Package 'brea'

October 6, 2016

Version 0.1.0

Date 2016-09-30

Title Bayesian Recurrent Event Analysis

Description A function to produce MCMC samples for posterior inference in semiparametric Bayesian discrete time competing risks recurrent events models.

Author Adam J King

Maintainer Adam J King <king@cpp.edu>

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-06 13:06:07

R topics documented:

`brea_mcmc` *Bayesian Discrete Survival Inference*

Description

This function performs Metropolis-Hastings exploration of the posterior distribution for Bayesian discrete time survival models. The models may have competing risks, semiparametric covariate effects (like arbitrary baseline hazards), and random effects shared across repeated events and correlated across competing risks.

Usage

```
brea_mcmc(x, y, S = 1000L, priors = NULL, n = NULL, K = NULL, store_re = FALSE)
```

Arguments

x	an integer matrix or dataframe all columns of which are factors specifying the values of the (discretized) predictors at each person-time point; specifically, the (i,j) entry is the value of predictor j at discrete person-time observation i
y	an integer matrix whose (i,j) entry counts the number of events of type j occurring at discrete person-time observation i
S	the number of MCMC iterations to perform
priors	a list with one element for each predictor variable (column of x) specifying the prior type to use for that predictor; see Details for more information
n	a vector of positive integers with length equal to the number of person-time observations whose entries equal the number of replicated observations that row stands for (defaults to 1 for each observation)
K	a vector of positive integers with length equal to the number of predictors giving the number of distinct values each discretized predictor may assume; this is not used if a dataframe of factors is provided for x
store_re	if TRUE, the random effects are stored from each MCMC iteration, and if FALSE they are not stored

Details

The data provided to the brea_mcmc function is specified at the person-time level: there is one row in x and y for each discrete time point each person or thing was at risk for event occurrence. All predictors in x must be encoded as factors (or their corresponding integer codes in the case that x is an integer matrix). The underlying type of predictor is specified in the priors argument, which is a list with one element for each predictor variable which specifies both the type of that predictor and the prior distribution to use. The allowed predictor types are:

- "cat" for categorical variables. The first element of the prior specification list is the string "cat", and second element is a positive conditional prior standard deviation parameter.

- "gmrf" for underlying continuous predictors; continuous predictors should be cut() before being included into x; Gaussian Markov random field (GMRF) priors are then used to smooth the effects of adjacent categories of the discretized continuous predictor. The first element of the prior specification list is the string "gmrf", the second element is a prior degrees of freedom for the scaled inverse chi-squared prior for the random walk increment variance, and the third element is a prior scale for the scaled inverse chi-squared.

- "re" for variables (like subject id numbers) that represent random effects. The first element of the prior specification list is the string "re", the second element is a prior degrees of freedom for an inverse Wishart prior for the random effects covariance, and the third element is a prior scale matrix for the random effects covariance.

- "zero" for predictors that are not used in the current MCMC run. This is provided as a convenient way to exclude predictors from certain runs. The first and only element of the prior specification list is the string "zero".

Value

A list providing the values from each of the S MCMC iterations for the intercept, the other linear predictor parameters, standard deviations of the GMRF random walk increments, and covariances of the random effects:

b_0_s	an R by S matrix (where R is the number of competing risks) whose (r,s) entry is the intercept for risk r in iteration s
b_m_s	a list whose m^th element is the R by K[m] by S array whose (r,k,s) entry is the value of the linear predictor contribution of value k for predictor m on risk r at MCMC iteration s
s_m_s	a list whose m^th element is, in the case of a GMRF prior, the R by S matrix whose (r,s) entry is the random walk standard deviation for predictor m and competing risk r at MCMC iteration s, or, in the case of a random effects priors, the R by R by S array whose (,,s) entry is the random effects covariance matrix for predictor m at MCMC iteration s
b_m_a	a list whose m^th element is the length K[m] vector giving the number of accepted Metropolis proposals for each value of predictor m across all S MCMC iterations

Author(s)

Adam King

Examples

```
# leukemia remission times data from Table 1 of "Regression Models and
# Life-Tables" (Cox, 1972)

# times of event occurrence or right censoring:
study_time <- c(6,6,6,6,7,9,10,10,11,13,16,17,19,20,22,23,25,32,32,34,35,
                1,1,2,2,3,4,4,5,5,8,8,8,8,11,11,12,12,15,17,22,23)

# event/censoring indicator (1 for event occurrence, 0 for right censoring):
cens <- c(0,1,1,1,1,0,0,1,0,1,1,0,0,0,1,1,0,0,0,0,0,
          1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)

# treatment assignment (1 for treatment, 2 for control):
treat <- c(rep(1L,21),rep(2L,21))

# total number of person-time observations among all subjects:
N <- sum(study_time)

# create and fill in person-time predictor and event indicator matrices:
x <- matrix(0L,N,2)  # first column is time, second column is treatment
y <- matrix(0L,N,1)
next_row <- 1  # next available row in the person-time matrices
for (i in seq_along(study_time)) {
  x[next_row:(next_row + study_time[i] - 1),] <- cbind(seq_len(study_time[i]),
                                                       rep(treat[i],study_time[i]))
```

```
  y[next_row:(next_row + study_time[i] - 1),] <- c(rep(0L,study_time[i] - 1),
                                                   as.integer(cens[i]))
  next_row <- next_row + study_time[i]
}

# group the time variable into 6-week intervals:
x[,1] <- cut(x[,1],seq(0,36,6),labels=FALSE)

# use GMRF prior for time, and categorical prior for treatment group:
priors <- list(list("gmrf",3,.01),list("cat",2))

# run 1,100 MCMC iterations:
fit <- brea_mcmc(x, y, 1100, priors)

# look at the structure of the returned posterior samples and acceptance counts:
str(fit)

# approximate posterior mean hazard ratio:
ss <- 101:1100  # use last 1,000 samples, discarding first 100 as burn-in
exp(2*mean(fit$b_m_s[[2]][1,2,ss]))
# hazard ratio of approximately 5
```

Index