

Package ‘hmm.discnp’

April 8, 2016

Version 0.2-4

Date 2016-04-08

Title Hidden Markov Models with Discrete Non-Parametric Observation Distributions

Author Rolf Turner

Maintainer Rolf Turner <r.turner@auckland.ac.nz>

Depends R (>= 0.99)

Description Fits hidden Markov models with discrete non-parametric observation distributions to data sets. Simulates data from such models. Finds most probable underlying hidden states, the most probable sequences of such states, and the log likelihood of a collection of observations given the parameters of the model.

LazyData true

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-04-08 11:33:08

R topics documented:

colifCount	2
fitted.hmm.discnp	3
hmm	4
lesionCount	8
logLikHmm	9
mps	10
pr	12
sim.hmm	13
sp	15
viterbi	16

Index	19
--------------	-----------

 colifCount

Coliform counts in sea-water samples

Description

Transformed counts of faecal coliform bacteria in sea water samples collected at seven locations near Sydney NSW, Australia. There were four “controls”: Longreef, Bondi East, Port Hacking “50”, and Port Hacking “100” and three “outfalls”: Bondi Offshore, Malabar Offshore and North Head Offshore. At each location measurements were made at four depths: 0, 20, 40, and 60 meters. A large fraction of the counts are missing values.

Usage

colifCount

Format

A data frame with 5432 observations on the following 6 variables.

y Transformed measures of the number of faecal coliform count bacteria. The original measures were obtained by a repeated dilution process.

locn a factor with levels Longreef, Bondi East, Port Hacking 50, Port Hacking 100, Bondi Offshore, Malabar Offshore and North Head Offshore.

depth a factor with levels 0 (0 metres), 20 (20 metres), 40 (40 metres), 60 (60 metres).

ma.com A factor with levels no and yes, indicating whether the Malabar sewage outfall had been commissioned.

nh.com A factor with levels no and yes, indicating whether the North Head sewage outfall had been commissioned.

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. Due to exigencies of weather, the unreliability of boats and other factors the collection times were actually highly irregular and have been rounded to the nearest week. Often no sample was obtained at a given site within a week of the putative collection time, in which the observed count is given as a missing value. In fact over **75%** of the counts are missing. See Turner et al. (1998) for more detail.

Modelling

The hidden Markov models applied to these data by Turner et al. (1998) and by Turner (2008) were much more complex and elaborate than those fitted in the examples in this package. See the references for details.

Source

The data were kindly supplied by Geoff Coade, of the New South Wales Environment Protection Authority (Australia)

References

T. Rolf Turner, Murray A. Cameron, and Peter J. Thomson. Hidden Markov chains in generalized linear models. *Canadian J. Statist.* **26** (1998) 107 – 125.

Rolf Turner. Direct maximization of the likelihood of a hidden Markov model. *Comp. Statist. Data Anal.* **52** (2008) 4147–4160.

fitted.hmm.discnp	<i>Fitted values of a discrete non-parametric hidden Markov model.</i>
-------------------	--

Description

Calculates the fitted values of a discrete non-parametric hidden Markov model as the conditional expectations of the observations, given the entire observation sequence (and the estimated parameters of the model).

Usage

```
## S3 method for class 'hmm.discnp'  
fitted(object, ...)
```

Arguments

object	An object of class <code>hmm.discnp</code> as returned by <code>hmm()</code> .
...	Not used.

Details

This function is essentially a wrapper for `sp()`. This latter function gets called with `mean=TRUE`. The fitted values are the conditional expected values of the observations, given the entire observation sequence. Note that the “conditional expected value” concept makes sense *only* if the observations are *numeric*. If this is not the case, an error is thrown. The observation sequence must be present in `object` (which will be the case if `object` was returned by `hmm()`). If it is not present an error is thrown.

Value

If the observations (stored in `object`) consist of a single sequence, then the object returned is the corresponding sequence of fitted values. If the observations consist of a list of observation sequences, then the object returned is a list of corresponding sequences of fitted values.

Author(s)

Rolf Turner <r.turner@auckland.ac.nz>

See Also

[sp\(\)](#)

Examples

```
# See the help for sim.hmm() for how to generate y.num.
## Not run:
fit.num <- hmm(y.num,K=2,verb=TRUE)
fv      <- fitted(fit.num)

## End(Not run)
```

hmm

Fit a hidden Markov model to discrete data.

