Package: PanCanVarSel (via r-universe)

September 5, 2024

Title Pan-Cancer Variable Selection

Version 0.0.4

Description Provides function for performing Bayesian survival regression using Horseshoe prior in the accelerated failure time model with log normal assumption in order to achieve high dimensional pan-cancer variable selection as developed in Maity et. al. (2019) <doi:10.1111/biom.13132>.

Depends R (>= 2.0.0)

Imports mvtnorm, Matrix, smoothmest, msm

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Repository https://maitya02.r-universe.dev

RemoteUrl https://github.com/maitya02/pancanvarsel

RemoteRef HEAD

RemoteSha 7a53dc16f97cad04bbddc2a43ca7553cb3856338

Contents

`hsaft` *Horseshoe shrinkage prior in Bayesian survival regression*

Description

This function employs the algorithm provided by van der Pas et. al. (2016) for log normal Accelerated Failure Rate (AFT) model to fit survival regression. The censored observations are updated according to the data augmentation of approach of Tanner and Wong (1984).

Usage

```
hsaft(ct, X, method.tau = c("fixed", "truncatedCauchy", "halfCauchy"),
  tau = 1, method.sigma = c("fixed", "Jeffreys"), Sigma2 = 1,
  burn = 1000, nmc = 5000, thin = 1, alpha = 0.05)
```

Arguments

`ct`	Response, a $n * 2$ matrix with first column as response and second column as right censored indicator, 1 is event time and 0 is right censored.
`X`	Matrix of covariates, dimension $n * p$.
`method.tau`	Method for handling τ. Select "truncatedCauchy" for full Bayes with the Cauchy prior truncated to [1/p, 1], "halfCauchy" for full Bayes with the half-Cauchy prior, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`tau`	Use this argument to pass the (estimated) value of τ in case "fixed" is selected for method.tau. Not necessary when method.tau is equal to"halfCauchy" or "truncatedCauchy". The default (tau = 1) is not suitable for most purposes and should be replaced.
`method.sigma`	Select "Jeffreys" for full Bayes with Jeffrey's prior on the error variance σ^2, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`Sigma2`	A fixed value for the error variance σ^2. Not necessary when method.sigma is equal to "Jeffreys". Use this argument to pass the (estimated) value of Sigma2 in case "fixed" is selected for method.sigma. The default (Sigma2 = 1) is not suitable for most purposes and should be replaced.
`burn`	Number of burn-in MCMC samples. Default is 1000.
`nmc`	Number of posterior draws to be saved. Default is 5000.
`thin`	Thinning parameter of the chain. Default is 1 (no thinning).
`alpha`	Level for the credible intervals. For example, alpha = 0.05 results in 95% credible intervals.

Details

The model is: t_i is response, c_i is censored time, $t_i^* = \min_{(} t_i, c_i)$ is observed time, w_i is censored data, so $w_i = \log t_i^*$ if t_i is event time and $w_i = \log t_i^*$ if t_i is right censored $\log t_i = X\beta + \epsilon, \epsilon \sim N(0, \sigma^2)$

Value

`SurvivalHat`	Predictive survival probability.
`LogTimeHat`	Predictive log time.
`BetaHat`	Posterior mean of Beta, a p by 1 vector.
`LeftCI`	The left bounds of the credible intervals.
`RightCI`	The right bounds of the credible intervals.
`BetaMedian`	Posterior median of Beta, a p by 1 vector.
`Sigma2Hat`	Posterior mean of error variance σ^2. If method.sigma = "fixed" is used, this value will be equal to the user-selected value of Sigma2 passed to the function.
`TauHat`	Posterior mean of global scale parameter tau, a positive scalar. If method.tau = "fixed" is used, this value will be equal to the user-selected value of tau passed to the function.
`BetaSamples`	Posterior samples of Beta.
`TauSamples`	Posterior samples of tau.
`Sigma2Samples`	Posterior samples of Sigma2.
`LikelihoodSamples`	Posterior Samples of likelihood.

