Package ‘flassomsm’

November 26, 2025

Type Package

Title Penalized Estimation for Multi-State Models with Lasso and Fused Penalties

Version 0.1.0

Maintainer Atanu Bhattacharjee <`atanustat@gmail.com`>

Description Provides a suite of methods for detecting influential subjects in longitudinal datasets, particularly when observations occur at irregular time points. The methods identify individuals whose response trajectories deviate significantly from the population pattern, enabling detection of anomalies or subjects exerting undue influence on model outcomes.

Imports dplyr, corpcor, future, future.apply, glmnet, mstate,numDeriv, penalized, progress, progressr, survival

Suggests ggplot2, rlang, mice

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

RoxygenNote 7.3.3

NeedsCompilation no

Author Atanu Bhattacharjee [aut, cre, ctb],
Gajendra Kumar Vishwakarma [aut, ctb],
Abhipsa Tripathy [aut, ctb]

Repository CRAN

Date/Publication 2025-11-26 09:30:02 UTC

Contents

`covselec` *covselec*

Description

This function is used for state-wise weighted biomarker selection for both lasso and fused type penalty and is applicable for a multi-state model encompassing finite number of states and transitions.

Usage

```
covselec(
  data,
  time_cols,
  status_cols,
  covariate_range,
  alphas,
  lambdas,
  method = "lasso",
  p_cut = 0.2,
  verbose = TRUE
)
```

Arguments

`data`	Multi state dataset with multiple state and time column in the data
`time_cols`	time-to-event columns for each state in a MSM
`status_cols`	status columns for corresponding states in a MSM
`covariate_range`	range of high-dimensional biomarkers in the MSM dataset
`alphas`	penalty parameter for lasso type penalty
`lambdas`	penalty parameter for fused type penalty
`method`	method to be selected for covariate selction either "lasso" or "fused"
`p_cut`	association between variables
`verbose`	Logical indicating whether to print progress messages

Details

This function deals with biomarker selection for each state of a multi-state model each phase involves selecting significant biomarkers for that state from two sets: one comprising covariates from the preceding state, and another consisting of all covariates from the current state that were previously unselected.If the total number of biomarkers is referred to as the overall count, and each state in a multi-state model has its own number of selected biomarkers, then in a model with four states, the number of selected biomarkers for the fourth state is determined using a weighted combination. Specifically, it depends on the number of biomarkers selected in the third state, a modified set from the second state, another modified set from the first state

Value

list containing selected biomarkers for each state

Author(s)

Atanu Bhattacharjee,Gajendra Kumar Vishwakarma,Abhipsa Tripathy

Examples

```
##
set.seed(123)
covselec(data= highDmsm,time_cols = c("state1", "state2","state3"),
status_cols = c("status1", "status2","status3"),
covariate_range = 8:107,alphas=c(0.40,0.45,0.60),
lambdas = c(0.1,0.15,0.20),method = "lasso", p_cut = 0.2,verbose = TRUE)
##
```

`flassomsm` *flassomsm*

Description

Fits a penalized regression model with combined Fusedlasso penalty using hybrid algorithm

Usage

```
flassomsm(
  msdata,
  X,
  p,
  lambda_lasso,
  lambda_fused,
  tol_outer = 1e-04,
  max_outer = 50,
  rho = 1,
  tol_admm = 1e-04,
  max_admm = 100,
```

```
  trace = TRUE,
  MSM_profile = FALSE,
  use_parallel = TRUE
)
```

Arguments

`msdata`	is a multi-state model in extended form having columns Tstart,Tstop,trans (covariates expanded transition wise)
`X`	expanded covariate matrix of the msdata
`p`	number of covariates in the dataset before expanding
`lambda_lasso`	parameter for lasso penalty
`lambda_fused`	parameter for fused penalty
`tol_outer`	tolerence limit in the outer loop of PIRLS algorithm to converge
`max_outer`	round of iterations until tolerence is reached for the outer loop
`rho`	augmented Lagrangian parameter
`tol_admm`	tolerence limit in the inner loop of ADMM algorithm to converge
`max_admm`	round of iterations until tolerence is reached for the inner loop
`trace`	logical triggering for status information
`MSM_profile`	logical triggering to return the results
`use_parallel`	logical flag to indicate whether to use parallel processing

