Package 'SAFEPG'

May 9, 2025

Type Package

Title A Novel SAFE Model for Predicting Climate-Related Extreme Losses

Version 0.0.1

Date 2025-05-01

Maintainer Qian Tang <qian-tang@uiowa.edu>

Description The goal of 'SAFEPG' is to predict climate-related extreme losses by fitting a frequency-severity model. It improves predictive performance by introducing a sign-aligned regularization term, which ensures consistent signs for the coefficients across the frequency and severity components. This enhancement not only increases model accuracy but also enhances its interpretability, making it more suitable for practical applications in risk assessment.

Depends R (>= 3.5.0)

License GPL-2

Imports stats, Matrix, methods

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation yes

Author Qian Tang [aut, cre],
Yikai Zhang [aut],
Boxiang Wang [aut]

Repository CRAN

Date/Publication 2025-05-09 09:40:02 UTC

Contents

coef.safe *Extract Model Coefficients from a ‘safe‘ Object*

Description

Retrieves the coefficients at specified values of ‘lambda‘ from a fitted ‘safe()‘ model.

Usage

```
## S3 method for class 'safe'
coef(object, s = NULL, ...)
```

Arguments

object	Fitted ‘safe()‘ object.
s	Values of the penalty parameter ‘lambda‘ for which coefficients are requested. Defaults to the entire sequence used during the model fit.
...	Not used.

Details

This function extracts coefficients for specified ‘lambda‘ values from a ‘safe()‘ object. If ‘s‘, the vector of ‘lambda‘ values, contains values not originally used in the model fitting, the ‘coef‘ function employs linear interpolation between the closest ‘lambda‘ values from the original sequence to estimate coefficients at the new ‘lambda‘ values.

Value

Returns a matrix or vector of coefficients corresponding to the specified ‘lambda‘ values.

See Also

safe, predict.safe

Examples

```
set.seed(1)
n <- 100
p <- 5
x <- matrix(rnorm(n * p), nrow = n, ncol = p)
beta_true <- rep(0.1, 5)
gamma_true <- c(rep(1, 3), -1, -1)
mu <- x %*% beta_true
k <- rpois(n, lambda = exp(mu))
alpha_val <- 1
theta <- exp(x %*% gamma_true) / alpha_val
y <- rgamma(n, shape = alpha_val, scale = theta)
lambda_val <- 1
```

```
fit <- safe(x, y, k, 1, ind_p = c(1, 1, 1, 0, 0))
coef(fit)
```

eccv.safe	*Cross-validation for selecting the tuning parameter of the SAFE model*

Description

Performs "electoral college" cross-validation for the `safe` function. Unlike a standard cross-validation approach, this method repeats the random partitioning into folds multiple times (controlled by `rep`), then selects the tuning parameter `lambda` via a majority vote across these repeated cross-validation runs. This function is largely similar [glmnet::cv.glmnet()].

Usage

```
eccv.safe(x, y, k, lambda, ind_p, rep = 24, nfolds = 5L, ...)
```

Arguments

`x`	A numeric matrix of dimensions $n \times p$, where n is the number of observations and p is the number of predictors.
`y`	A numeric vector of length n, representing the loss values (severity). These are assumed to follow a Gamma distribution with shape parameter `alpha` and scale parameter $\theta = \exp(x^\top \gamma)/\alpha$.
`k`	A numeric vector of length n, representing the number of claims (frequency), assumed to follow a Poisson distribution with mean $\mu = x^\top \beta$.
`lambda`	A user-supplied numeric vector of tuning parameters. The function will compute the solution for each value in `lambda`.
`ind_p`	A user-provided index or indicator specifying which predictors should share the same sign across frequency and severity.
`rep`	The number of repeated cross-validation cycles (default is 24). In each cycle, observations are randomly assigned to `nfolds` folds, cross-validation is performed for all `lambda` values, and a "vote" is cast for the `lambda` with the lowest error in that cycle.
`nfolds`	The number of folds in cross-validation. Default is 5.
`...`	Additional arguments passed to `safe`.

Details

The function computes the average cross-validation error and reports the best `lambda` that achieves the smallest cross-validation error.

Value

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

`lambda.min`	The tuning parameter `lambda` value that won the most votes (i.e., achieved the minimum cross-validation error most often).
`lambda`	The full sequence of candidate `lambda` values provided to the function.
`ncvmat`	A numeric matrix of dimension `rep` * the length of `lambda`, containing the cross-validation errors for each `lambda` across all `rep` runs.

Examples

```
set.seed(1)
n <- 100
p <- 5
x <- matrix(rnorm(n * p), nrow = n, ncol = p)
beta_true <- rep(0.1, 5)
gamma_true <- c(rep(1, 3), -1, -1)
mu <- x %*% beta_true
k <- rpois(n, lambda = exp(mu))
alpha_val <- 1
theta <- exp(x %*% gamma_true) / alpha_val
y <- rgamma(n, shape = alpha_val, scale = theta)
lambda_seq <- 10^seq(2, -8, length.out = 5)
fit <- eccv.safe(x, y, k, lambda=lambda_seq, ind_p = c(1, 1, 1, 0, 0))
```

`predict.safe` *Make Predictions from a ‘safe‘ Object*

Description

Produces fitted values for new predictor data using a fitted ‘safe()‘ object.

