Package ‘HMMextra0s’

September 17, 2018

Type Package
Title Hidden Markov Models with Extra Zeros
Version 1.0.0
Imports mvtnorm, ellipse
Suggests HiddenMarkov
Depends methods
Date 2018-09-12
Author Ting Wang - I am grateful to Jiancang Zhuang for some helpful suggestions and contributions
Maintainer Ting Wang <ting.wang@otago.ac.nz>

Description Contains functions for hidden Markov models with observations having extra zeros as defined in the following two publications, Wang, T., Zhuang, J., Obara, K. and Tsuruoka, H. (2016) <doi:10.1111/rssc.12194>; Wang, T., Zhuang, J., Buckby, J., Obara, K. and Tsuruoka, H. (2018) <doi:10.1029/2017JB015360>. The observed response variable is either univariate or bivariate Gaussian conditioning on presence of events, and extra zeros mean that the response variable takes on the value zero if nothing is happening. Hence the response is modelled as a mixture distribution of a Bernoulli variable and a continuous variable. That is, if the Bernoulli variable takes on the value 1, then the response variable is Gaussian, and if the Bernoulli variable takes on the value 0, then the response is zero too. This package includes functions for simulation, parameter estimation, goodness-of-fit, the Viterbi algorithm, and plotting the classified 2-D data. Some of the functions in the package are based on those of the R package ‘HiddenMarkov’ by David Harte.

License GPL (>= 2)

URL http://www.stats.otago.ac.nz/?people=ting_wang

NeedsCompilation yes
Repository CRAN

Date/Publication 2018-09-17 15:30:07 UTC

R topics documented:

HMMextra0s-package ......................................................... 2
cumdist.hmm0norm .......................................................... 4
Description

The DESCRIPTION file:

Package:     HMMextra0s
Type:        Package
Title:       Hidden Markov Models with Extra Zeros
Version:     1.0.0
Imports:     mvtnorm, ellipse
Suggests:    HiddenMarkov
Depends:     methods
Date:        2018-09-12
Author:      Ting Wang - I am grateful to Jiancang Zhuang for some helpful suggestions and contributions
Maintainer:  Ting Wang <ting.wang@otago.ac.nz>
Description: Contains functions for hidden Markov models with observations having extra zeros as defined in the following ...
License:     GPL(>=2)
URL:         http://www.stats.otago.ac.nz/?people=ting_wang
Packaged:    2018-09-12 01:56:30 UTC; twang
NeedsCompilation: yes

Index of help topics:

HMMextra0s-package               Hidden Markov Models with Extra Zeros
Viterbi.hmm0norm                 Viterbi Path of a 1-D HMM with Extra Zeros
Viterbi.hmm0norm2d               Viterbi Path of a Bivariate HMM with Extra Zeros
cumdist.hmm0norm                 Cumulative distribution of an HMM with Extra Zeros
hmm0norm                        Parameter Estimation of an HMM with Extra Zeros
hmm0norm2d                      Parameter Estimation of a bivariate HMM with Extra Zeros
In this framework, we are interested in estimating the transition probability matrix \( \Gamma = (\gamma_{ij})_{m \times m} \) that gives the distribution feature of observations \( Z_t \) at time \( t \). The probability of a first-order Markov chain in state \( j \) at time \( t \) given the previous states is 
\[
P(S_t = j|S_{t-1}, \ldots, S_1) = P(S_t = j|S_{t-1}).
\]
These states are not observable. The observation \( Y_t \) at time \( t \) depends on the state \( S_t \) of the Markov chain.

In this framework, we are interested in estimating the transition probability matrix \( \Gamma = (\gamma_{ij})_{m \times m} \) of the Markov chain that describes the migration pattern and the density function \( f(y_t|S_t = i) \) that gives the distribution feature of observations in state \( i \), where \( \gamma_{ij} = P(S_t = j|S_{t-1} = i) \).

