Doctoral Studies in Mathematical Sciences, 2017-2022

Analysis of Computer Experiments with Smooth Emulators

Asma Farid
ID: 170125910

Supervisor: Dr Hugo Maruri-Aguilar

Submitted in partial fulfilment of the requirements of the
Degree of Doctor of Philosophy
November 2022

School of Mathematical Sciences
Queen Mary University of London

Declaration of original work

This declaration is made on November 29, 2022.

I, Asma Farid, confirm that the research included within this thesis is my own work. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Asma Farid
Date: November 29, 2022

Acknowledgement

First and Foremost thanks to the Almighty for His blessings to achieve this milestone. Then, I would like to express my sincerest gratitude to my supervisor Dr Hugo Maruri-Aguilar. This thesis would not exist without all the suggestions, feedback, constant encouragement and patience provided by Dr Hugo throughout the years. I am deeply impressed by his conscientious and dedicated attitude as an educator and researcher. I would also like to thank the funding provider,HEC, to support my education and my stay in London that have been such a fantastic experience. I would like to acknowledge the panel Dr Steve Coad and Dr Jamie Griffin for their constructive feedback at each stage of my progression.

My life would not have been as enjoyable as it was without all the friendly members of SDSS group. I would like to thank each one of them for their support, who were always willing to share their knowledge, expertise as well as happiness and sorrows in daily life. My huge thanks will go to my most amazing friends Fauzia, Shafaq, Tayabba and my fellow researchers. In your own unique way, you guys have offered amazing support and made this time so colourful with your love and laughters.

Finally, I would like to thank my family; my loving husband, Kamran and my beautiful children Ayaan and Aiza for their ongoing love and support throughout my entire research period.

Above all I would like to thank my father, Muhammad Farid. I don't believe anyone has offered more guidance, pride and love than you, throughout my life. Without your continuous support and unconditional love this would not have been possible.

Abstract

This thesis is concerned with the methodology of smooth emulators for computer experiments. The work in this thesis is categorized into two parts. The first and the main part comprises methods with smooth emulators for the analysis of computer experiments. The second part is devoted to the extension of optimal designs for the smooth emulator. The methodology of the first part is primarily focussed on the problem of less than full rank design model matrix, and we combine the elements of ridge regression with a measure of smoothness to develop an improved emulator. Our analytical results show that mean square error of the parameter of smooth emulator improves over that for the standard regression parameter. We also extend the smooth emulator with the Gaussian process and compare the performance with the Gaussian emulator itself. We have applied our methods to model COVID transmission data as well as to data from simulated climate models. We have concluded from the analytical results and simulated studies that the smooth interpolator outperforms other emulators under certain given conditions which are described in the work. We conclude from the results that smooth emulator is useful particularly in rank deficient design model matrix. In addition, the smooth emulator provides a simple and cheap alternative in situations where Gaussian emulator is not a viable choice. In the second part of this work, we perform a detailed exploratory analysis to find some of the optimal designs for smooth interpolator that provides valuable insights into the properties of the optimal designs for smooth emulator. We also develop analytical results for D- and A-optimal designs for ridge regression not found in literature before.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Historically, Statistics has been the scientific discipline to create procedures and methods to carry out empirical research. This covers a wide variety of areas from sampling methods of data collection to estimation and analysis. Agricultural field experiments were the first to use designed experiments after the breakthrough work of [26] in the field of design of experiments. This section is motivated by the Introduction of Physical and Computer Experiments found in [69] and a summary of the same is given here.

Over the time many other areas and fields have been developed in this subject matter. For example, controlled clinical trials, which is a variant of designed experiments, have been extensively used in medical field. Similarly, simulation experiments are widely employed in industry, operations research, business studies, environment etc. to gain better understanding of physical phenomena under study. Unfortunately, the traditional techniques of physical experiments are not valid in case of computer experiments owing to the fact that physical experiments are accompanied by real life variations in variables and response. In order to increase the validity of physical experiments, some well known techniques are developed by Statisticians. The first among them is *randomization* which suggest to apply the experimental treatment randomly so as to prevent the nuisance variables from systemically affecting the response. The second method is the *blocking* which is the method of dividing the experimental material into homogeneous groups so that it may not supersede the treatment effect on response. Lastly, the phenomenon of *replication* is introduced which make the experiment to run on large scale. This minimizes the measurement variation across treatments.

In contrast to physical experiments, computer experiments may be deterministic and for a given set of input variables, the code run twice produces the same response. Moreover, the codes involved in a computer experiment may be time-consuming. Also, modelling of physical

processes is a complex activity owing to the fact that in any such process a large number of variables may involve along with uncontrolled variations that need to be taken into account. Therefore, efforts are being made to design efficient computer experiments that are aimed to model Physical phenomena in order to obtain optimal outputs and efficient predictions. One of the pioneer works in the domain of computer experiments dates back to 1955 with the publication of report [25]. In this work the first supercomputer MANIAC was used to develop a problem solving tool to virtually zoom in on a system and observe atomistic interactions at the molecular level, with a realism that was not possible before. It was a breakthrough work with the use of computer simulations for the first time. Computer models are the models that attempt to simulate a physical phenomenon under study. Computer experiments are preferred over physical experiments in following situations; large number of input variables, economic constraints to gather information on a particular research question or the ethical reasons. However, many complications arise in an effort to design an efficient model, partly because of the deterministic nature of computer experiments, i.e. absence of random error and partly because of the complexity of the underlying problem. In order to gain a better understanding of computer experiments, it is inevitable to distinguish three types of variables that can affect a computer code and hence the output, depending upon the phenomenon under study. A detailed note on these variables can be found in [69], a brief introduction of which is given here.

