Package ‘SAMprior’

September 27, 2023

Type   Package
Title  Self-Adapting Mixture (SAM) Priors
Version 1.1.1
Maintainer Peng Yang <py11@rice.edu>
Description Implementation of the SAM prior and generation of its operating characteristics for dynamically borrowing information from historical data. For details, please refer to Yang et al. (2023) <doi:10.1111/biom.13927>.
Depends RBesT, assertthat, checkmate, Metrics, ggplot2
Suggests rmarkdown, knitr, testthat (>= 2.0.0), foreach, purrr, rstanarm (>= 2.17.2), scales, tools, broom, tidyr, parallel
VignetteBuilder knitr
License GPL (>= 3)
Encoding UTF-8
RoxygenNote 7.2.3
NeedsCompilation no
Author Peng Yang [aut, cre] (<https://orcid.org/0000-0002-0630-9506>), Ying Yuan [aut] (<https://orcid.org/0000-0003-3163-480X>)
Repository CRAN
Date/Publication 2023-09-27 19:10:02 UTC

R topics documented:

<table>
<thead>
<tr>
<th>get_OC</th>
<th>SAM_prior</th>
<th>SAM_weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Index 14
Description

The get_OC function is designed to generate the operating characteristics of SAM priors (Yang, et al., 2023), including the relative bias, relative mean squared error, and type I error and power under a two-arm comparative trial design. As an option, the operating characteristic of robust MAP priors (Schmidli, et al., 2014) can also be generated for comparison.

Usage

get_OC(
  if.prior, theta.h, method.w, prior.odds, nf.prior, delta, n, n.t, decision, ntrial, if.MAP, weight, theta, theta.t, ...
)

## S3 method for class 'betaMix'
get_OC(
  if.prior, theta.h, method.w, prior.odds, nf.prior, delta, n, n.t, decision, ntrial, if.MAP, weight, theta, theta.t, ...
)
get_OC

## S3 method for class 'normMix'
get_OC(
  if.prior,  # Informative prior constructed from historical data, represented (approximately) as a mixture of conjugate distributions.
  theta.h,   # Estimate of the treatment effect based on historical data. If missing, the default value is set to be the posterior mean estimate from if.prior.
  method.w,  # Methods used to determine the mixture weight for SAM priors. The default method is LRT (Likelihood Ratio Test), the alternative option can be PPR (Posterior Probability Ratio). See SAM_weight for more details.
  prior.odds, # The prior probability of $H_0$ being true compared to the prior probability of $H_1$ being true using PPR method. The default value is 1. See SAM_weight for more details.
  nf.prior,  # Non-informative prior used for constructing the SAM prior and robust MAP prior.
  delta,     # Clinically significant difference used for the SAM prior.
  n,         # Sample size for the control arm.
  n.t,       # Sample size for the treatment arm.
  decision,  # Decision rule to compare the treatment with the control; see decision2S.
  ntrial,    # Number of trials simulated.
  if.MAP,    # Whether to simulate the operating characteristics of the robust MAP prior for comparison, the default value is FALSE.
  weight,    # Weight assigned to the informative prior component ($0 \leq \text{weight} \leq 1$) for the robust MAP prior, the default value is 0.5.
  ...,
  sigma
)

Arguments

if.prior  Informative prior constructed from historical data, represented (approximately) as a mixture of conjugate distributions.
theta.h   Estimate of the treatment effect based on historical data. If missing, the default value is set to be the posterior mean estimate from if.prior.
method.w  Methods used to determine the mixture weight for SAM priors. The default method is LRT (Likelihood Ratio Test), the alternative option can be PPR (Posterior Probability Ratio). See SAM_weight for more details.
prior.odds The prior probability of $H_0$ being true compared to the prior probability of $H_1$ being true using PPR method. The default value is 1. See SAM_weight for more details.
f.prior   Non-informative prior used for constructing the SAM prior and robust MAP prior.
delta     Clinically significant difference used for the SAM prior.
n         Sample size for the control arm.
n.t       Sample size for the treatment arm.
decision  Decision rule to compare the treatment with the control; see decision2S.
ntrial    Number of trials simulated.
if.MAP    Whether to simulate the operating characteristics of the robust MAP prior for comparison, the default value is FALSE.
weight    Weight assigned to the informative prior component ($0 \leq \text{weight} \leq 1$) for the robust MAP prior, the default value is 0.5.
get_OC

theta  A vector of the response rate (binary endpoints) or mean (continuous endpoints) for the control arm.
theta.t  A vector of the response rate (binary endpoints) or mean (continuous endpoints) for the treatment arm.
...
Additional parameters for continuous endpoints.
sigma  Variance to simulate the continuous endpoint under normality assumption.

