Estimation of Stochastic Differential Equations with Sim.DiffProc Package Version 3.2

by Arsalane Chouaib Guidoum[1] and Kamal Boukhetala[2]

February 9, 2016

Abstract

The Stochastic differential equations, especially diffusion processes, have been widely used in physical and biological sciences and in financial economics. In mathematical finance the success of the diffusion process can be attributed to its many attractive properties. However, all models involve unknown parameters or functions, which need to be estimated from observations of the process. The estimation of diffusion processes is therefore a crucial step in all applications, in particular, in applied finance. The main purpose in this vignette is to introduce the pseudo-maximum likelihood estimators for one-dimensional stochastic differential equations, the package implement `Sim.DiffProc` [Guidoum and Boukhetala, 2016] it explains how to use the function `fitsde` for these estimation techniques.

1 Introduction

The estimation based on discrete time observations is in general difficult. The main obstacle is the fact that SDEs do not explicitly specify the conditional dynamics associated with the sampling process. The general framework is given by the following one-dimensional (Itô) SDE:

$$dX_t = f(t, X_t, \underline{\theta})dt + g(t, X_t, \underline{\theta})dW_t, \qquad t \geq 0\,, X_0 = x_0, \tag{1}$$

where W_t is a standard Wiener process, $f : \Theta \times [0, T] \times \mathbb{R} \to \mathbb{R}$, called the drift coefficient, and $g : \Xi \times [0, T] \times \mathbb{R} \to \mathbb{R}^+$, called the diffusion coefficient, are known functions except the unknown parameters $\underline{\theta}$, $\Theta \subset \mathbb{R}$, $\Xi \subset \mathbb{R}$ and $\mathbb{E}(X_0^2) < \infty$. Parameters $\underline{\theta}$ in (1) are crucial for the characterization of dynamic phenomena being considered, naturally, researchers are interested in obtaining better estimates of the parameters using the observation data. There a rich literature and many books with applications in different fields and computer vision, e.g., Prakasa [1999], Sørensen [2000], Kutoyants [2004], Stefano [2008, 2011]. In practical situations the available data are discrete time series (in R class `'ts'`) data sampled over some time interval. Thus, the parameter estimation for discretely observed SDE is non-trivial and during the past decades it has generated a great deal of research effort, and has attracted the attention of lot of researchers. The following list is an attempt to summarise some contributions, despite the great number. However, it is certainly not a complete reference of the subject of the techniques to estimate in SDE's (e.g. Dacunha and Florens [1986], Dohnal [1987], Florens [1989], Genon [1990], Ozaki [1992], Yoshida [1992], Pedersen [1995], Kloeden et all [1996], Kessler [1997], Gallant and Long [1997], Shoji and Ozaki [1997, 1998], Hurn and Lindsay [1999], Florens [1999], Aït-Sahalia [1999, 2002], Nicolau [2002, 2004], Sørensen [2000, 2002, 2004], Hurn et all [2003], Alcock and Burrage [2004], Ogihara and Yoshida [2011], Uchida and Yoshida [2012], Brouste and Stefano [2013],. . .) to the best of our knowledge.

[1]Faculty of Mathematics. Department of Probabilities & Statistics. University of Science and Technology Houari Boumediene. BP 32 El-Alia, U.S.T.H.B, Algeria.
acguidoum@usthb.dz
[2]Faculty of Mathematics. University of Science and Technology Houari Boumediene. BP 32 El-Alia, U.S.T.H.B, Algeria.
kboukhetala@usthb.dz

There already exist a number of packages that can perform for estimation of SDE's with different methods in R Development Core Team [2015]; see sde [Stefano, 2015] it is the accompanying package to the book of [Stefano, 2008]. The yuima project package [Stefano et all, 2014] for simulation and inference for (multidimensional) SDE's; and PSM package [Stig and Søren, 2013] for estimation of linear and non-linear mixed-effects models using SDE's, are freely available on CRAN. The package Sim.DiffProc [Guidoum and Boukhetala, 2016] contains specific function `fitsde` for pseudo-maximum likelihood (also denoted quasi-maximum likelihood) estimators for one-dimensional SDE's, we implement four approximation scheme: **Euler**[1] [Florens, 1989, Yoshida, 1992], **Ozaki**[2] [Ozaki, 1992], **Shoji-Ozaki**[2] [Shoji and Ozaki, 1998] and **Kessler**[2] method [Kessler, 1997]. These approximation schemes do not approximate the transition density of (1) directly but the path of the process X_t in such a way that the discretized version of the process has a likelihood that is usable.

2 Pseudo-likelihood methods

To simplify equation (1), suppose that the infinitesimal coefficients do not depend on t, i.e. consider:

$$dX_t = f(X_t, \underline{\theta})dt + g(X_t, \underline{\theta})dW_t, \qquad t \geq 0\,, X_0 = x_0, \tag{2}$$

X_t is a time-homogeneous process. In this case the transition density (which we always assume to exist) depends only on Δt, x and y. Hence, we can write it in the form $p(\Delta t, x, y)$ (or $p(t_{i-1}, x, t_i, y)$). If Δt is constant and some regularity conditions it is known that $p(t-s, x, y)$ satisfies the equation of Fokker-Planck backward, it focuses on the variable x starting,

$$\frac{\partial p}{\partial s} = f(x)\frac{\partial p}{\partial x} + \frac{1}{2}g^2(x)\frac{\partial^2 p}{\partial x^2}, \tag{3}$$