The first type of variable is called controlled variable. Control variables are those variables that are fixed by the experimenter e.g. by an engineer or scientist in order to control the process. For example, amount of different chemicals in manufacturing of metal sheets.

The second type of variables are called environmental variables. These are the variables that affect the output and subject to change with specific users or conditions. e.g. in production of metal sheets, the humidity, temperature, light intensity may be considered as environmental variables that are subject to change at different points of time. These variables are also called noise variables.

The third type of variables are called model variables and they are associated with the uncertainty in mathematical modelling.

We provide a brief account of some of the examples details of which are provided in [69]. Examples of scientific and technological developments that have been made with the help of computer experiments are vast and rapidly growing. They have been employed to predict climate and weather, the performance of integrated circuits, the behaviour of controlled nuclear processes, fire studies, stresses in prosthetic devices, clinical trials etc.

A mathematical model for the evolution of fire in a single room with closed doors and windows

is developed by [16], where an object is ignited at some point below the ceiling. The input variables to this model include, the room floor area, ceiling height, height of burning object from the floor and heat release rate. The outputs are the temperature of the hot smoke layer and its distance as a function of time above the fire source. Since those initial attempt to model fire events, a vast amount of material on computer codes have been emerged to model wildfire evolutions as well as fires in confined places.
In an auto-mobile industry, [48], designed a computer model to determine the failure depth of the symmetrical rectangular blocks. This is an important application, since the sheet metal formed in this manner is utilized in many parts of the automotive industry.
Another example of the significance of computer experiments in clinical trials can be found in [14], that studied the design and analysis of total joint replacement with the aid of computer experiments. In an attempt to find the alternatives of the existing design in which two of the dimensions were allowed to vary, this model also had three patient specific variables, significant for structural response. A project is described by[5], where engineering specification require 31 design variables to optimally design helicopter blade. The objective function was a measure of the rotor vibration that combined the forces and moments on the rotor, each of these quantities were calculated by the computer code.
One of the applications of computer code is found in [40], where the authors highlighted the role of computer experiments in public policy. The model, they used, quantifies the state of the future over a window of 100 years, using several measures, one of which is Human development Index. There are 30 input variables and public policies are specified by some of the inputs.

1.0.1 Surrogate models for computer experiments

Computer simulations can serve to solve a wide range of problems including optimization, prediction, exploration of design space, uncertainty analysis and sensitivity analysis. An attempt to data handling and model building amid computer simulations often result in expensive computations. Consequently, the need had risen to replace the expensive computer simulations with alternative cheap surrogates. The surrogate modelling provides a framework to find an approximate function that may mimic the behaviour of the original system under study. Surrogates are constructed based on data driven bottom up approach. In other words, surrogates can be constructed for any process without any prior knowledge on the data generating mechanism. The problem of constructing the surrogates can be broadly classified into two categories [65]

1. Analysis problem: How to make effective use of data to develop a surrogate model that

results into accurate predictions at untried points.

2. Design problem: How to select design points to meet certain objectives of the study.

Many surrogate models have been developed and efficiently applied in lieu of simulation models. Some of the notable ones include: polynomial response surface models [6][50], Kriging in computer experiments [65], Radial basis functions [56] [9], neural networks [44] and support vector machines [18]. Surrogates may also be referred as meta-models, model emulation or proxy models. A review of surrogate models is provided by [60], where a detailed account of surrogate models is furnished.

Our prime focus in this thesis is to build a surrogate model that lays its foundations on polynomials and Gaussian processes. The use of Gaussian process as a surrogate in computer experiments is introduced in the seminal paper of Sacks [65]. The methodology of Gaussian process to make predictions at untried locations has its roots in spatial statistics and known as Kriging [37]. An excellent book in this reference is [76]. This book summarizes the previous work on the prediction of the random field based on observations at some locations of the random field and also introduce some new methods of kriging. A valuable resource to understand Gaussian regression within machine learning framework is [57]. It is vital to define here the terms simulator and emulator within the context of computer experiments that will be used in this thesis. In order to simulate the behaviour of real world systems complex models are developed that encompasses the detailed knowledge of the real world process. These models are in the form of equations which are implemented in computer programs. Such mathematical model and the computer program that implements it is referred as a simulator [53]. An emulator is the statistical approximation to the simulator. In other words simulator is a true function that maps the inputs to the output and emulator approximates this mapping. We say that the Gaussian Process surrogate model is the Gaussian process emulator that attempts to emulate the true model. Once an emulator is built for the computer model, different analysis can be done one of which includes Sensitivity Analysis. Sensitivity analysis is defined as the study of sensitivity of model outputs to variation in model inputs [53]. A good emulator is expected to produce efficiently close approximations of Sensitivity Analysis to that of the simulator. An emulator can also act as an efficient device to quantify uncertainty in the model output. Some of the sources of uncertainty include, random or unknown input, model specification and code uncertainty [53].

1.1 Motivation

The Gaussian surrogates assume the correlation structure among the observations which makes such models computationally expensive. We want to build a model without the need of the assumption of correlation among the observations. In other words we want to find methods to build an emulator without the Gaussian process. In addition the Gaussian surrogates in literature are developed for full rank design model matrix. We aim to build a flexible model which produces predictions both for full rank and less than full rank design model matrix. One of such motivation is provided by Smooth Supersaturated models [4]. This model is built with the introduction of a measure of Smoothness, the matrix K and solving a constraint problem details of which are given in Chapter (3). In order to give a flavour of this thesis we introduce a simple example. The notations used here will be followed by the explanation in the forthcoming chapters.

