IRSTI 27.35.43 https://doi.org/10.26577/ijmph-2019-i2-5

[1]Yu Jiang,[2]Junjun Lu,[3]Sihan Yin

School of Mathematics, Shanghai University of Finance and Economics, Shanghai 200433, P.R. China
e-mail:[1]jiang.yu@mail.shufe.edu.cn [2]e-mail: lujunjun0919@163.com [3]e-mail: shirleyysh@163.com

Bayesian inference approach to inverse problem in a fractional option pricing model

Abstract. As is well known to us, the Black-Scholes (B-S) model is an important and useful mathematical model for pricing a European options contract. However, because some strict assumptions in this model are not consistent with the real financial market, there are many limitations in practical applications. This paper investigates the inverse option problems (IOP) in a fractional option pricing model, which is derived from the finite moment log-stable (FMLS) model. We identify the model coefficients such as tail index α and the implied volatility σ from the measured data by using three statistical inversionschemeswhich are well known asMarkov Chain Monte Carlo (MCMC) algorithm, slice sampling algorithm and Hamiltonian/hybrid Monte Carlo (HMC) algorithm. Our numerical tests indicate that these Bayesian inference approaches can recover the unknown coefficients well.
Key words: FMLS model, statistical inversion, implied volatility, tail index, Bayesian Inference.

Introduction

As is well known to us, the Black-Scholes (B-S) model is an important and useful mathematical model for pricing a European options contract (cf. [1]). However, because some strict assumptions in this model are not consistent with the real financial market, there are many limitations in practical applications. In particular, the implied volatility of options derived from the B-S model is a constant and cannot fit to the actual "volatility smile" pattern. Recently, the fractional B-S option pricing model has begun to be widely concerned by assuming the price of the original asset is subject to the fractional Brownian motion, or even more general Lévy processes. Among these generalized B–S model, the finite moment log-stable (FMLS) model can effectively capture the leptokurtic feature observed in many financial markets (cf.[3, 4, 6 and 13]).

The stochastic differential equation corresponding to the FMLS model is as follows:

$$\frac{dS_t}{S_t} = \mu dt + \sigma dL_t^{\alpha,-1}, \qquad (1)$$

where S is native asset price, t is expected return time, μ and σ are expected rate of return and asset volatility, respectively. $L_t^{\alpha,-1}$ here denotes the maximally skewed Lévy stable process with a tail index $\alpha \in (0,2)$.

By assume $x_t = \ln S_t$ and according to the argument in [4], SDE(1)can be derived into the following fractional parabolic partial differential equations with the spatial-fractional derivatives:

$$\begin{cases} \frac{\partial V}{\partial t} + \left(r + \frac{1}{2}\sigma^\alpha \sec\frac{\alpha\pi}{2}\right)\frac{\partial V}{\partial x} - \left(\frac{1}{2}\sigma^\alpha \sec\frac{\alpha\pi}{2}\right) {}_{-\infty}D_x^\alpha V - rV = 0, \\ V(x,T;\alpha) = \Pi(x) := \max(e^x - K, 0), \end{cases} \qquad (2)$$

where V is option price, r is risk free rate, $\Pi(x)$ is payoff function with a given strike price K. Here ${}_{-\infty}D_x^\alpha(\cdot)$is the Weyl fractional operator defined as follows:

$${}_{-\infty}D_x^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)}\frac{\partial^n}{\partial x^n}\int_{-\infty}^{x}\frac{f(y)}{(x-y)^{\alpha+1-n}}\mathrm{d}y \quad n-1 \le \mathrm{Re}(\alpha) < n.$$

Similar withthe inverse option problem basing on the B-S model pioneered by Dupire [7], we come to our inverse option problem basing on the FMSL model as follows:

Inverse Problem: Recover the tail index α and the implied volatility σ from the measured option price $V(x)$ at a given t such that $T-t$ is a fixed constant.

However, because of the nonlinear dependency of V on the coefficients α and σ, the uniqueness and stability issues of this inverse problem are quite difficult. Thus, we only desire to have a fast and stable numerical inversion algorithm for solving this inverse problem. Usually, for this purpose, a regularized iterative algorithm such as the Levenberg-Marquardt (L-M) algorithm will be the first choice ([10]). Unfortunately, without a good enough initial guess, iterations in L-M algorithm will not converge. On the other hand, a statistical inversion algorithm such as Metropolis-Hastings Markov Chain Monte Carlo (MH-MCMC) algorithm is now widely used with great success for solving a variety of inverse problems ([11]). Here in this paper we will discuss how to apply the MH-MCMC algorithm to recover the unknown α and σ.

