Scientific Research Publishing

Advances in Pure Mathematics, 2025, 15(4), 303-318
https://www.scirp.org/journal/apm
ISSN Online: 2160-0384
ISSN Print: 2160-0368

Bayesian Inference Method for Stable Distributions

Mengyu Hu

Department of Mathematics, College of Science, Shanghai University, Shanghai, China
Email: 13786733555@163.com

How to cite this paper: Hu, M.Y. (2025) Bayesian Inference Method for Stable Distributions. *Advances in Pure Mathematics*, **15**, 303-318.
https://doi.org/10.4236/apm.2025.154015

Received: March 17, 2025
Accepted: April 19, 2025
Published: April 22, 2025

Abstract

Stable distributions are well-known for their desirable properties and can effectively fit data with heavy tail. However, due to the lack of an explicit probability density function and finite second moments in most cases, traditional parametric inference methods are no longer applicable. Bayesian Synthetic Likelihood is a likelihood-free Bayesian inference method based on model simulations, which effectively addresses parameter inference problems when the probability density function is not explicitly available. Semi-parametric Bayesian Synthetic Likelihood relaxes the normality assumption by incorporating semi-parametric estimation methods, but it performs poorly when applied to data with heavy tail and excess kurtosis. To improve this, we introduced adaptive Monte Carlo algorithm to enhance the convergence speed, and transformed kernel density estimation to increase the estimation accuracy. Numerical experiments and empirical analysis on stable distributions validated the superiority of the proposed improvements.

Keywords

Stable Distribution, Bayesian Synthetic Likelihood, Adaptive Monte Carlo Algorithm, Transformed Kernel Density Estimation

1. Introduction

Stable distributions are a class of distributions with excellent properties. They are the only distributions with an absorption domain, meaning that the sum of any random variables asymptotically converges to a stable distribution. As a result, stable distributions possess desirable theoretical properties. Additionally, stable distributions are particularly effective in fitting data with heavy-tail, making them widely applicable in fields such as financial markets and signal processing.

Stable distributions were first introduced by Lévy in 1925. However, the devel-

opment of parameter estimation methods for stable distributions has progressed slowly, mainly due to the following two reasons: 1) With a few exceptions, stable distributions do not have explicit probability density functions or distribution functions. As a result, traditional parameter estimation methods based on probability density functions are no longer applicable. 2) Stable distributions do not have finite second-order moments or higher-order moments. Therefore, traditional parameter estimation methods based on higher-order moments are also unsuitable. Currently, the main parameter estimation methods for stable distribution models include the quantile method [1], the fractional lower-order moment method [2], and the characteristic function method [3]. The quantile method requires estimating from a table based on calculated quantiles, which not only has a limited scope of application but also lacks high precision. The effectiveness of the fractional lower-order moment method depends on the choice of the order, which the user must determine. The characteristic function method estimates results based on sample size and initial parameter values by querying estimation interval tables, but it requires multiple queries to meet the convergence criteria, resulting in large computational effort and low accuracy. Therefore, there is a need to develop a simple and effective parameter estimation method for stable distribution models.

The Bayesian Synthetic Likelihood (BSL) is a likelihood-free inference method based on the Bayesian framework, first introduced by Price *et al.* in 2018 [4]. This method has also been applied to parameter inference in various models, such as the SDEMEM model [5], SDEs model [6], and ARCH model [7]. However, the fundamental assumption of this method is that the observed data follows a Gaussian distribution. When the observed data significantly deviates from a Gaussian distribution, this can lead to inaccurate estimates of the likelihood function, which in turn results in inaccurate estimates of the posterior distribution. Several studies have proposed improvements to the BSL method, which heavily relies on the Gaussian assumption. Fasiolo *et al.* [8] proposed a more flexible density estimator, called the extended empirical saddlepoint approximation, which relaxes the normality assumption. However, this method requires the user to select a shrinkage parameter, and both the accuracy of the results and computational efficiency depend on the choice of this parameter. An *et al.* [9] addressed this issue by using a semi-parametric estimation approach to estimate the likelihood function, leading to the new method called semiBSL. Numerical experiments show that this method significantly outperforms BSL and the extended empirical saddlepoint approximation when the observed data deviates from a Gaussian distribution. In 2022, An *et al.* [10] developed an R package for the MCMC-BSL, MCMC-semiBSL, and related extended methods to facilitate their subsequent use. Picchini *et al.* [11] improved the convergence speed of BSL by introducing a guided adaptive sampling method and demonstrated through numerical experiments that when the initial parameter values are far from the true values, the MCMC sampling method converges slowly and can even get trapped in local optima. Since semiBSL also

employs MCMC sampling, it theoretically faces the same issue.

Therefore, this paper proposes improvements to the semiBSL method. On the one hand, the Adaptive Monte Carlo (AM) algorithm is introduced to enhance the convergence speed of semiBSL. On the other hand, transformed kernel density estimation (TKDE) is incorporated to improve the estimation accuracy of semiBSL when dealing with data exhibiting "heavy-tail" characteristics. Simulation study and empirical study based on stable distributions are conducted, and the experimental results confirm the superiority of the proposed improvements.

2. Theoretical Foundations

2.1. Stable Distributions

There are several ways of defining a stable random variable. First, the initial definition of a stable random variable is based on the sums of random variables. A random variable X is said to have a stable distribution if for any $n \geq 2$, a positive number C_n and real number D_n exist such that:

$$X_1 + X_2 + \cdots + X_n \overset{\text{def}}{=} C_n X + D_n, \tag{1}$$

where $X_1, X_2, \cdots, X_n$ are independent copies of X.

Second, stable variables can be defined via their Lévy measure. For any $\alpha < 2$, the Lévy measure of a stable process is given by:

$$v(dx) = \left(\frac{C_+}{x^{1+\alpha}} 1_{x>0} + \frac{C_-}{x^{1+\alpha}} 1_{x<0} \right) dx, \tag{2}$$

where $1_{x>0}$ is an indicator function. The calculation of the characteristic function of the stable distribution is based on the Lévy-Khintchine formula. The characteristic function of S given by:

$$\Phi(t;S) = \begin{cases} \exp\left(i\mu t - |\sigma t|^{\alpha} \left(1 - i\beta (sign\, t) \tan\left(\frac{\pi\alpha}{2} \right) \right) \right), & \alpha \neq 1 \\ \exp\left(i\mu t - \sigma |t| \left(1 + i\beta \frac{2}{\pi} (sign\, t) \ln |t| \right) \right), & \alpha = 1 \end{cases} \tag{3}$$

where

$$sign\, t = \begin{cases} 1, & t > 0 \\ 0, & t = 0 \\ -1, & t < 0 \end{cases} \tag{4}$$

$0 < \alpha \leq 2$, $\sigma \geq 0$, $-1 \geq \beta \geq 1$, and $\mu \in \mathbb{R}$. The parameter α is the index of stability and it controls the behaviour of the left and right tails. When α is close to 2, the tail becomes thin. β, σ, and μ are the skewness, scale, and location parameters, respectively. When a random variable S follows a stable distribution, we denote it by $S(\alpha, \beta, \sigma, \mu)$.