Details

The `get_OC` function is designed to generate the operating characteristics of SAM priors, including the relative bias, relative mean squared error, and type I error, and power under a two-arm comparative trial design. As an option, the operating characteristics of robust MAP priors (Schmidli, et al., 2014) can also be generated for comparison.

The relative bias is defined as the difference between the bias of a method and the bias of using a non-informative prior. The relative mean squared error is the difference between the mean squared error (MSE) of a method and the MES of using a non-informative prior.

To evaluate type I error and power, the determination of whether the treatment is superior to the control is calculated based on function `decision2S`.

Value

Returns dataframe that contains the relative bias, relative MSE, type I error, and power for both SAM priors, as well as robust MAP priors. Additionally, the mixture weight of the SAM prior is also displayed.

Methods (by class)

- `get_OC(betaMix)`: The function is designed to generate the operating characteristics of SAM priors for binary endpoints.
- `get_OC(normMix)`: The function is designed to generate the operating characteristics of SAM priors for continuous endpoints.

References


Examples

```r
set.seed(123)
## Example of a binary endpoint
## Consider a randomized comparative trial designed to borrow information
## from historical data on the control. We assumed a non-informative prior
## beta(1, 1) and an informative prior beta(30, 50) after incorporating
## the historical data. The treatment is regarded as superior to the control
```
# if Pr(RR.t > RR.c | data) > 0.95, where RR.t and RR.c are response rates
# of the treatment and control, respectively. The operating characteristics
# were assessed under the scenarios of (RR.c, RR.t) = (0.3, 0.36) and (0.3, 0.56).
# OC <- get_OC(
## Informative prior constructed based on historical data
## if.prior = mixbeta(c(1, 30, 50)),
## ## Non-informative prior used for constructing the SAM prior
## nf.prior = mixbeta(c(1,1,1)),
## delta   = 0.2, ## Clinically significant difference
## n = 35,    ## Sample size for the control arm
## n.t = 70,  ## Sample size for the treatment arm
## ## Decision rule to compare the whether treatment is superior
## ## than the control
## decision = decision2S(0.95, 0, lower.tail=FALSE),
## ntrial  = 1000, ## Number of trials simulated
## ## Weight assigned to the informative component for MAP prior
## weight = 0.5,
## ## A vector of response rate for the control arm
## theta   = c(0.3, 0.36),
## ## A vector of response rate for the treatment arm
## theta.t = c(0.3, 0.56))
# OC

## Example of continuous endpoint
## Consider a randomized comparative trial designed to borrow information
## from historical data on the control. We assumed a non-informative prior
## N(0, 1e4) and an informative prior N(0.5, 2) after incorporating
## the historical data. The treatment is regarded as superior to the control
## if Pr(mean.t > mean.c | data) > 0.95, where mean.t and mean.c are mean
## of the treatment and control, respectively. The operating characteristics
## were assessed under the scenarios of (mean.c, mean.t) = (0.1, 0.1) and
## (0.5, 1.0).

sigma <- 2
prior.mean <- 0.5
prior.se <- sigma/sqrt(100)
# OC <- get_OC(
## Informative prior constructed based on historical data
## if.prior = mixnorm(c(1, prior.mean, prior.se)),
## ## Non-informative prior used for constructing the SAM prior
## nf.prior = mixnorm(c(1, 0, 1e4)),
## delta   = 0.2 * sigma, ## Clinically significant difference
## n = 100,    ## Sample size for the control arm
## n.t = 200,  ## Sample size for the treatment arm
## ## Decision rule to compare the whether treatment is superior
## ## than the control
## decision = decision2S(0.95, 0, lower.tail=FALSE),
## ntrial  = 1000, ## Number of trials simulated
## ## A vector of mean for the control arm
## theta   = c(0.1, 0.5),
## ## A vector of mean for the treatment arm
## theta.t = c(0.1, 1.0),
## sigma = sigma)
## OC
The **SAM_prior** function is designed to display the SAM prior, given the informative prior (constructed from historical data), non-informative prior, and the mixture weight calculated using **SAM_weight** function (Yang, et al., 2023).

