Parameter estimation and empirical analysis of fractional O-U process based on three machine learning algorithms

Yicun Li[1] and Yuanyang Teng[1]

[1]Zhejiang University

March 07, 2024

Abstract

Abstract Fractional O-U process is a very classical stochastic process which is used to describe the time series of financial volatility. Three parameters need to be estimated in this process, and the estimation method based on discrete observations can be realized by machine learning optimization algorithm. In this study, the parameter estimation method of fractional O-U process is briefly described, and three optimization algorithms, Newton method, quasi Newton method and genetic algorithm, are used to estimate the parameters. The comparison shows that genetic algorithm is relatively accurate and efficient. Finally, the minute data of stock index futures are estimated based on fractional O-U process. The results show that the estimation of theta and Hurst index is relatively accurate, and the estimation error of volatility is large.

Parameter estimation and empirical analysis of fractional O-U process based on three machine learning algorithms

Yicun Li*

(School of Management, Zhejiang University, Hangzhou, P. R. China., Email: 303984464@qq.com)

Yuanyang Teng†

(School of Management, Zhejiang University, Hangzhou, P. R. China., Email: 409220153@qq.com)

Abstract

Fractional O-U process is a very classical stochastic process which is used to describe the time series of financial volatility. Three parameters need to be estimated in this process, and the estimation method based on discrete observations can be realized by machine learning optimization algorithm. In this study, the parameter estimation method of fractional O-U process is briefly described, and three optimization algorithms, Newton method, quasi Newton method and genetic algorithm, are used to estimate the parameters. The comparison shows that genetic algorithm is relatively accurate and efficient. Finally, the minute data of stock index futures are estimated based on fractional O-U process. The results show that the estimation of theta and Hurst index is relatively accurate, and the estimation error of volatility is large.

Keywords

Fractional O-U process, parameter estimation, Newton method, quasi Newton method, genetic algorithm, real volatility

JEL classification

G12

1. Introduction

Many mechanisms or dynamical systems in nature and social life obey fractional O-U process, so it is of positive significance to study fractional O-U process for understanding these mechanisms. In view of the importance of fractional O-U process, how to estimate its parameters is a very important problem. El Mehdi Haress, Yao Zhong Hu [1] used Newton method to estimate the parameters of fractional O-U, and proved the convergence of the estimator. Ana prior, Marina kleptsyna, Paula milheiro Oliveira [2] for 2-D O-U process, they use the polar nature estimation method to estimate the drift term of random process. Yao Zhong Hu, David nualart, Hong Jun Zhou [3] used a least square estimator to estimate the drift term of O-U process, proved the convergence of $H \in (0,0.75)$ by using central limit theorem, and proved the convergence of $H \in (0.75,1)$ by using a non central limit theorem. Yao Zhong Hu, David nualart [4] also used a least square estimator to estimate the drift term of fractional O-U process. Weilin Xiao, Weiguo Zhang, Weidong Xu [5] estimate the drift term and diffusion term for discrete fractional O-U process on the premise that Hurst parameter is known. Istas and Lang [6] estimates three parameters of O-U process by quadratic variation methods. Kleptsyna and Le Breton [7] use MLE to estimate parameters for H $\in$ (0.5,1). Tudor and Viens [8] have also obtained the almost sure convergence of both the MLE and a version of the MLE using discrete observations for all H$\in$ (0,1). Irrespectively of the stability of the process, the MLE is known to be unbiased and consistent, by Basak and Lee [9] and Lin and Lototsky [10]. Proofs of asymptotic normality require certain regularity and ergodicity conditions under which the estimator is asymptotically normal with the rate of convergence $\sqrt{T}$, by Rao[11], Kutoyants[12].

At present, most of the literatures estimate the parameters of fractional O-U process based on mathematical methods and require some conditions for convergence. This paper continues the Newton method of El Mehdi Haress, Yaozhong Hu [1], further adopts the quasi Newton method, and creatively adds the swarm particle artificial intelligence algorithm such as genetic algorithm to estimate the parameters, and comprehensively compares the efficiency of Newton method, quasi Newton method and genetic algorithm in estimating the parameters.

The second part of this paper introduces the parameter estimation theory of fractional O-U process, proves the mathematical theory of Newton method and quasi Newton method, and introduces the algorithm flow of genetic algorithm. In the third part of the paper, the fractional O-U process is used to generate random simulation data under specific parameters, and then the above three methods are used to estimate the given parameters, and analysis the error of estimation. The fourth part is an empirical study, which estimates the fractional O-U process of real volatility series based on stock index futures data, and observes the estimation error of each quantile in the series. The fifth part is the conclusion, which summarizes the contribution of this paper.

2. Model and algorithm

2.1 parameter estimation method of fractional O-U process

A fractional O-U process $\left(X_t\right)_{t>0}$, describe the process of the following form:

$$\mathrm{d}X_t = -\theta X_t dt + \sigma dB_t^H$$

Among three parameters $\theta > 0$, for the sake of simplicity, set $\mathrm{X}_0 = 0$. Then discretize $\left(X_t\right)_{t>0}$, set h as time interval, we get $\left\{X_{\mathrm{h}}, X_{2h}, X_{3h} \dots X_{nh}\right\}$. Finally, three unknown parameters θ, σ, H are estimated by the discretized series $\left\{X_{\mathrm{h}}, X_{2h}, X_{3h} \dots X_{nh}\right\}$.

El Mehdi Haress, Yaozhong Hu [1], presents a method for estimating three unknown parameters by discrete data.