2.2. Bayesian Synthetic Likelihood

In Bayesian Synthetic Likelihood (MCMC-BSL), the objective is to simulate from

the summary statistic posterior given by

$$p(\boldsymbol{\theta} \mid \boldsymbol{s}_y) \propto p(\boldsymbol{\theta}) p(\boldsymbol{s}_y \mid \boldsymbol{\theta}) \tag{5}$$

where $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$ is the parameter that requires estimation with corresponding prior distribution $p(\boldsymbol{\theta})$. Here, $\boldsymbol{y}$ is the observed data that is subsequently reduced to a summary statistic $\boldsymbol{s}_y = S(\boldsymbol{y})$ where $S(\cdot)$ is the summary statistic function. The dimension of the statistic d must be at least the same size as the parameter dimension, *i.e.* $d \geq p$.

The BSL involves approximating $p(\boldsymbol{s}_y \mid \boldsymbol{\theta})$ with

$$p(\boldsymbol{s}_y \mid \boldsymbol{\theta}) \approx p_A(\boldsymbol{s}_y \mid \boldsymbol{\theta}) = N(\boldsymbol{s}_y \mid \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta})) \tag{6}$$

The mean $\boldsymbol{\mu}(\boldsymbol{\theta})$ and covariance $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ are not available in closed form but can be estimated via independent model simulations at $\boldsymbol{\theta}$. The procedure involves drawing $\boldsymbol{x}_{1:n} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$, where $\boldsymbol{x}_i \sim p(\cdot \mid \boldsymbol{\theta})$ for $i = 1, \cdots, n$, and calculating the summary statistic for each dataset, $\boldsymbol{s}_{1:n} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n)$, where $\boldsymbol{s}_i$ is the summary statistic for $\boldsymbol{x}_i$, $i = 1, \cdots, n$. These simulations can be used to estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ unbiasedly

$$\boldsymbol{\mu}_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{s}_i, \boldsymbol{\Sigma}_n(\boldsymbol{\theta}) = \frac{1}{n-1}\sum_{i=1}^{n} (\boldsymbol{s}_i - \boldsymbol{\mu}_n(\boldsymbol{\theta}))(\boldsymbol{s}_i - \boldsymbol{\mu}_n(\boldsymbol{\theta}))^{\mathrm{T}} \tag{7}$$

We can sample from the approximate posterior using MCMC, see **Algorithm 1**.

Algorithm 1: MCMC-BSL algorithm

Input: Summary statistic of the data, $\boldsymbol{s}_y$, the prior distribution, $\boldsymbol{p}(\boldsymbol{\theta})$, the proposal distribution q, the number of iterations, T, the number of simulation runs, n, the initial value of the chain $\boldsymbol{\theta}^{(0)}$.

Output: MCMC sample $(\boldsymbol{\theta}^{(0)}, \cdots, \boldsymbol{\theta}^{(T)})$ from the BSL posterior, $p_{A,n}(\boldsymbol{s}_y \mid \boldsymbol{\theta})$. Some samples can be discarded as burn-in if required.

1: Simulate $\boldsymbol{x}_{1:n} \sim p(\cdot \mid \boldsymbol{\theta}^{(0)})$ and compute $\boldsymbol{s}_{1:n}$

2: Compute $\boldsymbol{\mu}_n(\boldsymbol{\theta}^{(0)})$ and $\boldsymbol{\Sigma}_n(\boldsymbol{\theta}^{(0)})$ using (7)

3: for $i = 1, \cdots, T$ do

4: Draw $\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}^{(i-1)})$

5: Simulate $\boldsymbol{x}_{1:n}^* \sim p(\cdot \mid \boldsymbol{\theta}^*)$ and compute $\boldsymbol{s}_{1:n}^*$

6: Compute $\boldsymbol{\mu}_n(\boldsymbol{\theta}^*)$ and $\boldsymbol{\Sigma}_n(\boldsymbol{\theta}^*)$ using (7)

7: Compute $r = \min\left(1, \frac{p_{A,n}(\boldsymbol{s}_y \mid \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*) q(\boldsymbol{\theta}^{(i-1)} \mid \boldsymbol{\theta}^*)}{p_{A,n}(\boldsymbol{s}_y \mid \boldsymbol{\theta}^{(i-1)}) p(\boldsymbol{\theta}^{(i-1)}) q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(i-1)})}\right)$

8: if $U(0,1) < r$ then

9: Set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*, \boldsymbol{\mu}_n(\boldsymbol{\theta}^{(i)}) = \boldsymbol{\mu}_n(\boldsymbol{\theta}^*), \boldsymbol{\Sigma}_n(\boldsymbol{\theta}^{(i)}) = \boldsymbol{\Sigma}_n(\boldsymbol{\theta}^*)$

10: else

11: Set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}, \boldsymbol{\mu}_n(\boldsymbol{\theta}^{(i)}) = \boldsymbol{\mu}_n(\boldsymbol{\theta}^{(i-1)}), \boldsymbol{\Sigma}_n(\boldsymbol{\theta}^{(i)}) = \boldsymbol{\Sigma}_n(\boldsymbol{\theta}^{(i-1)})$

12: end

2.3. Semi-Parametric Bayesian Synthetic Likelihood

The difference between MCMC-semiBSL and MCMC-BSL lies in the treatment of the likelihood function. MCMC-BSL directly assumes the likelihood function to be a Gaussian distribution with unknown parameters, while MCMC-semiBSL employs a Copula model to estimate the likelihood function. The specific steps are as follows: First, kernel density estimation (KDE) is used to estimate each marginal density. Then, a Gaussian Copula function is applied to estimate the joint density. Finally, the resulting joint density is taken as the estimated likelihood function.

Based on kernel density estimation, the marginal density of the j-th component of the summary statistic $\boldsymbol{s}_y$ can be estimated as:

$$\hat{f}\left(s_y^j\right)=\frac{1}{n}\sum_{i=1}^{n}K_h\left(s_y^j-x_i^j\right),\ \ j=1,\cdots,d \tag{8}$$

where s_y^j represents the j-th component of the summary statistic $\boldsymbol{s}_y$, and x_i^j denotes the j-th component of the i-th simulated data. The kernel density estimator $K_h(u)$ is given by $K_h(u)=h^{-1}K(u/h)$, where h is the bandwidth, and $K(\cdot)$ is the kernel function, satisfying $K(\cdot)\geq 0$ and $\int K(\cdot)\mathrm{d}x=1$. There are many kernel functions that satisfy these conditions, and in MCMC-semiBSL, the Gaussian kernel function $K(u)=1/\sqrt{2\pi}\exp\left(-u^2/2\right)$ is used. The bandwidth is chosen according to Silverman [12], with $h=0.9n^{-0.2}\min\left(\delta,\mathrm{IQR}/1.34\right)$, where IQR denotes the interquartile range. The marginal distribution function $\hat{F}\left(s_y^j\right)$ can be directly estimated from the marginal density function.