### Usage

```r
SAM_prior(if.prior, nf.prior, weight, ...)  
## S3 method for class 'betaMix'
SAM_prior(if.prior, nf.prior, weight, ...)  
## S3 method for class 'gammaMix'
SAM_prior(if.prior, nf.prior, weight, ...)  
## S3 method for class 'normMix'
SAM_prior(if.prior, nf.prior, weight, ..., sigma)
```

### Arguments

- `if.prior` Informative prior constructed from historical data, represented (approximately) as a mixture of conjugate distributions.
- `nf.prior` Non-informative prior used for the mixture.
- `weight` Weight assigned to the informative prior component \((0 \leq \text{weight} \leq 1)\), which should be determined by **SAM_weight** function.
- `...` Additional parameters required for different endpoints.
- `sigma` Variance used for constructing the non-informative prior for continuous endpoints.

### Details

SAM prior is constructed by mixing an informative prior \(\pi_1(\theta)\), constructed based on historical data, with a non-informative prior \(\pi_0(\theta)\) using the mixture weight \(w\) determined by **SAM_weight** function to achieve the degree of prior-data conflict (Schmidli et al., 2015, Yang et al., 2023).

Let \(\theta\) and \(\theta_h\) denote the treatment effects associated with the current arm data \(D\) and historical data \(D_h\), respectively. Let \(\delta\) denote the clinically significant difference such that if \(|\theta_h - \theta| \geq \delta\), then \(\theta_h\) is regarded as clinically distinct from \(\theta\), and it is therefore inappropriate to borrow any information from \(D_h\). Consider two hypotheses:

\[
H_0: \theta = \theta_h, \ H_1: \theta = \theta_h + \delta \ or \ \theta = \theta_h - \delta.
\]
$H_0$ represents that $D_h$ and $D$ are consistent (i.e., no prior-data conflict) and thus information borrowing is desirable, whereas $H_1$ represents that the treatment effect of $D$ differs from $D_h$ to such a degree that no information should be borrowed.

The SAM prior uses the likelihood ratio test (LRT) statistics $R$ to quantify the degree of prior-data conflict and determine the extent of information borrowing.

$$ R = \frac{P(D|H_0, \theta_h)}{P(D|H_1, \theta_h)} = \frac{P(D|\theta = \theta_h)}{\max(P(D|\theta = \theta_h + \delta), P(D|\theta = \theta_h - \delta))}, $$

where $P(D|\cdot)$ denotes the likelihood function. An alternative Bayesian choice is the posterior probability ratio (PPR): 

$$ R = \frac{P(D|H_0, \theta_h)}{P(D|H_1, \theta_h)} = \frac{P(H_0)}{P(H_1)} \times BF, $$

where $P(H_0)$ and $P(H_1)$ is the prior probabilities of $H_0$ and $H_1$ being true. $BF$ is the Bayes Factor that in this case is the same as the LRT.

The SAM prior, denoted as $\pi_{sam}(\theta)$, is then defined as a mixture of an informative prior $\pi_1(\theta)$, constructed based on $D_h$ and a non-informative prior $\pi_0(\theta)$:

$$ \pi_{sam}(\theta) = w\pi_1(\theta) + (1 - w)\pi_0(\theta), $$

where the mixture weight $w$ is calculated as:

$$ w = \frac{R}{1 + R}. $$

As the level of prior-data conflict increases, the likelihood ratio $R$ decreases, resulting in a decrease in the weight $w$ assigned to the informative prior and thus a decrease in information borrowing. As a result, $\pi_{sam}(\theta)$ is data-driven and has the ability to self-adapt the information borrowing based on the degree of prior-data conflict.

