Package ‘crrstep’

January 15, 2026

Type Package

Title Stepwise Covariate Selection for the Fine & Gray Competing Risks Regression Model

Version 2025.1.1

Date 2025-12-31

Description Performs forward and backward stepwise regression for the proportional subdistribution hazards model in competing risks (Fine & Gray 1999). Procedure uses AIC, BIC and BICcr as selection criteria. BICcr has a penalty of k = log(n*), where n* is the number of primary events. This version includes improved handling of factors, interactions, and polynomial terms.

Depends R (>= 3.5.0)

Imports cmprsk, stats, utils

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 2)

Encoding UTF-8

LazyLoad yes

NeedsCompilation no

Repository CRAN

Author Ravi Varadhan [aut, cre],
Deborah Kuk [aut],
Leon Wang [ctb]

Maintainer Ravi Varadhan <ravi.varadhan@jhu.edu>

Date/Publication 2026-01-14 23:50:17 UTC

Contents

`crrstep-package` *Stepwise regression procedure for the Fine & Gray regression model in competing risks*

Description

Performs forward and backward stepwise regression for the Fine & Gray regression model in competing risks. Procedure uses AIC, BIC and BICcr as selection criteria. BICcr has a penalty of k = log(n*), where n* is the number of Type I events.

Details

Package:	crrstep
Type:	Package
Version:	2023.1.1
Date:	2023-08.21
License:	GPL (version 2)
LazyLoad:	yes

The package contains a singe function `crrstep`, which implements backward and forward stepwise regression for the Fine & Gray regression model. The Fine & Gray model (Fine & Gray, 1999) estimates the hazard that corresponds to the cumulative incidence function of a certain event type. Selection criteria that are can be used are: AIC, BIC and BICcr. BICcr is a selection criteria based on the work by Volinksy and Raftery in which the penalty is k = log(n*), where n* is the total number of Type I events.

Author(s)

Ravi Varadhan & Deborah Kuk.

Maintainers: Ravi Varadhan <rvaradhan@jhmi.edu>

References

Fine, J. P. and Gray, R. J. (1999). A proportional hazards model for the subdistribution of a competing risk. *Journal of the American Statistical Association*.

Volinsky, C. T. and Raftery, A. E. (2000). Bayesian information criterion for censored survival models. *Biometrics*.

Kuk, D. and Varadhan, R. (2013). Model selection in competing risks regression. *Statistics in Medicine*.

Examples

```
set.seed(123)
n <- 500
ftime <- rexp(n)
fstatus <- sample(0:2,n,replace=TRUE)
cov1 <- matrix(runif(5*n),nrow=n)
x61 <- as.factor(sample(3, size=n, rep=TRUE))
x71 <- as.factor(sample(5, size=n, rep = TRUE))
cov1 <- cbind(cov1, x61, x71)
dimnames(cov1)[[2]] <- c('x1','x2','x3','x4','x5', 'x6', 'x7')
formula1 <- ftime ~ 1 + x1 + x2 + x3 + x4 + x5 + as.factor(x6) + as.factor(x7)

crrstep(formula1, , fstatus, data = as.data.frame(cov1), direction = "backward", criterion = "BIC")
crrstep(formula1, , fstatus, data = as.data.frame(cov1), direction = "backward", criterion = "AIC")

ans2 <- crrstep(formula1, , fstatus, data = as.data.frame(cov1), direction = "forward",
failcode=2, criterion = "AIC")
ans2
```

`crrstep` *Stepwise Regression for Fine-Gray Competing Risks Model*

Description

Performs forward and backward stepwise variable selection for the Fine-Gray subdistribution hazard model in competing risks analysis. The procedure uses AIC, BIC, or BICcr as selection criteria. BICcr uses a penalty of $k = \log(d)$, where d is the number of events of interest.