After obtaining the marginal distribution, MCMC-semiBSL uses a Gaussian Copula function to model the dependence structure between the components of the summary statistics. The Gaussian Copula density function is given by:

$$c(u)=\frac{1}{\sqrt{\det(\boldsymbol{R})}}\exp\left\{-\frac{1}{2}\boldsymbol{\eta}^{\mathrm{T}}\left(\boldsymbol{R}^{-1}-\boldsymbol{I}_d\right)\boldsymbol{\eta}\right\} \tag{9}$$

where $\boldsymbol{I}_d$ is the d-dimensional identity matrix, $\boldsymbol{\eta}=\left(\Phi^{-1}(u_1),\cdots,\Phi^{-1}(u_d)\right)^{\mathrm{T}}$, $\Phi^{-1}(\cdot)$ is the inverse of the standard normal distribution, and $u_j=\hat{F}\left(s_y^j\right)$, for $j=1,\cdots,d$. If the correlation matrix $\boldsymbol{R}$ in the above expression can be estimated, the likelihood function can be estimated as:

$$\begin{aligned}p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}\right)=&\int\frac{1}{\sqrt{\det\left(\hat{\boldsymbol{R}}\right)}}\exp\left\{-\frac{1}{2}\hat{\boldsymbol{\eta}}_{\boldsymbol{s}_y}^{\mathrm{T}}\left(\hat{\boldsymbol{R}}^{-1}-\boldsymbol{I}_d\right)\boldsymbol{\eta}_{\boldsymbol{s}_y}\right\}\\&\times\prod_{j=1}^{d}\hat{f}\left(s_y^j\right)\prod_{i=1}^{n}p\left(S\left(\boldsymbol{x}_i\right)\mid\boldsymbol{\theta}\right)\mathrm{d}S\left(\boldsymbol{x}_i\right)\cdots S\left(\boldsymbol{x}_n\right)\end{aligned} \tag{10}$$

MCMC-semiBSL estimates the correlation matrix using the Gaussian rank correlation (GRC) method proposed by Boudt *et al.* in 2012 [13]. According to the GRC method, the (i,j)-th component $R_{i,j}$ of the correlation matrix $\boldsymbol{R}$ can be estimated as:

$$\hat{R}_{i,j}=\frac{\sum_{k=1}^{n}\Phi^{-1}\left(\frac{r\left(s_k^i\right)}{n+1}\right)\Phi^{-1}\left(\frac{r\left(s_k^j\right)}{n+1}\right)}{\sum_{k=1}^{n}\Phi^{-1}\left(\frac{k}{n+1}\right)^2} \tag{11}$$

where $r(\cdot):\mathbb{R}\to\mathcal{A},\mathcal{A}\equiv\{1,\cdots,n\}$ is the rank function, and $\Phi^{-1}(\cdot)$ is the inverse of the standard normal distribution. After estimating the likelihood function, similar to MCMC-BSL, the MCMC algorithm with iterations T is used to sample from $p(\boldsymbol{s}_y\mid\boldsymbol{\theta})$. The specific algorithm procedure is as follows:

Algorithm 2: MCMC-semiBSL algorithm

Input: Summary statistic of the data, $\boldsymbol{s}_y$, the prior distribution, $\boldsymbol{p}(\boldsymbol{\theta})$, the proposal distribution q, the number of iterations, T, the number of simulation runs, n, the initial value of the chain $\boldsymbol{\theta}^{(0)}$.

Output: MCMC sample $\left(\boldsymbol{\theta}^{(0)},\cdots,\boldsymbol{\theta}^{(T)}\right)$ from the BSL posterior, $p_{A,n}(\boldsymbol{s}_y\mid\boldsymbol{\theta})$. Some samples can be discarded as burn-in if required.

1: Simulate $\boldsymbol{x}_{1:n}\sim p\left(\cdot\mid\boldsymbol{\theta}^{(0)}\right)$ and compute $\boldsymbol{s}_{1:n}$

2: Compute $\hat{f}^{(0)}(\boldsymbol{s}_y)$ and $\hat{F}^{(0)}(\boldsymbol{s}_y)$ using (8)

3: Compute $\hat{\boldsymbol{R}}^{(0)}$ using (11)

4: Compute $p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(0)}\right)$ using (10)

5: for $i=1,\cdots,T$ do

6: Draw $\boldsymbol{\theta}^{*}\sim q\left(\cdot\mid\boldsymbol{\theta}^{(i-1)}\right)$

7: Simulate $\boldsymbol{x}_{1:n}^{*}\sim p\left(\cdot\mid\boldsymbol{\theta}^{*}\right)$ and compute $\boldsymbol{s}_{1:n}^{*}$

8: Compute $\hat{f}^{*}(\boldsymbol{s}_y)$ and $\hat{F}^{*}(\boldsymbol{s}_y)$ using (8)

9: Compute $\hat{\boldsymbol{R}}^{*}$ using (11)

10: Compute $p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{*}\right)$ using (10)

11: Compute $r=\min\left(1,\dfrac{p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{*}\right)p\left(\boldsymbol{\theta}^{*}\right)q\left(\boldsymbol{\theta}^{(i-1)}\mid\boldsymbol{\theta}^{*}\right)}{p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i-1)}\right)p\left(\boldsymbol{\theta}^{(i-1)}\right)q\left(\boldsymbol{\theta}^{*}\mid\boldsymbol{\theta}^{(i-1)}\right)}\right)$

12: if $U(0,1)<r$ then

13: Set $\boldsymbol{\theta}^{(i)}=\boldsymbol{\theta}^{*}$, $p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i)}\right)=p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{*}\right)$

14: else

15: Set $\boldsymbol{\theta}^{(i)}=\boldsymbol{\theta}^{(i-1)}$, $p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i)}\right)=p_{semiBSL}\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i-1)}\right)$

16: end

2.4. Adaptive Monte Carlo Algorithm

Markov Chain Monte Carlo (MCMC) is a stochastic simulation method based on Markov chains, used to estimate the numerical characteristics of complex probability distributions. The foundational algorithm of MCMC was proposed by W. K. Hastings in 1970, known as the Metropolis-Hastings (M-H) algorithm [14]. In the MCMC-semiBSL method, the M-H algorithm is also used. The basic idea is to generate samples from the target distribution by constructing a Markov sampling chain. The algorithm starts from an initial state and iteratively performs two steps: sampling from the proposal distribution and determining whether to accept the sample, until convergence is reached.

