Package ‘rarhsmm’

March 20, 2018

Type Package

Title Regularized Autoregressive Hidden Semi Markov Model

Version 1.0.7

Date 2018-03-19

Author Zekun (Jack) Xu, Ye Liu

Maintainer Zekun Xu <zekunxu@gmail.com>

Description Fit Gaussian hidden Markov (or semi-Markov) models with / without autoregressive coefficients and with / without regularization. The fitting algorithm for the hidden Markov model is illustrated by Rabiner (1989) <doi:10.1109/5.18626>. The shrinkage estimation on the covariance matrices is based on the method by Ledoit et al. (2004) <doi:10.1016/S0047-259X(03)00096-4>. The shrinkage estimation on the autoregressive coefficients uses the elastic net shrinkage detailed in Zou et al. (2005) <doi:10.1111/j.1467-9868.2005.00503.x>.

Depends R(>= 3.0.0)

License GPL

LazyData TRUE

Imports Rcpp (>= 0.12.9), glmnet

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-03-20 14:53:27 UTC

R topics documented:

`em.hmm` *EM algorithm to compute maximum likelihood estimate of Gaussian hidden Markov models with / without autoregressive structures and with / without regularization on the covariance matrices and/or autoregressive structures.*

Description

EM algorithm to compute maximum likelihood estimate of Gaussian hidden Markov models with / without autoregressive structures and with / without regularization on the covariance matrices and/or autoregressive structures.

Usage

```
em.hmm(y, mod, ntimes = NULL, tol = 1e-04, maxit = 100, arp = 0,
  cov.shrink = 0, auto.lambda = 0, auto.alpha = 0, print = TRUE)
```

Arguments

`y`	observed series
`mod`	list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices
`ntimes`	length of each homogeneous time series. Default to NULL, which means only homogeneous time series.
`tol`	tolerance for relative change. Default to 1e-4.
`maxit`	maximum number of iterations. Default to 100.
`arp`	order of autoregressive. Default to 0.
`cov.shrink`	shrinkage on the multivariate normal covariance matrix. Default to 0. See references.

`auto.lambda`	elastic net shrinkage on the autoregressive coefficients. Default to 0. See references.
`auto.alpha`	The elasticnet mixing parameter, with 0<=alpha<=1. The penalty is defined as (1-alpha)/2||.||_2^2+alpha||.||_1. alpha=1 is the lasso penalty, and alpha=0 the ridge penalty. Default to 0. Same as in the glmnet package.
`print`	Default to TRUE.

Value

a list containing the fitted parameters.

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67.2 (2005): 301-320.

Ledoit, Olivier, and Michael Wolf. "A well-conditioned estimator for large-dimensional covariance matrices." Journal of multivariate analysis 88.2 (2004): 365-411.

Examples

```
set.seed(332213)
data(finance)
x <- data.matrix(finance)
#log return
y <- x[-1,-51]
for(i in 2:nrow(x)){
 y[i-1,] <- log(x[i,-51]) - log(x[i-1,-51])
}
#annualize the log return
y <- y * 252

#first, fit a Gaussian HMM without autoregressive structure
m <- 2
#initialize the list of means
mu <- list(apply(y,2,mean), apply(y,2,mean))
#initialize the list of covariance matrices
sigma <- list(cov(y)*1.2,cov(y)*0.8)
#initialize the prior probability
delta <- c(0.5,0.5)
#initialize the transition probabilities
gamma <- matrix(c(0.9,0.1,0.2,0.8),2,2,byrow=TRUE)
mod1 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma)
#will not run without a shrinkage on the covariance matrices because the
#series is not long enough to reliably estimate the covariance structure
fit1 <- em.hmm(y=y,mod=mod1,cov.shrink=0.0001)
st1 <- viterbi.hmm(y=y,mod=fit1)
sp1 <- smooth.hmm(y=y,mod=fit1)
```

```
## Not run:
#second, fit a Gaussian HMM with 1st order autoregressive structure
auto <- list(matrix(rep(0,2500),50,50,byrow=TRUE),
             matrix(rep(0,2500),50,50,byrow=TRUE))
mod2 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto)
fit2 <- em.hmm(y=y,mod=mod2,ntimes=NULL,cov.shrink=0.0001,arp=1,
               auto.alpha=1,auto.lambda=0.1)
st2 <- viterbi.hmm(y=y,mod=fit2)
sp2 <- smooth.hmm(y=y,mod=fit2)

#third, fit a Gaussian HMM with 2nd order autoregressive structure
auto <- list(matrix(rep(0,5000),50,100,byrow=TRUE),
             matrix(rep(0,5000),50,100,byrow=TRUE))
mod3 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto)
fit3 <- em.hmm(y=y,mod=mod3,ntimes=NULL,cov.shrink=0.0001,arp=2,
               auto.alpha=1,auto.lambda=0.1)
st3 <- viterbi.hmm(y=y,mod=fit3)
sp3 <- smooth.hmm(y=y,mod=fit3)

## End(Not run)
```