First, the following estimation formula is constructed:

$$\begin{cases} \mathrm{f}_1(\sigma,\theta,H) := \dfrac{1}{2\pi}\sigma^2\Gamma(2H+1)\sin(\pi H)\displaystyle\int_0^\infty \dfrac{x^{1-2H}}{\theta^2+x^2}dx; \\ \mathrm{f}_2(\sigma,\theta,H) := \dfrac{1}{2\pi}\sigma^2\Gamma(2H+1)\sin(\pi H)\displaystyle\int_0^\infty \cos(hx)\dfrac{x^{1-2H}}{\theta^2+x^2}dx; \\ \mathrm{f}_3(\sigma,\theta,H) := \dfrac{1}{2\pi}\sigma^2\Gamma(2H+1)\sin(\pi H)\displaystyle\int_0^\infty \cos(2hx)\dfrac{x^{1-2H}}{\theta^2+x^2}dx; \end{cases} \cdots(1)$$

Equation (1) can be written as $\mathrm{f}(\sigma,\theta,H)=(\mathrm{f}_1(\sigma,\theta,H),\mathrm{f}_2(\sigma,\theta,H),\mathrm{f}_3(\sigma,\theta,H))^T$. And $\mathrm{f}_1,\mathrm{f}_2,\mathrm{f}_3$ can be approximated by formula (2).

$$\begin{cases} \mathrm{f}_1(\sigma,\theta,H)=\eta_\mathrm{n} = \dfrac{1}{n}\displaystyle\sum_{k=1}^{n} X_{kh}^2, \\ \mathrm{f}_2(\sigma,\theta,H)=\eta_{h,n} = \dfrac{1}{n}\displaystyle\sum_{k=1}^{n} X_{kh}X_{kh+h}, \\ \mathrm{f}_3(\sigma,\theta,H)=\eta_{2h,n} = \dfrac{1}{n}\displaystyle\sum_{k=1}^{n} X_{2kh}X_{2kh+2h}, \end{cases} \dots(2)$$

Therefore, the parameters to be estimated can be solved by equation (3)

$$\left(\tilde{\theta},\tilde{\mathrm{H}},\tilde{\sigma}\right) = f^{-1}\left(\eta_n,\eta_{h,n},\eta_{2h,n}\right)\dots(3)$$

2.2 Newton method

Propose nonlinear multivariate equations (4):

$$\begin{cases} f_1(x_1,x_2\dots x_n)=0 \\ f_2(x_1,x_2\dots x_n)=0 \\ \dots\dots\dots\dots\dots\dots\dots \\ f_n(x_1,x_2\dots x_n)=0 \end{cases} \dots(4)$$

Set:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathrm{F}(\boldsymbol{x}) = \begin{cases} f_1(x_1,x_2\dots x_n) \\ f_2(x_1,x_2\dots x_n) \\ \dots\dots\dots\dots\dots\dots \\ f_n(x_1,x_2\dots x_n) \end{cases} = 0$$

The Newton method is used to solve the equation as follows:

Firstly, Taylor expansion of $\mathrm{F}(\boldsymbol{x})$ is carried out at $\boldsymbol{x}^k$ point, which k represents k times iteration of $\boldsymbol{x}^k$, and the expansion ignores the high order terms:

$$\begin{cases} f_1(x_1,x_2\dots x_n)\approx f_1(x_1^k,x_2^k\dots x_n^k)+\frac{\partial f_1}{\partial x_1}(x_1-x_1^k)+\frac{\partial f_1}{\partial x_2}(x_2-x_2^k)\dots+\frac{\partial f_1}{\partial x_n}(x_n-x_n^k)=l_1(x_1,x_2\dots x_n) \\ f_2(x_1,x_2\dots x_n)\approx f_2(x_1^k,x_2^k\dots x_n^k)+\frac{\partial f_2}{\partial x_1}(x_1-x_1^k)+\frac{\partial f_2}{\partial x_2}(x_2-x_2^k)\dots+\frac{\partial f_2}{\partial x_n}(x_n-x_n^k)=l_2(x_1,x_2\dots x_n) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ f_n(x_1,x_2\dots x_n)\approx f_n(x_1^k,x_2^k\dots x_n^k)+\frac{\partial f_n}{\partial x_1}(x_1-x_1^k)+\frac{\partial f_n}{\partial x_2}(x_2-x_2^k)\dots+\frac{\partial f_n}{\partial x_n}(x_n-x_n^k)=l_n(x_1,x_2\dots x_n) \end{cases} \dots(5)$$

Set $\mathrm{L}(\boldsymbol{x})=\begin{bmatrix} l_1(x_1,x_2,\dots x_n) \\ l_2(x_1,x_2,\dots x_n) \\ \dots \\ l_n(x_1,x_2,\dots x_n) \end{bmatrix}$, $\mathrm{L}(\boldsymbol{x})$ is Taylor expansion of $\mathrm{F}(\boldsymbol{x})$, so the solution of $\mathrm{L}(\boldsymbol{x})$ is also the solution of $\mathrm{F}(\boldsymbol{x})$。

Then, set $\mathrm{L}(\boldsymbol{x}^{\mathrm{k+1}})=0$, according to equation (5), we can get:

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1},\frac{\partial f_1}{\partial x_2},\dots\frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1},\frac{\partial f_2}{\partial x_2},\dots\frac{\partial f_2}{\partial x_n} \\ \dots\dots\dots\dots \\ \frac{\partial f_n}{\partial x_1},\frac{\partial f_n}{\partial x_2},\dots\frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \mathrm{x}_1^{k+1}-x_1^k \\ \mathrm{x}_2^{k+1}-x_2^k \\ \dots\dots \\ \mathrm{x}_n^{k+1}-x_n^k \end{bmatrix} = -\begin{bmatrix} f_1(x_1^k,x_2^k\dots x_n^k) \\ f_2(x_1^k,x_2^k\dots x_n^k) \\ \dots\dots\dots \\ f_n(x_1^k,x_2^k\dots x_n^k) \end{bmatrix} \dots(6)$$