Let \( Z_t \) be a Bernoulli variable, with \( Z_t = 1 \) if an event is present at \( t \), and \( Z_t = 0 \), otherwise. Let \( X_t \) be the response variable (e.g., location of the tremor cluster in 2D space) at time \( t \). We set 
\[
P(Z_t = 0|S_t = i) = 1 - p_i 
\]
and 
\[
P(Z_t = 1|S_t = i) = p_i.
\]
We assume that, given \( Z_t = 1 \) and \( S_t = i \), \( X_t \) follows a univariate or bivariate normal distribution, e.g. for a bivariate normal,
\[
f(x_t|Z_t = 1, S_t = i) = \frac{1}{2\pi|\Sigma_i|^{1/2}} \exp \left( -\frac{1}{2}(x_t - \mu_i)^T \Sigma_i^{-1}(x_t - \mu_i) \right).
\]
The joint probability density function of \( Z_t \) and \( X_t \) conditional on the system being in state \( i \) at time \( t \) is
\[
f(x_t, z_t|S_t = i) = (1 - p_i)^{1-z_t} p_i \frac{1}{2\pi|\Sigma_i|^{1/2}} \exp \left( -\frac{1}{2}(x_t - \mu_i)^T \Sigma_i^{-1}(x_t - \mu_i) \right)^{z_t},
\]
where \( p_i, \mu_i = E(X_t|S_t = i, Z_t = 1) \) and \( \Sigma_i = Var(X_t|S_t = i, Z_t = 1) \) are parameters to be estimated.
Author(s)
Ting Wang - I am grateful to Jiancang Zhuang for some helpful suggestions and contributions
Maintainer: Ting Wang <ting.wang@otago.ac.nz>

References

cumdist.hmm0norm

Cumulative distribution of an HMM with Extra Zeros

Description
Calculates the cumulative distribution of an HMM with extra zeros.

Usage
cumdist.hmm0norm(x,HMMest)

Arguments
x
  x is a value at which the cumulative distribution is evaluated.
HMMest
  is a list which contains pie, gamma, sig, mu, and delta (the HMM parameter estimates).

Value
prob
  is the calculated cumulative distribution.

Author(s)
Ting Wang

References
**Examples**

```r
pie <- c(0.002, 0.2, 0.4)
gamma <- matrix(c(0.99, 0.007, 0.003,
                  0.02, 0.97, 0.01,
                  0.04, 0.01, 0.95), byrow=TRUE, nrow=3)
mu <- matrix(c(0.3, 0.7, 0.2), nrow=1)
sig <- matrix(c(0.2, 0.1, 0.1), nrow=1)
delta <- c(1, 0, 0)
y <- sim.hmm0norm(mu, sig, pie, gamma, delta, nsim=5000)
R <- as.matrix(y$x, ncol=1)
Z <- y$z
HMMEST <- hmm0norm(R, Z, pie, gamma, mu, sig, delta)
xx <- seq(0, 1, 0.05)
cumdist <- apply(t(xx), 2, cumdist.hmm0norm, HMMest=HMMEST)
```

---

**hmm0norm**  
*Parameter Estimation of an HMM with Extra Zeros*

**Description**

Calculates the parameter estimates of a 1-D HMM with observations having extra zeros.

**Usage**

```r
hmm0norm(R, Z, pie, gamma, mu, sig, delta, tol=1e-6, print.level=1, fortran = TRUE)
```

**Arguments**

- `R` is the observed data. R is a $T \times 1$ matrix, where $T$ is the number of observations.
- `Z` is the binary data with the value 1 indicating that an event was observed and 0 otherwise. Z is a vector of length $T$.
- `pie` is a vector of length $m$, the $j$th element of which is the probability of $Z = 1$ when the process is in state $j$.
- `gamma` is the transition probability matrix ($m \times m$) of the hidden Markov chain.
- `mu` is a $1 \times m$ matrix, the $j$th element of which is the mean of the (Gaussian) distribution of the observations in state $j$.
- `sig` is a $1 \times m$ matrix, the $j$th element of which is the standard deviation of the (Gaussian) distribution of the observations in state $j$.
- `delta` is a vector of length $m$, the initial distribution vector of the Markov chain.
- `tol` is the tolerance for testing convergence of the iterative estimation process. The default tolerance is 1e-6. For initial test of model fit to your data, a larger tolerance (e.g., 1e-3) should be used to save time.
print.level controls the amount of output being printed. Default is 1. If print.level=1, only the log likelihoods and the differences between the log likelihoods at each step of the iterative estimation process, and the final estimates are printed. If print.level=2, the log likelihoods, the differences between the log likelihoods, and the estimates at each step of the iterative estimation process are printed.

fortran is logical, and determines whether Fortran code is used; default is TRUE.

