Package ‘psborrow’

June 4, 2021

Title Bayesian Dynamic Borrowing with Propensity Score

Version 0.1.0

Description A tool which aims to help evaluate the effect of external borrowing using an integrated approach described in Lewis et al., (2019) <doi:10.1080/19466315.2018.1497533> that combines propensity score and Bayesian dynamic borrowing methods.

License GPL (>= 3) | file LICENCE

Encoding UTF-8

RoxygenNote 7.1.1

Imports dplyr, tidyr, data.table, rjags, mvtnorm, matrixcalc, ggplot2, foreach, doParallel, parallel, MatchIt, survival, futile.logger, methods, utils

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Collate 'add_cov.R' 'add_time.R' 'add_mcmc.R' 'simu_cov.R'
  'match_cov.R' 'simu_time.R' 'run_mcmc.R' 'get_summary.R'
  'utils.R'

Depends R (>= 3.5.0)

NeedsCompilation no

Author Yichen Lu [aut],
  Aijing Lin [ctb, cre]

Maintainer Aijing Lin <lina21@gene.com>

Repository CRAN

Date/Publication 2021-06-04 08:00:02 UTC

R topics documented:

  .clinClass-class ......................................................... 2
  .covClass-class ......................................................... 2
  .eventClass-class ....................................................... 2
  .priorClass-class ....................................................... 3
  c_.covClass-method ..................................................... 3
Index 18

.clinClass-class
S4 Class for specifying parameters for enrollment time, drop-out pattern and analysis start time

Description
S4 Class for specifying parameters for enrollment time, drop-out pattern and analysis start time

.covClass-class
S4 Class for setting up covariates

Description
S4 Class for setting up covariates

.eventClass-class
S4 Class for setting parameters for time-to-events

Description
S4 Class for setting parameters for time-to-events
The `priorClass-class` S4 Class for specifying prior distributions and predictors for MCMC methods.

**Description**

S4 Class for specifying prior distributions and predictors for MCMC methods.

**c,.covClass-method**

Concatenate multiple .covClass classes.

**Description**

Concatenate multiple .covClass classes.

**Usage**

```r
## S4 method for signature '.covClass'
c(x, ...)
```

**Arguments**

- `x`: A .covClass class with covariate information generated in `set_cov`
- `...`: Other .covClass classes with covariate information generated in `set_cov`

**Value**

A vector of .covClass classes.

**Examples**

```r
# combine two sets of covariates
covset1 = set_cov(n_cat = 2, n_cont = 0, mu_int = 0, mu_ext = 0, var = 1)
covset2 = set_cov(n_cat = 0, n_cont = 1, mu_int = 62, mu_ext = 65, var = 11)
cov_list = c(covset1, covset2)
```
**get_summary**

### Description
Generate summary statistics of a simulation scenario

### Usage
```
get_summary(dt)
```

### Arguments
- `dt` a `data.frame` containing summary statistics for the posterior samples from each simulation

### Value
a `data.frame` containing the mean and sd of posterior HR between treatment and control arm, the posterior mean and sd of HR between internal control and external control arm, reject rate, variance, bias and mse of the simulation set

---

**c,.priorClass-method**  
*Concatenate multiple .priorClasss class*

### Description
Concatenate multiple .priorClasss class

### Usage
```
## S4 method for signature '.priorClass'
c(x, ...)
```

### Arguments
- `x` A .priorClasss class with prior distribution information generated in `set_prior`
- `...` A .priorClasss class with prior distribution information generated in `set_prior`

### Value
A vector of .priorClasss classes
**match_cov**

**Match**

**Description**

Match

**Usage**

match_cov(dt, match)

**Arguments**

dt a list of matrix

match A vector of covariates name to match on

**Value**

a list of matrix containing matched cohort information

**Examples**

# match internal and external trial data using different covariates
smp = set_n(ssC = 140, ssE = 275, ssExt = 100)
covset1 = set_cov(n_cat = 2, n_cont = 0, mu_int = 0, mu_ext = 0, var = 1)
covset2 = set_cov(n_cat = 0, n_cont = 1, mu_int = 62, mu_ext = 65, var = 11)
cObj = c(covset1, covset2)
sample_cov <-
simu_cov(ssObj = smp, covObj = cObj, HR = 1, driftHR = 1.2, nsim = 2)

