

# UNCERTAINTY QUANTIFICATION WITH $\alpha$ -STABLE-PROCESS MODELS

Rui Tuo

*Chinese Academy of Sciences*

*Abstract:* In this article we consider using a class of  $\alpha$ -stable processes, which can be regarded as generalizations of the Gaussian processes, as the surrogate models for uncertainty quantification. We introduce a class of  $\alpha$ -stable processes, whose finite-dimensional distributions can be represented using independent stable random variables. This representation allows for Bayesian inference for the proposed statistical model. We can obtain the posterior distributions for the untried points as well as the model parameters through an MCMC algorithm. The computation for the representation requires some geometrical information given by the design points. We propose an efficient algorithm to solve this computational geometry problem. Two examples are given to illustrate the proposed method and its potential advantages.

*Key words and phrases:* Computer experiments, kriging, Lévy processes, stable distributions.

## 1. Introduction

In a computer experiment or uncertainty quantification problem, a complex computer model is used to simulate a physical process. The computer model may contain several control or noise factors as input variables, and the goal is to understand the function relationship between the computer input and the output. We refer to Santner, Williams and Notz (2003) for detailed discussion. A common scenario is that the computer model is so complex that each run of the code is costly. In this case, one has to reconstruct the function based on the computer outputs over a selected set of points in the input space. The reconstruction of the underlying target function is referred to as surrogate modeling, which is one of the core topics in computer experiments and uncertainty quantification.

Gaussian process models are an important class of surrogate models with statistical interpretation. Given a set of scattered points and the value of an underlying function of interest over these points, a Gaussian process model can give a prediction for the function value at an untried point, as well as its confidence interval. We refer to Rasmussen and Williams (2006); Santner, Williams and Notz

(2003); Banerjee, Carlin and Gelfand (2004) for the theory and implementation of Gaussian process modeling.

Gaussian process models perform well in fitting smooth functions. However, for functions with sudden jumps or oscillatory spikes, Gaussian process models can result in unstable prediction (Gramacy and Lee (2008); Chen, Wang and Wu (2010)). Furthermore, the predictive distributions given by a Gaussian process model are normal only, which may be too narrow to characterize all types of prediction uncertainties in reality. In this work, we focus on a broader family of stochastic processes that allows for conditional inference as Gaussian processes. The marginal distributions of these stochastic processes follow the stable distributions, which form a broader family than the normal.

We identify a class of stationary  $\alpha$ -stable random fields whose finite dimensional distributions can be represented using independent stable random variables. This specialization of  $\alpha$ -stable processes allows a feasible computational scheme for statistical inference. As a spatial model, the location of the input points characterizes the distributions of the independent stable random variables. We show that the derivation of these distributions reduces to solving a computational geometry problem. We propose an efficient algorithm to solve this problem. Based on this  $\alpha$ -stable process family, we propose statistical models for scattered spatial data. A Bayesian approach is utilized for the estimation and inference of the proposed model.

This paper is organized as follows. We define a class of  $\alpha$ -stable random field and study its probabilistic structure and properties in Section 2. In Section 3, we develop a Bayesian approach for estimation and inference in the proposed models. An efficient algorithm is presented in Section 4 to solve the computational geometry problem raised in Section 2.3. Numerical illustrations show the potential advantages in Section 5. Concluding remarks are made in Section 6. The technical proofs are given in Appendix A.

## 2. A Special Class of $\alpha$ -Stable Processes

In this section we state a new class of  $\alpha$ -stable processes and discuss their probabilistic structure and properties. A novel spatial statistical model is proposed based on this stochastic process family. First, we give a quick review of stable random variables and  $\alpha$ -stable processes in Section 2.1 and 2.2. For a detailed introduction to  $\alpha$ -stable random variables and vectors, stable integrals, and  $\alpha$ -stable processes, we refer to Samorodnitsky, Taqqu and Linde (1996).

**2.1. Stable random variables**

**Definition 1** (Stable Distributions). *A random variable  $X$  is said to have a stable distribution if for any positive number  $A$  and  $B$ , there is a positive number  $C$  and a real number  $D$  such that*

$$AX_1 + BX_2 \stackrel{d}{=} CX_3 + D,$$

where  $X_1, X_2$ , and  $X_3$  are independent copies of  $X$ , and “ $\stackrel{d}{=}$ ” denotes equality in distribution.

It can be proven that stable distributions form a four-parameter family (Samorodnitsky, Taqqu and Linde (1996)), denoted by  $S_\alpha(c, \beta, \mu)$ , where  $\alpha \in (0, 2]$  is the index parameter,  $\beta \in [-1, 1]$  the skewness parameter,  $\mu \in \mathbf{R}$  the shift parameter, and  $c \in (0, +\infty)$  the scale parameter. In general, the probability density of a stable distribution does not have an analytic expression.

A stable distribution with  $\alpha = 2$  is a normal distribution with mean  $\mu$  and standard deviation  $c$ . The parameter  $\beta$  has no effect in this case, although we usually set  $\beta = 0$  for convenience. For a non-Gaussian stable distribution, the parameters  $\alpha, \beta, c$  and  $\mu$  are unique. Stable distributions with  $\alpha < 2$  are heavy-tailed distributions and have infinite variance. The smaller  $\alpha$  is, the slower its tail decays. The skewness parameter  $\beta$  is a measure of asymmetry when  $\alpha < 2$ . When  $\beta = 0$ , the distribution is symmetric about  $\mu$ . The Cauchy distributions are stable distributions with  $\alpha = 1$  and  $\beta = 0$ .

The support of a stable random variable  $X$  is the whole real line unless  $\alpha < 1$  and  $\beta = \pm 1$ . When  $\alpha < 1$  and  $\beta = 1$ ,  $X \in [\mu, +\infty)$ ; when  $\alpha < 1$  and  $\beta = -1$ ,  $X \in (-\infty, \mu]$ .

Stable random variables can be generated using the numerical method proposed in Chambers, Mallows and Stuck (1976). An algorithm is given by Nolan (1997) to compute the probability density function of a stable distribution. Using this algorithm, maximum likelihood estimates for  $\alpha, \beta, c$ , and  $\mu$  can be obtained and the estimators are asymptotically normally distributed (Nolan (2001)).

**2.2. Convolution of  $\alpha$ -stable random measure**

Every stationary Gaussian process on  $\mathbf{R}^m$  can be represented by a convolution of the Gaussian random measure. Let  $G(\mathbf{x})$  be a stationary Gaussian process with mean 0. Then there exists a function  $K(\mathbf{x}) \in L^2(\mathbf{R}^m)$  such that

$$G(\mathbf{x}) = \int K(\mathbf{t} - \mathbf{x})W(d\mathbf{t}), \tag{2.1}$$

where  $W(\cdot)$  is a random measure satisfying

(C1): For any mutually disjoint measurable sets  $D_1, \dots, D_n$ , the random variables  $W(D_1), \dots, W(D_n)$  are independent;

(C2): If  $m(D)$  be the Lebesgue measure of set  $D$ ,

$$W(D) \sim N(0, m(D)).$$

If we change the distribution in condition C2 to a non-Gaussian distribution, we can use (2.1) to define a stationary non-Gaussian process. For many non-Gaussian infinitely divisible distributions, such a random measure exists. Here we consider the  $\alpha$ -stable random measures, which form a large family, including the Gaussian random measures.

**Definition 2** ( $\alpha$ -Stable Random Measure). *A random measure  $M(\cdot)$  is called  $\alpha$ -stable with skewness intensity  $\beta(\mathbf{x})$  if condition C1 is satisfied and*

$$M(D) \sim S_\alpha\left((m(D))^{1/\alpha}, \frac{\int_D \beta(\mathbf{x})m(d\mathbf{x})}{m(D)}, 0\right), \quad (2.2)$$

where  $0 < \alpha \leq 2$  and  $|\beta(x)| \leq 1$ .

