Package ‘phmm’

February 20, 2015

Version 0.7-5
Date 2013-07-29
Title Proportional Hazards Mixed-effects Model (PHMM)
Author Michael Donohue and Ronghui Xu
Maintainer Michael Donohue <mdonohue@ucsd.edu>
Depends survival, lattice, Matrix
Suggests lme4 (>= 1.0)
Description Fits proportional hazards model incorporating random effects using an EM algorithm using Markov Chain Monte Carlo at E-step.
License GPL-3
Collate 'formula.R' 'linear.predictors.R' 'phmm.R' 'pseudoPoisPHMM.R'
'traceHat.R'
NeedsCompilation yes
Repository CRAN
Date/Publication 2013-09-21 10:00:49

R topics documented:

phmm-package ................................................. 2
AIC.phmm ....................................................... 3
cAIC .............................................................. 5
linear.predictors ............................................... 7
loglik.cond ..................................................... 8
phmm ............................................................ 8
phmm.cond.loglik ............................................. 11
plot.phmm ..................................................... 11
pseudoPoisPHMM ............................................. 12
traceHat ......................................................... 14

Index 16
Description

Fits proportional hazards model incorporating random effects. The function implements an EM algorithm using Markov Chain Monte Carlo at the E-step as described in Vaida and Xu (2000).

Details

Package: phmm
Version: 0.2
Date: 2008-01-15
Depends: survival
Suggests: lme4
License: GPL2
Packaged: Fri Jul 11 10:33:57 2008; mdonohue
Built: R 2.8.0; universal-apple-darwin8.11.1; 2008-11-29 12:05:00; unix

Index:

- AIC.phmm: Akaike Information Criterion for PHMM
- cAIC: Conditional Akaike Information Criterion for PHMM
- e1582: Eastern Cooperative Oncology Group (EST 1582)
- linear.predictors: PHMM Design
- loglik.cond: PHMM conditional log-likelihood
- phmm: Proportional Hazards Model with Mixed Effects
- phmm-package: Proportional Hazards Model with Mixed Effects
- phmm.cond.loglik: PHMM conditional log-likelihood
- phmm.design: PHMM Design
- pseudoPoisPHMM: Pseudo poisson data for fitting PHMM via GLMM
- traceHat: Trace of the "hat" matrix from PHMM-MCEM fit

Author(s)

Ronghui Xu, Michael Donohue

Maintainer: Michael Donohue <mdonohue@ucsd.edu>

References

AIC.phmm


**Examples**

```r
n <- 50  # total sample size
clust <- 5  # number of clusters
clusters <- rep(1:nclust,each=n/clust)
beta0 <- c(1,2)
set.seed(13)
#generate phmm data set
Z <- cbind(Z1=sample(0:1,n,replace=TRUE),
          Z2=sample(0:1,n,replace=TRUE),
          Z3=sample(0:1,n,replace=TRUE))
b <- cbind(rep(rnorm(nclust),each=n/clust),rep(rnorm(nclust),each=n/clust))
WB <- matrix(0,n,2)
for( j in 1:2 ) WB[,j] <- Z[,j]*b[,j]
WB <- apply(WB,1,sum)
T <- -log(runif(n,0,1))*exp(-Z[,c('Z1','Z2')]*beta-WB)
C <- runif(n,0,1)
time <- ifelse(T<C,T,C)
event <- ifelse(T==C,1,0)
mean(event)
phmmd <- data.frame(Z)
phmmd$cluster <- clusters
phmmd$time <- time
phmmd$event <- event

fit.phmm <- phmm(Surv(time, event) ~ Z1 + Z2 + (-1 + Z1 + Z2 | cluster),
                 phmmd, Gbs = 100, Gbsvar = 1000, VARSTART = 1,
                 NINIT = 10, MAXSTEP = 100, CONVERG=90)
summary(fit.phmm)
```

---

**AIC.phmm**

*Akaike Information Criterion for PHMM*

**Description**

Function calculating the Akaike information criterion for PHMM fitted model objects, according to the formula 

$-2 \cdot \log \text{likelihood} + k \cdot \rho$, where $npar$ represents the number of parameters in the fitted model. The function returns a list of AIC calculations corresponding different likelihood estimations: conditional and marginal likelihoods calculated by Laplace approximation, reciprocal importance sampling, and bridge sampling (only implemented for nreff < 3). The default $k = 2$, is for the usual AIC.