Moreover, both L-M algorithm and MH-MCMC algorithm require for a fast forward solver, which can quickly get the accuracy numerical solution to our PDE model(2). Here we use the closed-form analytical solution given by Chen *et al.* [5] as follows:

$$V(x,t)=Ke^{-\gamma\tau}\int_{d_1}^{+\infty} f_{\alpha,0}(|m|)\mathrm{d}m-e^{x}\int_{d_1}^{+\infty} e^{-\tau-\tau^{1/\alpha}m} f_{\alpha,0}(|m|)\mathrm{d}m. \tag{3}$$

where

$$d_1=\frac{x-\ln K-(1-\gamma)\tau}{\tau^{\frac{1}{\alpha}}},$$

$$\tau=-\frac{\sigma^{\alpha}(T-t)}{2}\sec\frac{\alpha\pi}{2},$$

$$\gamma=-2r\left(\sigma^{\alpha}\sec\frac{\alpha\pi}{2}\right)^{-1},$$

and

$$f_{\alpha,0}(x)=\frac{1}{\pi}\sum_{n=1}^{\infty}\frac{\gamma(1+n/\alpha)}{n!}\sin\left(\frac{\pi n}{2}\right)(-x)^{n-1}.$$

This solution will reduce to B-S formula by setting $\alpha=2$.

The rest of this paper is organized as follows. In Section 2, three statistical inversion schemes for our inverse option problem are described and Section 3 is devoted to the numerical studies of our inversion schemes.

Statistical Inversion Schemes

In practices, the option price V is generally obtained on the different asset price $(S_1,\cdots,S_N)^{\mathrm{T}}$, and we denote:

$\mathbf{V}:=(V_1,\cdots,V_N)^{\mathrm{T}}=(V(S_1),\cdots,V(S_N))^{\mathrm{T}}$.

Now our inverse problem comes to the following nonlinear inverse problem:

$$\mathbf{V}=\mathbf{F}(\mathbf{x}),$$

with respect to unknown coefficients we intend to recover:

$$\mathbf{x}:=(\alpha,\sigma)^{\mathrm{T}}.$$

Here we denote the mapping $\mathbf{F}:\mathbb{R}^2\to\mathbb{R}^N$.

We can assume the noise $\mathbf{e}$ contained in observation

$$\mathbf{V}^{\mathbf{e}}=\mathbf{V}+\mathbf{e},$$

to be Gaussian type white noise, i.e. components of the random noise $\mathbf{e}$ are independent identically distributed (i.i.d.) such that $\mathbf{e}\sim N(0,\sigma_{\mathbf{e}}^2\mathbf{I})$, where σ_e is known noise level and $\mathbf{I}$ is an identity matrix. Thus, the posterior distribution is usually formulated as follows according to the knowledge of Bayesian inference (cf. [11]):

$$p(\mathbf{x}|\mathbf{V})\propto\exp\left(-\frac{1}{2\sigma_{\mathbf{e}}^2}\parallel\mathbf{V}^{\mathbf{e}}-\mathbf{F}(\mathbf{x})\parallel_2^2\right)p(\mathbf{x}).$$

The prior distribution here is simply assumed to be uniform, i.e.

$$p(\mathbf{x}):=\begin{cases}1, & \mathbf{x}\in D,\\ 0, & \mathbf{x}\notin D.\end{cases}$$

with a large enough admissible set D of $\mathbf{x}$. The above posterior distribution can be written into following form:

$$p(\mathbf{x}|\mathbf{V}) \propto \exp\left(-\frac{1}{2\sigma_{\mathbf{e}}^2} \| \mathbf{V}^e - \mathbf{F}(\mathbf{x}) \|_2^2\right).$$

One can get the maximum a posterior (MAP) estimator $\mathbf{x}_{\text{MAP}}$ of $\mathbf{x}$ such that:

$$\mathbf{x}_{\text{MAP}} = \underset{\mathbf{x}}{\operatorname{argmax}}\, p(\mathbf{x}|\mathbf{V}).$$

