COMBINING STRATEGIES FOR PARALLEL STOCHASTIC APPROXIMATION MONTE CARLO ALGORITHM OF BIG DATA

A Dissertation

by

FANG-YU LIN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Faming Liang
Co-Chair of Committee,	Raymond Carroll
Committee Members,	Michael Longnecker
	Ramalingam Saravanan
Head of Department,	Valen Johnson

December 2014

Major Subject: Statistics

ABSTRACT

Modeling and mining with massive volumes of data have become popular in recent decades. However, it is difficult to analyze on a single commodity computer because the size of data is too large. Parallel computing is widely used. As a natural methodology, the divide-and-combine (D&C) method has been applied in parallel computing. The general method of D&C is to use MCMC algorithm in each divided data set. However, MCMC algorith is computationally expensive because it requires a large number of iterations and is prone to get trapped into local optima. On the other hand, Stochastic Approximation in Monte Carlo algorithm (SAMC), a very sophisticated algorithm in theory and applications, can avoid getting trapped into local optima and produce more accurate estimation than the conventional MCMC algorithm does. Motivated by the success of SAMC, we propose parallel SAMC algorithm that can be utilized on massive data and is workable in parallel computing. It can also be applied for model selection and optimization problem. The main challenge of the parallel SAMC algorithm is how to combine the results from each parallel subset. In this work, three strategies to overcome the combining difficulties are proposed. From the simulation results, these strategies result in significant time saving and accurate estimation.

Synthetic Aperture Radar Interferometry (InSAR) is a technique of analyzing deformation caused by geophysical processes. However, it is limited by signal losses which are from topographic residuals. In order to analyze the surface deformation, we have to distinguish signal losses. Many methods assume the noise has second order stationary structure without testing it. The objective of this study is to examine the second order stationary assumption for InSAR noise and develop a parametric

nonstationary model in order to demonstrate the effect of making incorrect assumption on random field. It indicates that wrong stationary assumption will result in bias estimation and large variation.

DEDICATION

To my family

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my advisor, Dr. Faming Liang, an outstanding statistician in the field of statistical computation. Dr. Liang gave me unwavering support and indoctrination on research. His research attitude, insights, understanding and constant encouragement have not only been helpful in my Ph.D study but also influenced my personal philosophy about life. I will cherish these precious moments forever.

I would also like to thank Dr. Michael Longnecker for his constant support since I entered this department. His patience and guidance in many aspects are extremely helpful for me to accomplish my Ph.D degree.

I also deeply appreciate the other members of my committee, Dr. Raymond Carroll, Dr. Michael Longnecker, and Dr. Ramalingam Saravanan for their continuous assistance and valuable advice related to my research. I am also grateful to Dr. Genton, Dr. Jun, faculty members, students and colleagues at the Department of Statistics for their friendship and support. Thank you all.

Without my family's sacrifice and support, this work would not have been possible. They are my strong supporters and always love me. Whenever I need them, they are always there. Finally and most importantly, I would like to thank my husband, Hung-Ming Chou for his understanding, unconditional love and strong support so that I can accomplish my Ph. D study.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

This dissertation consists of two topics, Combining Strategies for Parallel Stochastic Approximation Monte Carlo Algorithm (SAMC) of Big Data and Stationary Test for InSAR Noise. The work can be applied across the fields of Stochastic Approximation Monte Carlo (SAMC), data mining, stationarity test and nonstationary model. The following paragraphs briefly introduce the motivation of two topics.

Enterprises collect billion bytes of data from mobile devices, software logs and so forth every day and increasingly look for insights into their data. Modeling big data from the need to answer business question, such as how to increase sales intelligence and how customer react of their recently ads, become more and more popular in decades. However, scientists encounter limitations for analyzing big data on a single commodity computer for two main reasons; volume of data is too big to fit in a single computer and big data requires a long processing time. The sophisticated way to overcome those limitations is parallel computation.

The basic idea of parallel computation is to divide the target problem into several smaller tasks and to perform the algorithm in each subset simultaneously. To choose a proper algorithm doing inference in subset, we inspired by the successful of stochastic approximation Monte Carlo (SAMC) in optimization. The stochastic approximation Monte Carlo (SAMC) algorithm is introduced in this study. The stochastic approximation Monte Carlo (SAMC) algorithm has been proposed by Liang et al. (2007). It addressed the shortages of the conventional MCMC algorithm trapped into local minima issue and improved the convergence of Wang-Landau algorithm (Wang and Landau (2001)). SAMC algorithm is a self-adjusting mechanism; it enables to explore whole sample domain during the updating estimator and is

not trapped by local energy minima even if the data structures are complex. Thus it is a sophisticate algorithm for Monte Carlo optimization. The main challenges in this research is how to combine the inference results after parallel computation. There is no statistical methodology engage in this aspect so far. In this dissertation, we propose three combining strategies for SAMC parallel computation and examine the calculating time and accuracy of estimation to compare with SAMC sequential computation.

Synthetic Aperture Radar (SAR) Interferometry is an important tool to study earthquake and volcanic deformation. However, to analyzed the deformation of surface through InSAR image is limited because of signal loss. The signal loss comes from topographic residuals, atmospheric delay and several error sources are difficult to define. To estimate the noise of InSAR image become the main challenge. Most of the studies assume the noise structure is the same as in the deforming structure; i.e. assuming that the noise has a second order stationary structure. However, the stationary assumption has never been tested. The objective of this research topic is to examine the second order stationary assumption for InSAR noise and demonstrate the effect of making incorrect stationarity assumption.

This chapter provides the backgrounds for two research topics with the arrangement as follows. Section 1.1 reviews the Stochastic Approximation Monte Carlo Algorithm and theory of dynamic weighted importance sampling. Section 1.2 provides the theory of stationary test. Section 1.3 displays the structure of this dissertation.

1.1 Stochastic Approximation Monte Carlo

Suppose that we are interested in sampling from a distribution which can be written by a proposal density with an unknown normalizing constant in the following

form,

$$p(X) = c\psi(x) \qquad x \in \mathcal{X} \tag{1.1}$$

where $\mathcal{X}$ is the sample space, c is the normalizing constant and $\psi(x)$ is a non-negative function. In Bayesian inference, $\psi(x)$ corresponds to the unnormalized posterior density $\pi(\theta|X) = \pi(\theta)f(X|\theta)$.

Suppose the sample space can be partitioned into m disjoint subregions according to a function defined by $U(X)$, $E_1 = \{x : U(X) \leq u_1\}$, $E_2 = \{x : u_1 < U(X) \leq u_2\}, \cdots$, $E_{m-1} = \{x : u_{m-2} < U(X) \leq u_{m-1}\}$, $E_m = \{x : U(X) > u_{m-1}\}$, where $u_1, u_2, \cdots, u_{m-1}$ are real pre-specified numbers which can be defined by the user.

SAMC tends to draw samples from each of the subregions with a pre-specified frequency. Without loss of generality, we assume that there is no empty subregion in the partition, that is, $\int_{E_i} \psi(x)dx > 0$ for $i = 1, \cdots, m$. Let $\pi = (\pi_1, \pi_1, \cdots, \pi_m)$ represent the vector of desired sampling frequencies of m disjoint subregions, where $0 < \pi_i < 1$ and $\sum_{i=1}^{m} \pi_i = 1$. π is called the desired sampling distribution in Liang et al. (2007). Let $\theta_i = log(\int_{E_i} \psi(x)dx/\pi_i)$ for $i = 1, \cdots, m$. For convenience, let $\theta^{(t)} = (\theta_1^{(t)}, \cdots, \theta_m^{(t)})$ denotes the working estimate of θ obtained at iteration t. If $\theta^{(t)}$ can be well estimated, sampling from target distribution $p_{\theta^{(t)}}(X)$ will result in a random walk and each subregions will be sampled with a frequency proportional to π. Because of updating $\theta^{(t)}$ in the SAMC algorithm, if a proposal is rejected, the weight of the subregion that the current sample belongs to will be adjusted to a larger value, and thus the proposal of jumping out from the current subregion will be less likely rejected in the next iteration. It overcomes the difficulty of local trap from conventional MCMC algorithm.

Let $x^{(t)}$ denote a sample drawn from MH algorithm with proposal distribution $q(x^{(t)}, y)$,where $x^{(t)}$ denotes the current state at iteration t and y denotes the pro-

posed state. The proposal distribution $q(x^{(t)}, y)$ is not necessarily symmetric and satisfy the following condition. For every $x \in \mathcal{X}$, there exist $\epsilon_1 > 0$ and $\epsilon_2 > 0$ such that $|x-y| < \epsilon_1 \Longrightarrow q(x^{(t)}, y) > \epsilon_2$. This is a common condition for proposal density in MCMC theory (Mengersen and Tweedie (1996)). The commonly used proposal density are normal, Cauchy and Student t-distribution in continuous system. For discrete system, we can defined the random walk on neighborhood of sample x in a specific way. Henceforth, The target density $p_{\theta^{(t)}}(X)$ for Stochastic Approximation Monte Carlo algorithm (SAMC) can be briefly explained in the following form,

$$p_{\theta^{(t)}}(X) \propto \sum_{i=1}^{m} \frac{\psi(x)}{e^{\theta_i^{(t)}}} I(x \in E_i) \qquad\qquad i = 1, \cdots, m \tag{1.2}$$

where $I(.)$ is an indicator function. Because $p_{\theta^{(t)}}(X)$ is invariant with respect to $\theta^{(t)}$, then adding to or subtracting a constant vector from $\theta^{(t)}$ will not change $p_{\theta^{(t)}}(X)$. To avoid vary truncation of $\theta^{(t)}$, we can keep $\theta^{(t)}$ in a compact set $\Theta = [-10^{100}, 10^{100}]^m$.

Define $H(\theta^{(t)}, x^{(t+1)}) = e^{(t+1)} - \pi$, where $e^{(t+1)} = (e_1^{(t+1)}, \cdots . e_m^{(t+1)})$ and $e_i^{(t+1)} = 1$ if $x^{(t+1)} \in E_i$ and 0 otherwise. Let γ_t denotes the gain factor (Robbins and Monro (1951)) which is a positive, non-decreasing sequence satisfying,

$$\sum_{t=1}^{\infty} \gamma_t = \infty \qquad\qquad \sum_{t=1}^{\infty} \gamma_t^{\zeta} < \infty$$

for some $\zeta \in (1, 2)$. In this study, we set

$$\gamma_t = \frac{t_0}{\max(t_0, t)}, \qquad\qquad t = 0, 1, 2, 3, \cdots$$

For some value $t_0 > 1$. To avoid the large variation of estimating $\theta^{(t)}$, γ_t should be very close to 0 at the end of the iteration. Therefore, t_0 control the speed of

convergence for $\theta^{(t)}$, a large t_0 will force the sampler to reach all subregions quickly. Liang et al. (2007) suggests t_0 can set to between $2m$ and $100m$, which m indicates the number of subregions. Let $J(x)$ represent the index of the subregions that the sample x belongs to, which takes value $(1, \cdots, m)$. With above notations, the SAMC algorithm can be describes as follows:

The SAMC algorithm:

1. (MH sampling) Simulate a sample $x^{(t)}$ by a MH algorithm with proposal distribution $q(x^{(t)}, y)$. To accept sample from proposal stage y with probability min(1,r), where transition rate r is

$$r = e^{\theta^{(t)}_{J(X^{(t)})} - \theta^{(t)}_{J(y^{(t)})}} \frac{\psi(y)q(y, x^{(t)})}{\psi(x^{(t)})q(x^{(t)}, y)}$$

2. (Weight updating) Set

$$\theta^* = \theta^{(t)} + \gamma_{t+1} H(\theta^{(t)}, x^{(t+1)})$$

If $\theta^* \in \Theta$, set $\theta^{(t+1)} = \theta^*$; otherwise, find a value c such that $\theta^* + cI_m \in \Theta$, c can be an arbitrary number that satisfies the condition $\theta^* + cI_m \in \Theta$.

We partition sample space in a flexible way in SAMC procedure, it will be allowed the existence of empty subregions. Liang et al. (2007) proved that when t $\to \infty$

$$\theta_i^{(t)} \to \begin{cases} C + \log(\int_{E_i} \psi(x)dx) - \log(\pi + d) & \text{if} \quad E_i \neq \emptyset \\ -\infty & \text{if} \quad E_i = \emptyset \end{cases} \tag{1.3}$$

holds almost surely, where $i = 1, \cdots, m$, $d = \sum_{j \in \{i: E_i = \emptyset\}} \pi_j/(m - m_0)$, m_0 is the number of empty subregions, and $C = -log(\int_{E_m} \psi(x)dx) + log(\pi_m + d)$.

Let $\hat{\pi}^{(t)} = P(X^{(t)} \in E_i)$ be the probability of sampling from the subregion E_i at iteration t. The equation (1.3) implies that as $t \to \infty$, $\hat{\pi}^{(t)}$ will converge to $\pi_i + d$ if $E_i \neq \emptyset$ and 0 otherwise, which is not dependent on its probability $\int_{E_i} p(x)dx$. It implies SAMC is capable of exploring the whole sample space.

The SAMC algorithm is a self-adjusting mechanism; it penalizes the over-visited subregions and rewards the under-visited subregions. This mechanism guarantees the algorithm not to be trapped by local optimal. We illustrate an example as follows to demonstrate the advantage of SAMC algorithm.

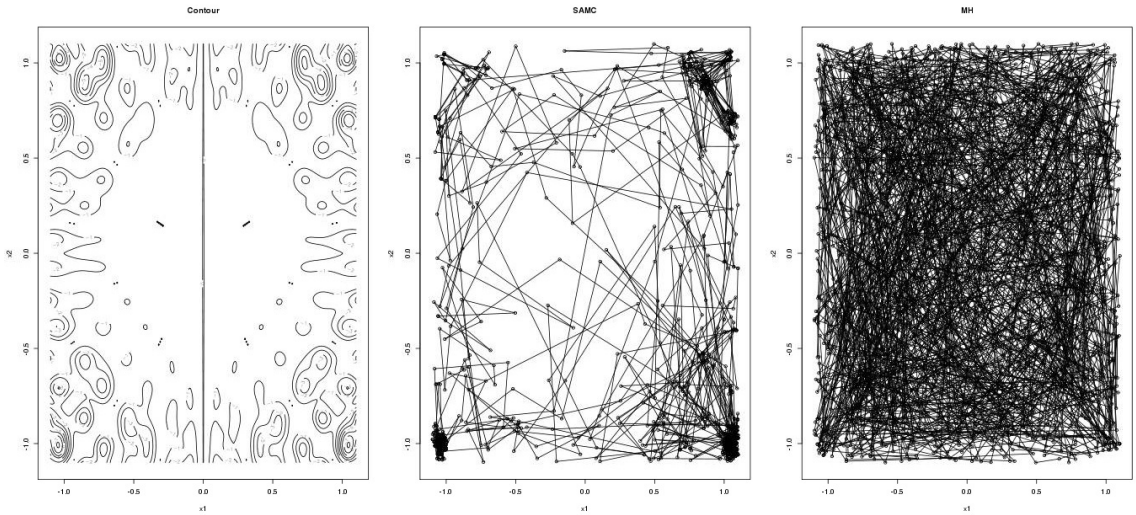


Figure 1.1: Comparison of SAMC algorithm and MH algorithm

Suppose the target density defined as $p(x) \propto e^{U(x)}$, where $x \in [-1.1, 1, 1]^2$ and $U(X) = -\{x_1 \sin(20x_2) + x_2 \sin(20x_1)\}^2 \cosh\{\sin(10x_1)x_1\} - \{x_1 \cos(10x_2) + x_2 \sin(10x_1)\}^2 \cosh\{\cos(10x_2)x_2\}$. Figure 1.1(a) shows contour plot of $U(x)$, it is clear that $U(x)$ has multitude of local minima. To investigate the benefit of SAMC algorithm compared with Metropolis Hasting algorithm, we run 20,000 iterations and

collect 2,000 samples at equally spaced time points. The proposal density for both algorithms are $N_2(x_t, 0.25^2 I_2)$. Figure 1.1(b) and 1.1(c) shows the evolving path of the 2,000 samples from SAMC and MH respectively. SAMC samples closely matches the contour plot of $U(x)$ whereas MH tends to sample uniformly in the sample space $\mathcal{X}$. Because we usually do not know the location of optimal values and how much the ratio of subregions' volume will be, we cannot control the simulation time spent on optimal regions in conventional MCMC algorithm. However, we can control the simulation time spent on optimal regions in SAMC by choosing the desired sampling distribution π. This scheme makes SAMC potentially more efficient than Metropolis Hasting in optimization, it increases the chance of locating the global optimizer.

Besides, user can partition sample space based on problem orientation. For examples, if our goal is to find the optimal of the target posterior density, we can partition the sample space according to the target density function; if our goal is model selection, then we can partition the sample space according to the index of models.

1.1.1 Dynamic Weighted Importance Sampling

From SAMC algorithm we have set samples $(x^{(1)}, \theta^{(1)}), \cdots, (x^{(n)}, \theta^{(n)})$ after weight updating stage. Liang (2009) showed SAMC is a dynamic weighting algorithms which in weight-estimation stage can evaluate $E_f h(\theta) = \int h(\theta) f(\theta|x)$ under the condition (A_1) and (A_2) (see Appendix A).

Let $\mathcal{A} \subset \mathcal{X}$ denote an arbitrary Borel set, and $\mathcal{A}^c$ denote the complementary set of $\mathcal{A}$. $\tilde{E}_1 = E_1 \cap \mathcal{A}, \tilde{E}_2 = E_1 \cap \mathcal{A}^c, \cdots, \tilde{E}_{2m-1} = E_m \cap \mathcal{A}, \tilde{E}_{2m} = E_m \cap \mathcal{A}^c$ are the new partition of sample space in $\mathcal{X}$, which denotes as an induced partition by $\mathcal{A}$. Thus $\tilde{J}(x_t)$ denote the index of the induce subregion x_t belongs to, and $\tilde{J}(x_1), \cdots, \tilde{J}(x_n)$ forms a sample from a nonhomogeneous Markov chain defined on the finite state

space $\{1, \cdots, 2m\}$. The transition matrices of the Markov chain can be defined as follows,

$$\tilde{P}_t(j|i) = \int_{\tilde{E}_i} \int_{\tilde{E}_j} [s_t(x, dy) + I(x \in dy)(1 - \int_x s_t(x, dz)]dx \tag{1.4}$$

where $s_t(x, dy) = q(x, dy)\min\{1, [f_{\theta_t}(y)q(y, x)]/[f_{\theta_t}(x)q(x, y)]\}$ and it follows from equation (1.4) that $f_{\theta_t}(x) \to f_\theta(x)$ almost surely. Thus, $\tilde{P}_t(j|i) \to \tilde{P}(j|i)$ almost surely. The stationary distribution of limiting Markov chain is $(\tilde{p}_1, \cdots, \tilde{p}_{2m})$, where $\tilde{p}_{2i-1} = \pi_i P_f(\mathcal{A}|E_i)$ and $\tilde{p}_{2i} = \pi_i P_f(\mathcal{A}^c|E_i)$ for $i = 1, \cdots, m$.

