Package 'BIPOD'

February 19, 2015

Type Package

Title BIPOD (Bayesian Inference for Partially Observed diffusions)

Version 0.2.1

Date 2014-03-18

Author Anders Chr. Jensen

Maintainer Anders Chr. Jensen <anders@math.ku.dk>

Description Bayesian parameter estimation for (partially observed) two-dimensional diffusions.

License GPL-3

Depends R(>= 3.0.2)

Imports Rcpp (>= 0.11.1)

LinkingTo Rcpp(>= 0.11.1), RcppArmadillo(>= 0.4.100.2.1)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-03-19 10:21:43

R topics documented:

BIPOD-package *Bayesian parameter estimation in two-dimensional diffusion models with affine drift.*

Description

This package use data augmentation and a Gibbs sampler to sample from the joint posterior of parameters and augmented data. The main function is 'Estfun' which produce an object of class 'BIPOD'. See the help page for 'Estfun' for an example.

Details

Package:	BIPOD
Type:	Package
Version:	0.2.1
Date:	2014-03-18
License:	GPL-3

BBSim DiffSim Estfun plot.BIPOD ShowModels

Author(s)

Anders Chr. Jensen

Maintainer: Anders Chr. Jensen <anders@math.ku.dk>

References

'Importance sampling techniques for estimation of diffusion models' by Papaspiliopoulos and Roberts, 'Statistical Methods for Stochastic Differential Equations', Monographs on Statistics and Applied Probability, Chapman and Hall, 2012 and 'Markov chain Monte Carlo approach to parameter estimation in the FitzHugh-Nagumo model' by Jensen et al, Phys. Rev. E 2012.

BBSim *Function for simulation of p dimensional Brownian bridge*

Description

Simulation of p-dimensional driftless SDE with constant diffusion, conditional on end points: dV_t=Sigma dW_t, conditional on V_0 and V_T. This function makes a call to C++ and it is therefore relatively fast.

Usage

```
BBSim(start, end, n, Sigma=diag(2), T, t0 = 0,seed = 1)
```

Arguments

start	Numerical vector of length p: Starting point for the process
end	Numerical vector of length p: Ending point for the process
n	Positive integer: Number of time points where the process is simulated
Sigma	p*p matrix: The diffusion matrix for the process
T	Positive number: End of time interval.
t0	Non negative number, defaults to 0. Start of time interval.
seed	Integer, defaults to 1. Specifies seed for random generator. If <=0 it is set randomly.

Details

An n*p matrix with columns representing simulations for each coordinate.

Value

An n*p matrix

Examples

```
(tmp <- BBSim(start = c(1,2),
              end   = c(3,5),
              n     = 10,
              Sigma = diag(2),
              T     = 2,
              t0    = 0,
              seed  = 1))
matplot(tmp,type="l")
```

DiffSim — *Simulation of a 2-dimensional diffusion process. See the Model argument for options.*

Description

Function for simulation of 2-dimensional diffusion processes, using the Euler-maruyama scheme.

Usage

```
DiffSim(n, start, Delta, driftpar, Sigma, seed, thin=1, Model)
```

Arguments

n	positive integer: Length of simulation.
start	Numerical vector: Starting point for the simulation.
Delta	Numerical: Time interval between observations.
driftpar	Numerical vector. Parameters of the FitzHugh-Nagumo model.
Sigma	2*2 diffusion matrix.
seed	Integer: Gives the seed for the random number generator. If seed <= 0 the seed is 'randomly' chosen.
thin	Integer: Output only every 'thin' simulation.
Model	Character specifying the model. Currently one of 'OU', 'FHN', 'FHN5' and 'CIR'.

Value

An (n/thin) by 2 matrix.

Examples

```
FH <- DiffSim(n        = 10000,
              start    = c(1,1),
              Delta    = .001,
              driftpar = c(10,0.6,1.5,0.0),
              Sigma    = diag(c(.5,.3)),
              seed     = 1,
              thin     = 100,
              Model    = "FHN")
matplot(FH,type="l")
```

Estfun *Parameter estimation for some two dimensional diffusions.*

Description

Applies a Gibbs sampler to parameters and augmented data for two-dimensional stochastic differential equations. Currently the Ornstein-Uhlenbeck, the stochastic FitzHugh-Nagumo model and the extended FitzHugh-Nagumo model are implemented.

Usage

```
Estfun(data, Delta, ImputeN = 5, seed, GibbsN = 1000,
       parKnown = list(), Start = c(0,0,0,0,1,1), diffPriorMean,
       diffPriorCovar, diffRW=diag(2), LatentPathStart,Model = NULL,
       driftPriorMean, driftPriorCovar, driftRW, LatentMeanY0 = 0,
       LatentVarY0 = 1, RWrhoPaths = 1, RWrho2PathPoints = 1)
```

Arguments

data	Data to estimate parameters from. Matrix or numeric. Dimensions must be n*1 or n*2 depending on whether second coordinate is observed.
Delta	Positive numeric: Time between observations.
ImputeN	Positive integer>=3: M-2 is the number of imputed data points between consecutive observed data.
seed	Positive integer giving the seed for the random number generator. Defaults to random.
GibbsN	Positive integer: Number of iterations of the Gibbs sampler.
parKnown	List of named values for the known drift and diffusion parameters.
Start	Numerical vector with starting values for the drift and diffusion parameters in the Gibbs sampler.
diffPriorMean	numerical vector of length 2. Prior mean for diffusion coefficients
diffPriorCovar	2*2 matrix. Prior variance for diffusion coefficients.
diffRW	Random walk variance for the MH step for the diffusion ocefficients.
LatentPathStart	Numeric of length one or same length as Data. Starting value for the latent path. If LatentPathStart is a single number then all starting values take this value.
Model	Charater, specifying the model. Currently the only options are 'OU','FHN' and 'FHN5'.
driftPriorMean	prior mean for the drift parameters
driftPriorCovar	Prior covariance for the drift parameters
driftRW	Covariance matrix for the RW update of the drift parameters
LatentMeanY0	Prior mean for the first data point of the unobserved coordinate.
LatentVarY0	Prior variance for the first data point of the unobserved coordinate. If 0, the point is fixed at first value of LatentPathStart.
RWrhoPaths	Numeric in [0,1]. Parameter for random walk update of the latent path between observation times. The value 0 samples a BB, the value 1 keeps the current value of the (skeleton) path
RWrho2PathPoints	Parameter for random walk update of the latent coordinate at observation times. The value 0 samples a middle point of a BB, the value 1 keeps the current value of the points

Details

More details for the help page will be added soon.