Consider a design $\mathcal{D} = \{0,\ \pm 1\}$ and the respective response vector $y = \{y_{-1}, y_0, y_1\}$. We discuss three cases which will built a foundation of this thesis.

1.1.1 Full rank design model matrix

Let $M = \{1, x\}$ be the basis to construct model terms. Then the design model matrix F is full rank and we can use simple linear regression to find the model parameters β_0 and β_1.

$$F = \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad F^T F = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}$$

The parameter vector β is straightforward to compute which gives $\beta_0 = \bar{y}$ and $\beta_1 = 1/2(y_1 - y_{-1})$.

1.1.2 Less than full rank design model matrix

Let $M = \{1, x, x^2, x^3\}$ which is supersaturated basis. The design model matrix for given basis is less than full rank.

$$F = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad F^T F = \begin{pmatrix} 3 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \end{pmatrix}$$

We present the matrix K introduced in [4] which is constructed from the measure of roughness and given as,

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 \\ 0 & 0 & 6 & 12 \end{pmatrix}$$

We introduce a model termed Smooth Ridge model which is formulated with the supersaturated basis and the matrix K. The detailed methodology of the model construction is delineated in Chapter (3). We present here the parameter vector of the proposed model which takes the form $\tilde{\beta}_\lambda = (F^T F + \lambda K)^{-1} F^T Y$. With the given design, basis and design model matrix it is straightforward to compute $\tilde{\beta}_\lambda$. It is interesting to note that as λ increases the parameter estimate associated to linear basis approaches to that of linear regression while the parameter estimate of non-linear basis goes to zero. For the given example we can see that,

$$\lim_{\lambda \to \infty} \tilde{\beta}_\lambda = \begin{pmatrix} \bar{y} \\ 1/2(y_1 - y_{-1}) \\ 0 \\ 0 \end{pmatrix}$$

1.1.3 Objectives

The core objectives of the work in this thesis are to:

1. Develop a methodology to build a smooth emulator for computer experiments particularly for rank deficient design model matrix.
2. Analyse and compare the proposed model with existing Gaussian Process emulators.
3. Identify the scope and limitations of the models under study.
4. Understanding the relationship among the response and the choice of design points.
5. Exploratory analysis of optimal designs for the smooth model.

1.2 Outline of the thesis

This thesis is composed of six chapters. Followed by the first chapter of Introduction, the second chapter exhibits a detailed review of literature that encompasses Gaussian Process emulator

in great depth. We begin with the introduction of notations in Section (2.1) to be used in the rest of the thesis. A detailed analysis of computer model [65] is furnished in Section (2.2) with the application of Markov Chain Mote Carlo to the same model in Section (2.3). An extension of Gaussian Process to Convex combination of Gaussian processes [29] is discussed in Section (2.4). The use of Gaussian Process to build an emulator for calibrated model was introduced by [34], an overview of which is presented in Section (2.5). A brief introduction of Neural networks, a widely adopted surrogate model, is given in Section (2.6).

In Chapter three we introduce the methodology to build Smooth Emulator. In Section (3.1) we review the methodology that provides a framework of Smooth Emulator. In Section (3.2) the Smooth Emulator named Smooth Ridge model is developed. The comparison of the variance of Smooth ridge estimator with that of regression estimator is given in Section3.3). Mean square error of Smooth Ridge estimator along with the Mean Square Error of model predictions is explained in Section (3.4) and (3.5) respectively. Comparisons of Smooth Ridge model with the existing models are carried out in Section (3.6) for simulated data and in Section (3.7) for COVID-19 transmission data. Introduction of Splines and their comparison with the proposed methodology is furnished in Section (3.8) and Section (3.9). The methodology of Sensitive Analysis for Smooth Ridge model is explained in Section (3.10) and construction of Sensitivity Indices is explained in Section (3.11). The Bayesian formulation of Smooth Ridge model is introduced in Section (3.12) followed by the final conclusions in Section (3.13).

Chapter four narrates the detailed comparisons of Smooth Ridge model and contemporary models for Bratley function and Levy function as simulators. Different choice of parameters, design size and dimensions are considered to evaluate the performance of Smooth Emulator in Sections (4.1-4.5). COVID data modelling is revisited in Section (4.6) and temperature data is employed for comparisons in Section (4.7). Sensitivity indices are computed for the COVID data and temperature data in Section (4.8).

Chapter five addresses the phenomenon of design selection for Smooth Ridge model. Section (5.1) describes some of the space filling designs followed by the overview of some of the criterion based designs in Section (5.2) and Section (5.3). The methodology to construct D- and A-optimal designs with two design points and replications is explained in Section (5.4). The analytical results for D- and A-optimal designs for Ridge regression are developed in Section (5.5). An exploratory analysis of designs based on prediction variance is expounded in Section (5.6).

The thesis is concluded with discussions and future recommendations in Chapter (6).

