

# Package ‘phmm’

September 22, 2009

**Version** 0.6.1

**Date** 2009-08-31

**Title** Proportional Hazards Mixed-effects Model (PHMM)

**Author** Michael Donohue, Ronghui Xu

**Maintainer** Michael Donohue <mdonohue@ucsd.edu>

**Depends** survival, lattice

**Suggests** lme4

**Description** Fits proportional hazards model incorporating random effects using an EM algorithm using Markov Chain Monte Carlo at E-step.

**License** GPL-2

**Repository** CRAN

**Date/Publication** 2009-09-22 19:46:08

## R topics documented:

|                             |    |
|-----------------------------|----|
| phmm-package . . . . .      | 2  |
| AIC . . . . .               | 3  |
| cAIC . . . . .              | 5  |
| linear.predictors . . . . . | 6  |
| loglik.cond . . . . .       | 7  |
| phmm . . . . .              | 7  |
| phmm.cond.loglik . . . . .  | 10 |
| phmm.design . . . . .       | 11 |
| plot.phmm . . . . .         | 12 |
| pseudoPoisPHMM . . . . .    | 13 |
| traceHat . . . . .          | 15 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>17</b> |
|--------------|-----------|

phmm-package

*Proportional Hazards with Mixed Model (PHMM)***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**

Vaida, F. and Xu, R. "Proportional hazards model with random effects", *Statistics in Medicine*, 19:3309-3324, 2000.

**See Also**

[bayesSurv](#)

**Examples**

```

N <- 100
B <- 100
n <- 50
nclust <- 5
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)
phmmdata <- data.frame(Z)
phmmdata$cluster <- clusters
phmmdata$time <- time
phmmdata$event <- event

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

```

---

AIC

*Akaike Information Criterion for PHMM*


---

**Description**

Function calculating the Akaike information criterion for PHMM fitted model objects, according to the formula  $-2 * \log - likelihood + k * rho$ , 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  $n_{\text{reff}} < 3$ ). The default  $k = 2$ , is for the usual AIC.

**Usage**

```

## 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**

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](#)

**Examples**

```
N <- 100
B <- 100
n <- 50
nclust <- 5
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)
phmndata <- data.frame(Z)
phmndata$cluster <- clusters
phmndata$time <- time
phmndata$event <- event

fit.phmm <- phmm(Surv(time, event)~Z1+Z2+cluster(cluster),
  ~-1+Z1+Z2, phmndata, 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.
```

```

poisphmdata <- pseudoPoisPHMM(fit.phmm)

library(lme4)
fit.lmer <- lmer(m~l+as.factor(time)+z1+z2+
  (-1+w1+w2|cluster)+offset(log(N)),
  poisphmdata, 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)@AICtab
AIC(fit.phmm)

```

---

cAIC

---

*Conditional Akaike Information Criterion for PHMM*


---

## Description

Function calculating the conditional Akaike information criterion (Vaida & Blanchard 2005) for PHMM fitted model objects, according to the formula  $-2 * \log - likelihood + k * rho$ , 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

```
cAIC(object, ..., k = 2)
```

## Arguments

|        |                                                                                                     |
|--------|-----------------------------------------------------------------------------------------------------|
| object | a fitted PHMM model object of class <code>phmm</code> ,                                             |
| ...    | 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**

- Vaida, F, & Blanchard, S. 2005. Conditional Akaike information for mixed-effects models. *Biometrika*, 92(2), 351-.
- Breslow, NE, Clayton, DG. (1993). Approximate Inference in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, Vol. 88, No. 421, pp. 9-25.
- 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-.
- Hodges, JS, & Sargent, DJ. 2001. Counting degrees of freedom in hierarchical and other richly-parameterised models. *Biometrika*, 88(2), 367-.

**See Also**

[phmm](#), [AIC](#)

**Examples**

```
## Not run:
data(e1582)
e1582.fit <- phmm(Surv(time, event)~z1+z2+z3+z4+z5+cluster(cluster),
  ~-1+z1, e1582, Gbs = 100, Gbsvar = 1000, VARSTART = 1,
  NINIT = 10, MAXSTEP = 50, CONVERG=40)
summary(e1582.fit)
cAIC(e1582)
## 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  $\beta'x_{ij} + w'_{ij}b_i$ .