Value

Displays the SAM prior as a mixture of an informative prior (constructed based on the historical data) and a non-informative prior.

Methods (by class)

- **SAM_prior(betaMix)**: The function calculates the SAM prior for beta mixture distribution. The default nf.prior is set to be mixbeta(c(1,1,1)) which represents a uniform prior Beta(1,1).
- **SAM_prior(gammaMix)**: The function calculates the SAM prior for gamma mixture distribution. The default nf.prior is set to be mixgamma(c(1,0.001,0.001)) which represents a vague gamma prior Gamma(0.001,0.001).
- **SAM_prior(normMix)**: The function calculates the SAM prior for normal mixture distribution. The default nf.prior is set to be mixnorm(c(1,summary(if.prior)['Var','mean'],sigma)) which represents a unit-information prior.
References


See Also

*SAM_weight*

Examples

```r
set.seed(123)
## Examples for binary endpoints
## Suppose that the informative prior constructed based on historical data is
## beta(40, 60)
prior.historical <- mixbeta(c(1, 40, 60))
## Data of the control arm
data.control <- rbinom(60, size = 1, prob = 0.42)
## Calculate the mixture weight of the SAM prior
wSAM <- SAM_weight(if.prior = prior.historical, 
                    delta = 0.15,                        ## Clinically significant difference
                    data = data.control)                ## Control arm data
## Assume beta(1,1) as the non-informative prior used for mixture
nf.prior <- mixbeta(nf.prior = c(1,1,1))
## Generate the SAM prior
SAM.prior <- SAM_prior(if.prior = prior.historical, ## Informative prior
                       nf.prior = nf.prior,               ## Non-informative prior
                       weight = wSAM)                  ## Mixture weight of the SAM prior
plot(SAM.prior)

## Examples for continuous endpoints
## Suppose that the informative prior constructed based on historical data is
## N(0, 3)
sigma <- 3
prior.mean <- 0
prior.se <- sigma/sqrt(100)
prior.historical <- mixnorm(c(1, prior.mean, prior.se), sigma = sigma)
## Data of the control arm
data.control <- rnorm(80, mean = 0, sd = sigma)
## Calculate the mixture weight of the SAM prior
wSAM <- SAM_weight(if.prior = prior.historical, 
                    delta = 0.2 * sigma,            ## Clinically significant difference
                    data = data.control)           ## Control arm data
## Assume unit-information prior N(0,3) as the non-informative prior used
## for the mixture
nf.prior <- mixnorm(nf.prior = c(1,prior.mean, sigma), 
```

## Generate the SAM prior
SAM.prior <- SAM_prior(if.prior = prior.historical, ## Informative prior
                       nf.prior = nf.prior, ## Non-informative prior
                       weight = wSAM ## Mixture weight of the SAM prior
)

plot(SAM.prior)

---

### SAM_weight

**Calculating Mixture Weight of SAM Priors**

**Description**

The `SAM_weight` function is designed to calculate the mixture weight of the SAM priors according to the degree of prior-data conflicts (Yang, et al., 2023).

**Usage**

SAM_weight(if.prior, theta.h, method.w, prior.odds, data, delta, ...)

```
## S3 method for class 'betaMix'
SAM_weight(if.prior, theta.h, method.w, prior.odds, data, delta, n, r, ...)

## S3 method for class 'normMix'
SAM_weight(if.prior, theta.h, method.w, prior.odds, data, delta, m, n, sigma, ...)

## S3 method for class 'gammaMix'
SAM_weight(if.prior, theta.h, method.w, prior.odds, data, delta, u, w, ...)
```

**Arguments**

- **if.prior**: Informative prior constructed based on historical data, represented (approximately) as a mixture of conjugate distributions.
- **theta.h**: Estimate of the treatment effect based on historical data. If missing, the default value is set to be the posterior mean estimate from `if.prior`.
Methods used to determine the mixture weight for SAM priors. The default method is "LRT" (Likelihood Ratio Test), the alternative option is "PPR" (Posterior Probability Ratio). See Details section for more information.

The prior probability of $H_0$ being true compared to the prior probability of $H_1$ being true using PPR method. The default value is 1. See Details section for more information.

