Package ‘intsurvbin’

October 3, 2019

Title Survival and Binary Data Integration
Version 0.0.4
Description Function to implement the horseshoe shrinkage prior in integrated survival and binary regression as developed in Maity et. al. (2019) <doi:10.1111/rssc.12377>.
Depends R (>= 2.1.1)
Imports msm, stats, tmvtnorm, mvtnorm, MHadaptive, mgcv
Suggests smoothmest
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
NeedsCompilation no
Author Arnab Maity [aut, cre],
Antik Chakraborty [aut],
Anirban Bhattacharya [aut],
Raymond Carroll [aut],
Bani K. Mallick [aut]
Maintainer Arnab Maity <arnab.maity@pfizer.com>
Repository CRAN
Date/Publication 2019-10-03 06:50:09 UTC

R topics documented:

afths ................................................................. 2
aftprobitsh .......................................................... 4
Index 8
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

afths(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, Xtest = NULL,
    cttest = NULL)

Arguments

c\_t survival response, a \( n \times 2 \) 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 \( n \times p \).

method.tau Method for handling \( \tau \). 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 \( \tau \) 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 Jeffreys's prior on the error variance \( \sigma^2 \), or "fixed" to use a fixed value (an empirical Bayes estimate, for example).

Sigma2 A fixed value for the error variance \( \sigma^2 \). 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.

Xtest test design matrix.

c\_ttest test survival response.
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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurvivalHat</td>
<td>Predictive survival probability</td>
</tr>
<tr>
<td>LogTimeHat</td>
<td>Predictive log time</td>
</tr>
<tr>
<td>Beta.sHat</td>
<td>Posterior mean of Beta, a ( p ) by 1 vector</td>
</tr>
<tr>
<td>LeftCI.s</td>
<td>The left bounds of the credible intervals</td>
</tr>
<tr>
<td>RightCI.s</td>
<td>The right bounds of the credible intervals</td>
</tr>
<tr>
<td>Beta.sMedian</td>
<td>Posterior median of Beta, a ( p ) by 1 vector</td>
</tr>
<tr>
<td>LambdaHat</td>
<td>Posterior median of ( \lambda ), a ( p \times 1 ) vector</td>
</tr>
<tr>
<td>Sigma2Hat</td>
<td>Posterior mean of error variance ( \sigma^2 ). If method.sigma = &quot;fixed&quot; is</td>
</tr>
<tr>
<td></td>
<td>used, this value will be equal to the user-selected value of Sigma2 passed</td>
</tr>
<tr>
<td></td>
<td>to the function</td>
</tr>
<tr>
<td>TauHat</td>
<td>Posterior mean of global scale parameter tau, a positive scalar</td>
</tr>
<tr>
<td>Beta.sSamples</td>
<td>Posterior samples of ( \beta )</td>
</tr>
<tr>
<td>TauSamples</td>
<td>Posterior samples of ( \tau )</td>
</tr>
<tr>
<td>Sigma2Samples</td>
<td>Posterior samples of Sigma2</td>
</tr>
</tbody>
</table>

References


Examples

```r
burnin <- 500
nmc <- 500
thin <- 1
y.sd <- 1  # standard deviation of the response

p <- 100  # number of predictors
ntrain <- 100  # training size
ntest <- 50  # test size
n <- ntest + ntrain  # sample size
q <- 10  # number of true predictors

beta.t <- c(sample(x = c(1, -1), size = q, replace = TRUE), rep(0, p - q))  # randomly assign sign
Sigma <- matrix(0.9, nrow = p, ncol = p)
for(j in 1:p)
{
 Sigma[j, j] <- 1
}
```
aftprobiths

Horseshoe shrinkage prior in integrated survival and binary regression

Description

This function provides the implementation of integrated survival and binary high dimensiona regression utilizing Horseshoe prior on the parameters.