However, MAP estimate is *point estimate*, different measured data $\mathbf{V}$ will come to different $\mathbf{x}$. To avoid this, one can compute the posterior conditional mean (CM) estimator from various point estimators:

$$\mathbf{x}_{\mathbf{CM}} := \int_{\mathbb{R}^2} \mathbf{x}\, p(\mathbf{x}|\mathbf{V}) \mathrm{d}\mathbf{x}.$$

Furthermore, it is hard to know the explicit form of $p(\mathbf{x}|\mathbf{V})$ in practice. Some sampling algorithm can be applied to obtain a set of samples $\mathbf{x}_k$ ($k = 1, \cdots, K$) drawn independently from the distribution $p(\mathbf{x}|\mathbf{V})$ (cf. [2, 11]), and thus $\mathbf{x}_{\mathbf{CM}}$ comes to a finite sum approximately

$$\mathbf{x}_{\mathbf{CM}} \approx \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k.$$

This is exactly the desired solution of our related inverse problem in the sense of Bayesian inference.

MH-MCMC Algorithm: in this paper, we first apply the most famous and popular sampling algorithm: Metropolis-Hastings algorithm ([8, 12]) shown as follows:

1. Generate $\mathbf{x}'$ from $q(\mathbf{x}'|\mathbf{x}_k) \sim N(\mathbf{x}_k, \Sigma)$ for given $\mathbf{x}_k$.

2. Calculate the choice

$$a(\mathbf{x}', \mathbf{x}_k) = \min\left\{1, \frac{p(\mathbf{x}'|\boldsymbol{V})}{p(\mathbf{x}_k|\boldsymbol{V})}\right\}.$$

3. Update $\mathbf{x}_k$ as $\mathbf{x}_{k+1} = \mathbf{x}'$ with probability $a(\mathbf{x}', \mathbf{x}_k)$, otherwise set $\mathbf{x}_{k+1} = \mathbf{x}_k$.

4. Here the proposal distribution $q(\mathbf{x}|\mathbf{y})$ is given as

$$q(\mathbf{x}|\mathbf{y}) \propto \exp\left(-\frac{1}{2\Sigma} \| \mathbf{x} - \mathbf{y} \|_2^2\right).$$

with given step sizes γ_α and γ_σ such that $\Sigma = \operatorname{diag}(\gamma_\alpha^2, \gamma_\sigma^2)$. For more details about MH-MCMC algorithm, we can refer to [2, 11].

However, the performance of MH-MCMC algorithm highly dependents on the specific choice of proposal distribution $q(\mathbf{x}|\mathbf{y})$. Without a carefully tuning of the step sizes γ_α and γ_σ, this algorithm will not lead to efficient samples. Therefore, we desire to have some sampling algorithm which will determine the step sizes "automatically". The following two well-known sampling algorithms introduced in [2] can be applied.

Slice Sampling Algorithm: the basic idea of this algorithm is to generate samples from the joint $(\mathbf{x}, u)$ space with an additional variable $u = p(\mathbf{x})$ where $p(\mathbf{x})$ is just the sampling distribution where we set it to the posterior distribution $p(\mathbf{x}|\mathbf{V})$. The procedure for finding the next sampling point $\mathbf{x}'$ from the current sampling point $\mathbf{x}$ is shown by following algorithm (see also Figure 1):

1. Generate a real value u from the uniform distribution $U(0, p(\mathbf{x}))$, and define the slice $S = \mathbf{x}{:}\, u < p(\mathbf{x})$.

2. Find a hyper rectangle $H := (L_1, R_1) \times \cdots \times (L_M, R_M)$ around $\mathbf{x}$, which contains the slice S as much as possible.

3. Generate the new sample $\mathbf{x}'$ uniformly in this hyperrectangle H.

Due to the existence of computational error, it is difficult to locate the hyper rectangle H exactly. A detailed numerical procedure about it can be found in [2]. Unfortunately, this numerical procedure always slows down the sampling.

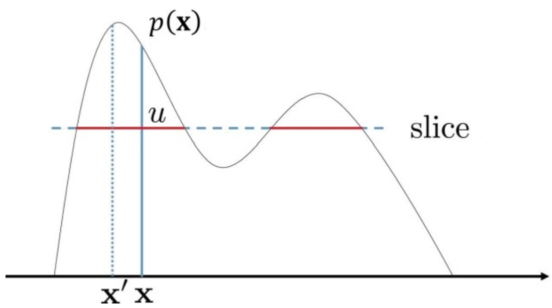

Figure 1 – Slice sampling algorithm.