As can be seen from the above algorithm, the M-H algorithm requires the specification of both the proposal distribution and the initial state before execution. The convergence rate of the M-H algorithm depends on the choice of these two factors. When the proposal distribution is poorly chosen or the initial state devi-

ates from the true distribution, the convergence rate of the M-H algorithm can be slow, and it may even become trapped in local optima. To address this limitation of the M-H algorithm, Haario *et al.* introduced the Adaptive Monte Carlo (AM) algorithm in 2001 [15]. This algorithm continuously adjusts the proposal distribution based on the sampled values during execution, thus accelerating the convergence rate of the algorithm. In the classic M-H algorithm, a Gaussian proposal distribution is typically used as follows:

$$q\left(\cdot \mid X^{(t-1)}\right) = N\left(\cdot \mid X^{(t-1)}, \boldsymbol{C}\right) \tag{12}$$

where $\boldsymbol{C}$ is an appropriate covariance matrix. The AM algorithm adjusts the covariance matrix $\boldsymbol{C}$ continuously during the execution of the algorithm based on the sampling results. The specific adjustment method is as follows:

Assume that at time $t-1$, the state samples drawn by the algorithm are $X^{(0)}, \cdots, X^{(t-1)}$, where $X^{(0)}$ is the set initial state. The candidate state at the next time step is then drawn from the proposal distribution $q_t\left(\cdot \mid X^{(0)}, \cdots, X^{(t-1)}\right)$:

$$q_t\left(\cdot \mid X^{(0)}, \cdots, X^{(t-1)}\right) = N\left(\cdot \mid X^{(t-1)}, \boldsymbol{C}_t\right) \tag{13}$$

$$\boldsymbol{C}_t = s_d cov\left(X^{(0)}, \cdots, X^{(t-1)}\right) + s_d \varepsilon \boldsymbol{I}_d \tag{14}$$

$$cov\left(X^{(0)}, \cdots, X^{(k)}\right) = \frac{1}{k}\left(\sum_{i=1}^{k} X^{(i)} X^{(i)\mathrm{T}} - (k+1)\bar{X}^{(k)}\bar{X}^{(k)\mathrm{T}}\right) \tag{15}$$

$$\bar{X}^{(k)} = \left(\frac{1}{k+1}\right)\sum_{i=1}^{k} X^{(i)} \tag{16}$$

where $\boldsymbol{I}_d$ is the d-dimensional identity matrix, $\varepsilon > 0$ is a small constant chosen by the user, and s_d is a scaling parameter that depends only on the dimensionality d. In this paper, s_d is set to $s_d = 2.4^2/d$ as suggested by Gelman *et al.* [16].

Theorem 1. Let $\boldsymbol{\pi}$ be the density of a target distribution supported on a bounded measurable subset $S \in \mathbb{R}^d$, and assume that $\boldsymbol{\pi}$ is bounded from above. Let $\varepsilon > 0$ and let μ_0 be any initial distribution on S. Then the AM chain (X_n) simulates properly the target distribution $\boldsymbol{\pi}$: for any bounded and measurable function $f: S \to \mathbb{R}$, the equality

$$\lim_{n\to\infty} \frac{1}{n+1}\left(f(X_0) + f(X_1) + \cdots + f(X_n)\right) = \int_S f(x)\pi \mathrm{d}x \tag{17}$$

holds almost surely.

Theorem 2. Assume that the finite-dimensional distributions of the stochastic process $(X_n)_{n=0}^{\infty}$ on the state space S, where the sequence of generalized transition probabilities (K_n) is assumed to satisfy the following three conditions:

(i) There are a fixed integer k_0 and a constant $\lambda \in (0,1)$ such that

$$\delta\left(\left(K_{n,\tilde{y}_{n-2}}\right)^{k_0}\right) \le \lambda < 1, \text{ for all } \tilde{y}_{n-2} \in S^{n-1} \text{ and } n \ge 2 \tag{18}$$

(ii) There are a fixed probability measure π on S and a constant $c_0 > 0$ such that

$$\left\|\pi K_{n,\tilde{y}_{n-2}} - \pi\right\| \le \frac{c_0}{n}, \text{ for all } \tilde{y}_{n-2} \in S^{n-1} \text{ and } n \ge 2 \tag{19}$$

(iii) We have the following estimate for the operator norm

$$\left\| K_{n,\tilde{y}_{n-2}} - K_{n+k,\tilde{y}_{n+k-2}} \right\| \le c_1 \frac{k}{n} \tag{20}$$

where c_1 is a fixed positive constant, $n,k \ge 1$ and one assumes that $\tilde{y}_{n+k-2}$ is a direct continuation of $\tilde{y}_{n-2}$.

Then, if $f : S \to \mathbb{R}$ is bounded and measurable, then the equality (17) holds almost surely.

In what follows, the auxiliary constants $c_i, i = 2,3,\cdots$ depend on S, ε or C_o, and their actual value is irrelevant to our purposes here.

Proof of Theorem 1. According to Theorem 2, it suffices to prove that the AM chain satisfies conditions (i)-(iii). In order to check condition (i), we observe that, directly from definition and by the fact that S is bounded, all the covariances $C = C_n\left(y_0,\cdots,y_{n-1}\right)$ satisfy the matrix inequality

$$0 < c_2 I_d \le C \le c_3 I_d \tag{21}$$

Hence the corresponding normal densities $N_c\left(\cdot - x\right)$ are uniformly bounded from below on S for all $x \in S$.

We next verify condition (iii). To that end, we assume that $n \ge 2$ and observe that, for given $\tilde{y}_{n+k-2} \in S^{n+k-1}$, one has

$$\left\| K_{n,\tilde{y}_{n-2}} - K_{n+k,\tilde{y}_{n+k-2}} \right\| \le 2\sup_{y\in S} \left| K_{n,\tilde{y}_{n-2}}\left(y;A\right) - K_{n+k,\tilde{y}_{n+k-2}}\left(y;A\right) \right| \tag{22}$$

Fix $y \in S$ and introduce $R_1 = C_n\left(y_0,\cdots,y_{n-2},y\right)$ together with $R_2 = C_{n+k}\left(y_0,\cdots,y_{n+k-2},y\right)$. According to the definition,

$$\begin{aligned}
&\left| K_{n,\tilde{y}_{n-2}}\left(y;A\right) - K_{n+k,\tilde{y}_{n+k-2}}\left(y;A\right) \right| = \left| M_{R_1}\left(y;A\right) - M_{R_2}\left(y;A\right) \right| \\
&\le \left| \int_{x\in A} \left(N_{R_1} - N_{R_2}\right)\left(x-y\right) \min\left(1, \frac{\pi(x)}{\pi(y)}\right) m\mathrm{d}x \right. \\
&\quad \left. + \chi_A(x) \int_{x\in\mathbb{R}^d} \left(N_{R_1} - N_{R_2}\right)\left(x-y\right) \times \left[1 - \min\left(1, \frac{\pi(x)}{\pi(y)}\right)\right] m\mathrm{d}x \right| \\
&\le 2\int_{x\in\mathbb{R}^d} \left| N_{R_1}(z) - N_{R_2}(z) \right| \mathrm{d}z \le 2\int_{x\in\mathbb{R}^d} \mathrm{d}z \int_0^1 \mathrm{d}s \left| \frac{\mathrm{d}}{\mathrm{d}s} N_{R_1+s(R_2-R_1)}(z) \right| \\
&\le c_5 \left\| R_1 - R_2 \right\|
\end{aligned} \tag{23}$$

In general, $\left\| C_t - C_{t-1} \right\| \le \frac{c_6}{t}$, for $t > 1$. We easily see that $\left\| R_1 - R_2 \right\| \le c_7\left(S, C_0, \varepsilon\right)\frac{k}{n}$, and hence the previous estimates yield (iii).