Details

This is the core function of the package.This function fits a penalized Cox-type regression model within the framework of a multi-state model. It is designed to handle transition-specific covariate effects across multiple states by incorporating a regularization approach that combines both the Lasso penalty and the Fused penalty. The penalization is of the L1 type, meaning it applies to the absolute values of the regression coefficients, encouraging sparsity in the model. Additionally, it penalizes the absolute differences between corresponding coefficients across different transitions, promoting similarity or grouping of effects across transitions when appropriate. This dual-penalty structure enables both variable selection and smoothing of covariate effects across related transitions, which is particularly useful in complex multi-state settings where covariate effects may share underlying patterns but still exhibit transition-specific behaviors. The parameters are estimated using a hybrid algorithm techinique combining PIRLS and ADMM together.

Value

A list with elements like matrix of estimated beta coefficients along with standard error and p value,number of iterations, aic (Akaike Information Criterion) value, gcv (GCV criterion) value and df (degrees of freedom)

Author(s)

Atanu Bhattacharjee,Gajendra Kumar Vishwakarma,Abhipsa Tripathy

Examples

```
##
set.seed(123)
data(msdata_3state)
covs1 <- msdata_3state[,9:10]
flassomsm(msdata = msdata_3state,X=msdata_3state[,c(11:dim(msdata_3state)[[2]])],
p = length(covs1),lambda_lasso = 0.3,lambda_fused = 0.5,tol_outer = 1e-4,
max_outer = 50, rho = 1, tol_admm = 1e-4, max_admm = 100,trace = TRUE,
MSM_profile = FALSE)

# For 2 covariates and 3 number of transitions


# Simulate msdata_4state instead of loading from disk
msdata_4state <- simdata(seed=123,n=1000,dist="weibull",cdist="exponential",
                 cparams=list(rate = 0.1),lambdas=c(0.1, 0.2, 0.3, 0.4),
                 gammas=c(1.5, 2, 2.5, 2.6),beta_list=list(c(-0.05, 0.01, 0.5, 0.6),
                 c(-0.03, 0.02, 0.07, 0.08),c(-0.04, 0.03, 0.04, -0.03),
                 c(-0.05,0.05,0.6,0.8)),cov_means=c(0,10,2,3),cov_sds=c(1,20,5,1.05),
                 trans_list=list(c(2, 3, 4, 5),c(3, 4, 5),c(4, 5), c(5), c()),
                 state_names=c("Tx", "Rec", "Death", "Reldeath", "srv"))

set.seed(123)
sub_msdata_4state <- msdata_4state[msdata_4state$id %in% sample(unique(msdata_4state$id), 10), ]
covs1 <- sub_msdata_4state[,9:10]
flassomsm(msdata = sub_msdata_4state,X=sub_msdata_4state[,c(13:32)],
         p = length(covs1),lambda_lasso = 0.5,lambda_fused = 0.6,tol_outer = 1e-4,
      max_outer = 50, rho = 1, tol_admm = 1e-4, max_admm = 100,trace = TRUE,MSM_profile = FALSE)

# For 2 covariates and 10 number of transitions
##
```

`flassomsm_admm` *flassomsm_admm*

Description

Fits a penalized regression model with combined Fusedlasso penalty using ADMM algorithm

Usage

```
flassomsm_admm(
  msdata,
  X,
  p,
  lambda_lasso,
  lambda_fused,
  tol_admm = 1e-04,
  max_admm = 100,
```

```
  rho = 1,
  trace = TRUE,
  MSM_profile = FALSE
)
```

Arguments

`msdata`	is a multi-state model in extended form having columns Tstart,Tstop,trans (covariates expanded transition wise)
`X`	expanded covariate matrix of the msdata
`p`	number of covariates in the dataset before expanding
`lambda_lasso`	parameter for lasso penalty
`lambda_fused`	parameter for fused penalty
`tol_admm`	tolerence limit in the algorithm to stop
`max_admm`	maximum number of iterations reached
`rho`	augmented Lagrangian parameter
`trace`	logical triggering for status information
`MSM_profile`	logical triggering to return the results

Details

This function fits a penalized Cox type regression model to a multi-state setting, where the penalty is a combination of Lasso penalty and Fused penalty. It applies L1 type penalization with the penalty applied to absolute transition-specific effects and pairwise difference between the corresponding transition effects and alternating direction method of multipliers (ADMM) algorithm.