Proposition 1.1.1 Assume the conditions (A_1) and (A_2). For a set of samples generated by SAMC, we have

$$\lim_{x\to\infty} \frac{1}{n} \sum_{i=1}^{n} e^{\theta_{tJ(x_t)} I(x_t \in \mathcal{A})} = \sum_{i=1}^{m} w_i P_f(\mathcal{A}|E_i) \quad \text{almost} \quad \text{surely} \tag{1.5}$$

Consider the set samples $(x^{(1)}, \theta^{(1)}), \cdots, (x^{(n)}, \theta^{(n)})$ generated by SAMC with the partition $E_1, \ldots, E_m$. Let $Y^{(1)}, \ldots, Y^{(n')}$ denote the distinct samples among $(x^{(1)}, \ldots, x^{(n)})$. Generate a random variable/vector Y such that

$$P(Y = y^{(i)}) = \frac{\sum_{t=1}^{n} e^{\theta^{(t)}_{J(x^{(t)})}} I(x^{(t)} = y^{(i)})}{\sum_{t=1}^{n} e^{\theta^{(t)}_{J(x^{(t)})}}}, \qquad i = 1, \ldots, n' \tag{1.6}$$

where $I(\cdot)$ is an indicator function, $J(x^{(t)})$ denotes the index of subregion that the sample $x^{(t)}$ belongs to, and $\theta^{(t)}_{J(x^{(t)})}$ is finite because we restricted Θ in a compact in SAMC algorithm.

Theorem 1.1 Assume the conditions (A_1) and (A_2). For a set of samples generated by SAMC, the random variable/vector Y generated in equation (1.6) is asymptoti-

cally distributed as $f(.)$

$$P(Y \in \mathcal{A}) \to \int_{\mathcal{A}} f(x)dx$$

The Theorem 1.1 implies that for an integrable function $h(x)$, the expectation $E_f h(x)$ can be estimate by

$$\hat{E}_f h(x) = \frac{\sum_{t=1}^{n} e^{\theta^{(t)}_{J(x^{(t)})}} h(x_t)}{\sum_{t=1}^{n} e^{\theta^{(t)}_{J(x^{(t)})}}} \tag{1.7}$$

As $n \to \infty$, $\hat{E}_f h(x) \to E_f h(x)$. From equation (1.7) we can estimate the posterior mean with dynamic weight parameter $\theta^{(t)}$ from SAMC algorithm.

1.2 Stationary Test

Suppose our goal is to test the second order stationary of InSAR noise and we introduce Jun and Genton (2012) the stationary test approach as a start. The idea of this approach is based on to divide the spatial domain into two disjoint sub-domains, D_1 and D_2, and use the test statistic that is based on the difference between empirical estimators of covariances at given lags from the disjoint spatial sub-domains.

Consider a spatial random field $\{Z(s) : s \in Z^2\}$. Suppose the observations are taken over a 2-dimensional space of integer lattice points $\{Z(s) : s \in D\}$ where $D \subset Z^2$ be a finite set of lattice points. We denote the covariance function of lattice points $Z(s)$ as $C(s, s+h) = Cov\{Z(s), Z(s+h)\}$, where $s, s+h \in D$. If the case of random field $Z(s)$ is weakly stationary, then covariance can be defined as $C(s, s+h) = C(0, h) = C_0(h)$ for all $s \in D$, h, where C_0 is a stationary covariance function.

Consider an empirical moment estimator of $Cov(Z(s+h), Z(s))$ at a given lag h,

$$\hat{A}_Z(h) = \frac{1}{N(h)} \sum_{s_i, s_j} Z(s_i) Z(s_j) \tag{1.8}$$

Here $N(h)$ is the number of pairs of locations where the spatial lag is h. Jun and Genton (2012) have proven that $A_Z(h)$ at a given lag h is asymptotically normally distributed with mean zero and a certain covariance if Z is weakly stationary.

Suppose we have a set of spatial lags $\Lambda = (h_1, ..., h_m)$, and we split the domain into two disjoint sub-domains D_1 and D_2. Then we can define empirical moment estimator $\hat{G}_i = \{A(h; D_i) : h \in \Lambda\}$ for $i = 1, 2$ and test the null hypothesis of weak stationarity as follow,

H0: $\hat{A}(k; D_1) = \hat{A}(k; D_2)$

Ha: $\hat{A}(k; D_1) \neq \hat{A}(k; D_2)$

where $\hat{A}(k; D_i)$ is calculated from equation (1.8) and $k \in \Lambda$. We can rewrite the null hypothesis to $X\hat{G}_n$=0, where X is contrast matrix and $\hat{G}_n$ is 2 by 1 sample variogram for disjoint spatial sub-domains. For example, if $\Lambda = \{(1, 0), (0, 1)\}$ then we can write contrast matrix $X = (I_2, -I_2)$.

Let $g = \sqrt{|D_n|}(\hat{G}_n - (G, G)^T)$. Because g is asymptotically normally distributed, the test statistic is asymptotic chi-squared limiting distribution given by,

$$T = (Xg)^T (X\Sigma X^T)^{-1} (Xg) \sim \chi^2_{rank(X)} \tag{1.9}$$

We can apply equation (1.9) for InSAR noise data with different spatial lags Λ to investigate the stationarity.

1.3 Dissertation Structure

The arrangement for this dissertation is described as follows; Chapter II develops three combining methodologies with its theory and application. Those methods ex-

amine through one typical simulation study and one application data by comparing processing time and estimation accuracy with non-parallel SAMC algorithm. Chapter III is independent of Chapter II and is dedicated to a spatial statistics fields for investigating the effect of nonstationarity via stationary test and demonstrating the second order stationary assumption for InSAR noise. Chapter IV gives a summary of this dissertation and points out some directions for future research.

2. COMBINING STRATEGIES FOR PARALLEL STOCASTIC APPROXIMATION MONTE CARLO ALGORITHM OF BIG DATA

2.1 Introduction

Growing modern computer technology allows us to collect massive volumes of data, such as atmospheric information from satellite, genetic assay data, credit card transaction records, search engine logs, and climate data. Enterprises and scientist collect billion bytes of data and look for insights from their data. Therefore, modeling and mining with such massive data to meet research or business demands has become more popular in decades. However, it has difficulty in practically analyzing on a single commodity computer because of two main reasons. First of all, the amount of data is too large to fit in the memory of a single computer. Second, it takes too much time to analyze an entire data using current statistical methods without employing sampling scheme.

Parallel computing is one way to overcome those circumstances. Parallel computation is different from sequential instruction computation, because it allows many calculations to be carried out simultaneously. Parallelism has played a significant role in high-performance computing for many years, with advancement of multi-core and cloud computing platforms accessing hundreds to thousands processors. It not only significantly decreases processing time but overcomes the obstacle of local computational resource encountered in sequential instruction computation.

The main idea of parallel computation is to break the target problem into discrete parts and each processor analyzes smaller part of problem concurrently. One efficient method to analyzing each small part of the problem without making unnecessarily restrictive assumptions is Markov chain Monte Carlo (MCMC) method.

MCMC method has played an important role in statistical computations in the last two decades. It is a classic algorithm for sampling from probability distributions by constructing a Markov chain that has the desired distribution. It is a powerful computational tool for complex data structures. The metropolis-Hasting (MH) sampler is originated in statistical physics Metropolis et al. (1953) and Hastings (1970), it is a typical Markov chain Monte Carlo (MCMC) algorithm. The MH algorithm is a simulation of stochastic processes having probability densities known up to a constant. The Metropolis-Hastings algorithm is useful in calculating the necessary normalization factor, which often extremely difficult. However, it has some flaws. The most important flaw is that the MH algorithm is prone to getting trapped into local optima in simulations when it is non-trivial to sample target distribution. Second, if the data is too large, it requires a large number of iterations to perform the MH algorithm.

To avoid trapping in local optima problems advanced Monte Carlo algorithms have been developed for decades, such as parallel tempering (Geyer (1991)), simulated tempering (Marinari and Parisi (1992)), evolutionary Monte Carlo (Liang and Wong (2001)), dynamic weighting (Wong and Liang (1997)), multicanonical sampling (Berg and Neuhaus (1991)), the Wang-Landau algorithm (Wang and Landau (2001)), equi-energy sampler (Kou et al. (2006)), Stochastic Approximation in Monte Carlo algorithm (Liang et al. (2007)), among others.

Stochastic Approximation in Monte Carlo algorithm (Liang et al. (2007)) is a very sophisticated method in theory and application. It can yield significant time saving and more accurate estimation compared to conventional MCMC algorithms such as Metropolis-Hasting in complex data structure. Motivated by the successes of SAMC in analyzing complex problems and advantage of parallel computing, we propose a parallel SAMC algorithm which can be utilized on big data and is workable

in parallel computing.

Although parallel computation improves the capability to compute big data, it is still unclear about how current statistical methods can be applied in parallel systems and how to combine the parallel results. Big data and parallel computing have posed great challenges for the current statistical methodology.

In this study, we propose three combining strategies for Stochastic Approximation Monte Carlo in parallel computation. The chapter structures are as follows. Section 2.1 briefly introduces the parallel computation concept, Section 2.2 provides the theory of three proposed methodologies, Section 2.3 evaluates the performance of the proposed methodologies through a simulation example, and Section 2.4 applies parallel SAMC in the Shuttle data.

2.2 Parallel Computation

Using sequential programs to implement the instructions is the main way in the traditional computation and data analysis. The process runs a program on a single computer and executes on after another when previous instructions have been done. Figure (2.1) illustrates the process of sequential programs where only one instruction can be executed at a time. In the world there are many complex and massive volumes of data, such as climate data, biological assay data, credit card transaction records, transportation logs, and satellite's data, which can not be practically analyzed on a single node computer because their sizes are too large to fit in computer's memory.

A single computer with multiple processors or number of computers connected by a network can be defined as parallel computing resources. Benefited by the growth of computer technologies, nowadays we can use multiple computer resources to solve numerous computational problems. The idea of parallel computing was born in this sophisticated sense. Figure (2.2) shows the procedure of parallel computing which is

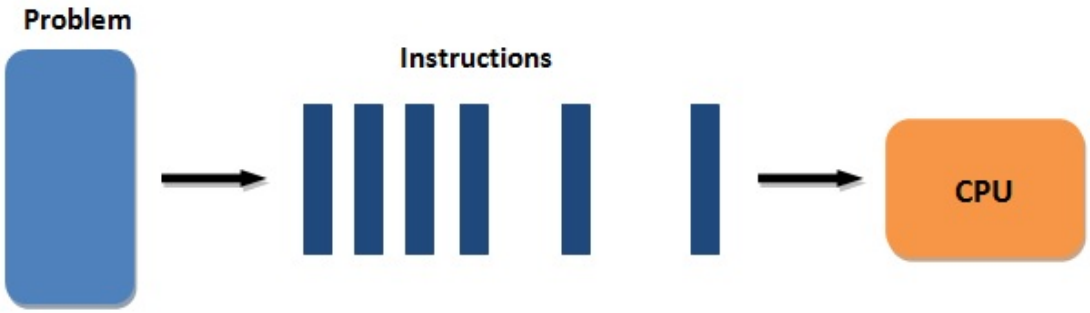


Figure 2.1: Structure of sequential computation

running tasks using multiple processors simultaneously and each processor works on its own section of the problem but can exchange information with each other.

There are several advantages performing parallel computing listed below,

1. Solving large problems

   Parallel computing can handle large and complex problems that it is impractical to solve on a sequential computing process, specially the CPU or memory are limited.

2. Saving process time

   Parallel computing divides a task into several sub-tasks and executes sub-tasks in parallel. If multiple processors perform computing tasks simultaneously, generally the execution speed will be faster.

3. Overcoming limited memory

   Single computer has finite memory resource. For large problems, using the memories from multiple computers may overcome this obstacle.

4. Using of non-local resources

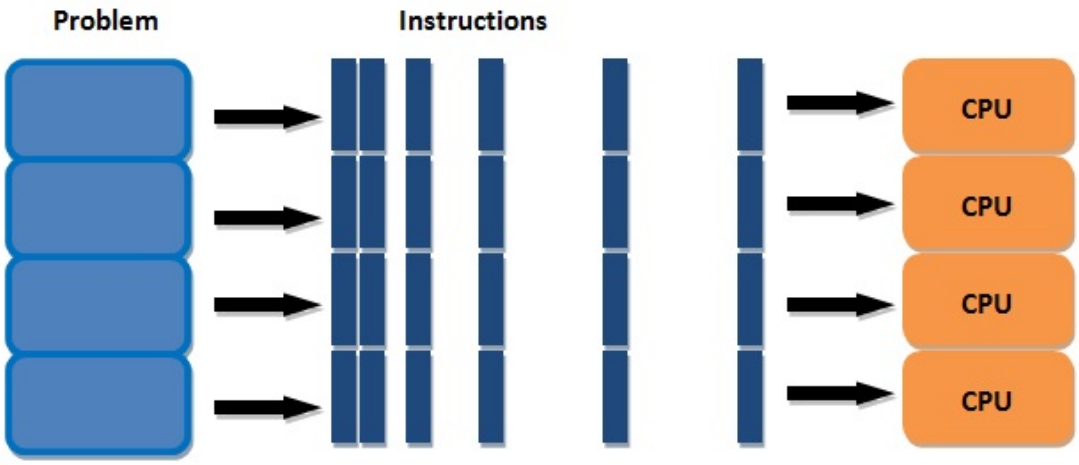


Figure 2.2: Structure of parallel computation

It can perform the computation on a wide area network when local computing resources are limited. For example: RIKEN Advance Institute for Computer Service (AICS) in Japan, contains 705,024 cores of CPU, National Supercomputing Center in Tianjin, China, contains 186,368 cores of CPU, and DOE/SC/OAK Ridge National Laboratory contains 224,162 cores of CPU (May, 2014).

Parallel computer has three type of memory architectures,

1. Shared memory

   Shared memory parallelism is the mechanism where all processors can access all memory as global space. Figure (2.3) illustrates the shared memory process, which can operate independently but share the same memory resources. In this parallelism, the data can be shared between tasks to make process faster and uniform due to the proximity of memory to CPUs.

2. Distributed memory

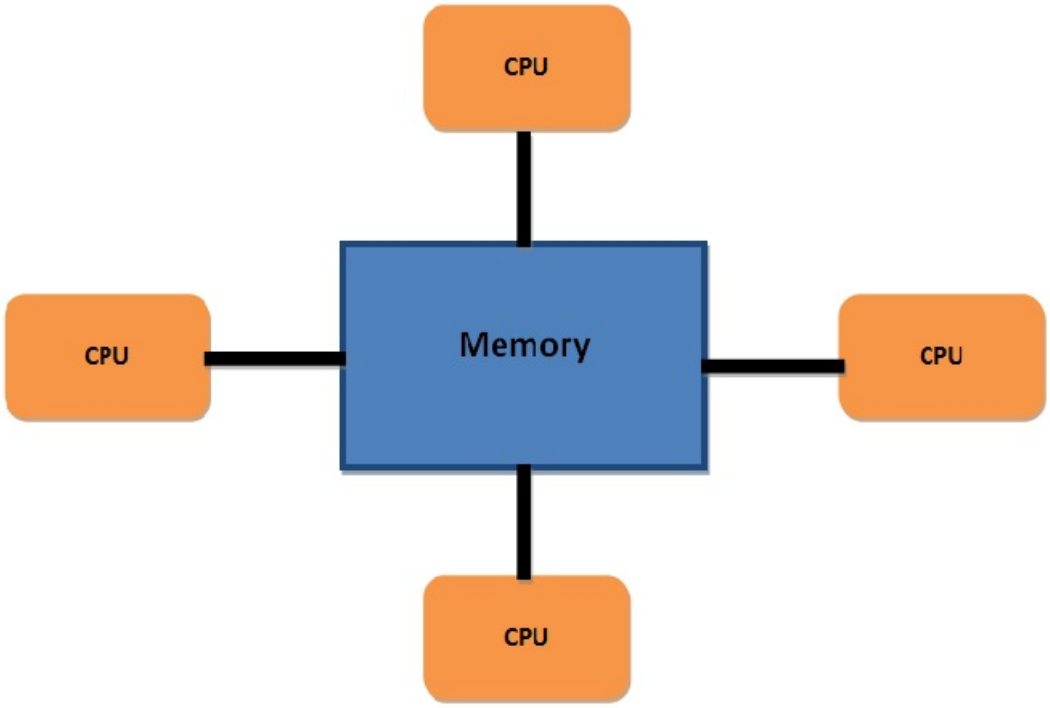


Figure 2.3: Shared memory parallelism

Distributed memory parallelism is that all processors have their own memory and operate independently. Figure (2.4) represents the distributed memory parallelism, each memory resides in one processor and is not accessible to other processors. In this parallelism, the memory systems require a communication network to connect inter-processor memory. When increase the number of processors during the task, the size of memory increase proportionately.