In order to check condition (ii), fix $\tilde{y}_{n-2} \in S^{n-1}$ and denote $C^* = C_{n-1}\left(y_0,\cdots,y_{n-2}\right)$. It follows that $\left\| C^* - C_n\left(y_0,\cdots,y_{n-2},y\right) \right\| \le c_8/n$, where c_8 does not depend on $y \in S$. We may therefore proceed exactly as in (22) and (23) to deduce that

$$\left\| K_{n,\tilde{y}_{n-2}} - M_{C^*} \right\| \le \frac{c_9}{n} \tag{24}$$

Since $\pi M_{C^*} = \pi$, we obtain

$$\left\| \pi - \pi K_{n,\tilde{y}_{n-2}} \right\| = \left\| \pi\left(M_{C^*} - K_{n,\tilde{y}_{n-2}}\right) \right\| \le \frac{c_9}{n} \tag{25}$$

which completes the proof of Theorem 1.

2.5. Transformed Kernel Density Estimation

Traditional kernel density estimation does not perform well for heavy-tail distributions. To address this issue, Wand *et al.* proposed a kernel density estimation method based on the Box-Cox transformation in 1991 [17]. The basic idea of the method is as follows: First, transform the original data so that the transformed data becomes more uniform. Then, apply traditional kernel density estimation to the transformed data. Finally, the resulting density function is inverted and transformed back to the original data's kernel density estimate. Based on this concept, this paper extends the transformed kernel density estimation to the framework of BSL.

In 1992, Park *et al.* [18] demonstrated the effectiveness of this method through numerical experiments. Since then, many studies have proposed different transformation methods, such as the power transformation method proposed by Bolance *et al.* [19], the Mobius-like mapping proposed by Clements *et al.* [20], the Champernowne transformation proposed by Buch *et al.* [21], and the inverse Beta(3,3) transformation proposed by Bolance *et al.* [22].

Among these, Bolance *et al.* introduced a quadratic transformation method that combines the Champernowne distribution function and the inverse Beta(3,3) function. This method first approximates the sample data distribution to a uniform distribution using the Champernowne distribution function, and then transforms the data, which has already been transformed once, to approximately follow a Beta(3,3) distribution using the inverse Beta(3,3) distribution function. This method offers a significant advantage over other methods because of the theory proven by Terrell and Scott in 1985 [23], which states that the Beta(3,3) density function minimizes the integrated mean squared error of kernel density estimation within the function space $\mathcal{A}=\left\{g: g \text{ is a density function and } g(x)=0,|x|\geq 1\right\}$. The kernel density estimation method adopted in this paper is an improved method of this quadratic transformation—the Beta kernel density estimation under the generalized Logistic transformation. This method requires fewer parameters and improves upon the original method by addressing the "boundary bias" issue. The specific method is as follows:

First, the original data is approximately transformed into a uniform distribution on the interval [0, 1]. To achieve this, the transformation function is set to be the cumulative distribution function (CDF) of the original data. However, the true distribution function is unknown, so a generalized Logistic function is used to estimate the empirical distribution function. The expression of the generalized Logistic function is as follows:

$$Y=\frac{1}{\left(1+\lambda\gamma \mathrm{e}^{-\alpha X}\right)^{\frac{1}{\lambda}}}\left(\alpha,\gamma>0;\lambda\geq 0\right) \tag{26}$$

where $\boldsymbol{\pi}=(\alpha,\gamma,\lambda)$ represents the parameter that needs to be estimated.

Next, the sample data $X_1,\cdots,X_n$ are arranged in ascending order as order statistics $X_{(1)},\cdots,X_{(n)}$, and the values of the empirical distribution function at $X_{(1)},\cdots,X_{(n)}$ are calculated as $Y_{(1)},\cdots,Y_{(n)}$. A regression is performed on the pairs

$\left(X_{(1)},Y_{(1)}\right),\cdots,\left(X_{(n)},Y_{(n)}\right)$, and the parameter estimates $\boldsymbol{\pi}=(\alpha,\gamma,\lambda)$ are obtained by least squares. Using the generalized Logistic function as an estimate for the distribution function $\hat{F}_{\hat{\boldsymbol{\pi}}}(x)=\dfrac{1}{\left(1+\hat{\lambda}\hat{\gamma}\mathrm{e}^{-\hat{\alpha}x}\right)^{1/\hat{\lambda}}}$, the original data is transformed as $Y_i=\hat{F}_{\hat{\boldsymbol{\pi}}}(X_i), i=1,\cdots,n$.

Then, a Beta kernel density estimate is applied to the transformed sample:

$$\hat{f}_Y(y)=\frac{1}{n}\sum_{i=1}^{n}K_{\frac{y}{h}+1,\frac{1-y}{h}+1}(Y_i) \tag{27}$$

where $K_{p,q}(\cdot)$ is the probability density function of the Beta(p, q) distribution, and h is the bandwidth.

Finally, the kernel density estimate of the transformed data is converted into the kernel density estimate of the original data as follows:

$$\hat{f}_X(x)=\left|\hat{F}'_{\hat{\boldsymbol{\pi}}}\right|\hat{f}_Y\left(\hat{F}_{\hat{\boldsymbol{\pi}}}(x)\right) \tag{28}$$

The AM algorithm and transformed kernel density estimation are incorporated into semiBSL, and the new method is named AM-semiTBSL. The detailed algorithmic process is shown in **Algorithm 3**.

Algorithm 3: AM-semiTBSL algorithm

Input: Summary statistic of the data, $\boldsymbol{s}_y$, the prior distribution, $\boldsymbol{p}(\boldsymbol{\theta})$, the initial proposal distribution q_0, the Initial covariance matrix, $\boldsymbol{C}_0$, the number of iterations, T, the number of simulation runs, n, the initial value of the chain $\boldsymbol{\theta}^{(0)}$.

Output: AM sample $\left(\boldsymbol{\theta}^{(0)},\cdots,\boldsymbol{\theta}^{(T)}\right)$ from the BSL posterior, $p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}\right)$. Some samples can be discarded as burn-in if required.