For convenience, we assume the skewness intensity  $\beta(x)$  is a constant, denoted by  $\beta$ . A more general setting can also be considered by imposing a parametric form for the skewness intensity  $\beta(x)$ . The statistical analysis can be conducted in a similar manner. If  $\alpha < 1$  and  $\beta = \pm 1$ , the support of the stable distribution is a half real line. Bayesian inference may be inconvenient if the support of the density functions depends on the parameters. To avoid this problem, we assume  $|\beta| < 1$  in this paper.

Let  $H(\mathbf{x}) = \int K(\mathbf{t} - \mathbf{x})M(d\mathbf{t})$ , where  $M(\cdot)$  is defined in (2.2). Then  $H(\cdot)$  is called an  $\alpha$ -stable stochastic process. This process is also stationary since  $M(\cdot)$  is invariant under any shift transformation. If we let  $\alpha = 2$  and  $\beta = 0$ , this process reduces to a stationary Gaussian process. The finite-dimensional distribution of an  $\alpha$ -stable process is called a multivariate  $\alpha$ -stable distribution.

Now we turn to the conditional inference problem. Suppose we have  $n$  inputs  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  and observe  $Y(\mathbf{x}_i)$  for  $i = 1, \dots, n$ . To establish a new statistical model using the conditional inference of the  $\alpha$ -stable processes, we assume the observations are from a linear transformation of the stationary  $\alpha$ -stable process  $H(\mathbf{x})$ ,

$$Y(\mathbf{x}) = \mu + cH(\mathbf{x}), \quad (2.3)$$

where  $c > 0$ . We can also change the constant  $\mu$  to a regression function, like the universal kriging for Gaussian process models (Santner, Williams and Notz (2003)).

Unlike that for a Gaussian process, the conditional inference for a stable process is intractable for a general kernel function  $K$ , since the density and conditional distributions of an  $\alpha$ -stable random vector are difficult to obtain. Karcher, Shmileva and Spodarev (2013) considers the linear predictors for  $\alpha$ -stable processes given the observational data. There is a lack of a framework for the likelihood-based or the Bayesian conditional inference for the  $\alpha$ -stable processes. The statistical inference for the max-stable processes, a class of relevant stochastic processes, is established. See Padoan, Ribatet and Sisson (2010) and the references therein. However, the existing work on max-stable processes cannot be adapted to the inference for the  $\alpha$ -stable processes because the parametrization scheme for the max-stable processes is not applicable for the  $\alpha$ -stable processes. In order to make the statistical inference feasible, we have to sacrifice some flexibility in  $K$ . In the next section we introduce a specific family of kernel functions and represent the finite-dimensional distributions of the corresponding  $\alpha$ -stable random fields using independent stable random variables.

### 2.3. Hyper-rectangle partition

For the rest of the paper we restrict  $K$  to be an axis-parallel hyper-rectangle with its center at 0,

$$K(\mathbf{x}) = I(|x_i| \leq d_i, i = 1, \dots, m), \tag{2.4}$$

where  $x_i$  is the  $i$ th entry of  $\mathbf{x}$ ,  $d_i$ 's are model parameters with  $d_i > 0$  for each  $i$ . The stationary  $\alpha$ -stable process of interest is still given by

$$H(\mathbf{x}) = \int K(\mathbf{t} - \mathbf{x})M(d\mathbf{t}). \tag{2.5}$$

**Remark 1.** For mathematical rigor we should use  $K(\mathbf{x}) = I(-d_i < x_i \leq d_i, i = 1, \dots, m)$  instead of (2.4) in order to make the stochastic process right-continuous, but the notation in (2.4) is more convenient. We set the center of the rectangle to be 0 also for convenience. Actually the location of the center has no effect on the probability structure of  $H(\cdot)$ , since  $M(\cdot)$  is invariant under a shift transformation.

**Remark 2.** If we let  $\alpha = 2$  and  $\beta = 0$ , (2.5) reduces to a Gaussian process with the correlation function

$$C(\mathbf{x}) = \frac{\prod_{i=1}^m (d_i - |x_i|)_+}{\prod_{i=1}^m d_i},$$

where  $x_i$  is the  $i$ th component of  $\mathbf{x}$ , and  $x_+ := xI(x > 0)$  for any  $x \in \mathbf{R}$ . This correlation function has a compact support  $\{\mathbf{x} : |x_i| \leq d_i\}$ . This correlation family has some similarity with more familiar ones. For instance, consider the

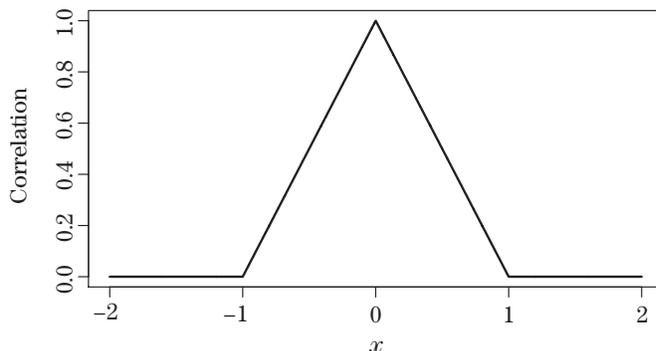


Figure 1. Correlation function for corresponding Gaussian process.

power exponential correlation family with  $C(\mathbf{x}) = \exp\{-\sum_{i=1}^m \theta_i x_i^p\}$  where  $\theta_i > 0$  are correlation parameters. In our model the correlation becomes stronger as  $d_i$  increases. Thus  $d_i$  plays a similar role as the correlation parameter  $\theta_i$  of the power exponential family. Figure 1 plots its one-dimensional version. From the figure we can see the non-zero part of this one-dimensional function forms a triangle and this correlation function is thus known as a triangular autocorrelation function (Stein (1999)).

We study the finite-dimensional distribution of (2.5). Denote the  $n$  inputs by  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . Let  $A_i$  be the rectangle region defined by the indicator function  $\mathbf{1}_{A_i}(\mathbf{x}) := K(\mathbf{x} - \mathbf{x}_i)$  for  $i = 1, \dots, n$ . By (2.2), (2.4), and (2.5) we have  $H(\mathbf{x}_i) \sim S_\alpha((m(A_i))^{1/\alpha}, \beta, 0)$ .

Since the  $m(A_i)$ 's are equal, the  $H(x_i)$ 's have the same marginal distribution. This property can also be shown by the stationarity of  $H$ . The joint distribution of  $(H(\mathbf{x}_1), \dots, H(\mathbf{x}_n))$  depends on the location of  $A_1, \dots, A_n$ . In order to study this spatial relationship, we need to consider the partition of  $\mathbf{R}^m$  generated by  $A_1, \dots, A_n$ .

Let  $\mathcal{A}$  be the algebra generated by  $A_1, \dots, A_n$ . Let  $\{I_0, I_1, \dots, I_s\}$  be the set of atoms in  $\mathcal{A}$ . If  $E \in \mathcal{A}$  and  $E \subset I_i$  then  $E = I_i$  for  $i = 0, \dots, s$ . Now we have the partition  $\mathbf{R}^m = \sum_{i=0}^s I_s$ , where  $\sum$  represents the disjoint union. Only one atom in  $\mathcal{A}$  is unbounded, which is  $\mathbf{R}^m - \cup_{i=1}^n A_i$ . Without loss of generality, we suppose this unbounded region is  $I_0$ . Thus  $I_1, \dots, I_s$  all have finite volumes. Figure 2 gives a 2D illustrating example of this hyper-rectangle partition.

We call  $I_i$ 's the *subregions* for  $i \geq 1$ . We say a subregion  $I$  belongs to a rectangle  $A$  if  $I \subseteq A$ .