Value

pie is the estimated probability of $Z = 1$ when the process is in each state.

mu is the estimated mean of the (Gaussian) distribution of the observations in each state.

sig is the estimated standard deviation of the (Gaussian) distribution of the observations in each state.

gamma is the estimated transition probability matrix of the hidden Markov chain.

delta is the estimated initial distribution vector of the Markov chain.

LL is the log likelihood.

Author(s)

Ting Wang

References


Examples

```r
pie <- c(0.002,0.2,0.4)
gamma <- matrix(c(0.99,0.007,0.003, 0.02,0.97,0.01, 0.04,0.01,0.95),byrow=TRUE, nrow=3)
mu <- matrix(c(0.3,0.7,0.2),nrow=1)
sig <- matrix(c(0.2,0.1,0.1),nrow=1)
delta <- c(1,0,0)
y <- sim.hmm0norm(mu,sig,pie,gamma,delta, nsim=5000)
R <- as.matrix(y$x,ncol=1)
Z <- y$z
yn <- hmm0norm(R, Z, pie, gamma, mu, sig, delta)
yn
```
Parameter Estimation of a bivariate HMM with Extra Zeros

**Description**

Calculates the parameter estimates of an HMM with bivariate observations having extra zeros.

**Usage**

\[
\text{hmmPnorm2d}(R, Z, \text{pie}, \text{gamma}, \text{mu}, \text{sig}, \text{delta}, \text{tol}=1e-6, \text{print.level}=1, \text{fortran} = \text{TRUE})
\]

**Arguments**

- **R** is the observed data. \( R \) is a \( T \times 2 \) matrix, where \( T \) is the number of observations.
- **Z** is the binary data with the value 1 indicating that an event was observed and 0 otherwise. \( Z \) is a vector of length \( T \).
- **pie** is a vector of length \( m \), the \( j \)th element of which is the probability of \( Z = 1 \) when the process is in state \( j \).
- **gamma** is the transition probability matrix \((m \times m)\) of the hidden Markov chain.
- **mu** is an \( m \times 2 \) matrix, the \( j \)th row of which is the mean of the bivariate (Gaussian) distribution of the observations in state \( j \).
- **sig** is a \( 2 \times 2 \times m \) array. The matrix \( \text{sig}[, , j] \) is the variance-covariance matrix of the bivariate (Gaussian) distribution of the observations in state \( j \).
- **delta** is a vector of length \( m \), the initial distribution vector of the Markov chain.
- **tol** is the tolerance for testing convergence of the iterative estimation process. Default is \( 1e-6 \). For initial test of model fit to your data, a larger tolerance (e.g., \( 1e-3 \)) should be used to save time.
- **print.level** controls the amount of output being printed. Default is 1. If \( \text{print.level}=1 \), only the log likelihoods and the differences between the log likelihoods at each step of the iterative estimation process, and the final estimates are printed. If \( \text{print.level}=2 \), the log likelihoods, the differences between the log likelihoods, and the estimates at each step of the iterative estimation process are printed.
- **fortran** is logical, and determines whether Fortran code is used; default is \text{TRUE}.

**Details**

Setting up initial values for the real world data can be challenging, especially when the model is large (the number of states is big). In the example below, we include a simple way to set up initial values. If the model is large, the model fitting process should be repeated for many different initial values. In the example below, we set the number of initial values to be \( N = 2 \) for the ease of compilation. For real-world data analysis, taking the 2D model for the tremor data in Wang et al. (2018) for example, we used at least \( N = 1000 \) initial values for the large models with more than 15 hidden states.
**Value**

- **pie** is the estimated probability of $Z = 1$ when the process is in each state.
- **mu** is the estimated mean of the bivariate (Gaussian) distribution of the observations in each state.
- **sig** is the estimated variance-covariance matrix of the bivariate (Gaussian) distribution of the observations in each state.
- **gamma** is the estimated transition probability matrix of the hidden Markov chain.
- **delta** is the estimated initial distribution vector of the Markov chain.
- **LL** is the log likelihood.

