Package ‘eglhmm’

November 1, 2023

Version 0.1-1
Date 2023-11-01
Title Extended Generalised Linear Hidden Markov Models
Maintainer Rolf Turner <rolfturner@posteo.net>
Imports dbd, nnet
Suggests R.rsp
VignetteBuilder R.rsp
LazyData true
ByteCompile true
NeedsCompilation yes
Depends R (>= 3.5)
License GPL (>= 2)
Author Rolf Turner [aut, cre]
Repository CRAN
Date/Publication 2023-11-01 08:40:02 UTC

R topics documented:

  anova.eglhmm .................................................. 2
  bcov .......................................................... 3
  eglhmm ......................................................... 4
  fitted.eglhmm ................................................ 12
  ionChannelData ................................................ 13
  miscprints .................................................... 14
  plot.eglhmm .................................................... 15
Test for a difference between two fitted eglhmm models.

Description

Apply a likelihood ratio test to determine whether the difference, between the log likelihood statistics of two fitted eglhmm models, is statistically significant.

Usage

```r
## S3 method for class 'eglhmm'
anova(object, ...)
```

Arguments

- `object` - An object of class "eglhmm" as returned by `eglhmm()`.
- `...` - Precisely one more object of class "eglhmm", to be compared with `object`.

Details

This anova method handles only comparisons between two models. The order of the arguments (i.e. which object is passed as "object" and which is passed as the sole entry of the `...` argument) is immaterial.

Value

A list with components

- `stat` the likelihood ratio statistic, i.e. the difference between the log likelihoods of the two models. That for the model with the smaller number of parameters is subtracted from that for the model with the larger number.
- `df` the degrees of freedom of the likelihood ratio statistic, i.e. the difference between the number of parameters of the respective models. The smaller number is subtracted from the larger.
- `pvalue` the p-value of the test as given by `pchisq(stat, df, lower.tail = FALSE)`.

This list has an attribute "details" consisting of a numeric vector of length four with entries `ll1` (the smaller of the log likelihoods), `ll2` (the larger of the log likelihoods), `np1` (the smaller of the parameter counts) and `np2` (the larger of the parameter counts).
**b cov**

**Author(s)**

Rolf Turner <rolfturner@posteo.net>

**Examples**

```r
fit1 <- eglhmm(formula=y~locn+depth,data=SydColCount,
cells=c("locn","depth"),distr="P",K=2,
method="em",verb=TRUE)
fit2 <- eglhmm(formula=y~locn+depth+ma.com+nh.com+bo.com,data=SydColCount,
cells=c("locn","depth"),distr="P",K=2,
method="em",verb=TRUE)
anova(fit1,fit2)
```

---

**Description**

Creates an estimate of the covariance matrix of the parameter estimates for an extended generalised linear hidden Markov model via parametric bootstrapping.

**Usage**

```r
b cov(object, nsim = 50, itmax = 500, verbose = TRUE)
```

**Arguments**

- **object** An object of class `eglhmm` as produced by `eglhmm()`.
- **nsim** The number of data sets to simulate, from which to estimate parameters. From each data set a vector of parameters is estimated; the estimated covariance matrix is the empirical covariance matrix of these `nsim` vectors.
- **itmax** The maximum number of iterations to be used in attempting to achieve convergence when fitting models to the simulated data sets. Note that if convergence is not achieved, the simulated data set being used is discarded (i.e. it “doesn’t count”) and a replacement data set is simulated.
- **verbose** Logical scalar. Should a “progress report” be printed out at each step of the fitting procedure?

**Value**

A list with components:

- **C_hat** The parametric bootstrap estimate of the covariance matrix of the parameter estimates.
- **nc.count** A count of the total number of times that the algorithm failed to converge during the bootstrapping procedure.
- **an.count** A count of the “anomalies” that occurred, i.e. the number of times that there was a decrease in the log likelihood. Present only if the method used in fitting the models is "em".
Remarks

Although this documentation refers to “extended generalised linear models”, the only such models currently (01/11/2023) available are the Gaussian model with the identity link, the Poisson model, with the log link, the Binomial model with the logit link, the Dbd (discretised beta distribution model), and the Multinom model. The latter two are generalised linear models only in the “extended” sense. Other models may be added at a future date.

When eglhmm() is called by bcov() the argument checkDecrLL is set equal to FALSE. This has an effect only when the method used in fitting the models is "em". In this case a decrease in the log likelihood is treated as meaning that the algorithm has converged. Setting checkDecrLL equal to FALSE is done so as to decrease the number of discarded data sets and thereby speed up the rate at which the iterations proceed.

Author(s)

Rolf Turner <rolfturner@posteo.net>

References

See the help for eglhmm() for references.

See Also

fitted.eglhmm() reglhmm() reglhmm.default() reglhmm.eglhmm()

Examples

# Takes too long.
fitP  <- eglhmm(y~locn+depth, data=SydColCount, distr="P", cells=c("locn","depth"),
               K=2, contr="sum", verb=TRUE, itmax=300)


#eglhmm
Fit (extended) generalised linear hidden Markov models.

Description

Fits an (extended) generalised linear model to a data set where the response in each “cell” of the model consists of a time series whose serial dependence is modelled by a hidden Markov model.
Usage

eglhmm(formula = NULL, response = NULL, data,
        distr = c("Gaussian", "Poisson", "Binomial", "Dbd", "Multinom", "discnp"),
        inclTau=TRUE,preSpecSigma=NULL, indep = NULL, size = NULL, nbot = NULL, ntop = NULL,
        cells = NULL, K = NULL, par0 = NULL, randStart = NULL,
        method = c("lm", "em", "bf"), optimiser = c("optim", "nlm"),
        optimMethod = "BFGS", nlmWarn = FALSE, lmc = 10, tolerance = NULL,
        digits = NULL, verbose = FALSE, itmax = 200,
        contrast = c("treatment", "sum", "helmert"),
        crit = c("CLL", "L2", "Linf", "ABSGRD"), breaks = NULL, hessian = FALSE,
        useAnalGrad = FALSE, ca = FALSE, checkDecrLL=TRUE)