Value

An object of class BIPOD.

`Drift`	Output of the Gibbs sampler for the drift parameters.
`Diff`	Output of the Gibbs sampler for the diffusion parameters.
`AccRate1`	Accept/reject (1/0) for each path interval and each iteration of the sampler.
`AccRate2`	Accept/reject (1/0) for each path endpoint of the latent coordinate and each iteration of the sampler. Only valid if second coordinate is latent.
`LatentPath`	Output of the Gibbs sampler for the endpoints of the latent path. Only valid when one coordinate is observed.
`diffAcc`	Accept/reject (1/0) for the MH step of the diffusion coefficient.
`Info`	List with information about the estimated model.
`driftPriormu`	Prior mean of the drift parameters.
`driftPriorOmega`	Prior variance in the drift parameters.
`driftRW`	Random Walk variance for updating drift parameters.

Author(s)

Anders Chr. Jensen

Examples

```
Data <- DiffSim(n=5000,
                start=c(0,0),
                Delta=.001,
                driftpar=c(10,5,1.5,.6),
                Sigma=diag(c(.5,.3)),
                seed=1,
                thin=100,
                Model="FHN")

A <- Estfun(data    = Data[,1],
            Delta   = .001*100,
            ImputeN = 5,
            seed    = 2,
            GibbsN  = 500,
            parKnown = list("drift3"=1.5,"drift4"=.6,"diff2"=.3),
            Start=c(10,10,10,10,1,1),
            diffPriorMean= c(0,0),
            diffPriorCovar= diag(2),
            diffRW = diag(c(.01,.02)),
            LatentPathStart = .5,
            Model="FHN",
            driftPriorMean = NULL,
            driftPriorCovar = NULL,
            driftRW = diag(4),
            LatentMeanY0 = 0,
```

```
            LatentVarY0 = 1,
            RWrhoPaths = 0,
            RWrho2PathPoints = 0)

class(A);names(A)
plot(A,type="trace",interval=1,theta=c(10,5,1.5,.6,.5,.3),subset=c(1,2,5))
### plot(A,type="movie",truepath=Data[,2],speed=.01,BY=10,interval=1)
```

`plot.BIPOD` *Plot function for class BIPOD*

Description

Graphical summaries of output from Gibbs sampler

Usage

```
## S3 method for class 'BIPOD'
plot(x, theta = NULL, subset = NULL, type, lag = 20,
       interval = NULL, treshold = 0.1, speed = 0.1, truepath = NULL,
       BY = 1, prop = c(.05,.95), diffPriorMean = NULL,
       diffPriorCovar = NULL, log = FALSE, ...)
```

Arguments

`x`	x: An object of class BIPOD.
`theta`	Optional numerical vector with true parameter values.
`subset`	Numeric vector. Which parameters should be used for plotting. Defaults to all.
`type`	Character choosing plotting type: Either "trace", "hist", "acp", "pairs","SDtrace", "accept", "movie" or "cover". See details.
`lag`	Positive integer: Number of lags used in autocorrelation plot
`interval`	Positive integer or numericla vector: If integer, used as burn in for the Gibbs sampler. If vector, used to subsample Gibbs output. Defaults to no subsampling and no burn in.
`treshold`	Positive numeric: Cut off for acceptance rate.
`speed`	Positive number used for type="movie": How much time to pause between each frame of the movie?
`truepath`	Vector or matrix with latent data. Optional.
`BY`	integer: Only relevant for type="movie". How many frames to skip for each iteration?
`prop`	Numeric vector with values between 0 and 1: Only relevant for type="cover". Specifies quantiles for the sampled paths
`diffPriorMean`	Numeric of length 2, only for type=="hist". The prior mean for the diffusion coefficients.

diffPriorCovar	2*2 matrix, only for type=="hist". The prior covariance for the diffusion coefficients.
log	Boolean, only for type=="hist". If TRUE, the prior density and the estimate of the diffusion coefficients are log transformed before plotting.
...	Additional arguments to be passed to matplot, density or acf.

Details

Different 'type'-argument gives different plots. More details to come...

ShowModels *Prints form of supported stochastic differential equations*

Description

Print function displaying the model structures currently supported. Used to fix the parametrization of the parameters.

Usage

```
ShowModels(Model)
```

Arguments

Model	Character specifying the model. Current options are "OU" for the Ornstein Uhlenbeck process, "FHN" for the stochastic FitzHugh-Nagumo process "FHN5" for the extended FitzHugh-Nagumo model and "CIR" for the Cox-Ingersoll-Ross model.

Details

This function is used to identity the parameter names in the supported models. This is necessary when specifying the "parKnown" argument in the "Estfun" function.

Value

List with three entries:

Model	A 2*1 Matrix with character entries.
Ndrift	Numeric giving the number of drift parameters
Ndiff	Numeric giving the number of diffusion parameters

Examples

```
ShowModels(Model="FHN")
```

Index