Only in simple cases can we solve these Partial differential equation (PDEs). Since the transition densities are generally unknown we cannot in principle obtain the MLE. Nevertheless, we will see some methods that can estimate these densities. With initial condition X_0 and θ the p-dimensional parameter of interest, by Markov property of diffusion processes, the likelihood has this form:

$$L_n(\theta) = \prod_{i=1}^{n} \log p_\theta(\Delta t, X_{i-1}, X_i) p_\theta(X_0), \tag{4}$$

and the log-likelihood is:

$$l_n(\theta) = \log L_n(\theta) = \sum_{i=1}^{n} \log p_\theta(\Delta t, X_{i-1}, X_i) + \log(p_\theta(X_0)), \tag{5}$$

(5) be the log-likelihood function associated with the (2). If some conditional moments of X_t are known but not the true transition density p, it is possible to estimate θ from a density h that although not belonging to the family of the true conditional density, is compatible in terms of moments with p. The h density is denoted as the pseudo true density. The pseudo maximum likelihood (PMLE) is defined as the solution of the following optimization problem:

$$\hat{\theta}_n = \underset{\theta \in \Theta}{\operatorname{argmax}}\; h_n(\theta | X_1, X_2, \ldots, X_n), \tag{6}$$

with:

$$h_n(\theta | X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} \log h_\theta(\Delta t, X_{i-1}, X_i) + \log(h_\theta(X_0)), \tag{7}$$

Under some conditions this techniques apply to high frequency data i.e., $\Delta t \to 0$ and $n\Delta t \to +\infty$.

[1] The implementation of this method in R is very easy see Stefano [2008, p. 122].

[2] The functions `dcOzaki`, `dcShoji` and `dcKessler` are available on `sde` package Stefano [2015], for approximated conditional law of a diffusion process.

The Sim.DiffProc package implements PMLE via the fitsde function. The interface and the output of the fitsde function are made as similar as possible to those of the standard mle function in the stats4 package of the basic R system. The main arguments to fitsde consist of a data of type a univariate time series (ts object) and initial values (start) for the optimizer. The drift and diffusion indicate drift and diffusion coefficient of the model (1), is an expression of two variables t, x and theta names of the parameters, and must be nominated by a vector of theta = (theta[1], theta[2],..., theta[p]) for reasons of symbolic derived in approximation methods. The start argument must be specified as a named list, where the names of the elements of the list correspond to the names of the parameters as they appear in the drift and diffusion coefficient. The pmle argument must be a character string specifying the method to use, can be either: "euler", "ozaki", "shoji" and "kessler". We can select the optimization method by the argument optim.method ("L-BFGS-B" is used by default), and further arguments to pass to optim function. The following we explain how to use this function to estimate a model (1) with different approximation methods, as well as other functions of type 'S3' are linking to a class 'fitsde'.

2.1 Euler method

Consider a process solution of the general stochastic differential equation (2), the Euler scheme produces the discretization ($\Delta t \to 0$):

$$X_{t+\Delta t} - X_t = f(X_t, \theta)\Delta t + g(X_t, \theta)(W_{t+\Delta t} - W_t),$$

the increments $X_{t+\Delta t} - X_t$ are then independent Gaussian random variables with mean: $\mathbb{E}_x = f(X_t, \theta)\Delta t$, and variance: $\mathrm{V}_x = g^2(X_t, \theta)\Delta t$. Therefore the transition density of the process can be written as:

$$p_\theta(t, y|x) = \frac{1}{\sqrt{2\pi t g^2(x,\theta)}} \exp\left(-\frac{(y - x - f(x,\theta)t)^2}{2tg^2(x,\theta)}\right)$$

then the log-likelihood is:

$$h_n(\theta|X_1, X_2, \dots, X_n) = -\frac{1}{2}\left(\sum_{i=1}^{n} \frac{(X_i - X_{i-1} - f(X_{i-1},\theta)\Delta)^2}{\sigma^2\Delta t} + n\log(2\pi\sigma^2\Delta t)\right) \tag{8}$$

the equation above is also called the locally Gaussian approximation. Florens [1989] and Yoshida [1992] showed that a consistent estimator of σ^2 is:

$$\hat{\sigma}^2 = \frac{1}{n\Delta t}\sum_{i=1}^{n}(X_i - X_{i-1})^2$$

As an example, we consider the Chan-Karolyi-Longstaff-Sanders (CKLS) model:

$$dX_t = (\theta_1 + \theta_2 X_t)dt + \theta_3 X_t^{\theta_4} dW_t, \qquad X_0 = 2 \tag{9}$$

with $\theta_1 = 1$, $\theta_2 = 2$, $\theta_3 = 0.5$ and $\theta_4 = 0.3$. Before calling fitsde, we generate sampled data X_{t_i}, with $\Delta t = 10^{-3}$, as following:

```
> f <- expression( (1+2*x) )
> g <- expression( 0.5*x^0.3 )
> sim    <- snssde1d(drift=f,diffusion=g,x0=2,M=1,N=1000,Dt=0.001)
> mydata <- sim$X
```

we set the initial values for the optimizer as $\theta_1 = \theta_2 = \theta_3 = \theta_4 = 1$, and we specify the coefficients drift and diffusion as expressions. you can use the upper and lower limits of the search region used by the optimizer (here using the default method "L-BFGS-B"), and specifying the method to use with pmle="euler". We can now use the function fitsde to estimate the parameters of (9) as follows:

```
> fx <- expression( theta[1]+theta[2]*x ) ## drift coefficient of model (9)
> gx <- expression( theta[3]*x^theta[4] ) ## diffusion coefficient of model (9)
> fitmod <- fitsde(data=mydata,drift=fx,diffusion=gx,start = list(theta1=1,
+                  theta2=1,theta3=1,theta4=1),pmle="euler")
```

