[Calibrating the Ornstein-Uhlenbeck (Vasicek) model](http://www.sitmo.com/article/calibrating-the-ornstein-uhlenbeck-model/)

In this article I’ll describe two methods for calibrating the model parameters of the Ornstein-Uhlenbeck process to a given dataset.

* The least squares regression method
* maximum likelihood method

Introduction

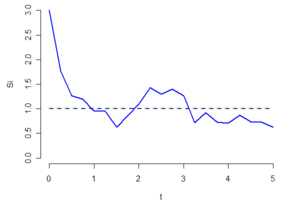
The stochastic differential equation (SDE) for the Ornstein-Uhlenbeck process is given by

\[ dS_t=\lambda(\mu-S_t)dt+\sigma dW_t \]

with \lambda the mean reversion rate, \mu the mean, and \sigma the volatility.

An example simulation

The table and figure below show a simulated scenario for the Ornstein-Uhlenbeck process with time step \delta=0.25, mean reversion rate \lambda=3.0, long term mean \mu=1.0 and a noise term of \sigma= 0.50. We will use this data to explain the model calibration steps.



A scenarios of a the Ornstein-Uhlenbeck process. The scenarios start at S(0)=3 and reverting to a long term mean of 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **i** | **t** | **S_i** | **N_{0,1}** |
| 0 | 0.00 | 3.0000 |  |
| 1 | 0.25 | 1.7600 | -1.0268 |
| 2 | 0.50 | 1.2693 | -0.4985 |
| 3 | 0.75 | 1.1960 | 0.3825 |
| 4 | 1.00 | 0.9468 | -0.8102 |
| 5 | 1.25 | 0.9532 | -0.1206 |
| 6 | 1.50 | 0.6252 | -1.9604 |
| 7 | 1.75 | 0.8604 | 0.2079 |
| 8 | 2.00 | 1.0984 | 0.9134 |
| 9 | 2.25 | 1.4310 | 2.1375 |
| 10 | 2.50 | 1.3019 | 0.5461 |
| 11 | 2.75 | 1.4005 | 1.4335 |
| 12 | 3.00 | 1.2686 | 0.4414 |
| 13 | 3.25 | 0.7147 | -2.2912 |
| 14 | 3.50 | 0.9237 | 0.3249 |
| 15 | 3.75 | 0.7297 | -1.3019 |
| 16 | 4.00 | 0.7105 | -0.8995 |
| 17 | 4.25 | 0.8683 | 0.0281 |
| 18 | 4.50 | 0.7406 | -1.0959 |
| 19 | 4.75 | 0.7314 | -0.8118 |
| 20 | 5.00 | 0.6232 | -1.3890 |

The following simulation equation is used for generating paths (sampled with fixed time steps of \delta=0.25). The equation is an exact solution of the SDE.

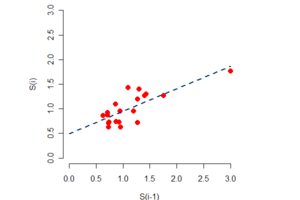
\begin{align*} S_{i+1} &=S_i  e^{-\lambda \delta}+\mu(1-e^{-\lambda \delta}) +\sigma\sqrt{\frac{1-e^{-2\lambda \delta}}{2\lambda}} N_{0,1} \end{align*}

The random numbers used in this example are shown in the last column of the table 1.

Calibration using least squares regression

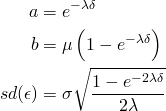
The relationship between consecutive observations S_i,S_{i+1} is linear with a iid normal random term \epsilon

\[ S_{i+1} = a S_i + b + \epsilon \]

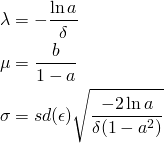


Least square fitting of a line to the data.