**Author(s)**

Ting Wang

**References**


**Examples**

```r
pie <- c(0.002, 0.2, 0.4)
gamma <- matrix(c(0.99, 0.007, 0.003,
                  0.02, 0.97, 0.01,
                  0.04, 0.01, 0.95), byrow=TRUE, nrow=3)
mu <- matrix(c(35.03, 137.01,
               35.01, 137.29,
               35.15, 137.39), byrow=TRUE, nrow=3)
sig <- array(NA, dim=c(2,2,3))
sig[,,1] <- matrix(c(0.005, -0.001,
                    -0.001, 0.01), byrow=TRUE, nrow=2)
sig[,,2] <- matrix(c(0.0007, -0.0002,
                    -0.0002, 0.0006), byrow=TRUE, nrow=2)
sig[,,3] <- matrix(c(0.002, 0.0018,
                    0.0018, 0.003), byrow=TRUE, nrow=2)
delta <- c(1,0,0)
y <- sim.hmm0norm2d(mu, sig, pie, gamma, delta, nsim=5000)
R <- y$x
Z <- y$z
yn <- hmm0norm2d(R, Z, pie, gamma, mu, sig, delta)
yn
# Setting up initial values when analysing real-world data
## nk is the number of states for the fitted model
### In this example we use nk=3

LL <- -10^200 ## A very small value to compare with
## the log likelihood from the model
```
nk <- 3

gamma <- array(NA, dim=c(nk, nk))
u <- array(NA, dim=c(nk, 2))
sig <- array(NA, dim=c(2, 2, nk))
pie <- array(NA, dim=c(1, nk))

kk <- 1
N <- 2
while(kk < N)
{
  temp <- matrix(rnorm(nk * nk, 0, 1), ncol=nk)
  diag(temp) = diag(temp) + rpois(1, 6) * apply(temp, 1, sum)
  temp <- temp * matrix(rep(1, ncol(temp)), ncol=ncol(temp), byrow=FALSE)
  gamma <- temp

  R1min <- min((temp)[1])
  R1max <- max((temp)[1])
  R2min <- min((temp)[2])
  R2max <- max((temp)[2])
  temp <- cbind(rnorm(nk, R1min, R1max), rnorm(nk, R2min, R2max))
  temp <- temp[order(temp[, 2]),]
  mu <- temp

  sdR1 <- sd((temp)[1])
  sdR2 <- sd((temp)[2])
  for (j in 1:nk){
    temp <- matrix(rnorm(4, 0, 0.001, max(sdR1, sdR2)), ncol=2)
    temp[1] <- temp[2] <- runif(1, -1, 1) * sqrt(prod(diag(temp)))
    sig[, , j] <- temp
  }

  pie <- matrix(sort(c(runif(1, 0, 0.01), runif(nk-1, 0, 1))), nrow = 1, byrow = TRUE)

  delta <- c(6, runif(nk-1, 0, 1))
  delta <- delta/sum(delta)

  tryCatch(
    temp <- hmm0norm2d(R, Z, pie, gamma, mu, sig, delta)
    kk <- kk+1
    if( LL <= temp$LL)
      {HMMest <- temp
      LL = HMMest$LL
      eval(parse(text=paste('HMM', kk, 'rest = HMMest', sep='')))
      eval(parse(text=paste('save(HMM', kk, 'est, file="HMM', kk, 'est.image"', sep=''))))
    }
  }, error=function(e){})
print(kk)
plotVitloc2d  

Plot the Classified 2-D Data of a Bivariate HMM With Extra Zeros

Description

Plot the classified 2-D data with different colours representing different hidden states (or different clusters) obtained from the Viterbi path and confidence contours.

Usage

plotVitloc2d(object, R, Z, HMMest, CI.level=0.95, npoints=100, cols=NA, cex.lab=1.5, cex.axis=1.5, cex=1, cex.text=2)

Arguments

object is a list containing y (the estimated Viterbi path) and v (the estimated probability of each time point being in each state). This object is returned from running Viterbi.hmm@norm2d(R, Z, HMMest).

R is the observed data. R is a T * 2 matrix, where T is the number of observations.

Z is the binary data with the value 1 indicating that an event was observed and 0 otherwise. Z is a vector of length T.

HMMest is a list which contains pie, gamma, sig, mu, and delta (the bivariate HMM parameter estimates).

CI.level is a scalar or a vector, the confidence level for the ellipse contour of each state. Default is 0.95.

npoints is the number of points used in the ellipse. Default is 100.

cols is a vector defines the colors to be used for different states. If col=NA, then the default colors will be used.

cex.lab specifies the size of the axis label text.

cex.axis specifies the size of the tick label numbers/text.

cex specifies the size of the points.

cex.text specifies the size of the text indicting the state number.