Usage

```
## S3 method for class 'safe'
predict(object, newx, s = NULL, ...)
```

Arguments

`object`	Fitted ‘safe()‘ object from which predictions are to be derived.
`newx`	Matrix of new predictor values for which predictions are desired. This must be a matrix and is a required argument.
`s`	Values of the penalty parameter ‘lambda‘ for which predictions are requested. Defaults to the entire sequence used during the model fit.
`...`	Not used.

Details

This function generates predictions at specified ‘lambda‘ values from a fitted ‘safe()‘ object. It is essential to provide a new matrix of predictor values (‘newx‘) at which these predictions are to be made.

Value

Returns the fitted values.

See Also

`safe`, `coef.safe`

Examples

```
set.seed(1)
n <- 100
p <- 5
x <- matrix(rnorm(n * p), nrow = n, ncol = p)
beta_true <- rep(0.1, 5)
gamma_true <- c(rep(1, 3), -1, -1)
mu <- x %*% beta_true
k <- rpois(n, lambda = exp(mu))
alpha_val <- 1
theta <- exp(x %*% gamma_true) / alpha_val
y <- rgamma(n, shape = alpha_val, scale = theta)
lambda_val <- 1
fit <- safe(x, y, k, 1, ind_p = c(1, 1, 1, 0, 0))
set.seed(234)
newx <- matrix(rnorm(n * p), nrow = n, ncol = p)
predict(fit, newx)
```

`safe` *Solve the sign-aligned frequency-severity (SAFE) model*

Description

This function fits a Poisson-Gamma regression framework that aligns the signs of certain predictors in both the Poisson frequency component and the Gamma severity component. The solution path is computed at a sequence of tuning parameter values (`lambda`).

Usage

```
safe(
  x,
  y,
  k,
  lambda,
  alpha = 1,
  eps = 1e-08,
  maxit = 1e+05,
  eps2 = 1e-08,
  ind_p,
  int_gam = NULL,
  int_beta = NULL
)
```

Arguments

`x`	A numeric matrix of dimensions $n \times p$, where n is the number of observations and p is the number of predictors.
`y`	A numeric vector of length n, representing the loss values (severity). These are assumed to follow a Gamma distribution with shape parameter `alpha` and scale parameter $\theta = \exp(x^\top \gamma)/\alpha$.
`k`	A numeric vector of length n, representing the number of claims (frequency), assumed to follow a Poisson distribution with mean $\mu = x^\top \beta$.
`lambda`	A user-supplied numeric vector of tuning parameters. The function will compute the solution for each value in `lambda`.
`alpha`	The shape parameter for the Gamma distribution (default is 1).
`eps`	Convergence tolerance for updating `beta` (default is `1e-8`).
`maxit`	An integer specifying the maximum number of iterations (default is `1e5`).
`eps2`	Convergence tolerance for updating `gamma` (default is `1e-8`).
`ind_p`	A user-provided vector specifying which predictors should share the same sign across frequency and severity.
`int_gam`	Optional numeric vector or matrix providing initial values for `gamma`; defaults to `NULL`.
`int_beta`	Optional numeric vector or matrix providing initial values for `beta`; defaults to `NULL`.

Details

The function uses an **accelerated proximal gradient descent** algorithm to simultaneously estimate `beta` (for the Poisson frequency model) and `gamma` (for the Gamma severity model) under a sign-alignment constraint for selected predictors (controlled by `ind_p`).

Value

An object of class `safe`, which is a list containing:

`beta`	A $p \times L$ matrix of frequency-model coefficients, where p is the number of predictors and L is the number of `lambda` values.
`gamma`	A $p \times L$ matrix of severity-model coefficients, with the same dimensions as `beta`.

`lambda`	The (sorted) sequence of `lambda` values used in the estimation.
`npass_beta`	Total number of iterations used to update `beta` across all `lambda` values.
`npass_gamma`	Total number of iterations used to update gamma across all `lambda` values.
`jerr`	An integer flag for warnings or errors; `0` indicates no issues encountered.

Examples

```
set.seed(1)
n <- 100
p <- 5
x <- matrix(rnorm(n * p), nrow = n, ncol = p)
beta_true <- rep(0.1, 5)
gamma_true <- c(rep(1, 3), -1, -1)
mu <- x %*% beta_true
k <- rpois(n, lambda = exp(mu))
alpha_val <- 1
theta <- exp(x %*% gamma_true) / alpha_val
y <- rgamma(n, shape = alpha_val, scale = theta)
lambda_val <- 1
fit <- safe(x, y, k, 1, ind_p = c(1, 1, 1, 0, 0))
```

Index