Arguments

formula A model formula specifying the linear predictor for the model. The formula should not include state as a predictor variable. The variable state gets added to the formula automatically. Ignored if the model is bivariate, i.e. if the length of response is 2.

response A character scalar or a length-2 vector of such scalars, specifying the name or names of the response(s). If response is not specified (i.e. if it is left as NULL) then formula (see below) must be specified and response is taken to be the left hand side of formula. (In this case, it is of course univariate.)

data A data frame with columns providing the response(s) and the predictor variables in the model.

distr Character string specifying the distribution of the response(s) ("emissions") variable(s). Currently (01/11/2023) the only distributions accommodated are Gaussian, Poisson, Binomial, Dbd, and Multinom. Note that "discnp" is just an alternative expression for "Multinom". Ignored if the response is bivariate, in which case distr is forcibly set equal to "Multinom". I.e. bivariate models are, currently, fitted only to data in which the emissions have the "Multinom" distribution.

inclTau Logical scalar. Should the transition probability matrix parameters “tau” be included in those that are estimated via the Hessian/gradient paradigm? In this case, they are included in the set of parameters to which the gradient and Hessian are applicable. If not, they are estimated via the method of moments as is done when the EM algorithm is used. In this latter case the dimensions of the Hessian are reduced (by a substantial amount if K is “large”).

preSpecSigma Numeric vector of length K (see below) with strictly positive entries. Ignored if distr is not equal to "Gaussian". This vector provides “pre-specified” values of the standard deviations sigma of the Gaussian distribution associated with each state. If preSpecSigma is specified, then it is used as the value of sigma throughout the fitting process, and sigma is not estimated from the data. If distr is "Gaussian" and preSpecSigma is specified, then an error will be thrown if the length of preSpecSigma is not equal to K, or if any entries of preSpecSigma fail to be strictly positive.
indep  Logical scalar; should the components of a bivariate model be considered to be independent? Ignored unless the model is bivariate (i.e. unless response is of length 2. If the model is bivariate and indep is not specified, an error is thrown.

size  Scalar integer specifying the number of trials in the experiment generating binomial data (see the size argument of \texttt{dbinom()}). Ignored unless distr is equal to "Binomial".

nbot  Scalar integer specifying the lower end (0 or 1) of the range of values of the discretised Beta distribution. Ignored unless distr is "Dbd".

ntop  Scalar integer specifying the upper end of the range of values of the discretised Beta distribution. Ignored unless distr is "Dbd".

cells  A character vector giving the names of the factors (columns of the data data frame) which determine what the “cells” of the model are considered to be. The cells correspond to the combinations of levels of the factors named by \texttt{cells}.

K  Scalar integer specifying the number of states of the hidden Markov model in question. If K is not specified and par0 (see below) is specified, and has a component tpm, then K is set equal to nrow(tpm). If par0 does not have a tpm component, an error is thrown. An error is also thrown in this setting if K is specified to a value different from nrow(tpm).

par0  A list comprising starting values parameter estimates, to be used by the various methods. (See \textbf{method} below.) This list may have components tpm (an estimate of the transition probability matrix), phi (a vector of estimates of the coefficients in the linear predictor in the generalised linear model) and Rho (a matrix, a list of two matrices, or a three dimensional array) that specifies the emission probabilities when distr is "Multinomial".

randStart  Either a logical scalar or a list of three logical scalars named tpm, phi, and Rho. If the former, it is converted internally into a list with entries named tpm, phi and Rho, all having the same value as the original argument. If tpm is TRUE then the (undocumented) function initialise() chooses entries for the starting value of tpm at random; likewise for phi and Rho. If left NULL, this argument defaults to list(tpm=FALSE,phi=FALSE,Rho=FALSE).

method  Character string specifying the method used to fit the model. This may be "lm" (Levenberg-Marquardt algorithm), "em" (EM algorithm) or "bf" ("brute force"). The latter calls upon \texttt{optim()} or \texttt{nlm()} to do the heavy lifting). If the response is bivariate, then method is forcibly (and silently) set equal to "em".

optimiser  Character string specifying which of \texttt{optim()} or \texttt{nlm()} should be used when method is "bf". Ignored unless method is "bf".

optimMethod  Character string specifying the optimisation method to be used by \texttt{optim()}. See \texttt{optim()} for details. Ignored unless method is "bf" and optimiser is "optim".

nlmWarn  The \texttt{nlm()} function sometimes produces, in the first few iterations, warnings to the effect “NA/Inf replaced by maximum positive value”. These warnings are almost surely irrelevant and are annoying. If nlmWarn is FALSE (the default) then these warnings are suppressed. This argument is provided to allow for the remote possibility that the user might want to see these warnings.

lmc  Positive numeric scalar. The initial “Levenberg-Marquardt constant”. Ignored unless method is "lm".
tolerance  Positive numeric scalar. The convergence tolerance to be used. What this value actually means depends upon method. If left as NULL it defaults to $1e^{-6}$ for the bivariate methods, to $\sqrt{\text{.Machine}$double.eps}$ for the "em" and "lm" methods, and to the default value of reltol used by optim() when method is "bf" and optimiser is "optim". It is ignored if method is "bf" and optimiser is "nlm".

digits  Integer scalar. The number of digits to which “progress reports” are printed when verbose (see below) is TRUE. There is a “sensible” default which is calculated in terms of tolerance. This argument is ignored if method is "bf".

verbose  Logical scalar; if TRUE, rudimentary “progress reports” are printed out at appropriate points during the iteration process. The nature of these “reports” varies with method.

itmax  Integer scalar. The maximum number of iterative steps to take. Has a somewhat different meaning when method is "bf", in which case the meaning depends on optimiser. For methods "em" and "lm", if convergence is not achieved by itmax steps, the function gives up, prints a message to this effect, and returns a value with a component converged=FALSE. This returned value may be used as a starting (the value of the argument par0) so that the iterations may be continued from where they left off. Unfortunately this facility is not available when method is "bf".

contrast  Text string specifying the contrast (in respect of unordered factors) (see contrasts() and options()) that will be used when the design matrix is constructed from the model formula. May be abbreviated (e.g. to "t", "s" or "h").