Data of the control arm from the current trial, see Methods section for more details.

Clinically significant difference used for the SAM prior.

Additional significant parameters required for different endpoints.

Number of subjects in the control arm for continuous endpoint.

Number of responses in the control arm for binary endpoint.

Mean estimate in the control arm for continuous endpoint.

Standard deviation in the control arm for continuous endpoint.

Number of events in the control arm for time-to-event endpoint.

Total observed time in the control arm for time-to-event endpoint.

Details

SAM prior is constructed by mixing an informative prior $\pi_1(\theta)$, constructed based on historical data, with a non-informative prior $\pi_0(\theta)$ using the mixture weight $w$ determined by SAM_weight function to achieve the degree of prior-data conflict (Schmidli et al., 2015, Yang et al., 2023).

Let $\theta$ and $\theta_h$ denote the treatment effects associated with the current arm data $D$ and historical data $D_h$, respectively. Let $\delta$ denote the clinically significant difference such that if $|\theta_h - \theta| \geq \delta$, then $\theta_h$ is regarded as clinically distinct from $\theta$, and it is therefore inappropriate to borrow any information from $D_h$. Consider two hypotheses:

$$ H_0 : \theta = \theta_h, \quad H_1 : \theta = \theta_h + \delta \text{ or } \theta = \theta_h - \delta. $$

$H_0$ represents that $D_h$ and $D$ are consistent (i.e., no prior-data conflict) and thus information borrowing is desirable, whereas $H_1$ represents that the treatment effect of $D$ differs from $D_h$ to such a degree that no information should be borrowed.

The SAM prior uses the likelihood ratio test (LRT) statistics $R$ to quantify the degree of prior-data conflict and determine the extent of information borrowing.

$$ R = P(D|H_0, \theta_h)/P(D|H_1, \theta_h) = P(D|\theta = \theta_h)/\max(P(D|\theta = \theta_h + \delta), P(D|\theta = \theta_h - \delta)), $$

where $P(D|\cdot)$ denotes the likelihood function. An alternative Bayesian choice is the posterior probability ratio (PPR):

$$ R = P(D|H_0, \theta_h)/P(D|H_1, \theta_h) = P(H_0)/P(H_1) \times BF, $$

where $P(H_0)$ and $P(H_1)$ is the prior probabilities of $H_0$ and $H_1$ being true. $BF$ is the Bayes Factor that in this case is the same as the LRT.
The SAM prior, denoted as $\pi_{sam}(\theta)$, is then defined as a mixture of an informative prior $\pi_1(\theta)$, constructed based on $D_h$ and a non-informative prior $\pi_0(\theta)$:

$$\pi_{sam}(\theta) = w\pi_1(\theta) + (1 - w)\pi_0(\theta),$$

where the mixture weight $w$ is calculated as:

$$w = R / (1 + R).$$

As the level of prior-data conflict increases, the likelihood ratio $R$ decreases, resulting in a decrease in the weight $w$ assigned to the informative prior and thus a decrease in information borrowing. As a result, $\pi_{sam}(\theta)$ is data-driven and has the ability to self-adapt the information borrowing based on the degree of prior-data conflict.

Value

The mixture weight of the SAM priors.

Methods (by class)

- **SAM_weight(betaMix)**: The function calculates the mixture weight of SAM priors for beta mixture distribution. The input data can be patient-level data (i.e., a vector of 0 and 1 representing the response status of each patient) or summary statistics (i.e., the number of patients and the number of responses).

- **SAM_weight(normMix)**: The function calculates the mixture weight of SAM priors for normal mixture distribution. The input data should be a vector of patient-level observations. The input data can be patient-level data (i.e., a vector of continuous response of each patient) or summary statistics (i.e., the mean estimate, number of subjects, and the standard deviation in the control arm).

- **SAM_weight(gammaMix)**: The function calculates the mixture weight of SAM priors for gamma mixture distribution. The input data can be patient-level data (i.e., a matrix with the first row as the censoring indicator and the second row recording the observed time) or summary statistics (i.e., the number of uncensored observations $u$ and total observed time $w$).