Usage

```
crrstep(formula, scope.min = ~1, etype, ..., subset, data,
        direction = c("backward", "forward"),
        criterion = c("AIC", "BICcr", "BIC"),
        crr.object = FALSE, trace = TRUE, steps = 100)

## S3 method for class 'crrstep'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'crrstep'
summary(object, conf.level = 0.95, ...)

## S3 method for class 'summary.crrstep'
print(x, digits = max(3L, getOption("digits") - 3L),
        signif.stars = getOption("show.signif.stars"), ...)

## S3 method for class 'crrstep'
coef(object, ...)
```

```
## S3 method for class 'crrstep'
vcov(object, ...)

## S3 method for class 'crrstep'
confint(object, parm, level = 0.95, ...)

## S3 method for class 'crrstep'
logLik(object, ...)

## S3 method for class 'crrstep'
AIC(object, ..., k = 2)

## S3 method for class 'crrstep'
BIC(object, ...)
```

Arguments

`formula` A formula object where the left-hand side (LHS) is the failure time variable and the right-hand side (RHS) specifies the linear predictors (e.g., `time ~ x1 + x2 + x3`). Factors, interactions, and polynomial terms are supported.

`scope.min` A one-sided formula specifying the minimum model (lower scope) for backward selection or the starting model for forward selection. Default is `~1` (null model).

`etype` An integer variable denoting the event type for each observation. By default, 1 indicates the event of interest, 2 indicates competing events, and 0 indicates censoring. These can be modified using `failcode` and `cencode` arguments passed to `crr`.

`...` Additional arguments passed to `crr`. Important arguments include:

`failcode` Integer value denoting the event of interest (default is 1).

`cencode` Integer value denoting censoring (default is 0).

For `print` and `summary` methods, additional arguments are passed to `printCoefmat`.

`subset` An optional vector specifying a subset of observations to use in the analysis.

`data` A data frame containing all variables referenced in the formula and `etype`. Only complete cases are used; rows with missing values in any predictor are automatically removed.

`direction` Character string specifying the direction of stepwise search. Either `"backward"` (default) starting from the full model, or `"forward"` starting from `scope.min`.

`criterion` Character string specifying the model selection criterion:

`"AIC"` Akaike Information Criterion (default): $-2\log L + 2p$

`"BIC"` Bayesian Information Criterion: $-2\log L + p\log(n)$

`"BICcr"` BIC for competing risks: $-2\log L + p\log(d)$

where L is the partial likelihood, p is the number of parameters, n is the sample size, and d is the number of events of interest.

`crr.object` Logical. If `TRUE`, returns the `crr` object from the final model instead of a `crrstep` object. Default is `FALSE`.

`trace` Logical. If `TRUE` (default), displays the stepwise model selection process including criterion values at each step.

`steps` Maximum number of steps allowed in the stepwise selection. Default is 100.

`x` An object of class `"crrstep"` or `"summary.crrstep"`.

`object` An object of class `"crrstep"`.

`digits` Integer specifying the number of significant digits to display.

`conf.level` Numeric value specifying the confidence level for confidence intervals. Default is 0.95.

`signif.stars` Logical. If `TRUE`, significance stars are printed alongside p-values.

`parm` A specification of which parameters to compute confidence intervals for. Can be a vector of numbers (indices) or names. If missing, all parameters are included.

`level` The confidence level for `confint`. Default is 0.95.

`k` Numeric penalty parameter for `AIC`. Default is 2.

Details

This function performs stepwise model selection for the Fine-Gray competing risks regression model implemented in `crr`. The implementation is based on the approach of `stepAIC` from the **MASS** package, adapted for the competing risks setting.

The function properly handles:

- Factor variables (automatically expanded to dummy variables)
- Interaction terms (e.g., `x1:x2`, `x1*x2`)
- Polynomial terms (e.g., `I(x^2)`)
- Complex terms (e.g., `I(x^2):factor`)

The BICcr criterion, proposed by Volinsky and Raftery (2000) and adapted for competing risks by Kuk and Varadhan (2013), uses the number of events of interest rather than the total sample size in the penalty term. This is appropriate because the effective sample size in survival analysis is determined by the number of events, not the total number of observations.

Missing values are handled by complete case analysis. Observations with missing values in any predictor variable are removed before model fitting, and a message is displayed if `trace = TRUE`.

Value

If `crr.object = TRUE`, returns the `crr` object from the final selected model.

Otherwise, returns an object of class `"crrstep"` with the following components:

`call` The matched function call.

`formula` The formula for the final selected model.

`direction` The direction of stepwise selection used.