References

Stephanie van der Pas, James Scott, Antik Chakraborty and Anirban Bhattacharya (2016). horseshoe: Implementation of the Horseshoe Prior. R package version 0.1.0. https://CRAN.R-project.org/package=horseshoe

Maity, A. K., Bhattacharyya, A., Mallick, B. K., and Baladandayuthapani, V. (2019) Data Integration and Variable Selection for Pan-Cancer Survival Prediction using Protein Expressions, Biometrics. <doi: 10.1111/biom.13132>

Examples

```
burnin <- 500   # number of burnin
nmc    <- 1000  # number of Markov Chain samples
y.sd   <- 1     # standard deviation of the data
p      <- 80    # number of covariates
n      <- 40    # number of samples
beta   <- as.vector(smoothmest::rdoublex(p))  # from double exponential distribution
x      <- mvtnorm::rmvnorm(n, mean = rep(0, p))  # from multivariate normal distribution
y.mu   <- x %*% beta  # mean of the data
y     <- as.numeric(stats::rnorm(n, mean = y.mu, sd = y.sd))  # from normal distribution
T      <- exp(y)   # AFT model
C      <- rgamma(n, shape = 1.75, scale = 3)  # censoring time
time   <- pmin(T, C)  # observed time is min of censored and true
status = time == T   # set to 1 if event is observed
ct     <- as.matrix(cbind(time = time, status = status))  # censored time

posterior.fit <- hsaft(ct, x, method.tau = "truncatedCauchy", method.sigma = "Jeffreys",
                       burn = burnin, nmc = nmc)
summary(posterior.fit$BetaHat)
```

`hsaftallcorr`	*This function extends the main function `hsaft` to create correlation among covariates.*

Description

This function extends the main function `hsaft` to create correlation among covariates.

Usage

```
hsaftallcorr(ct, X, method.tau = c("fixed", "truncatedCauchy",
  "halfCauchy"), tau = 1, method.sigma = c("fixed", "Jeffreys"),
  Sigma2 = 1, burn = 1000, nmc = 5000, thin = 1, alpha = 0.05, r,
  n.seq, pk)
```

Arguments

`ct`	Response, a $n * 2$ matrix with first column as response and second column as right censored indicator, 1 is event time and 0 is right censored.
`X`	Matrix of covariates, dimension $n * p$.
`method.tau`	Method for handling τ. Select "truncatedCauchy" for full Bayes with the Cauchy prior truncated to [1/p, 1], "halfCauchy" for full Bayes with the half-Cauchy prior, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`tau`	Use this argument to pass the (estimated) value of τ in case "fixed" is selected for method.tau. Not necessary when method.tau is equal to"halfCauchy" or "truncatedCauchy". The default (tau = 1) is not suitable for most purposes and should be replaced.
`method.sigma`	Select "Jeffreys" for full Bayes with Jeffrey's prior on the error variance σ^2, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`Sigma2`	A fixed value for the error variance σ^2. Not necessary when method.sigma is equal to "Jeffreys". Use this argument to pass the (estimated) value of Sigma2 in case "fixed" is selected for method.sigma. The default (Sigma2 = 1) is not suitable for most purposes and should be replaced.
`burn`	Number of burn-in MCMC samples. Default is 1000.
`nmc`	Number of posterior draws to be saved. Default is 5000.
`thin`	Thinning parameter of the chain. Default is 1 (no thinning).
`alpha`	Level for the credible intervals. For example, alpha = 0.05 results in 95% credible intervals.
`r`	number of groups.
`n.seq`	a vector of sample sizes for all groups.
`pk`	number of covariates in each group.

Value

`SurvivalHat`	Predictive survival probability.
`LogTimeHat`	Predictive log time.
`BetaHat`	Posterior mean of Beta, a p by 1 vector.
`LeftCI`	The left bounds of the credible intervals.
`RightCI`	The right bounds of the credible intervals.
`BetaMedian`	Posterior median of Beta, a p by 1 vector.
`Sigma2Hat`	Posterior mean of error variance σ^2. If method.sigma = "fixed" is used, this value will be equal to the user-selected value of Sigma2 passed to the function.
`TauHat`	Posterior mean of global scale parameter tau, a positive scalar. If method.tau = "fixed" is used, this value will be equal to the user-selected value of tau passed to the function.
`BetaSamples`	Posterior samples of Beta.
`TauSamples`	Posterior samples of tau.
`Sigma2Samples`	Posterior samples of Sigma2.
`BGHat`	Posterior samples of b which is a part of the mean of β.
`BPHat`	Posterior samples of b which is the other part of the mean of β.
`LikelihoodSamples`	Posterior Samples of likelihood.

