Package: PhaseType (via r-universe)

October 28, 2026

Title Inference for Phase-Type Distributions

Description Functions to perform Bayesian inference on absorption time data for Phase-type distributions. The methods of Bladt et al (2003) <doi:10.1080/03461230110106435> and Aslett (2012) <https://www.louisaslett.com/PhD_Thesis.pdf> are provided.

URL https://www.louisaslett.com/PhaseType/, https://github.com/louisaslett/PhaseType

BugReports https://github.com/louisaslett/PhaseType/issues

Version 0.3.0

Maintainer Louis Aslett <louis.aslett@durham.ac.uk>

Copyright The code in src/arms.c is slightly modified based on the research paper implementation written by Wally Gilks. All other code in all other files were written by the maintainer.

Imports coda, ggplot2, reshape

Suggests actuar

License GPL-2 | GPL-3

LazyLoad yes

NeedsCompilation yes

Author Louis Aslett [aut, cre], Wally Gilks [ctb]

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

Date/Publication 2024-09-15 03:00:53 UTC

RemoteUrl https://github.com/cran/PhaseType

RemoteRef HEAD

RemoteSha d7e240121e3a08c56ea84fd12a3c13997d51fb16

Contents

PhaseType-package 2
phtMCMC 3
phtMCMC2 6

Index **9**

`PhaseType-package` *Phase-type Distributions Toolbox*

Description

A collection of tools for working with Phase-type Distributions, including sampling methods and both frequentist and Bayesian inference.

Details

Package:	PhaseType
Type:	Package
Version:	0.2.1
Date:	2011-10-12
License:	GPL-2 | GPL-3
LazyLoad:	yes

Author(s)

Louis J. M. Aslett, <louis.aslett@durham.ac.uk> (https://www.louisaslett.com)

References

Aslett, L. J. M. (2012), MCMC for Inference on Phase-type and Masked System Lifetime Models. Ph.D. thesis, Trinity College Dublin..

Bladt, M., Gonzalez, A. & Lauritzen, S. L. (2003), 'The estimation of phase-type related functionals using Markov chain Monte Carlo methods', *Scandinavian Journal of Statistics* **2003**(4), 280-300.

Examples

```
library(actuar)
# Define the S matrix (columnwise)
S <- matrix(c(-3.6, 9.5, 9.5, 1.8, -11.3, 0, 1.8, 0, -11.3), 3)

# Define starting state distribution
pi <- c(1, 0, 0)

# Generate 50 random absorption times from the Phase-type with subgenerator S
# and starting distribution pi, which we will try to infer next
x <- rphtype(50, pi, S)


library(PhaseType)
```

```
# FIRST: descriptive model fit (Bladt et al. 2003)
# Prior on starting state
dirpi <- c(1, 0, 0)

# Gamma prior: shape hyperparameters (one per matrix element, columnwise)
nu <- c(24, 24, 1, 180, 1, 24, 180, 1, 24)

# Gamma prior: reciprocal scale hyperparameters (one per matrix row)
zeta <- c(16, 16, 16)

# Define dimension of model to fit
n <- 3

# Perform 20 MCMC iterations (fix inner Metropolis-Hastings to one iteration
# since starts in stationarity here).  Do more in practise!!
res1 <- phtMCMC(x, n, dirpi, nu, zeta, 20, mhit=1)
print(res1)

plot(res1)



# SECOND: mechanistic model fit (Aslett and Wilson 2011)
# Prior on starting state
dirpi <- c(1, 0, 0)

# Define the structure of the Phase-type generator
TT <- matrix(c(0,"R","R",0,"F",0,0,0,"F",0,0,0,0,"F","F",0), 4)

# Gamma prior: shape hyperparameters (one per model parameter)
nu <- list("R"=180, "F"=24)

# Gamma prior: reciprocal scale hyperparameters (one per model parameter)
zeta <- c("R"=16,"F"=16)

# Perform 20 MCMC iterations.  Do more in practise!!
res2 <- phtMCMC2(x, TT, dirpi, nu, zeta, 20)
print(res2)

plot(res2)
```

---

phtMCMC *MCMC for dense Phase-type models*

Description

Markov-chain Monte Carlo (MCMC) sampler for Bayesian inference on Phase-type models where data consist solely of time to entering the absorbing state (Bladt et al., 2003).

Consider an m+1 state continuous-time Markov chain (CTMC) where the final (m+1 st) state is absorbing. Then, given data consisting of first passage times to the last state, this function performs Bayesian inference on the rate parameters of the latent continuous-time Markov chain where the generator is assumed to be dense and all rates independent.