**Usage**

```r
# S3 method for class 'phmm'
AIC(object, ..., k = 2)
```
Arguments

object  a fitted PHMM model object of class phmm,
...  optionally more fitted model objects.
k  numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.

Value

Returns a list of AIC values corresponding to all available log-likelihood values from the fit. See phmm for details of the log-likelihood values.

References


See Also

phmm, AIC

Examples

n <- 50  # total sample size
nclust <- 5  # number of clusters
clusters <- rep(1:nclust,each=n/nclust)
beta0 <- c(1,2)
set.seed(13)
# generate phmm data set
Z <- cbind(Z1=sample(0:1,n,replace=TRUE),
            Z2=sample(0:1,n,replace=TRUE),
            Z3=sample(0:1,n,replace=TRUE))
b <- cbind(rep(rnorm(nclust),each=n/nclust),rep(rnorm(nclust),each=n/nclust))
wb <- matrix(0,n,2)
for(j in 1:2) wb[,j] <- Z[,j]*b[,j]
wb <- apply(wb,1,sum)
T <- -log(runif(n,0,1))*exp(-(Z[,c('Z1','Z2')]%*%beta0-wb)
C <- runif(n,0,1)
time <- ifelse(T<C,T,C)
event <- ifelse(T<=C,1,0)
mean(event)
phmmd <- data.frame(Z)
phmmd$cluster <- clusters
phmmd$time <- time
phmmd$event <- event

fit.phmm <- phmm(Surv(time, event) ~ Z1 + Z2 + (-1 | cluster),
                 phmmd, Gbs = 100, Gbsvar = 1000, VARSTART = 1,
                 NINIT = 10, MAXSTEP = 100, CONVERG=90)

# Same data can be fit with lmer,
# though the correlation structures are different.
cAIC

# Conditional Akaike Information Criterion for PHMM

Description

Function calculating the conditional Akaike information criterion (Vaida and Blanchard 2005) for PHMM fitted model objects, according to the formula $-2 \times \text{loglikelihood} + k \times \rho$, where $\rho$ represents the "effective degrees of freedom" in the sense of Hodges and Sargent (2001). The function uses the log-likelihood conditional on the estimated random effects; and trace of the "hat matrix", using the generalized linear mixed model formulation of PHMM, to estimate $\rho$. The default $k = 2$, conforms with the usual AIC.

Usage

```r
## S3 method for class 'phmm'
cAIC(object, method = "direct", ..., k = 2)
## S3 method for class 'coxph'
cAIC(object, ..., k = 2)
```

Arguments

- `object` A fitted PHMM model object of class `phmm`.
- `method` Passed to `traceHat`. Options include "direct", "pseudoPois", or "HaLee". The methods "direct" and "HaLee" are algebraically equivalent.
- `...` Optionally more fitted model objects.
- `k` numeric, the penalty per parameter to be used; the default $k = 2$ conforms with the classical AIC.
Value

Returns a numeric value of the cAIC corresponding to the PHMM fit.

References


See Also

phmm, AIC

Examples

```R
# Not run:
n <- 50 # total sample size
nclust <- 5 # number of clusters
clusters <- rep(1:nclust, each=n/nclust)
beta0 <- c(1,2)
set.seed(13)
# generate phmm data set
Z <- cbind(Z1=sample(0:1,n,replace=TRUE),
          Z2=sample(0:1,n,replace=TRUE),
          Z3=sample(0:1,n,replace=TRUE))
b <- cbind(rep(rnorm(nclust),each=n/nclust),rep(rnorm(nclust),each=n/nclust))
wb <- matrix(0,n,2)
for(j in 1:2) wb[,j] <- Z[,j]*b[,j]
wb <- apply(wb,1,sum)
T <- -log(runif(n,0,1))*exp(-Z[c('Z1','Z2')]%*%beta0-wb)
C <- runif(n,0,1)
time <- ifelse(T<C,T,C)
event <- ifelse(T<=C,1,0)
mean(event)
phmm <- data.frame(Z)
phmm$cluster <- clusters
phmm$time <- time
phmm$event <- event
fit.phmm <- phmm(Surv(time, event) ~ Z1 + Z2 + (-1 + Z1 + Z2 | cluster),
                 phmm, Gbs = 100, Gbsvar = 1000, VARSTART = 1,
                 NINIT = 10, MAXSTEP = 100, CONVERG=90)
```
# Same data can be fit with lmer, # though the correlation structures are different.
poisphmm <- pseudoPoisPHMM(fit.phmm)

