Package ‘RcppSMC’

August 30, 2020

Type Package

Title Rcpp Bindings for Sequential Monte Carlo

Version 0.2.2

Date 2020-08-30

Author Dirk Eddelbuettel, Adam M. Johansen and Leah F. South

Maintainer Dirk Eddelbuettel <edd@debian.org>

Description R access to the Sequential Monte Carlo Template Classes
by Johansen <doi:10.18637/jss.v030.i06> is provided. At present, four
additional examples have been added, and the first example from the JSS
paper has been extended. Further integration and extensions are planned.

License GPL (>= 2)

LazyLoad yes

Imports Rcpp (>= 0.11.0), methods

LinkingTo Rcpp, RcppArmadillo

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-08-30 14:40:03 UTC

R topics documented:

`blockpfGaussianOpt` *Block Sampling Particle Filter (Linear Gaussian Model; Optimal Proposal)*

Description

The `blockpfGaussianOpt` function provides a simple example for **RcppSMC**. It is based on a block sampling particle filter for a linear Gaussian model. This is intended only to illustrate the potential of block sampling; one would not ordinarily use a particle filter for a model in which analytic solutions are available. The 'optimal' block sampler in the sense of Doucet, Briers and Senecal (2006) can be implemented in this case.

The `simGaussian` function simulates data from the associated linear Gaussian state space model.

Usage

```
blockpfGaussianOpt(data, particles=1000, lag=5, plot=FALSE)
simGaussian(len)
```

Arguments

`data`	A vector variable containing the sequence of observations.
`particles`	An integer specifying the number of particles.
`lag`	An integer specifying the length of block to use.
`plot`	A boolean variable describing whether plot should illustrate the estimated path along with the uncertainty.
`len`	The length of the data sequence to simulate.

Details

The `blockpfGaussianOpt` function provides a simple example for **RcppSMC**. It is based on a simple linear Gaussian state space model in which the state evolution and observation equations are: x(n) = x(n-1) + e(n) and y(n) = x(n) + f(n) where e(n) and f(n) are mutually-independent standard normal random variables. The 'optimal' block-sampling proposal described by Doucet et al (2006) is employed.

The `simGaussian` function simulates from the same model returning both the state and observation vectors.

Value

The `blockpfGaussianOpt` function returns a matrix containing the final sample paths and a vector containing their weights. The logarithm of the estimated ratio of normalising constants between the final and initial distributions is also returned.

The `simGaussian` function returns a list containing the state and data sequences.

Author(s)

Adam M. Johansen and Dirk Eddelbuettel

References

A. Doucet, M. Briers, and S. Senecal. Efficient Block Sampling Strategies for sequential Monte Carlo methods. Journal of Computational and Graphical Statistics, 15(3):693-711, 2006.

Examples

```
sim <- simGaussian(len=250)
res <- blockpfGaussianOpt(sim$data,lag=5,plot=TRUE)
```

`LinReg` *Simple Linear Regression*

Description

A simple example based on estimating the parameters of a linear regression model using

* Data annealing sequential Monte Carlo (`LinReg`).

* Likelihood annealing sequential Monte Carlo (`LinRegLA`).

* Likelihood annealing sequential Monte Carlo with the temperature schedule, number of MCMC repeats and random walk covariance matrices adapted online (`LinRegLA_adapt`).

Usage

```
LinReg(model, particles = 1000, plot = FALSE)

LinRegLA(model, particles = 1000, temperatures = seq(0, 1, 0.05)^5)

LinRegLA_adapt(model, particles = 1000, resampTol = 0.5, tempTol = 0.9)
```

Arguments

`model`	Choice of regression model (1 for density as the predictor and 2 for adjusted density as the predictor).
`particles`	An integer specifying the number of particles.
`plot`	A boolean variable to determine whether to plot the posterior estimates.
`temperatures`	In likelihood annealing SMC the targets are defined as $P(y\|\theta)^{\gamma_t} P(\theta)$ where $0 = \gamma_0 \leq \ldots \leq \gamma_T = 1$ can be referred to as the temperatures, $P(y\|\theta)$ is the likelihood and $P(\theta)$ is the prior.
`resampTol`	The adaptive implementation of likelihood annealing SMC allows for the resampling tolerance to be specified. This parameter can be set to a value in the range [0,1) corresponding to a fraction of the size of the particle set or it may be set to an integer corresponding to an actual effective sample size.
`tempTol`	A tolerance for adaptive choice of the temperature schedule such that the conditional ESS is maintained at tempTol*particles.