Simulate $\boldsymbol{x}_{1:n}\sim p\left(\cdot\mid\boldsymbol{\theta}^{(0)}\right)$ and compute $\boldsymbol{s}_{1:n}$
Compute $\hat{f}^{(0)}\left(\boldsymbol{s}_y\right)$ and $\hat{F}^{(0)}\left(\boldsymbol{s}_y\right)$ using (28)
Compute $\hat{\boldsymbol{R}}^{(0)}$ using (11)
Compute $p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(0)}\right)$ using (10)
for $i=1,\cdots,T$ do
    Draw $\boldsymbol{\theta}^{*}\sim q\left(\cdot\mid\boldsymbol{\theta}^{(i-1)}\right)$
    Compute $\boldsymbol{C}_i$ using (14)
    Compute q_i using (13)
    Simulate $\boldsymbol{x}^{*}_{1:n}\sim p\left(\cdot\mid\boldsymbol{\theta}^{*}\right)$ and compute $\boldsymbol{s}^{*}_{1:n}$
    Compute $\hat{f}^{*}\left(\boldsymbol{s}_y\right)$ and $\hat{F}^{*}\left(\boldsymbol{s}_y\right)$ using (28)
    Compute $\hat{\boldsymbol{R}}^{*}$ using (11)
    Compute $p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{*}\right)$ using (10)
    Compute $r=\min\left(1,\dfrac{p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{*}\right)p\left(\boldsymbol{\theta}^{*}\right)q_i\left(\boldsymbol{\theta}^{(i-1)}\mid\boldsymbol{\theta}^{*}\right)}{p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i-1)}\right)p\left(\boldsymbol{\theta}^{(i-1)}\right)q_i\left(\boldsymbol{\theta}^{*}\mid\boldsymbol{\theta}^{(i-1)}\right)}\right)$
    if $U(0,1)<r$ then
        Set $\boldsymbol{\theta}^{(i)}=\boldsymbol{\theta}^{*}, p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i)}\right)=p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{*}\right)$
    else
        Set $\boldsymbol{\theta}^{(i)}=\boldsymbol{\theta}^{(i-1)}, p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i)}\right)=p\left(\boldsymbol{s}_y\mid\boldsymbol{\theta}^{(i-1)}\right)$
    end
end

3. Numerical Experiment

The performance of MCMC-semiBSL, AM-semiBSL, MCMC-semiTBSL, and AM-semiTBSL is compared through numerical simulations. All numerical simulations are performed on a computer configured with an AMD Ryzen 7 5800H 3.20 GHz processor and 16.0 GB of memory, using Matlab (R2021a).

To compare the estimation performance of different methods, the Total Variation Distance is used to measure the difference between the true posterior distribution and the posterior distribution estimated by the methods. Given two distributions, P and Q, the definition of the Total Variation Distance is as follows:

$$d_{tv}(P,Q)=\frac{1}{2}\int\left|p(x)-q(x)\right|\mathrm{d}x \tag{29}$$

where $p(x)$ and $q(x)$ are the probability density functions of distributions P and Q, respectively. $d_{tv}\in[0,1]$, $d_{tv}=0$ when the two distributions are identical, and $d_{tv}=1$ when the two distributions do not overlap at all. The Total Variation Distance is symmetric and satisfies the triangle inequality. By comparing the Total Variation Distance between the estimated posterior distribution of different methods and the true posterior distribution, the estimation performance of the different methods can be effectively assessed.

For the stable distributions provided in Section 2.1, the true parameter values are set as $\boldsymbol{\theta}=(\alpha,\beta,\sigma,\mu)=(0.7,0.5,1,0)$, and the observation data $\boldsymbol{y}=(y_1,\cdots,y_{50})$ are generated based on the model and parameters. In this numerical simulation, no dimensionality reduction of the original observed data into summary statistics is performed, and the complete observed data are directly treated as summary statistics. The initial values for the parameters of the four methods are set as $\boldsymbol{\theta}^{(0)}=(\alpha,\beta,\sigma,\mu)=(2,-1,1,0)$. Three sets of experiments are conducted for each method, with iteration numbers set as: $T=$ 10,000, $T=$ 20,000, and $T=$ 50,000. In each iteration, $n=$ 1000 simulated samples are generated.

Figure 1 shows the comparison of the estimated distributions and the true distribution under the four different iteration numbers, and Table 1 presents the Total Variation Distance between the estimated distributions and the true distribution for each case. The numerical simulation results indicate that: 1) As the number of iterations increases, the estimation accuracy of all four methods improves. 2) The AM algorithm effectively enhances the convergence speed, and under the same number of iterations, the two methods using the AM algorithm generally outperform the methods using MCMC sampling. 3) The use of kernel density estimation based on quadratic transformations can significantly improve the estimation accuracy, with MCMC-semiTBSL outperforming MCMC-semiBSL, and AM-semiTBSL outperforming AM-semiBSL. 4) Among the four methods, AM-semiTBSL achieves the best performance in terms of both accuracy and convergence speed, followed by MCMC-semiTBSL, then AM-semiBSL, and finally MCMC-semiBSL.

Table 1. The total variation distance between the results of different methods and the true distribution.

The number of iterations	Methods			
	MCMC-semiBSL	AM-semiBSL	MCMC-semiTBSL	AM-semiTBSL
T = 10,000	0.3027	0.3894	0.3427	0.1473
T = 20,000	0.2984	0.2829	0.2854	0.0362
T = 50,000	0.2115	0.2038	0.1306	0.0151

