University of Cape Town

Department of Statistical Sciences

Honours Thesis

Approximating the Gaussian Processes

Author:
Mishan Phiri

Student Number:
PHRMIS001

Supervised by: Professor Linda Haines

December 12, 2023

Contents

Acknowledgements

I would like to acknowledge the support of my supervisor Emeritus Professor Linda Haines and Benjamin Chiddy for their assistance during this project. My friends and family for cheering me onwards, and my Lord Jesus Christ for making sure I reach the end. Finally I would like to acknowledge the University of Cape Town's ICTS High-Performance Computing team: for assistance with setting up the simulation environment.

Abstract

The onset of big data has introduced new challenges in the geostatistics community. Kriging is a prominent prediction tool used in spatial statistics, but is has cubic time complexity and as datasets become larger and larger, kriging can potentially become computationally inefficient. Most users of this model have limited computing resources, and approximations to large datasets started being developed to scale the large datasets, to workable design spaces. This project makes use of kriging to predict land surface temperature using different subsampling methods to approximate the large dataset. Namely, random sampling, latin hypercube sampling, cover designs and local approximation gaussian process. Subsampling is simulated one thousand times to obtain performance measures and the methods are compared. We find that no method is perfect and choosing one depends on the objective of the analysis.

1 Introduction

Statistical Learning refers to tools used to make sense of complex datasets. With the onset of modern society and the Internet of Things (IoT), the size of complex datasets has exploded in both scale and scope. This has made statistical learning a critical toolkit for anyone wishing to understand high-dimensional and complex datasets. Statistical learning involves building a statistics-based model for predicting or estimating a response based on the observed or predictor values. This is known as supervised learning, and allows us to learn structures from data. The disciplines that dictate the core principles used in statistical learning were developed long ago before applications that can track and store the movement of billions of users to predict traffic were invented.

This type of data is called *spatial data* and is everywhere. Spatial data are characterised by the presence of coordinate values and a system of reference, which is considered when modelling the data. The analysis is concerned with hypothetical processes that generate the observed data. Statistical inference for spatial analysis is often challenging, as there are massive datasets generated from satellite imagery, location pings, or geographic borders. Hence, these observations cannot be assumed to be mutually independent, as the data points close in location are likely to be similar. The latter is referred to as spatial autocorrelation, and is useful for predicting values at unobserved locations. A popular tool in spatial analysis is *Gaussian Process Regression*, a non-parametric model that takes into account the correlation of the distance between points. It is a popular surrogate model used in spatial data analysis and is generally referred to as *kriging*.

Gaussian processes (GP) have revolutionised spatial statistics and computer simulation experiments. Instead of gathering our own data, we tend to generate data using computer codes. These techniques do not come without a caveat: The primary challenge when working with GPs is its computational complexity. $\mathcal{O}(n^3)$ operations are needed when evaluating the kernel and thus when the number of observations is sufficiently large the modeling process is no longer applicable due to resource limitations. In the case of Maximum Likelihood Estimation, the training set is limited to thousands, as the GP can be ruled out when the number of observations exceeds 2000 (Gramacy, 2020). In pursuit of faster results, geostatisticians have begun developing methods to handle high-dimensional datasets. Early solutions to the scalability problem include using tapered covariance functions, low-rank approximations, model surrogates and many more methods. However, these remedies are not without their own shortcomings. The approximate methods often oversmooth the data or have upper limits to the size of the data it can model (Heaton et al., 2019). Hence modern methods are developed to focus on parallelizing the computation.

First, this project briefly outlines the Bayesian framework of a Gaussian process. It then shifts to explaining gaussian process regression (kriging) in terms of spatial statistics in Section 2.3. Next, we discuss what happens when the gaussian process meets big data, and an overview of the remedies developed is discussed in Section 3. Thereafter, we examine what kriging is and how it makes predictions and uses its parameters. Subsequently, simulations are discussed in Section 4 to test the aforementioned remedies. Finally, the results are discussed, and the topic is concluded.

The Aim

This project focuses on comparing different methods developed to solve the big $\mathcal{O}(n^3)$ problem of gaussian process regression. Global and Local Approximation methods are used. Simulations were performed for each method, and prediction errors were tracked and compared.

2 Gaussian Process

Although a somewhat recent development in the context of machine learning (Williams and Rasmussen, 1995) the idea of probabilistically modelling a function has its origins in mid-century spatial statistics (Krige, 1951). In this context we are interested in predicting the value of f at an *unobserved* location $\mathbf{x}_*$ using n observations $\boldsymbol{f} = [f(\mathbf{x_1}), f(\mathbf{x_2}), \ldots, f(\mathbf{x_n})]^T$ obtained at n locations $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ within a study area (or input space $\mathcal{X}$). In his masters thesis, Ngwenya (2011) describes the spatial process as

$$F(\cdot) = \{F(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p\} \tag{2.1}$$

operating in p dimensions over the study area. Ngwenya then outlines the practical problem of predicting $F(\cdot)$ at $\mathbf{x}_* \in \mathcal{X}$ using $\boldsymbol{f} = [f(\mathbf{x_1}), f(\mathbf{x_2}), \ldots, f(\mathbf{x_n})]^T$ where we assume $\boldsymbol{f}$ is a realization of the n random variables $\mathbf{F} = [F(\mathbf{x_1}), F(\mathbf{x_2}), \ldots, F(\mathbf{x_n})]^T$. This problem is colloquially known as *Kriging* after its originator Danie Krige who in his 1951 masters thesis used it to predict the location of gold reserves in the Witwatersrand region of South Africa. While a physical process is typically modelled in 2 or 3 dimensions, there are no practical limitations when considering more than 2 inputs ($p > 2$).

When we consider the random variables $\mathbf{F}$ to be jointly Gaussian, the process is known as a Gaussian Process - formally defined as follows:

> *"A Gaussian Process is a collection of random variables, where any finite subset has (consistent) joint Gaussian distributions."* (Rasmussen, 2004)

2.1 Generalization of the Gaussian Distribution

A helpful way to think about the GP is as a generalization of the Gaussian distribution. While the Gaussian distribution is defined by its finite-dimensional mean vector and covariance matrix, the GP extends this concept into an infinite-dimensional space where it is defined over functions (Rasmussen and Williams, 2006). In this infinite-dimensional space the GP is characterized by its first two moments, both of which are themselves now functions. We describe a GP as

$$f \sim \mathcal{GP}(m, K) \tag{2.2}$$

meaning the function f is distributed as a GP with mean function m and *covariance function* or *kernel* K such that

$$\begin{aligned} m : \mathcal{X} \to \mathbb{R} &= \mathbb{E}[f(\mathbf{x})] \\ K : \mathcal{X} \times \mathcal{X} \to \mathbb{R} &= \mathbb{E}[(f(\mathbf{x})m(\mathbf{x}))(f(\mathbf{x}')m(\mathbf{x}'))] \end{aligned} \tag{2.3}$$

The mean function m encodes the central tendency of the function and is often assumed to be stationary. In the spatial statistics literature, this assumption is known as *Ordinary Kriging* when the value of $m(\mathbf{x})$ is unknown and *Simple Kriging* when $m(\mathbf{x})$ is known and usually taken to be 0. The covariance function K encodes information about the shape and structure we expect the function to have, or in other words how every location $\mathbf{x}$ is related to every other location $\mathbf{x}'$ (Garnett, 2023). A common choice is the squared exponential kernel, defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\{||\mathbf{x}\mathbf{x}'||^2\}. \tag{2.4}$$

What this tells us is that the covariance between $\mathbf{x}$ and $\mathbf{x}'$ (defined as Euclidean distance in 2.4) decays exponentially fast as $\mathbf{x}$ and $\mathbf{x}'$ become further apart (Gramacy, 2020). Note that $K(\mathbf{x}, \mathbf{x}) = 1$ and $K(\mathbf{x}, \mathbf{x}') < 1$ for $\mathbf{x} \neq \mathbf{x}'$. In addition, just as with the covariance matrix of a Gaussian distribution we require that $K(\mathbf{x}, \mathbf{x}')$ be *positive definite* and *symmetric* such that

$$\mathbf{x}^T K \mathbf{x} > 0 \ \forall \ \mathbf{x} \neq 0 \text{ and } K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x}) \text{ respectively.} \tag{2.5}$$

Besides the squared exponential there are many other popular types of popular kernels which share the above properties, most notably the Matérn which is used extensively in the context of machine learning (Gramacy, 2020). In what follows we briefly describe both of these kernels but focus on the squared exponential kernel due to its relative ease-of-use and flexibility. Ngwenya (2011) gives specifications of several other popular kernels.