Since  $A_i$ 's are all axis-parallel rectangles, it is easily shown that each  $I_i$  can

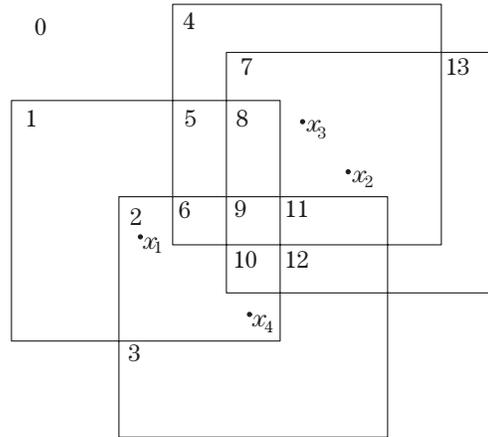


Figure 2. Example of hyper-Rectangle partition.

be represented by the disjoint union of some axis-parallel rectangles. In Section 4 we propose an effective algorithm for computing  $s$  and  $m(I_i)$  for each  $i$ , and in the following discussion we assume that these values are known.

Since  $I_i$ 's are disjoint, by condition C1 we have a representation for the joint distribution of  $(H(\mathbf{x}_1), \dots, H(\mathbf{x}_n))$ :

$$H(\mathbf{x}_i) = \sum_{I_j \subset A_i} v_j, i = 1, \dots, n, \tag{2.6}$$

where  $\{v_j\}_{j=1}^s$  are independent random variables and

$$v_j \sim S_\alpha((m(I_j))^{1/\alpha}, \beta, 0) \tag{2.7}$$

for  $j = 1, \dots, s$ .

### 2.4. Continuity

In this section we study the continuity of the proposed  $\alpha$ -stable processes. To this end, we use an existing result in the theory of Lévy processes.

For any  $l \in \{1, \dots, n\}$ , let

$$H_{x_1, \dots, x_{l-1}, x_{l+1}, x_n}(x) = H(x_1, \dots, x_{l-1}, x, x_{l+1}, x_n),$$

where  $x_1, \dots, x_{l-1}, x_{l+1}, x_n$  are constants. Let  $L(x)$  be the  $\alpha$ -stable Lévy process with  $L(x) \sim S_\alpha((xm(A_i)/d_l)^{1/\alpha}, \beta, 0)$  for  $x \geq 0$ , where  $d_l$  is defined in (2.4). We refer to Applebaum (2009) for more discussions. Because a Lévy process has stationary increments,  $L(x) - L(x - d_l) \stackrel{d}{=} L(d_l) \sim S_\alpha((m(A_i))^{1/\alpha}, \beta, 0)$ . Moreover, if  $\delta L(x) = L(x) - L(x - d_l)$ , then we can verify that

$$H_{x_1, \dots, x_{l-1}, x_{l+1}, x_n}(\cdot) \stackrel{d}{=} \delta L(\cdot).$$

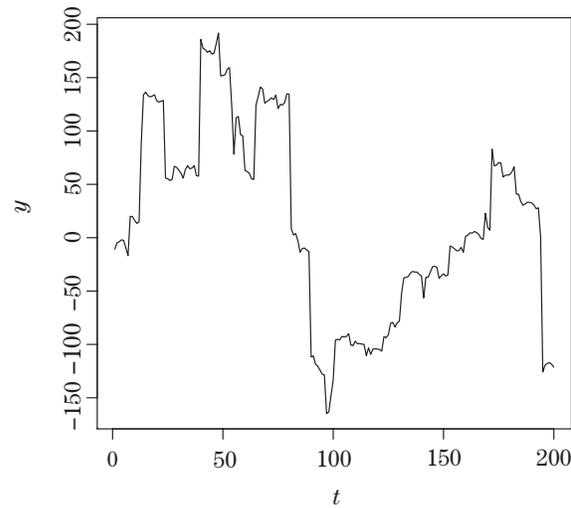


Figure 3. Realization of a stationary Cauchy process.

It is known that the sample path of a Lévy process does not have any other continuous component except a Brownian Motion with drift. Thus the sample path of  $Y(\mathbf{x})$  is discontinuous unless  $\alpha = 2$ . Figure 3 shows a realization of a stationary Cauchy process, which is the proposed  $\alpha$ -stable random field with  $\alpha = 1$  and  $\beta = 0$ . We can see the process fluctuates wildly with sudden jumps. Based on this, we expect that the proposed  $\alpha$ -stable process models have the capability of fitting functions with discontinuities.

**Definition 3** (Stochastic Continuity). *Given  $t$ , a random field  $X(\mathbf{t})$  is said to be continuous in probability at  $\mathbf{t}$  if for all  $\epsilon > 0$ ,*

$$\lim_{\mathbf{s} \rightarrow \mathbf{t}} P(|X(\mathbf{s}) - X(\mathbf{t})| \geq \epsilon) = 0.$$

If a stochastic process  $Y(\mathbf{x})$  is continuous in probability, the conditional distribution  $p(Y(\mathbf{x})|Y(\mathbf{x}_1), \dots, Y(\mathbf{x}_n))$  is continuous with respect to  $\mathbf{x}$  under the weak topology. This implies that the predictors of  $Y(\cdot)$ , such as the predictive median or mean, are continuous and interpolate the observed data. It is easily verified that the proposed  $\alpha$ -stable processes are continuous in probability.

### 3. Statistical Inference

In this section we develop a Bayesian approach for estimation and inference. The aim is to obtain the predictive distribution for an untried point as well as the posterior distributions for the parameters. Since there is no explicit form of

these distributions, Monte Carlo methods are used.

In Section 3.1 and 3.2, we suppose that  $\alpha, \beta, \mu,$  and  $c$  are known. Therefore, by (2.3) and (2.7), the value of  $H(\mathbf{x}_i)$  and the distribution of  $v_j$  are known for  $i = 1, \dots, n$  and  $j = 1, \dots, s$ . This assumption is used for convenience because we can update these parameters in each iteration of the Markov chain Monte Carlo (MCMC) scheme using the Gibbs sampler. Without loss of generality, we take  $\mu = 0$  and  $c = 1$  because if  $x \sim S_\alpha(c, \beta, \mu)$  then  $(x - \mu)/c \sim S_\alpha(1, \beta, 0)$ . The vector  $\mathbf{d} = (d_1, \dots, d_m)$  is fixed in the whole MCMC chain. We produce multiple chains with different  $\mathbf{d}$  to obtain its posterior mode. The parameter estimation problems are discussed in Section 3.3.

### 3.1. Sampling latent variables

From (2.6), although the  $H(\mathbf{x}_i)$ 's are correlated, they can be decomposed into independent random variables  $\{v_i\}_{i=1}^s$ . If we can sample these  $s$  latent variables, the statistical inference can be largely simplified. We rewrite (2.6) into the matrix form

$$H(\mathbf{X}) = \mathcal{I}V, \tag{3.1}$$

where  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ ,  $H(\mathbf{X}) = (H(\mathbf{x}_1), \dots, H(\mathbf{x}_n))^T$ ,  $V = (v_1, \dots, v_s)^T$  and  $\mathcal{I}$  is an  $n \times s$   $(0, 1)$  matrix.

The proof of the following lemma is given in Appendix A.

**Lemma 1.** *There exists at least one subregion, which belongs to only one rectangle.*

Let  $\mathbb{A} = \{A_1, \dots, A_n\}$ . Lemma 1 suggests that there exists  $i_1$  and  $j_1$ , such that  $I_{j_1}$  does not belong to any rectangle in  $\mathbb{A} - \{A_{i_1}\}$ . Therefore, the  $(i^*, j_1)$  entry of  $\mathcal{I}$  is zero for any  $i^* \neq i_1$ . Applying Lemma 1 to  $\mathbb{A} - \{A_{i_1}\}$ , we get  $i_2 \neq i_1$  and  $j_2 \neq j_1$  such that  $I_{j_2}$  does not belong to any rectangle in  $\mathbb{A} - \{A_{i_1}, A_{i_2}\}$ . Therefore, the  $(i^*, j_2)$  entry of  $\mathcal{I}$  is zero for any  $i^* \neq i_1, i_2$ . By repeating this scheme, we get  $\{i_k\}_{k=1}^n$  and  $\{j_k\}_{k=1}^n$ . The submatrix of  $\mathcal{I}$  formed by rows  $i_1, \dots, i_n$  and columns  $j_1, \dots, j_n$  has the form

$$\mathcal{I}_{\begin{matrix} (i_1 \dots i_n) \\ (j_1 \dots j_n) \end{matrix}} = \begin{pmatrix} 1 & & \mathbf{U} \\ & \ddots & \\ \mathbf{0} & & 1 \end{pmatrix}.$$

**Theorem 1.** *The matrix  $\mathcal{I}$  has an  $n \times n$  submatrix  $\mathcal{I}_1$  with  $|\det(\mathcal{I}_1)| = 1$ .*

Without loss of generality, we can assume  $\mathcal{I} = (\mathcal{I}_1, \mathcal{I}_2)$ . Theorem 1 implies that  $\mathcal{I}$  is of full row rank, which means that the algorithm discussed later does

not suffer from a singularity problem.