Hamiltonian Monte Carlo Algorithm: this sampling algorithm is also known as Hybrid Monte Carlo (HMC) algorithm. In this algorithm, the transition of sampling points is not through the proposed distribution $q(\mathbf{x}|\mathbf{y})$, but by solving the following Hamiltoniansystem:

$$\begin{cases} \frac{\partial \mathbf{q}'}{\partial t} = \frac{\partial H}{\partial \mathbf{p}'} = \frac{\partial K(\mathbf{p})}{\partial \mathbf{p}'}, \\ \frac{\partial \mathbf{p}'}{\partial t} = \frac{\partial H}{\partial \mathbf{q}'} = -\frac{\partial U(\mathbf{q})}{\partial \mathbf{q}'}. \end{cases}$$

where $\mathbf{q}$ is a statevariable, $U(\mathbf{q})$ is the potential energy of the dynamical system when in state $\mathbf{q}$, $\mathbf{p}$ is the momentumvariable, and $K(\mathbf{p})$ is the kinetic energy. When one state $(\mathbf{q},\mathbf{p})$ changes to another state $(\mathbf{q}',\mathbf{p}')$, the value of the following Hamiltonianisalways constant:

$$H(\mathbf{q},\mathbf{p}) = U(\mathbf{q}) + K(\mathbf{p}) \\ = U(\mathbf{q}') + K(\mathbf{p}') = H(\mathbf{q}',\mathbf{p}').$$

Based on this, we have the following HMC algorithm:

1. Calculate the potential energy $U(q) = p(\mathbf{q}|\mathbf{v})$ of the current state $\mathbf{q} = \mathbf{x}$.
2. Generate the momentum $\mathbf{p}$ from a given simply normal distribution $e^{-K(\mathbf{p})}$.
3. Update the sample $\mathbf{x}' = \mathbf{q}'$ by solving the above Hamiltonian system.

However, in practice, we can only solve the Hamiltonian equations numerically by applying the leapfrog scheme. Therefore, to ensure the samples are all in the same stable Markovchain, we use the “accept-reject" criterion to accept the candidate sample $\mathbf{q}'$or not:

$$a = \min\left(1, e^{-H(\mathbf{q}',\mathbf{p}')+H(\mathbf{q},\mathbf{p})}\right).$$

This indeed is similar to the one used in above MH-MCMC algorithm.

Numerical Test

In this section, we will test the performance of three algorithms for solving our inverse option price numerically.

Simulated Data: we firstly generate the noise free simulated data and the noisy simulated data which contains 20% relative Gaussian noise by using the closed-form analytical solution (3) (see). Here, the parameters in (3) are the same as the ones in Chen *et al.*: $K = 10$, $r = 0.1$, $T - t = 1$ (year) and $\mathbf{x} = (\alpha,\sigma)^{\mathrm{T}} = (1.75, 0.2440)^{\mathrm{T}}$.

Therefore, we test the sampling algorithms shown above under these simulated data one by one.The initial value of $\mathbf{x}_0$ is always set to $(\alpha_0,\sigma_0)^{\mathrm{T}} = (2, 0.5)^{\mathrm{T}}$.