Description

Uses the EM algorithm to perform a maximum likelihood fit of a hidden Markov model to discrete data where the observations come from one of a number of finite discrete distributions, depending on the (hidden) state of the Markov chain. These distributions are specified (non-parametrically) by a matrix $R = [\rho_{ij}]$ where $\rho_{ij} = P(Y = y_i | S = j)$, Y being the observable random variable and S being the hidden state.

Usage

```
hmm(y, yval=NULL, par0=NULL, K=NULL, rand.start=NULL, stationary=cis,
     mixture=FALSE, cis=TRUE, tolerance=1e-4, verbose=FALSE,
     itmax=200, crit='PCLL',keep.y=TRUE, data.name=NULL)
```

Arguments

y	A vector of discrete data or a list of such vectors; missing values are allowed.
yval	A vector of possible values for the data; it defaults to the sorted unique values of y. If any value of y does not match some value of yval, it will be treated as a MISSING VALUE.
par0	An optional (<i>named</i>) list of starting values for the parameters of the model, with components tpm (transition probability matrix) and Rho. The matrix Rho specifies the probability that the observations take on each value in yval, given the state of the hidden Markov chain. The columns of Rho correspond to states, the rows to the values of yval.

	<p>If <code>par0\$tpm</code> has row and column names these must be the same (otherwise an error is thrown). If <code>par0\$Rho</code> has column names these must be the same as the row and column names of <code>par0\$tpm</code>.</p> <p>If <code>par0</code> is not specified, starting values are created by the function <code>init.all()</code>.</p>
K	<p>The number of states in the hidden Markov chain; if <code>par0</code> is not specified K MUST be; if <code>par0</code> is specified, K is ignored.</p> <p>Note that K=1 is acceptable; if K is 1 then all observations are treated as being independent and the non-parametric estimate of the distribution of the observations is calculated in the obvious way.</p>
rand.start	<p>A list consisting of two logical scalars which must be named <code>tpm</code> and <code>Rho</code>, if <code>tpm</code> is TRUE then the function <code>init.all()</code> chooses entries for then starting value of <code>tpm</code> at random; likewise for <code>Rho</code>. This argument defaults to <code>list(tpm=FALSE, Rho=FALSE)</code>.</p>
stationary	<p>Logical scalar. If TRUE then the model is fitted under the stationarity assumption, i.e. that the Markov chain was in steady state at the time that observations commenced. In this case the initial state probability distribution is estimated as the stationary distribution determined by the (estimated) transition probability matrix. Otherwise if <code>cis</code> (see below) is TRUE the initial state probability distribution is estimated as the mean of the vectors of conditional probabilities of the states, given the observation sequences, at time <code>t=1</code>. If <code>stationary</code> is TRUE and <code>cis</code> is FALSE an error is given.</p>
mixture	<p>A logical scalar; if TRUE then a mixture model (all rows of the transition probability matrix are identical) is fitted rather than a general hidden Markov model.</p>
cis	<p>A logical scalar specifying whether there should be a constant initial state probability distribution. If <code>stationary</code> is FALSE and <code>cis</code> is FALSE then the initial state probability distribution for a given observation sequence is equal to 1 where the (first) maximum of the vector of conditional probabilities of the states, given the observation sequences, at time <code>t=1</code>, occurs, and is 0 elsewhere. If <code>stationary</code> is TRUE and <code>cis</code> is FALSE an error is given.</p>
tolerance	<p>If the value of the quantity used for the stopping criterion is less than <code>tolerance</code> then the EM algorithm is considered to have converged.</p>
verbose	<p>A logical scalar determining whether to print out details of the progress of the EM algorithm.</p>
itmax	<p>If the convergence criterion has not been met by the time <code>itmax</code> EM steps have been performed, a warning message is printed out, and the function stops. A value is returned by the function anyway, with the logical component "converged" set to FALSE.</p>
crit	<p>The name of the stopping criterion, which must be one of "PCLL" (percent change in log-likelihood; the default), "L2" (L-2 norm, i.e. square root of sum of squares of change in coefficients), or "Linf" (L-infinity norm, i.e. maximum absolute value of change in coefficients).</p>
keep.y	<p>Logical scalar; should the observations <code>y</code> be returned as a component of the value of this function?</p>
data.name	<p>An identifying tag for the fit; if omitted, it defaults to the name of data set <code>y</code> as determined by <code>deparse(substitute(y))</code>.</p>

Details

The hard work is done by a Fortran subroutine "recurse" (actually coded in Ratfor) which is dynamically loaded.