Originally the  $v_i$ 's are independent. Thus  $V$  has density  $Q(V) := \prod_{i=1}^s q_i(v_i)$ , where  $q_i(\cdot)$  is the density function of  $S_\alpha((m(I_i))^{1/\alpha}, \beta, 0)$ . Since  $H(\mathbf{X})$  is known, the goal is to sample  $V|H(\mathbf{X})$ . The density of  $V|H(\mathbf{X})$  has the form

$$p(V|H(\mathbf{X})) \propto Q(V) \quad \text{subject to: } H(\mathbf{X}) = \mathcal{I}V. \quad (3.2)$$

By Theorem 1, the support of  $V|H(\mathbf{X})$  is a linear subspace of  $\mathbf{R}^s$  with dimensionality  $s - n$ . It is difficult to implement the Gibbs sampler due to the constraint. An easy way to draw  $V$  from its conditional distribution uses a Metropolis algorithm

---

**Algorithm 1** Random-walk Metropolis

---

1. Start with  $V_0$  satisfying  $H(\mathbf{X}) = \mathcal{I}V_0$ .
  2. Draw  $\delta V \sim N(0, \sigma^2 I)$  subject to:  $\mathcal{I}\delta V = 0$ .
  3. Let  $V_{new} = V_t + \delta V$ .
  4. Let  $V_{t+1} = V_{new}$  with probability  $\min\{1, Q(V_{new})/Q(V_t)\}$ ; otherwise let  $V_{t+1} = V_t$ .
  5. Set  $t \leftarrow t + 1$  and go to Step 2.
- 

The parameter  $\sigma^2$  in Step 2 is a tuning parameter. The projection of a standard multivariate normal distribution to a linear subspace is still standard multivariate normal. Thus to draw from the normal distribution subject to the linear constraint in Step 2, we only have to compute an orthogonal basis for the null space of  $\mathcal{I}$ . One approach is to use the singular value decomposition of the sparse matrix  $\mathcal{I}$  and, by Theorem 1, we have to draw  $s - n$  independent normal random variables since  $v$  still has degrees of freedom  $s - n$ .

### 3.2. Prediction

Based on the sampling of  $V|H(\mathbf{X})$ , we can derive a Monte Carlo method to compute the predictive distributions. Suppose we want to predict for a new point  $\mathbf{x}_0$ . Denote its associated hyper-rectangle by  $A_0$ . By the definition of  $\alpha$ -stable random measure and (2.7), we have the decomposition  $v_i = \bar{v}_i + \bar{v}'_i$ , where  $\bar{v}_i \sim S_\alpha((m(I_i \cap A_0))^{1/\alpha}, \beta, 0)$  and  $\bar{v}'_i \sim S_\alpha((m(I_i - A_0))^{1/\alpha}, \beta, 0)$ , for  $i = 1, \dots, s$ . Since  $A_0 = \sum_{i=0}^s A_0 \cap I_i$ , invoking (2.6) we have

$$H(\mathbf{x}_0) = \bar{v}_0 + \sum_{i=1}^s \bar{v}_i, \quad (3.3)$$

where  $\bar{v}_0 \sim S_\alpha((m(I_0 \cap A_0))^{1/\alpha}, \beta, 0)$ . The objective is to obtain the predictive distribution  $p(H(\mathbf{x}_0)|H(\mathbf{X}))$ . Note that  $p(H(\mathbf{x}_0)|H(\mathbf{X})) = E[p(H(\mathbf{x}_0)|V)|H(\mathbf{X})]$ . In Algorithm 1 we have an MCMC scheme to sample  $V|H(\mathbf{X})$ . Denote the output sequence of the MCMC algorithm by  $V_1, \dots, V_N$ , where  $N$  is the Monte Carlo sample size. Then the predictive distribution can be approximated by

$$\hat{p}(H(\mathbf{x}_0)|H(\mathbf{X})) = \frac{1}{N} \sum_{i=1}^N p(H(\mathbf{x}_0)|V = V_i).$$

The remaining problem is to draw  $H(\mathbf{x}_0)|V$ . Invoking (3.3), we need to sample

$$\bar{v}_0 + \sum_{i=1}^s \bar{v}_i \Big| V.$$

Here the  $\bar{v}_i$ 's are mutually independent conditional on  $V$ . Thus we can sample  $\bar{v}'_i$ 's independently from the conditional distribution  $p(\bar{v}_i|v_i)$ . Let  $p_1(\cdot)$  be the density of  $S_\alpha((m(I_i \cap A_0))^{1/\alpha}, \beta, 0)$  and  $p_2(\cdot)$  be that of  $S_\alpha((m(I_i - A_0))^{1/\alpha}, \beta, 0)$ . Since  $v_i = \bar{v}_i + \bar{v}'_i$  and  $\bar{v}'_i$  is independent of  $\bar{v}_i$ , we have the conditional distribution

$$v_i|\bar{v}_i \sim S_\alpha((m(I_i - A_0))^{1/\alpha}, \beta, \bar{v}_i).$$

Therefore, by Bayes' Theorem the target conditional distribution has the form

$$p(\bar{v}_i|v_i) \propto p_1(\bar{v}_i)p_2(v_i - \bar{v}_i). \tag{3.4}$$

As an example, we study the conditional distribution when the  $v_i$ 's follow Cauchy distributions,  $p(v_i) \propto 1/(v_i^2 + c^2(m(I_i)^2))$ . By (3.4) we have

$$p(\bar{v}_i|v_i) \propto (\bar{v}_i^2 + c^2(m(I_i \cap A_0))^2)^{-1} \left( (v_i - \bar{v}_i)^2 + c^2(m(I_i - A_0))^2 \right)^{-1}. \tag{3.5}$$

This conditional distribution has finite first and second moments and therefore is no longer stable. When  $v_i = 0$  and  $m(I_i \cap A_0) = m(I_i - A_0)$ , (3.5) implies that the conditional distribution of  $cm(I_i \cap A_0)\bar{v}_i$  is the  $t$ -distribution with degrees of freedom three. Except for this special case, the distribution given by (3.5) is non-standard. This simple example suggests that the condition distributions for  $\alpha$ -stable processes can be complicated; conditional inference for these processes is much more difficult than the Gaussian processes.

### 3.3. Parameter estimation

In this section, we introduce algorithms to estimate the unknown parameters in our model:  $\alpha, \beta, c, \mu$  and  $\mathbf{d}$ . We propose a hybrid Bayesian approach for the parameter estimation. The main steps are listed in Algorithm 2. Note that the proposed approach is not fully Bayesian because we solely calculate the posterior

mode of  $\mathbf{d}$ . We do not look for the posterior density of  $\mathbf{d}$  because the MCMC sampling for  $\mathbf{d}$  is intractable. Note that the number of subregions relies on the value of  $\mathbf{d}$  (see Figure 2). Thus,  $\mathcal{I}$  may change if a different value of  $\mathbf{d}$  is considered. In this situation, Algorithm 1 breaks down. We have to fix  $\mathbf{d}$  in an MCMC chain.