Details

Williams (1959) considers two competing linear regression models for the maximum compression strength parallel to the grain for radiata pine. Both models are of the form

$y_i = \alpha + \beta(x_i - \bar{x}) + \epsilon_i,$

where ϵ_i $N(0, \sigma^2)$ and $i = 1, \ldots, 42$. Here y is the maximum compression strength in pounds per square inch. The density (in pounds per cubic foot) of the radiata pine is considered a useful predictor, so model 1 uses density for x. Model 2 instead considers the density adjusted for resin content, which is associated with high density but not with strength.

This example is frequently used as a test problem in model choice (see for example Carlin and Chib (1995) and Friel and Pettitt (2008)). We use the standard uninformative normal and inverse gamma priors for this example along with the transformation $\phi = log(\sigma^2)$ so that all parameters are on the real line and $\theta = [\alpha, \beta, \phi]$. The evidence can be computed using numerical estimation for both of the competing models. The log evidence is -309.9 for model 1 and -301.4 for model 2.

The `LinReg` function implements a data annealing approach to this example.

The `LinRegLA` function implements a likelihood annealing approach to this example.

The `LinRegLA_adapt` function implements a likelihood annealing approach to this example with adaptation of the temperature schedule, number of MCMC repeats and random walk covariance matrices.

Value

The `LinReg` function returns a `list` containing the final particle approximation to the target (θ and the corresponding weights) as well as the logarithm of the estimated model evidence.

The `LinRegLA` function returns a `list` containing the population of particles and their associates log likelihoods, log priors and weights at each iteration. The effective sample size at each of the iterations and several different estimates of the logarithm of the model evidence are also returned.

The `LinRegLA_adapt` function returns a `list` containing all of the same output as `LinRegLA`, in addition to the adaptively chosen temperature schedule and number of MCMC repeats.

Author(s)

Adam M. Johansen, Dirk Eddelbuettel and Leah F. South

References

B. P. Carlin and S. Chib. Bayesian model choice via Markov chain Monte Carlo. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 57(3):473-484, 1995.

N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 70(3):589-607, 2008.

Williams, E. (1959). Regression analysis. Wiley.

Examples

```
res <- LinReg(model=1, particles=1000, plot=TRUE)

res <- LinRegLA(model=1, particles=1000)
```

```
res <- LinRegLA_adapt(model=1, particles=1000)
```

`nonLinPMMH` *Particle marginal Metropolis-Hastings for a non-linear state space model.*

Description

The `nonLinPMMH` function implements particle marginal Metropolis Hastings for the non-linear state space model described in Section 3.1 of Andrieu et al. (2010).

Usage

```
nonLinPMMH(data, particles = 5000, iterations = 10000, burnin = 0,
  plot = FALSE)
```

Arguments

`data`	A vector of the observed data.
`particles`	An integer specifying the number of particles in the particle filtering estimates of the likelihood.
`iterations`	An integer specifying the number of MCMC iterations.
`burnin`	The number of iterations to remove from the beginning of the MCMC chain (for plotting purposes only).
`plot`	A boolean variable to determine whether to plot the posterior estimates and MCMC chain.

Details

This example uses particle marginal Metropolis Hastings to estimate the standard deviation of the evolution and observation noise in the following non-linear state space model:

$x(n) = 0.5x(n-1) + 25x(n-1)/(1 + x(n-1)^2) + 8cos(1.2n) + e(n)$ and

$y(n) = x(n)^2/20 + f(n)$

where e(n) and f(n) are mutually-independent normal random variables of variances var_evol and var_obs, respectively, and $x(0)\ N(0, 5)$.