3. Hybrid distributed-shared memory

   Hybrid distributed-shared parallelism is the combination of shared and distributed memory machines. The largest and the fastest computer in the world are Hybrid distributed-shared memory system. The main advantage of combining distributed-memory and shared-memory is that communication can itself

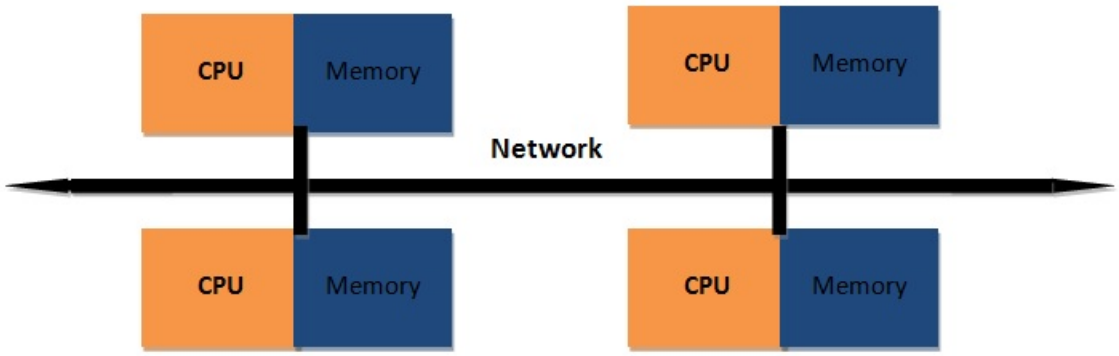


Figure 2.4: Distributed memory parallelism

be treated as another task or set of dependency tasks in a single instruction. Figure (2.5) illustrates the Hybrid distributed-shared memory. In each cluster of shared-memory, each machine computes its part of the solution and uses distributed-memory parallelism to communicate the computation results among others.

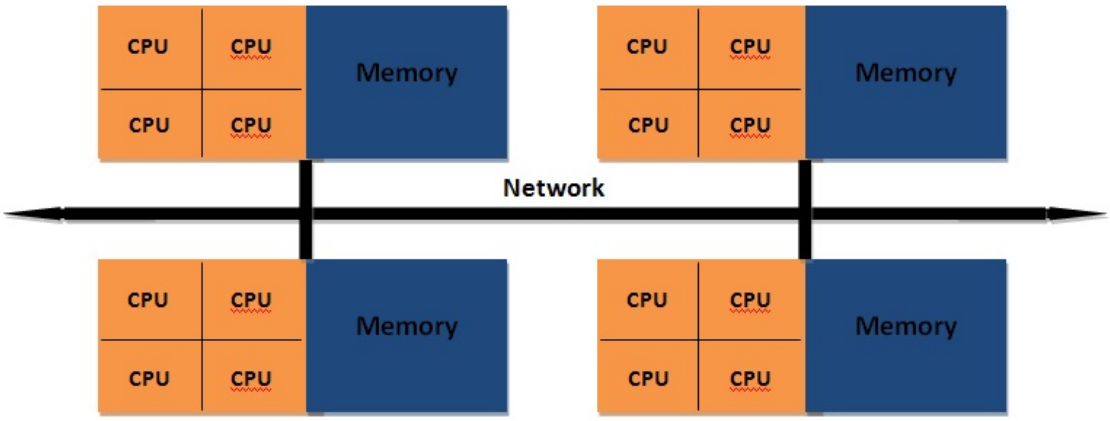


Figure 2.5: Hybrid distributed-shared memory parallelism

There are lots of parallel programming languages for automatically segmenting a task into multiple processes and/or threads to be executed on the available processors concurrently in a parallel system. Shared memory systems communicate by manipulating shared memory variables. Distributed memory systems communicate via message passing. OpenMP and POSIX Threads are the most widely used application passing interface in distributed memory system. Message Passing Interface (MPI) also plays a significant role for application program interface (API) in message passing system. It is a library specification for message-passing and can provide a powerful, efficient, and portable way to depict parallel programs. Nowadays, MPI can be used in C and FORTRAN for point-to-point communication in a parallel program.

2.3 Methodologies

The parallel SAMC algorithm can be viewed as constructing an efficient algorithm by partitioning our target density into disjoint small parts, performing a Bayesian analysis in each part, and combining the analysis results from each part to yield an overall posterior model. To make this procedure work well, data-splitting and data combination framework are essential.

The data-splitting procedure can be done according to the number of available computational servers we have and the complexity of the given problem. For example, if we aim to sample from a trail density which is a high dimension function, then we must not split into too many parts to ensure number of parameters $p << n_i$ in general cases. The randomly split data with the same number of observation in each part will contain approximately the same amount of information of the entire data. We recommended splitting data randomly n/k with a reasonable number of subset k which does not lead to a high dimension problem, but is workable for concurrent

computational units.

The SAMC algorithm will perform within each subsets with the same working prior, iteration number, proposal density and the same partition of the sample space and generate $(x^{(1)}, \theta^{(1)}), \cdots, (x^{(n)}, \theta^{(n)})$ posterior samples in each subsets. The posterior mean in each subset denoted as $E_i h(\theta) = \int h(\theta)\pi(\theta|\mathcal{Y}_i)$, where $\mathcal{Y}_i$ denotes the ith subset and $h(\theta)$ is an integrable function. By Theorem 1.1 we can estimate $E_i h(\theta) = \int h(\theta)\pi(\theta|\mathcal{Y}_i)$ by $\sum_{j=1}^{n} h(\theta_j^{(i)}) w_j^{(i)} / \sum_{j=1}^{n} w_j^{(i)}$, which will converge to $E_i h(\theta)$ almost surely as n $\to \infty$

Our objective is to estimate the posterior mean of big data. When SAMC has been done on each subset and the posterior samples for each subset are available, we can perform combining strategies to yield an overall posterior mean for big data. Three combining strategies describe as follows; section 2.3.1 describes Bayesian Inference via Divided-and-Combined (D&C), section 2.3.2 introduces Bayesian inference via Divided-and-Resample (D&R) and section 2.3.3 describes Weighted Combination via SAMC Importance Sampling.

2.3.1 Bayesian Inference via Divided-and-Combined

It is known that under mild regularity conditions, the maximum likelihood (ML) estimate $\hat{\theta}$ of θ is asymptotic normal $\hat{\theta} \sim N(\theta, \Sigma)$, where Σ denotes the associated large-sample covariance matrix. Let $\mathcal{Y}_1, \cdots, \mathcal{Y}_k$ denote the subsets from random partition of the big data $\mathcal{Y} = \{X_i : i = 1, \cdots, n\}$ each with sample size $n_k \approx n/k$. Let $\hat{\theta}_i$ denotes posterior mean estimator of each subset. For large n_k, we have approximately $\hat{\theta}_i \sim N(\theta, \Sigma_k)$, where Σ_k is the large sample covariance matrix associated with $\hat{\theta}_i$.

For the fixed effect model we assume that there exist a true mean effect which is shared by all individual studies. Each individual observed effects will be distributed

about θ with a within-study error ϵ_i, where ϵ_i is normal distribution with mean zero and variance Σ_k. That is,

$$\hat{\theta}_i = \theta + \epsilon_i \qquad \text{where} \quad \epsilon_i \sim N(0, \Sigma_k)$$

To calculate ML estimator based on the above formula, the likelihood function can be form as follows,

$$\text{log-likelihood} \propto \sum_{i=1}^{k} (-\log|\Sigma_i| - \frac{1}{2}(\hat{\theta}_i - \theta)^T \Sigma_i^{-1}(\hat{\theta}_i - \theta))$$

$$\frac{dl}{d\theta} = \sum_{i=1}^{k} ((\hat{\theta}_i - \theta)\Sigma_i^{-1}) = 0$$

$$\hat{\theta} = (\sum_{i=1}^{k} \hat{\theta}_i \Sigma_i^{-1})(\sum_{i=1}^{k} \Sigma_i^{-1})^{-1}$$

It can be defined as a weight average where the weights are determined by covariance matrix in each subset. The posterior mean of big data in the recombination step can be estimated by,

$$\tilde{\theta} = [\Sigma_1^{-1} + \cdots + \Sigma_k^{-1}]^{-1}[\Sigma_1^{-1}\hat{\theta}^{(1)} + \cdots + \Sigma_k^{-1}\hat{\theta}^{(k)}]$$

The Bayesian inference via divided-and-combined algorithm is defined as follows,

1. (*Partition*) Partition the big data into k number of subsets with each containable in the memory of a single computer node.

2. (*Parallel Simulation*) Simulate from the posterior of each subset in parallel using SAMC
   (a) (ω-estimation) Run SAMC to estimate ω_i's
   (b) (MH sampling) Simulate samples from $f_{\hat{w}}(x)$ using a MCMC algorithm

3. (*Aggregation*) Combine the kth posterior mean into the big data posterior mean with $\tilde{\theta} = [\Sigma_1^{-1} + \cdots + \Sigma_k^{-1}]^{-1}[\Sigma_1^{-1}\hat{\theta}^{(1)} + \cdots + \Sigma_k^{-1}\hat{\theta}^{(k)}]$

The flowchart of implementing parallel computing for Bayesian Inference via Divided-and-Combined (D&C) is shown in Figure (2.6). The parallel implementation involves two message passing interface (MPI) instructions: broadcast and reduce. The broadcast instruction is to assign the SAMC algorithm instructions from the master processor to slave processors, and each slave processors calculate posterior mean $\hat{\theta}_i$ concurrently. The reduce instruction is to reduce values on all processors to a single value on the master processor, which is performing the aggregation step. In this matter, Bayesian Inference via D&C combining strategy in parallelism procedure will save a lot of computational time.

2.3.2 Bayesian Inference via Divided-and-Resample

Markov Chain Monte Carlo (MCMC) methods have been proved to be a useful analyst tool for complex Bayesian inference. Let $\pi(\theta)$ be the prior distribution for the big data. The posterior distribution can be defined as

$$\pi(\theta|\mathcal{Y}) = \pi(\theta)\prod_{i=1}^{n} f(X_i|\theta) \qquad \text{for} \quad \theta \in \Theta$$

where $\mathcal{Y}$ represents the big data and $f(X_i|\theta)$ represents the likelihood function of big data. Suppose that a Monte Carlo sample $S_k = \{\theta^{(i,k)}; i = 1, \cdots, N_k\}$ is obtained from k-th posterior distribution, that is,

$$\pi_k(\theta) \propto \pi^{(k)}(\theta)\ell_k(\theta)$$

where the priors $\pi^{(k)}(\theta)$ are the working prior density. In terms of the working posteriors distribution $\pi_k(\theta)$, the desired posterior distribution can be written as

$$\pi(\theta|\mathcal{Y}) \propto \eta(\theta) \prod_{k=1}^{k} \pi_k(\theta)$$

where $\eta(\theta) \propto \pi(\theta) / \prod_{k=1}^{K} \pi^{(k)}(\theta)$, and $\theta \in \Theta$. That is, $\eta(\theta)$ equal to a constant if $\prod_{k=1}^{k} \pi^{(k)}(\theta) = \pi(\theta)$.

The interesting issues raised here include (1) creating a posterior sample $\{\theta^{(j)} : j = 1, \cdots, N\}$ from the individual Monte Carlo samples, and (2) approximating the posterior mean for any given function $h(\theta)$ based on the individual Monte Carlo samples and evaluating the associated Monte Carlo error. To address this problem can be challenging. However, it is not impossible because the individual sample $S_1, \cdots, S_k$ contain information in each working posterior density $\pi_k(\theta)$. The only challenge is that the recombination needs to be done via random set intersection. To extend the idea of random set intersection, we introduce the Dempster-Shafer theory.

Dempster's rule of combination is an application of random set intersection from Dempster-Shafer theory. It can be used to combine independent sets of probability mass density in specific situations. The rule is to derive the common shared belief function between different individual studies and ignore all the conflicting (non-shared) portions. Let $\mathcal{X}$ be the universal which represents all possible states of a system to be considered, and $2^{\mathcal{X}}$ be the power set which is the set of all subsets of $\mathcal{X}$. A, B, and C denote the subset of the power set. The belief function m(.) of each

subset is a mapping from power set to a real number [0,1] and it has two properties:

$$m(\emptyset) = 0$$
$$\sum_{A \in 2^X} m(A) = 1$$

The combination (joint belief function) is calculated from the two sets of masses function m_1 and m_2 in the following form,

$$m_{1,2}(\emptyset) = 0$$
$$m_{1,2}(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B) m_2(C)$$
$$\text{where} \quad K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$$

With the Dempster-Shafer theory of belief functions, we can generate an algorithm to demonstrate entire data posterior samples in the recombination operation. Here we consider a simple example.

Consider a simple case with a sample size n=2, X_1 and X_2. Suppose that the sample model is the Gaussian distribution with unknown mean θ and unit variance, i.e., $X_1, X_2 \sim N(\theta, 1)$. We divide the data into two (k =2) subsets consisting of a single observation for each subset and take a flat prior, $\pi(\theta) \propto 1$ for θ. Bayesian analysis is conducted on each subset using the working prior $\pi^{(k)}(\theta) \propto 1^{1/k}$. It produces the posterior distribution with posterior samples $S_k = \{\theta^{(i,k)} : i = 1, \cdots, N_k\}$ for k =1 and 2. By definition of Dempster-Shafer combination rules, consider the

target posterior probability in a small neighborhood of θ_0, $[\theta_0, \theta_0 + \Delta]$.

$$Pr(\theta \in [\theta_0, \theta_0 + \Delta] | X_1, X_2) \approx \Delta^{-1} \int_{\theta_0}^{\theta_0+\Delta} \int_{-\infty}^{\infty} \pi_1(\theta_1|X_1)[\pi_2(\theta_2|X_2) I_{\theta_1 \le \theta_2 \le \theta_1+\Delta}] d\theta_2 d\theta_1 \tag{2.1}$$

The posterior is defined in the expended space $(-\infty, \infty) \times (-\infty, \infty)$ for (θ_1, θ_2), and can be used in the recombination step. Equation (2.1) suggests to replace the posterior draws $\theta^{(i,2)}$ with a small intervals, i.e.,$[\theta^{(i,2)} - \Delta/2, \theta^{(i,2)} + \Delta/2]$. The combination algorithm with an acceptance-rejection rules is defined as follows,

1. (*Re-sample*) Take a sample θ_1^* from $S_1 = \{\theta^{(i,1)} : i = 1, \cdots, N_1\}$ and a sample θ_2^* from $S_2 = \{\theta^{(i,2)} : i = 1, \cdots, N_2\}$ at random

2. (*Recombine*) Accept θ_1^* if $\theta_1^* \in [\theta_2^* - \Delta/2, \theta_2^* + \Delta/2]$

3. Repeat step 1 and step 2 until we get enough recombination posterior samples

For computational efficiency, this recombination re-sampling method can be simplified to weighted sampling from $S_1 = \{\theta^{(i,1)} : i = 1, \cdots, N_1\}$ with weights depending on S_2. In a numerical example, we took $\theta_1 = -1$, $\theta_2 = 1$ and $N_1 = N_2 = 10,000$ and created a posterior sample of size 1,000 using the above recombination re-sampling algorithm. The results are shown in Figure (2.7) in terms of QQ-normal plots. It is shown that the posterior distribution $p(\theta_1|X_1)$ and $p(\theta_2|X_2)$ are Gaussian distribution with unit variance and mean $\theta_1 = -1$ and $\theta_2 = 1$ respectively. The target posterior based on both X_1 and X_2 is $N((X_1 + X_2)/2, 1/2) = N(0, 1/2)$. We see that the recombination re-sampling results are consistent with the theoretical results.

The Bayesian inference via divided-and-resample algorithm is defined as follows,

1. (*Partition*) Partition big data into k number of subsets with each containable in the memory of a single computer node.

2. (*Parallel Simulation*) Simulate from the posterior of each subset in parallel using SAMC

   (a) (ω-estimation) Run SAMC to estimate ω_i's

   (b) (MH sampling) Simulate samples from $f_{\hat{w}}(x)$ using a MCMC algorithm

3. (*Aggregation*) Combine the kth posterior mean into big data posterior mean with Recombination Re-sampling algorithm

   (a) Sample a posterior sample $x_i^{(1)}$ with probability respect to $\hat{g}_i = e^{\theta_i^{(1)}}$ from subset 1

   (b) Sample a posterior sample $x_i^{(2)}$ with probability respect to $\hat{g}_i = e^{\theta_i^{(2)}}$ from subset 2

   (c) Accept $x_i^{(1)}$ from subset 1 if $x_i^{(1)} \in [x_i^{(2)} - \Delta/2, x_i^{(2)} + \Delta/2]$. Otherwise return to step (a).

   (d) Repeat (a)-(c) for subset 3 & 4 and so forth until combined k subsets.

Figure (2.8) shows the flowchart of the implementation of parallel SAMC algorithm. The parallel implementation involves two main message passing interface (MPI) instructions, receive and send. The send instruction is to send the results from the slave processor to master processors and the receive instruction is the opposite way. In implementing the parallelism for Bayesian inference via Divided-and-Resample algorithm, the slaves send all the posterior samples with corresponding $\hat{g}_i = e^{\theta_i^{(1)}}$ to the master. The master received the data in order to perform the aggregation step of recombination re-sampling algorithm.

2.3.3 Weighted Combination via SAMC Importance Sampling

Importance sampling is more than a variance reduction method. It is an efficient approach to sample a posterior sample. In essence, we draw a posterior sample from an alternative distribution whose support is concentrated in the truncation region. The principle of importance sampling is,

$$\int_f xf(x)dx = \int_g x\frac{f(x)}{g(x)}g(x)dx = \int_g xw(x)g(x)dx$$

That is, sampling x from $f(x)$ distribution equivalent to sampling $x \times w(x)$ from $g(x)$ distribution, with importance sampling weight $w(x) = \frac{f(x)}{g(x)}$.

Let $\{Y_1, \ldots, Y_n\}$ denotes a single subject of the big data, and $\mathcal{Y}_1, \ldots, \mathcal{Y}_k$ be the subsets of big data, where $\mathcal{Y}_i$ is one of the subset. Let $\pi(\theta)$ be the prior distribution for big data. Under the assumption of i.i.d random variables and Bayesian rule, the posterior density can be shown by following,

$$\pi(\theta|\mathcal{Y}_1, \ldots, \mathcal{Y}_k) = \frac{f(\mathcal{Y}_1, \ldots, \mathcal{Y}_k|\theta)\pi(\theta)}{f(\mathcal{Y}_1, \ldots, \mathcal{Y}_k)} \propto \pi(\theta|\mathcal{Y}_1)\prod_{i=2}^{k} f(\mathcal{Y}_i|\theta) \tag{2.2}$$

where θ denotes the parameter vector of the model, $f(\mathcal{Y}_i|\theta)$ is the likelihood function for each subset of big data, and $\pi(\theta|\mathcal{Y}_1)$ denotes the posterior density of the first subset.