Value

A list with elements like matrix of estimated beta coefficients along with standard error and p value,number of iterations, aic (Akaike Information Criterion) value, gcv (GCV criterion) value and df (degrees of freedom)

Author(s)

Atanu Bhattacharjee,Gajendra Kumar Vishwakarma,Abhipsa Tripathy

Examples

```
##
set.seed(123)
data(msdata_3state)
covs1 <- msdata_3state[,9:10]
flassomsm_admm(msdata = msdata_3state,X=msdata_3state[,c(11:dim(msdata_3state)[[2]])],
p = length(covs1),lambda_lasso = 0.3,lambda_fused = 0.5,tol_admm = 1e-4,max_admm = 10,
rho = 1, trace = TRUE, MSM_profile = FALSE)

# For 2 covariates and 3 number of transitions
```

```
# Simulate msdata_4state instead of loading from disk
msdata_4state <- simdata(seed=123,n=1000,dist=”weibull”,cdist=”exponential”,
                 cparams=list(rate = 0.1),lambdas=c(0.1, 0.2, 0.3, 0.4),
                 gammas=c(1.5, 2, 2.5, 2.6),beta_list=list(c(-0.05, 0.01, 0.5, 0.6),
                 c(-0.03, 0.02, 0.07, 0.08),c(-0.04, 0.03, 0.04, -0.03),
                 c(-0.05, 0.05, 0.6, 0.8)),cov_means=c(0, 10, 2, 3),
                 cov_sds=c(1, 20, 5, 1.05),trans_list=list(c(2, 3, 4, 5),
           c(3,4,5),c(4,5),c(5),c()),state_names=c(”Tx”,”Rec”,”Death”,”Reldeath”,”srv”))
set.seed(123)
sub_msdata_4state <- msdata_4state[msdata_4state$id %in% sample(unique(msdata_4state$id), 10), ]
covs1 <- sub_msdata_4state[,9:10]
flassomsm_admm(msdata = sub_msdata_4state,X=sub_msdata_4state[,c(13:32)],
              p = length(covs1),lambda_lasso = 0.5,lambda_fused = 0.6,
              tol_admm = 1e-4,max_admm = 10,rho = 1, trace = TRUE,
              MSM_profile = FALSE)

# For 2 covariates and 10 number of transitions
##
```

`flassomsm_pirls` *flassomsm_pirls*

Description

Fits a penalized regression model with combined Fusedlasso penalty using PIRLS algorithm

Usage

```
flassomsm_pirls(
  msdata,
  X,
  p,
  lambda_lasso,
  lambda_fused,
  tol_lim = 1e-04,
  max_iter = 50,
  trace = TRUE,
  MSM_profile = FALSE,
  use_parallel = TRUE
)
```

Arguments

`msdata`	is a multi-state model in extended form having columns Tstart,Tstop,trans (covariates expanded transition wise)
`X`	expanded covariate matrix of the msdata
`p`	number of covariates in the dataset before expanding

`lambda_lasso`	parameter for lasso penalty
`lambda_fused`	parameter for fused penalty
`tol_lim`	tolerence limit in the algorithm to stop
`max_iter`	maximum number of iterations reached
`trace`	logical triggering for status information
`MSM_profile`	logical triggering to return the results
`use_parallel`	logical flag to indicate whether to use parallel processing

Details

This function fits a penalized Cox type regression model to a multi-state model, where the penalty is a combination of Lasso penalty and Fused penalty. It implements L1 type penalization with the penalty applied to absolute transition-specific effects and pairwise difference between the corresponding transition effects.

Value

A list with elements like matrix of estimated beta coefficients along with standard error and p value,number of iterations, aic (Akaike Information Criterion) value, gcv (GCV criterion) value and df (degrees of freedom)

Author(s)