---

em.semi *EM algorithm to compute maximum likelihood estimate of Gaussian hidden semi-Markov models with / without autoregressive structures and with / without regularization on the covariance matrices and/or autoregressive structures.*

Description

EM algorithm to compute maximum likelihood estimate of Gaussian hidden semi-Markov models with / without autoregressive structures and with / without regularization on the covariance matrices and/or autoregressive structures.

Usage

```
em.semi(y, mod, ntimes = NULL, tol = 1e-04, maxit = 100, arp = 0,
  cov.shrink = 0, auto.lambda = 0, auto.alpha = 0, print = TRUE)
```

Arguments

y	observed series
mod	list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. mod$d = list of state duration probabilities. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices

ntimes	length of each homogeneous time series. Default to NULL, which means only homogeneous time series.
tol	tolerance for relative change. Default to 1e-4.
maxit	maximum number of iterations. Default to 100.
arp	order of autoregressive. Default to 0.
cov.shrink	shrinkage on the multivariate normal covariance matrix. Default to 0. See references.
auto.lambda	elastic net shrinkage on the autoregressive coefficients. Default to 0. See references.
auto.alpha	The elasticnet mixing parameter, with 0<=alpha<=1. The penalty is defined as (1-alpha)/2||.||_2^2+alpha||.||_1. alpha=1 is the lasso penalty, and alpha=0 the ridge penalty. Default to 0. Same as in the glmnet package.
print	Default to TRUE.

Value

a list containing the fitted parameters.

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67.2 (2005): 301-320.

Ledoit, Olivier, and Michael Wolf. "A well-conditioned estimator for large-dimensional covariance matrices." Journal of multivariate analysis 88.2 (2004): 365-411.

Examples

```
## Not run:
set.seed(332213)
data(finance)
x <- data.matrix(finance)
#log return
y <- x[-1,-51]
for(i in 2:nrow(x)){
 y[i-1,] <- log(x[i,-51]) - log(x[i-1,-51])
}
#annualize the log return
y <- y * 252

#first, fit a Gaussian HMM without autoregressive structure
m <- 2
#initialize the list of means
mu <- list(apply(y,2,mean), apply(y,2,mean))
#initialize the list of covariance matrices
sigma <- list(cov(y)*1.2,cov(y)*0.8)
#initialize the prior probability
```

```
delta <- c(0.5,0.5)
#initialize the transition probabilities
gamma <- matrix(c(0,1,1,0),2,2,byrow=TRUE)
#initialize the state duration probabilities
d <- list(rep(0.1,10),rep(0.1,10))
mod1 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,d=d)
#will not run without a shrinkage on the covariance matrices because the
#series is not long enough to reliably estimate the covariance structure
fit1 <- em.semi(y=y,mod=mod1,cov.shrink=0.0001)
st1 <- viterbi.semi(y=y,mod=fit1)
sp1 <- smooth.semi(y=y,mod=fit1)

#second, fit a Gaussian HSMM with 1st order autoregressive structure
auto <- list(matrix(rep(0,2500),50,50,byrow=TRUE),
             matrix(rep(0,2500),50,50,byrow=TRUE))
mod2 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto,
             d=d,arp=1)
#increase auto.lambda to enforce stronger regularization for model to run
fit2 <- em.semi(y=y,mod=mod2,cov.shrink=0.001,arp=1,
               auto.alpha=0.8,auto.lambda=10)
sum(fit2$auto[[1]]==0)
sum(fit2$auto[[2]]==0)
st2 <- viterbi.semi(y=y,mod=fit2)
sp2 <- smooth.semi(y=y,mod=fit2)

#third, fit a Gaussian HSMM with 2nd order autoregressive structure
auto <- list(matrix(rep(0,5000),50,100,byrow=TRUE),
             matrix(rep(0,5000),50,100,byrow=TRUE))
mod3 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto,
             d=d,arp=2)
#increase auto.lambda to enforce stronger regularization for model to run
fit3 <- em.semi(y=y,mod=mod3,ntimes=NULL,cov.shrink=0.0001,arp=2,
               auto.alpha=0.8,auto.lambda=30)
sum(fit3$auto[[1]]==0)
sum(fit3$auto[[2]]==0)
st3 <- viterbi.semi(y=y,mod=fit3)
sp3 <- smooth.semi(y=y,mod=fit3)

## End(Not run)
```

