Package 'MRH'

February 26, 2016

Type Package

Title Multi-Resolution Estimation of the Hazard Rate

Version 2.2

Date 2016-02-23

Author Yolanda Hagar, Yuanting Chen, Vanja Dukic

Maintainer Yolanda Hagar <yolanda.hagar@colorado.edu>

Description Used on survival data to jointly estimate the hazard rate and the effects of covariates on failure times. Can accommodate covariates under the proportional and non-proportional hazards setting, and is ideal for analysis of survival data with long-term follow-up.

License GPL-2

Depends survival, KMsurv

Suggests R.rsp

VignetteBuilder R.rsp

Imports coda

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-02-26 17:29:11

R topics documented:

MRH-package *Multi-resolution estimation of the hazard rate.*

Description

This package uses the multi-resolution hazard estimator for estimation and inference for the hazard rate. The multi-resolution hazard estimator is a Polya tree-based Bayesian semi-parametric method for estimating the hazard rate jointly with covariate effects. This methodology splits the hazard rate into 2^M bins, and estimates a constant hazard rate within each bin using a tree-like structure, providing robust estimates of the hazard rate even through periods of sparse observations. This package allows for covariates in the model with or without the proportional hazards assumption.

Details

Package:	MRH
Type:	Package
Version:	1.0
Date:	2014-04-03
License:	GPL (>=2)

Author(s)

Yolanda Hagar, Yuanting Chen, Vanja Dukic

Maintainer: Yolanda Hagar <yolanda.hagar@colorado.edu>

References

Bouman, P., Dukic, V., Meng, X.L. (2005), Bayesian multiresolution hazard model with application to an AIDS reporting delay study. *Statistica Sinica*. **15,** 325–357.

Bouman, P., Dignam, J., Dukic, V. (2007), A multiresolution hazard model for multi-center survival studies: Application to Tamoxifen treatment in early stage breast cancer. *JASA*. **102,** 1145–1157.

Dukic, V., Dignam, J. (2007), Bayesian hierarchical multiresolution hazard model for the study of time-dependent failure patterns in early stage breast cancer. *Bayesian Analysis*. **2,** 591–610.

Hagar, Y., Albers, D., Pivavarov, R., Chase, H., Dukic, V., Elhadad, N. (2014), Survival analysis with Electronic Health Record data: Experiments with Chronic Kidney Disease. *Statistical Analysis and Data Mining*. **7,** 385–403.

Chen, Y., Hagar, Y., Dignam, J., Dukic, V. (2014), Pruned Multiresolution Hazard (PMRH) models for time-to-event data. *In review.* Available upon request via email to vanja.dukic@colorado.edu.

This work utilized the Janus supercomputer, which is supported by the National Science Foundation (award number CNS-0821794) and the University of Colorado Boulder. The Janus supercomputer

is a joint effort of the University of Colorado Boulder, the University of Colorado Denver and the National Center for Atmospheric Research.

Yolanda Hagar and Vanja Dukic were supported in part by grants NSF-DEB 1316334 and NSF-GEO 1211668.

`AnalyzeMultiple` *ANALYZE MULTIPLE MCMC CHAINS*

Description

Analyzes multiple MCMC chains and returns the median and alpha-level quantiles of the marginal posterior distribution for each parameter. The Gelman-Rubin test can be employed to check for convergence if the initial values for the parameter chains vary across the sample space.

Usage

```
AnalyzeMultiple(datalist, fileNames, alpha.level, maxStudyTime, GR = TRUE)
```

Arguments

`datalist`	A list object that contains one set of MCMC chains per list element.
`fileNames`	If datalist is empty, a list of filenames entered as a string of character names can be entered here.
`alpha.level`	The alpha-level bounds for the credible intervals for the parameter estimates.
`maxStudyTime`	The maximum observed or censored failure time, or the end of the study period.
`GR`	A TRUE or FALSE value denoting whether the Gelman-Rubin diagnostic convergence test should be performed on the chains. Default is TRUE, but can be set to FALSE if the user only wishes to obtain estimates across multiple data sets. If the Gelman-Rubin test is employed, the chains must have the same burn-in, thinning value, and maximum number of iterations, and the initialized parameter values must cover the parameter space. The most efficient way to ensure this is to set GR = TRUE when fitting the MRH object using the estimateMRH() function.

Details

If the Gelman-Rubin test is used, the MCMC chains must satisfy the assumptions needed for the test and the initial parameter values used to estimate the posteriors must be dispersed across the parameter space. In addition, the burn-in number, thinning value, and maximum number of iterations must be equal across all chains. This can be enforced in estimateMRH() by setting GR equal to TRUE. Because setting GR equal to TRUE fixes the burn-in number, thinning value, and maximum number of iterations, it is critical that the user enter values that are sufficient for convergence, while optimizing the run time for the fitted MRH model.

Value

hazardRate	The medians and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for the hazard rate within each bin.
beta	The medians and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for each covariate.
SurvivalCurve	The medians and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for the survival curve.
CumulativeHazard	The medians and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for the cumulative hazard over the course of the study.
d	The estimated cumulative hazard rate increments for each of the 2^M bins, with the medians and (alpha/2, 1-alpha/2) quantiles of the posterior distribution for each d reported.
H	The median and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for the cumulative hazard H at the end of the study.
Rmp	The medians and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for each of the split parameters (Rmp).
gamma	The medians and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for each of the gamma parameters (used in the posterior of the Rmp split parameters). This is only returned to the user if the gamma parameters were sampled in the original MCMC sampling routine.
k	The medians and alpha/2% and 1-alpha/2% quantiles of the posterior distribution for k (used in the posterior of the Rmp split parameters). This is only returned to the user if k was sampled in the original MCMC sampling routine.
gelman.rubin	If option is set to TRUE, the results of the Gelman-Rubin test. The chains have converged of the scale reduction factor is "close enough" to 1.

