Package ‘fastbackward’

May 8, 2026

Type Package

Title Fast Backward Elimination Based on Information Criterion

Version 1.0.1

Date 2024-5-29

Maintainer Jacob Seedorff <jacob-seedorff@uiowa.edu>

URL https://github.com/JacobSeedorff21/fastbackward

BugReports https://github.com/JacobSeedorff21/fastbackward/issues

Description Performs backward elimination with similar syntax to the stepAIC() function from the 'MASS' package. A bounding algorithm is used to avoid fitting unnecessary models, making it much faster.

License Apache License (>= 2)

Encoding UTF-8

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.1

Imports MASS

NeedsCompilation no

Author Jacob Seedorff [aut, cre]

Repository CRAN

Date/Publication 2024-05-29 22:40:02 UTC

Contents

fastbackward 2

Index **4**

`fastbackward` *Performs Fast Backward Elimination by AIC*

Description

Performs backward elimination by AIC, backward elimination is performed with a bounding algorithm to make it faster.

Usage

```
fastbackward(
  object,
  scope,
  scale = 0,
  trace = 1,
  keep = NULL,
  steps = 1000,
  k = 2,
  ...
)
```

Arguments

`object`	an object representing a model of an appropriate class. This is used as the initial model in the stepwise search.
`scope`	defines the range of models examined in the stepwise search. This should be missing or a single formula. If a formula is included, all of the components on the right-hand-side of the formula are always included in the model. If missing, then only the intercept (if included) is always included in the model.
`scale`	used in the definition of the AIC statistic for selecting the models, currently only for lm and aov models (see extractAIC for details).
`trace`	if positive, information is printed during the running of `fastbackward`. Larger values may give more detailed information. If `trace` is greater than 1, then information about which variables at each step are not considered for removal due to the bounding algorithm are printed.
`keep`	a filter function whose input is a fitted model object and the associated `AIC` statistic, and whose output is arbitrary. Typically `keep` will select a subset of the components of the object and return them. The default is not to keep anything.
`steps`	the maximum number of steps to be considered. The default is 1000 (essentially as many as required). It is typically used to stop the process early.
`k`	the multiple of the number of degrees of freedom used for the penalty. Only `k = 2` gives the genuine AIC: `k = log(n)` is sometimes referred to as BIC or SBC.
`...`	any additional arguments to extractAIC.

Details

The bounding algorithm allows us to avoid fitting models that cannot possibly provide an improvement in AIC. At a high-level, the algorithm basically works by identifying important predictors whose removal from the current model cannot possibly improve upon the current AIC.

Test statistics, p-values, and confidence intervals from the final selected model are not reliable due to the selection process. Thus, it is not recommended to use these quantities.

See more details at MASS::stepAIC.

Value

The stepwise-selected model is returned, with up to two additional components. There is an "anova" component corresponding to the steps taken in the search, as well as a "keep" component if the keep= argument was supplied in the call. The "Resid. Dev" column of the analysis of deviance table refers to a constant minus twice the maximized log likelihood: it will be a deviance only in cases where a saturated model is well-defined (thus excluding lm, aov and survreg fits, for example)

See Also

MASS::stepAIC, MASS::dropterm, and extractAIC

Examples

```
# Loading fastbackward
library(fastbackward)

# Using examples provided in MASS::stepAIC, but with fastbackward instead
## aov with quine dataset
quine.hi <- aov(log(Days + 2.5) ~ .^4, MASS::quine)
quine.nxt <- update(quine.hi, . ~ . - Eth:Sex:Age:Lrn)
quine.stp <- fastbackward(quine.nxt, trace = FALSE)
quine.stp$anova

## lm with cpus dataset
cpus1 <- MASS::cpus
for(v in names(MASS::cpus)[2:7])
 cpus1[[v]] <- cut(MASS::cpus[[v]], unique(quantile(MASS::cpus[[v]])),
                   include.lowest = TRUE)
cpus0 <- cpus1[, 2:8]  # excludes names, authors' predictions
cpus.samp <- sample(1:209, 100)
cpus.lm <- lm(log10(perf) ~ ., data = cpus1[cpus.samp,2:8])
cpus.lm2 <- fastbackward(cpus.lm, trace = FALSE)
cpus.lm2$anova

## glm with bwt dataset
example(birthwt, package = "MASS")
birthwt.glm <- glm(low ~ ., family = binomial, data = bwt)
birthwt.step <- fastbackward(birthwt.glm, trace = FALSE)
birthwt.step$anova
```

Index

aov, *2*

extractAIC, *2*, *3*

fastbackward, 2

lm, *2*

MASS::dropterm, *3*
MASS::stepAIC, *3*