Package 'seqMC'

June 10, 2017

Title Sequential Monte Carlo

Version 0.0.1

Description Sequential Monte Carlo for nonlinear/non-Gaussian state-space models. Implementation is based on the Gordon, Salmond and Smith (1993) Novel approach to nonlinear or non-Gaussian Bayesian state estimation

Depends R (>= 3.4.0)

License Apache License, Version 2.0

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

`seqMC` *Sequential Monte Carlo*

Description

Sequential Monte Carlo

Usage

```
seqMC(f, prob_y_given_x, x0, y, sample_method = c("systematic", "residual",
  "bootstrap"))
```

Arguments

`f`	function, when called with parameter k (time point) and x_k (state vector at time k), it would return x_k+1
`prob_y_given_x`	function, when called with parameter k (time point), y_k (observation vector at time k) and x_k (state vector at time k), it would return the conditional probability/density: Prob(y_k | x_k)

x0	matrix, sample of state vector at time 0, one sample vector per col.
y	matrix, observations, col 1 is observation at time 1, col 2 is observation at time 2, ... etc.
sample_method	character, specify sample method in the resample stage. Default systematic, means "systematic resampling".

Value

sample from posterior distribution of state vectors, a 3D array, with dimension of d x N x K, where d is the length of a state vector, N is the number of samples, K is the number of time steps.

Examples

```
f <- function(k, x) {
 0.5 * x + 25 * x / (1 + x * x) + 8.0 * cos(1.2 * (k-1)) + rnorm(length(x), sd=sqrt(10.0))
}

prob_y_given_x <- function(k, y, x) {
 as.numeric(dnorm(y - x * x / 20.0))
}

### simulate true path ###
K = 50
x = rep(0.0, K+1)
for (k in 1:K) {
 x[k+1] = f(k, x[k])
}
x = x[-1]
y = x * x / 20 + rnorm(length(x))

### estimate the posterior of state vector ####
N = 4000
x0 = matrix(rnorm(N, sd=sqrt(2)), nrow=1, ncol=N)
xhat = seqMC(f, prob_y_given_x, x0, matrix(y, nrow=1))
xhat_mean = apply(xhat, 3, mean)

alpha = 0.05
xhat_ci_lower = apply(xhat, 3, quantile, probs=alpha/2)
xhat_ci_upper = apply(xhat, 3, quantile, probs=1-alpha/2)

plot(x[-1], ylim=c(-40, 40), pch='*')
lines(xhat_ci_lower, lty='dotted')
lines(xhat_mean)
lines(xhat_ci_upper, lty='dotted')
```

Index