The estimated coefficients are extracted from the output object `fitmod` as follows:

```
> coef(fitmod)

   theta1    theta2    theta3    theta4
1.1201172 2.1384034 0.5288006 0.2701521
```

we can use the `summary` function to produce result summaries of output object `fitmod`:

```
> summary(fitmod)

Pseudo maximum likelihood estimation

Method:  Euler
Call:
fitsde(data = mydata, drift = fx, diffusion = gx, start = list(theta1 = 1,
    theta2 = 1, theta3 = 1, theta4 = 1), pmle = "euler")

Coefficients:
        Estimate Std. Error
theta1 1.1201172 1.57355762
theta2 2.1384034 0.19229249
theta3 0.5288006 0.03752510
theta4 0.2701521 0.03479513

-2 log L: -4298.304
```

The functions of type `S3` method (as similar of the standard `mle` function in the `stats4` package of the basic R system) for the class `'fitsde'` are the following:

- `coef`: which extracts model coefficients from objects returned by `'fitsde'`.
- `vcov`: returns the variance-covariance matrix of the parameters of a fitted model objects.
- `logLik`: extract log-likelihood.
- `AIC`: calculating Akaike's Information Criterion for fitted model objects.
- `BIC`: calculating Schwarz's Bayesian Criterion for fitted model objects.
- `confint`: computes confidence intervals for one or more parameters in a fitted model objects.

```
> vcov(fitmod)

             theta1        theta2        theta3        theta4
theta1  2.476083599 -0.2524732417 -0.0038764366  0.0038346123
theta2 -0.252473242  0.0369764031  0.0004688301 -0.0004628819
theta3 -0.003876437  0.0004688301  0.0014081329 -0.0012391759
theta4  0.003834612 -0.0004628819 -0.0012391759  0.0012107014

> AIC(fitmod)

[1] -4290.304
```

```
> confint(fitmod, level=0.95)

             2.5 %    97.5 %
theta1 -1.9639990 4.2042335
theta2  1.7615170 2.5152898
theta3  0.4552527 0.6023484
theta4  0.2019549 0.3383494
```

2.2 Ozaki method

The second approach we present is the Ozaki method (Ozaki [1992], Shoji and Ozaki [1997]), and it works for homogeneous stochastic differential equations. Consider the stochastic differential equation:

$$dX_t = f(X_t, \underline{\theta})dt + \sigma dW_t, \qquad t \geq 0\,, X_0 = x_0, \tag{10}$$

where σ is supposed to be constant. We just recall that the transition density for the Ozaki method is Gaussian, we have that: $X_{t+\Delta t}|X_t = x \sim \mathcal{N}(\mathbb{E}_x, \mathrm{V}_x)$, where:

$$\mathbb{E}_x = x + \frac{f(x)}{\partial_x f(x)}\left(e^{\partial_x f(x)\Delta t} - 1\right), \tag{11}$$

$$\mathrm{V}_x = \sigma^2 \frac{e^{2K_x \Delta t} - 1}{2K_x}, \tag{12}$$

with:

$$K_x = \frac{1}{\Delta t}\log\left(1 + \frac{f(x)}{x\partial_x f(x)}\left(e^{\partial_x f(x)\Delta t} - 1\right)\right)$$

It is always possible to transform equation (2) with a constant diffusion coefficient using the Lamperti transform (see Shoji and Ozaki [1998], Florens [1999], Aït-Sahalia [2002] and Stefano [2008, p. 40]).

Now we consider the Vasicek model, with $f(x,\theta) = \theta_1(\theta_2 - x)$ and constant volatility $g(x,\theta) = \theta_3$,

$$dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3 dW_t, \qquad X_0 = 5 \tag{13}$$

with $\theta_1 = 3$, $\theta_2 = 2$ and $\theta_3 = 0.5$, we generate sampled data X_{t_i}, with $\Delta t = 10^{-2}$, as following:

```
> f <- expression( 3*(2-x) )
> g <- expression( 0.5 )
> sim <- snssde1d(drift=f,diffusion=g,x0=5,Dt=0.01)
> HWV <- sim$X
```

we set the initial values for the optimizer as $\theta_1 = \theta_2 = \theta_3 = 1$, and we specify the coefficients drift and diffusion as expressions. Specifying the method to use with `pmle="ozaki"`, which can easily be implemented in R as follows:

```
> fx <- expression( theta[1]*(theta[2]- x) ) ## drift coefficient of model (13)
> gx <- expression( theta[3] )              ## diffusion coefficient of model (13)
> fitmod <- fitsde(data=HWV,drift=fx,diffusion=gx,start = list(theta1=1,theta2=1,
+                   theta3=1),pmle="ozaki")
> summary(fitmod)

Pseudo maximum likelihood estimation

Method:  Ozaki
Call:
fitsde(data = HWV, drift = fx, diffusion = gx, start = list(theta1 = 1,
    theta2 = 1, theta3 = 1), pmle = "ozaki")
```

```
        Estimate Std. Error
theta1 3.0844858 0.56115144
theta2 1.9897618 0.22305649
theta3 0.4675424 0.03306254

-2 log L: -329.5772
```