In order to get correctly importance weighted with respect to $\pi(\theta|\mathcal{Y}_1 \ldots \mathcal{Y}_k)$, we motivate by the principle of importance sampling, and rewrite our importance sampling weight as follows,

$$\frac{\pi(\theta|\mathcal{Y}_1, \ldots, \mathcal{Y}_k)}{\pi(\theta|\mathcal{Y}_1)} \propto \frac{f(\mathcal{Y}_1, \ldots, \mathcal{Y}_k|\theta)\pi(\theta)}{f(\mathcal{Y}_1|\theta)\pi(\theta)} = f(\mathcal{Y}_2|\theta) \times \cdots \times f(\mathcal{Y}_k|\theta) \tag{2.3}$$

It implies that we can draw posterior samples from the posterior density of first subset with importance sampling weight which is product of likelihood function for the rest of subsets. Suppose $(\theta_1^{(1)}, w_1^{(1)}, \ldots, \theta_n^{(1)}, w_n^{(1)})$ is a set of posterior sample we drawn from SAMC algorithm for the first subset of $\pi(\theta|\mathcal{Y}_1)$. Here $w_i = \int_{E_i} \psi(x)dx$ is invariant with respect to the importance weights (IWIW) (Liang (2009)). Thus, the correctly weighted $(w_1^{(1:k)}, \ldots, w_n^{(1:k)})$ for k subset of big data can be form as

$$w_l^{(1:k)} = w_l^{(1)} \prod_{i=2}^{k} f(\mathcal{Y}_i|\theta) \qquad l = 1, \ldots, n \tag{2.4}$$

The correctly weighted of the proposal density in the equation (2.4) can be substituted for any subset posterior density. Thus, the generalized correct weights are defined as,

$$w_l^{(1:k)} = w_l^{(j)} \prod_{i=1, i\neq j}^{k} f(\mathcal{Y}_i|\theta) \qquad l = 1, \ldots, n \tag{2.5}$$

We finally have a correctly weighted with posterior samples $(\theta_1, w_1^{(1:k)}, \ldots, \theta_n, w_n^{(1:k)})$ in each subset, and we can estimate posterior mean $\hat{\theta}^{(i)}$ $(i = 1, \cdots, k)$ for each subset by equation (2.5). Regarding contribution for big data posterior mean is not equal in each subset, we estimate it by weight average in following form,

$$\tilde{\theta} = [\Sigma_1^{-1} + \cdots + \Sigma_k^{-1}]^{-1}[\Sigma_1^{-1}\hat{\theta}^{(1)} + \cdots + \Sigma_k^{-1}\hat{\theta}^{(k)}]$$

where Σ_i $(i = 1, \cdots, k)$ is the covairance matrix associated with $\hat{\theta}^{(i)}$. The weighted combination via SAMC importance sampling algorithm is defined as follows,

1. (*Partition*) Partition the big data into k number of subsets with each containable in the memory of a single computer node.

2. (*Parallel Simulation*) Simulate from the posterior of each subset in parallel using SAMC

   (a) (ω-estimation) Run SAMC to estimate ω_i's

   (b) (MH sampling) Simulate samples from $f_{\hat{w}}(x)$ using a MCMC algorithm

3. (*Aggregation*) Combine the kth posterior mean into the big data posterior mean with weighted combination algorithm

   (a) Compute correctly weighted for k subset

   (b) Compute posterior mean with correctly weighted for each subset

   (d) Combine the kth posterior mean into the big data posterior mean with $\tilde{\theta} = [\Sigma_1^{-1} + \cdots + \Sigma_k^{-1}]^{-1}[\Sigma_1^{-1}\hat{\theta}^{(1)} + \cdots + \Sigma_k^{-1}\hat{\theta}^{(k)}]$

The implementation of parallel SAMC algorithm is depicted on a flowchart shown in Figure (2.9). The passing message instructions we used for weighted combination algorithm are broadcast and reduce. The slaves calculate the subset posterior mean with corresponding combined weights, then reduce them to the master processor for estimating big data posterior mean.

2.4 Simulation Study

In this section, we demonstrate the computational advantage of parallel SAMC algorithm by simulation studies. We consider the problem of estimating the logistic regression coefficients with five covariates $x_1, \cdots, x_5$. Let Y_i be the binary response and $x_i = (1, x_1, x_2, x_3, x_4, x_5)^T$. Then the logistic regression model can be written as

$$P(Y_i = 1) = \frac{\exp\{x_i^T\beta\}}{1 + \exp\{x_i^T\beta\}}, \qquad i = 1, \cdots, n$$

where $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T$ is coefficient vector for regression, and n is the number of observations. In our simulations, we set the true regression coefficients as

$\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T = (1, 2, 3, 4, 5, 6)^T$ and the sample size as $n = 10^6$. The predictors are drawn independently from the standard normal distribution. This example is modified from one example of Lin and Xi (2011).

To conduct parallel SAMC algorithm for Bayesian analysis in this example, we set an independent normal prior distribution for each component of β with mean 0 and variance 400. In order to explore the performance of parallel computing with different size of subset, we try three type of settings with $k = 10, 20$, and 40 and n_k are equal to 10^5, 5×10^4 and 2.5×10^4 by comparing it with big data $n = 10^6$. Recall that k denotes the number of subset and n_k denotes the size of each subset.

For each choice of (k, n_k), we run 20 independent parallel SAMC algorithm to demonstrate the CPU time saving and accuracy of parameter estimation. Each run consisted of 50,000 iterations. The proposal distribution used in MH step is a Gaussian random-walk $y \sim N(\beta, s^2)$, where s^2 is targeted to have a desired acceptance rate. In all simulations, we set $s = 0.15$ in order to keep a reasonable acceptance rate.

We perform parallel computation by using Brazos cluster at Texas A& M University, which is designed to meet the high-throughput computing and is capable of executing MPI applications. It contains 311 computing nodes with a total of 9.3TB of RAM. Table (2.1), Table (2.2) and Table (2.3) summarize the estimates of β in parallel SAMC with different number of subset in 20 independent runs for three combining methods. The results show the posterior mean, standard error of posterior mean in 20 independent runs and CPU process time in different number of subset. The CPU process time is measured by one run for different number of subsets. As number of subset k increase, the CPU time decrease dramatically, specially $k = 40$ is at least 20 times faster than big data estimation in three methods. As expected, the estimator is close to true parameter in different number of subset and the variation

is quite small in the proposed three methods. This example confirms the validity of the divided-and-combined strategy in parallel SAMC algorithm and also shows that parallel SAMC is quite robust to the choices of number of subset k. Parallel SAMC algorithm can indeed lead to a great saving of computational time without losing any information.

Table 2.1: Posterior mean estimation for Bayesian inference via D&C method in different number of subset

Subset	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$	CPU Time
k=1	0.991	2.000	3.011	4.018	5.019	6.018	127m 50s
	(0.0016)	(0.0015)	(0.0018)	(0.0018)	(0.0029)	(0.0025)	
k=10	0.989	2.004	3.011	4.021	5.018	6.023	13m 38s
	(0.0027)	(0.0035)	(0.0109)	(0.0140)	(0.0158)	(0.0218)	
k=20	0.996	1.992	2.988	3.976	4.978	5.980	7m 33s
	(0.0028)	(0.0049)	(0.0056)	(0.0085)	(0.0088)	(0.0111)	
k=40	0.978	1.961	2.928	3.916	4.897	5.867	6m 27s
	(0.0031)	(0.0058)	(0.0082)	(0.0107)	(0.0132)	(0.0137)	

Table 2.2: Posterior mean estimation for Bayesian inference via D&R method in different number of subset

Subset	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$	CPU Time
k=1	0.991	2.000	3.011	4.018	5.019	6.018	127m 50.47s
	(0.0016)	(0.0015)	(0.0018)	(0.0018)	(0.0029)	(0.0025)	
k=10	0.990	1.998	2.995	4.000	4.995	5.995	17m57.54s
	(0.0132)	(0.0203)	(0.0283)	(0.0319)	(0.0406)	(0.0533)	
k=20	0.997	1.992	2.985	3.972	4.979	5.975	13m0.29s
	(0.0179)	(0.0237)	(0.0359)	(0.0271)	(0.0424)	(0.0452)	
k=40	0.972	2.054	2.976	4.083	5.024	6.058	11m33.07s
	(0.0201)	(0.0254)	(0.0419)	(0.0352)	(0.0574)	(0.0471)	

Table 2.3: Posterior mean estimation for weighted combination via SAMC importance sampling method in different number of subset

Subset	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$	CPU Time
k=1	0.991	2.000	3.011	4.018	5.019	6.018	127m 50.47s
	(0.0016)	(0.0015)	(0.0018)	(0.0018)	(0.0029)	(0.0025)	
k=10	0.989	1.996	2.992	3.991	4.988	5.981	13m20s
	(0.0040)	(0.0065)	(0.0110)	(0.0090)	(0.0226)	(0.0247)	
k=20	1.017	1.994	2.986	4.006	5.002	5.993	8m37s
	(0.0032)	(0.0031)	(0.0058)	(0.0058)	(0.0063)	(0.0075)	
k=40	0.997	1.997	2.978	3.984	4.968	5.968	6m43s
	(0.0063)	(0.0066)	(0.0140)	(0.0135)	(0.0235)	(0.0228)	

The likelihood function for simulation example is not complicated and can be evaluated very fast. In next section we will use one real example for which the likelihood function is more complex in order to investigate the time saving and accuracy of estimation.

2.5 Application Data in Shuttle Data

Shuttle dataset is a datalog original from NASA which is about the position of radiator within the space shuttle "Challenger", specially an investigation ensued into the reliability of the shuttle's propulsion system. The full dataset is provided by Jason Catlett who was at the Basser Department of Computer Science, University of Sydney, Australia and data available at UCI machine learning repository (http://archive.ics.uci.edu/ml/).

There are seven target classes: Rad Flow, Fpv Close, Fpv Open, High, Bypass, Bpv Close, and Bpv Open, which represent the location of space shuttle and 9 explanation variables from three sensors monitored at one second. Approximately 80 percent of the data belong to Rad Flow class and only 6 observations in the class of Bpv Close. The data was provided as two portions, training dataset which con-

tains 43,500 observations and testing dataset which contains 14,500 observations. Our goal is to successful classify Rad Flow group (class 1) from 9 quantitative variables and shorten the computational time. We consider the problem of estimating the multinomial logistic regression coefficients with nine covariates $x_1, \cdots, x_9$. Let $Y_i = (1, \cdots, 7)$ be the seven outcomes and $x_i = (1, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)^T$. Then the multinomial logistic regression model can be written as

$$
\begin{aligned}
P(Y_i = 1) &= \frac{\exp\{x_i^T \beta_1\}}{1 + \sum_{k=1}^{6} \exp\{x_i^T \beta_k\}} \\
P(Y_i = 2) &= \frac{\exp\{x_i^T \beta_2\}}{1 + \sum_{k=1}^{6} \exp\{x_i^T \beta_k\}} \\
&\vdots \\
P(Y_i = 7) &= \frac{1}{1 + \sum_{k=1}^{6} \exp\{x_i^T \beta_k\}}
\end{aligned}
$$

where $\beta_k = (\beta_{0k}, \beta_{1k}, \beta_{2k}, \beta_{3k}, \beta_{4k}, \beta_{5k}, \beta_{6k}, \beta_{7k}, \beta_{8k}, \beta_{9k})^T$ is coefficient vector for regression. Before processing the data, we normalized 9 quantitative variables to be a mean 0 and unit variance and divided the training data into three and five subsets (k =3 and 5) on data splitting procedure. The observations we draw for each subset are constructed by randomly due to the unbalanced number of observations in seven class, the ranges from 0.014% to 78.4 %. Each subset contains 11,369 and 6,821 observations for Rad Flow (class 1) in three subsets and five subsets respectively. To conduct parallel SAMC algorithm, we set an independent normal prior distribution for each component of parameters with mean 0 and variance 400. The proposal distribution used in MH step is a Gaussian random-walk $y \sim N(\beta, 0.2^2)$ with reasonable acceptance rate and each subset we consisted of 50,000 iterations.

Table (2.4) summarizes the results of prediction rate for training data and testing data in three combining strategies we proposed in section 2.3. As shown in the table,

Table 2.4: Prediction rate and CPU time for three combing strategies

subset		Training Data %	Prediction Data %	CPU Times
k=1		78.99	79.51	7m23.719s
k=3				
	ML D&C	79.57	80.06	2m 38s
	Bayesian D&R	79.52	80.04	4m 18s
	Weighted combination	79.45	79.86	2m 37s
k=5				
	ML D&C	78.99	79.40	2m 31s
	Bayesian D&R	79.77	80.23	4m 41s
	Weighted combination	81.36	81.43	2m 27s

the prediction rate is consistently higher than 79.5% in all setting without losing much information and with more accuracy prediction rate compare with entire data estimation ($k = 1$). About the computational time, it is extremely faster than entire data estimation, partition $k = 5$ is three times faster than big data estimation. We note that the above settings are not necessarily optimal. In this application example we demonstrate that parallel SAMC algorithm with three combining strategies can be effectively used for big data problems.

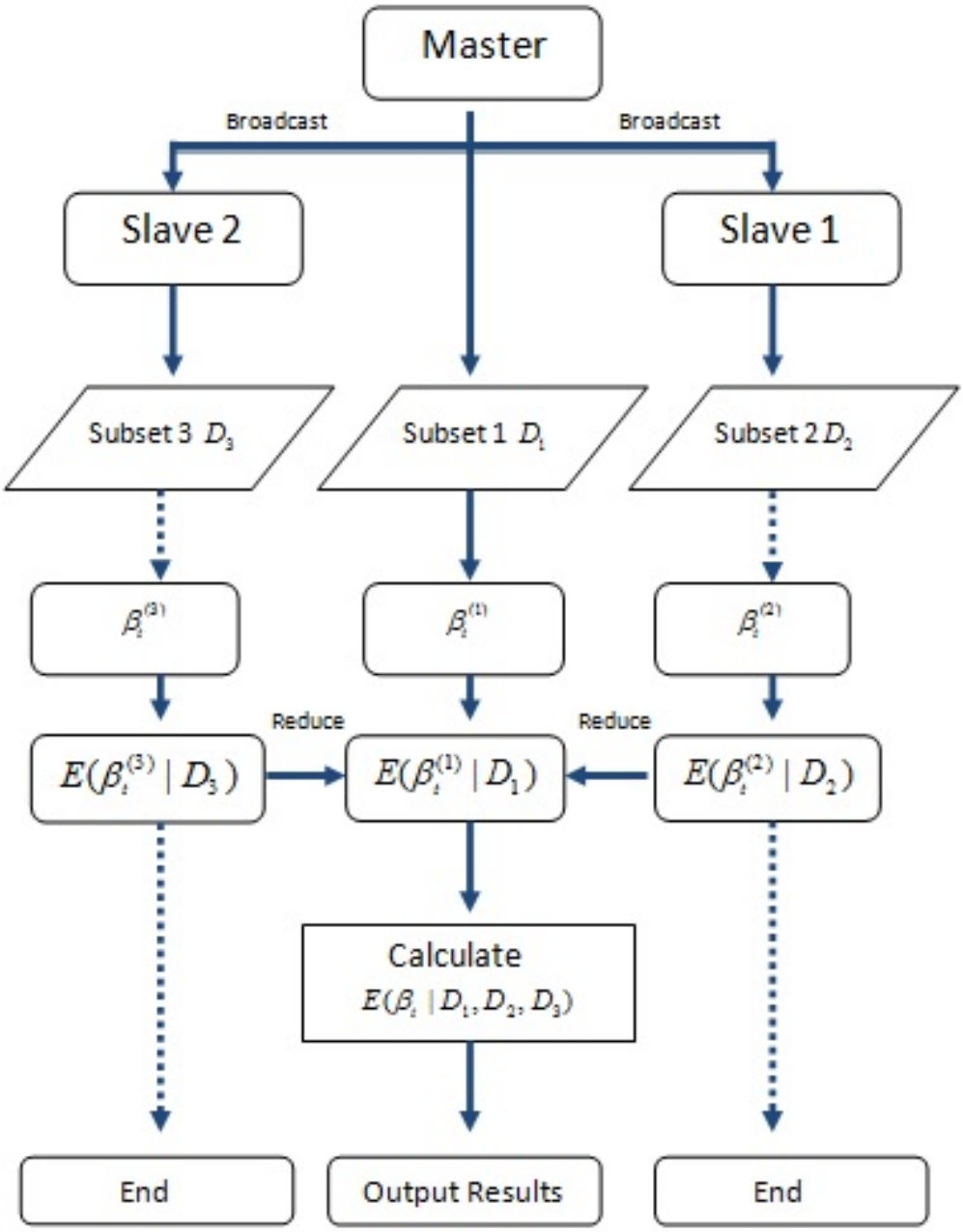


Figure 2.6: Flowchart for parallel computation in Bayesian inference via Divided-and-Combined

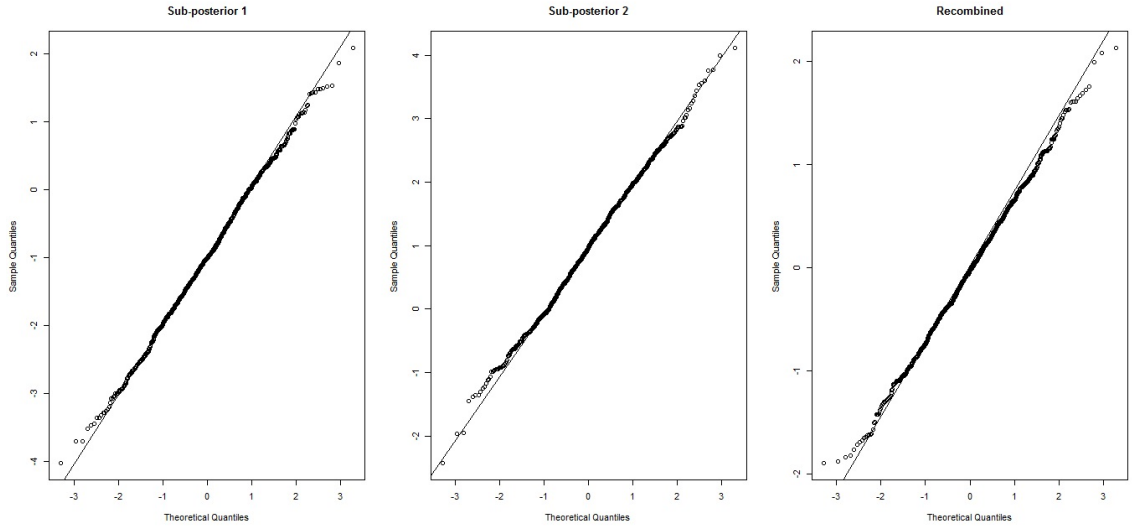


Figure 2.7: Q-Q plot for $\pi(\theta^{(1)}|X_1)$, $\pi(\theta^{(2)}|X_2)$ and $\pi(\theta|X_1, X_2)$