According to equation (6), $\boldsymbol{x}^{\mathrm{k+1}}$ can be solved as follows:

$$\begin{bmatrix} \mathrm{x}_1^{k+1} \\ \mathrm{x}_2^{k+1} \\ \dots \\ \mathrm{x}_\mathrm{n}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathrm{x}_1^{k} \\ \mathrm{x}_2^{k} \\ \dots \\ \mathrm{x}_\mathrm{n}^{k} \end{bmatrix} - \begin{bmatrix} \frac{\partial f_1}{\partial x_1},\frac{\partial f_1}{\partial x_2},\dots\frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1},\frac{\partial f_2}{\partial x_2},\dots\frac{\partial f_2}{\partial x_n} \\ \dots\dots\dots\dots \\ \frac{\partial f_n}{\partial x_1},\frac{\partial f_n}{\partial x_2},\dots\frac{\partial f_n}{\partial x_n} \end{bmatrix}^{-1} \begin{bmatrix} f_1(x_1^k,x_2^k\dots x_n^k) \\ f_2(x_1^k,x_2^k\dots x_n^k) \\ \dots\dots\dots \\ f_n(x_1^k,x_2^k\dots x_n^k) \end{bmatrix} \dots(7)$$

2.3 quasi Newton method

Although Newton method can approximate the solution of the equations, it is necessary to calculate the inverse matrix of Jacobian matrix repeatedly in the process of solving, which makes the calculation process very resource consuming. Therefore, a new quasi Newton method is proposed.

suppose K + 1 iteration is performed. The steps of quasi Newton method are as follows:

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^{k+1})(\boldsymbol{x} - \boldsymbol{x}^{k+1}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{k+1})^T \nabla^2 f(\boldsymbol{x}^{k+1})(\boldsymbol{x} - \boldsymbol{x}^{k+1}) \ldots (8)$$

Add a gradient operator ∇ to the left and right of equation (8)

$$\nabla f(\boldsymbol{x}) \approx \nabla f(\boldsymbol{x}^k) + \mathrm{H}_{k+1}(\boldsymbol{x} - \boldsymbol{x}^{k+1}) \ldots (9)$$

Where H_{k+1} is the Hessian matrix with iteration of K + 1.

In formula (9), set $\boldsymbol{x}=\boldsymbol{x}^k$,the following is obtained:

$$\nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^{\mathrm{k}}) \approx \mathrm{H}_{k+1}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \ldots (10)$$

set $\boldsymbol{y}_k = \nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^{\mathrm{k}})$, $\boldsymbol{s}_k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$

Then equation (10) can be rewritten as follows:

$$\boldsymbol{s}_k \approx \mathrm{H}_{k+1}^{-1} \boldsymbol{y}_k \ldots (11)$$

Equation (11) is quasi Newton condition, which constrains the Hessian matrix in the iterative process. Then, use D_{k+1} approximately substitute for the Hessian matrix, obtain equation (12) as follows:

$$\boldsymbol{s}_k = \mathrm{D}_{k+1} \boldsymbol{y}_k \ldots (12)$$

So far, formula (12) is the core iterative formula of quasi Newton method. The famous DFP Algorithm is introduced to solve equation (12). In equation (12), D_{k+1} can be disassembled into:

$$\mathrm{D}_{k+1} = \mathrm{D}_k + \Delta \mathrm{D}_k \ldots (13)$$

D_0 is set as an identity matrix at the beginning of iteration, so the key point is how to construct the $\Delta \mathrm{D}_k$ of each step.

The conjecture of DFP Algorithm is that $\Delta \mathrm{D}_k$ may be related to $\boldsymbol{s}_k$ 、 $\boldsymbol{y}_k$ 、 D_k , so the "undetermined method" is used to change the undetermined $\Delta \mathrm{D}_k$ into a special form, and then the quasi Newton condition is used to solve the problem.

The DFP set $\Delta \mathrm{D}_k$ to undetermined form

$$\Delta \mathrm{D}_k = \alpha \boldsymbol{u}\boldsymbol{u}^T + \beta \boldsymbol{v}\boldsymbol{v}^T \ldots (14)$$

In equation (14) α 、 β are undetermined coefficients, and $\boldsymbol{u}$ 、 $\boldsymbol{v}$ are undetermined vectors. Setting this form guarantees symmetry of $\Delta \mathrm{D}_k$ (because sum of $\boldsymbol{u}\boldsymbol{u}^T$ and $\boldsymbol{v}\boldsymbol{v}^T$ is a symmetric matrix).

Substituting formula (14) into formula (13) and combining with formula (12), we can obtain the following formula:

$$\boldsymbol{s}_k = \mathrm{D}_k \boldsymbol{y}_k + \alpha \boldsymbol{u}\boldsymbol{u}^T \boldsymbol{y}_k + \beta \boldsymbol{v}\boldsymbol{v}^T \boldsymbol{y}_k \ldots (15)$$

The transformation form is as follows:

$$\boldsymbol{s}_k = \mathrm{D}_k \boldsymbol{y}_k + \left(\alpha \boldsymbol{u}^T \boldsymbol{y}_k\right)\boldsymbol{u} + \left(\beta \boldsymbol{v}^T \boldsymbol{y}_k\right)\boldsymbol{v} \ldots (16)$$