Chapter 2

Models from Literature

A comprehensive literature review in the field of computer experiments along with the exploration of related domains is provided in this chapter. The purpose is to get a better insight of models pertaining to computer experiments and understand in detail the underlying methodology thereof. Most of the theoretical developments relevant to the reviewed literature are reproduced along with the simulation results. One of the pioneer work in the realm of computer experiments was [65]. This work builds an emulator by considering the deterministic output as a realization of stochastic process. This leads to introduce a bias correction term along with the trend in conventional regression problems. The emulator is build using elements of kriging method introduced by[43] where a predictor is defined in terms of linear combination of observed data. It is shown that predictions at untried points can be made with estimate of uncertainty. The concept of obtaining surrogate model with a combination of two Gaussian processes is explained by [29]. Their work addresses the situation when observations are supposed to be realizations of mixture of two independent Gaussian processes. The results are provided for one and two dimension data which suggest that convex combination of two Gaussian processes can improve predictions, particularly when the input region shows non homogeneous behaviour. The study of calibration parameters or context specific inputs for computer experiments was carried out by [34], in which a framework is provided for the model and analysis of data when true values of calibrated inputs are unknown. The approach used towards estimating and tuning of parameters in this study is based on Bayesian analysis and posterior distribution is obtained on calibration parameter to fit the model to the observed data. This study on calibration parameter and underlying methodology is further investigated by [78] in which the authors commented on unstable performance of [34] in calibration. They presented two frameworks, the one where physical observations are assumed to be non random

and a more realistic approach where they are considered random. Two computer models are developed namely, cheap computer simulations and expensive computer simulations which, supported by theoretical studies, show that the predictions made are not dependent heavily on the choice of priors contrary to *Kennedy'O Hagan* model [34]. Moreover, they provided asymptotic convergence of predictive distribution given by *Kennedy'O Hagan*. One of the many variants of calibration parameter is latent variable involved in generating the response. Deep Gaussian processes had caught an appreciable attention in recent years owing to its flexibility to capture latent structure present in the model. One of the work in the fast growing field is developed by [19]. In this work, the response is considered to be a result of multivariate Gaussian process, whereby, inputs are also governed by some Gaussian process. Therefore, there is structure of hidden and visible layers with nodes in each layer. In order to marginalize out the latent variables and obtain predictions, [19] explored the theory of variational inference and therefore defined a lower bound on the marginal likelihood of the data. It is shown that marginalising out the latent space with lower bound is very effective in learning the hierarchical features and abstract information even for smaller data sets. A comprehensive review of Computer experiments is accounted in [38]. Neural Networks have emerged as one of the most popular surrogate to computer simulations in recent times introduced by [44] in 1943.

The chapter is structured as follows: A comprehensive introduction of notations to be used in the rest of the document is given in Section (2.1). Detailed review of *Sacks* model [65] is accounted in Section (2.2) with implementation of Bayesian approach to the same model in Section (2.3). Convex combination of Gaussian process [29] is explicitly reviewed in Section (2.4). The detailed methodology of *Kennedy'O Hagan* model [34] is furnished in Section (2.5) followed by description of Neural Networks in Section (2.6).

2.1 Notation

The notation explained in this section will be followed in rest of the document. Consider a study with d-dimensional vector of input variables $\mathbf{x} = (x_j | j = 1, 2 \ldots) \in \mathbb{R}^d$. consider a design $\mathscr{D}$ be the set of n points taken from $\chi \subset \mathbb{R}^d$ such that $\mathscr{D} = \{\mathbf{s_i} | i = 1, 2 \ldots n\}$ which constitute the sample set of size n. While referring to the individual entries of each design point, the first sub-index will refer to the design point and the rest of the sub-indices to the entries, e.g. $\mathbf{s_i} = (s_{ij} | j = 1, 2 \ldots d)$. A real valued response obtained by the simulator $y(\mathbf{s_i})$ is observed at every design point $\mathbf{s_i} \in \chi$ and the column vector $\mathbf{y} = (y(\mathbf{s_i}) : i = 1, 2 \ldots n)$ be the collection of all these observations at n design points. For any indeterminate point x, $f(x)$ is chosen so as to approximate the true function responsible to generate the response surface.

$f(\mathbf{x}) = (f_1(\mathbf{x})\ f_2(\mathbf{x}) \dots f_k(\mathbf{x}))^T$is a vector of k functions on an indeterminate point x in χ. The design model matrix is the $n \times k$ matrix, built by evaluating the vector $f(\mathbf{x})$ at n design points and denoted as $F = \{f(\mathbf{s_i})|i = 1, 2 \dots n\}^T$.

2.2 Sacks Model

The pioneer work in the field of computer experiments was put forward by [65]. This model gave rise to the concept of simulated designs, different from the conventional experimental designs, owing to the fact that predictions in computer experiments are based on the emulator which tries to capture the true observations of the complex physical process under study. To this end different metamodels also termed surrogates have been developed as an interpolator to the simulated data. One of the widely adopted surrogates include kriging introduced by [43] in the domain of spatial statistics. In the context of computer experiments kriging assumes the underlying process to be superposition of the linear model and departures from the linear model. The most widely employed emulator in the realm of computer experiments are Gaussian emulators which also borrow some elements of kriging. The model proposed by [65] takes account of the Gaussian emulator. For any indeterminate point x, let *y(x)* be a deterministic response which in kriging, is considered as a realization of a stochastic process, *Y(x)*, where *Y(x)* depends on $\mathbf{x}$ in a standard regression manner while residual variation follows Gaussian Random Function (GRF). The resulting model has the form

$$Y(x) = \sum_{j=1}^{k} f_j(x)\beta_j + Z(x)$$

which equivalently can be written as

$$Y(x) = f(x)^T\beta + Z(x) \tag{2.1}$$

where $f(x)$ is a vector composed of $(f_1(\cdot), f_2(\cdot) \dots f_k(\cdot))^T$ which are k known regression functions, the vector of unknown regression coefficients is $\beta = (\beta_1, \beta_2 \dots \beta_k)^T$. The term $Z(x)$ is assumed to be a zero mean Gaussian random process with covariance

$$Cov(Z(x_i), Z(x_j)) = \sigma^2\varrho(x_i - x_j), \quad \text{where } \varrho \text{ is the correlation function} \tag{2.2}$$