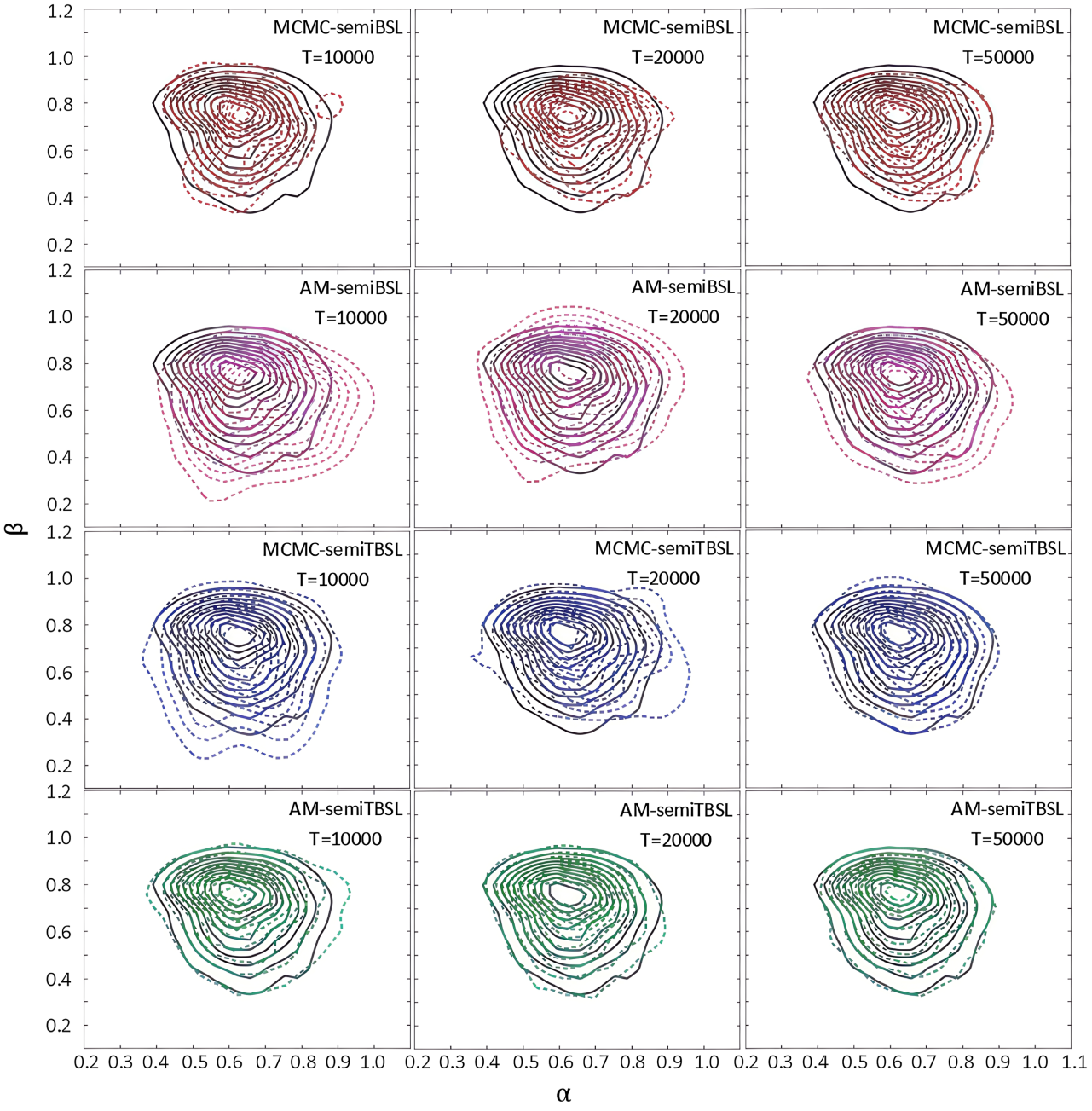

Figure 1. Numerical simulation results of different methods.

4. Empirical Analysis

The real data used in this section is sea clutter data collected by the IPIX radar at McMaster University in Canada. The analysis focuses on the first 10 distance bins from group #269, with a total of 1,310,720 samples.

Table 2 presents the basic statistical summary of the sea clutter sample data, where the skewness is −0.0252, indicating a slight "left skew" in the sample data. The kurtosis is 4.7240, which is greater than 3, indicating that the sample data is "peaked." This is also reflected in the normal Q-Q plot shown on the right side of Figure 2. These characteristics suggest that the sea clutter data exhibits non-Gaussian noise, making it suitable for modeling using stable distributions.

Table 2. The basic statistics of sea clutter data.

Mean	Standard deviation	Skewness	Kurtosis	Maximum value	Minimum value
2.7798×10^{-11}	1.0000	−0.0252	4.7240	5.8344	−6.7303

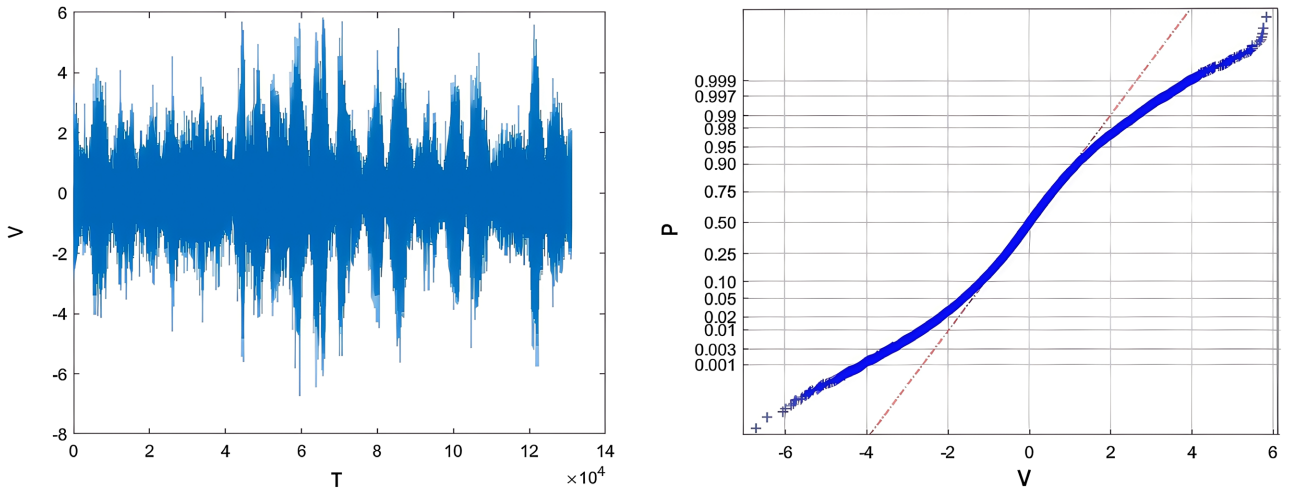

Figure 2. Time series plot and normal Q-Q plot of sea clutter data.

Stable distributions are used to model the sea clutter sample data, with the model parameters estimated using four methods: MCMC-semiBSL, AM-semiBSL, MCMC-semiTBSL, and AM-semiTBSL. To better compare the performance of the different methods and eliminate external factors, all four methods are initialized with the same parameter values $\boldsymbol{\theta}^{(0)} = (\alpha, \beta, \sigma, \mu) = (2, -1, 1, 0)$, iteration number T = 50,000, and model fitting sample size n = 1500. The Total Variation Distance is used to assess the estimation performance of the different methods.

From the results shown in Figure 3 and Table 3, it is evident that modeling radar clutter data using stable distributions is feasible. Furthermore, under the same number of iterations, the AM-semiTBSL method provides the best parameter estimates for the stable distribution. This demonstrates that incorporating the AM algorithm and transformed kernel density estimation effectively enhances the convergence speed and estimation accuracy of MCMC-semiBSL.

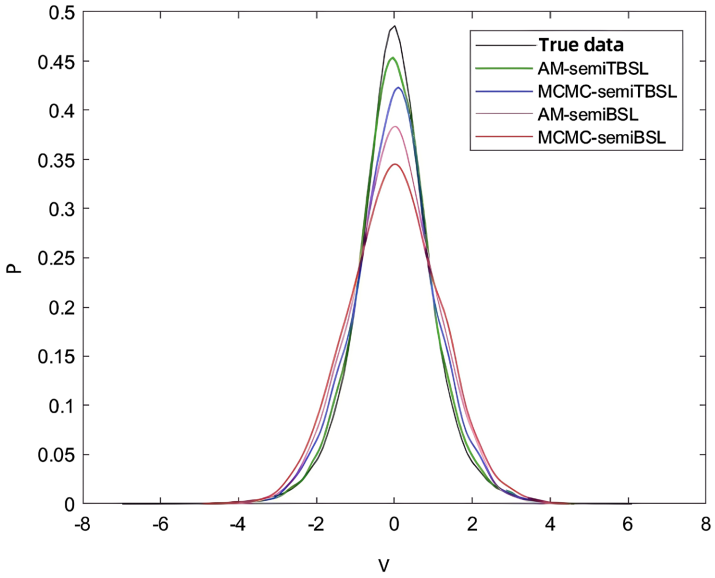

Figure 3. The estimated distributions of different methods and the true distribution.