Author(s)

Ting Wang and Jiancang Zhuang

References

Examples

```r
pie <- c(0.008, 0.2, 0.4)
gamma <- matrix(c(0.99, 0.007, 0.003, 0.02, 0.97, 0.01, 0.04, 0.01, 0.95), byrow=TRUE, nrow=3)
mu <- matrix(c(35.03, 137.01, 35.01, 137.29, 35.15, 137.39), byrow=TRUE, nrow=3)
sig <- array(NA, dim=c(2, 2, 3))
sig[,,1] <- matrix(c(0.005, -0.001, -0.001, 0.01), byrow=TRUE, nrow=2)
sig[,,2] <- matrix(c(0.0007, -0.0002, -0.0002, 0.0006), byrow=TRUE, nrow=2)
sig[,,3] <- matrix(c(0.002, 0.0018, 0.0018, 0.003), byrow=TRUE, nrow=2)
delta <- c(1, 0, 0)
y <- sim.hmm0norm2d(mu, sig, pie, gamma, delta, nsim=5000)
R <- y$x
Z <- y$z
HMMEST <- hmm0norm2d(R, Z, pie, gamma, mu, sig, delta)
Viterbi3 <- Viterbi.hmm0norm2d(R, Z, HMMEST)
plotVitloc2d(Viterbi3, R, Z, HMMEST)
```

Description

Plot the 2-D data, Viterbi path and the probability of each time point being in each state over time.

Usage

```r
plotVitpath2d(object, R, Z, HMMest, len.dat=96432, varb=8780, yearstart=2005, yearend=2012, cols=NA, cex.lab=1.5, cex.axis=1.5)
```

Arguments

- **object** is a list containing `y` (the estimated Viterbi path) and `v` (the estimated probability of each time point being in each state). This object is returned from running `Viterbi.hmm0norm2d(R, Z, HMMest).
- **R** is the observed data. `R` is a $T \times 2$ matrix, where $T$ is the number of observations.
- **Z** is the binary data with the value 1 indicating that an event was observed and 0 otherwise. `Z` is a vector of length $T$.
- **HMMest** is a list which contains `pie`, `gamma`, `sig`, `mu`, and `delta` (the bivariate HMM parameter estimates).
- **len.dat** is the length of the data, that is, the number of time points. Default is 96432.
varb is an integer indicating the length of data that will be plotted on each page. The default is 8780.

yearstart is the starting year of the data used. Default is 2005.

yearend is the end year of the data used. Default is 2012.

cols is a vector defines the colors to be used for different states. If col=NA, then the default colors will be used.

cex.lab specifies the size of the axis label text.

cex.axis specifies the size of the tick label numbers/text.

Details

The returned object has four panels. Top two panels: Observed latitudes and longitudes with the center \( \hat{\mu}_i \) of each state overlaid as the red lines; third panel: tracked most likely state sequence of the HMM; bottom panel: the estimated probability of the data being in each state, with blank representing the probability of being in the last state (typically the plot looks better if the last state represents the background state with the minimum proportion of tremor occurrence). Some example plots are in the supplementary file of the reference Wang et al. (2018).

Author(s)

Ting Wang and Jiancang Zhuang

References


Examples

```r
pie <- c(0.008, 0.2, 0.4)
gamma <- matrix(c(0.99, 0.007, 0.003, 
                 0.02, 0.97, 0.01, 
                 0.04, 0.01, 0.95), byrow=TRUE, nrow=3)
mu <- matrix(c(35.03, 137.01, 
              35.01, 137.29, 
              35.15, 137.39), byrow=TRUE, nrow=3)
sig <- array(NA, dim=c(2,2,3))
sig[,,1] <- matrix(c(0.005, -0.001, 
                   -0.001, 0.01), byrow=TRUE, nrow=2)
sig[,,2] <- matrix(c(0.0007, -0.0002, 
                   -0.0002, 0.0006), byrow=TRUE, nrow=2)
sig[,,3] <- matrix(c(0.002, 0.0018, 
                   0.0018, 0.003), byrow=TRUE, nrow=2)
delta <- c(1,0,0)
y <- sim.hmmnorm2d(mu, sig, pie, gamma, delta, nsim=5000)
R <- y$x
Z <- y$z
HMMEST <- hmmnorm2d(R, Z, pie, gamma, mu, sig, delta)
Viterbi3 <- Viterbi.hmmnorm2d(R, Z, HMMEST)
```
**Description**

Simulates the observed process and the associated binary variable of a 1-D HMM with extra zeros.