References

Stephanie van der Pas, James Scott, Antik Chakraborty and Anirban Bhattacharya (2016). horseshoe: Implementation of the Horseshoe Prior. R package version 0.1.0. https://CRAN.R-project.org/package=horseshoe

Maity, A. K., Bhattacharyya, A., Mallick, B. K., and Baladandayuthapani, V. (2019) Data Integration and Variable Selection for Pan-Cancer Survival Prediction using Protein Expressions, Biometrics. <doi: 10.1111/biom.13132>

Examples

```
# Examples for hsaftallcorr function
burnin <- 50   # number of burnin
nmc    <- 100  # number of Markov Chain samples
y.sd   <- 1     # standard deviation of the data
p      <- 80    # number of covariates
r      <- 5     # number of groups
p      <- 80    # number of covariate in each group
n1     <- 40    # sample size of 1st group
n2     <- 50    # sample size of 2nd group
n3     <- 70    # sample size of 3rd group
n4     <- 100   # sample size of 4th group
n5     <- 120   # sample size of 5th group
n      <- sum(c(n1, n2, n3, n4, n5))  # total sample size
n.seq  <- c(n1, n2, n3, n4, n5)
Beta   <- matrix(smoothmest::rdoublex(p * r), nrow = r, ncol = p, byrow = TRUE)
```

```
# from double exponential distribution
beta   <- as.vector(t(Beta))  # vectorize Beta
x1     <- mvtnorm::rmvnorm(n1, mean = rep(0, p))
x2     <- mvtnorm::rmvnorm(n2, mean = rep(0, p))
x3     <- mvtnorm::rmvnorm(n3, mean = rep(0, p))
x4     <- mvtnorm::rmvnorm(n4, mean = rep(0, p))
x5    <- mvtnorm::rmvnorm(n5, mean = rep(0, p))  # from multivariate normal distribution
y.mu1  <- x1 %*% Beta[1, ]
y.mu2  <- x2 %*% Beta[2, ]
y.mu3  <- x3 %*% Beta[3, ]
y.mu4  <- x4 %*% Beta[4, ]
y.mu5  <- x5 %*% Beta[5, ]
y1     <- stats::rnorm(n1, mean = y.mu1, sd = y.sd)
y2     <- stats::rnorm(n2, mean = y.mu2, sd = y.sd)
y3     <- stats::rnorm(n3, mean = y.mu3, sd = y.sd)
y4     <- stats::rnorm(n4, mean = y.mu4, sd = y.sd)
y5     <- stats::rnorm(n5, mean = y.mu5, sd = y.sd)
y      <- c(y1, y2, y3, y4, y5)
x      <- Matrix::bdiag(x1, x2, x3, x4, x5)
X      <- as.matrix(x)
y      <- as.numeric(as.matrix(y))  # from normal distribution
T      <- exp(y)   # AFT model
C      <- rgamma(n, shape = 1.75, scale = 3)  # censoring time
time   <- pmin(T, C)  # observed time is min of censored and true
status = time == T   # set to 1 if event is observed
ct     <- as.matrix(cbind(time = time, status = status))  # censored time
posterior.fit <- hsaftallcorr(ct, X, method.tau = "truncatedCauchy", method.sigma = "Jeffreys",
                              burn = burnin, nmc = nmc,
                              r = r, n.seq = n.seq, pk = p)
summary(posterior.fit$BetaHat)
```

`hsaftcovariatecorr` *This function extends the main function* `hsaft` *to create correlation among covariates.*

Description

This function extends the main function `hsaft` to create correlation among covariates.