The relationship between the linear fit and the model parameters is given by

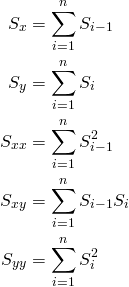


rewriting these equations gives

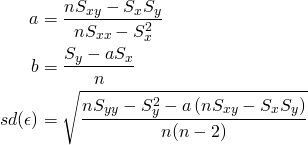


Calculating the least squares regression

Most software tools (Excel, Matlab, R, Octave, Maple, …) have built in functionality for least square regression. If its not available, a least square regression can easily be done by calculating the the quantities below:



from which we get the following parameters of the least square fit



Example

Applying the regression to the data in table 1 we get

|  |  |
| --- | --- |
| **Param** | **Value** |
| S_x | 22.5301 |
| S_y | 20.1534 |
| S_{xx} | 30.8338 |
| S_{xy} | 25.1973 |
| S_{yy} | 22.2222 |
| a | 0.4574 |
| b | 0.4924 |
| sd(\epsilon) | 0.2073 |

These results allow us to recover the model parameters:

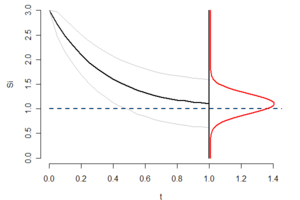
|  |  |
| --- | --- |
| **Param** | **Value** |
| \mu | 0.9075 |
| \lambda | 3.1288 |
| \sigma | 0.5831 |

Calibration using Maximum Likelihood estimates

Conditional probability density function

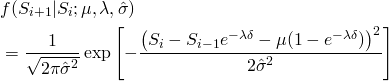
The conditional probability density function is easily derived by combining the simulation equation above with the probability density function of the normal distribution function:

\[ P(N_{0,1}=x) =\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2} \]



Conditional probability density function -red- of S at t=1.

The equation of the conditional probability density of an observation S_{i+1} given a previous observation S_i(with a \delta time step between them) is given by

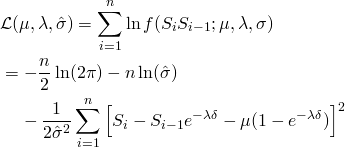


with

\[ \hat{\sigma}^2 =\sigma^2\frac{1-e^{-2\lambda\delta}}{2\lambda} \]

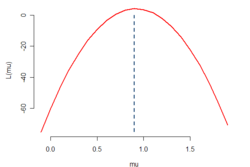
Log-likelihood function

The log-likelihood function of a set of observation (S_0, S_1, \ldots, S_n) can be derived from the conditional density function

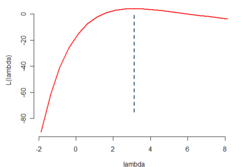


Maximum likelihood conditions

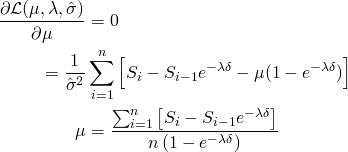
The maximum of this log-likelihood surface can be found at the location where all the partial derivatives are zero. This leads to the following set of constraints.

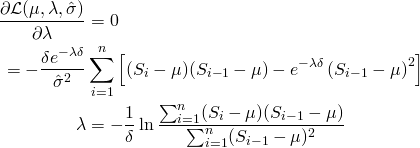


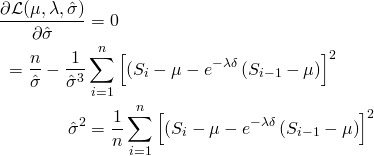
Log likelihood function as function of mu.



Log likelihood function as function of lambda.



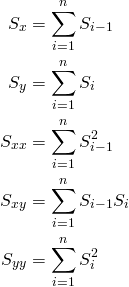




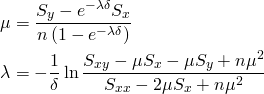
Solution of the conditions

The problem with these conditions is that the solutions depend on each other. However, both \lambda and \mu are independent of \sigma, and knowing either \lambda or \mu will directly give the value the other.  
The solution of \sigma can be found once both \lambda and \mu are determined. To solve these equations it is thus sufficient to find either \lambda or \mu.  
Finding \mu can be done by substituting the \lambda condition into the \mu.

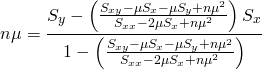
First we change notation of the \mu and \lambda condition using the same notation as before, i.e



which gives us:



substituting \lambda into \mu gives



removing denominators

\begin{align*} n\mu =\frac{S_y(S_{xx} -2\mu S_x + n\mu^2)-(S_{xy}-\mu S_x-\mu S_y + n\mu^2)S_x}{(S_{xx} -2\mu S_x + n\mu^2)-(S_{xy}-\mu S_x-\mu S_y + n\mu^2)} \end{align*}

collecting terms

\begin{align*} n\mu =\frac{(S_y S_{xx} - S_x S_{xy}) + \mu(S_x^2 - S_x S_y) + \mu^2 n(S_y-S_x)}{(S_{xx} - S_{xy}) + \mu(S_y-S_x)} \end{align*}

moving all \mu to the left

\[ n\mu(S_{xx}-S_{xy}) - \mu(S_x^2 - S_x S_y) =S_y S_{xx} - S_x S_{xy} \]

