

Vignette to fit the main model

2024-01-09

R Markdown

This is a vignette document on how to run the code associated with the article Maity, A. K., Pati, D., and Mallick, B. K. (2024). Bayesian Time-to-event Density Regression with Application to High Dimensional Data

```
require(mvtnorm)    # for multivariate normal distribution
```

```
## Loading required package: mvtnorm
```

```
library(msm)        # for truncated normal distribution
```

```
require(tmvtnorm)    # for truncated multivariate normal distribution
```

```
## Loading required package: tmvtnorm
```

```
## Loading required package: Matrix
```

```
## Loading required package: stats4
```

```
## Loading required package: gmm
```

```
## Loading required package: sandwich
```

```
require(truncdist)   # for random sample from truncated Gamma distribution
```

```
## Loading required package: truncdist
```

```
## Loading required package: evd
```

```
require(MHadaptive) # for makePositivedefinite function
```

```
## Loading required package: MHadaptive
```

```
## Loading required package: MASS
```

```
require(foreach)     # for foreach
```

```
## Loading required package: foreach
```

```
require(doParallel)  # for cluster creation
```

```
## Loading required package: doParallel
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
require(SuppDists)    # for rinuGauss function
```

```
## Loading required package: SuppDists
```

```
require(ipred)        # for integrated Brier Score
```

```
## Loading required package: ipred
```

```

require(survival)    # for Surv function

## Loading required package: survival
require(ggplot2)     # for confidence band

## Loading required package: ggplot2

n <- 150 # sample size
p <- 200 # number of covariates
x <- rmvnorm(n, mean = rep(0, p)) # design matrix
k <- 5 # number of true variables
beta.t <- c(sample(c(1, -1), size = k, replace = TRUE), rep(0, p - k))
linear.part <- as.vector(x %*% beta.t)
sigma.square <- 1

c <- 0.1
d <- 2

# Set the parameters
nburnin <- 1000
nmc <- 1000
thin <- 1
N <- 61
sig <- 2.5
nu <- 0.1
knots <- seq(from = c, to = d, length.out = N)
CM <- covariance_matrix(knots, sig, nu) # covariance matrix of g

# Data generation
xi <- rnorm(n, mean = linear.part, sd = sqrt(sigma.square)) # response
a <- 1
b <- min(xi) - 1
y <- (1/a)^(1/5) * (xi - b)^(1/5)

T <- exp(y) # AFT model
C <- rtrunc(n, spec = "gamma", shape = 1, scale = 10, a = 2) # 20% censor
time <- pmin(T, C) # observed time is min of censored and true
status = time == T # set to 1 if event is observed
1 - sum(status)/length(y) # censoring rate

## [1] 0.2066667

censor.rate <- 1 - sum(status)/length(y) # censoring rate
censor.rate

## [1] 0.2066667

ct <- as.matrix(cbind(time = time, status = status)) # censored time
X <- scale(x)

c <- min(log(time)) - 0.1
d <- max(log(time)) + 0.1
knots <- seq(from = c, to = d, length.out = N)
CM <- covariance_matrix(knots, sig, nu) # covariance matrix of g

```

```

# Training set and Test set
ntrain      <- 100
ntest       <- n - ntrain
cttrain     <- ct[1:ntrain, ]
cttest      <- ct[(ntrain + 1):n, ]
test.index1 <- which(cttest[, 2] == 1) # which are NOT censored
cttest1     <- cttest[test.index1, ]
Xtrain      <- X[1:ntrain, ]
Xtest       <- X[(ntrain + 1):n, ]

```

The following will fit the model and provide the significant variables. However, the function `gph()` and the file `helper.R` has to be loaded in the R console before running the following.

```

gph.fit <- gph(ct = cttrain, X = Xtrain, nburnin = nburnin, nmc = nmc, thin = thin, method.mcmc = "MH",

```

```

## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000
## [1] 1100
## [1] 1200
## [1] 1300
## [1] 1400
## [1] 1500
## [1] 1600
## [1] 1700
## [1] 1800
## [1] 1900
## [1] 2000

```

```

beta.post.gp <- gph.fit$BetaHat

cluster      <- kmeans(abs(beta.post.gp), centers = 2)$cluster # return cluster indices
cluster1     <- which(cluster == 1)
cluster2     <- which(cluster == 2)
min.cluster  <- ifelse(length(cluster1) < length(cluster2), 1, 2)
discovery    <- which(cluster == min.cluster)
discovery    # significant variables

```

```

## [1] 1 2 3 4 5

```