Author(s)

Yolanda Hagar

Examples

```
# Generate 3 chains for the same model.
# Set GR = TRUE so that the burn-in number,
# thinning value,
# and maximum number of iterations are the same
# across all MCMC chains, and so that initialized
# parameter values cover the parameter space.

# Note that the routine may produce a warning
# message that the algorithm has not converged,
# as typically more iterations are needed for convergence.
# However, for the purposes of this example, the number
# of iterations is sufficient.

data(cancer)
```

```
cancer$censorvar = cancer$status - 1
## Not run:
fit.lung1 = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, outfolder = 'MRH_lung1', GR = TRUE)
fit.lung2 = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, outfolder = 'MRH_lung2', GR = TRUE)
fit.lung3 = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, outfolder = 'MRH_lung2', GR = TRUE)
## End(Not run)

# Calculate the results of all three chains and the
# scale reduction factor entering the data sets in a list
## Not run:
AnalyzeMultiple(datalist =
list(read.table('MRH_lung1/MCMCchains.txt', header = TRUE),
read.table('MRH_lung2/MCMCchains.txt', header = TRUE),
read.table('MRH_lung2/MCMCchains.txt', header = TRUE)),
maxStudyTime = 960)
## End(Not run)

# Calculate the results of all three chains and the
# scale reduction factor entering the data file names
## Not run:
AnalyzeMultiple(fileNames = c('MRH_lung1/MCMCchains.txt',
'MRH_lung2/MCMCchains.txt', 'MRH_lung2/MCMCchains.txt'),
maxStudyTime = 960)
## End(Not run)
```

`CalcFunction` *CALCFUNCTION CALCULATES THE HAZARD RATE, CUMULATIVE HAZARD AND/OR SURVIVAL FUNCTION OF AN MRH OBJECT*

Description

This function calculates the hazard rate, cumulative hazard and/or the survival function of an MRH object. The alpha-level bounds are included in the calculation.

Usage

```
CalcFunction(mrhobject, function.type = c("h", "H", "S"),
maxStudyTime, alpha.level = 0.05)
```

Arguments

mrhobject	An MRH object: either the fitted model from the estimateMRH() routine or the text file of MCMC chains converted to an MRH object.
function.type	The function the user would like returned, with 'h' denoting the hazard rate, 'H' denoting the cumulative hazard, and 'S' denoting the survival function. The user may request any subset of these functions, and multiple functions can be requested and returned.
maxStudyTime	The maximum observed or censored failure time, or the end of the study period. This is only needed if the text file of the chains are used instead of the actived fitted MRH object.
alpha.level	The alpha.level for the bounds of the credible intervals.

Value

hazardrate	The hazard rate and alpha-level bounds. Returned to the user if 'h' is entered in function.type.
cumulhaz	The cumulative hazard functions and alpha-level bounds. Returned to the user if 'H' is entered in function.type.
survfunction	The survival function and alpha-level bounds. Returned to the user if 'S' is entered in function.type

Author(s)

Yolanda Hagar <yolanda.hagar@colorado.edu>

Examples

```
# The MRH fit portion of the example is from
# the estimateMRH() help page.  They do not need to
# be re-run if the code has previously
# been run and the outfolder ('MRH_lung') is saved.
data(cancer)
cancer$censorvar = cancer$status - 1
## Not run:
fit.lung = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200,
maxIter = 1000, thin = 1, outfolder = 'MRH_lung')
## End(Not run)

# Get the cumulative hazard with 99% credible interval bounds
## Not run:
CalcFunction(fit.lung, function.type = 'H',
alpha.level = .01)
## End(Not run)

# Get the hazard rate, cumulative hazard, and
# survival function using the MCMC chains text file
## Not run:
```

```
lung.chains = MRH(read.table('MRH_lung/MCMCchains.txt', header = TRUE))
## End(Not run)

# When the chains are read in from a file and used,
# the maximum study time
# must be entered to calculate the hazard rate.
# (i.e. CalcFunction(lung.chains) does not work)
## Not run:
CalcFunction(lung.chains, maxStudyTime = 960)
## End(Not run)
```

DIC *DIC CALCULATES INFORMATION CRITERION GIVEN THE MCMC CHAINS FROM THE estimateMRH routine.*

Description

The DIC function calculates the Deviance Information Criterion given the MCMC chains from an estimateMRH routine, using the formula: DIC = .5*var(D)+mean(D), where D is the chain of -2*log(L), calculated at each retained iteration of the MCMC routine. It also provides the Aikaike Information Criterion (AIC) = 2*p + D and the Bayesian Information Criterion (BIC) = p*ln(n) + D, where 'p' is the number of parameters in the model. Both AIC and BIC report the maximum (i.e. "worst") values in the chain.

Usage

```
DIC(mrhobject, n)
```

Arguments

mrhobject	The chains found in the MCMCchains.txt file, created using the estimateMRH routine, or the MRH results object.
n	The sample size of the original dataset. If n is not entered, the BIC calculation will not be returned to the user.