**See Also**

[phmm](#)

---

|             |                                        |
|-------------|----------------------------------------|
| loglik.cond | <i>PHMM conditional log-likelihood</i> |
|-------------|----------------------------------------|

---

**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 | <i>Proportional Hazards Model with Mixed Effects</i> |
|------|------------------------------------------------------|

---

**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, random, data, subset, na.action, Sigma, varcov, NINIT,  
      VARSTART, MAXSTEP, CONVERG, Gbs, Gbsvar, verbose)
```

## Arguments

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>formula</code>   | model formula for the fixed part of the model, i.e. the part that specifies $\beta'x_{ij}$ . See <code>survreg</code> for further details. Intercept is implicitly included in the model by estimation of the error distribution. As a consequence <code>-1</code> in the model formula does not have any effect on the model. The left-hand side of the <code>formula</code> must be an object created using <code>Surv</code> . If <code>random</code> is used then the formula must contain an identification of clusters in the form <code>cluster(id)</code> , where <code>id</code> is a name of the variable that determines clusters. |
| <code>random</code>    | formula for the random part of the model, i.e. the part that specifies $w'_{ij}b_i$ . If omitted, no random part is included in the model. E.g. to specify the model with a random intercept, say <code>random=~1</code> .                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>data</code>      | optional data frame in which to interpret the variables occurring in the formulas.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>subset</code>    | subset of the observations to be used in the fit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>na.action</code> | function to be used to handle any NAs in the data. The user is discouraged to change a default value <code>na.fail</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>Sigma</code>     | initial covariance matrix for the random effects. Defaults to "identity".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>varcov</code>    | constraint on <code>Sigma</code> . Currently only "diagonal" is supported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>NINIT</code>     | number of starting values supplied to Adaptive Rejection Metropolis Sampling (ARMS) algorithm.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>VARSTART</code>  | starting value of the variances of the random effects.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>MAXSTEP</code>   | number of EM iterations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>CONVERG</code>   | iteration after which Gibbs sampling size changes from <code>Gbs</code> to <code>Gbsvar</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>Gbs</code>       | initial Gibbs sampling size (until <code>CONVERG</code> iterations).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>Gbsvar</code>    | Gibbs sampling size after <code>CONVERG</code> iterations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>verbose</code>   | Set to <code>TRUE</code> to print EM steps.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## 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 `plot.phmm` method. For more on MCMC-EM convergence see Booth & Hobart (1999).

## Value

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

|                    |                                                             |
|--------------------|-------------------------------------------------------------|
| <code>steps</code> | a matrix of estimates at each EM step;                      |
| <code>bhat</code>  | empirical Bayes estimates of expectation of random effects; |

|        |                                                                                                                                                                                                                                    |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sdbhat | empirical Bayes estimates of standard deviation of random effects;                                                                                                                                                                 |
| coef   | the final parameter estimates for the fixed effects;                                                                                                                                                                               |
| var    | the estimated variance-covariance matrix;                                                                                                                                                                                          |
| loglik | a vector of length four with the conditional log-likelihood and marginal log-likelihood estimated by Laplace approximation, reciprocal importance sampling, and bridge sampling (only implemented for <code>nreff &lt; 3</code> ); |
| lambda | the estimated baseline hazard;                                                                                                                                                                                                     |
| Lambda | the estimated cumulative baseline hazard.                                                                                                                                                                                          |

## References

- Gilks WR & Wild P. (1992) Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* 41, pp 337-348.
- Vaida F & Xu R. 2000. "Proportional hazards model with random effects", *Statistics in Medicine*, 19:3309-3324.
- Gamst A, Donohue M, & Xu R (2009). Asymptotic properties and empirical evaluation of the NPMLE in the proportional hazards mixed-effects model. *Statistica Sinica*, 19, 997.
- Xu R, Gamst A, Donohue M, Vaida F, & Harrington DP. 2006. Using Profile Likelihood for Semi-parametric Model Selection with Application to Proportional Hazards Mixed Models. *Harvard University Biostatistics Working Paper Series*, Working Paper 43.
- Booth JG & Hobert JP. Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society, Series B* 1999; 61:265-285.

## See Also

[survfit](#), [Surv](#).

## Examples

```
N <- 100
B <- 100
n <- 50
nclust <- 5
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)
phmmdata <- data.frame(Z)
phmmdata$cluster <- clusters
phmmdata$time <- time
phmmdata$event <- event