Atanu Bhattacharjee,Gajendra Kumar Vishwakarma,Abhipsa Tripathy

Examples

```
##
set.seed(123)
data(msdata_3state)
covs1 <- msdata_3state[,9:10]
flassomsm_pirls(msdata = msdata_3state,X=msdata_3state[,c(11:dim(msdata_3state)[[2]])],
p = length(covs1),lambda_lasso = 0.3,lambda_fused = 0.5,tol_lim = 1e-4,
max_iter = 10, trace = TRUE, MSM_profile = FALSE,use_parallel = TRUE)

# For 2 covariates and 3 number of transitions


# Simulate msdata_4state instead of loading from disk
msdata_4state <- simdata(seed=123,n=1000,dist="weibull",cdist="exponential",
                 cparams=list(rate = 0.1),lambdas=c(0.1, 0.2, 0.3, 0.4),
                 gammas=c(1.5, 2, 2.5, 2.6),beta_list=list(c(-0.05,0.01,0.5,0.6),
                 c(-0.03, 0.02, 0.07, 0.08),c(-0.04, 0.03, 0.04, -0.03),
                 c(-0.05,0.05,0.6,0.8)),cov_means=c(0, 10, 2, 3),cov_sds=c(1,20,5,1.05),
                 trans_list=list(c(2, 3, 4, 5),c(3, 4, 5),c(4, 5), c(5), c()),
                 state_names=c("Tx", "Rec", "Death", "Reldeath", "srv"))

set.seed(123)
sub_msdata_4state <- msdata_4state[msdata_4state$id %in% sample(unique(msdata_4state$id), 10), ]
```

```
covs1 <- sub_msdata_4state[,9:10]
flassomsm_pirls(msdata = sub_msdata_4state,X=sub_msdata_4state[,c(13:32)],
         p = length(covs1),lambda_lasso = 0.5,lambda_fused = 0.6, tol_lim = 1e-4,max_iter = 10,
              trace = TRUE, MSM_profile = FALSE,use_parallel = TRUE)

# For 2 covariates and 10 number of transitions
##
```

`highDmsm` *Multi state data with high dimensional covariates*

Description

Multi state data with 3 states, 100 continuous biomarkers

Usage

```
data(highDmsm)
```

Format

A dataframe with 500 rows and 107 columns including all the states and covariates

id ID of subjects

state1 Time in days from transplantation to state-1 or last follow up

status1 Status of state-1

state2 Time in days from transplantation to state-2 or last follow up

status2 Status of state-1

state3,status3 Time in days from transplant to the respective states along with its status

v1,...,v100 The 100 biomarkers attached to the dataset

Examples

```
data(highDmsm)
```

`msdata_3state` *Short Multi state data*

Description

Simulated multi-state data with 3 states expanded in msdata format

Usage

```
data(msdata_3state)
```

Format

A dataframe with 12 rows and 2 variables expanded in wide format for 3 orders of transitions

id	ID of subjects
from	From which state the individual is shifting
to	To which state the individual is shifting
trans	Order of transition of a particular individual at a specific time
Tstart	Starting time of a transition
Tstop	Stop time of a transition
time	Total time duration of a particular transition
status	Indicator of that particular transition
x1,x2	2 continuous covariates originally in the dataset
x1.1,...,x2.3	The 2 covariates expanded in 3 orders of transitions

Examples

```
data(msdata_3state)
```

`msdata_4state` *Multi state data*

Description

Simulated multi-state data with 4 states expanded in msdata format

Usage

```
data(msdata_4state)
```

Format

A dataframe with 9198 rows and 4 variables expanded in wide format for 10 orders of transitions

id ID of subjects

from From which state the individual is shifting

to To which state the individual is shifting

trans Order of transition of a particular individual at a specific time

Tstart Starting time of a transition

Tstop Stop time of a transition

time Total time duration of a particular transition

status Indicator of that particular transition

x1,...,x4 4 continuous covariates originally in the dataset

x1.1,...,x4.10 The 4 covariates expanded in 10 orders of transitions

Examples

```
data(msdata_4state)
```

`prederr` *prederr*

Description

Evaluates the predictive performance of the multi-state model

Usage

```
prederr(
  msdata,
  X,
  beta_est,
  times,
  state_of_interest,
  trans_matrix,
  test_fraction = 0.2,
  quick = FALSE,
  verbose = FALSE
)
```

Arguments

`msdata`	is a multi-state model in extended form having columns Tstart,Tstop,trans (covariates expanded transition wise)
`X`	covariate matrix of the original covariates before expanding, (for example if the dataset initially contains 4 covariates then the matrix has to be formed with the 4 covariates only and not their expanded version)
`beta_est`	estimated beta coefficients from the fitted model
`times`	time points at which prediction error is to be calculated
`state_of_interest`	the target state for which prediction accuracy is evaluated
`trans_matrix`	transition matrix initially defined for multi-state model
`test_fraction`	fraction of subjects randomly assigned to test set
`quick`	specify whether the model will run complete code or quick mode
`verbose`	Logical indicating whether to print progress messages

Details

This function is designed to evaluate the predictive performance of a multi-state survival model using Brier score using subject specific covatiates and thier estimated coefficients from a penalized regression model. It also incorporates baseline hazards for each transtion from a Cox model using msfit() for more accurate prediction and also computes the predicted state probabilities with probtrans().