Details

The number of parameters 'p' is calculated as 2^M (one for each split parameter Rmp, and one for the cumulative hazard at H), plus 2 for a and lambda (parameters in the Gamma prior for H), and one for each covariate included under the proportional hazards assumption. If k and/or gamma are sampled (parameters in the prior for Rmp), the number of estimated parameters is increased by 1 for k and 2^M-1 for gamma. If a covariate is included under the non-proportional hazards assumption, the number of estimated parameters (excluding any covariates included under the proportional hazards assumption) is multiplied by the number of strata in the non-proportional covariate.

Value

DIC returns the DIC, AIC, and BIC values, as well as a summary of D (-2*loglike).

`loglik.summ`	The summary of the chain of -2*loglike values.
`ICtable`	Table containing the DIC, AIC and BIC values.

Author(s)

Yolanda Hagar <yolanda.hagar@colorado.edu> and Vanja Dukic

References

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., Rubin, D. (2004) *Bayesian Data Analysis*.

Spiegelhalter, D.J., Best, N.G., Carlin, B.P., van der Linde, A. (2002), Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*. **64,** 583–639.

Examples

```
# These MRH fit portion of the examples
# are from the estimateMRH() help page.
# They do not need to be re-run if the
# objects are already in the active workspace.
data(cancer)
cancer$censorvar = cancer$status - 1
## Not run:
fit.lung = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, outfolder = 'MRH_lung')

fit.lung.prune = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, prune = TRUE, outfolder = 'MRH_lung_prune')
## End(Not run)

# Compare the DIC of the pruned and unpruned models.
# The sample size must be entered for calculation of BIC.
# This number can be found in the ``MCMCInfo.txt" file
# in the output folder.
## Not run:
DIC(fit.lung, n = 227)
DIC(fit.lung.prune, n = 227)
## End(Not run)
```

`estimateMRH` *Joint estimation of the hazard rate and covariate effects using multi-resolution hazard methodology*

Description

estimateMRH is a function used to jointly estimate the hazard rate for time-to-event data using the multi-resolution hazard method. Covariates can be included with or without the proportional hazards assumption.

Usage

```
estimateMRH(formula, data, M, numberBins, maxStudyTime, outfolder = "MRHresults",
prune = FALSE, prune.alpha = 0.05, prune.levels = NULL, burnIn = 50000,
maxIter = 5e+05, thin = 10, Rmp.init = NULL, a.init = 10, lambda.init,
beta.init = NULL, k.fixed, gamma.fixed, GR = FALSE, convGraphs = TRUE,
fix.burnIn = FALSE, fix.thin = FALSE, fix.max = FALSE, continue.chain = FALSE)
```

Arguments

`formula`	A formula object, with the response on the left of a ~ operator, and the covariates on the right. The response must be a survival object as returned by the Surv function. The non-proportional covariates can be included in the formula by using the function nph(). (See Example.)
`data`	A data frame in which to interpret the variables named in the formula.
`M`	The value of M that dictates the time "resolution" of the hazard estimate, approximated via 2^M hazard increments.
`numberBins`	An optional value that can be used instead of 'M' to dictate the number of bins. This value must be a power of 2.
`maxStudyTime`	Allows the user to set the end of the study time to something other than the maximum observed or censored failure time. This is typically used if a specific bin width is desired.
`outfolder`	The name given to the folder (subdirectory of the working directory) that stores the output for the MRH estimation routine (output includes the MCMC chains text file, the convergence graphs, and hazard rate graphs). Note that this folder will be automatically placed in the working directory. A pathname for the folder may also be specified, but the path must also be accessible from the working directory.
`prune`	If set to TRUE, two adjacent bins that are constructed via the same split parameter (Rm,p, or 1-Rm,p) are merged if the estimated hazard rate in these two bins are statistically similar. Pruning hypothesis tests (with the null hypothesis that the two bins have equal hazard rates) are performed using a modified Fisher's exact test, based on the 2 by 2 table composed of the number of failures within the time interval and at-risk patients at the end of the time interval, for each pair of adjacent bins sharing a split parameter. If the test fails to reject the

null hypothesis at the alpha level, that split parameter (Rm,p) is set to 0.5. If a non-proportional covariate is used, each hazard rate is pruned separately. The user may also enter a pruning vector of their choice, which must contain a 0 or 1 for each Rmp in the model. For the non-proportional hazards model, the user-entered pruning indicator can either be a vector, which prunes all hazard rates the same, or it can be a matrix with each column containing a pruning vector for each hazard rate group, allowing the pruning for each hazard rate to be different. To make a user-defined pruning vector, please email Vanja Dukic <vanja.dukic@colorado.edu> or Yolanda Hagar <yolanda.hagar@colorado.edu>.

`prune.alpha` The significance level for Fisher's exact test if the "prune" option is set to TRUE. The default is alpha = 0.05.

`prune.levels` If pruning is used, the number of desired levels to be pruned (ranges from 1 to M). The default to prune all M levels. For multiple hazards, a single value or a vector with different pruning levels may be entered, with one value for each hazard.

`burnIn` The number of iterations in the burn-in for the MCMC chain. Default is 50,000. See details.

`maxIter` The maximum number of iterations in the MCMC routine. Default is 1,000,000.

`thin` The thinning parameter that denotes the thinning of the MCMC chain to reduce autocorrelation. Default value is 10. See Details.

`Rmp.init` The initial values for Rmp (split) parameters. If no values are entered, the default initial values is 0.5 for all Rmp.

`a.init` The initial value for "a", a parameter in the (Gamma) prior for the baseline cumulative hazard H. If no value is entered, the default value is 10.