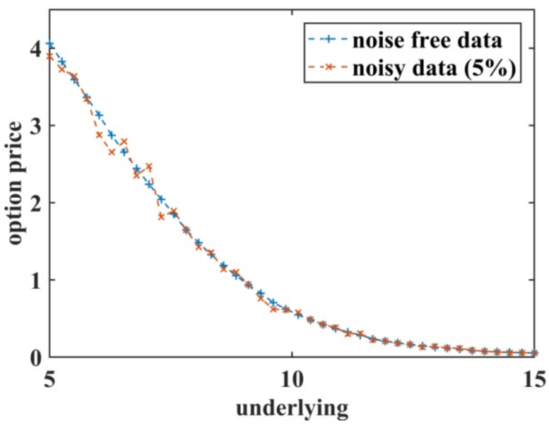

Figure 2 - Simulated data

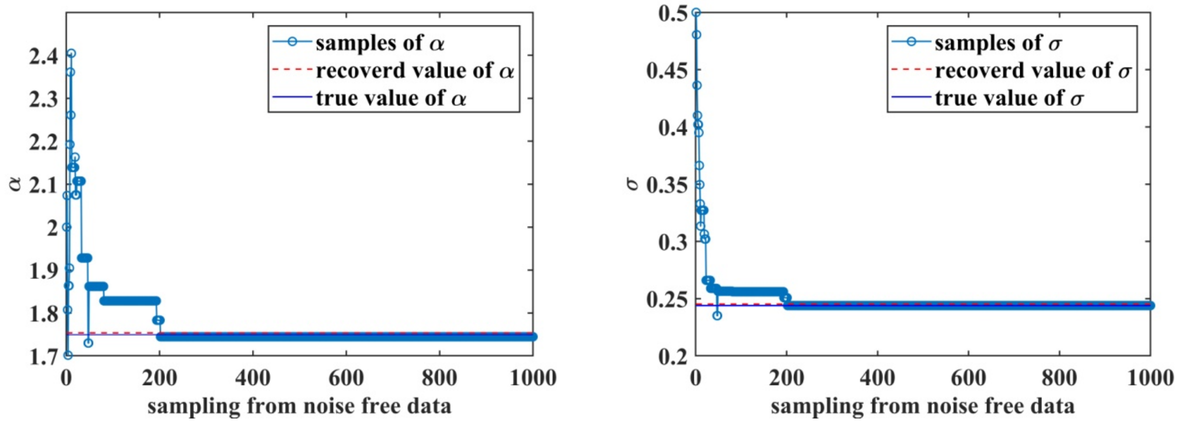

Figure 3 – Samples by applying MH-MCMC algorithm from noise free data: (up) α; (down) σ

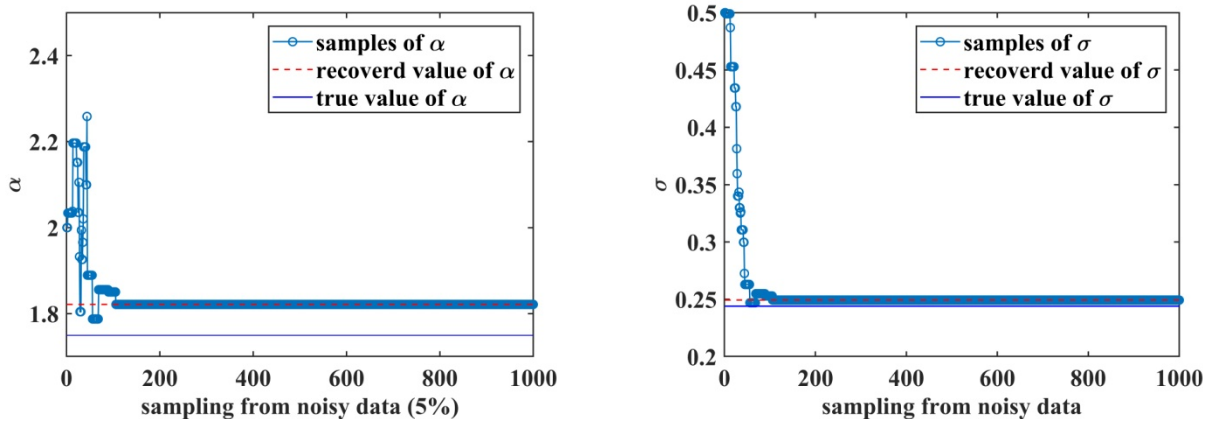

Figure 4 – Samples by applying MH-MCMC algorithm from noisy data (5%): (up) α; (down)σ