Following Andrieu, Doucet and Holenstein (2010), the priors are $var_evol\ IG(0.01, 0.01)$ and $var_obs\ IG(0.01, 0.01)$ where IG is the inverse gamma distribution.

Data can be simulated from the model using `simNonlin`.

Value

A `data.frame` containing the chain of simulated σ_v and σ_w values, as well as the corresponding log likelihood estimates and log prior values.

Author(s)

Adam M. Johansen, Dirk Eddelbuettel and Leah F. South

References

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(3):269-342, 2010.

See Also

simNonlin for a function to simulate from the model and pfNonlinBS for a simple bootrap particle filter applied to a similar non-linear state space model.

Examples

```
## Not run:
sim <- simNonlin(len=500,var_init=5,var_evol=10,var_obs=1,cosSeqOffset=0)
res <- nonLinPMMH(sim$data,particles=5000,iterations=50000,burnin=10000,plot=TRUE)

## End(Not run)
```

---

pfLineartBS *Particle Filter Example*

Description

The pfLineartBS function provides a simple example for **RcppSMC**. It is based on the first example in SMCTC and the discussion in Section 5.1 of Johansen (2009). A simple 'vehicle tracking' problem of 100 observations is solved with 1000 particles.

The pfLineartBSOnlinePlot function provides a simple default 'online' plotting function that is invoked during the estimation process.

The simLineart function simulates data from the model.

Usage

```
pfLineartBS(data, particles=1000, plot=FALSE, onlinePlot)
pfLineartBSOnlinePlot(xm, ym)
simLineart(len)
```

Arguments

data	A two-column matrix or dataframe containing x and y values. The default data set from Johansen (2009) is used as the default if no data is supplied.
particles	An integer specifying the number of particles.
plot	A boolean variable describing whether plot should illustrate the estimated path along with the data.

`onlinePlot`	A user-supplied callback function which is called with the x and y position vectors during each iteration of the algorithm; see pfExOnlinePlot for a simple example.
`xm`	Vector with x position.
`ym`	Vector with y position.
`len`	Length of sequence to simulate

Details

The `pfLineartBS` function provides a simple example for **RcppSMC**. The model is linear with t-distributed innovations. It is based on the `pf` example in the `SMCTC` library, and discussed in the Section 5.1 of his corresponding paper (Johansen, 2009). `simLineart` simulates from the model.

Using the simple `pfExOnlinePlot` function illustrates how callbacks into R, for example for plotting, can be made during the operation of SMC algorithm.

Value

The `pfLineartBS` function returns a `data.frame` containing as many rows as in the input data, and four columns corresponding to the estimated x and y coordinates as well as the estimated velocity in these two directions.

The `simLineart` function returns a list containing the vector of states and the associated vector of observations.

Author(s)

Adam M. Johansen and Dirk Eddelbuettel

References

A. M. Johansen. SMCTC: Sequential Monte Carlo in C++. Journal of Statistical Software, 30(6):1-41, April 2009. https://www.jstatsoft.org/v30/i06/paper

See Also

The SMCTC paper and code at https://www.jstatsoft.org/v30/i06/paper.

Examples

```
res <- pfLineartBS(plot=TRUE)
if (interactive()) ## if not running R CMD check etc
   res <- pfLineartBS(onlinePlot=pfLineartBSOnlinePlot)
```

pfNonlinBS *Nonlinear Bootstrap Particle Filter (Univariate Non-Linear State Space Model)*

Description

The `pfNonlinBS` function provides a simple example for **RcppSMC**. It is a simple “bootstrap” particle filter which employs multinomial resampling after each iteration applied to the ubiquitous "nonlinear state space model" following Gordon, Salmond and Smith (1993).