crit  Text string specifying the stopping criterion to be used. Possible values are “CLL” (scaled change in log likelihood), “L2” (scaled square root of the sum of squares of the changes in the parameter estimates), “Linf” (scaled maximum of the absolute value of the changes in the parameter estimates), and “ABSGRD” (scaled maximum of the absolute values of the entries of the gradient vector). The latter only makes sense for the Levenberg-Marquardt algorithm. This argument is ignored if method is "bf". It seems that the "bf" method effectively uses “CLL” when optimiser is "optim". When optimiser is "nlm" it seems that a combination of (something like) “ABSGRD” and “CLL” is used.

breaks  A vector of $K+1$ values used to construct a set of guesses at the states corresponding to each observation. These are in turn used to calculate an initial estimate of the transition probability matrix. There is a “sensible” default (produced by the undocumented function breaker()).

hessian  Logical scalar; should a Hessian matrix obtained by numerical differentiation be returned? Ignored unless method is "bf".

useAnalGrad  Logical scalar; should “analytical” calculation of the gradient be conducted? This argument is ignored unless the method is "bf".

ca  Logical scalar; “check analyticals”. Used only when the method is "bf" and optimiser is "nlm", and is passed on to nlm().

checkDecrLL  Logical scalar; “check for a decrease in the log likelihood”. Ignored unless the method is "em". Should the software check for a decrease in the log likelihood after an EM step? See the Remarks for further discussion.
Value

An object of class "eglhmm", consisting of a list with components:

call
The call by which this object was created. Present so that update() can be
applied to objects returned by eglhmm().

tpm
The estimated transition probability matrix.

ispd
The estimated initial state probability distribution.

phi
Except for the "Multinom" distribution this is the vector of estimated coeffi-
cients of the linear predictor in the generalised linear model. For the "Multinom"
distribution it consists of the entries of Rho (see below) with the final all-zero
column remove. In this case phi is of course redundant.

theta
The vector of parameter estimates that the estimation procedure actually works
with. It consists of the catenation of the non-redundant parameterization of the
transition probability matrix and the vector phi. It is redundant in the case of
the "Multinom" distribution.

Rho
A matrix, or a list of two matrices or a three dimensional array specifying the
emissions probabilities for a multinomial distribution. Present only if distr is
"Multinom".

log.like
The value of the log likelihood of the model evaluated at the parameter esti-
mates, i.e. the (approximately) maximal value of the log likelihood.

gradient
(Not present for the "em" method.) The gradient vector of the log likelihood at
the final parameter estimates; it should be effectively the zero vector.

numHess
(Present only if method is "bf" and only if the argument hessian is TRUE.)
A value of the Hessian matrix (see below), obtained by means of numerical
differentiation.

Hessian
(Present only if method is "lm").) The Hessian matrix, i.e. the matrix of second
partial derivatives of the log likelihood, evaluated at the final parameter esti-
mates. The inverse of the negative of this matrix constitutes an estimate of the
covariance matrix of the parameter estimates.

mu
A data frame with npred+1 columns where npred is the number of predictors
in the model. The rows contain, in their first npred entries, all possible com-
binations of the predictor values. The last (npred+1) entry of each row is the
fitted mean of the Gaussian distribution, as determined by that combination of
predictors. Present only if distr is "Gaussian".

sigma
Numeric vector of length K whose entries consist of the fitted standard deviations
for the underlying Gaussian distribution, corresponding to each of the states.
Present only if distr is "Gaussian" and preSpecSigma is not supplied.

preSpecSigma
Numeric vector equal to the preSpecSigma argument, with names "sigma1",
"sigma2", ..., "sigmaK" added. Present only if distr is "Gaussian" and
preSpecSigma is supplied.

stopCritVal
Numeric scalar equal to the value, assumed by the stopping criterion specified
by the argument crit, at the termination of the algorithm. If the algorithm con-
verged then stopCritVal will be less than tolerance. Not present if method
is "bf". If converged (see below) is NA then stopCritVal is NA also.
anomaly Logical scalar. Did an "anomaly" occur in an application of the EM algorithm? (See Remarks.) Present only if method was equal to "em". This entry of the returned value is provided mainly for use by the bcov() function. Note that anomaly is added to the returned object, irrespective of the value of checkDecrLL. When checkDecrLL is TRUE, anomaly is somewhat redundant, since it will be TRUE if and only if converged is NA. However when checkDecrLL is FALSE, anomaly is informative, since it is not possible to tell from other entries of the returned value when an anomaly has occurred.

converged A logical scalar. For the "lm", and "em" methods it is TRUE if convergence is achieved within itmax iterations and FALSE otherwise. For the "em" method, if checkDecrLL is TRUE, then converged may be NA. See Remarks for some discussion.

For the "bf" method converged is TRUE if the convergence component of the object returned by optim() is equal to 0 or if the code component of the object returned by nlm() is less than or equal to 2, and is FALSE otherwise. When nlm() is used, the value of converged has an attribute "code" equal to the actual value of the code component.

nstep The number of steps (iterations) actually used by the algorithm. For the "lm" and "em" methods this is the number of Levenberg-Marquardt steps, or EM steps, respectively, taken by the algorithm. For the "bf" method it is the counts component of the object returned by optim() when optimiser is "optim" and it is the iterations component of the object returned by nlm() when optimiser is "nlm".

mean A vector of the fitted mean values underlying each combination of observed predictors and state (i.e. corresponding to each entry of y in the data frame used to fit the model. See the description of data below. Present only if distr is "Gaussian".

sd A vector of the fitted values of the standard deviations underlying each combination of observed predictors and state, i.e. corresponding to each entry of y in the data frame used to fit the model. See the description of data below. Present only if distr is "Gaussian".

lambda A vector of estimated values of the Poisson parameter associated with each combination of observed predictors and state, i.e. corresponding to each entry of y in the data frame used to fit the model. See the description of data below. Present only if distr is "Poisson".

p A vector of estimated values of the “success” probabilities associated with each combination of observed predictors and state, i.e. corresponding to each entry of y in the data frame used to fit the model. See the description of data below. Present only if distr is "Binomial".

alpha A numeric vector of the fitted “alpha” parameters, of the discretised Beta distribution, corresponding to each observation. Present only if distr is "Dbd".

beta A numeric vector of the fitted “beta” parameters, of the discretised Beta distribution, corresponding to each observation. Present only if distr is "Dbd".

fy The values of the “emission probability (density)” function, calculated at each observed value, for each state (i.e. at each entry of y in data. See below.) These values are calculated using the (final) fitted parameters.
**message**
A (long) text string that is produced if the EM algorithm encounters the anomaly of a decrease in the log likelihood after an EM step. It warns the user that this has occurred and suggests consulting the help file for an explanation. Present only if method=="em", the anomaly referred to has occurred, and checkDecrLL is TRUE.