Value

A list with components:

Rho	The fitted value of the probability matrix Rho specifying the distributions of the observations (the "emission" probabilities).
tpm	The fitted value of the transition probability matrix tpm.
ispd	The fitted initial state probability distribution, or a matrix of (trivial or "deterministic") initial state probability distributions, one (column) of ispd for each observation sequence. If <code>stationary</code> is TRUE then ispd is assumed to be the (unique) stationary distribution for the chain, and thereby determined by the transition probability matrix tpm. If <code>stationary</code> is FALSE and <code>cis</code> is TRUE then ispd is estimated as the mean of the vectors of conditional probabilities of the states, given the observation sequences, at time $t=1$. If <code>cis</code> is FALSE then ispd is a matrix whose columns are trivial probability vectors, as described above.
log.like	The final value of the log likelihood, as calculated through recursion.
converged	A logical scalar saying whether the algorithm satisfied the convergence criterion before the maximum of <code>itmax</code> EM steps was exceeded.
nstep	The number of EM steps performed by the algorithm.
y	The observations (argument <code>y</code>). Present only if <code>keep.y</code> is TRUE.
data.name	An identifying tag, specified as an argument, or determined from the name of the argument <code>y</code> by <code>deparse(substitute(y))</code> .
stationary	The argument <code>stationary</code> .
cis	The argument <code>cis</code> .

Notes

The package *used* to require the argument `y` to be a *matrix* in the case of multiple observed series. If the series were of unequal length the user was expected to pad them out with NAs to equalize the lengths. In the revision from version 0.0-9 to 0.1-0 this was changed, requiring the argument `y` to be (more sensibly) a list when there are multiple series. (Unfortunately this help page was *not* correspondingly revised at the time of that transition nor of the next one, and the old "matrix" format was left as being the specified input format until the package was updated to version 0.1-2.)

The old matrix format is still permitted (and the matrix is internally changed into a list) but this is deprecated. In some future version of `hmm.discnp` this possibility will be removed.

If $K=1$ then `tpm`, `ispd`, `converged`, and `nstep` are all set equal to NA in the list returned by this function.

The estimate of `ispd` in the non-stationary setting is inevitably very poor, unless the number of sequences of observations (the length of the list `y`) is very large. We have in effect “less than one” relevant observation for each such sequence.

The returned values of `tpm` and `Rho` have row and column names. These are the same as the row and column names of the starting values of these matrices (as provided in `par0` if these exist. Otherwise they are taken to be the appropriate sequences of integers. Likewise the returned value of `ispd` is a named vector, the names being the same as the row (and column) names of `tpm` and the column names of `Rho`.

Warnings

The ordering of the (hidden) states can be arbitrary. What the estimation procedure decides to call “state 1” may not be what *you* think of as being state number 1. The ordering of the states will be affected by the starting values used.

At some time in the future the (deprecated) option of being able to specify argument `y` as a matrix (in the setting in which there are multiple data series to which a single model is being fitted) will no longer be permitted.

Author(s)

Rolf Turner <r.turner@auckland.ac.nz>

References

- Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE vol. 77, pp. 257 – 286, 1989.
- Zucchini, W. and Guttorp, P., "A hidden Markov model for space-time precipitation," Water Resources Research vol. 27, pp. 1917-1923, 1991.
- MacDonald, I. L., and Zucchini, W., "Hidden Markov and Other Models for Discrete-valued Time Series", Chapman & Hall, London, 1997.
- Liu, Limin, "Hidden Markov Models for Precipitation in a Region of Atlantic Canada", Master's Report, University of New Brunswick, 1997.