---

**Algorithm 2** Parameter Estimation

---

1. Choose a finite set of possible values for  $\mathbf{d}$ . Choose a current  $\mathbf{d}$  from this set.
  2. Given  $\mathbf{d}$  from the Step 1, sample  $\alpha, \beta, c$  and  $\mu$  using Gibbs sampling. This step is done along with the MCMC sampling of the latent variables discussed in Section 3.1. Details are given in Section 3.3.1.
  3. Using the MCMC samples obtained in Step 2, calculate the likelihood value for the current  $\mathbf{d}$ . Details are given in Section 3.3.2.
  4. Move to the next  $\mathbf{d}$  in the set. Repeat Step 2 and Step 3 until the likelihood value for each  $\mathbf{d}$  in the set is evaluated. Find the posterior mode of  $\mathbf{d}$ , i.e., the  $\mathbf{d}$  value which yields the greatest likelihood value.
- 

### 3.3.1. Gibbs sampling

In an MCMC chain, we fix  $\mathbf{d}$  to obtain a unified framework for data augmentation. In each iteration of this MCMC scheme, we update  $\alpha, \beta, \mu$  and  $c$  using the priors

$$\begin{aligned}
 p(\alpha) &\propto \alpha, \\
 p(\beta) &\propto (1 - |\beta|)_+, \\
 p(\mu) &\propto 1, \\
 c &\sim IG(\eta, \gamma),
 \end{aligned} \tag{3.6}$$

where  $IG$  is the inverse gamma distribution. We believe that in many problems, the true distributions cannot be too heavy-tailed or too skewed. Thus we choose non-uniform priors for  $\alpha$  and  $\beta$  in (3.6) to encourage the stable distributions not to be too far from the normal distributions.

Let  $Y = (y(x_1), \dots, y(x_n))^T$ . Using (2.3) and (3.1), we obtain the linear system

$$Y - \mu \mathbf{1} = \mathcal{I} \tilde{V}, \tag{3.7}$$

where  $\tilde{V} = cV$  and  $\mathbf{1}$  is a vector with all elements 1. If  $\tilde{v}_i$  is the  $i$ th component of  $\tilde{V}$ , then  $\tilde{v}_i \sim S_\alpha(c(m(I_i))^{1/\alpha}, \beta, 0)$ . The density of  $\tilde{V}$  is  $\tilde{Q}(\tilde{V}) := \prod_{i=1}^s \tilde{q}_i(\tilde{v}_i)$ , where  $\tilde{q}_i(\cdot)$  is the density function of  $S_\alpha(c(m(I_i))^{1/\alpha}, \beta, 0)$ . Therefore, we do not

have to update  $\tilde{V}$  during the Gibbs samplings of  $\alpha, \beta$ , and  $c$ , since (3.7) still holds after the Gibbs samplings. When updating  $\mu$ , we propose a transition of  $\tilde{V}$  to keep the identity (3.7).

Let the initial state be  $(\mu, \tilde{V})$ . Suppose we have a proposal move for the parameter  $\mu$  with a transition kernel  $q(\cdot, \cdot)$ . Now consider one iteration of the MCMC scheme. Suppose  $\mu$  moves to  $\mu'$  with  $\mu' \neq \mu$ . Then by (3.7), we should move  $\tilde{V}$  to some  $\tilde{V}'$  satisfying  $\mathbf{1} = \mathcal{I}(\tilde{V} - \tilde{V}')/(\mu' - \mu)$ . For the choice of  $\tilde{V}'$ , we suggest using the  $\mathbf{z}_0$  that solves the quadratic optimization problem

$$\min_{\mathbf{z}} \|\mathbf{z}\|^2 \quad \text{subject to: } \mathcal{I}\mathbf{z} = \mathbf{1}. \tag{3.8}$$

Since (3.8) is independent of  $\mu$ , we only have to solve this problem once for the whole MCMC chain. We propose the move  $\tilde{V}' = \tilde{V} + (\mu - \mu')\mathbf{z}_0$ , which keeps (3.7). If the proposal move for  $\mu$  is symmetric,  $q(x, x') = q(x', x)$  for any  $x \neq x'$ , then the proposed move for  $(\mu, \tilde{V})$  is also symmetric. Basically  $\mathbf{z}_0$  can be any  $\mathbf{z}$  satisfying  $\mathcal{I}\mathbf{z} = \mathbf{1}$ , but we use the least distance solution because it results in a small shift of  $\tilde{V}$ . We expect that, in this case, the variation of  $\tilde{Q}(\tilde{V})$  is also small so that the Metropolis scheme can achieve a relatively high acceptance rate.

For the Gibbs sampling of  $\alpha, \beta$ , and  $c$ , existing methods (Buckle (1995); Lombardi (2007)) can be extended and implemented in a straightforward manner.

### 3.3.2. Estimation of likelihood values

To compute the posterior mode for the parameter  $\mathbf{d}$ , we have to obtain the likelihood value  $P(Y|\mathbf{d})$ , where  $P(Y|\mathbf{d}) = \int P(Y|\alpha, \beta, c, \mu, \mathbf{d})p(d\alpha)p(d\beta)p(dc)p(d\mu)$ . We suggest a method to compute this value using the output from Algorithm 1. The proof of Theorem 2 is given in Appendix A.

**Theorem 2.** *Let  $X = (x_1, \dots, x_s)^T$  be a vector of continuous random variables with density  $p_1(x_1) \cdots p_s(x_s)$ , and the matrix  $A = (A_1, A_2)$ ,  $A_1 \in \mathbf{R}^{n \times n}$  be of full rank with  $A_2 \in \mathbf{R}^{n \times (s-n)}$ . Take  $Y = AX$  with density  $P(Y)$ . If  $X_1 = (x_1, \dots, x_n)$ ,  $X_2 = (x_{n+1}, \dots, x_s)$ ,  $P_1(X_1) = \prod_{i=1}^n p(x_i)$ ,  $P_2(X_2) = \prod_{i=n+1}^s p(x_i)$ , then*

$$\frac{1}{P(Y)} = |\det(A_1)|E\left[\frac{1}{P_1(X_1)} \middle| AX = Y\right].$$

From an MCMC chain we obtain a sequence of samples for the latent vector  $V$  that can be used to estimate the underlying likelihood value. Choosing  $\mathcal{I}_1$  satisfying the conclusion of Theorem 1, by Theorem 2 the likelihood value can be estimated by

$$\frac{1}{\hat{L}(Y|\mathbf{d})} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^n \frac{1}{q_{i,j}(v_{i,j})},$$

where  $q_{i,j}$  and  $v_{i,j}$  are  $q_j$  and  $v_j$  in the  $i$ th MCMC iteration, respectively.

#### 4. Computing Rectangle Partition

In this section we consider the solution to the geometric problem of hyper-rectangle partition raised in Section 2.3. It is very costly to obtain all the details of the geometric structure given by the rectangles, so we look for an efficient algorithm that provides only the necessary information. According to in Section 2.3 and Section 3, only two types of geometric information are required for statistical inference. The first type consists of the total number of the subregions and the volume of each subregion. The volumes of the subregions are used to determine the distributions of  $v_i$ 's defined by (2.6) and (2.7). The second type of information provides, for each rectangle, the subregions that belong to this rectangle. This linking information determines the matrix  $\mathcal{I}$  in equation (3.1). The algorithm proposed in this section provides both types of information without recording the shape of each subregion. To describe this algorithm, we use certain terminology from data structure. The algorithm is inspired by some ideas from *computational geometry*. A good introduction to this field is de Berg, Cheong and Van Kreveld (2008).