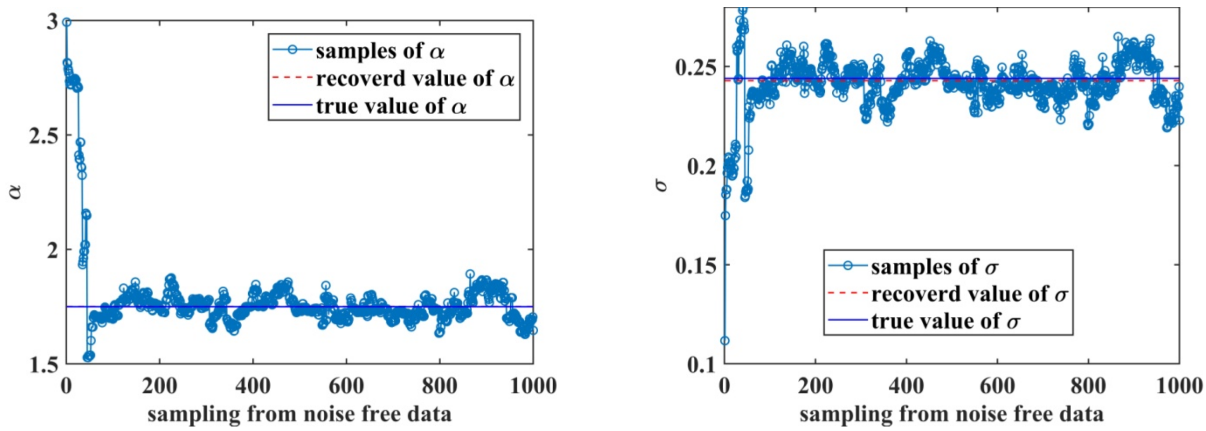

Figure 5 – Samples by applying slice sampling algorithm from noise free data: (up) α; (down) σ.

MH-MCMC Algorithm: the other hyper parameters used in MH-MCMC algorithm are set to be $\Sigma = \mathrm{diag}(0.25^2, 0.025^2)^{\mathrm{T}}$ and $\sigma_{\mathbf{e}} = 10^{-6}$. The total sampling time is 1000. Samples from noise free data are shown in, while samples from noisy data are shown in. We always set up some "burn-in" time, which is thought as the start point of stable Markov Chain. The mean value of samples among this "burn-in" time (=101) and the ending point (=1000) is computed and set to be our recovered result shown in Table 1. It is clear the MH-MCMC algorithm works well and does not trap in the any local minimums. Theoretically, a large number of samplings will final derive to a stable Markov chain, but in practice the computing cost will be very expansive, and thus we always need to manually choose the hyper parameters σ_e and Σ in MH-MCMC algorithm such that the Markov chain "converge" fast and stable. This is a big disadvantage of this MH-MCMC algorithm, and it will be quite interesting for us to try the other two sampling algorithms.

Slice Sampling Algorithm: the only hyper parameter needs to be set is$\sigma_e = 10^{-3}$. The total sampling time is also 1000. Samples from noise free data are shown in, while samples from noisy data are shown in. The mean value of samples among this "burn-in" time (=101) and the ending point (=1000) is computed and set to be our recovered result shown in Table 1.

Table 1 – Recovered results by applying MH-MCMC algorithm.

	α	σ
Initial value	2	0.5
Noise free data	1.7536	0.2453
Noisy data (5%)	1.8215	0.2493
True value	1.75	0.244

Table 2 – Recovered results by applying slice sampling algorithm.

	α	σ
Initial value	2	0.5
Noise free data	1.7509	0.2429
Noisy data (5%)	2.0617	0.2529
True value	1.75	0.244

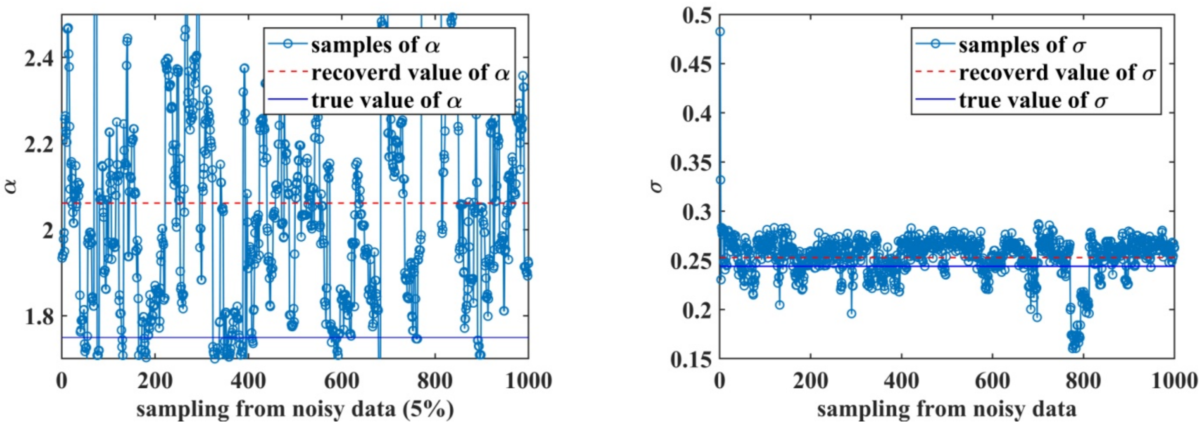

Figure 6 – Samples by applying slice sampling algorithmfrom noisy data (5%): (up) α; (down) σ