2.2 Bayesian Inference on f using Gaussian Processes

We may now ask ourselves: given a *training* set $D_n = (\mathbf{x}_n, f(\mathbf{x}_n))$ and a collection of unseen locations $\mathbf{x}_* \in \mathcal{X}$, what are the plausible realizations of $\boldsymbol{f}$ that could explain the observed values? In essence, our focus lies in understanding the conditional distribution of $f(\mathbf{x}_*)|D_n$. If we regard $f(\mathbf{x})$ to be the prior, then $f(\mathbf{x}_*)|D_n$ naturally becomes the posterior as emphasized by Gramacy (2020). The computation of this posterior distribution allows us to refine our initial beliefs on the form of f, represented by the prior in light of the training data. Consequently, we gain the ability to make inference on the underlying functional form of f across a set of unseen test cases. This objective aligns closely with the core principles of statistical learning, as was outlined in our introduction

2.2.1 Gaussian Process Posterior

For notational convenience, let $f(\mathbf{x}) \equiv \boldsymbol{f}$ be the *known* function values of the training cases $\mathbf{x} \in \mathcal{X}$ and $f(\mathbf{x}_*) \equiv \boldsymbol{f_*}$ be the function values corresponding to the *unseen* test cases $\mathbf{x}_* \in \mathcal{X}$. By definition, each of these quantities have a joint Gaussian distribution such that $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and $\boldsymbol{f}_* \sim \mathcal{N}(\boldsymbol{\mu}_*, \Sigma_*)$ where $\boldsymbol{\mu} = m(x_i),\ i = 1, 2, ..., n$ is the vector of training means and likewise $\boldsymbol{\mu}_*$ is the vector of test means. Equivalently $\Sigma = K(x_i, x_j),\ i, j = 1, 2, ..., n$ is the training set covariance matrix while Σ_* is the test set covariance matrix. Recall that by Equation 2.5 these are both valid covariance matrices. We can now write out the joint distribution between $\boldsymbol{f}$ and $\boldsymbol{f}_*$ as

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{bmatrix} \right). \tag{2.6}$$

Since we are interested in the function values at the unseen test locations conditional on the training data we can derive the following expression using the laws of the multivariate-normal distribution,

$$\boldsymbol{f}_* | \boldsymbol{f} \sim \mathbf{N}(\mu_* + \Sigma_*^T \Sigma^{-1}(\boldsymbol{f} - \boldsymbol{\mu}),\ \Sigma_{**} - \Sigma_*^T \Sigma^{-1} \Sigma_*). \tag{2.7}$$

This is the posterior distribution for any set of test cases $\mathbf{x}_* \in \mathcal{X}$ (Rasmussen, 2004). The corresponding **posterior** GP is therefore

$$\begin{aligned} f|D &\sim \mathcal{GP}(m_D, K_D) \\ m_D(x) &= m(x) + \Sigma(\mathbf{x}, x)^T \Sigma^{-1}(\boldsymbol{f} - \boldsymbol{m}) \\ K_D &= K(x, x') - \Sigma(\mathbf{x}, x)^T \Sigma^{-1} \Sigma(\mathbf{x}, x') \end{aligned} \tag{2.8}$$

where $\Sigma(\mathbf{x}, x)$ is the vector of covariances between every training case and $x_1, x_2, ..., x_n$ (Rasmussen, 2004). Notice that $m_D(x)$ is a linear predictor, in fact it is the Best Linear Unbiased Predictor (BLUP) for $\boldsymbol{f}_*$ as proven by Ngwenya (2011). Also note that the posterior variance K_D is equal to the prior variance minus a positive term meaning we have indeed *learnt* something from the training data (Gramacy, 2020).

As demonstrated by Equation 2.7 our path to Bayesian inference on an infinite-dimensional stochastic process requires merely the computation of a conditional Gaussian distribution. This insight allows us to make predictions based solely on data, without the need to *fit* any parameters or minimize some loss criterion. This is why GPs are so highly regarded as non-parametric regression tools (Gramacy, 2020). The equations in 2.8 are what's known as the *Kriging* equations and were first derived by Danie Krige in 1951, some 40 years before Williams and Rasmussen (1995) would reinterpret them through a Bayesian perspective.

2.2.2 1D Prediction Example

Although GPes are infinite-dimensional objects, by definition any finite subset has a joint Gaussian distribution. This is of great practical importance as it allows us to draw samples from f by simply computing the related distribution at n locations within $\mathcal{X}$. Consider the following example which is an adaption on one which is seen in *Gaussian Processeses in Machine Learning* (Rasmussen, 2004). Suppose we have the simple one dimensional process defined as

$$f \sim \mathcal{GP}(m, K) \text{ where } m = 2\mathbf{x}\cos(\mathbf{x}) \text{ and } K(\mathbf{x}, \mathbf{x}') = \exp\{||\mathbf{x}\mathbf{x}'||^2\} \tag{2.9}$$

Given a set of $\mathbf{x} \in \mathcal{X}^1$ we can compute a vector of means and covariance matrix using m and K, the result of which will be a Gaussian distribution with

$$\begin{aligned} \mu_i &= m(x_i) = 2x_i \cos(x_i) \;\; i = 1, 2, ..., n \\ \Sigma_{ij} &= K(x_i, x_j) = \exp\{||x_i x_j||^2\} \;\; i, j = 1, 2, ..., n \end{aligned} \tag{2.10}$$

If we now draw a random sample from $f|\mathbf{x} \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ the corresponding vector will have as its coordinates the function values $f(x)$ for each corresponding $x_1, x_2, ..., x_n$.

In Figure 1 we can see 100 random realizations of the GP defined in Equation 2.9 and shows why GPs are often referred to as *distributions over functions*. This feature makes them an ideal choice of prior for Bayesian inference on f. We now return to our earlier example described in Equation 2.9. Instead of generating random samples from f, we now obtain $n = 9$ training cases from the area of interest within our input space $(-6, 6) \subset \mathcal{X} = \mathbb{R}$. We then use the *Kriging* equations given in 2.8 to obtain the posterior distribution for all *unseen* test cases within the input space. Figure 2 below

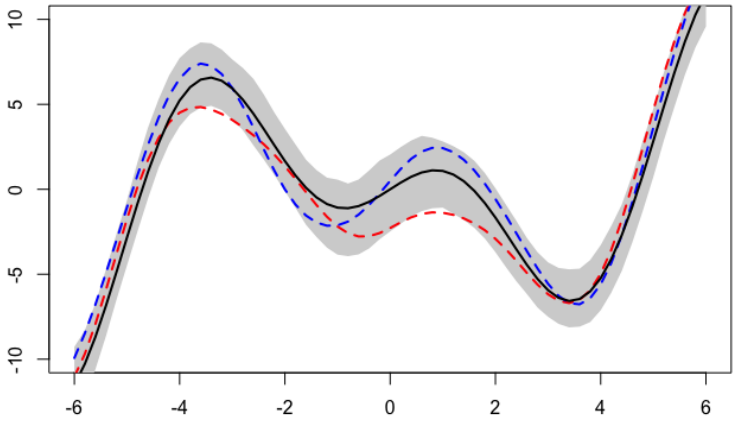

Figure 1: Two random realizations of f shown as dotted red and blue lines drawn from the GP defined in Equation 2.9 whose true form is shown as a solid black line. The shaded gray region represents 95% confidence intervals from 100 similar realizations of f.

shows 100 sample draws from this posterior distribution superimposed over the true response from f for these test locations (shown in black). Notice the sausage shape that the distribution of these samples take the further they are from the sampled locations. This is due to the fact that $K(x, x) = 1$ and $K(x, x') \to 1$ as $x' \to x$ which tells us we are interpolating the space between each of these sampled locations (Gramacy, 2020). This is known as *honouring* the training data in the context of geostatistics and is a reason why GPes are also popular tools in the modelling of large scale computer experiments where a quantification of uncertainty is required (Santner et al., 2003). Finally note that towards the edge of the study area, where we have less training examples, our posterior samples start to diverge from the true value of f. This warns us against making predictions at locations too far outside of the area of focus.

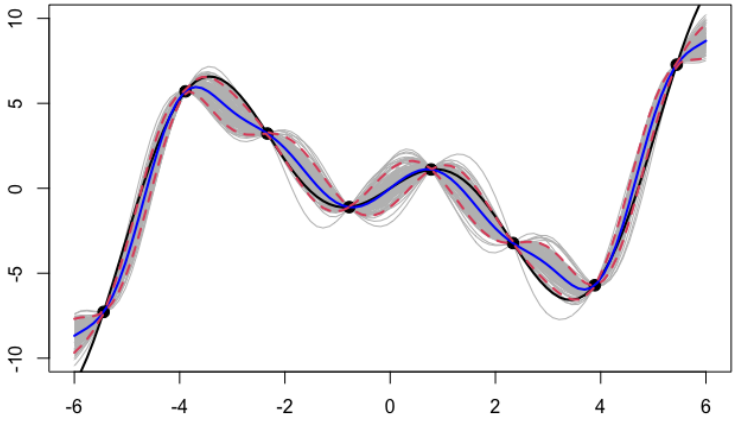

Figure 2: 100 samples from the posterior distribution of Equation 2.9 shown in grey with the posterior mean m_D shown in blue and the true response in black. 95% credibility intervals for m_D are shown as dotted red lines and the $n = 9$ training cases are shown as black dots.

2.3 Training a Gaussian Process

In the previous section, we discussed how a GP can be used within a Bayesian framework to make inferences about an unknown function f. We did this by specifying a prior mean and covariance function, which when combined with training data, allowed us to compute a posterior distribution. In spatial statistics literature, kriging is considered a parametric model. In this context, we would specify a covariance structure (known as a semivariogram) and fit its parameters through either ordinary least squares or maximum likelihood estimation. Surprisingly (or perhaps confusingly) this leads to the posterior gaussian process equations seen in Equation 2.8 without any Bayesian reasoning. Instead, we want to be able to choose the prior mean and covariance functions in light of the data; this is referred to as model *training* by Rasmussen (2004). The covariance function and mean were parameterised using a set of hyperparameters. At this point, we move away from the Bayesian interpretation given by Rasmussen (2004) and focus on the parametric approach to spatial statistics.

2.3.1 Hyperparameters of the Exponential Covariance Function

The squared exponential kenel introduced in Equation 2.4 is the covariance function. What this prior encodes is our belief that "covariance decays exponentially fast as $\mathbf{x}$ and $\mathbf{x}'$ become farther apart" (Gramacy, 2020). The covariance function is parameterised in the following way

$$K(x, x') = \tau^2 + \sigma^2 \left(1 - exp\left(\frac{-||x - x'||}{3\lambda}\right)\right) \tag{2.11}$$

with parameters $\boldsymbol{\theta} = \{\tau^2, \lambda, \sigma^2\}$. Below we provide some intuitions for each of the hyperparameters contained within $\boldsymbol{\theta}$:

- **Partial Sill** σ^2 - or *scale* controls the maximum covariance between any two observations x_i and x_j where $i, j = 1, 2, ...n$. The **sill**$= \tau^2 + \sigma^2$
- **Range** λ - or *length-scale* controls the rate-of-decay or how far-reaching the effect each observation x_i has on every other observation x_j where $i, j = 1, 2, ...n$.
- **Nugget** τ^2 - or *noise variance*is an extra source of variation applied individually to each observation.