library(lme4)
fit.lmer <- lmer(m~l+as.factor(time)+z1+z2+
    (-1+w1+w2|cluster)+offset(log(N)),
    as.data.frame(as(poisphmm, "matrix")), family=poisson)

fixef(fit.lmer)[c("z1","z2")]
fit.phmm$coef
VarCorr(fit.lmer)$cluster
fit.phmm$Sigma

logLik(fit.lmer)
fit.phmm$loglik

traceHat(fit.phmm)

summary(fit.lmer)
## End(Not run)

---

**linear.predictors**  
*PHMM Design*

**Description**

Internal function for extracting the linear predictors of the PHMM model from an object of class `phmm` returned by `phmm`.

**Usage**

`linear.predictors(x)`

**Arguments**

- **x** an object of class `phmm`.

**Value**

A vector of estimates from call to `phmm` of $\beta'x_{ij} + w_{ij}'h_i$.

**See Also**

`phmm`
loglik.cond  

*PHMM conditional log-likelihood*

**Description**

Function for computing log-likelihood conditional on the estimated random effects from an object of class `phmm` returned by `phmm`.

**Usage**

`loglik.cond(x)`

**Arguments**

- `x`: an object of class `phmm`.

**Value**

The PHMM log-likelihood conditional on the estimated random effects.

**See Also**

`phmm, phmm.cond.loglik`

---

**phmm**  

*Proportional Hazards Model with Mixed Effects*

**Description**

Fits a proportional hazards regression model incorporating random effects. The function implements an EM algorithm using Markov Chain Monte Carlo (MCMC) at the E-step as described in Vaida and Xu (2000).

**Usage**

`phmm(formula, data, subset, na.action = na.fail, Sigma = "identity", varcov = "diagonal", NINIT = 10, VARSTART = 1, MAXSTEP = 100, CONVERG = 90, Gbs = 100, Gbsvar = 1000, verbose = FALSE, maxtime = 120, random)`
phmm

Arguments

Formula

model formula for the fixed and random components of the model (as in \code{lmer}). An intercept is implicitly included in the model by estimation of the error distribution. As a consequence -1 in the model formula does not have any effect. The left-hand side of the \code{formula} must be a \code{Surv} object.

data

optional data frame in which to interpret the variables occurring in the formulas.

subset

subset of the observations to be used in the fit.

na.action

function to be used to handle any \code{NAs} in the data. The user is discouraged to change a default value \code{na.fail}.

Sigma

initial covariance matrix for the random effects. Defaults to "identity".

varcov

constraint on \code{Sigma}. Currently only "diagonal" is supported.

NINIT

number of starting values supplied to Adaptive Rejection Metropolis Sampling (ARMS) algorithm.

VARSTART

starting value of the variances of the random effects.

MAXSTEP

number of EM iterations.

CONVERG

iteration after which Gibbs sampling size changes from \code{Gbs} to \code{Gbsvar}.

Gbs

initial Gibbs sampling size (until CONVERG iterations).

Gbsvar

Gibbs sampling size after CONVERG iterations.

verbose

Set to \code{TRUE} to print EM steps.

maxtime

maximum time in seconds, before aborting EM iterations. Defaults to 120 seconds.

random

The argument \code{random} is no longer used. Random components are are expressed in \code{formula}.

Details

The proportional hazards model with mixed effects is equipped to handle clustered survival data. The model generalizes the usual frailty model by allowing log-linear multivariate random effects. The software can only handle random effects from a multivariate normal distribution. Maximum likelihood estimates of the regression parameters and variance components is gotten by EM algorithm, with Markov chain Monte Carlo (MCMC) used in the E-step.

Care must be taken to ensure the MCMC-EM algorithm has converged, as the algorithm stops after MAXSTEP iterations. No convergence criteria is implemented. It is advised to plot the estimates at each iteration using the \code{plot.phmm} method. For more on MCMC-EM convergence see Booth and Hobart (1999).

Value

The function produces an object of class "phmm" consisting of:
References


See Also

survfit, Surv.