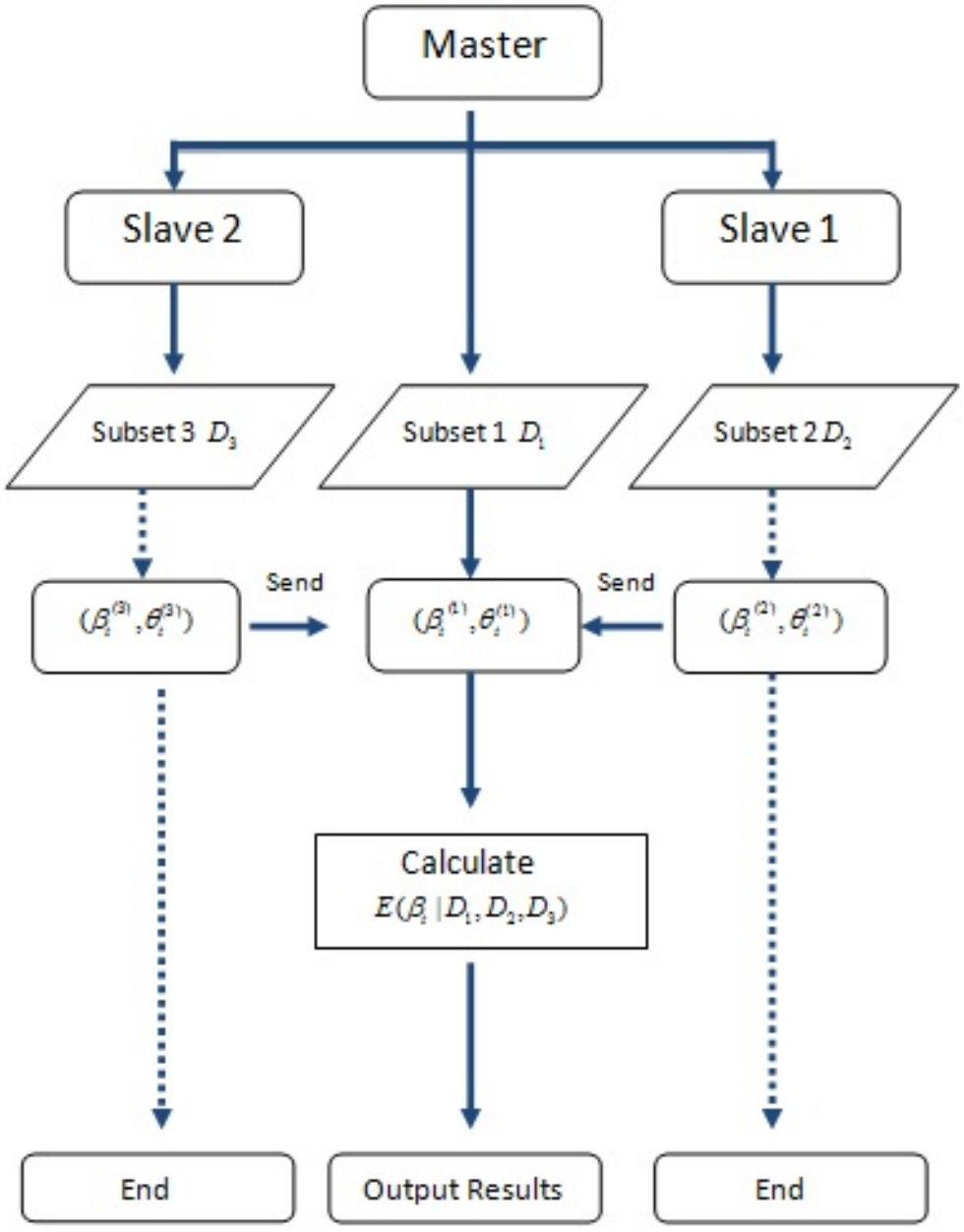


Figure 2.8: Flowchart for parallel computation in Bayesian inference via Divided-and-Resample algorithm

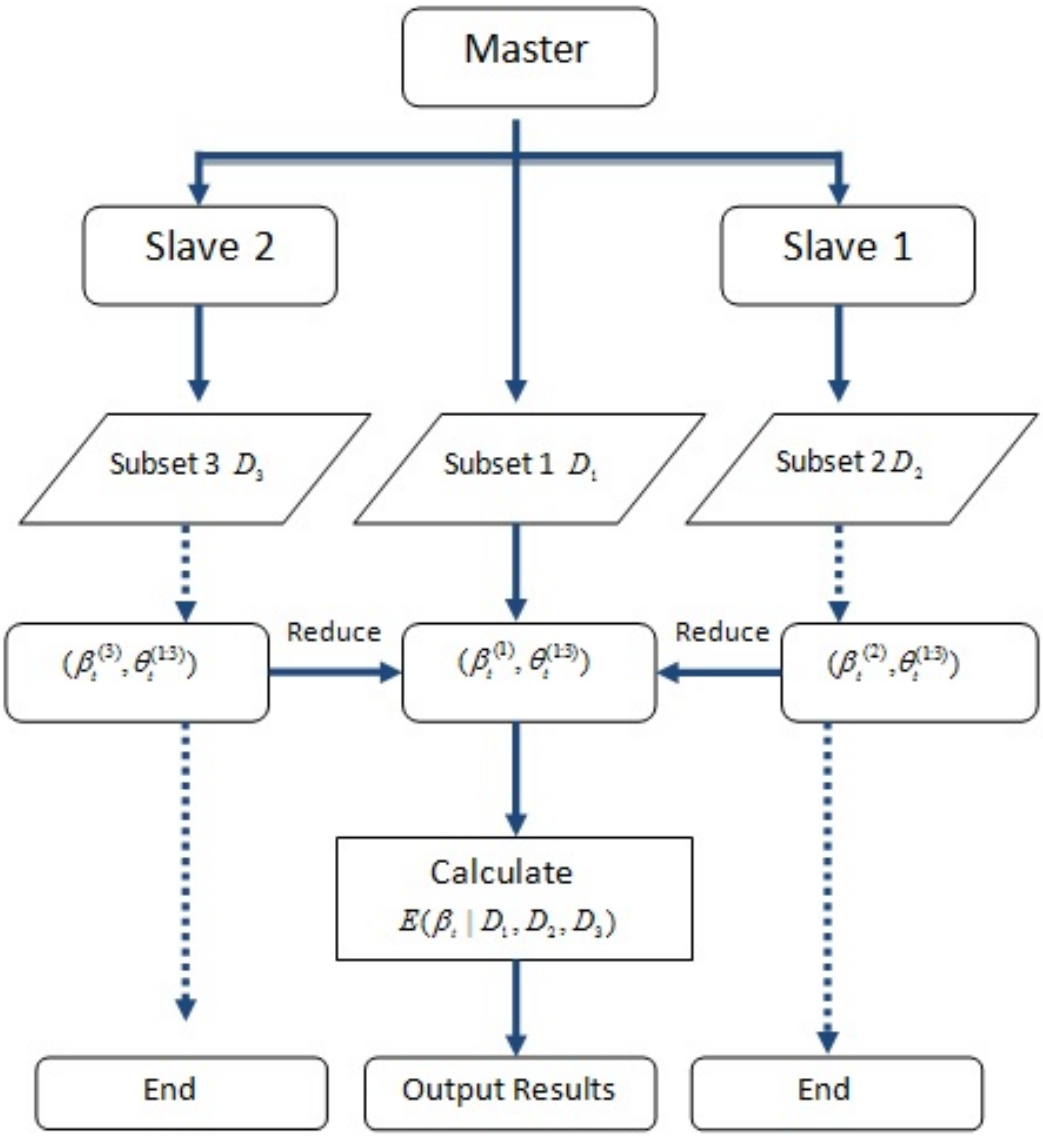


Figure 2.9: Flowchart for parallel computation in weighted combination via SAMC importance sampling algorithm

3. STATIONARY TEST FOR INSAR NOISE

Synthetic Aperture Radar (SAR) Interferometry is a synthesis of the SAR and a interferometry technique. It can provide ground maps for analyzing deformation caused by geophysical processes on earth with centimeter-level accuracy. This maps provides fine resolution and high measurement accuracy with wide spatial coverage. It is capable at delivering a day-and-night image on all-weather conditions as well. InSAR technique has played an important role in past decades for numerous studies, such as volcanic, earthquakes, post-seismic process, glacier flow, pressure change in aquifer. The technique of InSAR analyzing displacement is to compare interferometric processing of two SAR images recorded at different time, i.e. before deformation and after deformation, which is referred as differential SAR interferometry: dInSAR. It can provide vertical ground change information caused by geophysical process.

InSAR has become an important tool to study earthquake and volcanic deformation, however, it is limited by signal loss which are from topographic residuals, atmospheric delay and several error sources. In order to analyze the surface deformation, we have to distinguish other phase components which caused signal loss from InSAR images. It is not easy to distinguish between deformation and atmospheric signals in single interferograms because the error structure is different from image to image and is non-trivial to be estimated and to be included in calculations. Therefore, the InSAR noise has usually been ignored or incorrectly addressed, which leads to biases estimation of model parameters. Some studies have estimated the InSAR noise structure relative with power spectrum; Hanssen (2001) found that on average the InSAR noise has more power at the longer distances with a high power-law index for the spectrum (such as 8/3 and 5/3, spatial scales 0.5-2 km and $>$ 2 km respec-

tively). Other method to estimate InSAR noise is to analyze the non-deforming part of an interferogram and assume the noise structure is the same as in the deforming part; i.e. assuming that the noise has a second order stationary structure. This method has became a usual way to estimate InSAR noise; however, the stationary assumption has never been tested.

The objective of this study is to examine the second order stationary assumption for InSAR noise and to develop a non-stationary model in order to demonstrate the effect of making incorrect assumption on random field.

This Chapter is arranged as follows; Section 3.1 described the InSAR noise and investigated the second order stationary assumption for InSAR noise. Section 3.2 developed the nonstationarity covariance structure with gradually increasing the non-stationarity in the model to investigate the effect of nonstationarity in stationary test. Section 3.3 illustrated the effect of wrong covariance model fitting.

3.1 InSAR Noise

We have two InSAR noise dataset provided by Dr. Sigurjn Jnsson research team in King Abdullah University of Science and Technology shown in Figure 3.1. Each image represents a 129×129 (km) domain. It is not obvious to determine by eyes whether the InSAR noise is stationary or not. To examine the stationarity we used the method proposed by Jun and Genton (2012) shown in section 1.3.

Instead of calculating test statistics based on asymptotic Chi-squared distribution shown in equation (1.9), we can use a subsampling approach to calculate p-values. Subsampling approach has been widely used. In the spatial statistics fields, it can be applied to estimate the spatial covariance model (Hall (1988); Possolo (1991); Sherman and Carlstein (1994); Heagerty and Lumley (2000); Guan et al. (2004)). The fundamental idea of subsampling approach is to divide a target domain field D

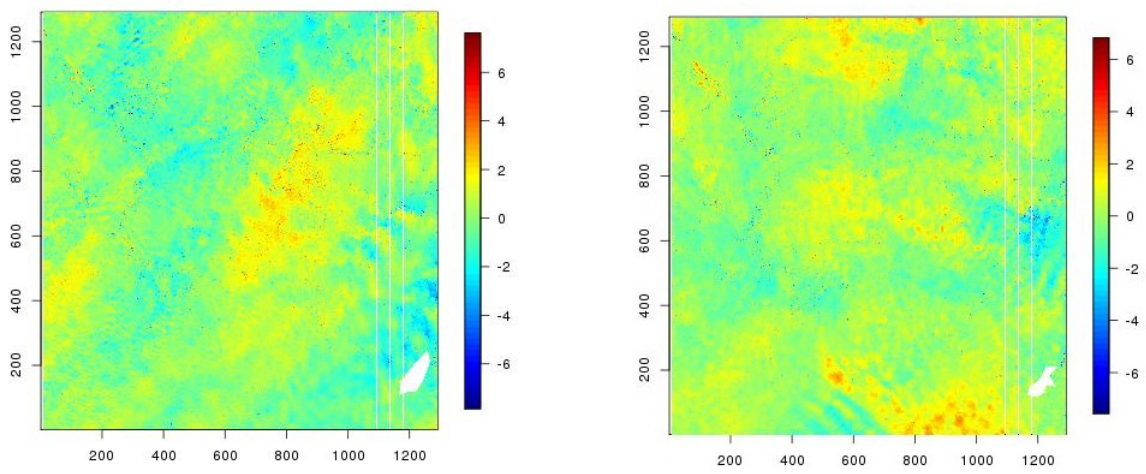


Figure 3.1: Image plot of InSAR noise

into over lapping subblocks. Each small subblock is contained in each split domain D. In practice, the subblock can be set cardinality to be of order $l^2(n)$, where $l(n) = cn^\alpha$ for some $c > 0$ and $\alpha \in (0, 1)$.

We estimate Σ by the subsampling estimator $\hat{\Sigma}$ given by follows,

$$\hat{\Sigma} = \frac{1}{k_n f_n} \sum_{i=1}^{k_n} |D^i_{l(n)}| (\hat{G}^i_{l(n)} - \bar{G}_n)(\hat{G}^i_{l(n)} - \bar{G}_n)^T \tag{3.1}$$

where k_n is total number of subblocks, $\hat{G}^i_{l(n)}$ is a vector of sample variogram calculated on each $D^i_{l(n)}$, $f_n = 1 - \frac{|D_{l(n)}|}{|D_n|}$ and $\bar{G}_n = \frac{1}{k_n} \sum_{i=1}^{k_n} \hat{G}^i_{l(n)}$. On each of these subblocks, we can define the test statistic,

$$T^i_{l(n)} = (X g^i_{l(n)})^T (X \hat{\Sigma} X^T)^{-1} (X g^i_{l(n)}) \tag{3.2}$$

$\hat{\Sigma} \to \Sigma$ and $T^i_{l(n)} \to \chi^2_{rank(X)}$ as $n \to \infty$ by Multivariate Slusky Theorem (Ferguson

(1996)). Following Politis et al. (1999), the p-value of the test statistics can be formed as follows,

$$p - value \approx \frac{1}{k_n} \sum_{i=1}^{k_n} I(T_{l(n)}^i > T) \tag{3.3}$$

T is the stationary test statistics in the whole target domain shown in equation (1.9). Null hypothesis is rejected if the calculated p-value is smaller than significance level α. To test the stationarity of InSAR noise. We consider the following lags,

$\Lambda_1 : h : (0,1),(1,0),(1,1),(0,2),(2,0),(2,2),(0,3),(3,0),(3,3),(4,0),(0,4),(4,4)$

$\Lambda_2 : \{h : (0,13),(13,0),(13,13),(0,15),(15,0),(15,15),(0,17),(17,0),(17,17),$
$(0,20),(20,0),(20,20)\}$

$\Lambda_3 : h : \Lambda_1, \Lambda_2, (8,8),(0,8),(8,0)$

$\Lambda_4 : h : (0,1),(0,2),(0,3),(0,4)$

$\Lambda_5 : h : (0,13),(0,15),(0,17),(0,20)$

$\Lambda_6 : h : \Lambda_4, \Lambda_5, (0,8)$

$\Lambda_7 : h : (1,0),(2,0),(3,0),(4,0)$

$\Lambda_8 : h : (13,0),(15,0),(17,0),(20,0)$

$\Lambda_9 : h : \Lambda_7, \Lambda_8, (8,0)$

Table 3.1 reports the p-values from the subsampling approach for two spatial images, Data 1 and Data 2. By splitting procedure for the stationary test into two images, we tried a horizontal split and vertical split. We also tried various subblock size of $l(n) = cn^\alpha$ for $\alpha = 0.5$ and c at 0.5 and 1 in order to get p-value through subsampling as well as to estimate covariance structure Σ, which correspond to subblocks of size 18×18 and 36×36 grids. The result shows that all the p-value are below 5% and it

rejects the null hypothesis of weak stationary. Therefore, the test statistic tells us that it is not second-order stationary, and it is not suitable to assume InSAR noise from non-deforming part is the same as deforming part.

Table 3.1: Stationary test for two InSAR noise

			Data 1 Vertical	Data 2 Vertical	Data 1 Horizontal	Data 2 Horizontal
Lag Size	Subblock	Lag	P-value	P-value	P-value	P-value
30 Pixels	18	Λ_1	0.0000	0.0159	0.0000	0.0000
		Λ_2	0.0000	0.0000	0.0000	0.0000
		Λ_3	0.0000	0.0000	0.0000	0.0000
		Λ_4	0.0000	0.0190	0.0000	0.0000
		Λ_5	0.0000	0.0004	0.0000	0.0000
		Λ_6	0.0000	0.0199	0.0000	0.0000
		Λ_7	0.0000	0.0028	0.0000	0.0000
		Λ_9	0.0000	0.0000	0.0000	0.0000
30 Pixels	36	Λ_1	0.0000	0.0000	0.0000	0.0000
		Λ_2	0.0000	0.0000	0.0000	0.0000
		Λ_3	0.0000	0.0000	0.0000	0.0000
		Λ_4	0.0000	0.0104	0.0000	0.0000
		Λ_5	0.0000	0.0000	0.0000	0.0000
		Λ_6	0.0000	0.0009	0.0000	0.0000
		Λ_7	0.0000	0.0000	0.0000	0.0000
		Λ_9	0.0000	0.0000	0.0000	0.0000

3.2 Nonstationary Covariance Structure

A stochastic random field $Z = z(x), x \in R$ is usually defined as,

$$z(x) = \mu(x) + \sigma(x)\varepsilon(x)$$

where $\mu(x) = E(Z(x))$, $\sigma^2(x) = Var(Z(x))$ and $\varepsilon(x)$ is a weakly stationary process with mean zero and unit variance. To deal with a nonstationary stochastic process

$Z = z(x), x \in R$, we can be deformed through a sequence of mean $E(Z(x))$ and second order moment $Var(Z(x))$.