`lambda.init` The initial value for "lambda", a parameter in the (Gamma) prior for the baseline cumulative hazard H. If no value is entered, the default value is -log(mean(delta))/a, where delta is the vector of censoring indicators. This initialization of lambda approximates the cumulative hazard rate divided by the value of a.

`beta.init` The initial value(s) for the beta (covariate effect) parameter(s). If no value is entered, the default values are obtained from the Cox proportional hazards estimates.

`k.fixed` Can either be a boolean indicator or a numeric value assigned to k, a parameter in the Beta prior for the split parameters Rmp. By default (or if set equal to TRUE), k is fixed at 0.5, implying zero prior correlation among the hazard increments. The user may also specify the fixed value of k by entering a numeric value for k.fixed greater than 0. When k is greater than 0.5, the increments are positively correlated a priori. Correspondingly, for k less than 0.5, the hazard increments will have negative prior correlation. If k.fixed is set to FALSE, then k will be sampled in the MCMC routine.

`gamma.fixed` Can either be a boolean indicator or a numeric vector of length 2^M-1 for values assigned to the gamma parameters (parameters in the Beta prior for the split parameters Rmp), with one gamma.mp associated with each Rmp, and E(Rmp) = gamma.mp. The gamma parameter allows the user to a priori "center" the baseline hazard in each bin. If gamma.fixed is omitted or set equal to TRUE, the values are fixed at 0.5, assuming centering occurs in the middle of the bin. The

user may also specify values (between 0 and 1) for each gamma. If gamma.fixed is set to FALSE, then each gamma parameter will be sampled in the MCMC routine.

`GR` Denotes whether the Gelman-Rubin test statistic will be applied to the results of the MCMC chain. See Details.

`convGraphs` Allows user to specify if graphs for convergence should be saved. Includes trace plots, density plots, and running means. Default is TRUE.

`fix.burnIn` If set equal to TRUE, the specified burn-in number (i.e. the ""burnIn"" value) will be fixed throughout the routine, and will not be increased, regardless of the convergence diagnostics performed by the routine. See details.

`fix.thin` If set equal to TRUE, the specified thinning value (i.e. the "thin" value) will be fixed throughout the routine, and will not be increased, regardless of the autocorrelation diagnostics performed by the routine. See details.

`fix.max` If set equal to TRUE, the specified maximum number of iterations (i.e. the "maxIter" value) will be fixed throughout the routine, and will not be decreased, regardless of convergence diagnostics. See details.

`continue.chain` If set to TRUE, the MCMC routine will continue running where the routine left off. The same output folder ("outfolder"") must be used, and parameters will be initialized with the last recorded sample from the MCMC chain in the previous run. Any thinning or burn-in specifications made in the model call will be ignored, and only values from the text file containing the chains will be used. The maximum number of new iterations is specified by "maxIter", and the new MCMC chains are appended to the existing text file in the output folder.

Details

This function returns the estimate of the hazard rate using the Multi-Resolution Hazard (MRH) method. The user must have the survival time and censoring variable for each subject. Parameters are estimated using MCMC.

After the first 100,000 MCMC iterations, the results are checked to determine an appropriate thinning value. The auto-correlation estimate, available via the coda package, is examined, and the first lag with an autocorrelation below 0.10 for any parameter is taken as the new thinning value. Default or user entered burn-in and thin values are taken as the minimum possible thinning values, and the routine only checks if the values should be greater (not smaller) than those specified by the user.

Every 100,000 iterations, the results are checked to determine if there is evidence of convergence and to determine an appropriate burn-in number. This is done through the Geweke diagnostic test using the geweke.diag() function available in the coda package. The geweke.diag() function returns the resulting z-score from the Geweke test for each parameter. If any of the z-scores are outside of the 0.005 range, then convergence is assumed not to have been reached. If convergence is not reached, 20,000 more iterations are burned, and the Geweke diagnostic test is performed again. This continues until there are fewer than 1,000 retained iterations left (i.e. retained after thinning). If there is no evidence of convergence, and if there are fewer than 1,000 retained iterations, the MCMC sampler runs another 100,000 times. This continues until the maximum number of iterations (maxIter) is reached, at which point the routine stops and the results are returned to the user. If the maximum number of iterations is reached before convergence and the user would like to continue the routine, the last parameter estimates in the output file can be used as initial values in

the another call of the routine. The user has the option to fix the burn-in number, maximum number of iterations, and thinning value if fix.burnIn, fix.max, or fix.thin are set equal to TRUE. The convergence routine will not change or check these values if they are fixed.

If the user would like to run the routine multiple times on the same data set and test for convergence using the Gelman-Rubin diagnostic test, the user should specify GR = TRUE when calling the MRH estimation function. This way the initial parameter values will be randomly sampled to explore the parameter space. Additionally, this will fix the user-entered or default burn-in number, and thinning value, and the number of MCMC iterations will be fixed at the maximum number of iterations (maxIter) so that the diagnostic test can be performed. NOTE: If the user desires less than the default 1,000,000 maximum number of iterations and has set GR = TRUE, maxIter should be changed. Likewise, if fix.max is set equal to TRUE, the routine will run exactly maxIter number of times, regardless of convergence.

