Package 'aftPenCDA'

April 7, 2026

Title Penalized AFT Estimation via Coordinate Descent

Version 0.1.0

Maintainer Suyeon Seon <seonsy429@gmail.com>

Description Provides penalized accelerated failure time (AFT) model estimation for right-censored and partly interval-censored survival data using induced smoothing and coordinate descent algorithms. Supported penalties include broken adaptive ridge (BAR), LASSO, adaptive LASSO, and SCAD. Core estimation routines are implemented in 'C++' via 'Rcpp' and 'RcppArmadillo' for computational efficiency. The methodology is related to Zeng and Lin (2008) <doi:10.1093/biostatistics/kxm034>, Xu et al. (2010) <doi:10.1002/sim.2576>, Dai et al. (2018) <doi:10.1016/j.jmva.2018.08.007>, and Choi et al. (2025) <doi:10.48550/arXiv.2503.11268>.

URL https://github.com/seonsy/aftPenCDA

BugReports https://github.com/seonsy/aftPenCDA/issues

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Suyeon Seon [aut, cre],
Taehwa Choi [aut],
Dipankar Bandyopadhyay [aut],
Dongha Kim [aut],
Seongoh Park [aut]

Repository CRAN

Date/Publication 2026-04-07 07:40:21 UTC

Contents

`aftpen` *Penalized AFT estimation for right-censored data*

Description

Fits a penalized accelerated failure time (AFT) model for right-censored survival data using induced smoothing and a penalized coordinate descent algorithm. Supported penalties include BAR, LASSO, adaptive LASSO, and SCAD.

Usage

```
aftpen(
  dt,
  lambda,
  se,
  type = c("BAR", "LASSO", "ALASSO", "SCAD"),
  r = 3.7,
  eps = 1e-08,
  max.iter = 100
)
```

Arguments

`dt`	A data frame whose first two columns are `y` and `d`, where `y` is the observed survival or censoring time and `d` is the event indicator (`1` for observed event and `0` for right-censoring). The remaining columns are treated as covariates.
`lambda`	A nonnegative tuning parameter controlling the amount of penalization.
`se`	A character string specifying the variance estimation method. **`"CF"`** Closed-form (plug-in) variance estimator based on the analytic expression of the estimating function (see Equation (6)). **`"ZL"`** Perturbation-based variance estimator using the resampling approach of Zeng and Lin (2008).
`type`	Penalty type. One of `"BAR"`, `"LASSO"`, `"ALASSO"`, or `"SCAD"`.
`r`	A positive tuning constant used in the SCAD penalty. Ignored unless `type = "SCAD"`. The default is `3.7`.
`eps`	Convergence tolerance for the outer penalized coordinate descent iterations. The default is `1e-8`.
`max.iter`	Maximum number of iterations for the outer penalized coordinate descent algorithm. The default is `100`.

Details

The function first calls the Rcpp backend `is_aft_cpp()` to obtain an initial estimator together with gradient and Hessian information. A Cholesky-based transformation is then applied, followed by coordinate-wise penalized updates.

For `type = "BAR"`, the update uses the internal `BAR_threshold()` operator. For `type = "LASSO"`, `"ALASSO"`, and `"SCAD"`, soft-thresholding-based updates are used.

Value

A list containing the following components:

- `beta`: final coefficient estimate on the original scale.

Examples

```
n = 100
p = 10
beta0 = c(rep(1,3),rep(0,7))
x = matrix(rnorm(n * p), n, p)
T = exp(x%*%beta0 + rnorm(n))
C = rexp(n, rate = exp(-2))
d = 1*(T<C)
y = pmin(T,C)
dt = data.frame(y,d,x)
fit <- aftpen(dt, lambda = 0.1, se = "CF", type = "BAR")
fit$beta
```

`aftpen_pic` *Penalized AFT estimation for clustered-partly interval censored data*

Description

Fits a penalized accelerated failure time (AFT) model for partly interval censored survival data using induced smoothing and a penalized coordinate descent algorithm. Supported penalties include BAR, LASSO, adaptive LASSO, and SCAD.