2.3.2 Matern Covariance Function

Although the squared exponential covariance function provides an introduction to Gaussian Processes, it is not always used in practice. In this section, we discuss the Matérn family of functions which are more popular in the machine learning community (Gramacy, 2020) and attractive in geostatistics. The function is parameterised as follows:

$$k_\nu(h) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{h}{\lambda}\right) K_\nu \left(\frac{h}{\lambda}\right) \tag{2.12}$$

Where:

- **Bessel function** K_ν - is a Bessel function of the second kind, it is essential for determining the correlation structure between points as well as for modeling smoothness
- **Smoothing Parameter** ν: determines how the Matérn will behave. At $\nu = 0.5$ is in fact the exponential covariance function, and as $\nu \to \infty$ the kernel approximates the Gaussian kernel.
- h: is the distance between two points

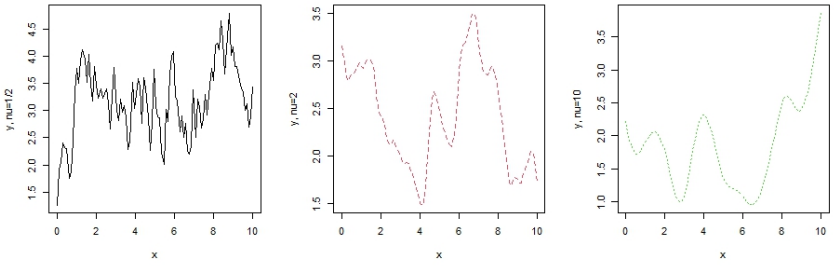

Figure 3: Sensitivity to Smoothing Parameter: From left to right Sample paths under Matèrn kernels with $\nu = 0.5$, $\nu = 2$ and $\nu = 10$

The selection of ν is important because the function is sensitive to this parameter. Figure 3 shows the sample paths under various values of ν, and it is evident that as ν increases, so does the smoothness of the plot. The literature supports the idea that smoothness should be learned from a training set; however, this poses a problem as datasets contain noise, and there is a lack of guidance on how to identify noise (Gramacy, 2020). This can result in a likelihood surface with many flat spots. The evaluation of K_ν is computationally expensive and challenging to implement for large values of n in the training set X_n. This slows down the runtime of the Matérn. The matérn can be re-expressed by defining ν as $\nu = p + \frac{1}{2}$ where $p > 0$ and p is an integer. Equation 2.12 can be reformulated as

$$k_{p+1/2}(h) = \frac{1}{2^{k-1}\Gamma(\nu)}(\frac{h}{\lambda})^\nu \tag{2.13}$$

This re-parametrization removes the Bessel Function for small p. This re-parametization is a member of the exponential family. It has been established that this kernel is very sensitive to our choice of ν it is important that a middle ground between the two extremes are established. It is hard to distinguish between values when $p \geq 3$, thus we are left with the three special cases cases were $p = 0, 1, 2$. This yields :

$$k(h) = exp - \frac{h^2}{\theta} \tag{2.14}$$

$$k_{3/2}(h) = \left(1 + r\frac{3}{\theta}\right) exp\left(-r\sqrt{\frac{3}{\theta}}\right) \tag{2.15}$$

$$k_{5/2}(h) = \left(1 + r\frac{5}{\theta} + \frac{5r^2}{3\theta}\right) exp\left(-r\sqrt{\frac{5}{\theta}}\right) \tag{2.16}$$

These special cases have no Bessel Functions, factorials or gammas. While the Matérn kernel allows for the adjustment of smoothness levels, and provides minimal control in this regard. Matérn family of functions have the potential to enhance numerical stability due to their improved covariance matrix properties.

2.4 Brief Overview of Parameter Estimation

2.4.1 Least Squares Method

These methods are based on Ordinary Least Squares, where these models are concerned with fitting curves in the generic semivariogram models (Ngwenya, 2011). They aim to minimise the sum of squares between the empirical semivariogram and the fitted semivariogram. These methods are popular for various reasons namely (Ngwenya, 2011):

- They make no distributional assumptions about the data or semivariogram
- They are quick and easy to use, and require minimal computational power
- The semivariogram fits are easy to interpret.

2.4.2 Likelihood method

The most popular method is spatial statistics for obtaining the model estimates are the maximum likelihood. These methods require the spatial distribution to be known. These models tend to produce biased estimates, as it does not take into account the loss in degrees of freedom from simultaneously estimating the parameters (Ngwenya, 2011). The only issue for this method is it is computationally expensive for larger datasets. Advantages of this model include:

- The parameters are estimated based on the actual data, and model residuals have a sensitive interpretation
- It is a statistical basis for parameter estimation.
- It allows for model comparison using AIC or BIC

2.4.3 Semivariogram

The semi-variance(γ) in geostatistics is a measure of spatial dependence and is one of the methods used to determine the model hyper-parameters . The relationship between points a distance apart h, can be expressed as the variance of the discrepancy between pairs. This is referred to as the semi-variance. It is half the expected squared distances at distance h (Ngwenya, 2011).
$2\gamma(\mathbf{h})$ is coined the variogram and half of it is the semivariogram $\gamma(h)$. It is the semi-variance γ_{ij} plotted against the distance h. This makes it easier to identify local trends and reveal geographic variations in an area of interest (Burgess and Webster, 1980).

The shape of the semi-variogram informs us of the covariance structure that the data follows and highlights the hyperparameters that define the Covariance Function. Hence the parameters are estimated from the data. Figure 4 shows the plot of a typical semivariogram. Semi-variance increases with h and is a continuous function. Burgess and Webster (1980) explains how Figure 4 illustrates the parameters of the

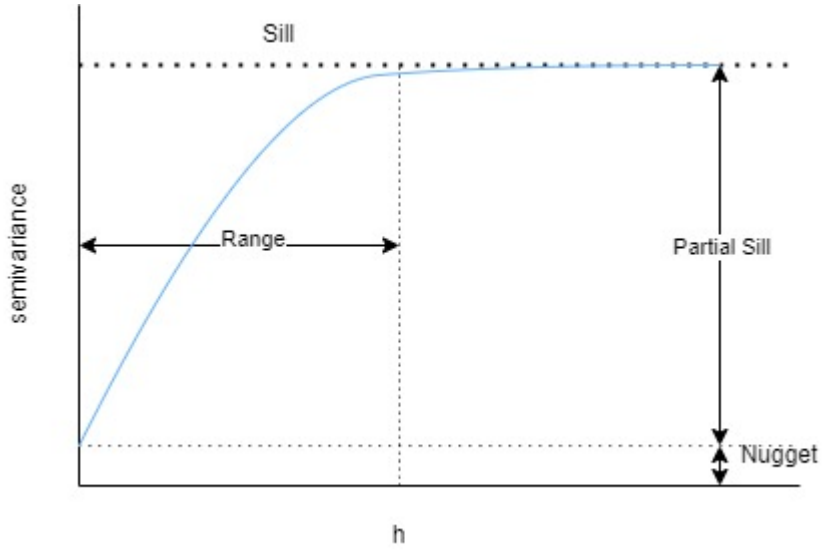

Figure 4: An example of a typical variogram with the parameters indicated

models, They explain that γ increases until a maximum which is the range. Points with a discrepancy less than the range are spatially dependent. The graph intercept is the nugget variance and is often zero. The value at which the variogram begins to show asymptotic behaviour is the sill, and which consists of the nugget and the partial sill that represents the range of variance attributed to spatial dependence. The hyperparameters are defined in Section 2.3.1.

3 Gaussian process meets Big Data

Big Data are high dimensional datasets that are often collected automatically or have various sources of information and are a result of uncontrolled observational studies. Large datasets magnify small effects due to the size of the dataset (Deldossi and Tommasi, 2021). This characteristic of big data led to the idea of using nearest neighbours or subsamples to represent larger datasets. Deldossi and Tommasi (2021) suggest that a set chosen to represent a dataset can either be descriptive or analytical. The former focuses on finite population parameters taken from a subset of the data. The latter takes a model-based approach where parameters are estimated from the full dataset. In their paper Optimal design subsampling, Deldossi and Tommasi (2021) proposes a purposeful selection subsampling strategy that allows us to select the most informative representation of the Full dataset

This section explores different methods for handling big data when it meats the GP. One of the GPs weakness is that it requires $\mathcal{O}(n^2)$ complexity to run. Hence, the focus has shifted from the development of traditional statistical models to techniques that can accommodate the size of datasets (Drovandi et al., 2017) and improve scalability while maintaining a desirable level of prediction accuracy. A variety of scalable GPs have been developed, which can be subdivided into two main categories: Global and Local Approximations (Liu et al., 2020). Global Approximations can be considered as an analytical survey and the Local Approximation a descriptive survey of the data

3.1 Global Approximations

These models approximate the kernel K of n observation through global distillation. Liu et al. (2020) suggests that this can be achieved by the following:

1. **Subset of the data**: using a subset/ subsample as a surogate of the training data with $m << n$ points results into a smaller kernel. This is the simplest strategy to approximate the full Process. These Subsets $\mathcal{D}$ carries out inference at a reduced time complexity $\mathcal{O}(m^3)$ as K only has m ¡¡ n data points. This method however struggles to produce a decent prediction variance due to the reduced sample size. When selecting a candidate subsample can either be done by be selected randomly or by a deterministic approach such as KD tree or active learning.
2. **Sparse Kernels**: this method aims to thin out the representative correlation matrix by removing entries close to 0 in K, therefore only the non-zero elements remain in K as a result time complexity is reduced to $\mathcal{O}(\alpha n^3)$ where $0 < \alpha < 1$, a fraction of the original time spent.
3. **Sparse Approximations**: uses a low-rank approximation as a representation of the full dataset measured. Eigenvalue decomposition to assist in choosing the first m eigenvalues used to approximate the Kernal. This results in a reduced complexity of $\mathcal{O}(nm^2)$