Equivalently,

$$Cov(Z(x_i), Z(x_j)) = \sigma^2 \varrho(h) \tag{2.3}$$

where, σ^2 is the process variance and $\varrho(h)$ is the correlation function between $Z(x_i)$ and $Z(x_j)$ that depends on the lag distance h between x_i and x_j. Any choice of correlation function is characterised by a parameter θ which is the length scale parameter and determines the speed of decay of the correlation between any two points. A detailed account of the various covariance kernels is dispensed in the Appendix (A.2).

2.2.1 MODEL ANALYSIS

The analysis requires that, at a design point s_i, we observe the response value $y(s_i)$. The following is the standard development from [65]. Let $\mathbf{y_s}$ be the vector of n observations, $\mathbf{y_s} = (y(\mathbf{s_1}), y(\mathbf{s_2}) \dots y(\mathbf{s_n}))^T$ at design $\mathscr{D} = \{\mathbf{s_i} | i = 1, 2 \dots n\}$. For predictions at an untried point x, the linear predictor $\hat{y}(x)$ is defined as

$$\hat{y}(x) = c(x)^T \mathbf{y_s} \tag{2.4}$$

For the sake of simplicity, bold notation is omitted for the development of the results in rest of the section. In order to find the best linear unbiased estimator (BLUE), the criterion of minimizing Mean Square Error is adopted. The frequentist approach we adopt replaces y_s and $y(x)$ by random quantities Y_s and $Y(x)$. Then the Mean Square Error is averaged over the random process.
This can be done by choosing a vector $c(x)$ of size $n \times 1$ to minimize

$$MSE(\hat{Y}(x)) = E(c(x)^T Y_s - Y(x))(c(x)^T Y_s - Y(x))^T \tag{2.5}$$
$$MSE(\hat{Y}(x)) = E\left(c(x)^T Y_s Y_s^T c(x) - Y(x) Y_s^T c(x) - c(x)^T Y_s Y(x) + Y(x) Y(x)^T\right)$$

subject to the constraint $\quad E(c(x)^T Y_s) = E(Y(x))$

It is important to provide expressions for some of the notation before derivation of MSE. The following object

$$R = \left(\varrho(s_i - s_j)\right)_{i,j} \quad 1 \le i \le n; 1 \le j \le n \tag{2.6}$$

is the $n \times n$ matrix of correlations between the process Z at the design sites. This matrix is constructed by the elements of $\varrho(s_i - s_j)$ evaluated at the design points. Denote by

$$r(x) = (\varrho(s_1 - x), \ldots \varrho(s_n - x))^T \tag{2.7}$$

is the correlation vector between the process Z at the design sites and at an untried input x.

In order to find the Mean Square Error of $\hat{y}(x)$, the expression given in Equation (2.5) is expanded as follows:

$$MSE(\hat{Y}(x)) = E\left(c(x)^T Y_s Y_s^T c(x) - 2c(x)^T Y_s Y(x) + Y(x)Y(x)\right), \tag{2.8}$$

where

$$E(Y_s Y_s^T) = \begin{pmatrix} E(Y(s_1)Y(s_1)) & E(Y(s_1)Y(s_2)) & \ldots & E(Y(s_1)Y(s_n)) \\ E(Y(s_2)Y(s_1)) & E(Y(s_2)Y(s_2)) & \ldots & E(Y(s_2)Y(s_n)) \\ \vdots & & & \\ E(Y(s_n)Y(s_1)) & E(Y(s_n)Y(s_2)) & \ldots & E(Y(s_n)Y(s_n)) \end{pmatrix}$$

To simplify this matrix we proceed as follows; let $\mu_s = \sum_{j=1}^{k} \beta_j f_j(s)$ so the expectation of every diagonal entry is,

$$\begin{aligned} E(Y^2(s_i)) &= Var\left(Y(s_i)\right) + \left(E(Y(s_i))\right)^2 \\ E(Y^2(s_i)) &= \sigma^2 + \mu^2(s_i) \qquad (2.9) \\ E(Y(s_i), Y(s_j)) &= E\left(\mu_{s_i} + Z(s_i))(\mu_{s_j} + Z(s_j)\right) \\ &= E\left(\mu_{s_i}\mu_{s_j} + \mu_{s_i} Z(s_j) + Z(s_i)\mu_{s_j} + Z(s_i)Z(s_j)\right) \\ E(Y(s_i), Y(s_j)) &= \mu_{s_i}\mu_{s_j} + \sigma^2\big(\varrho(s_i - s_j)\big) \qquad (2.10) \end{aligned}$$