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

```

---

phmm.cond.loglik     *PHMM conditional log-likelihood*

---

## Description

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 (Nx <code>nfixed</code> ) of covariates for fixed effects.   |
| beta  | Fitted fixed effects coefficients (p-vector).                               |
| w     | Numeric matrix (Nx <code>nrandom</code> ) of covariates for random effects. |
| b     | Numeric matrix (Nx <code>nrandom</code> ) of random effects estimates.      |

## Value

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

## See Also

[phmm](#), [loglik.cond](#)

---

phmm.design                      *PHMM Design*

---

### Description

Internal function for extracting design matrix from call to `phmm`. Code adapted from `bayesurvreg.design` function in the `bayesSurv` package.

### Usage

```
phmm.design(m, formula, random, data)
```

### Arguments

|                      |                                                    |
|----------------------|----------------------------------------------------|
| <code>m</code>       | match.call from call to <code>phmm</code> .        |
| <code>formula</code> | formula component from call to <code>phmm</code> . |
| <code>random</code>  | random component from call to <code>phmm</code> .  |
| <code>data</code>    | data component from call to <code>phmm</code> .    |

### Value

|                        |                                                                                                                                                                                                                                                                                            |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>n</code>         | number of observations (in the case of bivariate data, this is a number of single observations, i.e. $2 \times$ sample size) included in the dataset                                                                                                                                       |
| <code>ncluster</code>  | number of clusters included in the dataset. In the case of bivariate data this is equal to the number of bivariate observations. If there are no random effects included in the model and if the observations are not bivariate then <code>ncluster = n</code>                             |
| <code>nwithin</code>   | a~vector of length equal to <code>ncluster</code> with numbers of observations within each cluster. In the case of bivariate observations this is a~vector filled with 2's, if there are no random effects and if the observations are not bivariate then this is a~vector filled with 1's |
| <code>nY</code>        | number of columns in the response matrix $Y$ . This is equal to 2 if there are no interval-censored observations and equal to 3 if there is at least one interval censored observation in the dataset                                                                                      |
| <code>nZ</code>        | number of columns in the design matrix $Z$ . Note that the matrix $Z$ contains covariates for both fixed and random effects                                                                                                                                                                |
| <code>nfixed</code>    | number of fixed effects involved in the model. Note that possible intercept is always removed from the model                                                                                                                                                                               |
| <code>nrandom</code>   | number of random effects in the model, possible random intercept included                                                                                                                                                                                                                  |
| <code>randomInt</code> | TRUE/FALSE indicating whether the random intercept is included in the model                                                                                                                                                                                                                |
| <code>Y</code>         | response matrix. Its last column is always equal to the status indicator (1 for exactly observed event times, 0 for right-censored observations, 2 for left-censored observations, 3 for interval-censored observations).                                                                  |

|                           |                                                                                                                                                                                                                                                                           |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Z</code>            | design matrix containing covariates for fixed effects.                                                                                                                                                                                                                    |
| <code>W</code>            | design matrix containing covariates for random effects.                                                                                                                                                                                                                   |
| <code>Yinit</code>        | response matrix extracted from <code>formula</code> using <code>model.extract</code>                                                                                                                                                                                      |
| <code>Zinit</code>        | design matrix extracted from <code>formula</code> using <code>model.matrix</code> function                                                                                                                                                                                |
| <code>cluster</code>      | a~vector of length <code>n</code> with identifications of clusters (as given by <code>cluster</code> in <code>formula</code> )                                                                                                                                            |
| <code>indb</code>         | a~vector of length <code>nZ</code> identifying fixed and random effects. <code>indb[j] = -1</code> if the $j$ th column of matrix $Z$ is a fixed effects. it is equal to $l$ if the $j$ th column of matrix $Z$ corresponds to the $l$ th random effect (in C++ indexing) |
| <code>rnames.Z</code>     | row names of <code>Zinit</code>                                                                                                                                                                                                                                           |
| <code>names.random</code> | column names of the $Z$ matrix corresponding to the random effects. If there is the random intercept in the model, the first component of this vector is equal to "Intercept"                                                                                             |

## References

Arnost Komarek (2007). bayesSurv: Bayesian Survival Regression with Flexible Error and Random Effects Distributions. R package version 0.5-9. <http://www.karlin.mff.cuni.cz/~komarek>

## See Also

[phmm](#), [phmm](#).

---

plot.phmm

*Plots the convergence of MCMC-EM estimates from a PHMM*

---

## Description

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, ...)
```

## Arguments

`x` phmm object return by [phmm](#)  
`...` other arguments passed to [xyplot](#)

## References

Booth JG & Hobart JP. Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society, Series B* 1999; 61:265-285.