# match on covariates 1 and 2
match_cov(dt = sample_cov, match = c("cov1", "cov2"))

# match on all 3 covariates
match_cov(dt = sample_cov, match = c("cov1", "cov2", "cov3"))

**plot_bias**

**Plot bias**

**Description**

Plot bias

**Usage**

plot_bias(dt, HR = 1, driftHR = 1, pred = "none")
Arguments

- **dt**: a data.frame containing summary statistics for the posterior samples from each simulation generated with `get_summary`
- **HR**: pre-specified HR between treatment and control arm in the internal trial
- **driftHR**: pre-specified HR between external control arm and internal control arm
- **pred**: predictors to use when fitting exponential distribution in MCMC

Value

- a `ggplot` which is a bar plot containing bias implications corresponding to each prior, the pre-specified HR between internal treatment and control arms is 1

---

plot_hr

*Plot posterior hazard ratio between treatment and control*

Description

Plot posterior hazard ratio between treatment and control

Usage

```r
plot_hr(dt, HR = 0.67, driftHR = 1, pred = "none")
```

Arguments

- **dt**: a data.frame containing summary statistics for the posterior samples from each simulation generated with `get_summary`
- **HR**: pre-specified HR between treatment and control arm in the internal trial
- **driftHR**: pre-specified HR between external control arm and internal control arm
- **pred**: predictors to use when fitting exponential distribution in MCMC

Value

- a `ggplot` which is a point plot containing posterior hazard ratio between treatment and control arms corresponding to each prior
**plot_mse**  

**Plot MSE**

**Description**
Plot MSE

**Usage**
```
plot_mse(dt, HR = 1, driftHR = 1, pred = "none")
```

**Arguments**
- `dt`: a data.frame containing summary statistics for the posterior samples from each simulation generated with `get_summary`
- `HR`: pre-specified HR between treatment and control arm in the internal trial
- `driftHR`: pre-specified HR between external control arm and internal control arm
- `pred`: predictors to use when fitting exponential distribution in MCMC

**Value**
a ggplot which is a bar plot containing MSE implications corresponding to each prior, the pre-specified HR between internal treatment and control arms is 1

**plot_power**  

**Plot power**

**Description**
Plot power

**Usage**
```
plot_power(dt, HR = 0.67, driftHR = 1, pred = "none")
```

**Arguments**
- `dt`: a data.frame containing summary statistics for the posterior samples from each simulation generated with `get_summary`
- `HR`: pre-specified HR between treatment and control arm in the internal trial
- `driftHR`: pre-specified HR between external control arm and internal control arm
- `pred`: predictors to use when fitting exponential distribution in MCMC

**Value**
a ggplot which is a bar plot containing power implications corresponding to each prior
### plot_type1error
*Plot type 1 error*

#### Description
Plot type 1 error

#### Usage
```r
plot_type1error(dt, driftHR = 1, pred = "none")
```

#### Arguments
- `dt`: a `data.frame` containing summary statistics for the posterior samples from each simulation generated with `get_summary`
- `driftHR`: pre-specified HR between external control arm and internal control arm
- `pred`: predictors to use when fitting exponential distribution in MCMC

#### Value
A `ggplot` which is a bar plot containing type 1 error implications corresponding to each prior, the pre-specified HR between internal treatment and control arms is 1

### rej_est
*Generate summary statistics for the MCMC chains*

#### Description
Generate summary statistics for the MCMC chains

#### Usage
```r
rej_est(samples)
```

#### Arguments
- `samples`: an object of class `mcmc.list`

#### Value
A vector containing the mean, median, sd, reject rate for the MCMC chains
**run_mcmc**

Run MCMC for multiple scenarios with provided data

**Usage**

```
run_mcmc(dt, priorObj, n.chains, n.adapt, n.burn, n.iter, seed, path)
```

**Arguments**

- `dt`: a list of `matrix` containing simulated time-to-events information
- `priorObj`: an object of class `.priorClass` generated in `set_prior`
- `n.chains`: number of parallel chains for the model
- `n.adapt`: number of iterations for adaptation
- `n.burn`: number of iterations discarded as burn-in
- `n.iter`: number of iterations to monitor
- `seed`: the seed of random number generator. Default is the first element of `.Random.seed`
- `path`: file name for saving the output including folder path