Arranging terms in vector form, $E(Y_s Y_s^T)$ can be expressed as:

$$E(Y_s Y_s^T) = \mu_s \mu_s^T + \sigma^2 R. \tag{2.11}$$

Similarly, the expectation of $(Y_s Y(x))$ is,

$$E(Y_s Y(x)) = \begin{pmatrix} E(Y(s_1)Y(x)) \\ E(Y(s_2)Y(x)) \\ \vdots \\ E(Y(s_n)Y(x)) \end{pmatrix},$$

where for an individual entry we have,

$$\begin{aligned} E\left(Y(s_i)Y(x)\right) &= E(\mu_{s_i} + Z(s_i))(\mu_x + Z(x)) \\ &= E\left(\mu_{s_i}\mu_x + \mu_{s_i}Z(x) + \mu_x Z(s_i) + Z(s_i)Z(x)\right) \\ &= \mu_{s_i}\mu_x + E\left(Z(s_i), Z(x)\right). \end{aligned}$$

In summary,

$$E\left(Y(s_i)Y(x)\right) = \mu_{s_i}\mu_x + \varrho(s_i - x)$$

which can be expressed in vector form,

$$E\left(Y(s), Y(x)\right) = \mu_s\mu_x + r(x) \tag{2.12}$$

Here, recall that $E(Z(s_i)) = E(Z(x)) = 0$ and $r(x)$ is the correlation vector between design points and untried points, defined in Equation (2.7). The expectation of $Y^2(x)$ in Equation (2.8), gives the following result:

$$E\left(Y^2(x)\right) = V\left(Y(x)\right) + \left(E(Y(x)\right)^2 = \sigma^2 + \mu_x^2 \tag{2.13}$$

Collecting results from Equations (2.8, 2.10, 2.12 and 2.13), $MSE(\hat{Y}(x))$ is given as:

$$MSE(\hat{Y}(x)) = c(x)^T\sigma^2 Rc(x) + c(x)^T\mu_s\mu_s^T c(x) - 2c(x)^T\mu_s\mu_x - 2c(x)^T r(x) + \sigma_x^2 + \mu^2(x). \tag{2.14}$$

Given the condition $E\left(c(x)^T Y_s\right) = E\left(Y(x)\right) = \mu_x$ and $c(x)^T\mu_s = E\left(c(x)^T Y_s\right)$, the mean square error can be written as,

$$MSE(\hat{Y}_x) = \sigma^2\left(1 + c(x)^T Rc(x) - 2c(x)^T r(x)\right). \tag{2.15}$$

Development of predictor

The development of mean square error continues by developing an expression for the predictor $\hat{y}(x)$ given in Equation(2.4). This is done by minimizing mean square error subject to the constraint $F^T c(x) = f(x)$ in order to obtain the estimated value of c. This can be achieved by introducing Lagrange Multiplier.

Define Lagrangian function M as

$$M = \sigma^2 \left(1 + c(x)^T R c(x) - 2c(x)^T r(x)\right) + 2\sigma^2 \lambda \left(F^T c(x) - f(x)\right). \tag{2.16}$$

To minimize, differentiate Equation (2.16) with respect to $c(x)$ and equating the result to zero provide following equation $2\sigma^2 \lambda F^T + 2\sigma^2 c(x)^T R - 2\sigma^2 r(x)^T = 0$, that is

$$\lambda F^T + c(x)^T R = r(x)^T \tag{2.17}$$

Differentiating Equation (2.16) with respect to λ and equating the result to zero provides the constraint,

$$F^T c(x) = f(x) \tag{2.18}$$

Combining Equations (2.17 and 2.18) provides following form

$$\begin{pmatrix} 0 & F^T \\ F & R \end{pmatrix} \begin{pmatrix} \lambda(x) \\ c(x) \end{pmatrix} = \begin{pmatrix} f(x) \\ r(x) \end{pmatrix}$$

Assuming that the matrix on the left is invertible, the rearrangement yields

$$\begin{pmatrix} \lambda(x) \\ c(x) \end{pmatrix} = \begin{pmatrix} 0 & F^T \\ F & R \end{pmatrix}^{-1} \times \begin{pmatrix} f(x) \\ r(x) \end{pmatrix} \tag{2.19}$$

Matrix inversion in Equation (2.19) is done by the Inverse rule given by [42], where a detailed note is provided to find the inverses of 2×2 block matrices. The supporting theorem is given next.

Theorem 2.1. *Let a non singular matrix E be partitioned into 2×2 blocks as*

$$E = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \quad \textit{with } A \textit{ and } D \textit{ square matrices not necessarily of the same size.}$$

1. *Assume A is non singular. Then the matrix E is invertible if and only if the Schur complement $D - CA^{-1}B$ of A is invertible*

2. *Assume D is non-singular. Then the matrix E is invertible if and only if the Schur complement $A - BD^{-1}C$ of D is invertible*

Corollary 2.1.1. *Consider the matrix*

$$E = \begin{pmatrix} 0 & B \\ C & D \end{pmatrix}, \quad \textit{with } D \textit{ a square matrix.}$$

For squared diagonal partition with D non singular, it is invertible if and only if $BD^{-1}C$ is also invertible, and it has inverse

$$\begin{pmatrix} -(BD^{-1}C)^{-1} & (BD^{-1}C)^{-1}BD^{-1} \\ D^{-1}C(BD^{-1}C)^{-1} & D^{-1} - D^{-1}C(BD^{-1}C)^{-1}BD^{-1} \end{pmatrix}$$

Using Theorem (2.1) and Corollary (2.1.1), the inverse of matrix in Equation (2.19) can be found. After substituting we have the estimate of $c(x)$ as

$$c(x) = f(x)^T(F^TR^{-1}F)^{-1}F^TR^{-1} + r(x)^TR^{-1}(I - F(F^TR^{-1}F)^{-1}F^T)$$