See Also

[sim.hmm\(\)](#), [mps\(\)](#), [viterbi\(\)](#)

Examples

```
# 1.
Yval <- LETTERS[1:10]
Tpm <- matrix(c(0.75,0.25,0.25,0.75),ncol=2,byrow=TRUE)
Rho <- cbind(c(rep(1,5),rep(0,5)),c(rep(0,5),rep(1,5)))/5
rownames(Rho) <- Yval
set.seed(42)
xxx <- sim.hmm(rep(1000,5),tpm=Tpm,Rho=Rho,yval=Yval)
fit <- hmm(xxx,par0=list(tpm=Tpm,Rho=Rho))
print(fit$Rho)
```

```

# 2.
# See the help for sim.hmm() for how to generate y.num.
## Not run:
  fit.num      <- hmm(y.num,K=2,verb=TRUE)
  fit.num.mix  <- hmm(y.num,K=2,verb=TRUE,mixture=TRUE)
  print(fit.num[c("tpm","Rho")])

## End(Not run)
# Note that states 1 and 2 get swapped.

# 3.
xxx <- with(colifCount,split(y,f=locn))
fitColCount <- hmm(xxx,K=2) # Two states: above and below the thermocline.

# 4.
fitLesCount <- hmm(lesionCount,K=2) # Two states: relapse and remission.

```

lesionCount

Multiple sclerosis lesion counts for three patients.

Description

Lesion counts for three multiple sclerosis patients. The counts were obtained by magnetic resonance imaging, and were observed at monthly intervals.

Usage

```
lesionCount
```

Format

A list with three components each component being the sequence of counts for a given patient and consisting of a vector with non-negative integer entries.

Modelling

The hidden Markov models applied to these data by Albert et al. and by MacKay and Petkau were much more complex and elaborate than those fitted in the examples in this package. See the references for details.

Source

The data were originally studied by Albert et al., (1994). They are were also analyzed by Altman and Petkau (2005). The data were kindly provided by Prof. Altman.

References

Albert, P. S., McFarland, H. F., Smith, M. E., and Frank, J. A. Time series for modelling counts from a relapsing-remitting disease: application to modelling disease activity in multiple sclerosis. *Statistics in Medicine* **13** (1994) 453–466.

Altman, Rachel MacKay, and Petkau, A. John. Application of hidden Markov models to multiple sclerosis lesion count data. *Statistics in Medicine* **24** (2005) 2335–2344.

logLikHmm

Log likelihood of a hidden Markov model

Description

Calculate the log likelihood of a hidden Markov model with discrete non-parametric observation distributions.

Usage

```
logLikHmm(y, par)
```

Arguments

<code>y</code>	A sequence, or list of sequences, of observations from a hidden Markov model with discrete non-parametric observation distributions.
<code>par</code>	A list with entries <ul style="list-style-type: none"> • <code>Rho</code>: The matrix whose columns consist of the probabilities specifying the distributions of the observations. • <code>tpm</code>: The transition probability matrix of the Markov chain. • (Optionally) <code>ispd</code>: The vector of probabilities specifying the initial state probability distribution, or a matrix each of whose columns are trivial (“delta function”) vectors specifying the “most probable” initial state for each observation sequence. If the component <code>ispd</code> of <code>par</code> is not present then <code>ispd</code> is calculated as the stationary distribution determined by <code>tpm</code>.

This list may be an object returned by `hmm()`.

Details

The observation values (the entries of `y`, or of the sequences which constitute the entries of `y` if `y` is a list) must be members of the set of row names of `Rho`. More specifically, if `Rho` has no row names then these observation values must all be integers between 1 and `nrow(Rho)`. If `Rho` *has* row names then these observations must be character strings and must be members of `row.names(Rho)`.

Value

The loglikelihood of `y` given the parameter values specified in `par`.

Author(s)

Rolf Turner <r.turner@auckland.ac.nz>

References

See [hmm\(\)](#) for references.

See Also

[hmm\(\)](#), [pr\(\)](#), [sp\(\)](#)

Examples

```
P <- matrix(c(0.7,0.3,0.1,0.9),2,2,byrow=TRUE)
R <- matrix(c(0.5,0,0.1,0.1,0.3,
             0.1,0.1,0,0.3,0.5),5,2)
set.seed(42)
l11 <- sample(250:350,20,TRUE)
y.num <- sim.hmm(l11,P,R)
y.let <- sim.hmm(l11,P,R,yval=letters[1:5])
l11 <- logLikHmm(y.num,list(tpm=P,Rho=R))
row.names(R) <- letters[1:5]
l12 <- logLikHmm(y.let,list(tpm=P,Rho=R))
# Note: l12 differs from l11 since y.num and y.let were
# simulated from different random seeds.
l13 <- logLikHmm(y.let,list(tpm=P,Rho=R,ispd=c(0.5,0.5)))
fit <- hmm(y.num,K=2)
l14 <- logLikHmm(y.num,fit) # Use the fitted rather than the "true" parameters.
```

mps

Most probable states.