**Value**

a `data.frame` containing summary statistics of the posterior distribution for each simulation

**Examples**

```
# examples in vignette
```

---

**run_mcmc_p**

Run MCMC for multiple scenarios with provided data with parallel processing

**Description**

Run MCMC for multiple scenarios with provided data with parallel processing

**Usage**

```
run_mcmc_p(dt, priorObj, n.chains, n.adapt, n.burn, n.iter, seed, path)
```
Arguments

- **dt**: a list of matrix containing simulated time-to-events information
- **priorObj**: an object of class `priorClass` generated in `set_prior`
- **n.chains**: number of parallel chains for the model
- **n.adapt**: number of iterations for adaptation
- **n.burn**: number of iterations discarded as burn-in
- **n.iter**: number of iterations to monitor
- **seed**: the seed of random number generator. Default is the first element of `.Random.seed`
- **path**: file name for saving the output including folder path

Value

- a `data.frame` containing summary statistics of the posterior distribution for each simulation

Examples

```r
# similar to run_mcmc
```

---

**set_clin**  
*Specify parameters for enrollment time, drop-out pattern and analysis start time*

**Description**

This function allows user to specify the enrollment and drop-out rate, and the type of clinical cut-off Date. Both enrollment times and drop-out times follow piece-wise exponential distribution.

**Usage**

```r
set_clin(gamma, e_itv, CCOD, CCOD_t, etaC, etaE, d_itv)
```

**Arguments**

- **gamma**: A vector of rate of enrollment per unit of time
- **e_itv**: A vector of duration of time periods for recruitment with rates specified in `gamma`. Note that the length of `e_itv` should be same length as `gamma` or 1 less.
- **CCOD**: Type of analysis start time. Analysis starts at `CCOD_t` months after the first or last patient's enrollment if `CCOD = "fixed-first"` or `CCOD = "fixed-last"` respectively. Analysis starts when `CCOD_t` events have been observed if `CCOD = "event"`
set_cov

set_cov is a function in a statistical software package that is used to set up covariates for a clinical trial. It saves the mean, variance, and covariance among covariates. For technical details, see the vignette.

**Description**

This function saves the mean, variance, and covariance among covariates. For technical details, see the vignette.

**Usage**

```r
set_cov(n_cat, n_cont, mu_int, mu_ext, var, cov, prob_int, prob_ext)
```

**Arguments**

- **n_cat**: Number of binary variable
- **n_cont**: Number of continuous variable
- **mu_int**: Mean of covariates in the internal trial. All the covariates are simulated from a multivariate normal distribution. If left NULL, it uses default value 0 for all covariates. If provided one value, this value is used for all covariates
- **mu_ext**: Mean of covariates in the external trial
- **var**: Variance
- **cov**: Covariance
- **prob_int**: Probability of event in the internal trial
- **prob_ext**: Probability of event in the external trial

**Examples**

```r
# set the operational parameter values for the trial
# analysis starts at 64 time units after first patient in
set_clin(gamma = 10, e_itv = 4, etaC = 0.003, CCOD = "fixed-first", CCOD_t = 64)

# analysis starts at 12 time units after last patient in
set_clin(gamma = 2, e_itv = 18, etaC = 0.005, CCOD = "fixed-last", CCOD_t = 12)
```
mu_ext  Mean of covariates in the external trial. If left NULL, it uses the same mean as mu_int
var  Variance of covariates. If left NULL, it uses default value 0 for all covariates. If provided one value, it uses this value for all covariates
cov  Covariance between each pair of covariates. Covariance needs to be provided in a certain order and users are encouraged to read the example provided in the vignette. If left NULL, it uses default value 0 for all covariates. If provided one value, it uses this value for every pair of covariates
prob_int  Probability of binary covariate equaling 1 in the internal trial. If left NULL, it uses default value 0.5 for all covariates. If provided one value, it uses this value for all covariates
prob_ext  Probability of binary covariate equaling 1 in the external trial. If left NULL, it uses the same probability as prob_int