In equation (16), $\alpha \boldsymbol{u}^T \boldsymbol{y}_k$ and $\beta \boldsymbol{v}^T \boldsymbol{y}_k$ are two numbers, so we set $\alpha \boldsymbol{u}^T \boldsymbol{y}_k = 1$, $\beta \boldsymbol{v}^T \boldsymbol{y}_k = -1$. Substitute it into formula (16), we get that:

$$\boldsymbol{u} - \boldsymbol{v} = \boldsymbol{s}_k - \mathrm{D}_k \boldsymbol{y}_k \ldots (17)$$

If you want to make the above formula meets, you can take $\boldsymbol{u} = \boldsymbol{s}_k$ and $\boldsymbol{v} = \mathrm{D}_k \boldsymbol{y}_k$ then combine $\alpha \boldsymbol{u}^T \boldsymbol{y}_k = 1$ and $\beta \boldsymbol{v}^T \boldsymbol{y}_k = -1$ to get:

$$\alpha = \frac{1}{\mathrm{s}_k^T y_k}, \beta = -\frac{1}{(\mathrm{D}_k \boldsymbol{y}_k)^T \boldsymbol{y}_k} = -\frac{1}{\boldsymbol{y}_k^T \mathrm{D}_k \boldsymbol{y}_k}$$

substitute α 、 β and $\boldsymbol{u} = \boldsymbol{s}_k$, $\boldsymbol{v} = \mathrm{D}_k \boldsymbol{y}_k$ into formula (14), we can get:

$$\Delta \mathrm{D}_k = \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{s}_k^T \boldsymbol{y}_k} - \frac{\mathrm{D}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathrm{D}_k}{\boldsymbol{y}_k^T \mathrm{D}_k \boldsymbol{y}_k} \ldots (18)$$

So far, equation (18) is the iterative increment of formula (13) in DFP Algorithm, and then substitute D_{k+1} into formula (12), the solution of quasi Newton method can be obtained.

2.4 genetic algorithm

Genetic algorithm (GA) follows the principle of "survival of the fittest". It is a kind of randomized search algorithm which draws lessons from natural selection and natural genetic mechanism in biological world. Genetic algorithm simulates the evolution process of an artificial population. Through the selection, crossover and mutation mechanisms, a group of candidate individuals are retained in each iteration, and the process is repeated. After several generations of evolution, the fitness of the population reaches the "approximate optimal" state under ideal conditions.

The main steps of the algorithm are as follows:

1. Initialization population: initialization population is to generate a series of solutions randomly for target problems.

2. Calculate the given fitness: substitute the randomly generated solution into the target problem, calculate the result of the problem, and then evaluate the result of the solution based on some criteria suitable for the target problem.

3. Ranking according to individual fitness: ranking the evaluation results of solutions, the

solutions with high fitness rank first, and the solutions with poor fitness rank second.

4. Selection operation: select the superior individuals and eliminate some poor individuals.

5. Crossover operation: gene crossover operation is carried out on the selected dominant individuals, that is to say, the dominant solutions are fused in accordance with the target problem.

6. Mutation operation: to a certain extent, the combined dominant solution is randomly perturbed, so that the perturbed combined solution has a certain probability to deviate from the original dominant solution, but still can roughly retain the performance of the dominant solution.

7. Judge whether the iteration of the dominant solution reaches the convergence condition. If convergence condition meets, stop iteration, otherwise return back to step 2 and loop step 2-7.

Through the above eight steps of the cycle, the final randomly generated solution will reach the convergence condition after several rounds of iteration, that is, the fitness of the dominant solution will not change, which means that the solution at this time at least reaches the local optimal. In this way, the objective of the optimization problem is approximately obtained.

3. Simulation

The simulation is to give the parameters of fractional O-U, randomly generate the sequence, and then calculate η_{n}、$\eta_{h,n}$、$\eta_{2h,n}$, and estimate the three unknown parameters θ,σ,H according to the steps in the parameter estimation method of 2.1 fractional O-U process, by using 2.2 Newton method, 2.3 quasi Newton method, and 2.4 genetic algorithm according to equations (1), (2), (3).

El Mehdi Haress, Yao Zhong Hu [1] used Newton's method to test and simulate two groups of parameters $\theta=6,\sigma=2,H=0.7$ and $\theta=6,\sigma=2,H=0.4$. Therefore, this study also uses three different algorithms to simulate these two groups of parameters for comparison.

First, 100 groups of random fractional O-U processes based on parameters $\theta=6,\sigma=2,H=0.7$ are randomly generated. As shown in Figure 1.

