

Package ‘seqMC’

June 11, 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:

batchSeqMC	1
seqMC	2
update.seqMC	3
Index	5

batchSeqMC	<i>Sequential Monte Carlo</i>
------------	-------------------------------

Description

Sequential Monte Carlo

Usage

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

Arguments

<code>f</code>	function, when called with parameter <code>t</code> (time point) and <code>x_t</code> (state vector at time <code>t</code>), it would return <code>x_(t+1)</code>
<code>prob_y_given_x</code>	function, when called with parameter <code>t</code> (time point), <code>y_t</code> (observation vector at time <code>t</code>) and <code>x_t</code> (state vector at time <code>t</code>), it would return the conditional probability: $\text{Prob}(y_t x_t)$
<code>x0</code>	matrix, sample of state vector at time 0, each col is a sample of state at time 0.
<code>y</code>	matrix of <code>T</code> cols, observations, col 1 is observation at time 1, col 2 is observation at time 2, ... etc. <code>T</code> is the number of time points.
<code>sample_method</code>	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 $d \times N \times T$, where d is the length of a state vector, N is the number of samples, T is the number of time steps.

seqMC

seqMC creates a seqMC object

Description

seqMC creates a seqMC object

Usage

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

Arguments

<code>f</code>	function, when called with parameter <code>t</code> (time point) and <code>x_t</code> (state vector at time <code>t</code>), it would return <code>x_(t+1)</code>
<code>prob_y_given_x</code>	function, when called with parameter <code>t</code> (time point), <code>y_t</code> (observation vector at time <code>t</code>) and <code>x_t</code> (state vector at time <code>t</code>), it would return the conditional probability: $\text{Prob}(y_t x_t)$
<code>x0</code>	matrix, sample of state vector at time 0, each col is a sample of state at time 0.
<code>sample_method</code>	character, specify sample method in the resample stage. Default systematic, means "systematic resampling".

Value

a seqMC object, which can be updated at each time point. e.g. `obj = update(obj, y)`

Examples

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

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

x0 = matrix(rnorm(4000, sd=2), nrow=1, ncol=4000)

mod = seqMC(f, prob_y_given_x, x0)

### simulate true path ###
x0 = 0.1
T = 50
x = rep(0.0, T)
x[1] = f(0, x0)
for (t in 1:(T-1)) {
  x[t+1] = f(t, x[t])
}
y = x * x / 20 + rnorm(length(x))

### estimate the posterior of state vector given y[t] ####
xhat = sapply(1:T, function(t) {
  mod <- update(mod, y[t])
  c(mean(mod$x), quantile(mod$x, c(0.025, 0.975)))
})

plot(x, ylim=c(-40, 40), pch='*')
lines(xhat[1,])
lines(xhat[2,], lty='dotted')
lines(xhat[3,], lty='dotted')
```

update.seqMC

Update seqMC object after an observation (if y is not missing) or simply update the object to next time point (if y is missing).

Description

Update seqMC object after an observation (if y is not missing) or simply update the object to next time point (if y is missing).

Usage

```
## S3 method for class 'seqMC'
update(obj, y)
```

Arguments

obj	seqMC object.
y	observation at time obj\$t + 1.

Value

seqMC object updated to next time point.

Index

`batchSeqMC`, [1](#)

`seqMC`, [2](#)

`update.seqMC`, [3](#)