Usage

```
hsaftcovariatecorr(ct, X, method.tau = c("fixed", "truncatedCauchy",
  "halfCauchy"), tau = 1, method.sigma = c("fixed", "Jeffreys"),
  Sigma2 = 1, burn = 100, nmc = 500, thin = 1, alpha = 0.05, r,
  n.seq, pk)
```

Arguments

`ct`	Response, a $n * 2$ matrix with first column as response and second column as right censored indicator, 1 is event time and 0 is right censored.
`X`	Matrix of covariates, dimension $n * p$.
`method.tau`	Method for handling τ. Select "truncatedCauchy" for full Bayes with the Cauchy prior truncated to [1/p, 1], "halfCauchy" for full Bayes with the half-Cauchy prior, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`tau`	Use this argument to pass the (estimated) value of τ in case "fixed" is selected for method.tau. Not necessary when method.tau is equal to"halfCauchy" or "truncatedCauchy". The default (tau = 1) is not suitable for most purposes and should be replaced.
`method.sigma`	Select "Jeffreys" for full Bayes with Jeffrey's prior on the error variance σ^2, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`Sigma2`	A fixed value for the error variance σ^2. Not necessary when method.sigma is equal to "Jeffreys". Use this argument to pass the (estimated) value of Sigma2 in case "fixed" is selected for method.sigma. The default (Sigma2 = 1) is not suitable for most purposes and should be replaced.
`burn`	Number of burn-in MCMC samples. Default is 1000.
`nmc`	Number of posterior draws to be saved. Default is 5000.
`thin`	Thinning parameter of the chain. Default is 1 (no thinning).
`alpha`	Level for the credible intervals. For example, alpha = 0.05 results in 95% credible intervals.
`r`	number of groups.
`n.seq`	a vector of sample sizes for all groups.
`pk`	number of covariates in each group.

Value

`SurvivalHat`	Predictive survival probability.
`LogTimeHat`	Predictive log time.
`BetaHat`	Posterior mean of Beta, a p by 1 vector.
`LeftCI`	The left bounds of the credible intervals.
`RightCI`	The right bounds of the credible intervals.
`BetaMedian`	Posterior median of Beta, a p by 1 vector.
`Sigma2Hat`	Posterior mean of error variance σ^2. If method.sigma = "fixed" is used, this value will be equal to the user-selected value of Sigma2 passed to the function.
`TauHat`	Posterior mean of global scale parameter tau, a positive scalar. If method.tau = "fixed" is used, this value will be equal to the user-selected value of tau passed to the function.
`BetaSamples`	Posterior samples of Beta.
`TauSamples`	Posterior samples of tau.

`Sigma2Samples`	Posterior samples of Sigma2.
`BHat`	Posterior samples of b which is the mean of β.
`LikelihoodSamples`	Posterior Samples of likelihood.

References

Stephanie van der Pas, James Scott, Antik Chakraborty and Anirban Bhattacharya (2016). horseshoe: Implementation of the Horseshoe Prior. R package version 0.1.0. https://CRAN.R-project.org/package=horseshoe

Maity, A. K., Bhattacharyya, A., Mallick, B. K., and Baladandayuthapani, V. (2019) Data Integration and Variable Selection for Pan-Cancer Survival Prediction using Protein Expressions, Biometrics. <doi: 10.1111/biom.13132>