Many literature has plenty of examples of addressing nonstationary spatial covariance structure. However, they do not provide a general model for nonstationary covariance model. Obled and Creutin (1986) provide a general approach called empirical orthogonal function to model nonstationary space-time process. Sampson and Guttorp (1992) present a new framework for generating nonstationary covariance model by directly modeling $\varepsilon(x) = \gamma(f(x))$, where γ is a weakly stationary process and f represents a smooth continuous bijective function. It is equivalent to deal with a nonstationary variogram model $\gamma(||f(x) - f(y)||)$, where $||f(x) - f(y)||$ denotes Euclidean distance between site locations in a bijective transformation of geographic coordinates system x and y, and γ is a isotropic and stationary variogram model.

There are some literature discussing the property of deformation function f. Sampson and Guttorp (1992) develop the nonparametric approach for estimating deformation function f by defining f as a thin-plate spline. Iovleff and Perrin (2004) estimate the space deformation using a simulated annealing algorithm. Those methodologies deal with deformation function by using the nonparametric approaches. Perrin and Monestiez (1999) propose a parametric family of bijective functions, called the Radial Basis Deformation function, which is analogy to the Radial Basis Function (RBF) (Powell (1987)). It has an advantage that the bijective condition is ensured for parametric family. In this study, we will introduce Radial Basis Deformation function and to use this approach conducting a nonstationary random field.

3.2.1 Definition of Radial Basis Deformation

We introduce the parametric family of radial basis deformations from Geography-plane $x \in R^2$ into the deformation-plane $f(x) \in R^2$ defined as follows,

$$f(x) = c + (x - c)\Phi(||x - c||), \forall f \in B \tag{3.4}$$

where c is the coordinates in our target domain, which is the center of the deformation and $\Phi(.)$ is a Radial function form R^+ to R. Powell (1987) proved that any function that satisfies the property $\Phi(x) = \Phi(||x||)$ is a radial function. Perrin and Monestiez (1999) introduce the choice of parametric radial function form Φ as follows:

$$\text{exponential} : \Phi(\rho) = 1 + b\exp(-a\rho) \tag{3.5}$$

$$\text{Gaussian} : \Phi(\rho) = 1 + b\exp(-a\rho^2) \tag{3.6}$$

where $\rho = ||x - c||$ and $a > 0$

Each deformation function is a warping from the geography-plane into the deformation-plane, which is saying that a pair of two-dimensional functions that maps a position x in the G-plane to position y in the D-plane.

$$y = f(x) = (f(x_1), f(x_2))$$

Therefore, a nonstationary covariance structure can be formed as follows:

$$D(x_1, x_2) = \gamma_\beta(||f(x_1) - f(x_2)||) \tag{3.7}$$

where $||f(x_1) - f(x_2)||$ denotes the Euclidean distance between x_1 and x_2 in D-plane.

f represent a smooth bijective radial basis deformation function, and γ represents a stationary and isotropic variogram function with parameter β.

3.2.2 Definition of Parametric Family in Radial Basis Function

We work on Gaussian radial basis deformation as a start which can be defined as follows:

$$f(x) = c + (x - c)(1 + b\exp(-a||x - c||^2)) \tag{3.8}$$

where c is two dimension coordinate system and represents the center of deformation, a is a range parameter and b is the intensity of stretching (b >0) or shrinking (b <0). $b = 0$ indicates no deformation, i.e. $f(x) = x$.

Perrin and Monestiez (1999) proved that Gaussian Radial Deformation $f(x)$ will be an bijective function with parameters $a > 0$, $c \in R$ and $b \in (-1, \frac{1}{2}\exp(\frac{3}{2}))$ (See Appendix A). We now illustrate the behavior of parameters for Gaussian deformation in one dimensional case shown in Figure (3.2). We use a sequence of x between -1 and 1 and γ_β is set to be an exponential variogram. When parameter $b = 0$, the Gaussian deformation function $f(x) = x$, which has no deformation and indicates exponential variogram in our case. In Figure (3.2), we fixed parameter b=2.24 and c=(0,0) to show the behavior of deformation parameter a. It is more curvier from center (0,0) when deformation parameter a is getting smaller. There is only a slight difference when we increase the parameter a from 1 to 10. It is obvious that parameter a controls the range of deformation. When deformation parameter a gets small, it results in larger deformation.

Figure (3.3) shows the case when we fixed range parameter $a = 1$ and chose various values of parameter b in the range between -1 and 2.24. $b \in (-1, \frac{1}{2}\exp(\frac{3}{2})) = b \in (-1, 2.24)$. It shows that $f(x)$ stretched upward on positive coordinates side when $b > 0$ and shrink down when $b < 0$.

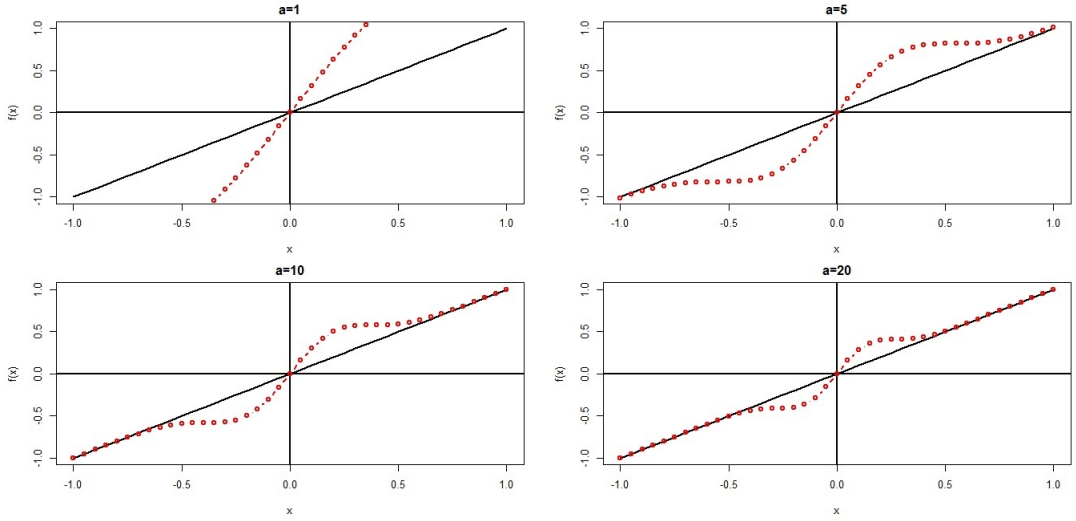


Figure 3.2: RBF deformation parameter a in one dimension system

Now we illustrate the pattern of parameter changed in two dimension coordinate system shown in Figure (3.4). We chose a target domain $Z(s) \in (0,1) \times (0,1)$ and deformation from center c=(0.5,0.5) to investigate the behavior of change in deformation parameter a and b. We use γ_β to be an exponential variogram in this case as well. In Figure (3.4), the upper three plots represent deformation parameter $b = -0.9$ with three different values of deformation parameter a, 0.1, 1, 10 respectively. It shrinks in the center much more on $a = 0.1$ than $a = 1$ or $a = 10$. The lower three plots in Figure (3.4) represent deformation parameter $b = 2.24$, it diverges from center point $c = (0.5, 0.5)$ on $a = 0.1$ than $a = 1$ or $a = 10$ as well. However, it diverges too much when $a = 0.1$ which results in no deformation at all on the D-plane.

In next section, we demonstrate the behavior of deformation parameter by evaluated the power of stationary test in a simulation example.

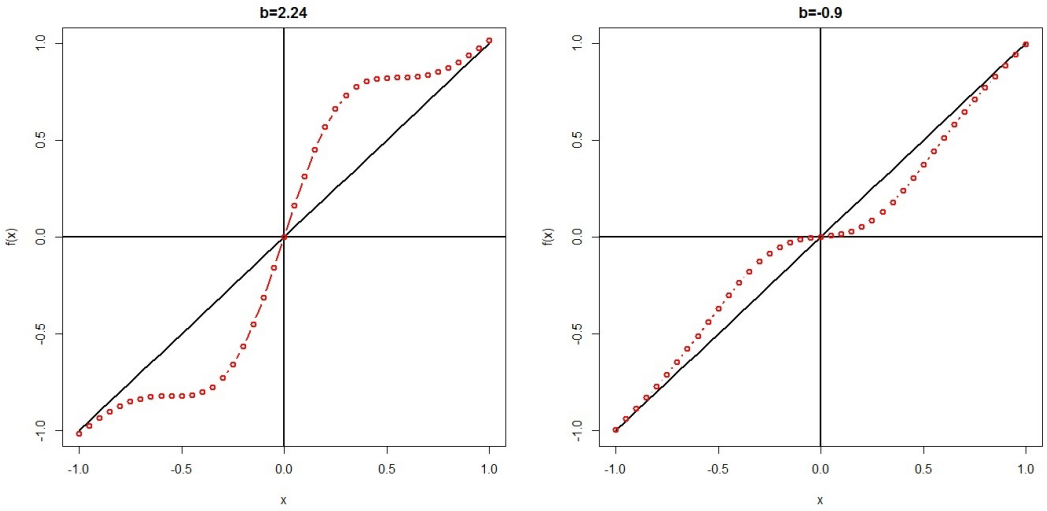


Figure 3.3: RBF deformation parameter b in one dimension system

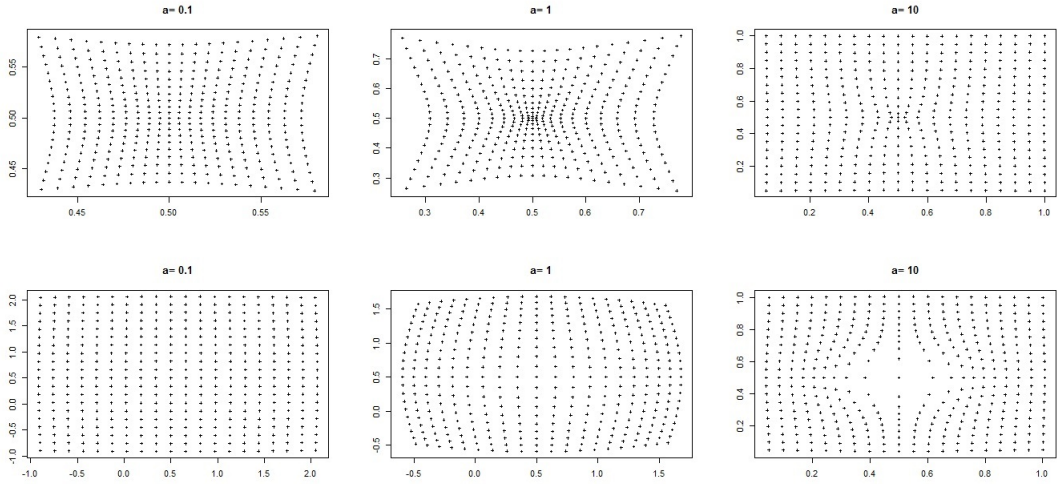


Figure 3.4: RBF deformation parameter in two dimension coordinate system

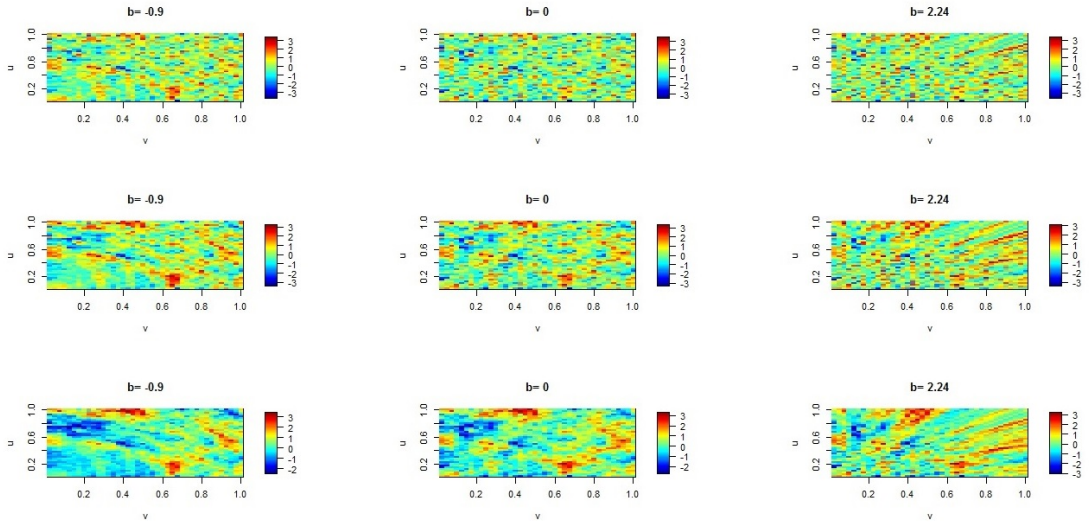
3.3 Simulation Study

Our goal is to simulate random fields from non-stationary covariance structure and by gradually increasing nonstationarity in the model to see the effect of non-stationarity in the stationary test. We consider a mean zero weakly stationary (isotropic) Gaussian random field $Z(s)$, $s \in D$ and use Gaussian radial deformation function to generate D-plane, to test the stationarity and calculate the power of stationary test. The covariance structure for assessing the test statistics power is an exponential variogram model as follows:

$$\gamma(h; \alpha, \beta) = -\alpha \exp(\frac{-h}{\beta}) \tag{3.9}$$

We test the case when $\alpha = 1$ and $\beta = 0.0167, 0.0333, 0.0833$, where variogram parameter β represents different strength of dependence (weak to strong). We generate 5000 realizations for each combination of (α, β) in the domain of $(0, 1) \times (0, 1)$ with 20x20 or 40x40 square grids. The Gaussian Radial basis deformation is performed here and to compute the Euclidean distance in D-plane. Figure 3.5 represent the image plot for D-plane in three combination of exponential variogram parameters and three combination of deformation parameters. It is non-trivial to observe by eye that which one has non-stationary covariance structure.

We split the domain into two 10x20 or 20x40 neighboring squares, compute empirical estimator $\hat{A}_Z(h) = \frac{1}{N(h)} \sum_{s_i, s_j} Z(s_i)Z(s_j)$ in rectangular domains and use



Note: the upper three plots represent low spatial dependency ($\beta = 0.0167$) with fixed deformation parameter $a = 1$ and various deformation parameter b; the middle three plots represent the median spatial dependency ($\beta = 0.0333$); and the lower three plots represent the high spatial dependency ($\beta = 0.0833$).

Figure 3.5: Image plot of random field with various RBF deformation parameters

following lags set:

$$\Lambda_1 : h : (1,0), (0,1)$$

$$\Lambda_2 : h : (1,0), (0,1), (1,1), (-1,1)$$

$$\Lambda_3 : h : (1,0), (0,1), (1,1), (-1,1), (2,1), (1,2), (-2,1), (-1,2)$$

We try a horizontal split for spatial domain and use subsampling approach to calculate p-values. Table 3.2 reports the power test from subsampling approach in 20x20 and 40x40 spatial domain. The empirical power is quite small when deformation parameter $a = 0.1$ and $b = 2.24$. It has been shown in Figure (3.4) which indicates nearly no deformation. The empirical power values does not change much through deformation parameter b when deformation parameter $a = 10$. The de-

formation seems to be too concentrated around the center, which does not reflect on whole target domain. When we have extremely deformation parameter b, say $b = 2.24$ or $b = -0.9$ with deformation parameter $a = 1$, we get 90 percent power value.

We continue to investigate the effect of various subblocks and spatial lags with fixed deformation parameter $a = 1$. We use criterion $l(n) = cn^{\alpha}$ for $\alpha = 0.5$ and c at 0.5, 1 and 1.5 respectively, which corresponds to number subblock 2,4 and 7. The results are shown in Table 3.3 and Table 3.4. The power values are significantly bigger in 40x40 domain, 90% of statistical power when deformation parameter b is close to 2.24.

3.4 Effect of Wrong Fitting

In the previous section, we analyzed the structure of InSAR noise in examples of deformation interferograms and tested the stationary assumption. It proved that stationary assumption does not hold for InSAR noise. If we try to estimate deformation process with wrong covariance structure assumption, it will lead to a non-accurate and biased estimation. We now turn our attention to investigate the effect of wrong covariance fitting.