If you want to have confidence intervals θ_1 and θ_2 parameters only, using the command `confint` as following:

```
> confint(fitmod,parm=c("theta1","theta2"),level=0.95)

          2.5 %   97.5 %
theta1 1.984649 4.184322
theta2 1.552579 2.426944
```

2.3 Shoji-Ozaki method

An extension of the method to Ozaki the more general case where the drift is allowed to depend on the time variable t, and also the diffusion coefficient can be varied is the method Shoji and Ozaki [1998]. Consider the stochastic differential equation:

$$dX_t = f(t, X_t, \underline{\theta})dt + g(X_t, \underline{\theta})dW_t, \qquad t \geq 0\,, X_0 = x_0, \tag{14}$$

the transition density for the Shoji-Ozaki method is Gaussian, we have that: $X_{t+\Delta t}|X_t = x \sim \mathcal{N}\left(\mathrm{A}_{(t,x)}x, \mathrm{B}^2_{(t,x)}\right)$, where:

$$\mathrm{A}_{(t,x)} = 1 + \frac{f(t,x)}{x\mathrm{L}_t}\left(e^{\mathrm{L}_t\Delta t} - 1\right) + \frac{\mathrm{M}_t}{x\mathrm{L}_t^2}\left(e^{\mathrm{L}_t\Delta t} - 1 - \mathrm{L}_t\Delta t\right), \tag{15}$$

$$\mathrm{B}_{(t,x)} = g(x)\sqrt{\frac{e^{2\mathrm{L}_t\Delta t} - 1}{2\mathrm{L}_t}}, \tag{16}$$

with:

$$\mathrm{L}_t = \partial_x f(t,x) \quad \text{and} \quad \mathrm{M}_t = \frac{g^2(x)}{2}\partial_{xx} f(t,x) + \partial_t f(t,x).$$

for more details, can be found in the original works Shoji and Ozaki [1997, 1998]. As an example, we consider the following model:

$$dX_t = a(t)X_t dt + \theta_2 X_t dW_t, \qquad X_0 = 10 \tag{17}$$

with: $a(t) = \theta_1 t$, and we generate sampled data X_{t_i}, with $\theta_1 = -2$, $\theta_2 = 0.2$ and time step $\Delta t = 10^{-3}$, as following:

```
> f <- expression(-2*x*t)
> g <- expression(0.2*x)
> sim <- snssde1d(drift=f,diffusion=g,N=1000,Dt=0.001,x0=10)
> mydata <- sim$X
```

we set the initial values for the optimizer as $\theta_1 = \theta_2 = 1$, and we specify the method to use with `pmle="shoji"`,

```
> fx <- expression( theta[1]*x*t ) ## drift coefficient of model (17)
> gx <- expression( theta[2]*x )   ## diffusion coefficient of model (17)
> fitmod <- fitsde(data=mydata,drift=fx,diffusion=gx,start = list(theta1=1,
+                    theta2=1),pmle="shoji",lower=c(-3,0),upper=c(-1,1))
> summary(fitmod)
```

```
Pseudo maximum likelihood estimation

Method:  Shoji
Call:
fitsde(data = mydata, drift = fx, diffusion = gx, start = list(theta1 = 1,
    theta2 = 1), pmle = "shoji", lower = c(-3, 0), upper = c(-1,
    1))

Coefficients:
         Estimate Std. Error
theta1 -1.8737101 0.36086610
theta2  0.2081152 0.00465562

-2 log L: -2872.243

> vcov(fitmod)

              theta1        theta2
theta1  1.302243e-01 -6.778558e-06
theta2 -6.778558e-06  2.167479e-05

> logLik(fitmod)

[1] 1436.121

> confint(fitmod,level=0.9)

              5 %       95 %
theta1 -2.4672820 -1.2801382
theta2  0.2004574  0.2157731
```

2.4 Kessler method

Kessler [1997] proposed to use a higher-order Itô-Taylor expansion to approximate the mean and variance in a conditional Gaussian density. Consider the stochastic differential equation (1), the transition density by Kessler method is: $X_{t+\Delta t}|X_t = x \sim \mathcal{N}(\mathbb{E}_x, \mathrm{V}_x)$, where:

$$\mathbb{E}_x = x + f(t,x)\Delta t + \left(f(t,x)\partial_x f(t,x) + \frac{1}{2}g^2(t,x)\partial_{xx}g(t,x)\right)\frac{(\Delta t)^2}{2}, \tag{18}$$

$$\begin{aligned}\mathrm{V}_x = x^2 + (2f(t,x)x + g^2(t,x))\Delta t + \Big(&2f(t,x)\left(\partial_x f(t,x)x + f(t,x) + g(t,x)\partial_x g(t,x)\right)\\ &+ g^2(t,x)\left(\partial_{xx}f(t,x)x + 2\partial_x f(t,x) + \partial_x g^2(t,x) + g(t,x)\partial_{xx}g(t,x)\right)\Big)\frac{(\Delta t)^2}{2} - \mathbb{E}_x^2.\end{aligned} \tag{19}$$

In the framework consider by this approximation, see Kessler [1997] for the result for the maximum likelihood estimator.