Examples

```
# Examples for hsaftcovariatecorr function
burnin <- 50   # number of burnin
nmc    <- 100  # number of Markov Chain samples
y.sd   <- 1     # standard deviation of the data
p      <- 80    # number of covariates
r      <- 5     # number of groups
p      <- 80    # number of covariate in each group
n1     <- 40    # sample size of 1st group
n2     <- 50    # sample size of 2nd group
n3     <- 70    # sample size of 3rd group
n4     <- 100   # sample size of 4th group
n5     <- 120   # sample size of 5th group
n      <- sum(c(n1, n2, n3, n4, n5))  # total sample size
n.seq  <- c(n1, n2, n3, n4, n5)
Beta   <- matrix(smoothmest::rdoublex(p * r), nrow = r, ncol = p, byrow = TRUE)
# from double exponential distribution
beta   <- as.vector(t(Beta))  # vectorize Beta
x1     <- mvtnorm::rmvnorm(n1, mean = rep(0, p))
x2     <- mvtnorm::rmvnorm(n2, mean = rep(0, p))
x3     <- mvtnorm::rmvnorm(n3, mean = rep(0, p))
x4     <- mvtnorm::rmvnorm(n4, mean = rep(0, p))
x5    <- mvtnorm::rmvnorm(n5, mean = rep(0, p))  # from multivariate normal distribution
y.mu1  <- x1 %*% Beta[1, ]
y.mu2  <- x2 %*% Beta[2, ]
y.mu3  <- x3 %*% Beta[3, ]
y.mu4  <- x4 %*% Beta[4, ]
y.mu5  <- x5 %*% Beta[5, ]
y1     <- stats::rnorm(n1, mean = y.mu1, sd = y.sd)
y2     <- stats::rnorm(n2, mean = y.mu2, sd = y.sd)
y3     <- stats::rnorm(n3, mean = y.mu3, sd = y.sd)
y4     <- stats::rnorm(n4, mean = y.mu4, sd = y.sd)
y5     <- stats::rnorm(n5, mean = y.mu5, sd = y.sd)
y      <- c(y1, y2, y3, y4, y5)
x      <- Matrix::bdiag(x1, x2, x3, x4, x5)
X      <- as.matrix(x)
y      <- as.numeric(as.matrix(y))  # from normal distribution
```

```
T      <- exp(y)   # AFT model
C      <- rgamma(n, shape = 1.75, scale = 3)  # censoring time
time   <- pmin(T, C)  # observed time is min of censored and true
status = time == T   # set to 1 if event is observed
ct     <- as.matrix(cbind(time = time, status = status))  # censored time

posterior.fit <- hsaftcovariatecorr(ct, X, method.tau = "truncatedCauchy",
                                    method.sigma = "Jeffreys",
                                    burn = burnin, nmc = nmc,
                                    r = r, n.seq = n.seq, pk = p)
                                    summary(posterior.fit$BetaHat)
```

`hsaftgroupcorr` *This function extends the main function* `hsaft` *to create correlation among groups.*

Description

This function extends the main function `hsaft` to create correlation among groups.

Usage

```
hsaftgroupcorr(ct, X, method.tau = c("fixed", "truncatedCauchy",
  "halfCauchy"), tau = 1, method.sigma = c("fixed", "Jeffreys"),
  Sigma2 = 1, burn = 1000, nmc = 5000, thin = 1, alpha = 0.05, r,
  n.seq, pk)
```

Arguments

`ct`	Response, a $n * 2$ matrix with first column as response and second column as right censored indicator, 1 is event time and 0 is right censored.
`X`	Matrix of covariates, dimension $n * p$.
`method.tau`	Method for handling τ. Select "truncatedCauchy" for full Bayes with the Cauchy prior truncated to [1/p, 1], "halfCauchy" for full Bayes with the half-Cauchy prior, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`tau`	Use this argument to pass the (estimated) value of τ in case "fixed" is selected for method.tau. Not necessary when method.tau is equal to"halfCauchy" or "truncatedCauchy". The default (tau = 1) is not suitable for most purposes and should be replaced.
`method.sigma`	Select "Jeffreys" for full Bayes with Jeffrey's prior on the error variance σ^2, or "fixed" to use a fixed value (an empirical Bayes estimate, for example).
`Sigma2`	A fixed value for the error variance σ^2. Not necessary when method.sigma is equal to "Jeffreys". Use this argument to pass the (estimated) value of Sigma2 in case "fixed" is selected for method.sigma. The default (Sigma2 = 1) is not suitable for most purposes and should be replaced.

`burn`	Number of burn-in MCMC samples. Default is 1000.
`nmc`	Number of posterior draws to be saved. Default is 5000.
`thin`	Thinning parameter of the chain. Default is 1 (no thinning).
`alpha`	Level for the credible intervals. For example, alpha = 0.05 results in 95% credible intervals.
`r`	number of groups.
`n.seq`	a vector of sample sizes for all groups.
`pk`	number of covariates in each group.