**par0**
The starting values used in the estimation procedure. Either those provided by the argument par0 or those created by the (undocumented) function initialise.

**cells**
A character vector indicating the names of the factors specifying the “cells” of the model. (Equal to the cells argument.)

**formula**
The formula for the model that was fitted; equal to the formula argument, augmented by state.

**distr**
Text string specifying the distribution of the response variable. Equal to the distr argument of this function.

**nbot**
Integer scalar. The lower endpoint of the range of values of the discretised beta distribution. Equal to the value of the nbot argument of this function. Present only if distr is "Dbd".

**ntop**
Integer scalar. The upper endpoint of the range of values of the discretised beta distribution. Equal to the value of the nbot argument of this function. Present only if distr is "Dbd".

**size**
Scalar integer equal to the number of trials in the “experiments” generating the data. Equal to the size argument of this function. Present only if distr is "Binomial".

**tolerance**
The convergence tolerance used to fit the model. Equal to the tolerance argument.

**crit**
Character scalar specifying the stopping criterion that was used. Equal to the crit argument of this function. Not present if method is "bf".

**contrast**
Text string specifying the contrast for unordered factors that was used in fitting the model. Equal to the contrast argument of this function.

**method**
The method ("lm", "em", or "bf") used to fit the model. Equal to the method argument.

**stationary**
Logical scalar. Was a stationary Markov chain fitted? Currently (01/11/2023) stationary is always TRUE.

**data**
The data frame to which the model was fitted. It is a rearrangement of the data argument, with rows of that argument replicated K times (once for each state). A state column (factor) has been added, as has a column cf ("cell factor"), which indicates, by means of a single factor, which cell of the model a given row of data corresponds to, is added. The aforementioned rearrangement consists of ordering the cells in the order of the levels of cf. When distr is "Multinom" the "response" variables are coerced into factors.

**bicm**
Numerical scalar. The number by which npar is multiplied to form the BIC criterion. It is essentially the log of the number of observations. See the code of eglhmm() for details.

**AIC**
Numerical scalar. The Akaike Information criterion, calculated as \(-2\times\text{ll} + 2\times\text{npar}\) where l1 is the log likelihood of the fitted model and npar is the number of fitted parameters.
BIC  Numerical scalar. The Bayesian Information criterion, calculated as \(-2*ll + \text{bicm}*npar\) where \(ll\) is the log likelihood of the fitted model, \(npar\) is the number of fitted parameters, and \(\text{bicm}\) is the log of the number of observations.

missFrac  The fraction or proportion of missing values in the observations.

Remarks

**Available models:** Although this documentation refers to (extended) “generalised linear models”, the only such models currently (01/11/2023) available are the Gaussian model with the identity link, the Poisson model, with the log link, and the Binomial model with the logit link. When \(\text{distr}\) is "Dbd" or "Multinom" the model fitted is is a generalised linear model only in a rather extended sense. Even the Gaussian model is not strictly speaking a generalised linear model, since the (state dependent) standard deviations are estimated by a method separate from the generalised linear model paradigm. Other models may be added at a future date.

**Decrease in the log likelihood:** If \(\text{method}\) is equal to "EM" there may be a *decrease* (!!!) in the log likelihood at some EM step. This is “theoretically impossible” but can occur in practice due to an intricacy in the way that the EM algorithm treats \(\text{ispd}\) when \(\text{stationary}\) is TRUE. It turns out to be effectively impossible to maximise the expected log likelihood unless the term in that quantity corresponding to \(\text{ispd}\) is ignored (whence it is ignored). Ignoring this term is “asymptotically negligible” but can have the unfortunate effect of occasionally leading to a decrease in the log likelihood. If \(\text{method}\) is equal to "em", then the object returned by \text{eglhmm}() has a component \text{anomaly} which is TRUE if such a decrease in the log likelihood was detected, and FALSE otherwise.

If such a decrease/anomaly is detected, then (provided that \text{checkDecrLL} is TRUE) the algorithm terminates and the \text{converged} component of the returned value is set equal to \text{NA}. The algorithm issues a message to the effect that the decrease occurred. The message suggests that another method be used and that perhaps the results from the penultimate EM step (which are returned by this function) be used as starting values. This of course is not possible if the response is bivariate, in which case only the EM algorithm is applicable.

Note that if \text{checkDecrLL} is FALSE, then the algorithm proceeds “normally”. That is, it treats the decrease in the log likelihood to mean that the “increase” in the log likelihood is less than \text{tolerance} and deems convergence to be achieved.

The value of \text{checkDecrLL} is set to FALSE in the function \text{bcov}() so as to speed up the rate at which the iterations proceed. In other circumstances it is probably judicious to leave it at its default value of TRUE.

Author(s)

Rolf Turner <rolfturner@posteo.net>

References


fitted.eglhmm

Predict method for extended generalised linear hidden Markov models.

Description

Predicted values based on an extended generalised linear hidden Markov model object.

Usage

## S3 method for class 'eglhmm'
fitted(object, ...)

Arguments

object  
An object of class eglhmm as returned by eglhmm().