##### 4.1. An illustration

We use a simple example to illustrate how the proposed algorithm works. Consider two overlapping 2D-rectangles, shown in Figure 4 with solid lines. Denote the upper rectangle by Rectangle 1 and the lower one by Rectangle 2. At the beginning, we compute the projection of these rectangles to 1D space, as shown by the dashed line on the left. Then we obtain a 1D version of the rectangle partition problem that is trivial to solve. We can easily figure out that the two line segments (projections of the two rectangles) form three disjoint segments, denoted by Segment I, II, and III. The length of each segment can be obtained, too. In addition, we record the following linking information: Segment I belongs to Rectangle 1 only; Segment II belongs to both rectangles; and Segment III belongs to Rectangle 2 only. Now we return to the original 2D problem. The structure on the second dimension can be represented by the four sides of the two rectangles on the second dimension. We denote these four sides by Side i, ii, iii, iv from left to right. Now we initialize an empty *queue* for each of the disjoint

segments in the 1D problem, denoted by Queue I, II and, III. Then we scan the second dimension from  $-\infty$  to  $+\infty$ . The first object appears is Side i, which is the start line of Rectangle 1. From the 1D problem, we already know that Segment I and II belong to Rectangle 1. Thus we insert 1 to Queue I and II, and leave Queue III unchanged. Then we see Side ii. We obtain three small rectangle regions between Side i and ii, divided by the dotted lines in Figure 4. Each small rectangle is associated with a segment in the 1D problem. Now read the three queues. Queue I and II contain 1 only, and Queue III is empty (denoted by  $\phi$  in Figure 4), which means that the first and second region belong to Rectangle 1 only and the third region does not belong to any rectangle. Thus we see the first 2D subregion, indexed by 1. Currently its area is the sum of the volume of the first two small rectangles. The areas of these rectangles are the distance between Side i and ii times the lengths of Segment I and II, respectively. Then we update the queues again. Since Rectangle 2 starts from Side ii, we insert 2 to Queue II and III, and leave Queue I unchanged. Next we move to Side iii. We obtain three new small rectangle regions between Side ii and iii, so Queue I contains 1 only, Queue II contains 1 and 2, and Queue III contains 2 only. Thus the first rectangle still belongs to subregion indexed by 1, and the other two belongs to two new subregions, indexed by 12 and 2, respectively. Here the term “12” denotes the subregion that belongs to Rectangles 1 and 2. Update the area of the subregion indexed by 1 by adding the area of the first small rectangle. Also initialize the subregions indexed by 12 and 2 and calculate their areas. Then update the queues. Since Rectangle 1 ends at Side iii, the term 1 should leave all queues. Because Queue III does not contain 1, it remains unchanged. We can see that the queue data structure (i.e., first-in-first-out) works here because all rectangles are congruent. Finally we turn to Side iv. The status of the queues are: empty, 2, 2. Thus we update the area of the subregion indexed by 2 by adding the areas of the last two rectangles between Side iii and iv. Since we have met all the four sides, the algorithm terminates.

## 4.2. Algorithm

A fundamental concept in the proposed methodology is subregions. As shown in Sections 2.3, a subregion is generated by the intersection of a class of rectangles. The set of rectangle labels is called the *index* of the subregion in Section 4.1. If two subregions hold the same index, they should be merged as one. In our statistical inference procedure, we also need the volume of the subregion. Here we define Subregion as a data type: for Subregion  $r$ ,  $r\$index$  and  $r\$volume$  denote

I		1	1	$\phi$	
II		1	12	2	
III		$\phi$	2	2	
		i	ii	iii	iv

Figure 4. Computational geometry algorithm.

its index and volume. Given an index, there is an associated volume. Such a relationship can be described using an *associative table*, also called a *map*. An associative table defines an association between its *Key* and its *Value*. It can be realized using data structures like *binary search trees* or *hash tables*. Here we call the data structure consisting of Subregions *Table*. For Table  $t$ , we use the notation  $t[id]$  to denote the volume of subregion in  $t$  with the index  $id$ . Note that  $t[id]$  may not exist if  $t$  does not contain a subregion with such an index.

A basic operation is to add a new part of a subregion to a table. This sub-function is given in Algorithm 3.

---

**Algorithm 3** Sub-function Merge.
 

---

 Input (Table  $t$ , Subregion  $r$ )

 If  $t[r\$index]$  exists

 $t[r\$index] \leftarrow t[r\$index] + r\$volume$ 

Else

 Add  $(r\$index, r\$volume)$  to  $t$ .

End If

 Output  $t$ 


---

The algorithm, given a set of intervals, determine the generated subregions and their lengths. This step is trivial and can be done by a standard implementation of the Queue data structure. For a general  $m$ -dimensional problem, we first generate  $m$  associative tables given by the projection to each of the coordinate axes. Then the algorithm proceeds in a recursive manner, by adding one dimension to an existing  $d$ -dimensional table. The ascending dimension procedure is described in Algorithm 4, where  $t1$  denotes the  $d$ -dimensional table and  $t2$  denotes the one-dimensional table to be added. Finally, the main algorithm (shown in Algorithm 5) proceeds by successively adding all dimensions.

**Algorithm 4** Sub-function Ascent

---

Input (Table  $t1$ , Table  $t2$ )  
 Initiate Subregion  $r3$ , Table  $t3$   
 For each  $r1$  in  $t1$ ,  $r2$  in  $t2$   
    $r3\$index \leftarrow r1\$index \cap r2\$index$   
    $r3\$volume \leftarrow r1\$volume * r2\$volume$   
   Call  $Merge(t3, r3)$   
 End For  
 Output  $t3$

---

**Algorithm 5** Main Algorithm for Calculation the Subregions and Their Volumes

---

For each dimension  $i$ , generate table  $S[i]$   
 For  $i$  in  $2 : m$   
    $S[1] = Ascent(S[1], S[i])$   
 End For  
 Output  $S[1]$

---

**4.3. Prediction at untried points**

We move to the prediction problem. Suppose we have  $n$  input points denoted by  $\{x_1, \dots, x_n\}$  and  $n^*$  untried points of interest denoted by  $\{x_1^*, \dots, x_{n^*}^*\}$ . Denote associated rectangles for the input and untried points by  $\{A_1, \dots, A_n\}$  and  $\{A_1^*, \dots, A_{n^*}^*\}$ , respectively. If we implement Algorithm 4 to these  $n + n^*$  rectangles, we can get  $m(A_i \cap A_j^*)$  and  $m(A_j^* - \cap_{k=1}^n A_k)$  for  $1 \leq i \leq n$  and  $1 \leq j \leq n^*$ . These volumes are necessary. The algorithm also gives the “interaction” terms between the untried points,  $m(A_i^* \cap A_j^*)$ . When  $n^*$  is large, there are a large number of such terms. However, these terms are not needed in the prediction, if we are not interested in the joint distribution of the untried points. Therefore we suggest a modification of the algorithm that updates the volumes separately for the untried points. The computation for the rectangle given by the input points follows the same scheme.

We have shown how to compute the properties of the subregions generated by  $\{A_1, \dots, A_n\}$ . Denote these (bounded) subregions by  $I_1, \dots, I_s$ . Clearly, it suffices to obtain  $m(A_j^* \cap I_i)$  for each  $i, j$ , because  $m(A_j^* - \cap_{k=1}^n A_k) = m(A_j^*) - \sum_{i=1}^s m(A_j^* \cap I_i)$ .

The volume  $m(A_j^* \cap I_j)$  can be calculated similarly as were those in Section 4.2. The remainder of the algorithm proceeds similarly as Algorithm 3-5 by replacing  $t[i]$  with  $t_j[i]$  for each  $j$ .

Table 1. Numerical results for the discontinuous function given by the proposed method and the Gaussian process model. SD = standard deviation.

$i$	Stable process model		Gaussian process model	
	mean	SD	mean	SD
1	-1.000	0.017	-0.989	0.168
2	-1.000	0.017	-0.952	0.227
3	-1.000	0.016	-0.937	0.168
4	-1.000	0.016	-0.983	0.031
5	-1.000	0.016	-1.088	0.114
6	-1.000	0.017	-1.202	0.194
7	-1.000	0.017	-1.237	0.171
8	-1.000	0.016	-1.108	0.058
9	-0.018	1.001	-0.778	0.085
10	-0.015	1.001	-0.281	0.181
11	-0.009	1.001	0.281	0.181
12	-0.005	1.002	0.778	0.085
13	1.000	0.016	1.108	0.058
14	1.000	0.017	1.237	0.171
15	1.000	0.017	1.202	0.194
16	0.999	0.016	1.088	0.114
17	1.000	0.015	0.983	0.031
18	1.000	0.017	0.937	0.168
19	1.000	0.017	0.952	0.227
20	1.000	0.016	0.989	0.168

## 5. Numerical Illustrations

In this section we use two examples to illustrate the proposed method and demonstrate its potential advantages.