**Usage**

```
sim.hmm0norm(mu, sig, pie, gamma, delta, nsim = 1, seed = NULL)
```

**Arguments**

- `pie` is a vector of length $m$, the $j$th element of which is the probability of $Z = 1$ when the process is in state $j$.
- `gamma` is the transition probability matrix ($m \times m$) of the hidden Markov chain.
- `mu` is a $1 \times m$ matrix, the $j$th element of which is the mean of the (Gaussian) distribution of the observations in state $j$.
- `sig` is a $1 \times m$ matrix, the $j$th element of which is the standard deviation of the (Gaussian) distribution of the observations in state $j$.
- `delta` is a vector of length $m$, the initial distribution vector of the Markov chain.
- `nsim` is an integer, the number of observations to simulate.
- `seed` is the seed for simulation. Default `seed=NULL`.

**Value**

- `x` is the simulated observed process.
- `z` is the simulated binary data with the value 1 indicating that an event was observed and 0 otherwise.
- `mcy` is the simulated hidden Markov chain.

**Author(s)**

Ting Wang

**References**

Examples

```r
pie <- c(0.002, 0.2, 0.4)
gamma <- matrix(c(0.99, 0.007, 0.003,
                 0.02, 0.97, 0.01,
                 0.04, 0.01, 0.95), byrow=TRUE, nrow=3)
mu <- matrix(c(0.3, 0.7, 0.2), nrow=1)
sig <- matrix(c(0.2, 0.1, 0.1), nrow=1)
delta <- c(1, 0, 0)
y <- sim.hmm0norm(mu, sig, pie, gamma, delta, nsim=5000)
```

**Description**

Simulates the observed process and the associated binary variable of a bivariate HMM with extra zeros.

**Usage**

```r
sim.hmm0norm2d(mu, sig, pie, gamma, delta, nsim = 1, seed = NULL)
```

**Arguments**

- `pie` is a vector of length `m`, the `j`th element of which is the probability of `Z = 1` when the process is in state `j`.
- `gamma` is the transition probability matrix (`m * m`) of the hidden Markov chain.
- `mu` is an `m * 2` matrix, the `j`th row of which is the mean of the bivariate (Gaussian) distribution of the observations in state `j`.
- `sig` is a `2 * 2 * m` array. The matrix `sig[,]` is the variance-covariance matrix of the bivariate (Gaussian) distribution of the observations in state `j`.
- `delta` is a vector of length `m`, the initial distribution vector of the Markov chain.
- `nsim` is an integer, the number of observations to simulate.
- `seed` is the seed for simulation. Default seed=NULL.

**Value**

- `x` is the simulated observed process.
- `z` is the simulated binary data with the value 1 indicating that an event was observed and 0 otherwise.
- `mcy` is the simulated hidden Markov chain.

**Author(s)**

Ting Wang
References


Examples

```r
pie <- c(0.002, 0.2, 0.4)
gamma <- matrix(c(0.99, 0.007, 0.003,
                  0.02, 0.97, 0.01,
                  0.04, 0.01, 0.05), byrow=TRUE, nrow=3)
mu <- matrix(c(35.03, 137.01,
             35.01, 137.29,
             35.15, 137.39), byrow=TRUE, nrow=3)
sig <- array(NA, dim=c(2, 2, 3))
sig[,1] <- matrix(c(0.005, -0.001,
                  -0.001, 0.01), byrow=TRUE, nrow=2)
sig[,2] <- matrix(c(0.0007, -0.0002,
                  -0.0002, 0.0006), byrow=TRUE, nrow=2)
sig[,3] <- matrix(c(0.002, 0.0018,
                  0.0018, 0.003), byrow=TRUE, nrow=2)
derta <- c(1, 0, 0)
y <- sim.hmm0norm2d(mu, sig, pie, gamma, delta, nsim=5000)
```

Viterbi.hmm0norm

Viterbi Path of a 1-D HMM with Extra Zeros

Description

Finds the most probable sequence of hidden states of an observed process.