We consider the following Hull-White (extended Vasicek) model:

$$dX_t = a(t)(b(t) - X_t)dt + \sigma(t)dW_t, \qquad X_0 = 2 \tag{20}$$

with: $a(t) = \theta_1 t$ and $b(t) = \theta_2\sqrt{t}$, the volatility depends on time: $\sigma(t) = \theta_3 t$. We generate sampled data of (20), with $\theta_1 = 3$, $\theta_2 = 1$ and $\theta_3 = 0.3$, time step $\Delta t = 10^{-3}$, as following:

```
> f <- expression(3*t*(sqrt(t)-x))
> g <- expression(0.3*t)
> sim <- snssde1d(drift=f,diffusion=g,M=1,N=1000,x0=2,Dt=0.001)
> mydata <- sim$X
```

we set the initial values for the optimizer as $\theta_1 = \theta_2 = \theta_3 = 1$, and we specify the method to use with `pmle="kessler"`,

```
> ## drift coefficient of model (20)
> fx <- expression( theta[1]*t* ( theta[2]*sqrt(t) - x ) )
> ## diffusion coefficient of model (20)
> gx <- expression( theta[3]*t )
> fitmod <- fitsde(data=mydata,drift=fx,diffusion=gx,start = list(theta1=1,
+                    theta2=1,theta3=1),pmle="kessler")
> summary(fitmod)

Pseudo maximum likelihood estimation

Method:  Kessler
Call:
fitsde(data = mydata, drift = fx, diffusion = gx, start = list(theta1 = 1,
    theta2 = 1, theta3 = 1), pmle = "kessler")

Coefficients:
        Estimate  Std. Error
theta1 3.1334952 0.333744885
theta2 1.1253709 0.158830540
theta3 0.2957026 0.006615639

-2 log L: -8492.931
```

3 The `fitsde()` in practice

3.1 Estimation of attractive model

We propose the following dispersion models family [Boukhetala, 1996]:

$$dR_t = \left(\frac{0.5\theta_3^2 R_t^{\theta_2 - 1} - \theta_1}{R_t^{\theta_2}} \right) dt + \theta_3 dW_t, \tag{21}$$

where: $2\theta_1 > \theta_3^2$ condition to ensure attractiveness; we generate sampled data of this model, with $\theta_1 = 5$, $\theta_2 = 1$ and $\theta_3 = 0.2$, $\Delta t = 10^{-3}$, as following:

```
> theta1 = 5; theta2 = 1; theta3 = 0.2
> f <- expression( ((0.5*theta3^2 *x^(theta2-1) - theta1)/ x^theta2) )
> g <- expression( theta3 )
> sim <- snssde1d(drift=f,diffusion=g,M=1,N=1000,x0=3,Dt=0.001)
> mydata <- sim$X
```

we use `fitsde` function to estimate the parameters of model (21) as follows:

```
> fx <- expression( ((0.5*theta[3]^2 *x^(theta[2]-1) - theta[1])/ x^theta[2])  )
> gx <- expression(theta[3])
> fitmod <- fitsde(mydata,drift=fx,diffusion=gx, start = list(theta1=1,theta2=1,
+                  theta3=1),lower=c(0,0,0),pmle="euler")
> coef(fitmod)

   theta1    theta2    theta3
5.0176690 1.0077608 0.2040635
```

for to calculate the bias and confidence intervals of estimators it is easy, we can proceed as follows:

```
> true <- c(theta1,theta2,theta3)    ## True parameters
> bias <- true-coef(fitmod)
> bias

      theta1       theta2       theta3
-0.017669042 -0.007760805 -0.004063473

> confint(fitmod)

           2.5 %    97.5 %
theta1 4.4894051 5.5459330
theta2 0.9594991 1.0560225
theta3 0.1951120 0.2130149
```

3.2 Model selection via AIC

The aim is to try to identify the underlying continuous model on the basis of discrete observations using AIC (Akaike Information Criterion) statistics. Uchida and Yoshida [2005] develop the AIC statistics defined as:

$$\text{AIC} = -2h_n\left(\hat{\theta}_n^{\text{PML}}\right) + 2\dim(\Theta), \tag{22}$$

where $\hat{\theta}_n^{\text{PML}}$ is the pseudo maximum likelihood estimator and h_n the local Gaussian approximation of the true log-likelihood. When comparing several models for a given data set, the models such that the AIC is lower is preferred.

Let the following models:

$$\begin{aligned}
dX_t &= \theta_1 X_t dt + \theta_2 X_t^{\theta_3} dW_t, && \text{(true model)}\\
dX_t &= (\theta_1 + \theta_2 X_t)dt + \theta_3 X_t^{\theta_4} dW_t, && \text{(competing model 1)}\\
dX_t &= (\theta_1 + \theta_2 X_t)dt + \theta_3 \sqrt{X_t} dW_t, && \text{(competing model 2)}\\
dX_t &= \theta_1 dt + \theta_2 X_t^{\theta_3} dW_t, && \text{(competing model 3)}
\end{aligned}$$

We generate data from true model with parameters $\underline{\theta} = (2, 0.3, 0.5)$, initial value $X_0 = 2$ and $\Delta t = 10^{-3}$, as following:

```
> f <- expression( 2*x )
> g <- expression( 0.3*x^0.5 )
> sim <- snssde1d(drift=f,diffusion=g,M=1,N=1000,x0=2,Dt=0.001)
> mydata <- sim$X
```