definition
A .covClass class containing covariate information

Description
Set up time-to-events

Usage
set_event(event, lambdaC, beta, shape, t_itv, change, keep)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Distribution of time-to-events: event = &quot;pwexp&quot; for piece-wise exponential distribution. event = &quot;weibull&quot; for weibull distribution</td>
</tr>
<tr>
<td>lambdaC</td>
<td>Baseline hazard rate of internal control arm. Specify a vector for piece-wise hazard with duration specified in t_itv if event = &quot;pwexp&quot;</td>
</tr>
<tr>
<td>beta</td>
<td>covariates’ coefficients. NULL if no covariates are found in ssObj</td>
</tr>
<tr>
<td>shape</td>
<td>shape of weibull distribution if event = &quot;weibull&quot;. NULL if event = &quot;pwexp&quot;</td>
</tr>
<tr>
<td>t_itv</td>
<td>a vector indicating interval lengths where the exponential rates provided in lambdaC apply. Note that the length of t_itv is at least 1 less than that of lambdaC and that the final value rate in lambdaC applies after time sum(t_itv). NULL if event = &quot;weibull&quot;</td>
</tr>
<tr>
<td>change</td>
<td>operations applied to covariates for simulating time-to-events</td>
</tr>
<tr>
<td>keep</td>
<td>original covariates to keep when simulate time-to-events</td>
</tr>
</tbody>
</table>
set_n

Value

A .eventClass class containing time-to-events information
A matrix containing simulated time-to-events information

Examples

# time-to-event follows a weibull distribution
set_event(event = "weibull", shape = 0.9, lambdaC = 0.0135)

# time-to-event follows a piece-wise exponential distribution
set_event(event = "pwexp", t_itv = 1, lambdaC = c(0.1, 0.02))

Description

This function conducts validity check and generates a matrix with two binary variables indicating

1. if the observation belongs to the external trial
2. if the observation belongs to the treatment arm.

Usage

set_n(ssC, ssE, ssExt)

Arguments

ssC Number of observations in the internal control arm. Default is 100
ssE Number of observations in the internal experiment arm. Default is the same number of observations as ssC
ssExt Number of observations in the external control arm. Default is the same number of observations as ssC

Value

A matrix containing external trial indicator and treatment indicator
set_prior

Specify prior distributions and predictors for MCMC methods

Usage

set_prior(pred, prior, r0, alpha, sigma)

Arguments

pred Predictors to include in the weibull distribution. No covariates except for treatment indicator is included if pred = NULL. Only propensity score generated using a logistic regression model on all covariates and treatment indicator are included if pred = ps. All covariates and treatment indicator are included if pred = all

prior Prior distribution for the precision parameter that controls the degree of borrowing. Half-cauchy distribution if prior = "cauchy". No external data is included in the data if prior = "no_ext". External control arm is assumed to have the same baseline hazards as internal control arm if prior = "full_ext". Other options include "gamma" and "unif"

r0 Initial values for the shape of the weibull distribution for time-to-events

alpha Initial values for log of baseline hazard rate for external and internal control arms. Length of alpha should be 1 if prior = "full_ext" or prior = "no_ext", and equal to 2 otherwise

sigma Initial values for precision parameter if prior = "cauchy". If left NULL, default value 0.03 is used

Value

a .priorClass class containing survival data and prior information

Examples

# hierachical Bayesian model with precision parameter follows a half-cauchy distribution
set_prior(pred = "none", prior = "cauchy", r0 = 1, alpha = c(0, 0), sigma = 0.03)

# hierachical Bayesian model with precision parameter follows a gamma distribution
set_prior(pred = "none", prior = "gamma", r0 = 1, alpha = c(0, 0))

# conventional Bayesian model to not borrow from external control arm
set_prior(pred = "none", prior = "no_ext", alpha = 0)

# conventional Bayesian model to fully borrow from external control arm
set_prior(pred = "none", prior = "full_ext", alpha = 0)
Description

This function generates continuous and binary covariates through simulating from a multivariate normal distribution. Outcomes are further converted to binary variables using quantiles of the normal distribution calculated from the probability provided. Then the covariates are added to the external trial and treatment arm indicators.

