 0, 1, rep(0, p-5))
y <- as.numeric(as.matrix(X) %*% beta + rnorm(n, 1))

stab <- stability(
  X, y,
  family = "gaussian",
  K = 10,
  B = 50,
  verbose = TRUE
)

## Logistic regression example
set.seed(1)
n <- 200; p <- 10
X <- as.data.frame(matrix(rnorm(n*p), n, p))
names(X) <- paste0("x", 1:p)
eta <- 1.5*X$x1 - 2*X$x2 + 1*X$x5
prob <- 1/(1 + exp(-eta))
y <- rbinom(n, 1, prob)

stab <- stability(
  X, y,
  family = "binomial",
  K = 8,
  B = 100,
  verbose = TRUE
)
```

Index

build_paths, 2

confusion_metrics, 4

launch_app, 6

multipath_aic, 6

plausible_models, 8

stability, 10