...  
Not used.
Value

A vector of fitted values of the same length as that of the observed values (i.e. length equal to the row dimension of the data frame to which the model was fitted. This data frame is equal to object$data but with repeated rows corresponding to different states collapsed to a single row. The row dimension of this data frame is thus nrow(object$data)/K where K is the number of states in the model. This data frame, with columns cf and state omitted, is returned as an attribute data of the vector of fitted values.

Remark

Although this documentation refers to “generalised linear models”, the only such models currently (01/11/2023) available are the Gaussian model with the identity link, the Poisson model, with the log link, and the Binomial model with the logit link. Other models may be added at a future date.

Author(s)

Rolf Turner <rolftturner@posteo.net>

References

See the help for eglhmm() for references.

See Also

reglhmm() reglhmm.default() reglhmm.eglhmm() bcov()

Examples

loc4 <- c("LngRf","BondiE","BondiOff","MlbrOff")
SCC4 <- SydColCount[SydColCount$locn %in% loc4,]
SCC4$locn <- factor(SCC4$locn) # Get rid of unused levels.
rownames(SCC4) <- 1:nrow(SCC4)
fit <- eglhmm(y~locn+depth,data=SCC4,cells=c("locn","depth"),
K=2,distr="P",contr="sum",verb=TRUE)
fv <- fitted(fit)
with(attr(fv,"data"),plot(y[locn=="BondiOff" & depth=="40"],
xlab="time",ylab="count"))
with(attr(fv,"data"),lines(fv[locn=="BondiOff" & depth=="40"]))

ionChannelData

Description

Time series of observations, made by means of patch clamps, of current in picoamps, across cell membranes.
miscprints

Note

These data are **not** immediately available in the eg1hmm package. Their presence would cause the size of the data directory to exceed 4.5 Mb., which is unacceptably large. Consequently these data sets have been placed in a separate “data only” package called ionChannelData, which is available from github. This package may be obtained by executing the command:

```r
install.packages("ionChannelData",repos="https://rolfturner.r-universe.dev")
```

After having installed the ionChannelData package, you may load it via `library(ionChannelData)` and then access the data sets in the usual way, e.g. `X <- ic25kHz_12_sgmnt1`.

Alternatively (after having installed the ionChannelData package) you may use the `::` syntax to access a single data set, e.g. `X <- ionChannelData::ic25kHz_12_sgmnt1`.

You can access the documentation via, e.g. `?ionChannelData::ionChannelData`.

miscprints

**Specialised print methods.**

**Description**

Print objects of class "RhoExpForm", "RhoProbForm" and "kitty", appropriately.

**Usage**

```r
## S3 method for class 'RhoExpForm'
print(x, ...)
## S3 method for class 'RhoProbForm'
print(x, ...)
## S3 method for class 'kitty'
print(x, ...)
```

**Arguments**

- `x` An object of class "RhoExpForm", "RhoProbForm" or "kitty" respectively.
- `...` Not used. Present for compatibility with the generic `print()` function.

**Details**

The methods `print.RhoExpForm()` and `print.RhoProbForm()` are present essentially for debugging purposes only. The method `print.kitty()` is present to improve the appearance of printed output from eg1hmm when there is a "message" component of this output. None of these methods would normally be called by users.

**Value**

None.
Description

For each specified model cell plot an array, with one panel for each state, of the probability mass or density functions corresponding to the given cell and state. The plots are produced with `type="h"` for probability mass functions, and with `type="l"` for probability density functions.

Usage

```r
## S3 method for class 'eglhmml'
plot(x, ..., wcells = NULL, col = "red",
     nrnc = NULL, ntop = NULL, xlab = NULL, ylab = NULL,
     xlim = NULL, ylim = NULL, main = NULL, cex.main = 1.5)
```

Arguments

- `x` An object of class "eglhmml" as returned by the function `eglhmml()`.
- `...` Not used.
- `wcells` Character vector specifying the cells of the model to be plotted. Defaults to all cells (i.e. the levels of `x$data$cf`).
- `col` The colour for the (vertical) lines of the plots.
- `nrnc` An integer vector of length two specifying the dimensions of the array of plots that is produced. The first entry is the number of rows, the second the number of columns. The product of the entries must be greater than or equal to `K`, the number of states in the model.
- `ntop` The largest `x`-value to be used in plots of the Poisson distribution. Defaults to the maximum of the upper $10^{-7}$ quantile of all of the Poisson distributions that are to be plotted. Ignored unless `x$distr` is "Poisson".
- `xlab` An optional label for the `x`-axes of the panels in the array of plots. Defaults to "x".
- `ylab` An optional label for the `y`-axes of the panels in the array of plots. Defaults to "probability" for probability mass functions and to "probability density" for probability density functions.
- `xlim` An optional vector of length two, specifying the `x`-limits for the plots. Defaults to `c(0,ntop)` if `x$distr"Poisson"` and to `c(0,x$sizet) if `x$distr is "Binomial". There is no default if `x$distr is "Gaussian".
- `ylim` An optional vector of length two, specifying the `y`-limits for the plots. Defaults to `c(0,M)` where `M` is the maximum of all of the probabilities or probability density values that are to be plotted.
Optional character vector specifying overall titles for each array panel of plots. Defaults to the names of the model cells. If the length of \texttt{main} is less than the number (\texttt{nwc}) of cells to be plotted, then \texttt{main} is replicated to have length \texttt{nwc}. If the length of \texttt{main} is greater than \texttt{nwc} then entries with index greater than \texttt{nwc} are ignored. If you wish there to be no overall titles for the arrays of plots, specify \texttt{main=""}.

\texttt{cex.main} Expansion factor for the text in the main title (determining the size of the text). Ignored if \texttt{main} is set equal to the empty string.

Details

If plotting is interactive, then the arrays of plots are displayed one at a time, and (except for the last of the plots) the user is prompted with the string "Go?" after each array is plotted. Press \texttt{<return>} to see the next plot.

Value

None.