We test the performance of the AIC statistics for the four competing models, we can proceed as follows:

```
> ## True model
> fx <- expression( theta[1]*x )
> gx <- expression( theta[2]*x^theta[3] )
> truemod <- fitsde(data=mydata,drift=fx,diffusion=gx,start = list(theta1=1,
+                    theta2=1,theta3=1),pmle="euler")
> ## competing model 1
> fx1 <- expression( theta[1]+theta[2]*x )
> gx1 <- expression( theta[3]*x^theta[4] )
> mod1 <- fitsde(data=mydata,drift=fx1,diffusion=gx1,start = list(theta1=1,
+            theta2=1,theta3=1,theta4=1),pmle="euler")
> ## competing model 2
> fx2 <- expression( theta[1]+theta[2]*x )
> gx2 <- expression( theta[3]*sqrt(x) )
```

```
> mod2 <- fitsde(data=mydata,drift=fx2,diffusion=gx2,start = list(theta1=1,
+           theta2=1,theta3=1),pmle="euler")
> ## competing model 3
> fx3 <- expression( theta[1] )
> gx3 <- expression( theta[2]*x^theta[3] )
> mod3 <- fitsde(data=mydata,drift=fx3,diffusion=gx3,start = list(theta1=1,
+           theta2=1,theta3=1),pmle="euler")
> ## Computes AIC
> AIC <- c(AIC(truemod),AIC(mod1),AIC(mod2),AIC(mod3))
> Test <- data.frame(AIC,row.names = c("True mod","Comp mod1","Comp mod2",
+                    "Comp mod3"))
> Test

                AIC
True mod  -4867.597
Comp mod1 -4865.617
Comp mod2 -4867.587
Comp mod3 -4810.597

> Bestmod <- rownames(Test)[which.min(Test[,1])]
> Bestmod

[1] "True mod"
```

the estimates under the different models,

```
> Theta1 <- c(coef(truemod)[[1]],coef(mod1)[[1]],coef(mod2)[[1]],coef(mod3)[[1]])
> Theta2 <- c(coef(truemod)[[2]],coef(mod1)[[2]],coef(mod2)[[2]],coef(mod3)[[2]])
> Theta3 <- c(coef(truemod)[[3]],coef(mod1)[[3]],coef(mod2)[[3]],coef(mod3)[[3]])
> Theta4 <- c("",coef(mod1)[[4]],"","")
> Parms  <- data.frame(Theta1,Theta2,Theta3,Theta4,row.names = c("True mod",
+                      "Comp mod1","Comp mod2","Comp mod3"))
> Parms

              Theta1    Theta2    Theta3    Theta4
True mod   1.8978518 0.3007418 0.4928129
Comp mod1 -0.1776481 1.9283156 0.3007623 0.4927636
Comp mod2 -0.1742726 1.9276524 0.2972557
Comp mod3  8.2200670 0.2914395 0.5297354
```

3.3 Application to real data

We make use of real data of the U.S. Interest Rates monthly form 06/1964 to 12/1989 (see Figure 1) available in package `Ecdat` [?], and we estimate the parameters $\underline{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)$ of CKLS model (9). These data have been analyzed by Stefano et all [2014] with `yuima` package, here we confirm the results of the estimates by several approximation methods.

```
> data(Irates)
> rates <- Irates[, "r1"]
> X <- window(rates, start = 1964.471, end = 1989.333)

> plot(X)
```

we can now use all previous methods by `fitsde` function to estimate the parameters of model (9) as follows:

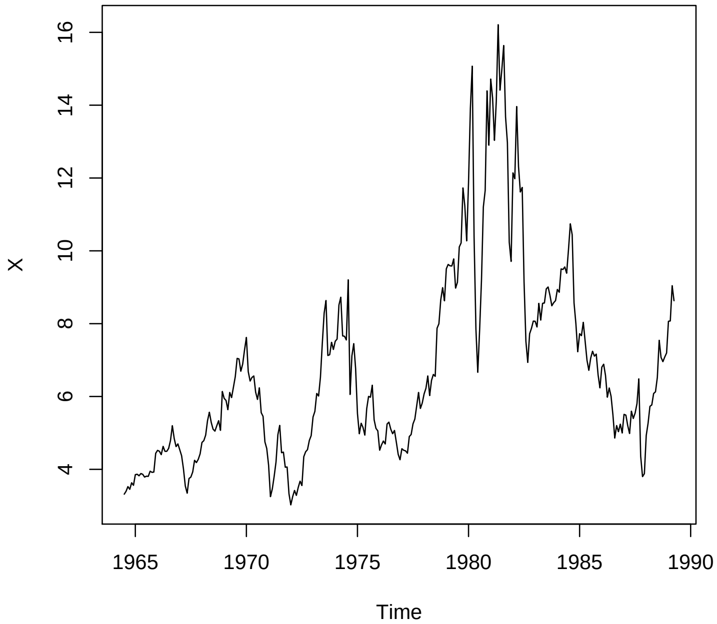

Figure 1: The U.S. Interest Rates monthly data from 06/1964 to 12/1989.

```
> fx <- expression( theta[1]+theta[2]*x ) ## drift coefficient of model (9)
> gx <- expression( theta[3]*x^theta[4] ) ## diffusion coefficient of model (9)
> pmle <- eval(formals(fitsde.default)$pmle)
> fitres <- lapply(1:4, function(i) fitsde(X,drift=fx,diffusion=gx,pmle=pmle[i],
+                   start = list(theta1=1,theta2=1,theta3=1,theta4=1)))
> Coef <- data.frame(do.call("cbind",lapply(1:4,function(i) coef(fitres[[i]]))))
> Info <- data.frame(do.call("rbind",lapply(1:4,function(i) logLik(fitres[[i]]))),
+                    do.call("rbind",lapply(1:4,function(i) AIC(fitres[[i]]))),
+                    do.call("rbind",lapply(1:4,function(i) BIC(fitres[[i]]))),
+                    row.names=pmle)
> names(Coef) <- c(pmle)
> names(Info) <- c("logLik","AIC","BIC")
> Coef

            euler    kessler      ozaki      shoji
theta1  2.0769516  2.1433505  2.1153154  2.1015009
theta2 -0.2631871 -0.2743368 -0.2690547 -0.2664674
theta3  0.1302158  0.1259800  0.1265225  0.1316708
theta4  1.4513173  1.4691660  1.4649140  1.4513080

> Info

           logLik      AIC      BIC
euler   -237.8786 483.7572 487.1514
kessler -237.7845 483.5690 486.9632
ozaki   -237.8356 483.6712 487.0654
shoji   -237.8786 483.7572 487.1514
```

In Figure 2 we reports the sample mean of the solution of the model (23) with the estimated parameters and real data, their empirical 95% confidence bands (from the 2.5th to the 97.5th

percentile), we can proceed as follows:

$$dX_t = (2.076 - 0.263X_t)dt + 0.130X_t^{1.451}dW_t, \tag{23}$$

```
> f <- expression( (2.076-0.263*x) )
> g <- expression( 0.130*x^1.451 )
> mod <- snssde1d(drift=f,diffusion=g,x0=X[1],M=200, N=length(X),t0=1964.471,
+                    T=1989.333)
> mod

Ito Sde 1D:
        | dx = (2.076 - 0.263 * x) * dt + 0.13 * x^1.451 * dw
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process        | N  = 298.
        | Number of simulation   | M  = 200.
        | Initial value          | x0 = 3.317.
        | Time of process        | t in [1964.471,1989.333].
        | Discretization         | Dt = 0.08342953.

> plot(mod,plot.type="single",type="n",ylim=c(0,30))
> lines(X,col=4,lwd=2)
> lines(time(mod),mean(mod),col=2,lwd=2)
> lines(time(mod),bconfint(mod,level=0.95)[,1],col=5,lwd=2)
> lines(time(mod),bconfint(mod,level=0.95)[,2],col=5,lwd=2)
> legend("topleft",c("real data","mean path",paste("bound of", 95,"% confidence")),
+        inset = .01,col=c(4,2,5),lwd=2,cex=0.8)
```

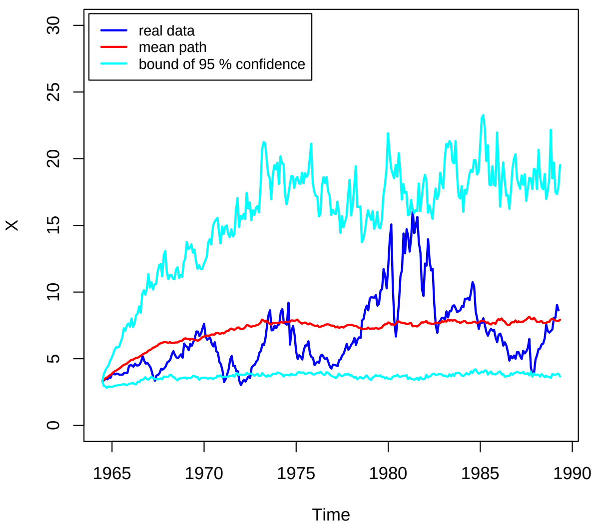

Figure 2: Real data vs the empirical mean of 200 simulated trajectories of model (23).

4 Summary

The asymptotic approach to statistical estimation is frequently adopted because of its general applicability and relative simplicity. In this work we explained the use of `fitsde` function in

Sim.DiffProc package, which is based on pseudo-maximum likelihood estimator for one-dimensional stochastic differential equations, with different approximation methods and some examples of applications.

References

Dacunha, D.C. and Florens, D.Z. (1986). Estimation of the Coefficients of a Diffusion from Discrete Observations. *Stochastics*. 19, 263–284.

Florens, Z.D. (1989). Approximate discrete time schemes for statistics of diffusion processes. *Statistics*. 20, 547–557.

Dohnal, G. (1987). On estimating the diffusion coefficient. *J. Appl.Prob.*, 24, 105–114.

Genon, V.C. (1990). Maximum constrast estimation for diffusion processes from discrete observation. *Statistics*, 21, 99–116.

Aït-Sahalia, Y. (1999). Transition densities for interest rate and other nonlinear diffusions. *The Journal of Finance*, 54, 1361–1395.

Aït-Sahalia, Y. (2002). Maximum likelihood estimation of discretely sampled diffusions: a closed-form approximation approach. *Econometrica*. 70, 223–262.

Kessler, M. (1997). Estimation of an ergodic diffusion from discrete observations. *Scand. J. Statist.*, 24, 211-229.

Ozaki, T. (1992). A bridge between nonlinear time series models and nonlinear stochastic dynamical systems: A local linearization approach. *Statistica Sinica*, 2, 25-83.

Shoji, L. and Ozaki, T. (1997). Comparative study of estimation methods for continuous time stochastic processes. *J.Time Ser. Anal.,* 18, 485–506.