Description

Calculates the most probable hidden state underlying each observation.

Usage

```
mps(y, object = NULL, tpm, Rho, ispd=NULL)
```

Arguments

y The observations for which the underlying most probable hidden states are required. May be a sequence of observations, or a list each component of which constitutes a (replicate) sequence of observations. If **y** is missing it is set equal to the **y** component of **object**, given that that **object** and that component exist. Otherwise an error is given.

object	An object describing a fitted hidden Markov model, as returned by <code>hmm()</code> . In order to make any kind of sense, object should bear some reasonable relationship to <code>y</code> .
tpm	The transition probability matrix for a hidden Markov model; ignored if object is non-null. Should bear some reasonable relationship to <code>y</code> .
Rho	A matrix specifying the probability distributions of the observations for a hidden Markov model; ignored if object is non-null. Should bear some reasonable relationship to <code>y</code> .
ispd	A vector specifying the initial state probability distribution for a hidden Markov model, or a matrix each of whose columns are trivial (“delta function”) vectors specifying the “most probable” initial state for each observation sequence. This argument is ignored if object is non-null. It should bear some reasonable relationship to <code>y</code> . If both <code>ispd</code> and <code>object</code> are NULL then <code>ispd</code> is taken to be the stationary distribution of the chain, calculated from <code>tpm</code> .

Details

For each t the maximum value of $\gamma_t(i)$, i.e. of the (estimated) probability that the state at time t is equal to i , is calculated, and the corresponding index returned. These indices are interpreted as the values of the (most probable) states. I.e. the states are assumed to be $1, 2, \dots, K$, for some K .

Value

If `y` is a single observation sequence, then the value is a vector of corresponding most probable states.

If `y` is a list of replicate sequences, then the value is a list, the j -th entry of which constitutes the vector of most probable states underlying the j -th replicate sequence.

Warning

The *sequence of most probable states* as calculated by this function will not in general be the *most probable sequence of states*. It may not even be a *possible* sequence of states. This function looks at the state probabilities separately for each time t , and not at the states in their sequential context.

To obtain the most probable sequence of states use `viterbi()`.

Author(s)

Rolf Turner <r.turner@auckland.ac.nz>

References

Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE vol. 77, pp. 257 – 286, 1989.

See Also

`hmm()`, `sim.hmm()`, `viterbi()`

Examples

```
# See the help for sim.hmm() for how to generate y.num.
## Not run:
fit.num <- hmm(y.num,K=2,verb=TRUE)
s.1 <- mps(y.num,fit.num)
s.2 <- mps(y.num,tpm=P,ispd=c(0.25,0.75),Rho=R) # P and R as in the help
                                           # for sim.hmm().

# The order of the states has gotten swapped; 3-s.1[,1] is much
# more similar to s.2[,1] than is s.1[,1].

## End(Not run)
```

pr

Probability of state sequences.

Description

Calculates the conditional probability of one or more state sequences, given the corresponding observations sequences (and the model parameters).

Usage

```
pr(s,y,object=NULL,tpm,Rho,ispd=NULL)
```

Arguments

s	A sequence of states of the underlying Markov chain, or a list of such sequences.
y	A sequence of observations from a hidden Markov model, corresponding to the state sequence s, or a list of such sequences corresponding to the state sequences in the list s. If y is or consists of a single sequence, it is assumed to correspond to each of the state sequences in s in turn. Otherwise the length of the list y must be the same as the length of the list s (otherwise an error is given). If y is missing, it is extracted from object provided that object and its y component are not NULL. Otherwise an error is given.
object	An object of class <code>hmm.discnp</code> as returned by <code>hmm()</code> .
tpm	The transition probability matrix of the chain. Ignored (and extracted from object instead) if object is not NULL.
Rho	The matrix of probabilities specifying the distribution of the observations, given the underlying state. The rows of this matrix correspond to the possible values of the observations, the columns to the states. Ignored (and extracted from object instead) if object is not NULL.
ispd	The vector specifying the initial state probability distribution of the Markov chain. Ignored (and extracted from object instead) if object is not NULL. If both ispd and object are NULL then ispd is taken to be the stationary distribution of the chain, calculated from tpm.

Value

The probability of s given y , or a vector of such probabilities if s and y are lists.