Then the expression $c(x)^Ty_s$ in Equation (2.4) can be evaluated to obtain the predictor $\hat{y}(x)$:

$$\hat{y}(x) = f(x)^T(F^TR^{-1}F)^{-1}F^TR^{-1}y_s + r(x)^TR^{-1}(y_s - F(F^TR^{-1}F)^{-1}F^Ty_s). \tag{2.20}$$

Maximum Likelihood Estimation:Profile Likelihood

In Sacks model (2.1), the vector of stochastic response variables Y_s follows a multivariate normal distribution with mean vector $F\beta$ and covariance matrix σ^2R, that is $Y_s \sim \mathcal{N}(F\beta,\, \sigma^2R)$. Let the model parameters be denoted by $\phi = (\beta, \sigma^2, \theta)$ which contains all the parameters associated with the correlations in the matrix R. In order to estimate model parameters, we use method of maximum likelihood. The likelihood can be written as

$$L(Y_s|\beta, \sigma^2, \theta) = \frac{1}{(2\pi)^{\frac{n}{2}}|\sigma^2R|^{\frac{1}{2}}}e^{-\frac{1}{2}\left((Y_s-F\beta)^T(\sigma^2R)^{-1}(Y_s-F\beta)\right)} \tag{2.21}$$

Taking logarithm of the likelihood and simplifying terms, gives the following expression

$$l(\beta, \sigma^2, \theta) = -\frac{n}{2}ln2\pi - nlog\sigma - \frac{1}{2}ln|R| - \frac{1}{2\sigma^2}(Y^TR^{-1}Y_s - 2\beta^TF^TR^{-1}Y_s + \beta^TF^TR^{-1}F) \tag{2.22}$$

Differentiating the above log likelihood with respect to β and σ^2 and equating the differential to zero, provides the optimal values of β and σ^2.

$$\frac{\partial l(\beta, \sigma^2, \theta)}{\partial \beta} = -\frac{1}{2\sigma^2}(-2F^T R^{-1} Y_s + 2F^T R^{-1} F\beta)$$

Equating the above expression to zero, the usual weighted least squares estimate of β is obtained

$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} Y_s. \tag{2.23}$$

Similarly, the estimator of

$$\hat{\sigma}^2 = \frac{Y_s^T R^{-1}(I - F(F^T R^{-1} F)^{-1} F^T) R^{-1} Y_s}{n} \tag{2.24}$$

Profile likelihood [75] is used for the the estimation of θ. This likelihood is obtained by substituting the estimated values of β and σ^2 back in Equation (2.22). The profile likelihood expression takes the following form after substituting estimated values of β and σ^2 and taking logarithm.

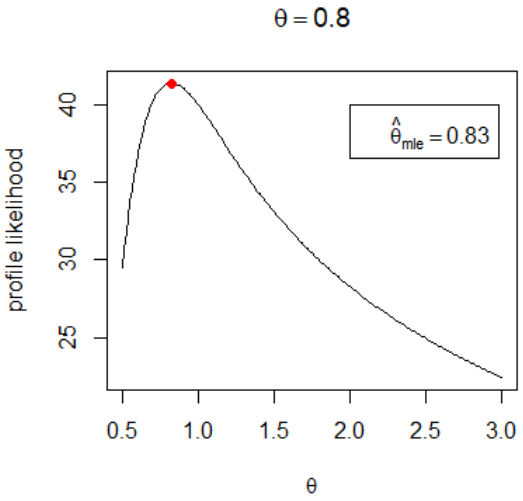
$$l_p(\theta) = -\frac{n}{2} log(2\pi) + \frac{n}{2} log(n) - \frac{n}{2} log(C) - \frac{1}{2} log|R| - \frac{1}{2} \tag{2.25}$$

where, $C = Y^T R^{-1}(I - F(F^T R^{-1} F)^{-1} F^T) R^{-1} Y$.

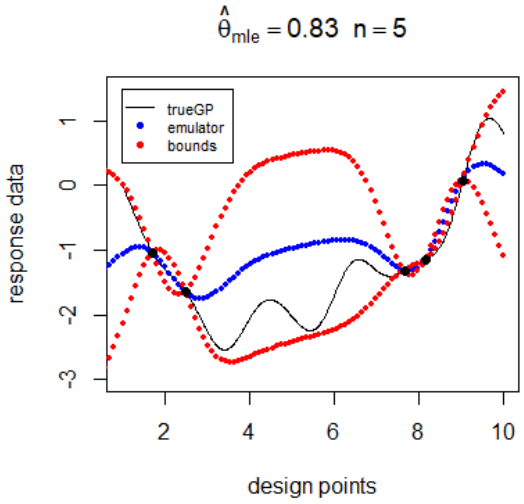
2.2.2 Example with synthetic data

In this section we provide an example of synthetic data where the experimental codes are developed with the aid of `R` in order to obtain the profile likelihood estimate of the parameter θ. A random data is generated from the Gaussian process for the input variable x in the range $[0, 10]$ with true $\theta = 0.8$. A random sample of 10 design points is selected from 500 Gaussian data points and the likelihood estimates of β and σ^2 are found with the use of Equations (2.23, 2.24). A random search is made among a grid of 200 values of θ over the region [0.5 , 3] to estimate the parameter θ. The profile likelihood estimate $\hat{\theta} = 0.83$ is the one that minimizes the profile likelihood in Equation (2.25). After estimating the parameter vector $\phi = (\beta, \sigma^2, \theta)$ we want to study the behaviour of mean squared prediction error for different sample size. For this purpose we chose random samples of size $n = \{5, 10, 15\}$ from the Gaussian process. For each of these samples we construct design model matrix F and correlation matrix R with the estimated value of $\theta = 0.83$. We compute the predictor $\hat{y}(x)$ in Equation (2.20) and mean square error of $\hat{y}(x)$ in Equation (2.15) for 100 untried points. In addition, the confidence bounds are also constructed given by the expression $\hat{y}(x) \pm MSE(\hat{y}(x))$. The results are displayed in the