There may be instances in which the user may desire to combine some of the hazard rate bins, particularly if there is evidence that the hazard rate does not change from one bin to the next. In these cases, a pruning indicator can be used to denote which bins can be combined, with a "0" indicating the bin should not be combined with another bin, and a "1" indicating it should. While it is possible to specify this manually, the user can use the built-in pruning function by setting prune = TRUE, which performs Fisher's exact test to determine if bins can be combined based on observed failures and censoring in each bin. Based on the results of the hypothesis tests, certain bins may be pruned. If there are multiple hazards being calculated (i.e. the non-proportional hazards model is used), by default each hazard rate is pruned separately. The user may create their own pruning vector or matrix, although care must be taken in accounting for the tree-like structure of the Rmp parameters. For assistance with creation of a pruning vector or matrix, please contact Vanja Dukic <vanja.dukic@colorado.edu> or Yolanda Hagar <yolanda.hagar@colorado.edu>.

Value

estimateMRH returns a list with class MRH that contains the results of the MCMC algorithm. The routine also writes a file of the thinned and burned chain of MCMC iterations and saves pdf graphs of the hazard function with credible bounds in an MRH results folder (with the default title "MRHresults"). The components returned in the MRH fitted object are:

`summary`	A summary table containing the rounded estimates and central credible intervals for the estimates of the within-bin hazard rate (labeled "h.binj" for the jth bin) and the covariate effects. Estimates are calculated as the median of the thinned and burned MCMC chain, and bounds on the credible intervals are calculated as the 2.5% and 97.5% of the thinned and burned MCMC chain.
`hazardRate`	The estimate and lower and upper bounds of the 95% central credible interval for the hazard rate in each bin (labeled "h.binj" for the hazard rate of bin j).
`beta`	The estimate and lower and upper bounds of the 95% central credible interval for the covariate effects. This can include the log-ratios of the non-proportional hazards if the non-proportional assumption is used, which are labeled with "binj" following the covariate name to denote the log-ratio estimate within the jth bin.
`SurvivalCurve`	The estimate and lower and upper bounds of the 95% central credible interval for the survival curve. In the non-proportional hazards setting, separate survival curves and credible intervals are provided for each strata of the covariate.

`CumulativeHazard`
: The estimate and lower and upper bounds of the 95% central credible interval for the cumulative hazard. In the non-proportional hazards setting, separate cumulative hazard curves and credible intervals are provided for each strata of the covariate.

`d`
: The estimate and lower and upper bounds of the 95% central credible interval for the within-bin cumulative hazard (denoted as 'd') for each bin.

`H`
: The estimate and lower and upper bounds of the 95% central credible interval for the baseline cumulative hazard H at the end of the study (denoted as 'H00' or 'H(tJ)' in manuscripts).

`Rmp`
: The estimate and lower and upper bounds of the 95% central credible interval for the split parameters Rmp.

`gamma`
: The estimate and lower and upper bounds of the 95% central credible interval for the gamma parameter (used in the prior for the Rmp split parameters). This is only returned to the user if k.fixed is set equal to FALSE when fitting the model.

`k`
: The estimate and lower and upper bounds of the 95% central credible interval for the k parameters (used in the priors for the Rmp split parameters). This is only returned to the user if gamma.fixed is set equal to FALSE when fitting the model.

`AIC`
: Akaike's Information Criterion, calculated as 2p-2ln(L), where k is the number of parameters in the model and ln(L) is the minimum of the likelihood values calculated at each chain iteration. The number of parameters 'p' is calculated as 2^M (one for each split parameter Rmp, and one for the cumulative hazard at H), plus 2 for a and lambda (parameters in the Gamma prior for H), and one for each covariate included under the proportional hazards assumption. If k and/or gamma (parameters in the prior for Rmp) are sampled, the number of estimated parameters is increased by 1 for k and 2^M-1 for gamma. If a covariate is included under the non-proportional hazards assumption, the number of estimated parameters (excluding any covariates included under the proportional hazards assumption) is multiplied by the number of strata in the non-proportional covariate.

`BIC`
: Bayesian Information Criterion, calculated as -2ln(L) + p*log(n)

`DIC`
: Deviance Information Criterion, calculated as .5*var(-2*ln(L)) + mean(-2ln(L)), where ln(L) is calculated at each retained chain iteration.

`burnIn`
: The number of iterations burned during the MCMC routine.

`thin`
: The thinning parameter used to account for autocorrelation.

`TotalIters`
: The number of iterations the MCMC algorithm performed before convergence.

`convergence`
: A TRUE/FALSE indicator denoting whether there is evidence that the algorithm converged based on methods used in the MRH routine.

`gelman.rubin.used`
: Denotes whether the Gelman-Rubin option was used in the routine.

`fix.thin`
: A TRUE/FALSE indicator denoting whether the thinning value was fixed by the user.

fix.burnin	A TRUE/FALSE indicator denoting whether the burn-in value was fixed by the user.
fix.max	A TRUE/FALSE indicator denoting whether the maximum value was fixed by the user.
InitialValues	The initial values of the parameters at the start of the MCMC chain.
gamma.fixed	Returns the fixed value of gamma or FALSE if the gamma parameters are sampled.
k.fixed	Returns the fixed value of k or FALSE if k is sampled.
runtime	The amount of time the routine ran in hours
outfolder	The pathname of the folder containing the MCMC chains and the graphical output produced by the routine.
maxStudyTime	The maximum time in the study (either as a censored observation or an observed failure)

Author(s)

Yolanda Hagar <yolanda.hagar@colorado.edu>

References

P. Bouman, J. Dignam, V. Dukic, XL. Meng. (2005) Bayesian multiresolution hazard model with application to an aids reporting delay study. *Statistica Sinica*, **102**, 1145–1157.

Bouman, P., Dignam, J., Dukic, V. (2007), A multiresolution hazard model for multi-center survival studies: Application to Tamoxifen treatment in early stage breast cancer. *JASA*. **102,** 1145–1157.

