

Package ‘DRLAP2’

August 25, 2026

Type Package

Title Dynamic Reinforcement Learning and Adaptive Progressive
Censoring

Version 0.1.1

Description Implements Maximum Likelihood Estimation (MLE) and Bayesian Markov Chain Monte Carlo (MCMC) sampling algorithms for progressive censoring models, with support for dynamic reinforcement learning environment simulation and accelerated computational routines written in C++.

License GPL (>= 3)

Encoding UTF-8

Imports Rcpp, stats

LinkingTo Rcpp

NeedsCompilation yes

Config/roxygen2/version 8.1.0

Author Okechukwu J. Obulezi [aut, cre]

Maintainer Okechukwu J. Obulezi <oj.obulezi@unizik.edu.ng>

Repository CRAN

Date/Publication 2026-08-25 15:10:16 UTC

Contents

calc_reward	2
compute_drl_sums_cpp	2
drl_log_target_beta_cpp	3
fit_drl_bayes	3
fit_drl_mle	4
init_drl_env	5

Index	6
--------------	----------

calc_reward	<i>Calculate Reward Signal</i>
-------------	--------------------------------

Description

Calculate Reward Signal

Usage

```
calc_reward(t_i, r_i, lambda = 0.5)
```

Arguments

t_i	Observed failure time
r_i	Chosen removal action
lambda	Cost penalty scaling factor

Value

A numeric scalar representing the computed net reward signal (information gain minus action removal cost).

compute_drl_sums_cpp	<i>Compute DRL Sums (C++)</i>
----------------------	-------------------------------

Description

Computes internal sum terms S0, S1, and S2 used in DRL parameter estimation.

Usage

```
compute_drl_sums_cpp(t, R, beta)
```

Arguments

t	Numeric vector of positive time or survival values.
R	Numeric vector of response or censoring counts.
beta	Double specifying the beta shape parameter.

Value

A named list containing S0, S1, and S2.

Examples

```
t_vals <- c(1.2, 2.5, 3.1)
R_vals <- c(0, 1, 0)
compute_drl_sums_cpp(t = t_vals, R = R_vals, beta = 1.5)
```

drl_log_target_beta_cpp

Compute Log Target Density for Beta (C++)

Description

Evaluates the conditional log-target distribution for the beta parameter in DRL models.

Usage

```
drl_log_target_beta_cpp(beta, theta, t, R, a1, b1)
```

Arguments

beta	Double specifying the current beta parameter (must be positive).
theta	Double specifying the current theta scale parameter.
t	Numeric vector of positive time values.
R	Numeric vector of response counts.
a1	Double specifying shape hyperparameter a1.
b1	Double specifying rate hyperparameter b1.

Value

Double representing the evaluated log-target value.

Examples

```
t_vals <- c(1.2, 2.5, 3.1)
R_vals <- c(0, 1, 0)
drl_log_target_beta_cpp(beta = 1.5, theta = 0.8, t = t_vals, R = R_vals, a1 = 1.0, b1 = 1.0)
```

fit_drl_bayes

Bayesian MCMC Sampler for DRL-AP2 Weibull Model

Description

Bayesian MCMC Sampler for DRL-AP2 Weibull Model

Usage

```
fit_drl_bayes(
  t,
  R,
  priors = list(a1 = 0.1, b1 = 0.1, a2 = 0.1, b2 = 0.1),
  N_MC = 10000,
  N_burn = 2000,
  proposal_sd = 0.05
)
```

Arguments

t	Numeric vector of failure times
R	Numeric vector of removal actions
priors	List containing prior hyperparameters a1, b1, a2, b2
N_MC	Total MCMC draws
N_burn	Burn-in draws to discard
proposal_sd	Standard deviation for Gaussian candidate proposal on beta

Value

An object of S3 class "drl_bayes", which is a list containing:

bayes_estimates	A named numeric vector of posterior mean point estimates for parameters alpha and beta.
hpd_alpha	A named numeric vector of length 2 giving the 95% equal-tailed credible interval limits for alpha.
hpd_beta	A named numeric vector of length 2 giving the 95% equal-tailed credible interval limits for beta.
chains	A data.frame containing post-burn-in MCMC draws for alpha and beta.

fit_drl_mle	<i>Maximum Likelihood Estimation for DRL-AP2 Weibull Model (C++ Accelerated)</i>
-------------	--

Description

Maximum Likelihood Estimation for DRL-AP2 Weibull Model (C++ Accelerated)

Usage

```
fit_drl_mle(t, R, alpha = 0.05, max_iter = 100, tol = 1e-07)
```

Arguments

t	Numeric vector of failure times
R	Numeric vector of progressive removal actions
alpha	Significance level (default = 0.05)
max_iter	Maximum Newton-Raphson iterations
tol	Convergence threshold

Value

A list containing point estimates and asymptotic confidence intervals

init_drl_env	<i>Initialize MDP State Observer for Dynamic Censoring</i>
--------------	--

Description

Initialize MDP State Observer for Dynamic Censoring

Usage

```
init_drl_env(n, m, budget)
```

Arguments

n	Total initial units at start of experiment
m	Required total number of failure observations
budget	Initial operational budget

Value

An object of class DRLState

Index

`calc_reward`, [2](#)

`compute_drl_sums_cpp`, [2](#)

`drl_log_target_beta_cpp`, [3](#)

`fit_drl_bayes`, [3](#)

`fit_drl_mle`, [4](#)

`init_drl_env`, [5](#)