### 5.1. Handling discontinuity

The first example considers the interpolation problem for a discontinuous function.

$$F(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{if } x < 0. \end{cases}$$

The design consists of six points:  $(-0.25, -0.15, -0.05, 0.05, 0.15, 0.25)$ . We want to predict for 20 untried points, given by  $i/42 - 0.25$  for  $1 \leq i \leq 20$ . Thus the true values for the first 10 points are  $-1$ , and those for the rest are  $1$ . We compare the proposed method and the Gaussian process model. For the proposed method, we chose  $d = 0.45$ . We ran MCMC for 15,000 iterations with 5,000 burn-in runs. For the Gaussian process model, we assumed the correlation

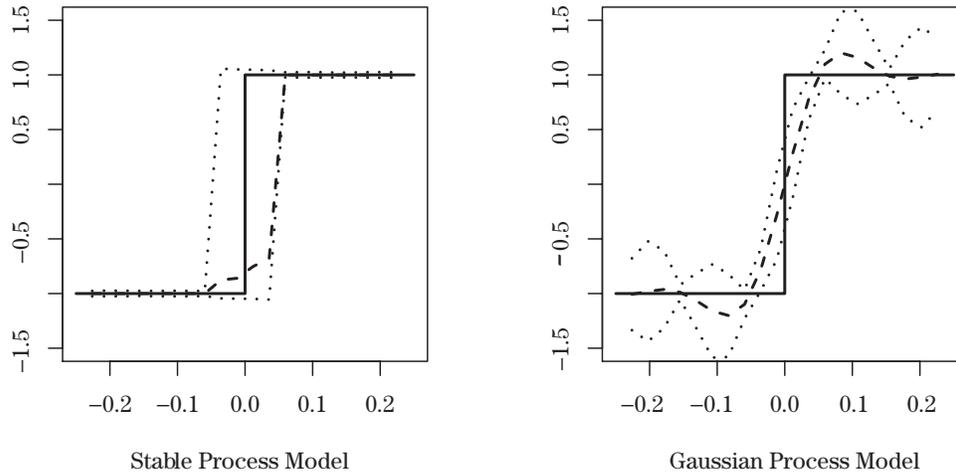


Figure 5. Credible limits given by the stable and Gaussian models. The solid lines denote the true function. The dashed lines denote the predictive median given by the two methods respectively. The dotted lines denote the 0.05 and 0.95 credible limits given by the two methods.

function  $r(x) = \exp\{-\theta x^2\}$ , where  $\theta$  is the unknown correlation parameter. We implemented a frequentist approach for the inference of the Gaussian process model. We compared the mean and standard deviation for each  $i$  given by the two methods. The results are given in Table 1.

For the Gaussian process model, the predictive means are not accurate for  $i = 1, \dots, 8$  and  $i = 13, \dots, 20$ , since the predictive curve of a Gaussian process model oscillates near the discontinuity (the Gibbs phenomenon (Gottlieb and Shu (1997))). The confidence interval given by the Gaussian process model also fails to capture the true uncertainty of the prediction. It underestimates the standard deviations for the points for  $i = 9, 10, 11, 12$ , while the confidence intervals for  $i = 1, \dots, 8$  and  $i = 13, \dots, 20$  are too wide.

We plot the 0.05, 0.5, 0.95 credible limits given by the stable and Gaussian models in Figure 5, in which the stable process model gives much better results than the Gaussian process model. While the Gaussian process model has inflation of uncertainty for  $i = 1, \dots, 8$  and  $i = 13, \dots, 20$  and deflation for  $i = 9, 10, 11, 12$ , the stable process model gives reasonable credible intervals. Here the standard deviations are large for  $i = 9, 10, 11, 12$  and the predictors are nearly exact with very small credible intervals for  $i = 1, \dots, 8$  and  $i = 13, \dots, 20$ . Figure 6 plots the predictive densities for  $i = 9, 10, 11, 12$ . We can see that each of the predictive densities has two peaks around  $-1$  and  $1$  respectively. A Gaussian process model

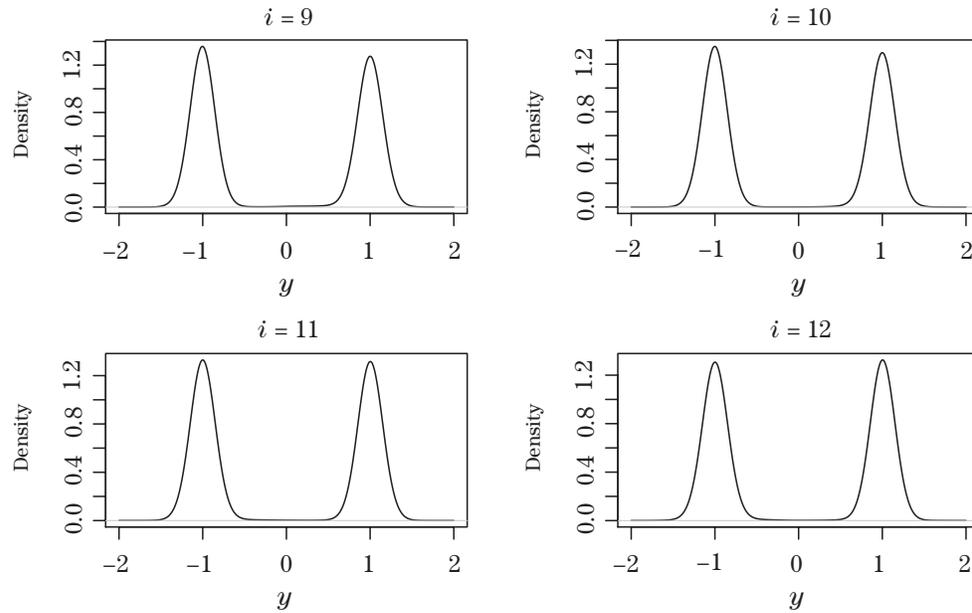


Figure 6. Predictive densities for  $i = 9, 10, 11, 12$ .

cannot give similar results because the conditional distribution is Gaussian and unimodal. A frequentist method using stable processes, as in Karcher, Shmileva and Spodarev (2013), cannot give this result because its predictor is a single point.

## 5.2. Determination of gaussianity

To determine if the true process is Gaussian, for estimation efficiency, we only need to study the parameter estimation problem for  $\alpha$ .

We chose a maximin distance Latin hypercube of 30 points in  $[0, 1]^2$  as the design points. The observations were simulated from a stationary Gaussian process with mean zero and correlation function  $r(\mathbf{x}) = \exp\{-10\|\mathbf{x}\|^2\}$ . We used the proposed model with  $\mathbf{d} = (0.4, 0.4)$  to fit the data. We generated four independent realizations of this Gaussian process on the design points. For each realization we ran MCMC for 1,000 iterations under the proposed model. The initial values of  $\alpha$  in the four chains were 1.3. In Figure 7, we plot the MCMC track of  $\alpha$  for each realization of the Gaussian process.