Author(s)

Rolf Turner <rolftwner@posteo.net>

See Also

\texttt{eglhmm()}

Examples

\begin{verbatim}
loc4 <- c("LngRf","BondiE","BondiOff","MlbrOff")
SCC4 <- SydColCount[SydColCount$locn %in% loc4,]
SCC4$locn <- factor(SCC4$locn) # Get rid of unused levels.
rownames(SCC4) <- 1:nrow(SCC4)
fit <- eglhmm(y~locn+depth,data=SCC4,cells=c("locn","depth"),
               K=2,distr="P",verb=TRUE)
plot(fit)
allcells <- levels(fit$data$cf)
wcells <- allcells[grep("\..60",allcells)]
plot(fit,wcells=wcells,main=c("Longreef","Bondi East","Bondi Offshore",
                           "Malabar Offshore"),ntop=12)
\end{verbatim}

\texttt{postHocGradHess} Obtain gradient and Hessian, post hoc.

Description

Calculates the gradient and Hessian of the log likelihood of an extended generalised hidden Markov model, from the components of a "eglhmm" object, or in certain circumstances, simply extracts these quantities from that object.
Usage

postHocGradHess(object, inclTau=TRUE)

Arguments

object  An object of class "eglhmm" as returned by eglhmm().
inclTau Logical scalar; should the vector of "tau" parameters be included in the parameters under consideration?

Details

If object is the result of fitting a bivariate model (i.e. if it inherits from "eglhmm.bivariate" then an error is thrown. (No gradient or Hessian is available in this case.)

If object$method is "lm" and if inclTau matches the value used in fitting method, then the appropriate gradient and Hessian have already been calculated and are simply extracted from object. If inclTau does not match the value used in fitting method, then the gradient and Hessian are recalculated (by the undocumented function getHgl()) with the value of inclTau being that specified by the function argument.

If object$method is "em" or "bf", then the gradient and Hessian are calculated (by the undocumented function getHgl()) with the value of incl Tau being that specified by the function argument.

If object$method is "bf" then the gradient has been calculated numerically and the Hessian may have been calculated numerically (if the argument hessian of eglhmm() was set equal to TRUE). The corresponding value of the gradient will comprise a component, named "numGrad", of the list returned by this function. The corresponding value of the Hessian, if this was indeed calculated, will comprise a component, named "numHess", of the list returned by this function.

Value

A list with components

- gradient: The gradient of the log likelihood.
- Hessian: The Hessian of the log likelihood.
- numGrad: The numerically calculated gradient of the log likelihood. Present only if object$method is "bf".
- numHess: The numerically calculated Hessian of the log likelihood. Present only if object$method is "bf" and if argument hessian was set equal to TRUE in the call to eglhmm() that produced object.

Author(s)

Rolf Turner <rolfturner@posteo.net>
reglhmm

Simulate data from a hidden generalised linear Markov model.

Description

Takes a specification of the model and simulates the data from that model. The model may be specified in terms of the individual components of that model (the default method). The components include a data frame that provides the predictor variables, and various parameters of the model. For the "reglhmm" method the model is specified as a fitted model, an object of class "eglhmm".

References


See Also

eglhmm()

Examples

```r
fit.em <- eglhmm(y~locn+depth,data=SydColCount,distr="P",
                  cells=c("locn","depth"),K=2,method="em",verb=TRUE)
gh.em <- postHocGradHess(fit.em) # Calculates using inclTau=TRUE.

gh.em.noTau <- postHocGradHess(fit.em,inclTau=FALSE)

fit.lm <- eglhmm(y~locn+depth,data=SydColCount,distr="P",
                 cells=c("locn","depth"),K=2,verb=TRUE)
gh.lm <- postHocGradHess(fit.lm) # Just extracts the relevant components.

gh.lm.noTau <- postHocGradHess(fit.lm,inclTau=FALSE)

fit.bf <- eglhmm(y~locn+depth,data=SydColCount,distr="P",
                 cells=c("locn","depth"),K=2,method="bf",verb=TRUE,
                 hessian=TRUE)
gh.bf <- postHocGradHess(fit.bf) # Calculates using inclTau=TRUE; also
                          # extracts numerically computed quantities.

gh.bf.noTau <- postHocGradHess(fit.bf,inclTau=FALSE) # Calculates; also
                          # extracts numerically # computed quantities.
```
Usage

reglhmm(x,...)
## Default S3 method:
reglhmm(x, formula, response, cells=NULL, data=NULL, nobs=NULL,
        distr=c("Gaussian","Poisson","Binomial","Dbd","Multinom"),
        phi, Rho, sigma, size, ispd=NULL, ntop=NULL, zeta=NULL,
        missFrac = 0, fep=NULL,
        contrast=c("treatment","sum","helmert"),...)
## S3 method for class 'eglhmm'
reglhmm(x, missFrac = NULL, ...)

Arguments

x For the default method, the transition probability matrix of the hidden Markov chain. For the "eglhmm" method, an object of class "eglhmm" as returned by the function eglhmm()

formula The formula specifying the generalised linear model from which data are to be simulated. Note that the predictor variables in this formula must include a factor state, which specifies the state of the hidden Markov chain. Note also that this formula must determine a design matrix having a number of columns equal to the length of the vector phi of model coefficients provided in object (and to the length of psi in the case of the Gaussian distribution). If this condition is not satisfied, an error is thrown.
It is advisable to use a formula specified in the manner y~0+state+... where ... represents the predictors in the model other than state. Of course phi must be supplied in a manner that is consistent with this structure.

response A character vector of length 2, specifying the names of the responses. Ignored unless distr is "Multinom". If distr is "Multinom" and if response is provided appropriately, then the simulated data are bivariate multinomial.

cells A character vector specifying the names of the factors which determine the "cells" of the model. These factors must be columns of the data frame data. (See below.) Each cell corresponds to a time series of (simulated) observations. If cells is not supplied (left equal to NULL) then the model is taken to have a single cell, i.e. data from a "simple" hidden Markov model is generated. The parameters of that model may be time-varying, and still depend on the predictors specified by formula.

data A data frame containing the predictor variables referred to by formula, i.e. the predictors for the model from which data are to be simulated. If data is not specified, the nobs (see below) must be. If data is not specified then formula must have the structure y ~ state or preferably y ~ 0 + state. Of course phi must be specified in a consistent manner.

nobs Integer scalar. The number of observations to be generated in the setting in which the generalised linear model in question is vacuous. Ignored if data is supplied.

distr Character string specifying the distribution of the “emissions” from the model, i.e., of the observations. This distribution determines “emission probabilities”.
**phi**
A numeric vector specifying the coefficients of the linear predictor of the generalised linear model. The length of phi must be equal to the number of columns of the design matrix determined by `formula` and `data`. The entries of phi must match up appropriately with the columns of the design matrix.