Table 3. The total variation distance between the estimated distributions of different methods and the true distribution.

	MCMC-semiBSL	AM-semiBSL	MCMC-semiTBSL	AM-semiTBSL
d_{tv}	0.3612	0.2971	0.1639	0.0854

5. Conclusion

This paper addresses the issue of poor performance of semi-parametric Bayesian synthetic likelihood when handling data with heavy-tail by introducing transformed kernel density estimation. It also tackles the slow convergence of MCMC sampling and the tendency to fall into local optima within the semi-parametric Bayesian synthetic likelihood framework by incorporating the Adaptive Monte Carlo (AM) algorithm. Subsequently, numerical experiments based on stable distributions were conducted. The results show that the transformed kernel density estimation improves the estimation accuracy of the semi-parametric Bayesian synthetic likelihood, while the Adaptive Monte Carlo algorithm enhances its convergence speed. Finally, empirical analysis on a real dataset, the sea clutter data, further confirms the superiority of the proposed improvements.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Sisson, S.A., Fan, Y. and Tanaka, M.M. (2007) Sequential Monte Carlo without Likelihoods. *Proceedings of the National Academy of Sciences*, **104**, 1760-1765. https://doi.org/10.1073/pnas.0607208104

[2] Price, L.F., Drovandi, C.C., Lee, A. and Nott, D.J. (2017) Bayesian Synthetic Likelihood. *Journal of Computational and Graphical Statistics*, **27**, 1-11. https://doi.org/10.1080/10618600.2017.1302882

[3] Picchini, U. and Forman, J.L. (2019) Bayesian Inference for Stochastic Differential Equation Mixed Effects Models of a Tumour Xenography Study. *Journal of the Royal Statistical Society Series C*: *Applied Statistics*, **68**, 887-913. https://doi.org/10.1111/rssc.12347

[4] Maraia, R., Springer, S., Härkönen, T., Simon, M. and Haario, H. (2023) Bayesian Synthetic Likelihood for Stochastic Models with Applications in Mathematical Finance. *Frontiers in Applied Mathematics and Statistics*, **9**, Article 1187878. https://doi.org/10.3389/fams.2023.1187878

[5] Moriña, D., Fernández-Fontelo, A., Cabaña, A., Arratia, A. and Puig, P. (2023) Estimated Covid-19 Burden in Spain: ARCH Underreported Non-Stationary Time Series. *BMC Medical Research Methodology*, **23**, Article No. 75. https://doi.org/10.1186/s12874-023-01894-9

[6] Fasiolo, M., Wood, S.N., Hartig, F. and Bravington, M.V. (2018) An Extended Empirical Saddlepoint Approximation for Intractable Likelihoods. *Electronic Journal of Statistics*, **12**, 1544-1578. https://doi.org/10.1214/18-ejs1433

[7] An, Z., Nott, D.J. and Drovandi, C. (2019) Robust Bayesian Synthetic Likelihood via a Semi-Parametric Approach. *Statistics and Computing*, **30**, 543-557. https://doi.org/10.1007/s11222-019-09904-x

[8] An, Z., South, L.F. and Drovandi, C. (2022) BSL: An R Package for Efficient Parameter Estimation for Simulation-Based Models via Bayesian Synthetic Likelihood. *Journal of Statistical Software*, **101**, 1-33. https://doi.org/10.18637/jss.v101.i11

[9] Picchini, U., Simola, U. and Corander, J. (2023) Sequentially Guided MCMC Proposals for Synthetic Likelihoods and Correlated Synthetic Likelihoods. *Bayesian Analysis*, **18**, 1099-1129. https://doi.org/10.1214/22-ba1305

[10] Silverman, B.W. (1986) Density Estimation for Statistics and Data Analysis. CRC Press.

[11] Boudt, K., Cornelissen, J. and Croux, C. (2011) The Gaussian Rank Correlation Estimator: Robustness Properties. *Statistics and Computing*, **22**, 471-483. https://doi.org/10.1007/s11222-011-9237-0

[12] Hastings, W.K. (1970) Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, **57**, 97-109. https://doi.org/10.1093/biomet/57.1.97

[13] Haario, H., Saksman, E. and Tamminen, J. (2001) An Adaptive Metropolis Algorithm. *Bernoulli*, **7**, 223-242. https://doi.org/10.2307/3318737

[14] Gelman, A., Roberts, G.O. and Gilks, W.R. (1996) Efficient Metropolis Jumping Rules. In: Bernardo, J.M., *et al.*, Eds., *Bayesian Statistics 5*, Oxford University Press, 599-608. https://doi.org/10.1093/oso/9780198523567.003.0038

[15] Haario, H., Saksman, E. and Tamminen, J. (2001) An adaptive Metropolis Algorithm. *Bernoulli*, **7**, 223-242. https://doi.org/10.2307/3318737

[16] Gelman, A., Carlin, J.B., Stern, H.S., *et al.* (2004) Bayesian Data Analysis. 2nd Edition, CRC Press.

[17] Wand, M.P., Marron, J.S. and Ruppert, D. (1991) Transformations in Density Estimation. *Journal of the American Statistical Association*, **86**, 343-353. https://doi.org/10.1080/01621459.1991.10475041

[18] Park, B.U., Chung, S.S. and Seog, K.H. (1992) An Empirical Investigation of the Shifted Power Transformation Method in Density Estimation. *Computational Statis-*

tics & Data Analysis, **14**, 183-191. https://doi.org/10.1016/0167-9473(92)90172-c

[19] Bolancé, C., Guillen, M. and Nielsen, J.P. (2003) Kernel Density Estimation of Actuarial Loss Functions. *Insurance*: *Mathematics and Economics*, **32**, 19-36. https://doi.org/10.1016/s0167-6687(02)00191-9

[20] Clements, A., Hurn, S. and Lindsay, K. (2003) Mobius-Like Mappings and Their Use in Kernel Density Estimation. *Journal of the American Statistical Association*, **98**, 993-1000. https://doi.org/10.1198/016214503000000945

[21] Buch-larsen, T., Nielsen, J.P., Guillén, M. and Bolancé, C. (2005) Kernel Density Estimation for Heavy-Tailed Distributions Using the Champernowne Transformation. *Statistics*, **39**, 503-516. https://doi.org/10.1080/02331880500439782

[22] Bolancé, C. (2010) Optimal Inverse $\beta(3, 3)$ Transformation in Kernel Density Estimation. *Sort-Statistics and Operations Research Transactions*, **34**, 223-238.

[23] Terrell, G.R. and Scott, D.W. (1985) Oversmoothed Nonparametric Density Estimates. *Journal of the American Statistical Association*, **80**, 209-214. https://doi.org/10.1080/01621459.1985.10477163