Usage

```
phtMCMC(x, states, beta, nu, zeta, n, mhit=1, resume=NULL, silent=FALSE)
```

Arguments

`x`	a vector of absorption times (or times at which censoring occurs).
`states`	an integer describing how many states (ie what dimension) the fitted Phase-type's generator should be.
`beta`	a vector of length m representing the Dirichlet prior on the starting state of the latent continuous-time Markov chain. Entries should sum to 1.
`nu`	a list of the Gamma shape hyper-parameters for the prior of each parameter in the continuous-time Markov chain. These should match the dense generator matrix filled **column-wise**.
`zeta`	a list of the Gamma reciprocal scale hyper-parameters for the prior of each parameter in the continuous-time Markov chain. These should match the dense generator matrix, with one zeta value **per row**.
`n`	the total number of MCMC iterations to compute.
`mhit`	the number of Metropolis-Hastings iterations to perform when sampling the latent process.
`resume`	`NULL` indicates a new chain is to be sampled. Otherwise, passing in an object of class `phtMCMC` (as returned by a call to this function) picks up a previously run MCMC chain at its end an continues for another `n` iterations.
`silent`	setting to `TRUE` suppresses the feedback about current iteration. Highly recommended for batch files or else output fills with unnecessary iteration updates.

Details

Usage of this function effectively involves specification of the number of states for the CTMC generator matrix, the parameter priors in Gamma form and providing absorption time data.

If you want to specify structure on your generator (for example, to model a stochastic process about which some underlying mechanism is known), then consider using `phtMCMC2` instead. This function is best illustrated by the example below which is more fully discussed on this webpage.

Value

`phtMCMC` returns an object of class `"phtMCMC"`.

An object of class `"phtMCMC"` is a list containing at least the following components:

`samples`	an object of class `"mcmc"` containing the MCMC samples for each parameter in the model.

data	a vector containing the original data used in the inference.
vars	a list of the distinct variable names in the CTMC generator.
TT	the naming scheme for the generator which was fitted.
beta	the Dirichlet prior probability mass function on the starting states.
nu	a list of the prior nu hyper-parameters used when phtMCMC2 was called.
zeta	a list of the prior zeta hyper-parameters used when phtMCMC2 was called.
iterations	the number of MCMC iterations contained in the object.
MHit	the number of iterations used if method="MHRS"

Note

Please feel free to email <louis.aslett@durham.ac.uk> with any queries or if you encounter errors when running this function.

Author(s)

Louis J.M. Aslett <louis.aslett@durham.ac.uk> (https://www.louisaslett.com/)

References

Bladt, M., Gonzalez, A. and Lauritzen, S. L. (2003), 'The estimation of Phase-type related functionals using Markov chain Monte Carlo methods', *Scandinavian Actuarial Journal* 2003(4), 280-300.

See Also

phtMCMC2

Examples

```
# Some pre-simulated absorption times
x <- c(1.45353415045187, 1.85349532001349, 2.01084961814576, 0.505725921290172,
1.56252630012213, 3.41158665930278, 1.52674487509487, 4.3428662377235,
8.03208018151311, 2.41746547476986, 0.38828086509283, 2.61513815012196,
3.39148865480856, 1.82705817807965, 1.42090953713845, 0.851438991331866,
0.0178808867191894, 0.632198596390046, 0.959910259815998, 1.83344199966323)

# Prior on starting state
dirpi <- c(1, 0, 0)
# Gamma prior: shape hyperparameters (one per matrix element, columnwise)
nu <- c(24, 24, 1, 180, 1, 24, 180, 1, 24)
# Gamma prior: reciprocal scale hyperparameters (one per matrix row)
zeta <- c(16, 16, 16)
# Define dimension of model to fit
n <- 3
# Perform 20 MCMC iterations (fix inner Metropolis-Hastings to one iteration
# since starts in stationarity here).  Do more in practice!
res <- phtMCMC(x, n, dirpi, nu, zeta, 6, mhit=1)
print(res)
```

```
plot(res)
```

---

phtMCMC2 *MCMC for structured Phase-type models*

Description

Markov-chain Monte Carlo (MCMC) sampler for Bayesian inference on Phase-type models where data consist solely of time to entering the absorbing state (Aslett and Wilson, 2011, and Aslett, 2012).

Consider an m+1 state continuous-time Markov chain (CTMC) where the final (m+1 st) state is absorbing. Then, given data consisting of first passage times to the last state, this function performs Bayesian inference on the rate parameters of the latent continuous-time Markov chain where the generator has some fixed structure.

Usage

```
phtMCMC2(x, TT, beta, nu, zeta, n, censored=rep(FALSE, length(x)),
         C=matrix(1.0, nrow=dim(TT)[1], ncol=dim(TT)[2]), method="ECS", mhit=1,
         resume=NULL, silent=FALSE)
```

Arguments

x	a vector of absorption times (or times at which censoring occurs).
TT	a matrix describing the structure of the Phase-type distribution to be used in the inference. Each element of the matrix should either be a string for the variable name, or zero to indicate the transition there is prohibited. See Details and Examples sections below.
beta	a vector of length m representing the Dirichlet prior on the starting state of the latent continuous-time Markov chain. Entries should sum to 1.
nu	a list of the Gamma shape hyper-parameters for the prior of each parameter in the continuous-time Markov chain. The tag = value arguments to the list creation should be such that tag matches the parameter name given in TT and value is the numeric value of the hyper-parameter.
zeta	a list of the Gamma reciprocal scale hyper-parameters for the prior of each parameter in the continuous-time Markov chain. The tag = value arguments to the list creation should be such that tag matches the parameter name given in TT and value is the numeric value of the hyper-parameter.
n	the total number of MCMC iterations to compute.
censored	a vector of TRUE/FALSEs of the same length as x indicating whether the corresponding observation is censored.
C	a numeric matrix of the same dimensions as TT which specifies any constant scalar multiples of the parameters in TT. Entries must be non-negative.