Examples

n <- 50 # total sample size
nclust <- 5 # number of clusters
clusters <- rep(1:nclust,each=n/nclust)
beta0 <- c(1,2)
set.seed(13)
#generate phmm data set
Z <- cbind(Z1=sample(0:1,n,replace=TRUE),
           Z2=sample(0:1,n,replace=TRUE),
           Z3=sample(0:1,n,replace=TRUE))
b <- cbind(rep(rnorm(nclust),each=n/nclust),rep(rnorm(nclust),each=n/nclust))
wb <- matrix(0,n,2)
for( j in 1:2 ) wb[,j] <- Z[,j]*b[,j]
wb <- apply(wb,1,sum)
T <- -log(runif(n,0,1))*exp(-Z[,c('Z1','Z2')]*%*%beta0-wb)
C <- runif(n,0,1)
time <- ifelse(T<C,T,C)
event <- ifelse(T<=C,1,0)
mean(event)
phmmd <- data.frame(Z)
phmmd$cluster <- clusters
phmmd$time <- time
phmmd$event <- event

fit.phmm <- phmm(Surv(time, event) ~ Z1 + Z2 + (-1 + Z1 + Z2 | cluster),
                 phmmd, Gbs = 100, Gbsvar = 1000, VARSTART = 1,
                 NINIT = 10, MAXSTEP = 100, CONVERG=90)
summary(fit.phmm)
Function for computing log-likelihood conditional on the estimated random effects from the data and specified parameter estimates of a PHMM.

Usage

```
phmm.cond.loglik(time, delta, z, beta, w, b)
```

**Arguments**

- `time`: Follow-up time (right censored data).
- `delta`: The status indicator (0=alive, 1=dead; or TRUE=dead, FALSE=alive).
- `z`: Numeric matrix (Nxnfixed) of covariates for fixed effects.
- `beta`: Fitted fixed effects coefficients (p-vector).
- `w`: Numeric matrix (Nxnrandom) of covariates for random effects.
- `b`: Numeric matrix (Nxnrandom) of random effects estimates.

**Value**

The PHMM log-likelihood conditional on the estimated random effects.

**See Also**

- `phmm`, `loglik.cond`

---

Plots the value of each parameter of the model at each iteration of the MCMC-EM algorithm. For more on MCMC-EM convergence see Booth & Hobart (1999).

**Usage**

```
# S3 method for class 'phmm'
plot(x, ...)
```
pseudoPoisPHMM

Arguments

x phmm object return by phmm

... other arguments passed to xyplot

References


See Also

phmm.

```
summary(pseudoPoisPHMM)
```

pseudopoisphmm

Pseudo poisson data for fitting PHMM via GLMM

Description

Function for generating a pseudo Poisson data set which can be used to fit a PHMM using GLMM software. This follows the mixed-model extension Whitehead (1980), who described how to fit Cox (fixed effects) models with GLM software.

Usage

pseudoPoisPHMM(x)

Arguments

x an object of class phmm.

Value

A data.frame with columns:

References


See Also

phmm, traceHat
Examples

```r
## Not run:

n <- 50  # total sample size
nclust <- 5  # number of clusters
clusters <- rep(1:nclust, each=n/nclust)
beta0 <- c(1, 2)
set.seed(13)

# generate phmm data set
Z <- cbind(Z1=sample(0:1, n, replace=TRUE),
          Z2=sample(0:1, n, replace=TRUE),
          Z3=sample(0:1, n, replace=TRUE))
b <- cbind(rep(rnorm(nclust), each=n/nclust), rep(rnorm(nclust), each=n/nclust))
wb <- matrix(0, n, 2)
for (j in 1:2) wb[, j] <- Z[, j]*b[, j]
wb <- apply(wb, 1, sum)
T <- -log(runif(n, 0, 1))*exp(-Z[, c('Z1', 'Z2')]*beta0 - wb)
C <- runif(n, 0, 1)
time <- ifelse(T<C, T, C)
event <- ifelse(T<=C, 1, 0)
mean(event)

phmm <- data.frame(Z)
phmm$cluster <- clusters
phmm$time <- time
phmm$event <- event

fit.phmm <- phmmSurr(time, event) ~ Z1 + Z2 + (-1 + Z1 + Z2 | cluster),
             pgmm, Gs = 100, Gsvar = 1000, VARSTART = 1,
             NINIT = 10, MAXSTEP = 100, CONVERG=90)

# Same data can be fit with lmer,
# though the correlation structures are different.
poisphmm <- pseudoPoisPHMM(fit.phmm)

library(lme4)
fit.lmer <- lmer(m~l+as.factor(time)+z1+z2+
                 (-1+w1+w2|cluster)+offset(log(N)),
                 as.data.frame(as(poisphmm, "matrix")), family=poisson)

fixef(fit.lmer)[c("z1","z2")]
fit.phmm$coef
VarCorr(fit.lmer)$cluster
fit.phmm$Sigma

logLik(fit.lmer)
fit.phmm$loglik

traceHat(fit.phmm)

## End(Not run)
```
traceHat