Dukic, V., Dignam, J. (2007), Bayesian hierarchical multiresolution hazard model for the study of time-dependent failure patterns in early stage breast cancer. *Bayesian Analysis*. **2,** 591–610.

Chen, Y., Hagar, Y., Dignam, J., Dukic, V. (2014), Pruned Multiresolution Hazard (PMRH) models for time-to-event data. *In review.* Available upon request via email to vanja.dukic@colorado.edu.

http://amath.colorado.edu/faculty/vdukic/software/MRH.html

Examples

```
#####################################################
# NOTE: Examples may take a few minutes, so please be
# patient. In addition, warning messages about
# convergence may appear as more iterations are
# typically needed for chain convergence.

########### PROPORTIONAL HAZARDS EXAMPLE ############
# Examine the NCCTG lung cancer data set (from the survival package),
# and quantify how age, gender, and physician rated
# Karnofsky performance scores affect survival times (in days).
# Assume the hazards are proportional for all covariates.

data(cancer)

# Adjust "status" so that it is a 0/1
```

```
# variable (currently it is 1 = censored, 2 = observed death)
cancer$censorvar = cancer$status - 1

# Run the estimateMRH routine.  Set the maximum
# study time to 960 days, which makes each bin
# 120 days long.  This censors 0 extra subjects
# (see FindBinWidth() for an example).  Save
# the output in a folder titled 'MRH_lung'
# (default is 'MRHresults').
# Generally it is recommended to use a higher burn-in value,
# thinning value, and maximum number
# of iterations, but for illustrative purposes
# these values have been lowered.
# Note that the routine may produce a warning
# message that the algorithm has not converged,
# as typically more iterations are needed for convergence.
# However, for the purposes of this example, the number
# of iterations is sufficient.
## Not run:
fit.lung = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, outfolder = 'MRH_lung')
## End(Not run)


# See all items returned in the model fit
## Not run:
fit.lung

## End(Not run)
# See the main summary
## Not run:
fit.lung$summary

## End(Not run)
# NOTE: If estimateMRH is run as a background job,
# or if the output folder has been saved for use
# at a later instance, then  the fit can be calculated
# using the as.MRH() and summary.MRH() functions.
# See the those help pages or the vignette for
# more information.

# Run the same model as above, but with pruning.
# Save the output in a folder titled 'MRH_lung_prune'
## Not run:
fit.lung.prune = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, prune = TRUE, outfolder = 'MRH_lung_prune')
## End(Not run)
```

```
############ NON-PROPORTIONAL HAZARDS EXAMPLE ############
# Examine the tongue data set (from the KMsurv package), and
# quantify how the rumor DNA profile
# affects survival time (in weeks).
data(tongue)

# Fit the MRH model, including tumor type using
# the non-proportional hazards model.
# With 16 bins (M = 4), each bin represents 25 weeks.
# Generally it is recommended to use a higher burn-in value,
# thinning value, and maximum number
# of iterations, but for illustrative purposes
# these values have been lowered.
# Note that the routine may produce a warning
# message that the algorithm has not converged,
# as typically more iterations are needed for convergence.
# However, for the purposes of this example, the number
# of iterations is sufficient.
## Not run:
fit.tongue = estimateMRH(formula = Surv(time, delta) ~
nph(type), data = tongue, M = 4,
burnIn = 200, maxIter = 2000, thin = 1,
outfolder = 'MRH_tongue_nph')
## End(Not run)

# Get the time-varying hazard ratios
## Not run:
fit.tongue$beta

## End(Not run)
```

`FindBinWidth` *A pre-processing function that calculates the length of time per bin for different values of M.*

Description

Before fitting a MRH model, the optimal number of bins must be determined. The MRH methodology divides the total study time in to 2^M bins, so the choice of M can be determined through biological rationale or can be based on the ideal length of time per bin. In some instances, the number of bins may be relatively easy to determine. However, there are many cases where it is not clear what the ideal bin length should be. In these instances, the FindBinWidth() provides a table of lengths of time per bin for different units of time (ranging from seconds to years) for different M values (ranging from M = 2 to M = 10).

Usage

```
FindBinWidth(time, delta, time.unit, maxStudyTime)
```

Arguments

time	The vector of failure times for the subjects in the study.
delta	The censoring vector, with a '1' denoting the an observed failure, and a '0' denoting a censored failure.
time.unit	The unit of time for the failures, with options for seconds ('s'), minutes ('min'), days ('d'), weeks ('w'), months ('mon'), and years ('y').
maxStudyTime	The maximum failure time (observed or censored) or the length of the study.

Author(s)

Yolanda Hagar

Examples

```
# Examine the options for the NCCTG lung cancer data set (from the survival package)
data(cancer)

# Code the censoring variable delta as 0/1 instead of 1/2
cancer$censorvar = cancer$status - 1

# The time unit in the cancer data set is in days, so specify time.unit as "d".
FindBinWidth(cancer$time, cancer$censorvar, time.unit = 'd')

# None of the bin options show an optimal/rounded length.
# Set maxStudyTime to 960.  This will show the results if we use 8 bins, with 120 days per bin.
FindBinWidth(cancer$time, cancer$censorvar, time.unit = 'd', maxStudyTime = 8*120)

# Results show rounded bin lengths, and we see that with the shortened maximum study time
# zero extra failures are censored.
```

MRH *Converts an MRH MCMC chains text file into an MRH object*

Description

NOTE: This function converts the MRH MCMC chains file to an MRH object. For help with the MRH package, please see MRH-package for more information.