Usage

```
aftpen_pic(
  dt,
  lambda,
  se,
  type = c("BAR", "LASSO", "ALASSO", "SCAD"),
  r = 3.7,
  eps = 1e-08,
  max.iter = 100
)
```

Arguments

dt	A data frame containing PIC survival data. It must include L, R, delta, and id, where L and R define the observation interval, delta indicates whether the failure time is exactly observed (1) or censored (0), and id is the cluster identifier. The remaining columns are treated as covariates.
lambda	A nonnegative tuning parameter controlling the amount of penalization.
se	A character string specifying the variance estimation method.
	"CF" Closed-form (analytic plug-in) variance estimator based on the estimating function.
	"ZL" Perturbation-resampling variance estimator following Zeng and Lin (2008).
type	Penalty type. One of "BAR", "LASSO", "ALASSO", or "SCAD".
r	A positive tuning constant used in the SCAD penalty. Ignored unless type = "SCAD". The default is 3.7.
eps	Convergence tolerance for the outer penalized coordinate descent iterations. The default is 1e-8.
max.iter	Maximum number of iterations for the outer penalized coordinate descent algorithm. The default is 100.

Details

The input data dt are assumed to arise from clustered partly interval-censored survival data with informative cluster sizes.

Specifically, observations are grouped into clusters, where each cluster shares a latent frailty variable that affects both the failure times and the cluster size. As a result, the number of observations within each cluster is not fixed but depends on the underlying frailty, leading to an informative cluster size structure.

For each subject, the failure time follows an accelerated failure time (AFT) model, and the observed data consist of an interval (L, R) together with an indicator delta. When $L = R$ (i.e., delta = 1), the observation is exact; otherwise (delta = 0), the observation is censored and may correspond to left-censoring, right-censoring, or interval-censoring depending on the relationship between the true failure time and the inspection times.

The function first calls the Rcpp backend is_aftp_pic_cpp() to obtain an initial estimator together with gradient and Hessian information. A Cholesky-based transformation is then applied, followed by coordinate-wise penalized updates.

For type = "BAR", the update uses the internal BAR_threshold() operator. For "LASSO", "ALASSO", and "SCAD", soft-thresholding-based updates are used.

Value

A list containing the following components:

- beta: final coefficient estimate on the original scale.

Examples

```
set.seed(1)

## simplified generator for clustered partly interval-censored data
n <- 50
p <- 2
beta0 <- c(1, 1)
clu_rate <- 0.5
exactrates <- 0.8
left <- 0.001
right <- 0.01

## cluster-level frailty and informative cluster sizes
eta <- 1 / clu_rate
v <- rgamma(n, shape = eta, rate = eta)
m <- ifelse(v > median(v), 5, 3)
id <- rep(seq_len(n), m)
vi <- rep(v, m)

## subject-level covariates and failure times
N <- sum(m)
x <- matrix(rnorm(N * p), ncol = p)
T <- as.vector(exp(x %*% beta0 + vi * log(rexp(N))))

## build (L, R, delta)
L <- R <- delta <- numeric(N)
index <- rbinom(N, 1, exactrates)

for (i in seq_len(N)) {
  if (index[i] == 1) {
    L[i] <- T[i]
    R[i] <- T[i]
    delta[i] <- 1
  } else {
    U <- cumsum(c(1e-8, runif(10, left, right)))
    LL <- U[-length(U)]
    RR <- U[-1]

    if (T[i] < min(LL)) {
      L[i] <- 1e-8
      R[i] <- min(LL)
      delta[i] <- 0
    } else if (T[i] > max(RR)) {
      L[i] <- max(RR)
      R[i] <- 1e8
      delta[i] <- 0
    } else {
      idd <- which(T[i] > LL & T[i] < RR)
      if (length(idd) == 1) {
        L[i] <- LL[idd]
        R[i] <- RR[idd]
        delta[i] <- 0
```

```
      } else {
        L[i] <- T[i]
        R[i] <- T[i]
        delta[i] <- 1
      }
    }
  }
}

dt <- data.frame(
  L = L, R = R, delta = delta, id = id,
  x1 = x[, 1], x2 = x[, 2]
)

fit <- aftpen_pic(dt, lambda = 0.001, se = "CF", type = "BAR")
fit$beta
```

Index