Value

`SurvivalHat`	Predictive survival probability.
`LogTimeHat`	Predictive log time.
`BetaHat`	Posterior mean of Beta, a p by 1 vector.
`LeftCI`	The left bounds of the credible intervals.
`RightCI`	The right bounds of the credible intervals.
`BetaMedian`	Posterior median of Beta, a p by 1 vector.
`Sigma2Hat`	Posterior mean of error variance σ^2. If method.sigma = "fixed" is used, this value will be equal to the user-selected value of Sigma2 passed to the function.
`TauHat`	Posterior mean of global scale parameter tau, a positive scalar. If method.tau = "fixed" is used, this value will be equal to the user-selected value of tau passed to the function.
`BetaSamples`	Posterior samples of Beta.
`TauSamples`	Posterior samples of tau.
`Sigma2Samples`	Posterior samples of Sigma2.
`BHat`	Posterior samples of b which is the mean of β.
`LikelihoodSamples`	Posterior Samples of likelihood.

References

Stephanie van der Pas, James Scott, Antik Chakraborty and Anirban Bhattacharya (2016). horseshoe: Implementation of the Horseshoe Prior. R package version 0.1.0. https://CRAN.R-project.org/package=horseshoe

Maity, A. K., Bhattacharyya, A., Mallick, B. K., and Baladandayuthapani, V. (2019) Data Integration and Variable Selection for Pan-Cancer Survival Prediction using Protein Expressions, Biometrics. <doi: 10.1111/biom.13132>

Examples

```
# Examples for hsaftgroupcorr function
burnin <- 50   # number of burnin
nmc    <- 100  # number of Markov Chain samples
y.sd   <- 1     # standard deviation of the data
p      <- 80    # number of covariates
```

```
r      <- 5     # number of groups
p      <- 80    # number of covariate in each group
n1     <- 40    # sample size of 1st group
n2     <- 50    # sample size of 2nd group
n3     <- 70    # sample size of 3rd group
n4     <- 100   # sample size of 4th group
n5     <- 120   # sample size of 5th group
n      <- sum(c(n1, n2, n3, n4, n5))  # total sample size
n.seq  <- c(n1, n2, n3, n4, n5)
Beta   <- matrix(smoothmest::rdoublex(p * r), nrow = r, ncol = p, byrow = TRUE)
# from double exponential distribution
beta   <- as.vector(t(Beta))  # vectorize Beta
x1     <- mvtnorm::rmvnorm(n1, mean = rep(0, p))
x2     <- mvtnorm::rmvnorm(n2, mean = rep(0, p))
x3     <- mvtnorm::rmvnorm(n3, mean = rep(0, p))
x4     <- mvtnorm::rmvnorm(n4, mean = rep(0, p))
x5     <- mvtnorm::rmvnorm(n5, mean = rep(0, p))  # from multivariate normal distribution
y.mu1  <- x1 %*% Beta[1, ]
y.mu2  <- x2 %*% Beta[2, ]
y.mu3  <- x3 %*% Beta[3, ]
y.mu4  <- x4 %*% Beta[4, ]
y.mu5  <- x5 %*% Beta[5, ]
y1     <- stats::rnorm(n1, mean = y.mu1, sd = y.sd)
y2     <- stats::rnorm(n2, mean = y.mu2, sd = y.sd)
y3     <- stats::rnorm(n3, mean = y.mu3, sd = y.sd)
y4     <- stats::rnorm(n4, mean = y.mu4, sd = y.sd)
y5     <- stats::rnorm(n5, mean = y.mu5, sd = y.sd)
y      <- c(y1, y2, y3, y4, y5)
x      <- Matrix::bdiag(x1, x2, x3, x4, x5)
X      <- as.matrix(x)
y      <- as.numeric(as.matrix(y))  # from normal distribution
T      <- exp(y)   # AFT model
C      <- rgamma(n, shape = 1.75, scale = 3)  # censoring time
time   <- pmin(T, C)  # observed time is min of censored and true
status = time == T   # set to 1 if event is observed
ct     <- as.matrix(cbind(time = time, status = status))  # censored time
posterior.fit <- hsaftgroupcorr(ct, X, method.tau = "truncatedCauchy", method.sigma = "Jeffreys",
                                burn = burnin, nmc = nmc,
                                r = r, n.seq = n.seq, pk = p)
summary(posterior.fit$BetaHat)
```

Index