When the MRH model is run as a background routine or if the user quits the R workspace after the routine finishes running, the MRH object is not available in the workspace. The MRH() and as.MRH() functions convert the MCMC chains text file produced by the estimateMRH() routine in to an MRH object. This then allows the user to use the summary and plotting functions on the MRH object.

Usage

```
MRH(x)
as.MRH(x)
is.MRH(x)
```

Arguments

x	The MCMC chains text file produced by the MRH estimation routine.

Author(s)

Yolanda Hagar <yolanda.hagar@colorado.edu>

Examples

```
# The MRH fit example is from the estimateMRH() help page.  It does not need to
# be re-run if the code has previously been run and the outfolder ('MRH_tongue_nph') is
# saved.
data(tongue)
## Not run:
fit.tongue = estimateMRH(formula = Surv(time, delta) ~ nph(type), data = tongue, M = 4,
burnIn = 200, maxIter = 2000, thin = 1, outfolder = 'MRH_tongue_nph')
## End(Not run)

## Not run:
MRHchains = read.table('MRH_tongue_nph/MCMCchains.txt', header = TRUE)
MRHobject = MRH(MRHchains)
# When the chains are read in, the maximum study time must
# be entered in the summary function, and the total number
# of subjects (n) must be entered for BIC to be calculated.
summary(MRHobject, maxStudyTime = max(tongue$time))
DIC(MRHobject, n = nrow(tongue))
## End(Not run)
```

MRHdata *SIMULATED SURVIVAL DATA SET*

Description

This data set is a simulated survival data set, which is useful for code examples, particularly in the vignette. In addition to the failure times and a censoring variable, the data set also includes gender, treatment, and age as covariates. This treatment hazards are non-proportional, and there are periods of sparse observations and high rates of censoring, which is similar to studies with longer follow-up periods.

Usage

```
data(MRHdata)
```

Format

A data frame with 3000 observations on the following 5 variables:

`time` A numeric vector containing the survival time from start of treatment to failure.

`delta` A numeric vector of the censoring variable, which equals '1' if the failure is observed, and '0' otherwise.

`gender` A numeric binary vector, with 0 denoting males and 1 denoting females.

`treatment` A numeric vector with three treatment groups. The treatment hazard rates were simulated under the non-proportional hazards assumption.

`age` A numeric vector for age, with the measurements standardized.

`nph` *INTERNAL FUNCTION NEEDED FOR NON-PROPORTIONAL HAZARDS MODELS.*

Description

nph is used in the MRH survival model formula to denote that a variable should be modeled under the non-proportional hazards assumption.

Usage

```
nph(x)
```

Arguments

`x`	x is the covariate that will be modeled under the non-proportional hazards assumption. It must be a categorical variable, although it need not be entered using the "factor" class in R. A separate hazard rate will be estimated for each group in x.

Author(s)

Yolanda Hagar <yolanda.hagar@colorado.edu> and Vanja Dukic

References

Dukic, V., Dignam, J. (2007), Bayesian hierarchical multiresolution hazard model for the study of time-dependent failure patterns in early stage breast cancer. *Bayesian Analysis*. **2,** 591–610.

Examples

```
# Examine the tongue data set (from the KMest package), and
# quantify how the rumor DNA profile (1=Aneuploid Tumor, 2=Diploid Tumor)
# affects survival time (in weeks).
data(tongue)

# Fit the MRH model, including tumor type using the
# non-proportional hazards model.
# With 16 bins (M = 4), each bin represents 25 weeks.
# Generally it is recommended to use a higher burn-in value,
# thinning value, and maximum number
```

```
# of iterations, but for illustrative purposes these values
# have been lowered.
## Not run:
fit.tongue = estimateMRH(formula = Surv(time, delta) ~
nph(type), data = tongue, M = 4,
burnIn = 200, maxIter = 2000, thin = 1,
outfolder = 'MRH_tongue_nph')
## End(Not run)
```

`plot.MRH` *Plots the hazard rate (with credible bands) of an MRH object.*

Description

plot.mrh plots the hazard rate of an MRH object with the 95% credible interval bounds included as dashed lines.

Usage

```
## S3 method for class 'MRH'
plot(x, maxStudyTime, main = "", xlab = "Time",
ylab = "Hazard Rate", plot.type = 'h',
interval = TRUE, alpha.level = 0.05, smooth.graph = FALSE,
smooth.df = NULL, combine.graphs = TRUE, log.ratio = TRUE,...)
```

Arguments

`x`	x is an MRHobject: Either 1) an MRH fitted object or 2) MCMC chains produced by the estimateMRH routine, converted to an MRH object using MRH() or as.MRH().
`maxStudyTime`	The maximum observed or censored failure time needs to be entered when the text file of MCMC chains is used instead of the active MRH fitted object for calculation of the hazard rate.
`main`	The main title of the graph.
`xlab`	The label of the x-axis.
`ylab`	The label of the y-axis.
`plot.type`	Denotes whether the hazard rate, cumulative hazard, survival function or hazard ratio (for the case of non-proportional hazards) should be plotted, specified with 'h', 'H', 'S', or 'r' respectively. Default plot is the hazard rate.
`interval`	Set to TRUE or FALSE if the credible interval bounds should also be included in the plot. Default is TRUE.
`alpha.level`	1 minus the credible level of the interval. Default is 0.05.
`smooth.graph`	Should be set equal to TRUE if the user would like a graph of the smoothed hazard rate using smooth.spline(). Default is set to FALSE.

smooth.df	The degrees of freedom to be used in the smoothed graph of the hazard rate. If the user does not enter a value, it is set to 1/2 of the number of bins.
combine.graphs	If the MCMC chains are from a non-proportional hazards model, combine.graphs can be set to FALSE if the user would like to view the hazard rates or ratios separately. The default is TRUE, combining the hazard rates or ratios in to one graph.
log.ratio	In the non-proportional hazards models, this denotes whether the log-ratio of the hazard rates should be plotted over time (log.ratio = TRUE) or if the ratio of the hazard rates should be plotted instead (log.ratio = FALSE). The default is a plot of the log-ratio of the hazard rates.
...	Arguments to be passed to methods, such as graphical parameters (see par).

