

Package ‘phmm’

January 2, 2012

Version 0.7-1

Date 2010-12-04

Title Proportional Hazards Mixed-effects Model (PHMM)

Author Michael Donohue, Ronghui Xu

Maintainer Michael Donohue <mdonohue@ucsd.edu>

Depends survival, lattice, Matrix

Suggests lme4

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

License GPL-3

Repository CRAN

Date/Publication 2010-12-05 09:34:10

R topics documented:

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

Index	17
--------------	-----------

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[bayessurvreg1](#)**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)

```

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 <- 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)

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

```

poisphmmd <- 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(poisphmmd, "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)@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, method = "direct", ..., k = 2)
```

Arguments

object	A fitted PHMM model object of class phmm.
method	Passed to <code>traceHat</code> . 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

- 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:
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)

# Same data can be fit with lmer,
# though the correlation structures are different.
poisphmmd <- 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(phisphmmd, "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)@AICtab
AIC(fit.phmm)
cAIC(fit.phmm)

## 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, 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)
```

Arguments

formula	model formula for the fixed and random components of the model (as in 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 formula must be a 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 NAs in the data. The user is discouraged to change a default value <code>na.fail</code> .
Sigma	initial covariance matrix for the random effects. Defaults to "identity".
varcov	constraint on 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 Gbs to Gbsvar.
Gbs	initial Gibbs sampling size (until CONVERG iterations).
Gbsvar	Gibbs sampling size after CONVERG iterations.
verbose	Set to TRUE to print EM steps.
maxtime	maximum time in seconds, before aborting EM iterations. Defaults to 120 seconds.
random	The argument random is no longer used. Random components are expressed in 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 [plot.phmm](#) method. For more on MCMC-EM convergence see Booth & Hobart (1999).

Value

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

steps	a matrix of estimates at each EM step;
bhat	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 nreff < 3);
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 Semiparametric 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 <- 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
```

```

phmd$time <- time
phmd$event <- event

fit.phmm <- phmm(Surv(time, event) ~ Z1 + Z2 + (-1 + Z1 + Z2 | cluster),
  phmd, 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 (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](#)

Value

A data.frame with columns:

time	the event time;
N	the number at risk at time time;
m	the number at risk (in the same cluster with same covariates) at time time;
cluster	the integer cluster indicator;
N	the number at risk at time time;
fixed effects covariates	denoted z1, z2, etc.;
random effects covariates	denoted w1, w2, etc.;
linear.predictors	the linear predictors from the phmm 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 <- 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)

# Same data can be fit with lmer,
# though the correlation structures are different.
poisphmmd <- 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(poisphmmd, "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 approximation via a generalized linear mixed-effects model (GLMM) (Donohue, et al, submitted).

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

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 <- 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)

# Same data can be fit with lmer,
# though the correlation structures are different.
poisphmmd <- 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(phisphmmd, "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)
```

Index

*Topic **package**

phmm-package, 2

*Topic **survival**

AIC, 3

cAIC, 5

linear.predictors, 7

loglik.cond, 8

phmm, 8

phmm.cond.loglik, 11

plot.phmm, 12

pseudoPoisPHMM, 12

traceHat, 14

traceHat, 5, 13, 14

xyplot, 12

AIC.phmm (AIC), 3

cAIC (cAIC), 5

linear.predictors (linear.predictors),
7

loglik.cond (loglik.cond), 8

phmm.cond.loglik (phmm.cond.loglik), 11

pseudoPoisPHMM (pseudoPoisPHMM), 12

traceHat (traceHat), 14

AIC, 3, 4, 6

AIC.phmm, 15

bayessurvreg1, 3

cAIC, 5

linear.predictors, 7

lmer, 9

loglik.cond, 8, 11

phmm, 4, 6, 7, 8, 8, 11–13, 15

phmm-package, 2

phmm.cond.loglik, 8, 11

plot.phmm, 9, 12

pseudoPoisPHMM, 12

Surv, 9, 10

survfit, 10