Global approximations tend to capture global trends better and filter out local trends due to the decreased sample size, yet they still maintain their capability to capture long-term spatial correlations,

3.2 Local Approximations

These methods divide and conquer the data for subspace learning. Subsets of the training data with $m << n$ points are clustered and analysed in parallel while ignoring points outside each cluster, and the results are combined. Hyperparameters are either joined for the entire training set or unique to each cluster; however, this affects time complexity (Chalupka et al., 2013).
Downfalls of this method include poor predictions along the cluster boundaries due to the discontinuities in data sets (Chalupka et al., 2013).
The Local Approximation Gaussian Process (LaGP) was developed to address the scalability problem through a transductive approach (Heaton et al., 2019; Gramacy, 2016), where fitting is tailored to each problem as opposed to the inductive approach of fitting, followed by the prediction and improvement of the model's performance. A special case occurs when the algorithm trains a GP predictor on the nearest m neighbours to s. The data are subset into D_m where:

$$D_m = \{Y(s_i) : s_i \epsilon N_m(s)\} \tag{3.1}$$

where $N_m(s)$ are the m nearest neighbours to s in terms of Euclidean Distance (Heaton et al., 2019). If the data has a well-defined Covariance structure or conforms to the modelling assumption later discussed in section **??** then m can take on any positive integer value.

LaGP Active Learning Algorithm

1. Start with a nearest neighbour set for s $D_{m_0}(s)$ where $m_0 < m$.
2. For $i = m_0 + 1, ..., m$ choose s_i to add to $D_{m_0}(s)$. selecting s_i according to some criterion that minimises mean prediction error.
3. Repeat until there are m observations in $D_{m_0}(s)$.

However finding the best m is a huge search as there, as there are $\frac{N!}{m!(N-m)!}$ possible options. Moreover, calculations of each s_i are statistically independent and can be run in parallel. This nature allows the model to be computationally efficient (Heaton et al., 2019). Despite the method's shortfalls, Local methods outperform Global methods with regard to predictions on larger data sets with time complexity of $O(m^2 n)$ vs. Global methods with $O(m)$ time complexity (Chalupka et al., 2013). Hybrid methods exist that combine both approaches.

3.3 Focus in this Project

The experiments carried out in this section aim to answer two questions: How do we take a subsample and how large should the subsample be? The choice of the subsample selection depends on the research objectives. In this case, we aimed to predict the surface temperature of the area of interest. The ideal subsample size balances the trade-off between statistical power and computational efficiency. If the subsample is too small, it may oversmooth the data; however, if it is too large, it will overfit the dataset and become computationally expensive. Careful consideration of these questions can help ensure that the choice of the subsample is a representation

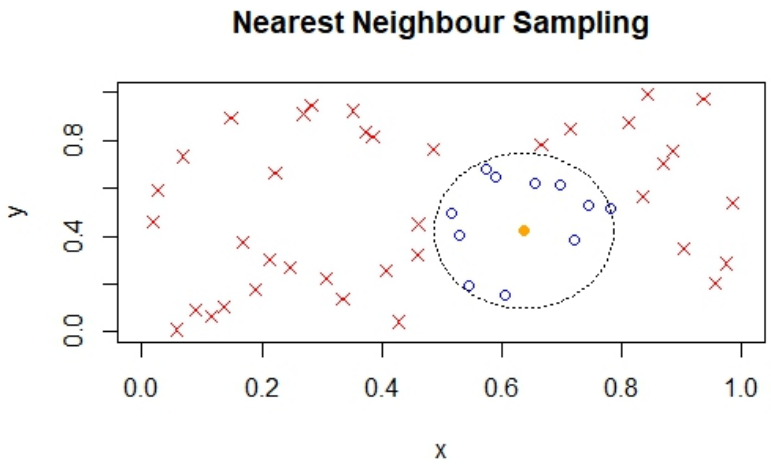

Figure 5: Nearest Neighbor Plot: This plot highlights the nearest neighbor relationship, with blue points identified as the nearest neighbors to the orange points.

of our full dataset.

3.3.1 Sampling Methods for Subsetting Data

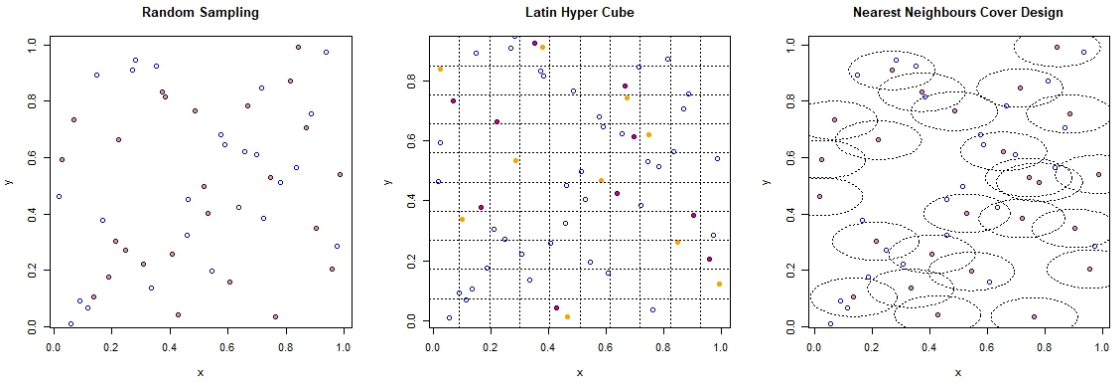

Figure 6: Comparison of Sampling Techniques: from left to right: Random, LHS and Cover Design. Each method offers a different approach to sample selection, impacting the distribution of points within the parameter space

When data is collected, little priori knowledge is available about which model is appropriate and which computer experiments and designs would facilitate diverse modelling methods (Lin and Tang, 2022). Therefore it is appropriate to use designs that represent all portions of the design region. Now the question is posed as to how do we sample a dataset to get the best representation of the design space. Two of the most popular space-filling schemes are latin hypercube sampling (LHS), and cover designs. Both are based on a geometric criterion but offer optimal spread in different senses (Lin and Tang, 2022). LHSs are random, so they disperse in a probabilistic sense, targeting a certain uniformity property. cover designs are more deterministic, and seek spread in terms of distance, by max min discrepancy between points. In

this section, we will briefly look at Random sampling and move on to discuss the LHS and cover designs.

Random Sampling The random sampling method will be considered first, as our trivial subsample to be improved on. Random sampling is a fundamental method in statistics and research that involves selecting a subset of individuals or items from a larger population in such a way that each member of the population has an equal and independent chance of being included in the sample. The key idea behind random sampling is to create a sample that is representative of the entire population, making it a valuable tool for drawing valid inferences and making generalizations about the population based on the characteristics of the sample.

Latin Hypercube sampling Lin and Tang (2022) highlights that the goal of this sampling scheme is to guarantee a certain degree of spread in the design, while otherwise enjoying the properties of a random uniform sample. LHSs accomplish that by dividing the design region evenly into cubes. If the design region is divided into cubes with the same number of observations, there is exactly one observation randomly in each interval (Gramacy, 2020). The observation in each cube is chosen at random. Since the location of the point within the selected cube is random, the LHS does not prevent two sampled points from being located nearby one another, but it does guarantee a spread sample.

A Latin hypercube of n runs for k factors is represented by an n × k matrix In the case of spatial subsampling, there are two factors namely Longitude and Latitude. each column of which is a permutation of n equally spaced levels, with one observation in each row. A LHS design $\mathbf{D}$, is an $n \times k$ matrix with (i, j)th entries being:

$$d_{ij} = \frac{l_{ij} + (n-1)/2 + u_{ij}}{n}, \quad i = 1, \ldots, n, j = 1, \ldots, k$$

where l_{ij} is an entry in an $n \times k$ latin hypercube L and $u_{ij} \sim U(0, 1)$. Latin hypercube designs have exactly one point in each of the n intervals, and this property is referred to as one-dimensional uniformity (Lin and Tang, 2022). The variance of the sample mean under Latin hypercube sampling is smaller than that under simple random sampling. The extent of the variance reduction depends on the extent to which the function considered is additive in the inputs.

LHS gives randomly spaced points and not randomly spaced observations. Therefore, we use a special of LHS that finds the nearest neighbour (NN) to the randomly spaced points in the latin hypercube.

Nearest Neighbour LHS

Nearest Neighbours is an important problem in knowledge discovery and data mining. Distances are measured using any Minkowski L_m distance metrics., defined for any integer $m \geq 1$ the distance between the points in the latin hypercube $l = l_1, ..., l_d$ and $z = z_1, ..., z_d$ observed points is defined as the m^{th} root of

$$\sum_{1 \leq i \leq d} |z_i - l_i|^m \tag{3.2}$$

L_1 and L_2, are known as the manhattan and euclidean distance metrics. The distance between any two points can be computed in $O(d)$ time (Arya et al., 1998). As a result, given any point l_i, the closest observed data point can be calculated. We use this method to update the LHS, and find the nearest neighbour to our random point l_i in z. This method cannot prevent two random points from having the same NN in set z. Therefore negligible amounts of the sample are lost, so the subsample approximates the sample size.