`criterion` The selection criterion used.

`criterion_value` The value of the selection criterion for the final model.

`coefficients` Named numeric vector of estimated coefficients.

`std.error` Standard errors of the coefficients.

`zvalue` Wald z-statistics (`coefficients` / `std.error`).

`pvalue` Two-sided p-values based on the normal distribution.

`conf.int` Matrix of confidence intervals for the coefficients.

`conf.level` The confidence level used for `conf.int`.

`loglik` Log partial likelihood of the final model.

`loglik_null` Log partial likelihood of the null model.

`n` Number of observations used.

`nevent` Number of events of interest.

`converged` Logical indicating whether the `crr` algorithm converged.

`is_null_model` Logical indicating whether the final model is the null model.

`crr_fit` The `crr` object from the final model.

The `summary` method returns an object of class `"summary.crrstep"` containing:

`call` The matched function call.

`formula` The formula for the final selected model.

`direction` The direction of stepwise selection.

`criterion` The selection criterion used.

`criterion_value`
The criterion value for the final model.

`coefficients` Formatted coefficient table with estimates, hazard ratios, standard errors, z-values, and p-values.

`conf.int` Confidence intervals for hazard ratios.

`conf.level` The confidence level used.

`loglik` Log partial likelihood of the final model.

`loglik_null` Log partial likelihood of the null model.

`lr_test` List containing the likelihood ratio test statistic, degrees of freedom, and p-value.

`n` Number of observations.

`nevent` Number of events of interest.

`converged` Logical indicating convergence.

`is_null_model` Logical indicating if the null model was selected.

Author(s)

Ravi Varadhan and Deborah Kuk

References

Fine, J. P. and Gray, R. J. (1999). A proportional hazards model for the subdistribution of a competing risk. *Journal of the American Statistical Association*, 94(446), 496-509.

Volinsky, C. T. and Raftery, A. E. (2000). Bayesian information criterion for censored survival models. *Biometrics*, 56(1), 256-262.

Kuk, D. and Varadhan, R. (2013). Model selection in competing risks regression. *Statistics in Medicine*, 32(18), 3077-3088.

See Also

`crr` for the underlying Fine-Gray model, `stepAIC` for stepwise selection in other model classes, `AIC`, `BIC` for information criteria

Examples

```
# Load required package
library(cmprsk)

# Simulate example data
set.seed(123)
n <- 500
ftime <- rexp(n)
fstatus <- sample(0:2, n, replace = TRUE)
cov1 <- matrix(runif(5 * n), nrow = n)
x6 <- as.factor(sample(3, size = n, replace = TRUE))
x7 <- as.factor(sample(5, size = n, replace = TRUE))
cov1 <- cbind(cov1, x6, x7)
dimnames(cov1)[[2]] <- c('x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7')
mydata <- as.data.frame(cov1)
mydata$ftime <- ftime
mydata$fstatus <- fstatus

# Define formula with factors
formula1 <- ftime ~ x1 + x2 + x3 + x4 + x5 + as.factor(x6) + as.factor(x7)

# Backward selection with BIC
fit1 <- crrstep(formula1, etype = fstatus, data = mydata,
                direction = "backward", criterion = "BIC")
print(fit1)

# Forward selection with AIC (event of interest = 2)
fit2 <- crrstep(formula1, etype = fstatus, data = mydata,
                direction = "forward", criterion = "AIC",
                failcode = 2, trace = FALSE)
print(fit2)
summary(fit2)

# Extract model components
coef(fit2)
confint(fit2)
confint(fit2, level = 0.90)
```

```
AIC(fit2)
BIC(fit2)
logLik(fit2)
vcov(fit2)

# Model with interactions
formula2 <- ftime ~ x1 + x2 + x3 + x4 + x4:as.factor(x6) + x5:as.factor(x7)
fit3 <- crrstep(formula2, etype = fstatus, data = mydata,
                direction = "backward", criterion = "AIC",
                trace = FALSE)
print(fit3)
summary(fit3)

# Return crr object for further analysis
fit_crr <- crrstep(formula1, etype = fstatus, data = mydata,
                   direction = "backward", criterion = "BICcr",
                   crr.object = TRUE, trace = FALSE)
```

Index