Usage

Viterbi.hmm0norm(R, Z, HMMest)

Arguments

- `R` is the observed data. `R` is a `T` * `1` matrix, where `T` is the number of observations.
- `Z` is the binary data with the value 1 indicating that an event was observed and 0 otherwise. `Z` is a vector of length `T`.
- `HMMest` is a list which contains `pie`, `gamma`, `sig`, `mu`, and `delta` (the HMM parameter estimates).

Value

- `y` is the estimated Viterbi path.
- `v` is the estimated probability of each time point being in each state.
Author(s)
Ting Wang

References

Examples
```
pie <- c(0.002, 0.2, 0.4)
gamma <- matrix(c(0.99, 0.007, 0.003, 
                  0.02, 0.97, 0.01, 
                  0.04, 0.01, 0.95), byrow=TRUE, nrow=3)
mu <- matrix(c(0.3, 0.7, 0.2), nrow=1)
sig <- matrix(c(0.2, 0.1, 0.1), nrow=1)
delta <- c(1, 0, 0)
y <- sim.hmm@norm(mu, sig, pie, gamma, delta, nsim=5000)
R <- as.matrix(y$x, ncol=1)
Z <- y$z
HMMest <- hmm@norm(R, Z, pie, gamma, mu, sig, delta)
Viterbi3 <- Viterbi.hmm@norm(R, Z, HMMest)
```

Viterbi@norm2d Viterbi Path of a Bivariate HMM with Extra Zeros

Description
Finds the most probable sequence of hidden states of an observed process of a bivariate HMM with extra zeros.

Usage
```
Viterbi.hmm@norm2d(R, Z, HMMest)
```

Arguments
- **R**
  - is the observed data. R is a $T \times 2$ matrix, where $T$ is the number of observations.
- **Z**
  - is the binary data with the value 1 indicating that an event was observed and 0 otherwise. Z is a vector of length $T$.
- **HMMest**
  - is a list which contains pie, gamma, sig, mu, and delta (the bivariate HMM parameter estimates).

Value
- **y**
  - is the estimated Viterbi path.
- **v**
  - is the estimated probability of each time point being in each state.
Author(s)
Ting Wang

References

Examples

```
pie <- c(0.002,0.2,0.4)
gamma <- matrix(c(0.99,0.007,0.003,
                 0.02,0.97,0.01,
                 0.04,0.01,0.95),byrow=TRUE,nrow=3)
mu <- matrix(c(35.03,137.01,
              35.01,137.29,
              35.15,137.39),byrow=TRUE,nrow=3)
sig <- array(NA,dim=c(2,2,3))
sig[,]1 <- matrix(c(0.005,-0.001,
                 -0.001,0.01),byrow=TRUE,nrow=2)
sig[,]2 <- matrix(c(0.0007,-0.0002,
                 -0.0002,0.0006),byrow=TRUE,nrow=2)
sig[,]3 <- matrix(c(0.002,0.0018,
                 0.0018,0.003),byrow=TRUE,nrow=2)
delta <- c(1,0,0)
y <- sim.hmm0norm2d(mu,sig,pie, gamma, delta, nsim=5000)
R <- y$x
Z <- y$z
HMMEST <- hmm0norm2d(R, Z, pie, gamma, mu, sig, delta)
Viterbi3 <- Viterbi.hmm0norm2d(R,Z,HMMEST)
```
Index

*Topic **goodness-of-fit**
  cumdist.hmm0norm, 4

*Topic **inversion**
  Viterbi.hmm0norm, 15
  Viterbi.hmm0norm2d, 16

*Topic **methods**
  hmm0norm, 5
  hmm0norm2d, 7

*Topic **optimize**
  hmm0norm, 5
  hmm0norm2d, 7

*Topic **package**
  HMMextra0s-package, 2

*Topic **plot, 2-D classification, Viterbi**
  plotVitloc2d, 10

*Topic **plot, classification, Viterbi**
  plotVitpath2d, 11

*Topic **simulation**
  sim.hmm0norm, 13
  sim.hmm0norm2d, 14

  cumdist.hmm0norm, 3, 4
  hmm0norm, 3, 5
  hmm0norm2d, 3, 7
  HMMextra0s (HMMextra0s-package), 2
  HMMextra0s-package, 2
  plotVitloc2d, 3, 10
  plotVitpath2d, 11
  sim.hmm0norm, 3, 13
  sim.hmm0norm2d, 3, 14
  Viterbi.hmm0norm, 3, 15
  Viterbi.hmm0norm2d, 3, 16