`finance` *NYSE stock closing price data*

Description

A dataset containing the daily closing price of 50 NYSE stocks from 2015-01-02 to 2016-12-30. The first 50 columns are the daily closing price for each of the 50 stocks, and the last column is the date.

Usage

```
finance
```

Format

A data frame with 504 rows and 51 variables:

Details

The abbreviation for the 50 stocks are as follows: "AMD","BAC","AAPL","FTR","BBRY","GE","RAD","MU","CHK","F", "VALE","FCX","PBR","XOM","INTC","MSFT","WWAV","QQQ","ABEV", "VZ","HPQ","KKFC","PFE","SWN","T","A "LULU","MT","WFT","CLF","SNA","S","C","HCP","DRYS","FMC", "CSCO","KMI","AES","X","SIRI","WLL","COP","

`hmm.predict` *1-step forward prediction for (autoregressive) Gaussian hidden Markov model*

Description

1-step forward prediction for (autoregressive) Gaussian hidden Markov model

Usage

```
hmm.predict(y, mod)
```

Arguments

`y`	observed series
`mod`	list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices

Value

1-step forward state probabilities and forecasts

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(15562)
m <- 2
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0.8,0.2,0.1,0.9),2,2,byrow=TRUE)
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
mod <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto,arp=2)
sim <- hmm.sim(2000,mod)
y <- sim$series
state <- sim$state
fit <- em.hmm(y=y, mod=mod, arp=2, tol=1e-5)
forecast <- hmm.predict(y=y,mod=fit)
```

`hmm.sim` *Simulate a Gaussian hidden Markov series with / without autoregressive structures*

Description

Simulate a Gaussian hidden Markov series with / without autoregressive structures

Usage

```
hmm.sim(ns, mod)
```

Arguments

`ns`	length of the simulated series
`mod`	list consisting of at least the following items: mod$m = number of states, mod$delta = vector of prior probabilities, mod$gamma = matrix of state transition probabilities. mod$mu = list of means, mod$sigma = list of covariance matrices. For autoregressive hidden markov models, we also need the additional items: mod$auto = list of autocorrelation matrices. mod$arp = order of autoregressive.

Value

a list containing simulated series and states

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(135)
#Gaussian HMM 3 hidden states (no autoregressive structure)
m <- 3
mu <- list(c(3),c(-2),c(0))
sigma <- list(as.matrix(1), as.matrix(0.8),as.matrix(0.3))
delta <- c(0.3,0.3,0.4)
gamma <- matrix(c(0.8,0.1,0.1,0.1,0.8,0.1,0.1,0.1,0.8),3,3,byrow=TRUE)
mod1 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma)
sim1 <- hmm.sim(1000,mod1)
y1 <- sim1$series
fit1 <- em.hmm(y=y1, mod=mod1)

#AR(2) Gaussian HMM with 3 hidden states
m <- 2
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0.8,0.2,0.1,0.9),2,2,byrow=TRUE)
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
mod2 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto,arp=2)
sim2 <- hmm.sim(2000,mod2)
y2 <- sim2$series
fit2 <- em.hmm(y=y2, mod=mod2, arp=2)
```

`hsmm.predict` *1-step forward prediction for (autoregressive) Gaussian hidden semi-Markov model*

Description

1-step forward prediction for (autoregressive) Gaussian hidden semi-Markov model

Usage

```
hsmm.predict(y, mod)
```

Arguments

y	observed series
mod	list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. mod$d = list of state duration probabilities. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices

Value

1-step forward state probabilities and forecasts

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(15562)
m <- 2
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0,1,1,0),2,2,byrow=TRUE)
d <- list(c(0.4,0.2,0.1,0.1,0.1,0.1),c(0.5,0.3,0.2))
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
mod <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,
            auto=auto,arp=2,d=d)
sim <- hsmm.sim(2000,mod)
y <- sim$series
state <- sim$state
fit <- em.semi(y=y, mod=mod, arp=2,tol=1e-5)
forecast <- hsmm.predict(y=y,mod=fit)
```

hsmm.sim *Simulate a Gaussian hidden semi-Markov series with / without autoregressive structures*

Description

Simulate a Gaussian hidden semi-Markov series with / without autoregressive structures

Usage

```
hsmm.sim(ns, mod)
```

Arguments

ns	length of the simulated series
mod	list consisting of at least the following items: mod$m = number of states, mod$delta = vector of prior probabilities, mod$gamma = matrix of state transition probabilies. mod$mu = list of means, mod$sigma = list of covariance matrices. mod$d = list of state duration probabilities. For autoregressive hidden markov models, we also need the additional items: mod$auto = list of autocorrelation matrices. mod$arp = order of autoregressive.

Value

a list containing simulated series and states

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(351)
#Gaussian HSMM 3 hidden states (no autoregressive structure)
m <- 3
mu <- list(c(3),c(-2),c(0))
sigma <- list(as.matrix(1), as.matrix(0.8),as.matrix(0.3))
delta <- c(0.3,0.3,0.4)
gamma <- matrix(c(0,0.5,0.5,0.5,0,0.5,0.5,0.5,0),3,3,byrow=TRUE)
d <- list(c(0.4,0.3,0.2,0.1), c(0.5,0.25,0.25), c(0.7,0.3))
mod1 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,d=d)
sim1 <- hsmm.sim(500,mod1)
y1 <- sim1$series
fit1 <- em.semi(y=y1, mod=mod1)

## Not run:
#AR(2) Gaussian HSMM with 3 hidden states
m <- 2
```

```
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0,1,1,0),2,2,byrow=TRUE)
d <- list(c(0.4,0.2,0.1,0.1,0.1,0.1),c(0.5,0.3,0.2))
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
mod2 <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,
            auto=auto,arp=2,d=d)
sim2 <- hsmm.sim(2000,mod2)
y2 <- sim2$series
fit2 <- em.semi(y=y2, mod=mod2, arp=2)

## End(Not run)
```

`mvdnorm` *multivariate normal density*

Description

multivariate normal density

Usage

```
mvdnorm(x, mean, sigma, logd)
```

Arguments

`x`	matrix with each row as an observed vector of multivariate normal RVs
`mean`	vector of means
`sigma`	variance-covariance matrix
`logd`	whether log transformation should be used

Value

a vector of density values for each observed vector

mvrnorm — *multivariate normal random number generator*

Description

multivariate normal random number generator

Usage

```
mvrnorm(n, mu, sigma)
```

Arguments

n	number of random vectors to generate
mu	vector of means
sigma	variance-covariance matrix

Value

a matrix with each row as a realization of multivariate normal random vector

package-rarhsmm — *Regularized Autoregressive Hidden Semi Markov Models*

Description

Package:	rarhsmm
Type:	Package
Version:	1.0.7
Date:	2018-03-19
License:	GPL
LazyLoad:	yes
LazyData:	yes

Author(s)

Zekun Xu <zekunxu@gmail.com>

Ye Liu <yliu87@ncsu.edu> Maintainer: Zekun Xu <zekunxu@gmail.com>

`rmultinomial` *multinomial random variable generator*

Description

multinomial random variable generator

Usage

```
rmultinomial(n, k, prob, label)
```

Arguments

`n`	number of random variables to generate
`k`	number of categories
`prob`	vector of probabilities summing up to 1
`label`	vector of labels for each category

Value

multinomial random variables

`smooth.hmm` *Calculate the probability of being in a particular state for each observation.*

Description

Calculate the probability of being in a particular state for each observation.