Cover Designs Royle and Nychka (1998) defines he cover design as a method of constructing spatial designs D_z based on a geometric criterion. This sampling method ignores the covariance structure of the process. It is perfect for modelling when there is little prior knowledge of the region. Sampling points are selected to minimize a criterion that is a function of the distance between the sampled locations ($\mathbf{z}$) and the out of sample locations ($\mathbf{u} \in D_z$). This criterion has been coined the "coverage" criterion, and is descibed as how well a sample set "covers" the area of interest. It has become common practice to select inputs to cover the available space as uniformly as possible (Pronzato and Müller, 2012). In the case of a minimax-distance criterion, "covers" is described as the set of points that minimise the maximum distance between points. Essentially minisinng the maximum L_m distance where rewritten in equation 3.3. This equation converges to 0 as point z approaches the Design space for m¿0 and the "coverage" criterion is the average of "coverages" for each point and is defined as $C_{m,M}(\mathcal{D}_z)$ in equation 3.3 when $q > 0$.

$$C_{m,M}(\mathcal{D}_z) = max_{z \in Z} min_{u_i} ||\mathbf{z} - \mathbf{u}_i|| \tag{3.3}$$

Where the subscribt of C denotes a minimax criteria. We aim to find a subset that minimises $C_{m,M}(\mathcal{D}_z)$ as m and M$\rightarrow -\infty$ and ∞ respectfully. The "coverage" converges to a criterion associated with minimax space-filling designs as $C_{-\infty,\infty}$ which is a maximum over minimums. We use the point exchange algorithm. The algorithm uses a random starting configuration, there is no significant relationship between the starting design and the best design. The basic idea behind the algorithm is that if swapping a point in the design with a point in the candidate set reduces the coverage criterion, then the point in the design is moved to the candidate set and the candidate set point is added to the design (Royle and Nychka, 1998). The algorithm can be summarized as follows

Point Swapping Algorithm

1. Compute the distance matrix $\mathbf{D}$ and a vector of its row sums $\mathbf{r}$
2. Choose a starting design $C_{m,M}(\mathcal{D}_z)$ and compute r
3. For each point in the Design set compute N criterion values by iteratively replacing out-of-sample points with $\mathbf{x}$
4. Swap $\mathbf{x}$ with the out-of-sample points that decrease the overall initial criterion.
5. Re-calculate $C_{m,q}(\mathcal{D}_z)$ and r until no further swaps can be made.

Although this algorithm always converges, it does not guarantee an optimal design, so using different starting designs is beneficial (Royle and Nychka, 1998). A large N has two effects on this algorithm, firstly it increases the likelihood of finding a

local minimum. Secondly, this algorithm slows down as N increases, to decrease the run time the candidate set for swapping can be decreased to a set of M nearest neighbours. Depending on the design space this can reduce run time by 50 to 80% at no cost to optimally.

4 Project Implementation

This experiment is based on a case study competition for analyzing Large Spatial Datasets (Heaton et al., 2019). The paper explored approximates of the Gaussian Process that are more computationally efficient at handling big data. The models explored in the case study were variations of Local Approximation methods such as Low rank, sparse covariance methods, sparse precision and algorithmic approaches. The paper ignored Global Approximations, such as the basic subsampling methods. They recorded and compared the MAE, RMSE, Run Time and cores used by each model. In this project. R is used as the main programming language (R Core Team, 2023). Code is available at request

4.1 The Dataset

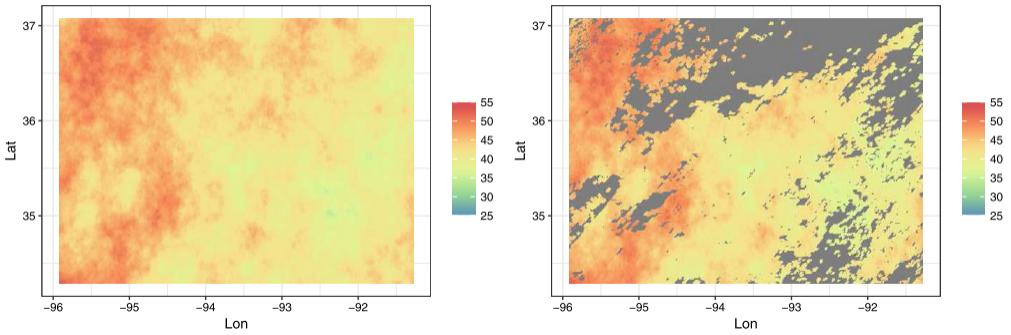

Figure 7: Simulated Surface Temperature : From left to right the full and training simulated data.

The dataset is simulated from a real dataset consisting of 150000 observations of daytime land surface temperatures measured by the Terra instrument on the MODIS satellite on 4 August 2016 (Heaton et al., 2019). They were observed on a 500×300 grid, with longitude values ranging from -95.91 to -91.28 and latitude values ranging from 34.29 to 37.07. This simulated dataset was created by fitting a Gaussian Process on a sample of 2500, with a constant mean and exponential covariance function and nugget effect to obtain 150 0000 simulated points. The resulting parameter estimates for the range, spatial variance, nugget and constant mean are 1.3, 16.40, 0.05 and 44.49 respectively (Heaton et al., 2019). The analysis is done on a subset of 10 000, and a subsample of 1000 referred to as the pilot study. All datasets excluding the 1000 subset are available at `https://github.com/finnlindgren/heatoncomparison`.

4.2 Tools

We will use Global and Local Approximations to approximate the full dataset. For each approximation model, we draw subsamples of size $n = 100, 200, ..., 600$ for our pilot study, and $n = 1000, 2000, ..., 4000$ for the larger sample 1000 times to obtain a distribution of results. In each run, we took sample n and used it to predict the response at each point. Each method was compared in terms of the root mean squared error (RMSE), mean absolute error (MAE), and run time in seconds for

sampling, training, and prediction of the model. Ordinary Kriging, was used to make predictions at a point. The approximation models we use are listed below:

- Global Approximations
    1. Random Sampling: Selecting each observation with an equal probability
    2. Nearest Neighbour Latin Hypercube: Selecting points that cover the design space, by randomly selecting one observation in a grid
    3. Cover Design: selecting a subsample that minimises a set of criteria.
- Local Approximation
    1. LaGP: selected a set of nearest neighbours to predict a point, where 20% of the sample is used as nearest neighbours.

The package `gstat`, written by Pebesma (2006) provides functionality for univariate and multivariate geostatistical analysis. It is used in conjunction with the package `sp` (Pebesma and Bivand, 2005), as it provides classes for defining spatial data. This allows us to fit the kriging models and variograms and predict global approximations. In contrast, we use `laGP`, a package developed by Gramacy (2016) for our local approximations. It was developed specifically to solve the big-N problem in kriging. The methods for local approximation leverage parallel computing. It can be used to model, predict, and estimate the hyperparameters.

4.2.1 Machinery

Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: hpc.uct.ac.za. For the sake of uniformity, all samples were computed using 20 cores and OpenMPI for parallelization.

4.3 Pilot Study Results

The Pillot Study was conducted on a sample size of 1000. The model was trained on 708 observations. From these 708 observations samples of [1, 2, ..., 6]$\times$100 were drawn without replacement and used to train the model and predict the 292 test set.

4.3.1 Hyperparameter Selection

Global Approx: Figure 8 shows 3 possible Covariance Functions. The points are the true covariance structure, while the solid line shows the fitted semivariance given the estimated model parameters for the training set, summarised in Table 1. To describe the spatial variation of the points in question, we compared three

Covariance Function	Nugget	Sill	Range	Kappa
Matérn: $\nu = 1.5$	0.549	6.909	0.105	1.5
Matérn: $\nu = 2.5$,	0.621	6.46	0.073	2.5
Exponential	0	11.142	0.415	0

Table 1: Variogram in Figure 8 estimated Parameter values

semi-variance models on the basis that we have no prior knowledge of the underlying

model. It is visually evident that the covariance function that best estimates the true structure is the Exponential Kernel. Therefore we choose our model parameters to be nugget = 0, Sill = 11.142, Range = 0.415 and no smoothing parameter for our globalapproximations.

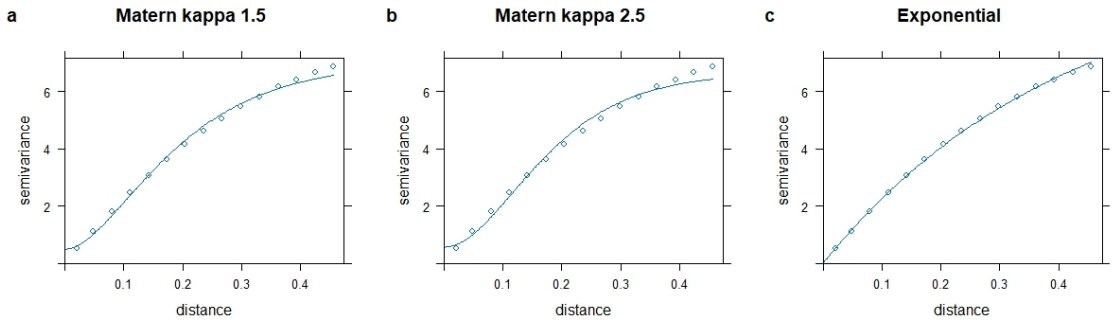

Figure 8: Variogram Comparison: (a) Matérn: $\nu = 1.5$, (b) Matérn $\nu = 2.5$, and (c) Exponential Semivariogram plots. A visual representation showcasing the different characteristics and behaviors of these variogram models for hyperparameter selection

Local Approximate Gaussian Process We use built-in functions in the `laGP` package to estimate the nugget and range from informative priors. The functions generate priors and initial values that control the hyperparameters. It is important to note that, unlike the `gstat` package, laGP estimates a covariance function and hyperparameter for each neighbourhood. This is not a necessary step but it is essential in establishing stable behaviour. The range is derived from a Gamma prior with a shape = 3/2 and a scale = 0.576, on the other hand, the nugget estimation is derived from a Gamma prior with shape 3/2 and scale = 2.164. The prior functions are plotted in Figure 9