From Figure 7, in each realization  $\alpha$  appears stationary after 200 iterations with its mean values eventually above 1.8. Based on these results, we suggest an empirical rule: declare the true process Gaussian if the proposed method gives

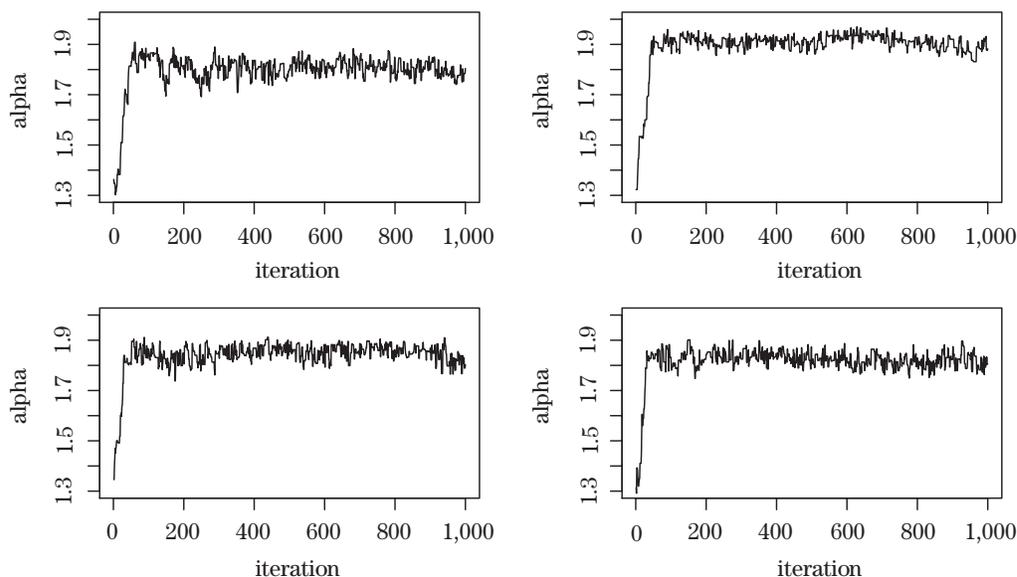


Figure 7. MCMC tracks of  $\alpha$  for four independent realizations of the Gaussian process.

$\alpha > 1.8$ . This Gaussian process does not belong to the proposed stable process family since its correlation function does not have the form (2.6) and yet the proposed method can still lead to relatively large  $\hat{\alpha}$  values.

## 6. Discussion

Most of the computational work described in Section 3.2 can be done using standard tools except for the sampling of (3.4). There should be an efficient algorithm for generating independent samples in (3.4). The development of such an algorithm is of value because it can improve the accuracy in prediction. Algorithm 1 has room for improvement because it is not efficient for large  $s$ . We suggest employing the hybrid Monte Carlo algorithm using Hamiltonian dynamics (Neal (2010)). An efficient version of this algorithm for the proposed models needs further development.

Further research on the complexity of the computational geometry algorithm is also warranted. Similar to the orthogonal range searching problem (de Berg, Cheong and Van Kreveld (2008)), the complexity of the proposed algorithm relies on the total number of the subregions,  $s$  in (2.7). Numerical experience indicates that  $s$  decreases as  $d_i$  decreases. Thus further investigation on an accurate upper bound of  $s$  is of interest.

## Acknowledgment

The author would like to thank his thesis advisor Professor C. F. Jeff Wu for discussions that inspired the current research and for his comments and suggestions. Tuo's work is supported by NSFC 11501551, 11271355, 11671386, and the National Center for Mathematics and Interdisciplinary Sciences, CAS. This research is part of the doctoral thesis completed while visiting Georgia Institute of Technology under partial support of NSF grant DMS 1007574. The author is grateful to an associate editor and referees for helpful comments.

## Appendix A: Proofs

*Proof of Lemma 1.* Sort  $\{\mathbf{x}_i : i \in F\}$  using the lexicographical order. Let the smallest element be  $A_{i_0}$ . Then  $A_{i_0}$  has vertex  $x_{i_0} - \mathbf{d}$ . This vertex cannot be covered by any rectangle in  $\{A_i : i \in F - \{i_0\}\}$ . Suppose this vertex belongs to subregion  $I_{j_0}$ . Since  $I_{j_0}$  is an atom,  $I_{j_0}$  does not belong to any rectangle in  $\{A_i : i \in F - \{i_0\}\}$ . This completes the proof.

*Proof of Theorem 2.* Let  $c_1, c_2 \in \mathbf{R}^n$ . We use the notation  $a_1 < a_2$  for any  $a_1, a_2 \in \mathbf{R}^n$  if each component of  $a_1$  is less than the same component of  $a_2$ . By the  $\pi - \lambda$  Theorem (Durrett (2010)), it suffices to prove that

$$\int_{c_1 < Y < c_2} \frac{1}{P(Y)} dP = |\det(A_1)| \int_{c_1 < AX < c_2} E\left[\frac{1}{P_1(X_1)} \middle| AX\right] dP.$$

Define rectangle  $C = \{x : c_1 < x < c_2\}$ . Here the left side =  $Vol(C)$ , where  $Vol(C)$  is the Volume of  $C$ . For the integral on the right, we have

$$\begin{aligned} & \int_{c_1 < AX < c_2} E\left[\frac{1}{P_1(X_1)} \middle| AX\right] dP \\ &= E\left[\frac{1}{P_1(X_1)} I(c_1 < AX < c_2)\right] \\ &= \int_{c_1 < AX < c_2} \frac{1}{P_1(X_1)} P_1(X_1) P_2(X_2) dX_1 dX_2 \\ &= \int_{c_1 < A_1 X_1 + A_2 X_2 < c_2} P_2(X_2) dX_1 dX_2 \\ &= \int_{X_2} \left( \int_{c_1 - A_2 X_2 < A_1 X_1 < c_2 - A_2 X_2} P_2(X_2) dX_1 \right) dX_2 \\ &= Vol(\mathbf{A}_1^{-1} C) \end{aligned}$$

$$= |\det(A_1)^{-1}| \text{Vol}(C),$$

where  $\text{Vol}(\mathbf{A}_1^{-1}C)$  is the volume of  $C$  under the linear transformation defined by  $\mathbf{A}_1^{-1}$ . This completes the proof.

## References

- Applebaum, D. (2009). *Lévy Processes and Stochastic Calculus*. Cambridge University Press.
- Banerjee, S., Carlin, B. P. and Gelfand, A. E. (2004). *Hierarchical Modeling and Analysis for Spatial Data*, volume 101. Chapman & Hall.
- Buckle, D. J. (1995). Bayesian inference for stable distributions. *Journal of the American Statistical Association*, 605–613.
- Chambers, J. M., Mallows, C. L. and Stuck, B. W. (1976). A method for simulating stable random variables. *Journal of the American Statistical Association*, 340–344.
- Chen, R. B., Wang, W. and Wu, C. F. J. (2010). Building surrogates with overcomplete bases in computer experiments with applications to bistable laser diodes. *IIE Transactions* **43**, 39–53.
- de Berg, M., Cheong, O. and Van Kreveld, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag New York Inc.
- Durrett, R. (2010). *Probability: Theory and Examples*. Cambridge University Press.
- Gottlieb, D. and Shu, C. W. (1997). On the gibbs phenomenon and its resolution. *SIAM Review*, 644–668.
- Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association* **103**, 1119–1130.
- Karcher, W., Shmileva, E. and Spodarev, E. (2013). Extrapolation of stable random fields. *Journal of Multivariate Analysis* **115**, 516–536.
- Lombardi, M. J. (2007). Bayesian inference for  $\alpha$ -stable distributions: A random walk mcmc approach. *Computational Statistics & Data Analysis* **51**, 2688–2700.
- Neal, R. M. (2010). Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo* **54**, 113–162.
- Nolan, J. P. (1997). Numerical calculation of stable densities and distribution functions. *Communications in Statistics. Stochastic Models* **13**, 759–774.
- Nolan, J. P. (2001). Maximum likelihood estimation and diagnostics for stable distributions. *Lévy Processes: Theory and Applications*, 379–400.
- Padoan, S. A., Ribatet, M. and Sisson, S. A. (2010). Likelihood-based inference for max-stable processes. *Journal of the American Statistical Association* **105**, 263–277.
- Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.
- Samorodnitsky, G., Taqqu, M. S. and Linde, R. W. (1996). Stable non-gaussian random processes: stochastic models with infinite variance. *Bulletin of the London Mathematical Society* **28**, 554–555.
- Santner, T. J., Williams, B. J. and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer Verlag.

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Verlag.

Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China  
100190.

E-mail: tuorui@amss.ac.cn

(Received October 2015; accepted February 2017)