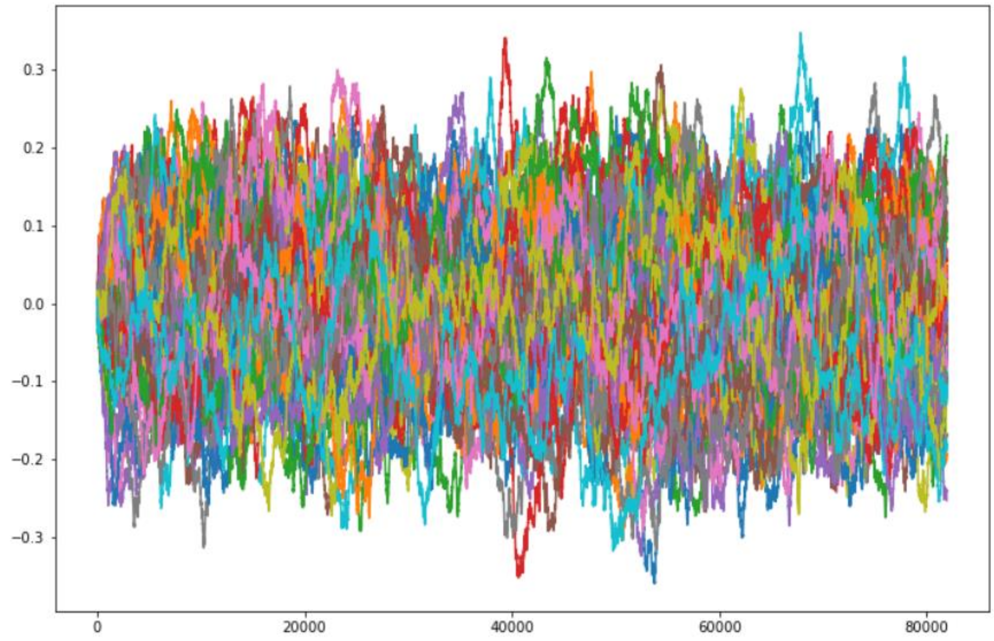

Figure 1 100 groups of the fractional U-O process based on parameters $\theta=6,\sigma=2,H=0.7$

From 100 groups of the fractional U-O process based on parameters $\theta=6,\sigma=2,H=0.7$, we get 100 groups of η_{n}、$\eta_{h,n}$、$\eta_{2h,n}$, which are used to estimate θ,σ,H. Statistics of η_{n}、$\eta_{h,n}$、$\eta_{2h,n}$ showed in Table 1.

Table 1 η_{n}、$\eta_{h,n}$、$\eta_{2h,n}$ calculated from parameters $\theta=6,\sigma=2,H=0.7$

	η_{n}	$\eta_{h,n}$	$\eta_{2h,n}$
mean	0.003958	0.003955	0.00395
std	0.001196	0.001196	0.001196

According to the parameter estimation method based on 2.1 fractional O-U process, we use η_{n}、$\eta_{h,n}$、$\eta_{2h,n}$ in Table 1, Newton method, quasi Newton method and genetic algorithm are used to estimate the fractional U-O process parameters, and the results are shown in Table 2.

Table 2 The parameter estimation based on $\theta=6,\sigma=2,H=0.7$

H=0.7,theta=6,sigma=2		
Panel A: parameter estimation of Newton		
	mean	std
theta	3.783848	21.9009
H	0.7412	0.2726
sigma	0.6778	0.9445
Panel B: parameter estimation of qusi-Newton		
	mean	std
theta	7.6179	1.88
H	0.9095	0.0125
sigma	1.5069	2.5492
Panel C: parameter estimation of GA		
	mean	std
theta	7.0234	1.6905
H	0.5884	0.0972
sigma	1.3647	1.1949

From the estimation results in Table 2, the performance of the three algorithms is: genetic algorithm > quasi Newton method > Newton method. Based on the theoretical analysis of the algorithm, Newton's method and quasi Newton's method are based on the usage of gradient to approximate the local optimum. However, the structure of the equations needed to solve the optimization problem in this study is more complex, which may be approximated by gradient or similar way, and may fall into local optimum or doesn't converge. In fact, in Newton's method, there are many final results that do not converge. The genetic algorithm is a kind of particle swarm algorithm, its crossover and mutation operation is conducive to the optimization of the dominant population, at the same time, it also has a certain probability to jump out of the local optimal, so it is close to the global optimal, so from the results, the performance of genetic algorithm is the best.

Next, make a group of parameter $\theta=6, \sigma=2, H=0.4$, $H<0.5$ means that the 100 groups of random O-U process sequences have anti historical memory, as shown in Figure 2.

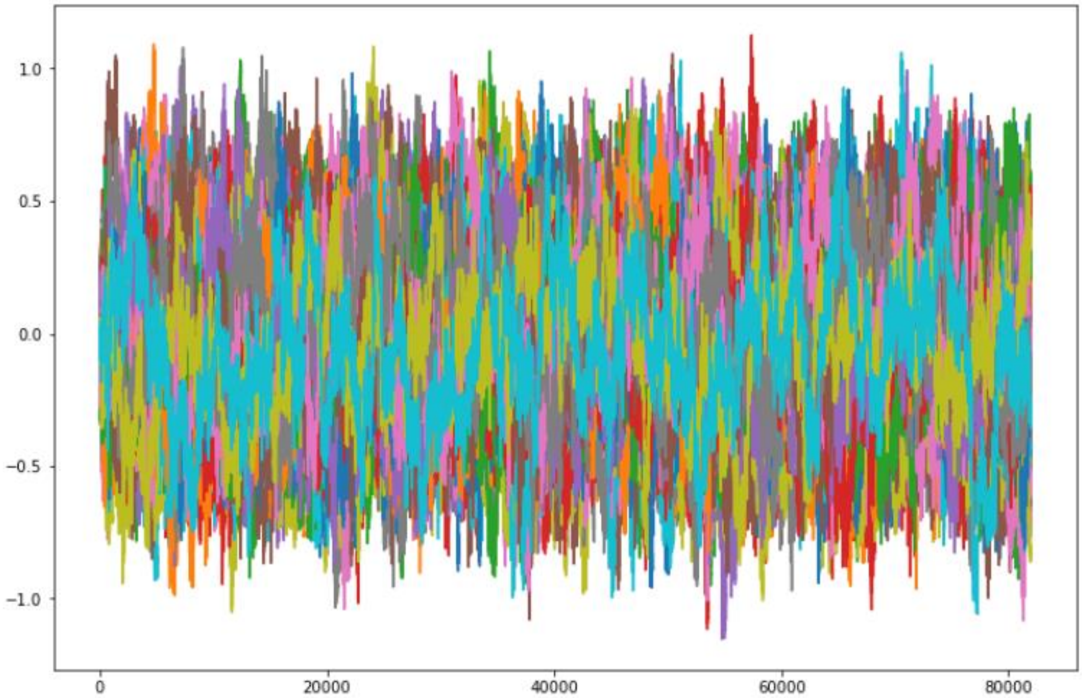

Figure 2 100 groups of the fractional U-O process based on parameters $\theta=6, \sigma=2, H=0.4$