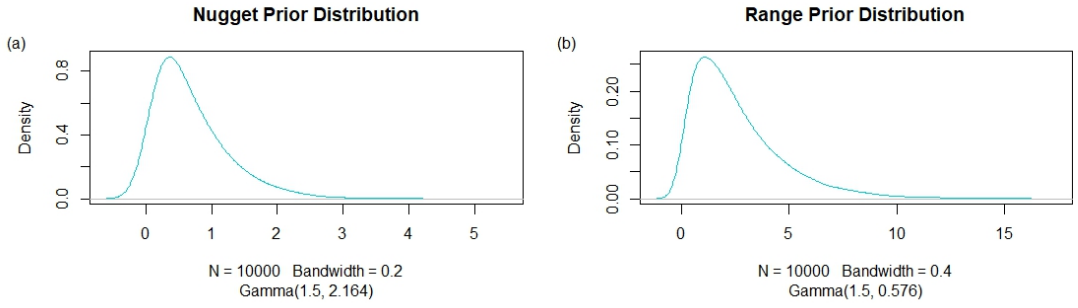

Figure 9: Prior Distributions for the LaGP

4.3.2 Results

Comparing Global and Local Methods: First, we compare the performance of the approximation methods, and the aggregated results are summarised in Table 2 ,3 and 4. In terms of prediction accuracy, The Cover models have significantly less prediction error in plots (a) and (b) in Figure 10. Meanwhile, the Random and

Sampling method	Random		LHS		Cover design		laGP	
Subsample	mean	sd	mean	sd	mean	sd	mean	sd
100	1.234	0.084	1.221	0.083	1.094	0.055	1.993	0.221
200	1.024	0.04	1.025	0.054	0.926	0.027	1.621	0.086
300	0.924	0.029	0.935	0.040	0.839	0.017	1.518	0.049
400	0.867	0.024	0.883	0.033	0.789	0.012	1.47	0.035
500	0.825	0.19	0.850	0.028	0.775	0.008	1.448	0.026
600	0.793	0.017	0.829	0.025	0.766	0.006	1.424	0.021

Table 2: Pilot study RMSE summary results. A table summarising the distribution of RMSE from each sampling method

Sampling method	Random		LHS		Cover design		laGP	
Subsample	mean	sd	mean	sd	mean	sd	mean	sd
100	0.111	0.084	0.126	0.091	0.083	0.063	0.246	0.184
200	0.062	0.048	0.070	0.052	0.036	0.028	0.148	0.107
300	0.044	0.034	0.05	0.037	0.022	0.016	0.098	0.078
400	0.038	0.029	0.040	0.030	0.015	0.011	0.089	0.066
500	0.032	0.024	0.033	0.025	0.009	0.007	0.073	0.053
600	0.029	0.021	0.030	0.023	0.031	0.007	0.062	0.044

Table 3: Pilot study MAE summary results. A table summarising the distribution of MAE from each sampling method

LHS methods are very similar in terms of prediction error; however, as the sample size approaches the superset, the random sample marginally outperforms the LHS. A question that arises is whether this gap is sufficiently significant to differentiate between the two models. The local approximation (LA) method had the highest average RMSE across all samples and a smoother slope. An increase in the number of NN does not seem to affect the quality of the prediction. Figure 10 highlights the trends in the average results. The plot for the run time (c) was not unexpected. The LA model grows fast and exponentially but still performs poorly with regard to the prediction error. The time function for the LHS and the Cover design are both increasing functions, while the time taken for the random sample is exponentially increasing. It takes about as much time as the LA runtime. This is an unexpected and interesting result, as the other two methods are algorithms with many bells and whistles, as they require time to find the nearest neighbours and swap points around. It is indeed peculiar. The models all take on average less than one second to subsample, fit and predict, this is a desirable result.

Sampling method	Random		LHS		Cover design		laGP	
Sample size	mean	sd	mean	sd	mean	sd	mean	sd
100	0.052	0.126	0.13	0.086	0.243	0.064	0.021	0.046
200	0.129	0.127	0.02	0.096	0.274	0.065	0.087	0.045
300	0.262	0.132	0.028	0.089	0.31	0.063	0.208	0.048
400	0.442	0.129	0.037	0.102	0.372	0.063	0.401	0.049
500	0.690	0.134	0.047	0.102	0.425	0.022	0.641	0.048
600	1.023	0.136	0.057	0.103	0.485	0.024	1.002	0.057

Table 4: Pilot study run time summary results. A table summarising the distribution of run time from each sampling method

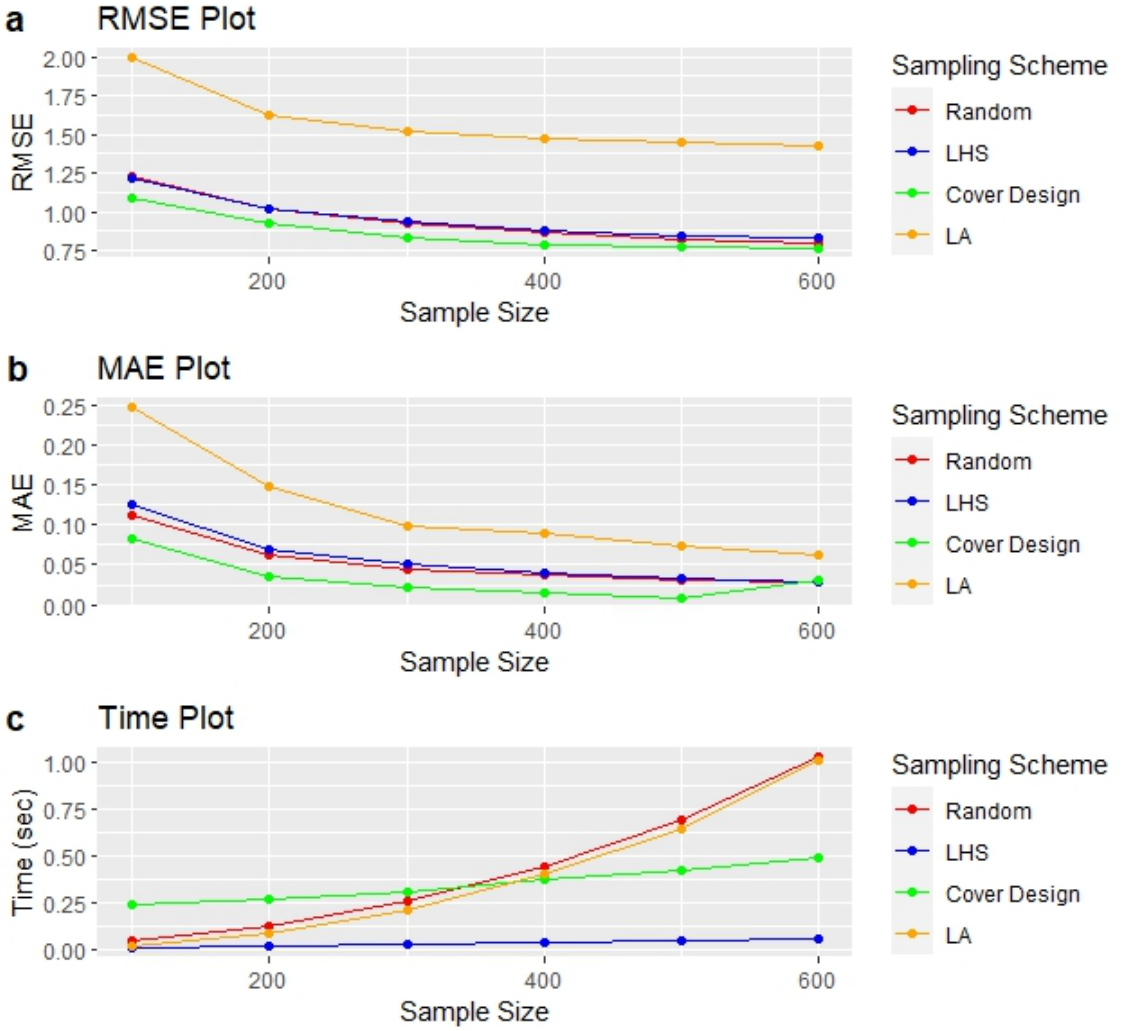

Figure 10: Comparison of Average Results: The above plots provide a comparison of the average performance measures for each 1000 simulations for their respective sample. From top down: (a) RMSE, (b) MAE, (c) Computing Time

Assessing Prediction error (RMSE) The difference between the errors for the global approximations was small; therefore, we assessed whether the gap between the RMSE distributions was significant. To achieve this, the values of all RMSE for each sampling scheme were aggregated, and we conducted two tests: the Q-Q plot and the Kolmogorov-Smirnov (KS) test, to determine if there was a significant difference.