Shoji, L. and Ozaki, T. (1998). Estimation for nonlinear stochastic differential equations by a local linearization method. *Stochastic Analysis and Applications*, 16, 733-752.

Nicolau, J. (2002). New Technique for Simulating the Likelihood of Stochastic Differential Equations. *The Econometrics Journal*. 5(1).

Nicolau, J. (2004). Introduction to the estimation of stochastic differential equations based on discrete observations. *Autumn School and International Conference, Stochastic Finance.*

Boukhetala, K. (1996). Modelling and simulation of a dispersion pollutant with attractive centre. ed by Computational Mechanics Publications, Southampton ,U.K and Computational Mechanics Inc, Boston, USA, 245–252.

Stefano, M.I. (2008). *Simulation and inference for stochastic differential equations: with R examples.* Springer-Verlag, New York.

Stefano, M.I. (2015). sde: Simulation and Inference for Stochastic Differential Equations. *R package version 2.0.14*. http://CRAN.R-project.org/package=sde

Stefano, M.I. (2011). *Option Pricing and Estimation of Financial Models with R.* John Wiley & Sons Ltd.

Alexandre Brouste, Masaaki Fukasawa, Hideitsu Hino, Stefano M. Iacus, Kengo Kamatani, Yuta Koike, Hiroki Masuda, Ryosuke Nomura, Teppei Ogihara, Yasutaka Shimuzu, Masayuki Uchida, Nakahiro Yoshida. (2014). The yuima Project: A Computational Framework for Simulation and Inference of Stochastic Differential Equations. *Journal of Statistical Software*, 57(4). 1–51. http://www.jstatsoft.org/v57/i04/.

Brouste, A. and Stefano, M.I. (2013). Parameter Estimation for the Discretely Observed Fractional Ornstein-Uhlenbeck Process and the yuima R Package. *Computational Statistics*. 28(4), 1129–1147.

Guidoum, A. C. and Boukhetala, K. (2016). Sim.DiffProc: Simulation of Diffusion Processes. R package version 3.2. http://CRAN.R-project.org/package=Sim.DiffProc

Guidoum, A. C. and Boukhetala, K. (2014). Quasi Maximum Likelihood Estimation for One-Dimensional Diffusion Processes. In submission (The R Journal).

R Development Core Team (2015). *R: A Language and Environment for Statistical Computing*. Vienna, Austria. http://www.R-project.org/

Stig, M. and Søren, K. (2013). PSM: Non-Linear Mixed-Effects modelling using Stochastic Differential Equations. *R package version 0.8-10*. http://CRAN.R-project.org/package=PSM

Croissant, Y. (2015). Ecdat: Data Sets for Econometrics. *R package version 0.2-9*. http://CRAN.R-project.org/package=Ecdat

B.L.S. Prakasa Rao. (1999). *Statistical Inference for Diffusion Type Processes*. Arnold, London and Oxford University press, New York.

Florens, D. (1999). Estimation of the diffusion coefficient from the crossings, *Stat. Infer. Stochastic Process*. 1, 175–195.

Kutoyants, Y.A. (2004). *Statistical Inference for Ergodic Diffusion Processes*. Springer, London.

Sørensen, H. (2000). Inference for Diffusion Processes and Stochastic Volatility Models. Ph.D. thesis, Department of Mathematical Sciences, University of Copenhagen.

Sørensen, H. (2002). Estimation of diffusion parameters for discretely observed diffusion processes. *Bernoulli*, 8, 491–508.

Sørensen, H. (2004). Parametric inference for diffusion processes observed at discrete points in time: a survey. *International Statistical Review*, 72, 337–354.

Hurn, A.S. and Lindsay, K.A. (1999). Estimating the parameters of stochastic differential equations. *Mathematics And Computers In Simulation*. 48, 373–384.

Hurn, A.S., Lindsay, K.A. and Martin, V.L. (2003). On the efficacy of simulated maximum likelihood for estimating the parameters of stochastic differential equations. *Journal of Time Series Analysis*. 24(1), 45–63.

Pedersen, A.R. (1995). A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations. *Scand. J. Statist.*, 22(1), 55–71.

Yoshida, N. (1992). Estimation of diffusion processes from discrete observations. *Journal of Multivariate Analysis*. 41, 220–242.

Gallant, A and Long, J. (1997). Estimating stochastic differential equations efficiently by minimum chi-square. *Biometrika*. 84, 124–141.

Alcock, J and Burrage, K. (2004). A genetic estimation algorithm for parameters of stochastic ordinary differential equations. *Computational Statistics & Data Analysis*. 47, 255–275.

Ogihara, T. and Yoshida, N. (2011). Quasi-Likelihood Analysis for the Stochastic Differential Equation with Jumps. *Statistical Inference for Stochastic Processes*. 14(3), 189–229.

Uchida, M. and Yoshida, N. (2005). AIC for ergodic diffusion processes from discrete observations. *preprint MHF 2005-12, March 2005, Faculty of Mathematics, Kyushu University*. Fukuoka, Japan.

Uchida, M. and Yoshida, N. (2012). Adaptive Estimation of an Ergodic Diffusion Process Based on Sampled Data. *Stochastic Processes and their Applications.* 122(8), 2885–2924.

Kloeden, P., Platen, E., Schurz and Sørensen, M. (1996). On Effects of Discretization on Estimators of Drift Parameters for Diffusion Processes. *Journal of Applied Probability.* 33, 1061–1076.