From 100 groups of the fractional U-O process based on parameters $\theta=6, \sigma=2, H=0.4$, we get 100 groups of η_{n} 、$\eta_{h,n}$ 、$\eta_{2h,n}$ which are used to estimate θ, σ, H . Statistics of η_{n} 、$\eta_{h,n}$ 、 $\eta_{2h,n}$ showed in Table 3

Table 3 η_{n} 、 $\eta_{h,n}$ 、 $\eta_{2h,n}$ calculated from parameters $\theta=6, \sigma=2, H=0.4$

	η_{n}	$\eta_{h,n}$	$\eta_{2h,n}$
mean	0.0351	0.0344	0.0338
std	0.0047	0.0047	0.0047

Then, according to 100 groups of η_{n} 、 $\eta_{h,n}$ 、 $\eta_{2h,n}$ in Table 3, Newton method, quasi Newton method and genetic algorithm are used to estimate again, and the results are shown in Table 4.

Table 4 The parameter estimation based on $\theta=6, \sigma=2, H=0.4$

H=0.4,theta=6,sigma=2		
Panel A: parameter estimation of Newton		
	mean	std

theta	Nan	Nan
H	Nan	Nan
sigma	Nan	Nan
Panel B: parameter estimation of qusi-Newton		
	mean	std
theta	6.712858	2.25364
H	0.779601	0.020399
sigma	2.029061	2.754953
Panel C: parameter estimation of GA		
	mean	std
theta	6.746417	2.129206
H	0.601235	0.09987
sigma	3.228041	1.267521

From the results in Table 4, the performance of the algorithm is still genetic algorithm > quasi Newton method > Newton method. The reason why Newton method is Nan is that it is not easy to converge to the real value through the approximation of first-order Taylor expansion, and the final error is large. Although the H estimation of genetic algorithm is greater than 0.5, compared with other algorithms, it still has the smallest error and the fastest running speed.

To sum up, three algorithms are used to estimate the O-U process of the above two groups of parameters. From the results, genetic algorithm is a relatively better parameter estimation algorithm, so the following empirical research is carried out with genetic algorithm.

4. Empirical research

4.1 parameter estimation and prediction of fractional O-U process

This empirical study uses the RV constructed by the minute data of the closing price of Shanghai stock index. The length of time is from November 21, 2016 to November 20, 2019. The data characteristics of the whole dataset are shown in Table 5. The formula of RV is (19).

$$\mathrm{rv}_t = \log\left(\sum_{i=1}^{M} \log\left(\frac{P_{i\Delta}}{P_{(i-1)\Delta}}\right)\right), \quad \mathrm{t} = 1,2,3\ldots\mathrm{n}\ldots(19)$$

Where Δ is the time interval, which is 1 minute in this study, and M is 10 minutes, which represents the RV calculated every 10 minutes.

Table 5 statistical characteristics of stock index and logarithmic RV

	No. of obs	mean	std	skewness	kurtosis
index	176534	3046.46789	251.6178511	-0.408650443	-0.75881
log(rv) of index	17652	-1.39E+01	1.13E+00	0.902773474	2.027883

The main research object of fractional O-U process is volatility, so this study focuses on logarithmic RV. Figure 3 is a linear graph of time series of RV, from which we can see that the fluctuation of RV has certain resilience.

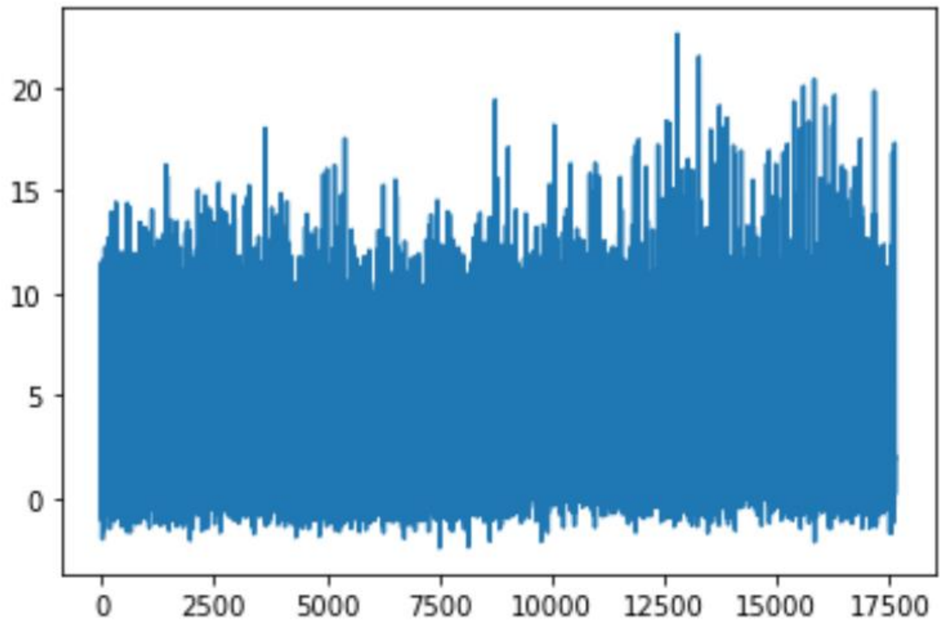

Figure 3 logarithmic RV linear graph

Figure 4 is the frequency distribution of RV. It can be seen from the figure that the distribution of RV is basically bell shaped in small fluctuation areas, but has a positive long tail.