Trace of the "hat" matrix from PHMM-MCEM fit

Description
Compute trace of the "hat" matrix from PHMM-MCEM fit using a direct approximation method
(Donohue, et al, submitted), an approximation via hierarchical likelihoods (Ha et al, 2007), or an

Usage
traceHat(x, method = "direct")

Arguments
x an object of class phmm,
method acceptable values are "direct", "pseudoPois", or "HaLee",

Value
The trace of the "hat" matrix which can be used as a measure of complexity of the model.

References
Donohue, M, Xu, R, Vaida, F, Haut R. Model Selection for Clustered Data: Conditional Akaike
Information under GLMM and PHMM. Submitted.
in Medicine, Vol. 26, pp. 4790-4807.
Whitehead, J. (1980). Fitting Cox's Regression Model to Survival Data using GLIM. Journal of
the Royal Statistical Society. Series C, Applied statistics, 29(3), 268-.

See Also
phmm, AIC.phmm

Examples
## Not run:
n <- 50  # total sample size
nclust <- 5  # number of clusters
clusters <- rep(1:nclust, each=n/nclust)
betatheta <- c(1,2)
set.seed(13)
#generate phmm data set
Z <- cbind(Z1=sample(0:1,n,replace=TRUE),
Z2 = sample(0:1, n, replace=TRUE),
Z3 = sample(0:1, n, replace=TRUE))
b <- cbind(rep(rnorm(nclust), each=n/nclust), rep(rnorm(nclust), each=n/nclust))
wb <- matrix(0, n, 2)
for (j in 1:2) wb[, j] <- Z[, j]*b[, j]
wb <- apply(wb, 1, sum)
T <- -log(runif(n, 0, 1))*exp(-Z[, c('Z1', 'Z2')]*beta0 - wb)
C <- runif(n, 0, 1)
time <- ifelse(T < C, T, C)
event <- ifelse(T <= C, 1, 0)
mean(event)

phmm <- data.frame(Z)
phmm$cluster <- clusters
phmm$time <- time
phmm$event <- event

fit.phmm <- phmm(Surv(time, event) ~ Z1 + Z2 + (-1 + Z1 + Z2 | cluster),
phmm, Gbs = 100, Gbsvar = 1000, VARSTART = 1, NINIT = 10, MAXSTEP = 100, CONVERG=90)

# Same data can be fit with lmer,
# though the correlation structures are different.
poisphmm <- pseudoPoisPHMM(fit.phmm)

library(lme4)
fit.lmer <- lmer(m~-1+as.factor(time)+z1+z2+(-1+w1+w2|cluster)+offset(log(N)),
as.data.frame(as(poisphmm, "matrix")), family=poisson)

fixef(fit.lmer)[c("z1","z2")]
fit.phmm$coeff
VarCorr(fit.lmer)$cluster
fit.phmm$Sigma

logLik(fit.lmer)
fit.phmm$loglik

traceHat(fit.phmm)

## End(Not run)
Index

*Topic package
  phmm-package, 2
*Topic survival
  AIC.phmm, 3
cAIC, 5
  linear.predictors, 7
  loglik.cond, 8
  phmm, 8
  phmm.cond.loglik, 11
  plot.phmm, 11
  pseudoPoisPHMM, 12
  traceHat, 14

AIC, 4, 6
AIC.phmm, 3, 14

cAIC, 5

linear.predictors, 7
lmer, 9
loglik.cond, 8, 11

phmm, 4, 6–8, 8, 11, 12, 14
phmm-package, 2
phmm.cond.loglik, 8, 11
plot.phmm, 9, 11
pseudoPoisPHMM, 12

Surv, 9, 10
survfit, 10

traceHat, 5, 12, 14

xypplot, 12