Author(s)

Yolanda Hagar <yolanda.hagar@colorado.edu> and Vanja Dukic

Examples

```
# These MRH fit portion of the examples are from the
# estimateMRH() help page.
# They do not need to be re-run if the objects
# are already in the active workspace.
data(cancer)
cancer$censorvar = cancer$status - 1
## Not run:
fit.lung = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, outfolder = 'MRH_lung')
## End(Not run)

data(tongue)
## Not run:
fit.tongue = estimateMRH(formula = Surv(time, delta) ~
nph(type), data = tongue, M = 4,
burnIn = 200, maxIter = 2000, thin = 1, outfolder = 'MRH_tongue_nph')
## End(Not run)

# Plot the hazard rate, cumulative hazard,
# and survival function of the lung model side-by-side:
## Not run:
par(mfrow = c(1,3))
plot(fit.lung, main = 'Hazard rate')
plot(fit.lung, plot.type = 'H', main = 'Cumulative hazard')
plot(fit.lung, plot.type = 'S', main = 'Survival function')
## End(Not run)

# Plot the hazard rates for the tongue (i.e. non-proportional hazards)
# model with smoothed estimates.
## Not run:
plot(fit.tongue, smooth.graph = TRUE)
```

```
## End(Not run)
# Separate the graphs for less crowding
## Not run:
plot(fit.tongue, smooth.graph = TRUE, combine.graphs = FALSE)
## End(Not run)
# Plot the hazard ratios
## Not run:
plot(fit.tongue, plot.type = 'r')
## End(Not run)

# Plot the hazard rate of the lung model using the chains.
# This requires maxStudyTime to be entered.
## Not run:
lung.chains = MRH(read.table('MRH_lung/MCMCchains.txt', header = TRUE))
plot(lung.chains, maxStudyTime = 960)
## End(Not run)
```

`summary.MRH` *Summarizes the results of an MRH object*

Description

This function summarizes the MRH object (either the fitted object or the MCMC chains) and returns estimates of the covariate effects, the hazard rate, the survival curve, the cumulative hazard, and the log ratio for non-propoortional hazards models. If desired, the user can specify the alpha-level for the credible intervals.

Usage

```
## S3 method for class 'MRH'
summary(object, alpha.level, maxStudyTime, ...)
```

Arguments

`object`	object is an MRHobject: Either 1) an MRH fitted object or 2) MCMC chains produced by the estimateMRH routine, converted to an MRH object using MRH() or as.MRH().
`alpha.level`	The width of the credible intervals for the parameters, with the lower bound calculated as the alpha/2 percentile, and the upper bound calculated as the 1-alpha/2 percentile.
`maxStudyTime`	The maximum study period (or the censoring time) used in the analysis. This is only required if the MCMC chains are used instead of the fitted MRH model.
`...`	Arguments to be passed to methods, such as graphical parameters (see par).

Value

d	The hazard increments within each bin (i.e. the cumulative hazard within each bin)
hazardRate	The hazard rate across the bins
SurvivalCurve	The survival curve (i.e. S(t)) over the course of the study.
CumulativeHazard	The cumulative hazard (i.e. H(t)) over the course of the study.
beta	The estimated covariate effects. If the model contains both PH and NPH parameters, the covariate effects will be denoted with 'betaPH' and 'betaNPH'.
H	The parameter H, which is the cumulative hazard at the end of the study.
Rmp	The split parameters.
maxStudyTime	The maximum study time (i.e. the censoring time).

Author(s)

Yolanda Hagar <yolanda.hagar@colorado.edu> and Vanja Dukic

Examples

```
# These MRH fit portion of the examples are from the
# estimateMRH() help page.
# They do not need to be re-run if the objects
# are already in the active workspace.
data(cancer)
cancer$censorvar = cancer$status - 1
## Not run:
fit.lung = estimateMRH(formula = Surv(time, censorvar) ~
age + as.factor(sex) + ph.karno, data = cancer,
M = 3, maxStudyTime = 960, burnIn = 200, maxIter = 1000,
thin = 1, outfolder = 'MRH_lung')
## End(Not run)

data(tongue)
## Not run:
fit.tongue = estimateMRH(formula = Surv(time, delta) ~
nph(type), data = tongue, M = 4,
burnIn = 200, maxIter = 2000, thin = 1, outfolder = 'MRH_tongue_nph')
## End(Not run)

# Summarize the models
## Not run:
summary(fit.lung)
summary(fit.lung, alpha.level = .01)

# Read in from the saved output file, converting to an MRH object,
# then summarize.
mcmc.lung = as.MRH(read.table('MRH_lung/MCMCchains.txt', header = TRUE))
summary(mcmc.lung, maxStudyTime = 960)
summary(mcmc.lung, maxStudyTime = 960, alpha.level = .01)
```

```
## End(Not run)
```

Index