method	the sampling method to use for the latent stochastic process: method="ECS" is Exact Conditional Sampling as described in Aslett and Wilson (2011); method="MHRS" is Metropolis-Hastings + Rejection Sampling as in the original work of Bladt et al. (2003).
mhit	the number of Metropolis-Hastings iterations to perform when sampling the latent process. Ignored unless method="MHRS".
resume	NULL indicates a new chain is to be sampled. Otherwise, passing in an object of class phtMCMC (as returned by a call to this function) picks up a previously run MCMC chain at its end an continues for another n iterations.
silent	setting to TRUE suppresses the feedback about current iteration. Highly recommended for batch files or else output fills with unnecessary iteration updates.

Details

Usage of this function effectively involves specification of the structure of an absorbing CTMC generator matrix, the parameter priors in Gamma form and providing absorption time data.

The generator matrix TT is specified as a matrix containing strings (ie text) naming the parameters in each element of the matrix. This allows constraints to be imposed (see Aslett & Wilson, 2011) and is best illustrated by the example below which is more fully discussed on this webpage.

Value

phtMCMC2 returns an object of class "phtMCMC".

An object of class "phtMCMC" is a list containing at least the following components:

samples	an object of class "mcmc" containing the MCMC samples for each parameter in the model.
data	a vector containing the original data used in the inference.
vars	a list of the distinct variable names detected in the CTMC generator.
TT	the original CTMC generator passed in when phtMCMC2 was called.
beta	the Dirichlet prior probability mass function on the starting states.
nu	a list of the prior nu hyper-parameters used when phtMCMC2 was called.
zeta	a list of the prior zeta hyper-parameters used when phtMCMC2 was called.
iterations	the number of MCMC iterations contained in the object.
censored	the original vector of TRUE/FALSEs of the same length as x indicating whether the corresponding observations are censored.
method	a string indicating the method used for sampling of the latent process
MHit	the number of iterations used if method="MHRS"

Note

Please feel free to email <louis.aslett@durham.ac.uk> with any queries or if you encounter errors when running this function.

Author(s)

Louis J.M. Aslett <louis.aslett@durham.ac.uk> (https://www.louisaslett.com/)

References

Aslett, L. J. M. and Wilson, S. P. (2011), 'Markov chain Monte Carlo for inference on Phase-type models', ISI 2011 Proceedings, (https://www.louisaslett.com/Proceedings/ISI_2011.html).

Aslett, L. J. M. (2012), 'MCMC for Inference on Phase-type and Masked System Lifetime Models', PhD Thesis, Trinity College Dublin (https://www.louisaslett.com/PhD_Thesis.html).

Bladt, M., Gonzalez, A. and Lauritzen, S. L. (2003), 'The estimation of Phase-type related functionals using Markov chain Monte Carlo methods', *Scandinavian Actuarial Journal* 2003(4), 280-300.

See Also

phtMCMC

Examples

```
# Some pre-simulated absorption times
x <- c(1.45353415045187, 1.85349532001349, 2.01084961814576, 0.505725921290172,
1.56252630012213, 3.41158665930278, 1.52674487509487, 4.3428662377235,
8.03208018151311, 2.41746547476986, 0.38828086509283, 2.61513815012196,
3.39148865480856, 1.82705817807965, 1.42090953713845, 0.851438991331866,
0.0178808867191894, 0.632198596390046, 0.959910259815998, 1.83344199966323)

# Prior on starting state
dirpi <- c(1, 0, 0)
# Define the structure of the Phase-type generator
TT <- matrix(c(0,"R","R",0,"F",0,0,0,"F",0,0,0,0,"F","F",0), 4)
# Gamma prior: shape hyperparameters (one per model parameter)
nu <- list("R"=180, "F"=24)
# Gamma prior: reciprocal scale hyperparameters (one per model parameter)
zeta <- c("R"=16,"F"=16)
# Perform 20 MCMC iterations.  Do more in practice!!
res <- phtMCMC2(x, TT, dirpi, nu, zeta, 20)
print(res)

plot(res)
```

Index

∗ **bayesian**
phtMCMC, 3
phtMCMC2, 6
∗ **inference**
phtMCMC, 3
phtMCMC2, 6
∗ **mcmc**
PhaseType-package, 2
phtMCMC, 3
phtMCMC2, 6
∗ **phase-type**
PhaseType-package, 2
phtMCMC, 3
phtMCMC2, 6
∗ **pht**
PhaseType-package, 2

PhaseType *(*PhaseType-package*)*, 2
PhaseType-package, 2
phtMCMC, 3, *8*
phtMCMC2, *4, 5*, 6