Usage

```r
simu_cov(ssObj, covObj, driftHR, HR, nsim, seed, path)
```

Arguments

- **ssObj**: an object of class `covClass` generated in `set_n`
- **covObj**: an object of class `covClass` generated in `set_cov`
- **driftHR**: hazard ratio of external control and internal control arms
- **HR**: a list of hazard ratio of treatment and control arms
- **nsim**: number of simulation. Default is 5
- **seed**: the seed of R’s random number generator. Default is the first element of `.Random.seed`
- **path**: file name for saving the output including folder path

Value

a list of matrix containing simulated covariates information

Examples

```r
# simulate patient-level data with 1 continuous covariate
sample = set_n(ssC = 10, ssE = 20, ssExt = 40)
cov1 = set_cov(n_cat = 0, n_cont = 1, mu_int = 0, mu_ext = 0, var = 1)
simu_cov(ssObj = sample, covObj = cov1, HR = 0.5, driftHR = 1, nsim = 2)

# simulate patient-level data with 1 binary and 2 continuous covariate
cov2 = set_cov(n_cat = 1, n_cont = 2, mu_int = 0, mu_ext = 0, var = 1,
               cov = 0.3, prob_int = 0.2, prob_ext = 0.3)
simu_cov(ssObj = sample, covObj = cov2, HR = 0.5, driftHR = 1, nsim = 2)
```
simu_time

Simulate time-to-events for multiple scenarios

Description

Simulate time-to-events for multiple scenarios

Usage

simu_time(dt, eventObj, clinInt, clinExt, seed, path)

Arguments

dt
  a list of matrix generated in simu_cov containing simulated covariates information

eventObj
  an object of class .eventClass generated in set_event including event information

clinInt
  an object of class .clinClass generated in set_clin including internal trial information

clinExt
  an object of class .clinClass generated in set_clin including external trial information

seed
  the seed of R’s random number generator. Default is the first element of .Random.seed

path
  file name for saving the output including folder path

Value

a list of matrix containing simulated time-to-events information

Examples

# simulate patient-level data without covariates
# simulate survival time following weibull distribution

# simulate trial indicator and set hazard ratios
sample = set_n(ssC = 10, ssE = 20, ssExt = 40)
sample_hr <- simu_cov(ssObj = sample, HR = 1, driftHR=c(1,1.2), nsim = 10)

# enrollment pattern, drop-out, analysis start time
c_int = set_clin(gamma = 2, e_itv = 10, etaC = 0.5, CCOD = "fixed-first", CCOD_t = 64)
c_ext = c_int

# simulate time-to-event with a weibull distribution
evt1 <- set_event(event = "weibull", shape = 0.8, lambdaC = 0.01)
simu_time(dt = sample_hr, eventObj = evt1, clinInt = c_int, clinExt = c_ext)

# simulate time-to-event with an exponential distribution
```r
evt2 <- set_event(event = "pwexp", t_itv = 1, lambdaC = c(0.1, 0.02))
simu_time(dt = sample_hr, eventObj = evt2, clinInt = c_int, clinExt = c_int)
```
Index

* classes
  .covClass-class, 2

* class
  .clinClass-class, 2
  .eventClass-class, 2
  .priorClass-class, 3

* constructor
  match_cov, 5
  set_clin, 10
  set_cov, 11
  set_event, 12
  set_n, 13
  set_prior, 14

* helper
  c,.covClass-method, 3
  c,.priorClass-method, 4

* method
  c,.covClass-method, 3
  c,.priorClass-method, 4
  get_summary, 4, 6–8
  match_cov, 5
  plot_bias, 5
  plot_hr, 6
  plot_mse, 7
  plot_power, 7
  plot_type1error, 8
  rej_est, 8
  run_mcmc, 9
  run_mcmc_p, 9
  set_clin, 10, 16
  set_cov, 3, 11, 15
  set_event, 12, 16
  set_n, 13, 15
  set_prior, 4, 9, 10, 14
  simu_cov, 15, 16
  simu_cov,matrix-method(simu_cov), 15
  simu_time, 16