Usage

```
smooth.hmm(y, mod)
```

Arguments

`y`	observed series
`mod`	list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices

Value

a matrix containing the state probabilities

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(15562)
m <- 2
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0.8,0.2,0.1,0.9),2,2,byrow=TRUE)
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
mod <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto,arp=2)
sim <- hmm.sim(2000,mod)
y <- sim$series
state <- sim$state
fit <- em.hmm(y=y, mod=mod, arp=2)
stateprob <- smooth.hmm(y=y,mod=fit)
head(cbind(state,stateprob),20)
```

---

smooth.semi — *Calculate the probability of being in a particular state for each observation.*

Description

Calculate the probability of being in a particular state for each observation.

Usage

```
smooth.semi(y, mod)
```

Arguments

y	observed series

`mod` list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. mod$d = list of state duration probabilities. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices

Value

a matrix containing the state probabilities

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(15562)
m <- 2
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0,1,1,0),2,2,byrow=TRUE)
d <- list(c(0.4,0.2,0.1,0.1,0.1,0.1),c(0.5,0.3,0.2))
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
mod <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,
            auto=auto,arp=2,d=d)
sim <- hsmm.sim(2000,mod)
y <- sim$series
state <- sim$state
fit <- em.semi(y=y, mod=mod, arp=2)
stateprob <- smooth.semi(y=y,mod=fit)
head(cbind(state,stateprob),20)
```

---

viterbi.hmm *Viterbi algorithm to decode the latent states for Gaussian hidden Markov model with / without autoregressive structures*

Description

Viterbi algorithm to decode the latent states for Gaussian hidden Markov model with / without autoregressive structures

Usage

```
viterbi.hmm(y, mod)
```

Arguments

y	observed series
mod	list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices

Value

a list containing the decoded states

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(135)
m <- 2
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0.8,0.2,0.1,0.9),2,2,byrow=TRUE)
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
mod <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,auto=auto,arp=2)
sim <- hmm.sim(2000,mod)
y <- sim$series
state <- sim$state
fit <- em.hmm(y=y, mod=mod, arp=2)
state_est <- viterbi.hmm(y=y,mod=fit)
sum(state_est!=state)
```

viterbi.semi — *Viterbi algorithm to decode the latent states for Gaussian hidden semi-Markov model with / without autoregressive structures*

Description

Viterbi algorithm to decode the latent states for Gaussian hidden semi-Markov model with / without autoregressive structures

Usage

```
viterbi.semi(y, mod)
```

Arguments

y	observed series
mod	list consisting the at least the following items: mod$m = scalar number of states, mod$delta = vector of initial values for prior probabilities, mod$gamma = matrix of initial values for state transition probabilies. mod$mu = list of initial values for means, mod$sigma = list of initial values for covariance matrices. mod$d = list of state duration probabilities. For autoregressive hidden markov models, we also need the additional items: mod$arp = scalar order of autoregressive structure mod$auto = list of initial values for autoregressive coefficient matrices

Value

a list containing the decoded states

References

Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

Examples

```
set.seed(135)
m <- 2
mu <- list(c(3,4,5),c(-2,-3,-4))
sigma <- list(diag(1.3,3),
            matrix(c(1,-0.3,0.2,-0.3,1.5,0.3,0.2,0.3,2),3,3,byrow=TRUE))
delta <- c(0.5,0.5)
gamma <- matrix(c(0,1,1,0),2,2,byrow=TRUE)
auto <- list(matrix(c(0.3,0.2,0.1,0.4,0.3,0.2,
                     -0.3,-0.2,-0.1,0.3,0.2,0.1,
                      0,0,0,0,0,0),3,6,byrow=TRUE),
            matrix(c(0.2,0,0,0.4,0,0,
                      0,0.2,0,0,0.4,0,
```

```
                     0,0,0.2,0,0,0.4),3,6,byrow=TRUE))
d <- list(c(0.5,0.3,0.2),c(0.6,0.4))
mod <- list(m=m,mu=mu,sigma=sigma,delta=delta,gamma=gamma,
           auto=auto,arp=2,d=d)
sim <- hsmm.sim(2000,mod)
y <- sim$series
state <- sim$state
fit <- em.semi(y=y, mod=mod, arp=2)
state_est <- viterbi.semi(y=y,mod=fit)
sum(state_est!=state)
```

Index