Warning

The conditional probabilities will be tiny if the sequences involved are of any substantial length. Underflow may be a problem. The implementation of the calculations is not sophisticated.

Author(s)

Rolf Turner <r.turner@auckland.ac.nz>

See Also

[hmm\(\)](#), [mps\(\)](#), [viterbi\(\)](#), [sp\(\)](#), [fitted.hmm.discnp\(\)](#)

Examples

```
# See the help for sim.hmm() for how to generate y.num.
## Not run:
fit.num <- hmm(y.num,K=2,verb=TRUE)
# Using fitted parameters.
s.vit.1 <- viterbi(y.num,fit.num)
pr.vit.1 <- pr(s.vit.1,object=fit.num)
# Using true parameters from which y.num was generated.
s.vit.2 <- viterbi(y.num,tpm=P,Rho=R)
pr.vit.2 <- pr(s.vit.2,y.num,tpm=P,Rho=R)

## End(Not run)
```

sim.hmm

Simulate discrete data from a hidden Markov model.

Description

Simulates one or more replicates of discrete data from a model such as is fitted by the function `hmm()`.

Usage

```
sim.hmm(nsim,tpm,Rho,ispd=NULL,yval=NULL,verb=FALSE)
```

Arguments

<code>nsim</code>	Vector of the lengths of the sequences of observations to be generated.
<code>tpm</code>	The transition probability matrix for the underlying hidden Markov chain(s). Note that the rows of <code>tpm</code> must sum to 1. Ignored if <code>ncol(Rho)==1</code> .
<code>Rho</code>	A matrix specifying the probability of an observation taking on one of a set of possible values, given the state of the underlying hidden Markov chain. Note that the columns of <code>Rho</code> must sum to 1. If <code>ncol(Rho)==1</code> the simulated data are i.i.d. from the distribution specified by the single column of <code>Rho</code> .
<code>ispd</code>	A vector specifying the initial state probability distribution of the chain. If this is not specified it is taken to be the stationary distribution of the chain, calculated from <code>tpm</code> .
<code>yval</code>	Vector of numbers or character strings constituting the possible values of the observations. If not supplied it is taken to equal <code>1:nrow(Rho)</code> .
<code>verb</code>	Logical scalar. If TRUE then the overall index of the simulated value that has been reached is printed out every 1000 iterations. Useful for reassurance when very “large” simulations are undertaken.

Value

If `length(nsim)==1` then the value returned is a vector of length `nsim`. If `length(nsim)>1` then the value returned is a list of the same length as `nsim`, each component of which is an independent vector of simulated observations. The length of component `i` of this list is equal to `nsim[i]`. The values of the observations are entries of `yval`.

Author(s)

Rolf Turner <r.turner@auckland.ac.nz>

See Also

[hmm\(\)](#)

Examples

```
P <- matrix(c(0.7,0.3,0.1,0.9),2,2,byrow=TRUE)
R <- matrix(c(0.5,0,0.1,0.1,0.3,
             0.1,0.1,0,0.3,0.5),5,2)

set.seed(42)
l11 <- sample(250:350,20,TRUE)
y.num <- sim.hmm(l11,P,R)
y.let <- sim.hmm(l11,P,R,yval=letters[1:5])
```

sp *Calculate the conditional state probabilities.*

Description

Returns the probabilities that the underlying hidden state is equal to each of the possible state values, at each time point, given the observation sequence. Also can return the fitted conditional means, if requested, given that the observations are numeric.

Usage

```
sp(y, object = NULL, tpm, Rho, ispd=NULL, means = FALSE)
```

Arguments

y	The observations on the basis of which the probabilities of the underlying hidden states are to be calculated. May be a sequence of observations, or a list each component of which constitutes a (replicate) sequence of observations. If y is missing it is set equal to the y component of object, given that that object and that component exist. Otherwise an error is given.
object	An object of class <code>hmm.discnp</code> as returned by <code>hmm()</code> .
tpm	The transition probability matrix for the underlying hidden Markov chain. Ignored if object is not NULL. Ignored if object is not NULL (in which case tpm is extracted from object).
Rho	The matrix of probabilities specifying the distribution of the observations, given the underlying state. The rows of this matrix correspond to the possible values of the observations, the columns to the states. Ignored if object is not NULL (in which case Rho is extracted from object).
ispd	Vector specifying the initial state probability distribution of the underlying hidden Markov chain. Ignored if object is not NULL (in which case ispd is extracted from object). If both object and ispd are NULL then ispd is calculated to be the stationary distribution of the chain as determined by tpm.
means	A logical scalar; if means is TRUE then the conditional expected value of the observations (given the observation sequence) is calculated at each time point. If means is TRUE and the observation values are not numeric, then an error is given.