Usage

```
pfNonlinBS(data, particles=500, plot=FALSE)
```

Arguments

`data`	A vector variable containing the sequence of observations.
`particles`	An integer specifying the number of particles.
`plot`	A boolean variable describing whether a plot should illustrate the (posterior mean) estimated path along with one and two standard deviation intervals.

Details

The `pfNonlinbs` function provides a simple example for **RcppSMC**. It is based on a simple nonlinear state space model in which the state evolution and observation equations are: x(n) = 0.5 x(n-1) + 25 x(n-1) / (1+x(n-1)^2) + 8 cos(1.2(n-1))+ e(n) and y(n) = x(n)^2 / 20 + f(n) where e(n) and f(n) are mutually-independent normal random variables of variances 10.0 and 1.0, respectively. A boostrap proposal (i.e. sampling from the state equation) is used, together with multinomial resampling after each iteration.

Value

The `pfNonlinBS` function returns two vectors, the first containing the posterior filtering means; the second the posterior filtering standard deviations.

Author(s)

Adam M. Johansen, Dirk Eddelbuettel and Leah F. South

References

N. J. Gordon, S. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings-F, 140(2):107-113, April 1993.

See Also

`simNonlin` for a function to simulate from the model and `nonLinPMMH` for an example of particle marginal Metropolis Hastings applied to a non-linear state space model.

Examples

```
sim <- simNonlin(len=50)
res <- pfNonlinBS(sim$data,particles=500,plot=TRUE)
```

---

radiata *Radiata pine dataset (linear regression example)*

Description

This dataset was originally presented in Table 5.1 of Williams (1959) where two non-nested linear regression models were considered.

Usage

```
radiata
```

Format

A data frame with 42 rows and three variables:

y Maximum compression strength (response) in pounds per square inch

x1 Density (predictor 1) in pounds per cubic foot

x2 Adjusted density (predictor 2) in pounds per cubic foot

Source

E. Williams. Regression analysis. Wiley, 1959.

---

simNonlin *Simulates from a simple nonlinear state space model.*

Description

The `simNonlin` function simulates data from the models used in `link{pfNonlinBS}` and `link{nonLinPMMH}`.

Usage

```
simNonlin(len = 50, var_init = 10, var_evol = 10, var_obs = 1,
  cosSeqOffset = -1)
```

Arguments

`len`	The length of data sequence to simulate.
`var_init`	The variance of the noise for the initial state.
`var_evol`	The variance of the noise for the state evolution .
`var_obs`	The variance of the observation noise.
`cosSeqOffset`	This is related to the indexing in the cosine function in the evoluation equation. A value of -1 can be used to follow the specification of Gordon, Salmond and Smith (1993) and 0 can be used to follow Andrieu, Doucet and Holenstein (2010).

Details

The `simNonlin` function simulates from a simple nonlinear state space model with state evolution and observation equations:

$x(n) = 0.5x(n-1) + 25x(n-1)/(1 + x(n-1)^2) + 8cos(1.2(n + cosSeqOffset)) + e(n)$ and

$y(n) = x(n)^2/20 + f(n)$

where $e(n)$ and $f(n)$ are mutually-independent normal random variables of variances var_evol and var_obs, respectively, and $x(0)\ N(0, var_init)$.

Different variations of this model can be found in Gordon, Salmond and Smith (1993) and Andrieu, Doucet and Holenstein (2010). A cosSeqOffset of -1 is consistent with the former and 0 is consistent with the latter.

Value

The `simNonlin` function returns a list containing the state and data sequences.

Author(s)

Adam M. Johansen, Dirk Eddelbuettel and Leah F. South

References

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(3):269-342, 2010.

N. J. Gordon, S. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings-F, 140(2):107-113, April 1993.

See Also

`pfNonlinBS` for a simple bootrap particle filter applied to this model and `nonLinPMMH` for particle marginal Metropolis Hastings applied to estimating the standard deviation of the state evolution and observation noise.

Index