The simple Mogi point source model has been playing an important role to explain observed inflation and deflation on volcanoes process for decades. The main idea is based on an elastically expanded point source in a half space. The geologists use the surface deformation results from the InSAR data and the Mogi source model (Mogi (1958)), and invert the ground deformations to find the locations and intensities of several deformation sources. It can describe the observed line-of-sight (LOS) data at

Table 3.2: Statistical power for stationary test with various deformation parameter

		20x20			40x40		
beta	b	a=0.1	a=1	a=10	a=0.1	a=1	a=10
	-0.9	29.62	30.98	4.60	97.82	27.96	4.80
	-0.7	10.62	15.62	4.54	67.90	13.30	4.62
	-0.4	5.16	6.46	4.56	12.82	5.88	4.52
	-0.1	4.46	4.58	4.50	4.86	4.66	4.52
0.0167	0	4.48	4.48	4.40	4.56	4.56	4.56
	0.5	4.44	4.78	4.34	5.14	6.10	4.62
	1.1	4.40	5.80	4.58	5.12	15.34	4.72
	1.6	4.44	7.02	4.68	4.92	38.70	4.98
	2.1	4.46	8.72	5.22	4.74	73.40	5.32
	2.24	4.46	9.92	5.54	4.68	80.22	5.46
	-0.9	36.80	41.60	5.76	96.34	40.82	6.52
	-0.7	17.78	26.20	5.54	68.04	26.64	6.54
	-0.4	8.90	10.98	5.54	22.08	12.30	6.40
	-0.1	6.08	5.96	5.68	7.30	6.82	6.22
0.0333	0	5.66	5.66	5.66	6.28	6.28	6.28
	0.5	4.74	9.40	5.72	10.90	15.02	6.18
	1.1	4.28	18.68	5.98	14.64	52.82	6.20
	1.6	4.40	26.54	6.70	13.36	87.34	6.46
	2.1	4.42	32.46	7.62	11.04	97.88	7.22
	2.24	4.40	34.06	7.82	10.54	98.70	7.56
	-0.9	53.34	43.72	14.50	87.86	43.46	17.64
	-0.7	36.16	30.60	14.34	56.96	32.60	17.60
	-0.4	22.80	20.12	14.22	28.94	21.68	17.28
	-0.1	16.18	15.18	14.42	18.74	17.36	17.12
0.0833	0	14.50	14.50	14.50	17.16	17.16	17.16
	0.5	9.82	16.34	15.02	18.14	23.94	17.18
	1.1	7.20	27.84	15.60	29.48	57.66	17.48
	1.6	6.38	41.14	16.28	39.50	88.34	17.94
	2.1	5.68	54.28	17.40	47.04	97.38	18.64
	2.24	5.60	57.56	17.84	48.42	98.18	18.76

Table 3.3: Statistical power for stationary test in 20× 20 domain with various subblocks

		0.0167			0.0333			0.0833		
subblock	b	Lag 2	Lag 4	Lag 8	Lag 2	Lag 4	Lag 8	Lag 2	Lag 4	Lag 8
	-0.9	30.98	22.06	24.62	41.60	32.82	38.56	43.72	39.74	51.12
	-0.7	15.62	10.94	15.20	26.20	19.34	25.46	30.60	27.40	38.70
	-0.4	6.46	5.18	9.84	10.98	8.76	14.94	20.12	16.08	27.42
	-0.1	4.58	4.04	8.86	5.96	4.58	11.32	15.18	12.38	23.00
2	0	4.48	3.92	8.74	5.66	4.42	11.12	14.50	11.64	22.10
	0.5	4.78	4.24	9.46	9.40	7.18	14.66	16.34	13.48	24.26
	1.1	5.80	5.30	10.92	18.68	17.56	24.92	27.84	27.78	38.18
	1.6	7.02	8.76	15.28	26.54	35.06	41.04	41.14	48.88	55.08
	2.1	8.72	21.38	29.26	32.46	55.46	60.80	54.28	70.42	73.38
	2.24	9.92	26.24	34.40	34.06	60.56	64.26	57.56	74.52	76.46
	-0.9	31.52	31.96	71.08	39.04	39.08	77.22	16.38	38.60	79.70
	-0.7	19.74	21.34	62.86	26.36	27.88	70.12	9.14	27.20	71.74
	-0.4	10.32	14.06	55.54	13.58	16.78	61.42	4.54	17.82	63.04
	-0.1	7.80	11.96	53.38	8.76	12.62	55.48	3.20	14.08	58.48
4	0	7.96	12.02	53.26	8.40	12.66	55.14	3.02	13.76	58.10
	0.5	9.06	12.62	53.24	12.96	17.12	59.16	4.66	18.60	63.68
	1.1	10.28	14.92	55.78	23.60	30.88	70.70	11.26	36.36	77.50
	1.6	11.40	20.00	61.30	31.00	46.76	80.96	20.48	57.58	87.80
	2.1	12.66	32.38	74.52	35.30	62.06	89.90	30.20	73.32	93.62
	2.24	13.70	35.94	77.60	36.80	64.50	91.34	32.74	76.42	94.54

Table 3.4: Statistical power for stationary test in 40× 40 domain with various sub-blocks

		0.0167			0.0333			0.0833		
subblock	b	Lag 2	Lag 4	Lag 8	Lag 2	Lag 4	Lag 8	Lag 2	Lag 4	Lag 8
	-0.9	27.96	19.46	11.78	40.82	31.24	21.34	43.46	35.80	31.48
	-0.7	13.30	9.94	6.82	26.64	19.22	13.2	32.60	25.58	23.42
	-0.4	5.88	5.18	4.14	12.30	8.92	6.52	21.68	17.20	16.62
	-0.1	4.66	4.32	3.78	6.82	5.32	4.76	17.36	14.50	14.28
2	0	4.56	4.36	3.82	6.28	5.22	4.76	17.16	14.22	14.18
	0.5	6.10	5.18	4.16	15.02	11.06	8.10	23.94	18.74	18.36
	1.1	15.34	11.52	7.70	52.82	40.20	29.08	57.66	46.78	42.84
	1.6	38.70	28.68	18.90	87.34	78.32	68.84	88.34	80.94	76.92
	2.1	73.40	63.56	53.26	97.88	96.56	95.60	97.38	95.74	95.96
	2.24	80.22	72.90	66.00	98.70	97.60	97.68	98.18	96.80	97.54
	-0.9	23.32	17.80	15.12	33.20	25.10	21.36	33.42	25.38	22.80
	-0.7	13.42	10.64	10.72	21.92	17.72	15.48	22.56	16.92	16.30
	-0.4	6.62	6.34	7.90	10.94	9.14	9.98	12.02	9.16	10.18
	-0.1	5.20	5.38	7.08	6.18	6.10	7.84	8.24	6.64	7.72
4	0	5.10	5.36	7.06	6.02	5.80	7.80	8.06	6.60	7.58
	0.5	6.98	6.52	7.66	14.26	11.38	11.84	15.32	11.70	11.40
	1.1	15.98	13.08	12.50	51.60	41.58	35.42	53.62	42.22	36.56
	1.6	38.68	30.02	25.54	86.24	77.68	71.12	85.56	77.22	69.62
	2.1	70.44	60.84	55.34	97.58	95.64	93.90	96.34	93.36	91.38
	2.24	76.88	69.20	65.62	98.28	97.02	96.22	97.24	94.88	93.74
	-0.9	24.44	23.74	35.84	31.82	30.34	42.48	34.06	32.00	44.40
	-0.7	15.56	17.06	29.26	23.00	23.68	36.26	23.92	23.06	35.00
	-0.4	8.66	11.86	24.52	13.06	15.10	27.80	12.68	13.94	25.54
	-0.1	7.50	10.98	23.68	8.06	11.18	24.26	8.00	9.52	21.38
7	0	7.42	11.02	23.66	7.78	10.78	24.10	7.70	9.10	21.12
	0.5	8.92	12.18	24.70	17.00	17.80	30.88	17.26	17.08	29.56
	1.1	18.98	19.32	32.40	53.30	48.64	59.72	56.36	51.86	60.08
	1.6	40.28	37.20	48.06	85.06	81.06	84.94	85.34	81.58	84.86
	2.1	69.86	64.10	73.58	96.76	94.80	96.74	95.34	93.42	95.38
	2.24	74.76	69.96	79.40	97.84	96.08	97.86	96.26	94.52	96.42

location (x,y) as,

$$d_o(x, y) = i^T u + \epsilon \tag{3.10}$$

where i is a unit vector pointing towards the satellite, u is the 3-dimensional ground displacement predicted at (x, y) by a Mogi source with model parameters (volume change ΔV, the depth d, and the source location (x', y') and ϵ is InSAR noises. The objective of this study is to estimate the volume change ΔV of displacement. To evaluate the effect of making wrong stationary assumption for InSAR noise, we simulate with three different type of covariance structures; uncorrelated, stationary and non-stationary.

We simplify the equation (3.10) to a linear regression model with true parameter of volume change $b_{vc} = 0.005$.

$$Y = Xb_{vc} + \epsilon$$

Based on the information of displacement parameter ΔV equal to 0.005 km^{3m}, depth is 5.75 , deformation location $(x', y') = (0.3, 0.7)$, we can generate the corresponding X and Y. We simulate 1000 realization for three types of InSAR noise: identical independent noise, stationary covariance noise and nonstationary covariance noise in a target domain 32x32 pixel grid, where a pixel spacing 1 km, and $(x, y) \in (0, 1) \times (0, 1)$. The InSAR noise $\epsilon \sim N(0, \sigma^2 I)$ for identical independent noise; $\epsilon \sim N(0, \Sigma_\theta)$ for exponential variogram with parameter $\theta = (\alpha, \beta) = (\sigma^2, 0.0167)$, and $\epsilon \sim N(0, \Sigma_{\theta,\gamma})$ with Gaussian Radial deformation with exponential variogram parameter $\theta = (\alpha, \beta) = (\sigma^2, 0.0167)$ and deformation parameter $\gamma = (a, b, c)$. We compare four different scale of variances, standard error from 0.00325 to 0.052. The

correct model fitting results are shown in Figure (3.6).

In this examination, we perform three types of scenario; fitting with i.i.d covariance structure through ordinary least square method, fitting with exponential variogram covariance structure and fitting with nonstationary variogram model through generalize least square method. We first consider estimating volume change ΔV by ordinary least square. The equation is defined as follows,

$$b_{vc} = (X^T X)^{-1} X^T Y$$

Figure (3.7) shows the ordinary least square result for three different noise covariance structures in 1000 simulations. There is no surprise that independent noise covariance structure has smaller variation of volume change estimation than correlated noise covariance structure when we estimated by OLS.

We continue to estimate the volume change with generalize least square. The general equation for GLS is defined as follows,

$$b_{vc} = (X^T \Sigma X)^{-1} X^T \Sigma^{-1} Y$$

To estimate the volume change b_{vc} with three different noise structures by using generalize least square, we have to estimate covariance matrix Σ first. We used Newton-type algorithm to estimate the likelihood function from Gaussian random field with mean zero and stationary covariance Σ_θ and nonstationary covariance $\Sigma_{\theta,\gamma}$. Then re-write the estimating equation for b_{vc} as follows,

$$\hat{b_{vc}} = (X^T \hat{\Sigma}_\theta X)^{-1} X^T \hat{\Sigma}_\theta^{-1} Y$$
$$\hat{b_{vc}} = (X^T \hat{\Sigma}_{\theta,\gamma} X)^{-1} X^T \hat{\Sigma}_{\theta,\gamma}^{-1} Y$$

Table 3.5: Estimation of exponential variogram for three types of InSAR noise

Sigma		iid	stationary	nonstationary
0.00325	mean of alpha	0.0000106	0.0000105	0.0000104
	mean of beta	0.0044857	0.0165676	0.0250997
0.0065	mean of alpha	0.0000421	0.0000421	0.0000417
	mean of beta	0.0045062	0.0165676	0.0250997
0.013	mean of alpha	0.0001691	0.0001686	0.0001666
	mean of beta	0.0044532	0.0165676	0.0250997
0.052	mean of alpha	0.0027050	0.0115928	0.0026657
	mean of beta	0.0043897	0.0165679	0.0250997

Figure (3.8) represents the generalize least square results with fitting exponential variogram $\hat{\Sigma}_\theta$ for three different types of InSAR noise in 1000 simulations. The stationary noise and independent noise both have smaller variation of volume change estimation than nonstationary noise. Generalize least square estimation corrected the accuracy for stationary noise. However, the b_{vc} estimation has larger variation when true model is the nonstationary covariance structure but we fitted with the stationary covariance. Table 3.5 shows the exponential covariance parameters (α, β) estimation results for three InSAR noises. The true exponential variogram parameter β is 0.0167 but nonstationary model will result in 1.5 times more of dependence. The results suggests that we will increase the strength of dependence when we fit stationary model to a non-stationary covariance structure.

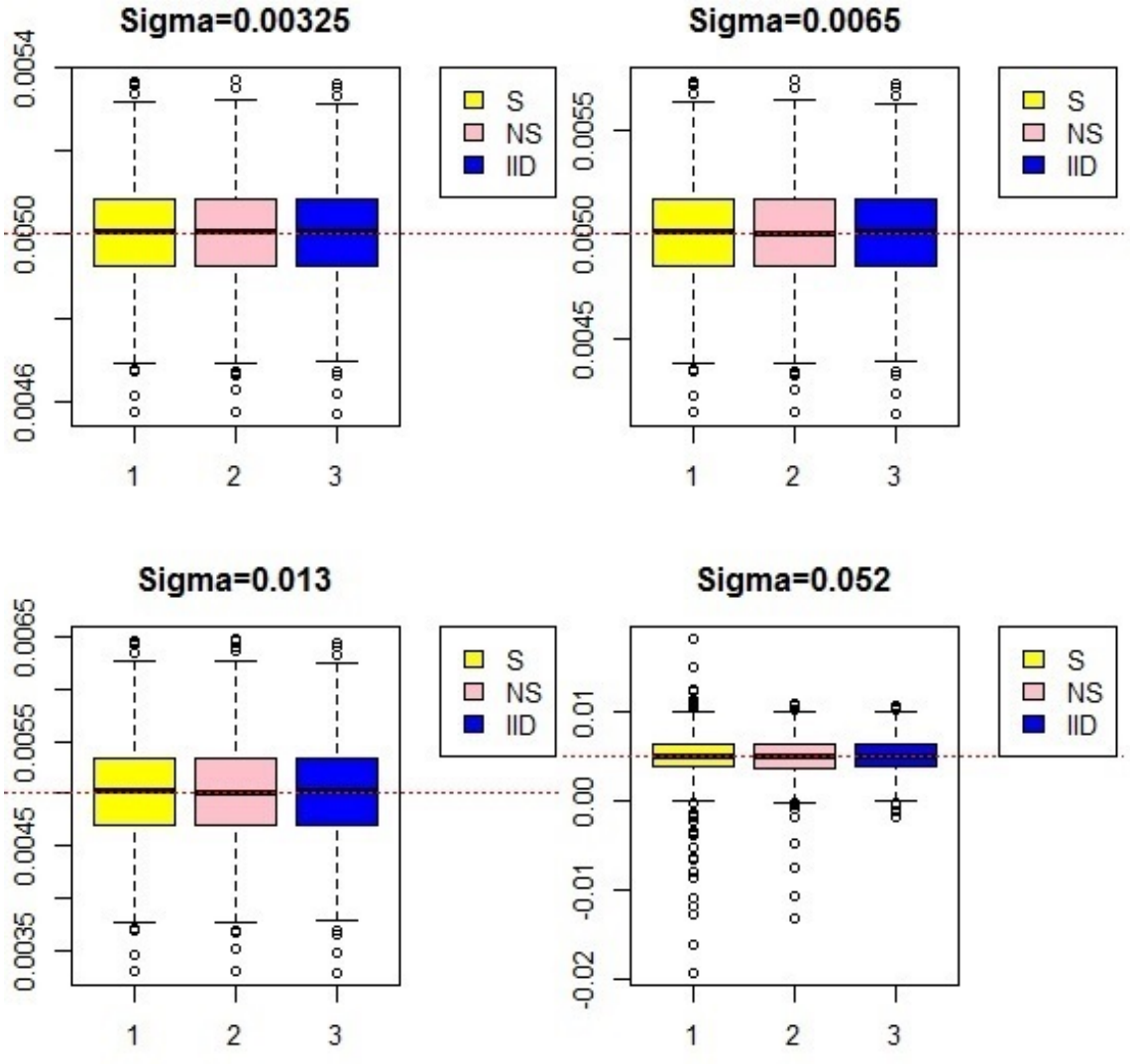
Figure (3.9) represents the generalize least square result with nonstationary covariance function $\hat{\Sigma}_{\theta,\gamma}$ for three different noise covariance structure in 1000 simulations. It is obvious that nonstationary covariance structure has smaller variation than stationary and i.i.d covariance structure when we estimate with nonstationary covariance model.

Table 3.6 shows more statistics details for three covariance model estimation.

Table 3.6: Statistics index of volume change estimation in three types of InSAR noise

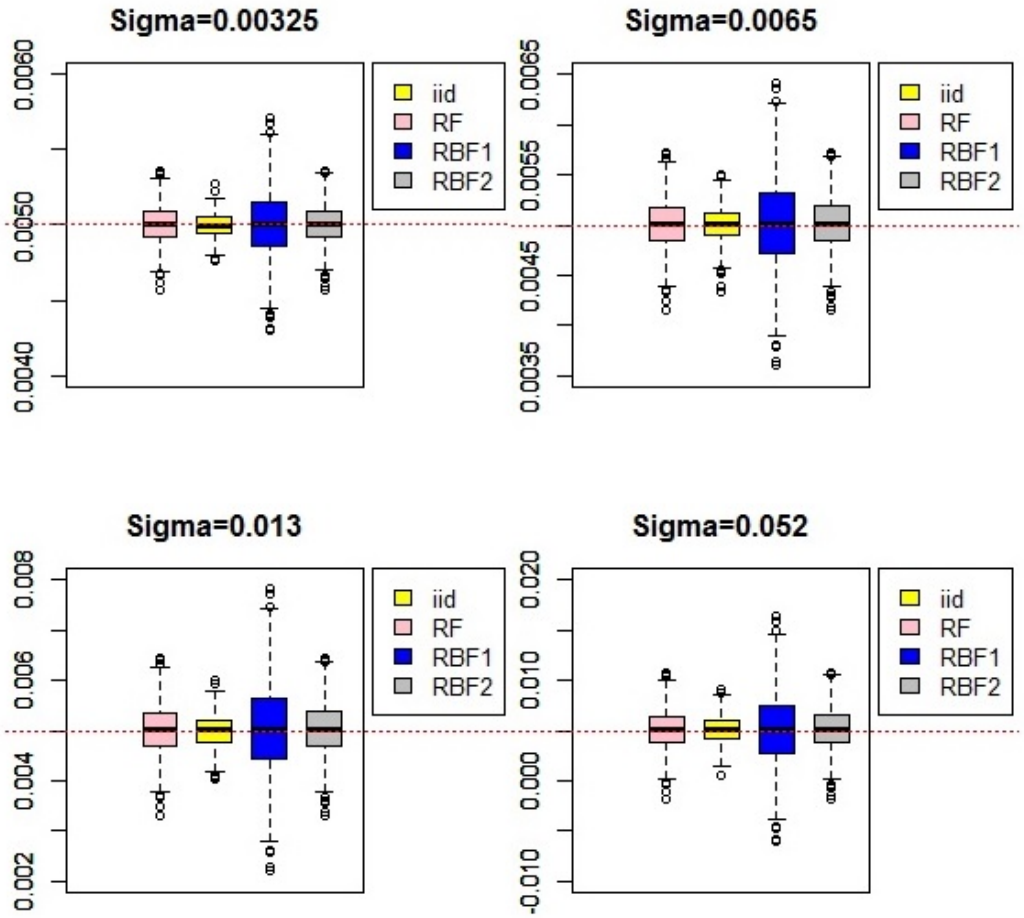
	Mean	Median	SD	MSE ($\times 10^3$ %)	95% Percentile
NS	0.005010	0.005005	0.000157	0.0025	0.004703 0.005318
S	0.005007	0.005000	0.000219	0.0048	0.004578 0.005436
IID	0.005007	0.005007	0.000217	0.0047	0.004581 0.005433

Even though stationary and i.i.d covariance structure can be estimated as good as non-stationary model for volume change; however, it produces a much wider confidence interval and mean square error is 1.9 times more than correct model.