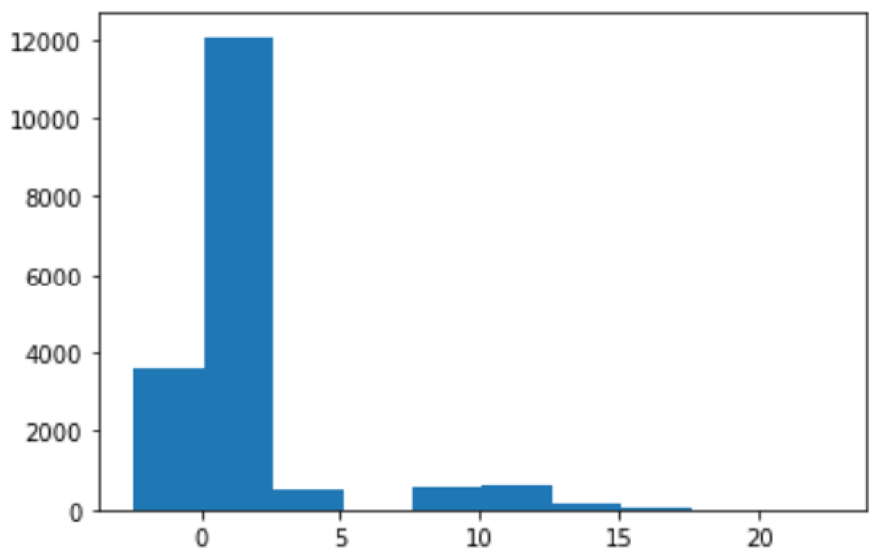

Figure 4 histogram of logarithmic RV

Then the data set is divided into training set and test set. The training set is from November 21, 2016 to December 28, 2018, and the test set is from January 2, 2019 to November 20, 2019. The parameters of fractional O-U process are estimated with training set data, and the results are shown in Table 6.

Table 6 parameters of fractional O-U process estimated by genetic algorithm in training set

GA	theta	H	sigma

mean	6.744176	0.581836414	3.266967513
std	2.095349	0.114964958	1.177182794

Then, 100 groups of 50 000 samples of fractional O-U process random sequences are randomly generated with the parameters in Table 6. The generated fractional O-U process random sequences are shown in Figure 5.

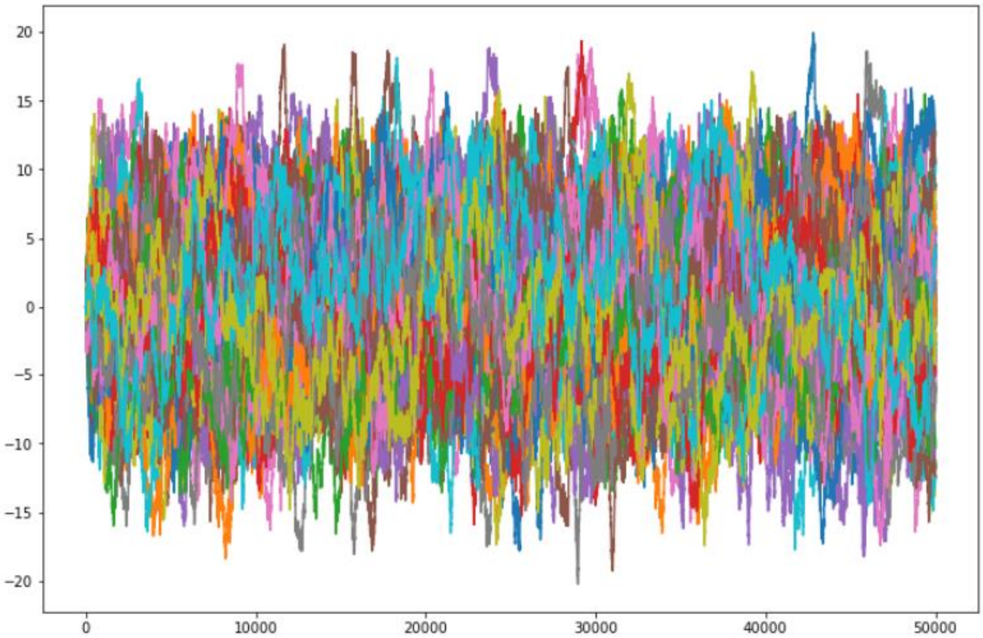

Figure 5 random sequence of fractional O-U process generated by training set estimation parameters

Take the test set from January 2, 2019 to November 20, 2019 to calculate RV, and normalize the initial value to 0, as shown in Figure 6.