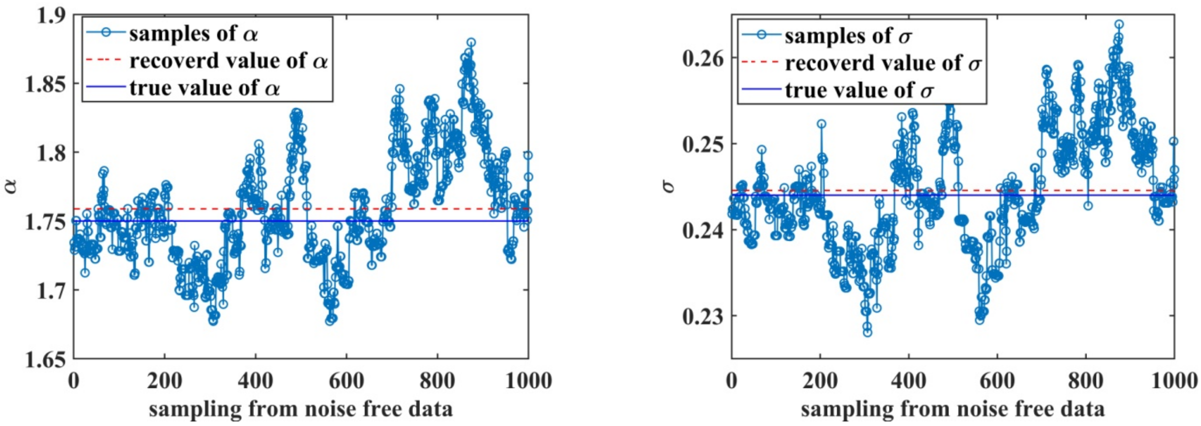

Figure 7 – Samples by applying HMC algorithm from noise free data: (up) α; (down) σ

It is clear that the samples always transit with probability one without stopping. Therefore, these samples generated by slice sampling algorithm can exhibit the real random characteristics of the posterior distribution $p(\mathbf{x}|\mathbf{V})$ we desire to recover in every statistical inverse problem.

Hamiltonian Monte Carlo Algorithm: similar to slice sampling algorithm, the only hyper parameter needs to be set is $\sigma_{\mathbf{e}} = 10^{-1}$. The total sampling time is again 1000. Samples from noise free data are shown in, while samples from noisy data are shown in. The mean value of samples among this "burn-in" time (=101) and the ending point (=1000) is computed and set to be our recovered result shown in Table 1.

Similar to slice sampling algorithm, the samples draw by HMC algorithm usually transit stalely without stopping. These samples generated can also exhibit the real random characteristics of the posterior distribution $p(\mathbf{x}|\mathbf{V})$ we desire to recover. However, the numerical computation of Hamiltonian system in each sampling is quite time consuming, and thus HMC algorithm is much slower than the other two in this paper.

Table 3 – Recovered results by applying HMC algorithm

	α	σ
Initial value	2	0.5
Noise free data	1.7588	0.2446
Noisy data (5%)	1.9255	0.2539
True value	1.75	0.244