**See Also**[phmm](#).

---

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:

|                                        |                                                                                                   |
|----------------------------------------|---------------------------------------------------------------------------------------------------|
| <code>time</code>                      | the event time;                                                                                   |
| <code>N</code>                         | the number at risk at time <code>time</code> ;                                                    |
| <code>m</code>                         | the number at risk (in the same cluster with same covariates) at time <code>time</code> ;         |
| <code>cluster</code>                   | the integer cluster indicator;                                                                    |
| <code>N</code>                         | the number at risk at time <code>time</code> ;                                                    |
| <code>fixed effects covariates</code>  | denoted <code>z1, z2, etc.</code> ;                                                               |
| <code>random effects covariates</code> | denoted <code>w1, w2, etc.</code> ;                                                               |
| <code>linear.predictors</code>         | the linear predictors from the <code>phmm</code> fit (excluding the cumulative hazard estimates). |

**References**

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](#), [traceHat](#)

**Examples**

```

## Not run:
N <- 100
B <- 100
n <- 50
nclust <- 5
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)
phmmdata <- data.frame(Z)
phmmdata$cluster <- clusters
phmmdata$time <- time
phmmdata$event <- event

fit.phmm <- phmm(Surv(time, event)~Z1+Z2+cluster(cluster),
                ~-1+Z1+Z2, phmmdata, 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.
poisphmmdata <- pseudoPoisPHMM(fit.phmm)

library(lme4)
fit.lmer <- lmer(m~-1+as.factor(time)+z1+z2+
                (-1+w1+w2|cluster)+offset(log(N)),
                poisphmmdata, 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 approximation via a generalized linear mixed-effects model (GLMM) (Donohue, et al, submitted).

## Usage

```
traceHat(x, method="direct")
traceHat.default(z, w, cluster, Sigma, fitted)
```

## Arguments

|                      |                                                                                                  |
|----------------------|--------------------------------------------------------------------------------------------------|
| <code>x</code>       | an object of class <code>phmm</code> ,                                                           |
| <code>method</code>  | acceptable values are "direct", "pseudoPois", "HaLee" or <code>coxph</code> (no random effects), |
| <code>z</code>       | matrix of fixed effects covariates,                                                              |
| <code>w</code>       | matrix of random effects covariates,                                                             |
| <code>cluster</code> | integer valued cluster indices,                                                                  |
| <code>Sigma</code>   | variance-covariance matrix of fit,                                                               |
| <code>fitted</code>  | linear predictors (fixed and random).                                                            |

## Value

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

## References

- Breslow, NE, Clayton, DG. (1993). Approximate Inference in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, Vol. 88, No. 421, pp. 9-25.
- Donohue, M, Xu, R, Vaida, F, Haut R. Model Selection for Clustered Data: Conditional Akaike Information under GLMM and PHMM. Submitted.
- Ha, ID, Lee, Y, MacKenzie, G. (2007). Model Selection for multi-component frailty models. *Statistics 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 <- 100
B <- 100
n <- 50
nclust <- 5
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)
phmmdata <- data.frame(Z)
phmmdata$cluster <- clusters
phmmdata$time <- time
phmmdata$event <- event

fit.phmm <- phmm(Surv(time, event)~Z1+Z2+cluster(cluster),
                 ~-1+Z1+Z2, phmmdata, 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.
poisphmmdata <- pseudoPoisPHMM(fit.phmm)

library(lme4)
fit.lmer <- lmer(m~-1+as.factor(time)+z1+z2+
                (-1+w1+w2|cluster)+offset(log(N)),
                poisphmmdata, 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)

```

# Index

## \*Topic **package**

phmm-package, 1

## \*Topic **survival**

AIC, 3

cAIC, 5

linear.predictors, 6

loglik.cond, 7

phmm, 7

phmm.cond.loglik, 10

phmm.design, 11

plot.phmm, 12

pseudoPoisPHMM, 13

traceHat, 15

AIC, 3, 4, 6

AIC.phmm, 15

bayesSurv, 2

cAIC, 5

linear.predictors, 6

loglik.cond, 7, 10

model.extract, 12

model.matrix, 12

phmm, 4, 6, 7, 7, 10, 12, 13, 15

phmm-package, 1

phmm.cond.loglik, 7, 10

phmm.design, 11

plot.phmm, 8, 12

pseudoPoisPHMM, 13

Surv, 8, 9

survfit, 9

survreg, 8

traceHat, 13, 15

xyplot, 12