Firstly, we plot a Quantile-Quantile (QQ) plot to assess whether the three sampling schemes are approximately the same or have similar shapes. We see that the left plot in Figure 11 follows closely to the straight line through the origin. There is a slight derivation at the tails, evidence that there are differences in the tail behaviour. Based on the plot alone we can say that LHS and Random Sample produce similar results. On the other hand, the distribution between the random sample and the cover design clearly deviates from the straight line. To support our claims, we performed the KS

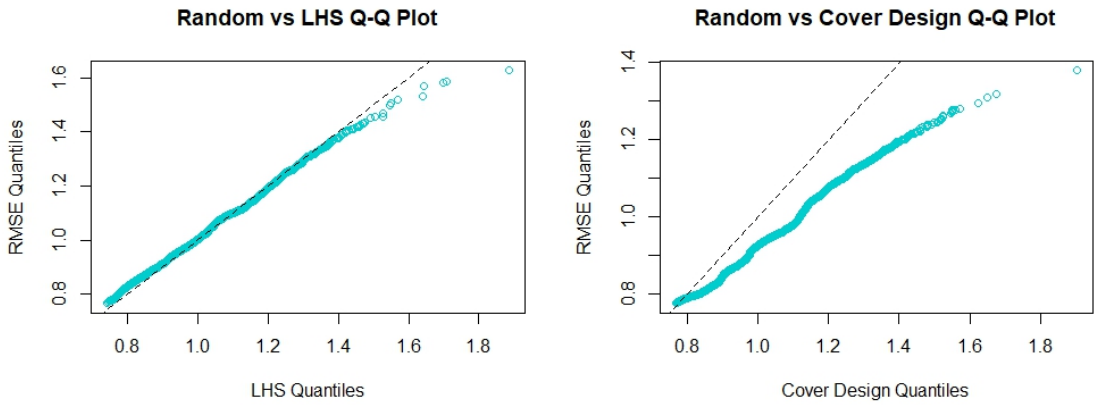

Figure 11: Enter Caption

Test	Alternative Hypothesis	p-value	Conclusion
Two sided	The random sample and LHS sample come from the same distributions	0	Reject the null Hypothesis
Two sided	The Random Sample and Cover Design are from the same distribution	0	Reject the null hypothesis

Table 5: Kolmogorov-Smirnov Tests

test to compare the two samples via a statistical test. The null hypothesis states that two samples follow the same distribution. This test measures the maximum absolute difference between the cumulative distribution functions of two samples We perform this test with three alternative hypotheses, and the results are summarised in Table 5. For both tests we reject the null hypothesis and conclude that they are all sampled from the same distribution. This conflicts with the conclusion drawn from the QQ plot

Sensitivity to Covariance Function: Secondly, we assess the global result's sensitivity to the choice of Covariance Function. The figures to the left in Figure 12 are the results of the Global Approaches in Figure 10 and to the right we have the same models but using the estimates provided for a Matérn Kernel. It is obvious that the plots are very similar in terms of trend and scale when we look at RMSE and MAE. A noticeable dissimilarity is that the Cover Design subsample becomes increasingly unstable as it draws near to the sample size. This is demonstrated in the MAE plot (d), where the matérn covariance function error no longer decreases smoothly but starts fluctuating, this is also the case for the exponential covariance function, although this fluctuation occurs later. All methods exhibit a decreasing RMSE as the sample size increases. The RMSE for the LHS and random sample are almost indistinguishable, but as the sample size approaches the superset, the random sample marginally outperforms the LHS. This plot raises the question of whether this distinction is significant and worth investigating. The training of the cover design with the matérn function was more time-consuming than expected, due to the complexity of the model and the slower processing speed of the Matérn function. Despite this, the cover design produced better results with a longer runtime. Although the Left-Hand Side (LHS) reduced computing time, it resulted in

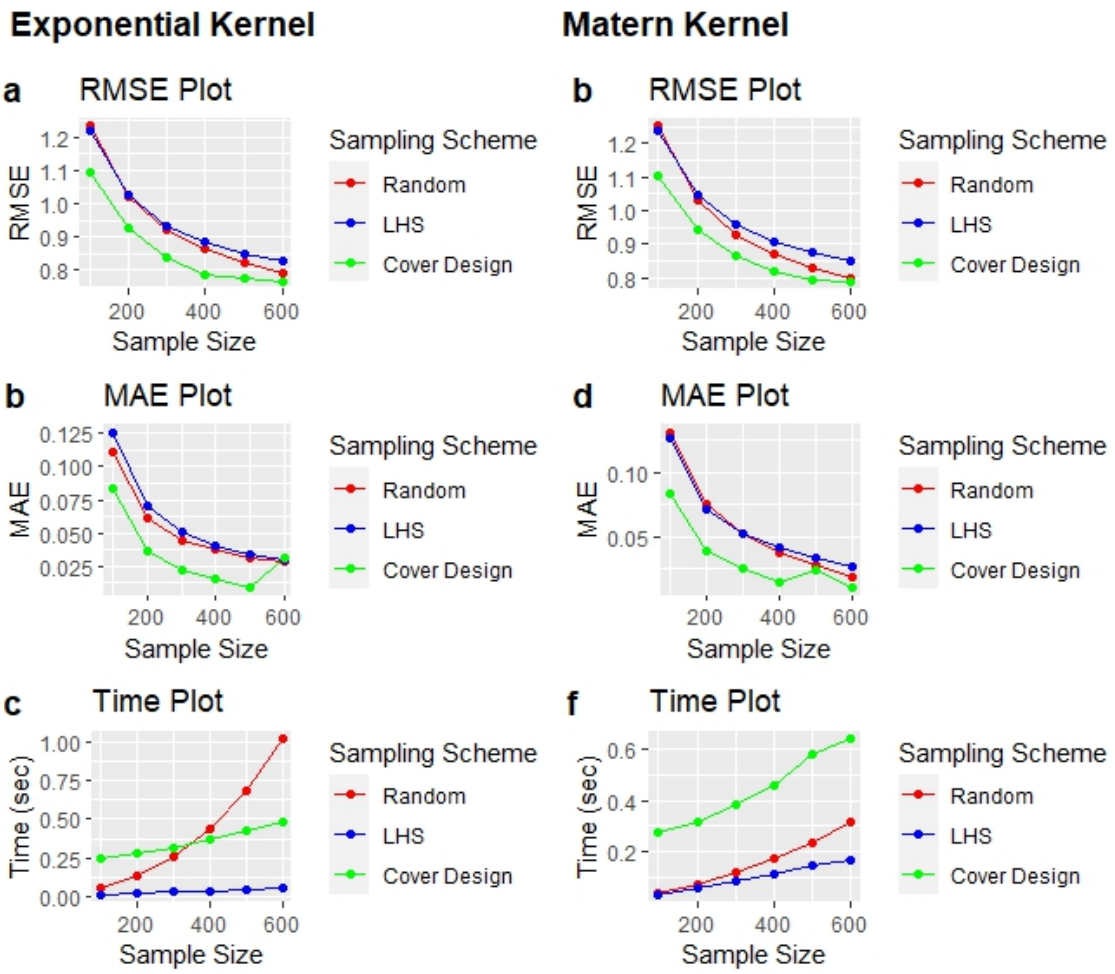

Figure 12: This plot compares the previous results of the Global Approximations estimated using the Exponential Kernel versus using the Matern Kernal, to address the Kriging's sensitivity to its Kernel

Subsampling method	Random sample		LHS		laGP	
subsample size	mean	sd	mean	sd	mean	sd
1000	0.02	0.015	0.027	0.02	0.049	0.038
2000	0.012	0.009	0.027	0.02	0.029	0.022
3000	0.009	0.007	0.28	0.017	0.022	0.017
4000	0.006	0.004	0.28	0.16	0.017	0.013

Table 6: Larger Dataset MAE summary results. A table summarising the distribution of MAE from each sampling method

Subsampling method	Random sample		LHS		laGP	
subsample size	mean	sd	mean	sd	mean	sd
1000	0.719	0.012	0.731	0.024	1.293	0.014
2000	0.639	0.007	0.654	0.020	1.146	0.075
3000	0.600	0.006	0.616	0.017	0.954	0.041
4000	0.575	0.005	0.594	0.015	0.900	0.017

Table 7: Larger Dataset RMSE summary results. A table summarising the distribution of RMSE from each sampling method

less accurate predictions.

4.4 Larger Sample Results

GPs are still relatively fast for smaller datasets, With 20 cores we can reach speeds of up to a tenth of a second, as the big N problem comes into effect at $n > 2000$. To assess whether the remedies introduced in Section 3 can properly represent our super set, the same analysis in the pilot study was conducted on a sample of 10000. We run simulations with $1000, 2000, 3000$ and 4000 subsample sizes for the global Approximations. For the Local Approximations, we sample 1000, 2000, 3000 and 4000 and use $10-20\%$ of the sample size as our nearest neighbour sample to approximate the local covariance functions. We use the same parameters in section 4.3.1 for our models. In this section, the only global approximations that are considered are the random sample and LHS as the cover design was terminated prematurely due to the model's extended runtime. Tables 6, 7 and 8, sumarise the results. The MAE standard deviation in Figure 13 appear large, but are actually very small ranges acording to the Table 6. The MAE results are all very similar and overlap.

Comparing Models The figure displayed in Figure 13 depicts the aggregated results and standard deviation of the larger sample, with the Root Mean Squared Error (RMSE) of the Local Approximation being higher than that of the Random

Subsampling method	Random sample		LHS		laGP	
subsample size	mean	sd	mean	sd	mean	sd
1000	3.499	0.230	0.621	0.099	27.724	0.277
2000	16.173	1.373	1.765	0.118	107.641	2.036
3000	41.174	2.692	3.551	0.214	250.622	12.044
4000	71.663	3.178	5.423	0.332	467.281	34.493

Table 8: Larger Dataset run time summary results. A table summarising the distribution of run time from each sampling method