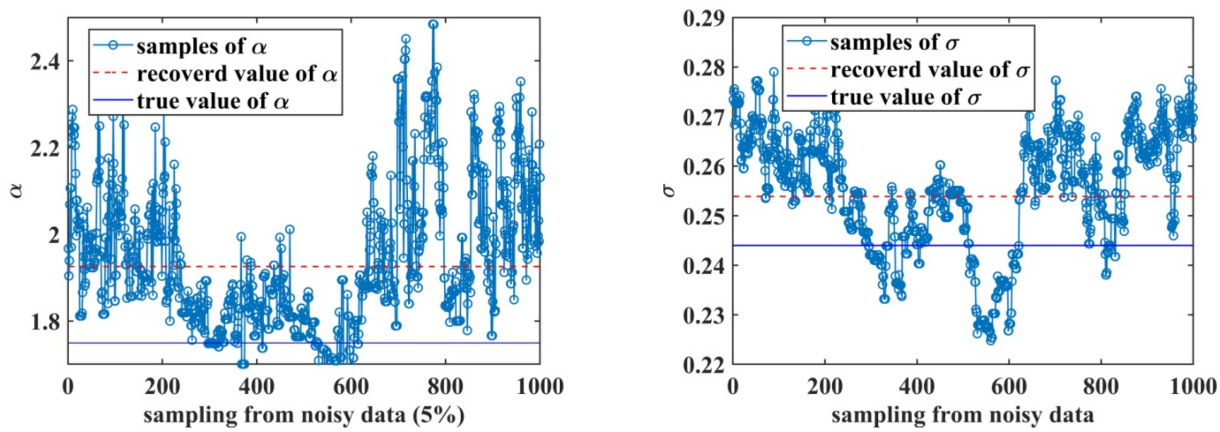

Figure 8 – Samples by applying HMC algorithm from noisy data (5%): (up) α; (down) σ

Conclusion: all of these three sampling algorithms can solve our invers option problem well. The recovery of the implied volatility σ is much better than the recovery of the tail index αbecause of the high nonlinearity of the problem corresponding to α.

Also, here isa short summary of the main advantage and disadvantage of inversion algorithms involved in this paper:

L-M algorithm

Disadvantage: good initial value $\mathbf{x}_0$ is required, otherwise it is easy to fall into local minimum.

Advantage: if the initial value is properly selected, the convergence speed is fast and the result is accuracy.

MH-MCMC algorithm

Disadvantage: need to carefully choose a proposal distribution, otherwise the rejected rate will be quite high and need to have a large number of samples to draw/recover the posterior distribution $p(\mathbf{x}|\mathbf{V})$.

Advantage: if the proposal distribution is properly chosen, the recovered posterior distribution $p(\mathbf{x}|\mathbf{V})$ is good.

Slice sampling algorithm

Disadvantage: finding a proper slice in each sampling is time consuming.

Advantage: no need of the proposal distribution and the recovered posterior distribution $p(\mathbf{x}|\mathbf{V})$ is good.

HMC algorithm

Disadvantage: numerical computation of Hamiltonian system in each sampling is quite time consuming.

Advantage: no need of the proposal distribution and the recovered posterior distribution $p(\mathbf{x}|\mathbf{V})$ is good.

Acknowledgment

This research was supported by National Natural Science Foundation of China (No. 11971121).

Reference

1. Black F., Scholes M.S., "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy* vol. 81, no. 3 (1973):637-654.

2. Bishop C.M., *Pattern Recognition and Machine Learning*. NewYork: Springer-Verlag Inc.,2006.

3. Carr P., Wu L., "The Finite Moment Log Stable Process and Option Pricing." *Journal of Finance* vol.58, no. 2 (2003):753-778.

4. Cartea A., Del-Castillo-Negrete D., "Fractional Diffusion Models of Option Prices in Markets with Jumps." *Birkbeck Working Papers in Economics and Finance* vol.374, no. 2 (2006):749-763.

5. Chen W., Xu X., Zhu S.P., "Analytically pricing European-style options under the modified Black-Scholes equation with a spatial-fractional derivative." *Quarterly of Applied Mathematics* vol. 72, no. 3 (2014):597-611.

6. Cont R., Tankov P.,*Financial Modelling with Jump Processes*. London: Chapman & Hall/CRC Press, 2003.

7. Dupire B., "Price with a smile." *RISK*vol. 7, (1994): 18-20.

8. Hastings W.K., "Monte Carlo Sampling Methods Using Markov Chains and Their Applications."*Biometrika* vol.57, no 1 (1970):97-109.

9. Liu J.S.,*Monte Carlo Strategies in Scientific Computing*. New York: Springer-Verlag Inc.,2004.

10. Kaltenbacher B., Neubauer A., Scherzer O.,*Iterative regularization methods for nonlinear ill-posed problems.*KG Berlin: Walter de Gruyter GmbH & Co,2008.

11. Kaipio J.P., Somersalo E., *Statistical and Computational Inverse Problems.* New York: Springer-Verlag Inc., 2005.

12. Metropolis N., Rosenbluth A.W., Rosenbluth M.N., et al. "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics*vol. 21(2004):1087-1092.

13. Schoutens W., *Lévy Processes in Finance: Pricing Financial Derivatives.*Hoboken: John Wiley & Sons Inc., 2003.