Value

Gives a list of objects like brier score at specified time points, Integrated Brier score,predicted probabilities, their true states and the time points

Author(s)

Atanu Bhattacharjee,Gajendra Kumar Vishwakarma,Abhipsa Tripathy

Examples

```
##
set.seed(123)
data(msdata_3state)
tmat_3state <- mstate::transMat(x = list(c(2, 3), c(3), c()),
                        names = c("State1", "State2", "State3"))
beta_est1 <- c(0.13, -0.16, -0.12, -0.20, -0.15, -0.55)
prederr(msdata = msdata_3state, X = msdata_3state[, 9:10], beta_est = beta_est1,
times = seq(0.5, 1.5, length.out = 5),state_of_interest = 2,
trans_matrix = tmat_3state,test_fraction = 0.2,quick = TRUE,verbose = TRUE)


data(msdata_4state)
set.seed(123)
sub_msdata_4state <- msdata_4state[msdata_4state$id %in% sample(unique(msdata_4state$id), 10), ]
```

```
tmat_4state <- mstate::transMat(x = list(c(2, 3, 4,5), c(3, 4, 5), c(4,5), c(5),c()),
                                names = c("Tx", "Lrc","Fp", "Dp", "srv"))
beta_est1 <- as.numeric (c(0.13,-0.16,-0.12,-0.20,-0.15,-0.55,-0.35,-0.28,-0.34,-0.12))
times1 <- seq(0.5, 1.5, length.out = 5)
prederr(msdata = sub_msdata_4state, X = sub_msdata_4state[,9],
       beta_est = beta_est1,times = times1,state_of_interest = 2,
       trans_matrix = tmat_4state,test_fraction = 0.2,quick = TRUE,verbose = TRUE)

##
```

`simdata` *simdata*

Description

Simulation of a multi-state model with n no. of states.

Usage

```
simdata(
  seed = 123,
  n = 1000,
  dist = "weibull",
  cdist = "exponential",
  cparams = list(rate = 0.1),
  lambdas,
  gammas,
  beta_list,
  cov_means,
  cov_sds,
  trans_list,
  state_names
)
```

Arguments

`seed`	Random seed for reproducibility
`n`	Number of subjects
`dist`	distribution to follow for baseline hazard ("exponential", "weibull", "gompertz")
`cdist`	distribution to follow for censoring distribution ("uniform", "exponential", "weibull")
`cparams`	parameter vector for censoring distribution
`lambdas`	scale parameter of the baseline distribution
`gammas`	shape parameter of the baseline distribution
`beta_list`	a list containing coefficients for the covariates to be generated, each value corresponds to one transition

`cov_means`	mean value of each of the covariates
`cov_sds`	standard deviation of each of the covariates
`trans_list`	transition matrix of the multi-state model based on number of states
`state_names`	states of the multi-state model

Details

This function is used for simulating a multi-state model with n no. of states and status corresponding to each state along with a number of covariates both continuous or categorical.

Value

a multi-state dataframe with given number of states, corresponding status and the covariate vector

Author(s)

Atanu Bhattacharjee,Gajendra Kumar Vishwakarma,Abhipsa Tripathy

Examples

```
##
msdata_4state <- simdata(seed=123,n=1000,dist="weibull",cdist="exponential",
                 cparams=list(rate = 0.1),lambdas=c(0.1, 0.2, 0.3, 0.4),
                 gammas=c(1.5, 2, 2.5, 2.6),beta_list=list(c(-0.05, 0.01, 0.5, 0.6),
                 c(-0.03, 0.02, 0.07, 0.08),c(-0.04, 0.03, 0.04, -0.03),
               c(-0.05, 0.05, 0.6, 0.8)),cov_means=c(0, 10, 2, 3),cov_sds=c(1,20,5,1.05),
                 trans_list=list(c(2, 3, 4, 5),c(3, 4, 5),c(4, 5), c(5), c()),
                 state_names=c("Tx", "Rec", "Death", "Reldeath", "srv"))
##
```

Index