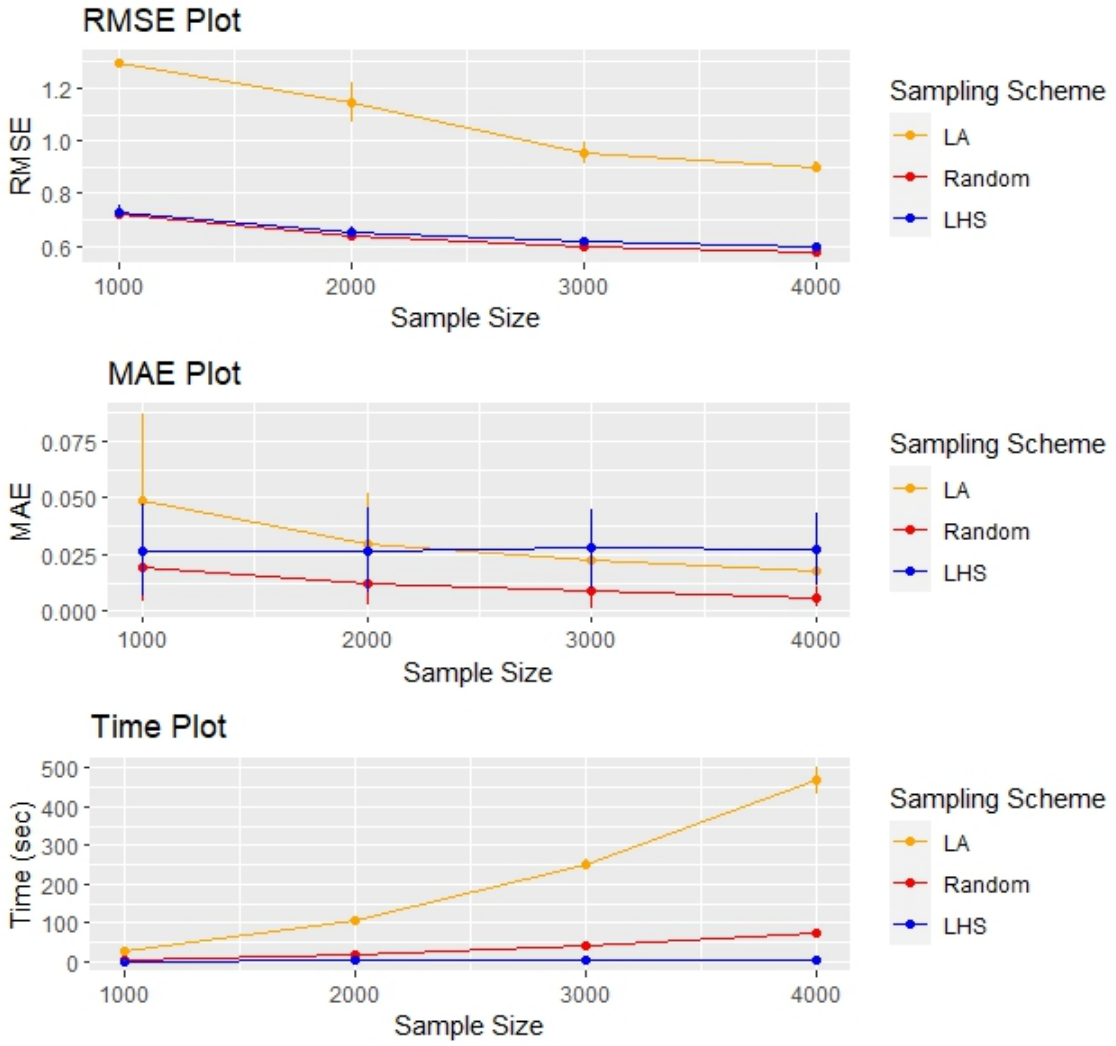

Figure 13: Results of the 10 000: From left to right: RMSE, MAE and Time plot

Sample and LHS, but decreasing at a faster rate. The LA results also exhibit a higher standard deviation for its smaller samples than the other methods. The Mean Absolute Error (MAE) of the LA has stabilized and is decreasing at a faster rate. The random and LHS models do not exhibit a lot of variation when it comes to their results. Once again the random and LHS have similar results as their pilot samples. The LHS MAE is increasing with the sample size. In terms of time, the random subsample requires an average of less than 100 seconds to compute, while the time it takes to compute the LA is increasing exponentially and so is the variation of the time it takes to run, meanwhile, the LHS computes in seconds.

5 Discussion

5.1 Random vs. LHS vs. cover design

The question of how to select a subsample still stands. The choice of the method depends on the desired outcome. If the primary goal is to minimise runtime, subsampling methods are recommended. They simplify the modelling process and produce reasonable prediction errors. The LHS produced results at efficient speeds, it is also the only method explored in this paper as it does not appear to be affected by the size of n. The LHS has advantageous characteristics as it first selects a reasonable subsample, that covers the design space and secondly, it has linear time complexity. The only downside is that it does not select a sample that is a robust approximation when compared to its competitors. The random sample and LHS were toe-to-toe in terms of RMSE and MAE. In spatial statistics, data exhibit spatial autocorrelation, which means that nearby locations tend to have similar values, and random sampling might not be able to capture structure. Therefore it is expected that LHS would produce better results than the random sample. However, this is not the case. This raises the question of whether the LHS captures that structure. The subsampling method that outperformed all its competitors was the cover design. The caveat with this method is it took long to converge for larger samples. Given that this was a simulation exercise of 1000 repetitions, the cover design failed to keep up. Despite this, it is the most reasonable choice, as it approximated the space well, and produced robust errors.

5.2 laGP vs. gstat

The results of the analysis were surprising, as the literature suggested that laGP's are faster and more robust. The results of this project contradict the expectations presented by the literature. Furthermore, the laGP results were disappointing as it used the most resources but performed poorly, against simpler methods. laGP works well on supercomputers but is computationally inefficient elsewhere. This raises the question of for whom the package was designed, as it appears to be less user-friendly. Generally, it is difficult for beginners to manoeuvre spatial packages, but laGP was exceptionally challenging. It is easier to understand the many configurations for each model given prior knowledge of some basic computer science concepts.

5.3 A note on computation limitations

Two of the methods utilized in this project were computationally too expensive to be feasible on the large dataset, specifically, the cover design, which was unable to keep pace with the substantial increase in data size. Despite laGP being marketed as one of the faster methods available, the authors failed to acknowledge that it was optimized for performance on supercomputers. The maximum average runtime in this project for the laGP is 467 seconds processed in parallel on 20 cores. It was a nightmare to run on an ordinary computer. The same goes for the cover design, however, no data is available to estimate how long it took on average to sample. However, a contributor to the model's prolonged computation time was the inclusion of 1000 simulations, although the sample sizes did not make an insignificant contribution to the run time. Ultimately, the models have demonstrated effective-

ness, and if there are 20 cores, they can be executed instantaneously. Therefore, I would not dismiss them just yet.

6 Conclusion

This project provides an overview of the Gaussian Process and its application in spatial statistics. It then delves into the challenges that arise when using kriging models with large datasets and the remedies that have been developed to address these issues. The LHS performs similarly to a random sample of spatial data, which is limited as it does not take into account spatial autocorrelation. The cover model was the most promising model, but it failed to produce enough results for analysis when we scaled the sample size from 1000 to 10,000. The study also notes that more complex models, such as the Local approximation GP, require more processing time but still result in a higher level of error compared to simpler models. The findings indicate that there are no perfect solutions, and each approach has its own drawbacks.

References

S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

T. Burgess and R. Webster. Optimal interpolation and isarithmic mapping of soil properties: I the semi-variogram and punctual kriging. *Journal of soil science*, 31(2):315–331, 1980.

K. Chalupka, C. K. Williams, and I. Murray. A framework for evaluating approximation methods for gaussian process regression. *Journal of Machine Learning Research*, 14:333–350, 2013.

W. Dai, Y. Song, and D. Wang. A subsampling method for regression problems based on minimum energy criterion. *Technometrics*, 65(2):192–205, 2023.

L. Deldossi and C. Tommasi. Optimal design subsampling from big datasets. *Journal of Quality Technology*, 54(1): 93–101, 2021.

C. C. Drovandi, C. Holmes, J. M. McGree, K. Mengersen, S. Richardson, and E. G. Ryan. Principles of experimental design for big data analysis. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 32 (3):385, 2017.

R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

R. B. Gramacy. laGP: Large-scale spatial modeling via local approximate gaussian processes in R. *Journal of Statistical Software*, 72(1):1–46, 2016. doi: 10.18637/jss.v072.i01.

R. B. Gramacy. *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Chapman Hall/CRC, Boca Raton, Florida, 2020. `http://bobby.gramacy.com/surrogates/`.

J. S. HaiYing Wang, Min Yang. Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association*, 114(525):393–405, 2019. ISSN 0162-1459.

M. J. Heaton, A. Datta, A. O. Finley, R. Furrer, J. Guinness, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D. Hammerling, M. Katzfuss, et al. A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics*, 24:398–425, 2019.

K. Krauth, E. V. Bonilla, K. Cutajar, and M. Filippone. Autogp: Exploring the capabilities and limitations of gaussian process models. *arXiv preprint arXiv:1610.05392*, 2016.

D. G. Krige. A statistical approaches to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52:119–139, 1951.

C. D. Lin and B. Tang. Latin hypercubes and space-filling designs. *arXiv preprint arXiv:2203.06334*, 2022.

H. Liu, Y.-S. Ong, X. Shen, and J. Cai. When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems*, 31(11):4405–4423, 2020.

S. Liu, Y. Wang, and F. Sun. Two-dimensional projection uniformity for space-filling designs. *Canadian Journal of Statistics*, 51(1):293–311, 2023.

M. Z. Ngwenya. Investigating 'optimal' kriging variance estimation: Analytic and bootstrap estimators. Master's thesis, University of Cape Town, South Africa, 2011.

E. Pebesma and R. S. Bivand. S classes and methods for spatial data: the sp package. *R news*, 5(2):9–13, 2005.

E. J. Pebesma. The gstat package. *Á Bwww. gstat. org*, 2006.

L. Pronzato and W. G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701, 2012.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023. URL `https://www.R-project.org/`.

C. E. Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-28650-9. doi: 10.1007/978-3-540-28650-9_4. URL `https://doi.org/10.1007/978-3-540-28650-9_4`.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X.

J. A. Royle and D. Nychka. An algorithm for the construction of spatial coverage designs with implementation in splus. *Computers & Geosciences*, 24(5):479–488, 1998.

T. Santner, B. Williams, B. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. Springer, 2003. ISBN 9780387954202. URL https://books.google.co.za/books?id=01itua2QzFkC.

S. Surjanovic and D. Bingham. Virtual library of simulation experiments. Accessed September 2023, 2013. URL https://www.sfu.ca/~ssurjano/index.html.

C. Williams and C. Rasmussen. Gaussian processes for regression. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL https://proceedings.neurips.cc/paper_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf.

H. Zhang, Y. Zhan, J. Li, C.-Y. Chao, Q. Liu, C. Wang, S. Jia, L. Ma, and P. Biswas. Using kriging incorporated with wind direction to investigate ground-level pm2. 5 concentration. *Science of the Total Environment*, 751: 141813, 2021.