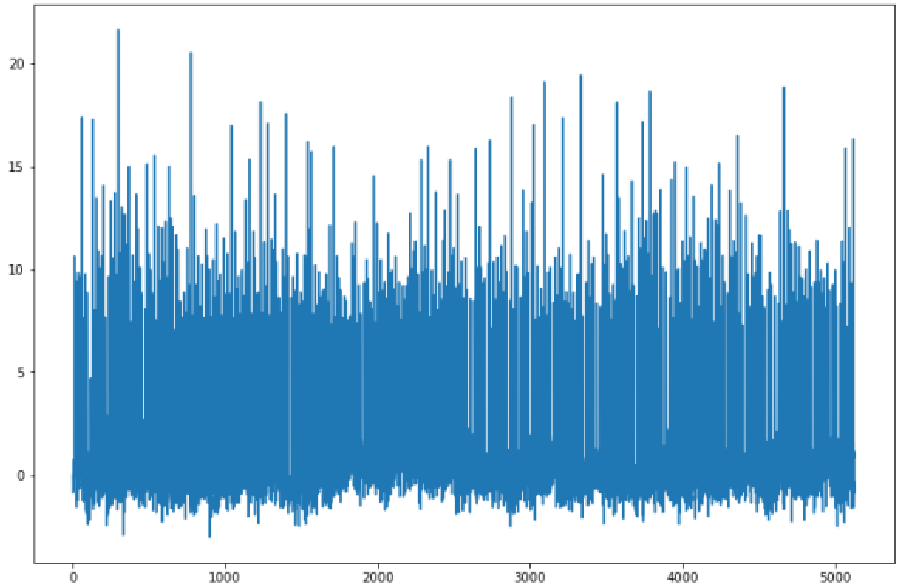

Figure 6 RV sequence of test set calculation

Comparing Figure 5 and Figure 6, the structure of fractional O-U process assumes a mean symmetric sequence, but in reality, the RV of stock index futures tends to be longer in positive tail, shorter in negative tail and asymmetric in reality. So next, we examine the prediction error of each quantile in the test set.

Table 7 prediction of fractional O-U process in each quantile of stock index futures test set

	Test set	Fractional O-U process forecast	Absolute error percentage
0%quantile	-3.0187602	-23.01062168	6.622540431
10%quantile	-1.0088317	-6.120521524	5.066940123
20%quantile	-0.6799955	-3.971917773	4.841093943
30%quantile	-0.4166668	-2.458337494	4.900007717
40%quantile	-0.1773466	-1.162987505	5.557709034
50%quantile	0.05762205	0.042818873	0.256901207
60%quantile	0.33190683	1.261312791	2.800201333
70%quantile	0.65942546	2.570562692	2.89818538
80%quantile	1.16846836	4.101424509	2.510086053
90%quantile	2.61425997	6.203339771	1.372885573
100%quantile	21.623571	21.54931026	0.003434249

It can be seen from table 7 that the more negative the tail, the worse the prediction effect of fractional O-U process, and the more positive the tail, the better the prediction effect. Especially for the 100% quantile, which is the position of the maximum value, the absolute value percentage of prediction error is only 0.003. It is speculated that this difference is due to the asymmetry of the market, because in China's A-share market, we can only do long but not short. Even if there are means of securities lending, the source of securities lending is very limited in reality.

5. Conclusion

In this study, three different optimization algorithms are used to estimate the parameters of fractional O-U process. The results show that genetic algorithm is faster and more accurate than Newton method and quasi Newton method. Then, the genetic algorithm is used to estimate the parameters of fractional O-U process from 2016-11-21 to 2018-12-28 training set data, and then the parameters are substituted into the fractional O-U process to generate 100 groups of random sequences, which are compared with the test set data from 2016-11-21 to 2019-11-20. Taking 10% as the step, from 0% quantile to 100%, the percentage of prediction error of each quantile is compared. It is found that fractional O-U process is more accurate in predicting the positive tail of real stock index futures RV, but exaggerates the negative tail. The positive and negative tails of real stock index futures are not symmetrical, which may be caused by the non formation mechanism of China's A-share market. But on the whole, fractional O-U process does well describe the basic characteristics of stock index futures RV, which is of great significance to the understanding of volatility.

References

[1] Haress, E. M., Hu, Y. (2021). Estimation of all parameters in the fractional Ornstein–Uhlenbeck model under discrete observations. Statistical Inference for Stochastic Processes, 1-25.

[2] Prior, A., Kleptsyna, M., Milheiro-Oliveira, P. (2017). On maximum likelihood estimation of the drift matrix of a degenerated O–U process. Statistical Inference for Stochastic Processes, 20(1), 57-78.

[3] Hu, Y., Nualart, D., & Zhou, H. (2019). Parameter estimation for fractional Ornstein–Uhlenbeck processes of general Hurst parameter. Statistical Inference for Stochastic Processes, 22(1), 111-142.

[4] Hu, Y., & Nualart, D. (2010). Parameter estimation for fractional Ornstein–Uhlenbeck processes. Statistics & probability letters, 80(11-12), 1030-1038.

[5] Xiao, W., Zhang, W., & Xu, W. (2011). Parameter estimation for fractional Ornstein–Uhlenbeck processes at discrete observation. Applied Mathematical Modelling, 35(9), 4196-4207.

[6] Istas, J., & Lang, G. (1997). Quadratic variations and estimation of the local Hölder index of a Gaussian process. In Annales de l'Institut Henri Poincare (B) probability and statistics (Vol. 33, No. 4, pp. 407-436).

[7]Kleptsyna, M. L., & Le Breton, A. (2002). Statistical analysis of the fractional Ornstein–Uhlenbeck type process. Statistical Inference for stochastic processes, 5(3), 229-248.

[8] Tindel, S., Tudor, C. A., & Viens, F. (2003). Stochastic evolution equations with fractional Brownian motion. Probability Theory and Related Fields, 127(2), 186-204.

[9] Basak, G. K., & Lee, P. (2008). Asymptotic properties of an estimator of the drift coefficients of multidimensional Ornstein-Uhlenbeck processes that are not necessarily stable. Electronic Journal of Statistics, 2, 1309-1344.

[10] Lin, N., & Lototsky, S. V. (2011). Undamped harmonic oscillator driven by additive Gaussian white noise: a statistical analysis. Communications on Stochastic Analysis, 5(1), 13.

[11] Prakasa Rao, B. L. S. (1999). Statistical inference for diffusion type processes. Kendall's Lib. Statist., 8.

[12]Kutoyants, Y. A. (2013). Statistical inference for ergodic diffusion processes. Springer Science & Business Media.