Details

Then conditional mean value at time t is calculated as

$$\sum_k \gamma_t(k) \mu_k$$

where $\gamma_t(k)$ is the conditional probability (given the observations) that the hidden Markov chain is in state k at time t , and μ_k is the expected value of an observation given that the chain is in state k .

Arguments

y	The observations for which the underlying most probable hidden states are required. May be a sequence of observations, or a list each entry of which constitutes an independent sequence of observations. If y is missing (and if object is not NULL) then y is extracted from object, provided that the y component of object is present. Otherwise an error is given.
object	An object describing a hidden Markov model, as fitted to the data set y by <code>hmm()</code> .
tpm	The transition probability matrix for a hidden Markov model; ignored if object is non-null.
Rho	A matrix specifying the probability distributions of the observations for a hidden Markov model; ignored if object is non-null. Should bear some reasonable relationship to y. If Rho has row names then these must include all values of the observations. If Rho does not have row names then the values of the observations must be integers between 1 and <code>nrow(Rho)</code> .
ispd	The initial state probability distribution for a hidden Markov model; ignored if object is non-null. Should bear some reasonable relationship to y. If object and ispd are both NULL then ispd is set equal to the stationary distribution calculated from tpm.
log	Logical scalar. Should logarithms be used in the recursive calculations of the probabilities involved in the Viterbi algorithm, so as to avoid underflow? If log is FALSE then underflow is avoided instead by a normalization procedure. The quantity <code>delta</code> (see Rabiner 1989, page 264) is replaced by <code>delta/sum(delta)</code> at each step. It should actually make no difference whether log is set to TRUE. I just included the option because I could. Also the HMM package uses the logarithm approach so setting <code>log=TRUE</code> might be of interest if comparisons are to be made between the results of the two packages.

Details

Applies the Viterbi algorithm to calculate “the” most probable robable state sequence underlying each observation sequences.

Value

If y consists of a single observation sequence, the value is the underlying most probable observation sequence, or a matrix whose columns consist of such sequences if there is more than one (equally) most probable sequence.

If y consists of a list of observation sequences, the value is a list each entry of which is of the form described above.

Warning

There *may* be more than one equally most probable state sequence underlying a given observation sequence. This phenomenon appears to be unlikely to occur in practice.

Thanks

The correction made to the code so as to avoid underflow problems was made due to an inquiry and suggestion from Owen Marshall.

Author(s)

Rolf Turner <r.turner@auckland.ac.nz>

References

Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE vol. 77, pp. 257 – 286, 1989.

See Also

[hmm\(\)](#), [sim.hmm\(\)](#), [mps\(\)](#), [pr\(\)](#), [viterbi\(\)](#)

Examples

```
# See the help for sim.hmm() for how to generate y.num and y.let.
## Not run:
fit.num <- hmm(y.num,K=2,verb=TRUE)
v.1 <- viterbi(object=fit.num)
v.2 <- viterbi(y.num,tpm=P,Rho=R) # P and R as in the
                                # help for sim.hmm().
# The order of the states has gotten swapped; 3-v.1[[1]] is much
# more similar to v.2[[1]] than is v.1[[1]].

fit.let <- hmm(y.let,K=2,verb=TRUE)
v.3 <- viterbi(object=fit.let) # Works.
v.4 <- viterbi(y.let,tpm=P,Rho=R) # Throws an error (R has no row names.)

## End(Not run)
```

Index

*Topic **datasets**

colifCount, 2
lesionCount, 8

*Topic **models**

fitted.hmm.discnp, 3
hmm, 4
logLikHmm, 9
mps, 10
pr, 12
sim.hmm, 13
sp, 15
viterbi, 16

colifCount, 2

fitted.hmm.discnp, 3, 13, 16

hmm, 3, 4, 9–18

lesionCount, 8
logLikHmm, 9

mps, 7, 10, 13, 16, 18

pr, 10, 12, 16, 18

sim.hmm, 7, 11, 13, 18
sp, 3, 4, 10, 13, 15

viterbi, 7, 11, 13, 16, 16, 18