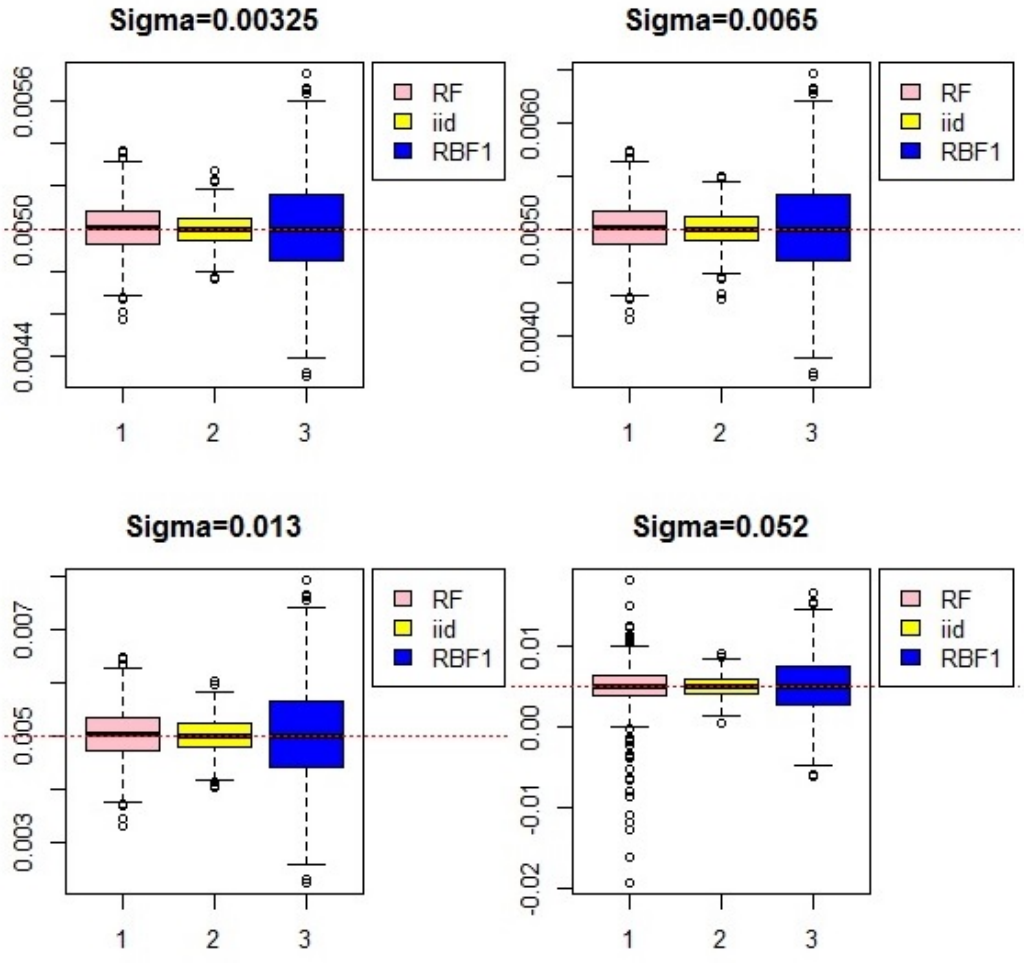
Note: part a)-d) indicates difference standard error of noise. S represents stationary noise estimating with stationary covariance model; NS represent nonstationary noise estimating with non-stationary covariance model and iid indicates stationary noise estimating with uncorrelated covariance model.

Figure 3.6: Boxplot of correct estimation for Volume Change



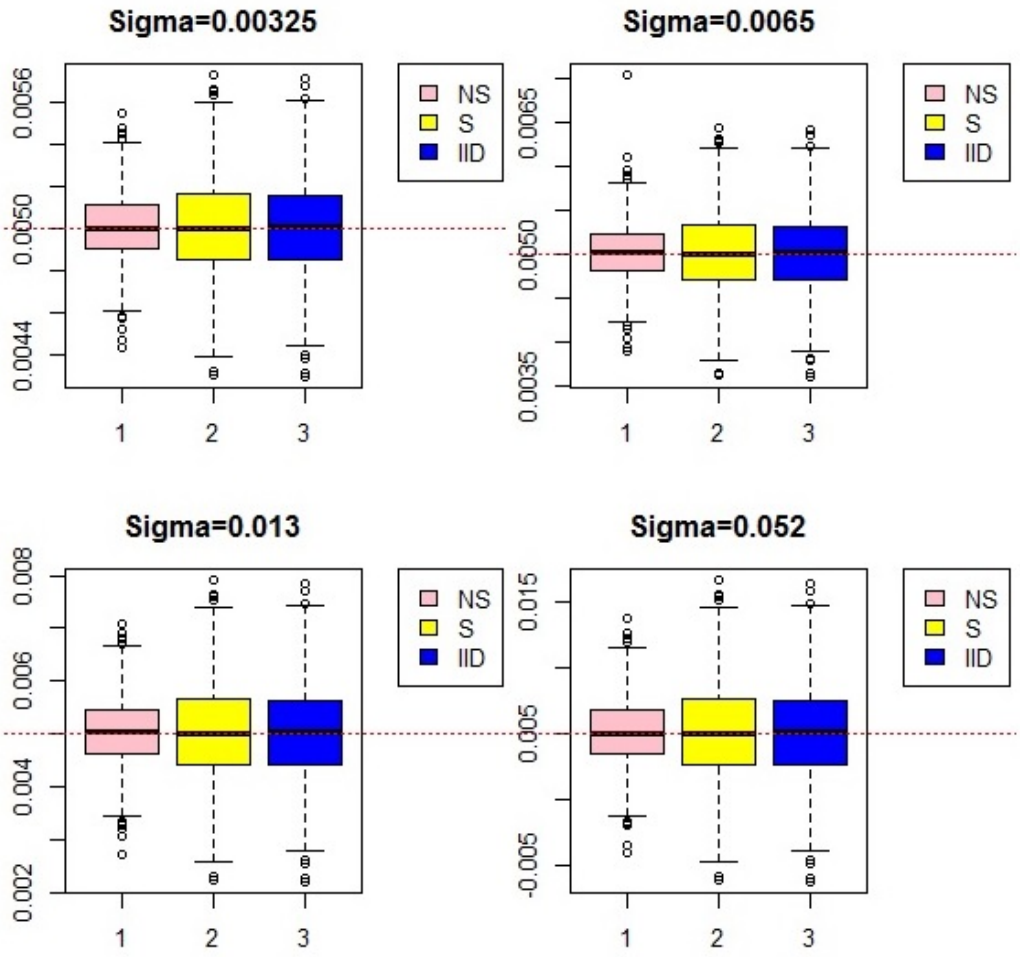
Note: a. For i.i.d random variable we used normal distribution with mean zero and different sigma values. b. For Gaussian random field we used exponential with parameter beta=0.0167 and alpha (represents the variance) exactly the same with i.i.d setting. c. For Radial Basis Deformation we based on exponential variogram with alpha exactly the same with i.i.d setting and beta=0.0167. The deformation parameter for RBF1, we used a=1, b=-0.9 and c=(0.05,0.05). The deformation parameter for RBF2, we used a=1, b=2.24 and c=(0.05,0.05)

Figure 3.7: Boxplot of OLS estimation



Note: part a)-d) indicate difference sigma for noise setting. For i.i.d random variable we used normal distribution with mean zero and constance variance. For Gaussian random field we used exponential with parameter beta=0.0167 and alpha (represents the variance) exactly the same with i.i.d setting. For Radial Basis Deformation we based on exponential variogram with alpha exactly the same with i.i.d setting and beta=0.0167. The deformation parameter for RBF1, we used a=1, b=2.24 and c=(0.05,0.05).

Figure 3.8: Boxplot of GLS estimation with fitting stationary covariance structure



Note: part a)-d) indicate difference sigma for noise setting. For i.i.d random variable we used normal distribution with mean zero and constance sigma values. For Gaussian random field we used exponential with parameter beta=0.0167 and alpha (represents the variance) exactly the same with i.i.d setting. For Radial Basis Deformation we based on exponential variogram with alpha exactly the same with i.i.d setting and beta=0.0167. The deformation parameter for RBF1, we used a=1, b=2.24 and c=(0.05,0.05)

Figure 3.9: Boxplot of GLS estimation with fitting nonstationary covariance structure

4. SUMMARY AND CONCLUSIONS

In this dissertation we work on two parts. In the first part, we propose a new algorithm for big data analytic in Bayesian fields; parallel computation for Stochastic Approximation Monte Carlo algorithm. In addition, we demonstrate how parallel SAMC algorithm can be applied to estimate the parameter of the model and propose combining strategies to aggregate the information from parallel results.

SAMC algorithm is a Stochastic Approximation Monte Carlo algorithm. It is an adaptive MCMC algorithm that has been proven in theory and application. Unlike other conventional MCMC algorithm, such as Metropolis-Hasting, SAMC algorithm can avoid being trapped into local minima by additionally updating weight parameter in each iteration step. Because of that, SAMC can converge to the desired target distribution quickly. It can perfectly be applied to subset analytic.

The parallel computation is a modern technology which allows to break a target problem into several smaller pieces and to calculate these pieces simultaneously. Although the idea of parallel computation is very straightforward, however, there is no statistical literature discussing how to combine the parallel results and how accurate the estimator is in parallel computing.

The Bayesian inference via Divided-and-Combined method is the first method we propose for aggregation step. It can be treated as a fixed effect problem with an unknown mean effect and within-study error. It is straightforward to solve mean effect by ML estimator. This aggregation step can be done very fast because there is no subsets communication during the parallel computation. In other words, each subset only evaluates its posterior mean without communicating with other subsets.

Compared with the first aggregation method, we propose a Bayesian inference via

Divided-and-Resample method, which allows communication among other subsets in the aggregation step. The basic idea of recombination re-sampling algorithm is to derive the common shared belief function between different subsets. The numerical results show that this method has more accurate estimation than Bayesian inference via Divided-and-Combined aggregation but it will take longer processing time.

By taking the advantage of fast processing time in first aggregation method and more accurate estimation for communicating with subsets in the second aggregation method, we propose the third aggregation method: weighted combination via SAMC importance sampling. It calculates the correctly weighted parameter $w^{(1:k)}$ for all subsets to evaluate big data posterior mean. The numerical results shows that it is not only as fast as the first aggregation method but also more accurate than second aggregation method.

The strength of parallel computation is to reduce the processing time. Our numerical results suggest that parallel SAMC algorithm can save the process time compared with sequential computation. As we increase the subsets which break from the big data, it dramatically decreases the calculating time. The application of space shuttle data shows that parallel SAMC algorithm keeps the same prediction rate (80%) as we do the entire data inference. Therefore, we conclude that parallel SAMC algorithm is a new, promising method which is working perfectly in big data problem.

The second part of this dissertation focuses on a deformation framework for a stationary spatial random field and investigates the effect of making wrong stationary assumption. We use an InSAR data to demonstrate an example of non-stationary random field via stationary test. It is non-trivial to examine by eye whether the fields is stationary or not. Therefore to make a stationary assumption for InSAR noise data is not appropriate. We also propose a deformation framework to investi-

gate the nonstationarity with gradually increasing the deformation parameters. The numerical study suggests that as dependency of deformation parameter b increases to the boundary, the power test for stationary test achieves 90% in larger testing random field, which indicates non-stationary model for sure. That is, instead of generating a whole new non-stationary model, we can use deformation framework in a stationary model to create a parameter nonstationarity random field.

REFERENCES

Berg, B. A. and Neuhaus, T. (1991), "Multicanonical algorithms for first order phase transitions," *Physics Letters B*, 267, 249–253.

Ferguson, T. S. (1996), *A Course in Large Sample Theory*, New York: Chapman and Hall/CRC, 1996th ed.

Geyer, C. J. (1991), "Markov chain Monte Carlo maximum likelihood," in *Computing Science and Statistics : Proceedings of the 23rd Symposium on the Interface (pp.153-163)*, Seattle, WA : Interface Foundation.

Guan, Y., Sherman, M., and Calvin, J. A. (2004), "A nonparametric test for spatial isotropy using subsampling," *Journal of the American Statistical Association*, 99, 810–821.

Hall, P. (1988), "On confidence intervals for spatial parameters estimated from nonreplicated data," *Biometrics*, 44, 271–277.

Hanssen, R. F. (2001), *Radar Interferometry: Data Interpretation and Error Analysis*, Dordrecht: Springer.

Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, 97–109.

Heagerty, P. J. and Lumley, T. (2000), "Window subsampling of estimating functions with application to regression models," *Journal of the American Statistical Association*, 95, 197–211.

Iovleff, S. and Perrin, O. (2004), "Estimating a nonstationary spatial structure using simulated annealing," *Journal of Computational and Graphical Statistics*, 13, 90–105.

Jun, M. and Genton, M. G. (2012), "A test for stationarity of spatio-temporal random

fields on planar and spherical domains," *Statistica Sinica*, 22.4, 1737.

Kou, S. C., Zhou, Q., and Wong, W. H. (2006), "Equi-energy sampler with applications in statistical inference and statistical mechanics," *The Annals of Statistics*, 34, 1581–1619.

Liang, F. (2009), "On the use of stochastic approximation Monte Carlo for Monte Carlo integration," *Statistics & Probability Letters*, 79, 581–587.

Liang, F., Liu, C., and Carroll, R. J. (2007), "Stochastic approximation in Monte Carlo computation," *Journal of the American Statistical Association*, 102, 305–320.

Liang, F. and Wong, W. H. (2001), "Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models," *Journal of the American Statistical Association*, 96, 653–666.

Lin, N. and Xi, R. (2011), "Aggregated estimating equation estimation," *Statistics and Its Interface*, 4, 73–83.

Marinari, E. and Parisi, G. (1992), "Simulated tempering: a new Monte Carlo scheme," *Europhysics Letters*, 19, 451.

Mengersen, K. L. and Tweedie, R. L. (1996), "Rates of convergence of the Hastings and Metropolis algorithms," *The Annals of Statistics*, 24, 101–121.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, 21, 1087–1092.

Mogi, K. (1958), "Relations between the eruptions of various volcanoes and the deformation of the ground surfaces around them," Tech. Rep. 36, Bull Earthquake Res. Inst.

Obled, C. and Creutin, J. D. (1986), "Some developments in the use of empirical orthogonal functions for mapping meteorological fields," *Journal of Climate and*

Applied Meteorology, 25, 1189–1204.

Perrin, O. and Monestiez, P. (1999), *Modelling of Non-Stationary Spatial Structure Using Parametric Radial Basis Deformations*, no. 10 in Quantitative geology and geostatistics, Springer Netherlands.

Politis, D. N., Romano, J. P., and Wolf, M. (1999), *Subsampling*, New York: Springer, 1999th ed.

Possolo, A. (1991), "Subsampling a random field," *Lecture Notes-Monograph Series*, 20, 286–294.

Powell, M. J. D. (1987), *Algorithms for Approximation*, New York, NY, USA: Clarendon Press.

Robbins, H. and Monro, S. (1951), "A stochastic approximation method," *The Annals of Mathematical Statistics*, 22, 400–407.

Sampson, P. D. and Guttorp, P. (1992), "Nonparametric estimation of nonstationary spatial covariance structure," *Journal of the American Statistical Association*, 87, 108–119.

Sherman, M. and Carlstein, E. (1994), "Nonparametric estimation of the moments of a general statistic computed from spatial data," *Journal of the American Statistical Association*, 89, 496–500.

Wang, F. and Landau, D. P. (2001), "Efficient, multiple-range random walk algorithm to calculate the density of states," *Physical Review Letters*, 86, 2050–2053.

Wong, W. H. and Liang, F. (1997), "Dynamic weighting in Monte Carlo and optimization," *Proceedings of the National Academy of Sciences*, 94, 14220–14224.

APPENDIX A

PROOFS OF CHAPTER II

(A_1) The sequence $\{\gamma_t\}$ is positive and non-increasing, and satisfies the conditions:

$$\lim_{t\to\infty} \gamma_t = 0, \qquad \sum_{t=1}^{\infty} \gamma_t = \infty, \qquad \sum_{t=1}^{\infty} \gamma_t^{\eta} < \infty$$

for some $\eta \in (1, 2)$

(A_2) Let P_θ denote the MH transition kernel for a given $\theta \in \Theta$. For any $\theta \in \Theta$, P_θ is ψ-irreducible and aperiodic (Meyn and Tweedie, 1995). In addition, there exist a function V: $X \to [1, \infty)$ and a constant $\alpha \geq 2$ such that for any compact subset $\mathcal{K} \subset \Theta$,

1. there exist a set $\mathcal{C} \subset X$, an integer l, constants $0 < \lambda < 1$, b, ζ , $\delta > 0$ and a probability measure ν such that

2. there exists a constant c such that for all $(\theta, \theta') \subset \mathcal{K} \times \mathcal{K}$

APPENDIX B

PROOFS OF CHAPTER III

If G is a open subset of R^2 and if f : G $\rightarrow R^2$ is an injective and continuously differentiable function that Jocobian determinat $\det(J_f(x)) \neq 0$ for all x $\in G$, then $f(G)$ is a open subset of R^2 and f defines a diffeomorphism from G onto $f(G)$.

For Gaussian radial deformation with c=c(0,0), $f = \rho(1 + b\exp(-a\rho^2))$, which the Jacobian determinant is $\det(J_f(x)) = 1 + bexp(-a\rho^2)[1 - 2a\rho^2]$. Now we must to determine the parameter a and b in which the $\det(J_f(x))$ is not equal to zero.

Let $g(\rho) = 1 + bexp(-a\rho^2)[1 - 2a\rho^2]$, then take the first derivative to $g(\rho)$ and set $g^{'}(\rho) = 0$

$$g^{'}(\rho) \Rightarrow -2ab\exp(-a\rho^2)[3 - 2a\rho^2] = 0$$
$$\rho = \sqrt{\frac{3}{2a}}$$

when $g^{'}(\rho) > 0 \Leftrightarrow b < \frac{1}{2}\exp(\frac{3}{2})$

$g^{'}(\rho) < 0 \Leftrightarrow b > -1$

The above have proved that when $b \in [-1, \frac{1}{2}\exp(\frac{3}{2})]$ the function f is a bijective function.