Final results: The maximum likelihood equations

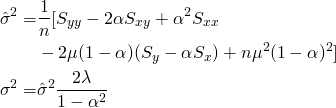
mean:

\[ \mu = \frac{S_y S_{xx} - S_x S_{xy}}{n(S_{xx}-S_{xy}) - (S_x^2 - S_x S_y)} \]

mean reversion rate:

\begin{align*} \lambda = -\frac{1}{\delta}\ln \frac{S_{xy}-\mu S_x-\mu S_y + n\mu^2}{S_{xx} -2\mu S_x + n\mu^2} \end{align*}

variance:



with \alpha =e^{-\lambda\delta}

Example

Calculating the sums based on table 1 we get

|  |  |
| --- | --- |
| **Param** | **Value** |
| S_x | 22.5301 |
| S_y | 20.1534 |
| S_{xx} | 30.8338 |
| S_{xy} | 25.1973 |
| S_{yy} | 22.2222 |

These results allow us to recover the model parameters:

|  |  |
| --- | --- |
| **Param** | **Value** |
| \mu | 0.9075 |
| \lambda | 3.1288 |
| \sigma | 0.5532 |

Matlab Code

Least Squares Calibration

|  |
| --- |
| function [mu,sigma,lambda] = OU\_Calibrate\_LS(S,delta)  n = length(S)-1;    Sx = sum( S(1:end-1) );  Sy = sum( S(2:end) );  Sxx = sum( S(1:end-1).^2 );  Sxy = sum( S(1:end-1).\*S(2:end) );  Syy = sum( S(2:end).^2 );    a = ( n\*Sxy - Sx\*Sy ) / ( n\*Sxx -Sx^2 );  b = ( Sy - a\*Sx ) / n;  sd = sqrt( (n\*Syy - Sy^2 - a\*(n\*Sxy - Sx\*Sy) )/n/(n-2) );    lambda = -log(a)/delta;  mu = b/(1-a);  sigma = sd \* sqrt( -2\*log(a)/delta/(1-a^2) );  end |

Maximum Likelyhood Calibration

|  |
| --- |
| function [mu,sigma,lambda] = OU\_Calibrate\_ML(S,delta)  n = length(S)-1;    Sx = sum( S(1:end-1) );  Sy = sum( S(2:end) );  Sxx = sum( S(1:end-1).^2 );  Sxy = sum( S(1:end-1).\*S(2:end) );  Syy = sum( S(2:end).^2 );    mu = (Sy\*Sxx - Sx\*Sxy) / ( n\*(Sxx - Sxy) - (Sx^2 - Sx\*Sy) );  lambda = -log( (Sxy - mu\*Sx - mu\*Sy + n\*mu^2) / (Sxx -2\*mu\*Sx + n\*mu^2) ) / delta;  a = exp(-lambda\*delta);  sigmah2 = (Syy - 2\*a\*Sxy + a^2\*Sxx - 2\*mu\*(1-a)\*(Sy - a\*Sx) + n\*mu^2\*(1-a)^2)/n;  sigma = sqrt(sigmah2\*2\*lambda/(1-a^2));  end |

Example Usage

|  |
| --- |
| S = [ 3.0000 1.7600 1.2693 1.1960 0.9468 0.9532 0.6252 ...  0.8604 1.0984 1.4310 1.3019 1.4005 1.2686 0.7147 ...  0.9237 0.7297 0.7105 0.8683 0.7406 0.7314 0.6232 ];    delta = 0.25;    [mu1, sigma1, lambda1] = OU\_Calibrate\_LS(S,delta)  [mu2, sigma2, lambda2] = OU\_Calibrate\_ML(S,delta) |

gives

|  |
| --- |
| mu1 = 0.90748788828331  sigma1 = 0.58307607458526  lambda1 = 3.12873217812387    mu2 = 0.90748788828331  sigma2 = 0.55315453345189  lambda2 = 3.12873217812386 |