References


Examples

```r
set.seed(123)
## Examples for binary endpoints
## Example 1: no prior-data conflict
## Suppose that the informative prior constructed based on historical data is
## beta(40, 60)
prior.historical <- mixbeta(c(1, 40, 60))
## Data of control arm
data.control <- rbinom(60, size = 1, prob = 0.42)
## Calculate the mixture weight of the SAM prior
```
SAM_weight

wSAM <- SAM_weight(if.prior = prior.historical,
                      delta = 0.15,  # Clinically significant difference
                      data = data.control # Control arm data
)
print(wSAM)

## Example 2: in the presence of prior-data conflict, where the current data
## has 12 responses in 60 patients
wSAM <- SAM_weight(if.prior = prior.historical,
                      delta = 0.15,  # Clinically significant difference
                      # Methods to determine mixture weight for the SAM priors
                      # by Posterior Probability Ratio
                      method.w = 'PPR',
                      # Prior odds of favoring no prior-data conflicts to
                      # the presence of prior-data conflict
                      prior.odd = 1/9,
                      n = 60,        # Number of patients in the control arm
                      r = 12         # Number of responses in the control arm
)
print(wSAM)

## Example 3: in the presence of prior-data conflict, where the current data
## has 12 responses in 60 patients
wSAM <- SAM_weight(if.prior = prior.historical,
                      delta = 0.15,  # Clinically significant difference
                      n = 60,        # Number of patients in the control arm
                      r = 12         # Number of responses in the control arm
)
print(wSAM)

## Examples for continuous endpoints
## Example 1: no prior-data conflict
## Suppose that the informative prior constructed from historical data is
## N(0, 3)
sigma <- 3
prior.mean <- 0
prior.se <- sigma/sqrt(100)
prior.historical <- mixnorm(c(1, prior.mean, prior.se), sigma = sigma)
## Data of the control arm
data.control <- rnorm(80, mean = 0, sd = sigma)
wSAM <- SAM_weight(if.prior = prior.historical,
                      delta = 0.3 * sigma,  # Clinically significant difference
                      data = data.control # Control arm data
)
print(wSAM)

## Example 2: in the presence of prior-data conflict, where the current data
## has mean of 0.5
data.control <- rnorm(80, mean = 1, sd = sigma)
wSAM <- SAM_weight(if.prior = prior.historical,
                      delta = 0.3 * sigma,  # Clinically significant difference
                      data = data.control # Control arm data
)
## Examples for survival endpoints

### Example 1: no prior-data conflict

Suppose the survival times from historical data follows \( \exp(1) \) distribution with random censoring time follows \( \mathcal{U}(0.5, 5) \) distribution.

```r
T_hi <- rexp(100, rate = 1)
C_hi <- runif(100, min = 0.5, max = 5)
```

Indicators of the uncensored events:

```r
delta_hi <- as.numeric(T_hi < C_hi)
```

Observed survival times from historical data:

```r
U_hi <- T_hi
U_hi[delta_hi == 0] <- C_hi[delta_hi == 0]
```

Construct the informative prior based on simulated historical data:

```r
prior.historical <- mixgamma(c(1, sum(delta_hi), sum(U_hi)),
param = c('ab', likelihood = 'exp'))
```

### Example 2: in the presence of prior-data conflict, where the current survival times follows \( \exp(2) \) distribution with random censoring time follows \( \mathcal{U}(0.5, 5) \) distribution.

```r
T_ci <- rexp(100, rate = 0.95)
C_ci <- runif(100, min = 0.5, max = 5)
```

Indicators of the uncensored events:

```r
delta_ci <- as.numeric(T_ci < C_ci)
```

Observed survival times from control data:

```r
U_ci <- T_ci
U_ci[delta_ci == 0] <- C_ci[delta_ci == 0]
```

Data of the control arm:

```r
data.control <- rbind(sum(delta_ci), sum(U_ci))
```

Weighted survival analysis:

```r
wSAM <- SAM_weight(if.prior = prior.historical, 
 delta = 0.2, 
 data = data.control 
)
```

Print the results:

```r
print(wSAM)
```
Index

decision2S, 3, 4
get_OC, 2
SAM_prior, 6
SAM_weight, 3, 6, 8, 9, 10