Usage

aftprobiths(ct, z, X, burn = 1000, nmc = 5000, thin = 1, alpha = 0.05, Xtest = NULL, cttest = NULL, ztest = NULL)
Arguments

- **ct**: survival response, a $n \times 2$ matrix with first column as response and second column as right censored indicator, 1 is event time and 0 is right censored.
- **z**: binary response, a $n \times 1$ vector with numeric values 0 or 1.
- **X**: Matrix of covariates, dimension $n \times p$.
- **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.
- **Xtest**: test design matrix.
- **cttest**: test survival response.
- **ztest**: test binary response.

Value

- **Beta.sHat**: Posterior mean of $\beta$ for survival model, a $p$ by 1 vector.
- **Beta.bHat**: Posterior mean of $\beta$ for binary model, a $p$ by 1 vector.
- **LeftCI.s**: The left bounds of the credible intervals for Beta.sHat.
- **RightCI.s**: The right bounds of the credible intervals for Beta.sHat.
- **LeftCI.b**: The left bounds of the credible intervals for Beta.bHat.
- **RightCI.b**: The right bounds of the credible intervals for Beta.bHat.
- **Beta.sMedian**: Posterior median of $\beta$ for survival model, a $p$ by 1 vector.
- **Beta.bMedian**: Posterior median of $\beta$ for binary model, a $p$ by 1 vector.
- **SigmaHat**: Posterior mean of variance covariance matrix.
- **LambdaHat**: Posterior mean of $\lambda$, a $p \times 1$ vector.
- **TauHat**: Posterior mean of $\tau$, a $2 \times 1$ vector.
- **Beta.sSamples**: Posterior samples of $\beta$ for survival model.
- **Beta.bSamples**: Posterior samples of $\beta$ for binary model.
- **LambdaSamples**: Posterior samples of $\lambda$.
- **TauSamples**: Posterior samples of $\tau$.
- **SigmaSamples**: Posterior samples of variance covariance matrix.
- **DIC.s**: DIC for survival model.
- **DIC.b**: DIC for binary model.
- **SurvivalHat**: Predictive survival probability.
- **LogTimeHat**: Predictive log time.

References

Examples

```r
burnin <- 50
nmc <- 150
thin <- 1
y.sd <- 1  # standard deviation of the response

p <- 100  # number of predictors
ntrain <- 100  # training size
ntest <- 50  # test size
n <- ntest + ntrain  # sample size
q <- 10  # number of true predictors

beta.t <- c(sample(x = c(1, -1), size = q, replace = TRUE), rep(0, p - q))  # randomly assign sign
Sigma <- matrix(0.9, nrow = p, ncol = p)
for(j in 1:p)
{
  Sigma[j, j] <- 1
}
x <- mvtnorm::rmvnorm(n, mean = rep(0, p), sigma = Sigma)  # correlated design matrix
zmean <- x %*% beta.t
tmean <- x %*% beta.t
yCorr <- 0.5
yCov <- matrix(c(1, yCorr, yCorr, 1), nrow = 2)

y <- mvtnorm::rmvnorm(n, sigma = yCov)
t <- y[, 1] + tmean
z <- ifelse((y[, 2] + zmean) > 0, 1, 0)
X <- scale(as.matrix(x))  # standardization
z <- as.numeric(as.matrix(c(z)))
t <- as.numeric(as.matrix(c(t)))
T <- exp(t)  # AFT model
C <- rgamma(n, shape = 1.75, scale = 3)  # 42% censoring time
time <- pmin(T, C)  # observed time is min of censored and true
status <- time == T  # set to 1 if event is observed
cj <- as.matrix(cbind(time = time, status = status))  # censored time

# Training set
ztrain <- z[1:ntrain]
cctrain <- cj[1:ntrain, ]
Xtrain <- X[1:ntrain, ]

# Test set
ztest <- z[(ntrain + 1):n]
cctest <- cj[(ntrain + 1):n, ]
Xtest <- X[(ntrain + 1):n, ]
```
posterior.fit.joint <- aftprobiths(ct = cttrain, z = ztrain, X = Xtrain,
    burn = burnin, nmc = nmc, thin = thin,
    Xtest = Xtest, cttest = cttest, ztest = ztest)

posterior.fit.joint$Beta.sHat
Index

afths, 2
aftprobiths, 4