**Rho**
A matrix, or a list of two matrices or a three dimensional array specifying the emissions probabilities for a multinomial distribution. Ignored unless `distr` is "Multinomial".

**sigma**
A numeric vector of length equal to the number of states. Its i'th entry is the standard deviation of the (Gaussian) distribution corresponding to the i'th state. Ignored unless `distr` is "Gaussian".

**size**
Integer scalar. The number of trials (sample size) from which the number of "successes" are counted, in the context of the binomial distribution. (I.e. the size parameter of `rbinom()`. Ignored unless `distr` is "Binomial".

**ispd**
An optional numeric vector specifying the initial state probability distribution of the model. If `ispd` is not provided then it is taken to be the stationary/steady state distribution determined by the transition probability matrix `x`. If specified, `ispd` must be a probability vector of length equal to the number of rows (equivalently the number of columns) of `x`.

**ntop**
Integer scalar, strictly greater than 1. The maximum possible value of the db distribution. See `db()`. Used only if `distr` is "Dbd".

**zeta**
Logical scalar. Should zero origin indexing be used? I.e. should the range of values of the db distribution be taken to be \{0,1,2,...,ntop\} rather than \{1,2,...,ntop\}? Used only if `distr` is "Dbd".

**missFrac**
A non-negative scalar, less than 1. Data will be randomly set equal to NA with probability `miss.frac`. Note that for the "eglhmm" method, if "miss.frac" is not supplied then it is extracted from `object`.

**fep**
A list of length 1 or 2. The first entry of this list is a logical scalar. If this is TRUE, then the first entry of the simulated emissions (or at least one entry of the first pair of simulated emissions) is forced to be "present", i.e. non-missing. The second entry of `fep`, if present, is a numeric scalar, between 0 and 1 (i.e. a probability). It is equal to the probability that both entries of the first pair of emissions are present. It is ignored if the emissions are univariate. If the emissions are bivariate but the second entry of `fep` is not provided, then this second entry defaults to the "overall" probability that both entries of a pair of emission are present, given that at least one is present. This probability is calculated from `nafrac`.

**contrast**
A character string, one of "treatment", "helmert" or "sum", specifying what contrast (for unordered factors) to use in constructing the design matrix. (The contrast for ordered factors, which is has no relevance in this context, is left at its default value of "contr.poly".) Note that the meaning of the coefficient vector phi depends on the contrast specified, so make sure that the contrast is the same as what you had in mind when you specified phi!!! Note that for the "eglhmm" method, contrast is extracted from `x`.

... Not used.
Value

A data frame with the same columns as those of data and an added column, whose name is determined from formula, containing the simulated response.

Remark

Although this documentation refers to “generalised linear models”, the only such models currently (01/11/2023) available are the Gaussian model with the identity link, the Poisson model, with the log link, and the Binomial model with the logit link. The Multinomial model, which is also available, is not exactly a generalised linear model; it might be thought of as an “extended” generalised linear model. Other models may be added at a future date.

Author(s)

Rolf Turner <rolfturner@posteo.net>

References


See Also

fitted.eglhmm(), bcov()

Examples

loc4 <- c("LnGrf","BondiE","BondiOff","MlbrOff")
SCC4 <- SydColCount[SydColCount$locn %in% loc4,]
SCC4$locn <- factor(SCC4$locn) # Get rid of unused levels.
rownames(SCC4) <- 1:nrow(SCC4)
Tpm <- matrix(c(0.91,0.09,0.36,0.64),byrow=TRUE,ncol=2)
Phi <- c(0,log(5),-0.34,0.03,-0.32,0.14,-0.05,-0.14)
# The "state effects" are 1 and 5.
Dat <- SCC4[,1:3]
fm1a <- y~0+state+locn+depth
cells <- c("locn","depth")
# The default method.
X <- reglhmm(Tpm,formula=fm1a,cells=cells,data=Dat,distr="P",phi=Phi,
miss.frac=0.75,contrast="sum")
# The "eglhmm" method.
fit <- eglhmm(y~locn+depth,data=SCC4,cells=cells,K=2,
verb=TRUE,distr="P")
Y <- reglhmm(fit)
# Vacuous generalised linear model.
Z <- reglhmm(Tpm,formula=y~0+state,nobs=300,distr="P",phi=log(c(2,7)))
# The "state effects" are 2 and 7.
Reorder the states of a fitted eglhmm model so that the state effects are in decreasing order.

Usage

```r
## S3 method for class 'eglhmm'
reorder(x, ...)
```

Arguments

- `x` An object of class "eglhmm" as returned by the function `eglhmm()`.
- `...` Not used.

Details

The states of a fitted hidden Markov model are usually in a rather arbitrary order, which can sometimes make it difficult to compare different fits. This function reorders the states so that the state corresponding to the "largest state effect" comes first (and so on down the line). What is meant by "largest state effect" depends on whether the distribution used in the model is "Dbd". If the distribution is not "Dbd", then what is meant is simply the largest of those entries of \( \phi \) which correspond to state. (The vector \( \phi \) is the vector of coefficients of the linear predictor in the model. Note that, since the formula for the model is constructed as \( y \sim 0 + \text{state} + \ldots \), the "state" coefficients are unconstrained and there are as many of them as there are states.)

If the distribution in question is "Dbd" then things are a bit more complicated. We calculate the theoretical expected values for "Dbd"s with parameters \( \alpha = \text{alpha[k]} \) and \( \beta = \text{beta[k]} \) where \( \text{alpha[k]} \) and \( \text{beta[k]} \) are the parameter values corresponding to the \( k \)th state. The states are then ordered according to the decreasing order of these expected values. These expected values are the expected values of the emissions given that all predictors other than the state predictors are zero.

Value

An object of class c("eglhmm","reordered") which is identical to the argument \( x \) in most respects. The components which (may) differ are:

- `tpm`
- `ispd`
- `phi`
- `theta`
- `Hessian`
- `gradient`
- `mean` and `sd`, or `lambda` or `p`, or `alpha` and `beta` (depending on which distribution is being used)
The entries of these components will have the same numerical values as before but, given that the ordering of the states has actually changed, will have different orderings, corresponding to the new ordering of the states.

Note that the attribute preSpecSigma of the component theta, may differ from what it was in \( x \).

The returned value will also have a component neworder which is an integer vector providing the indices of the reordering of the states. It also currently (01/11/2023) has a component newlog.like. This should (if there is any justice in the world — but there isn’t!) have the same value as the component log.like. Once I am confident that everything is working as it should, the newlog.like component will be removed.

Author(s)

Rolf Turner <rolfturner@posteo.net>

See Also

eglhmm()

Examples

```r
loc4 <- c("LngRf","BondiE","BondiOff","MlbrOff")
SCC4 <- SydColCount[SydColCount$locn %in% loc4,]
SCC4$locn <- factor(SCC4$locn) # Get rid of unused levels.
rownames(SCC4) <- 1:nrow(SCC4)
fit  <- eglhmm(y~locn+depth,data=SCC4,cells=c("locn","depth"),
               K=2,distr="P",verb=TRUE)
ofit <- reorder(fit)
```

SydColDat

*Sydney coliform bacteria data*

Description

Transformed counts of faecal coliform bacteria in sea water at seven locations: Longreef, Bondi East, Port Hacking “50”, and Port Hacking “100” (controls) and Bondi Offshore, Malabar Offshore and North Head Offshore (outfalls). At each location measurements were made at four depths: 0, 20, 40, and 60 meters.

Usage

SydColCount
SydColDisc
Format

Data frames with 5432 observations on the following 6 variables.

\( y \)  Transformed measures of the number of faecal coliform count bacteria in a sea-water sample of some specified volume. The original measures were obtained by a repeated dilution process. For SydColCount the transformation used was essentially a square root transformation, resulting values greater than 150 being set to \( \text{NA} \). The results are putatively compatible with a Poisson model for the emission probabilities.

For SydColDisc the data were discretised using the \texttt{cut()} function with breaks given by \( c(0,1,5,25,200,\text{Inf}) \) and labels equal to \( c(\text{“lo”, “mlo”, “m”, “mhi”, “hi”}) \).

Note that in the SydColDisc data there are 180 fewer missing values (\( \text{NAs} \)) in the \( y \) column than in the SydColCount data. This is because in forming the SydColCount data (transforming the original data to a putative Poisson distribution) values that were greater than 150 were set equal to \( \text{NA} \), and there were 180 such values.

\( \text{locn} \)  a factor with levels “LngRf” (Longreef), “BondiE” (Bondi East), “PH50” (Port Hacking 50), “PH100” (Port Hacking 100), “BondiOff” (Bondi Offshore), “MlbrOff” (Malabar Offshore) and “NthHdOff” (North Head Offshore)

\( \text{depth} \)  a factor with levels “0” (0 metres), “20” (20 metres), “40” (40 metres) and “60” (60 metres).

\( \text{ma.com} \)  A factor with levels no and yes, indicating whether the Malabar sewage outfall had been commissioned.

\( \text{nh.com} \)  A factor with levels no and yes, indicating whether the North Head sewage outfall had been commissioned.

\( \text{bo.com} \)  A factor with levels no and yes, indicating whether the Bondi Offshore sewage outfall had been commissioned.

Details

The observations corresponding to each location-depth combination constitute a time series. The sampling interval is ostensibly 1 week; distinct time series are ostensibly synchronous. The measurements were made over a 194 week period. See Turner et al. (1998) for more detail.

Oh for fuck’s sake!!!

Blah.

Source

Geoff Coade, of the New South Wales Environment Protection Authority (Australia)

References


Examples

# Select out a subset of four locations:
loc4 <- c("LngRf", "BondiE", "BondiOff", "MlbrOff")
SCC4 <- SydColCount[SydColCount$locn %in% loc4,]
SCC4$locn <- factor(SCC4$locn)  # Get rid of unused levels.
rownames(SCC4) <- 1:nrow(SCC4)
Index

* datagen
  - reglhmm, 18
* datasets
  - ionChannelData, 13
  - SydColDat, 23
* hplot
  - plot.eglhmm, 15
* htest
  - anova.eglhmm, 2
* models
  - bcov, 3
  - eglhmm, 4
  - fitted.eglhmm, 12
  - reglhmm, 18
* print
  - miscprints, 14
* utilities
  - postHocGradHess, 16
anova.eglhmm, 2
bcov, 3, 11–13, 21
contrasts, 7
db, 20
dbinom, 6
eglhmm, 2–4, 4, 12, 13, 15–19, 22, 23
fitted.eglhmm, 4, 12, 12, 21
ic25kHz_12_sgmnt1 (ionChannelData), 13
ic25kHz_13_sgmnt2 (ionChannelData), 13
ic25kHz_14_sgmnt2 (ionChannelData), 13
ic25kHz_15_sgmnt2 (ionChannelData), 13
ic50kHz_06_sgmnt2 (ionChannelData), 13
ic50kHz_08_sgmnt2 (ionChannelData), 13
ic50kHz_09_sgmnt1 (ionChannelData), 13
ic50kHz_10_sgmnt1 (ionChannelData), 13
ionChannelData, 13
miscprints, 14
nlm, 6, 7, 9
optim, 6, 9
options, 7
plot.eglhmm, 15
postHocGradHess, 16
print.kitty (miscprints), 14
print.RhoExpForm (miscprints), 14
print.RhoProbForm (miscprints), 14
reglhmm, 4, 13, 18
reglhmm.default, 4, 12, 13
reglhmm.eglhmm, 4, 12, 13
reorder.eglhmm, 22
SydColCount (SydColDat), 23
SydColDat, 23
SydColDisc (SydColDat), 23