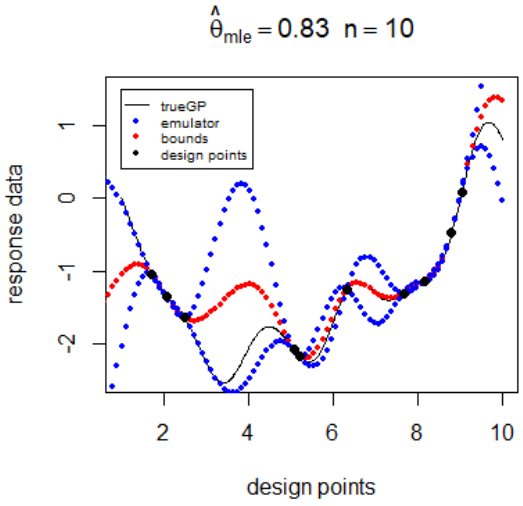
Figures (2.1).



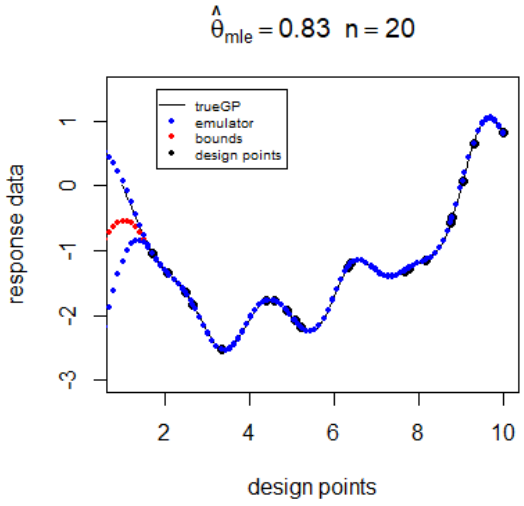
(a) profile likelihood of θ for synthetic example



(b) MSE plot for $n = 5$



(c) MSE plot for $n = 10$



(d) Emulator plot for $n = 20$

Figure 2.1: Plots for emulator and bounds for n={5,10,20}

It is evident from Figures (2.1) that adding more design points or increasing the sample size n results in improved approximation of the underlying process by the emulator. For example the emulator (red dots) becomes closer to the true process (black line) as training data points increases from $n = 5$ to $n = 20$ in Figures (2.1 (a,b,c)). In addition, confidence bounds are wider for smaller sample size and become narrower with increasing sample size. This is explained by

the fact that adding more data points reduces the prediction error hence mean squared error is small for large samples giving rise to small prediction bounds. In addition, we observe that prediction bounds are wider for the region with no design point Figure (2.1 (a,b,c)).

2.2.3 Analysis of Circuit-Simulator Data

We revisit the circuit-simulator example studied by [65] and reproduce the results in this section. The input data in this example is comprised of six transistor dimensions and the response is measure of asynchronization of two clocks. The observed data consist of 32 realizations. The correlation function used in [65] is the product exponential given as

$$\varrho(w,x) = \prod_{j=1}^{6} \exp(-\theta_j (w_j - x_j)^2)$$

The `R` package `DiceKriging` developed by [63] is employed to find maximum likelihood estimator of the correlation parameters $\theta_1, \theta_2, \ldots, \theta_6$. The first 16 data values are used for estimation of parameters θ_i for $i = 1, \ldots 6$, and *Gauss* is used as the covariance function among the available covariance functions in `DiceKriging`. The employment of *Gauss* function is justified by the fact that it has the closest form to the one used in [65], which is not readily available in the package. The estimated values thus obtained are transformed in order to obtain the closest estimates of the correlation parameters. The transformation is done as follows

$$\text{Gauss exponent} : \varrho(x_i, x_j) = exp(-\frac{1}{2}(|x_i - x_j|\theta)^2) \tag{2.26}$$

$$\text{Product Power exponent} : \varrho(x_i, x_j) = exp(-\theta^*(|x_i - x_j|)^p) \tag{2.27}$$

$$\theta^* = \frac{1}{2}\theta^2 \qquad \text{for} \quad p = 2$$

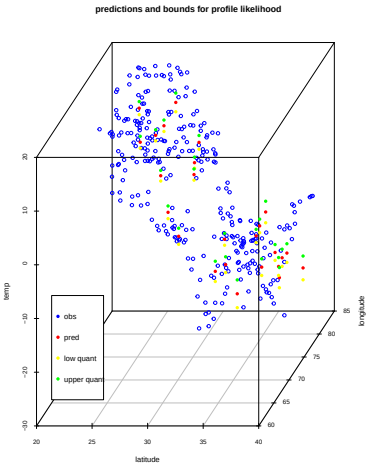
Table 2.1: **Estimation of parameters for transistor data**

	Sack's estimates [65]	Gauss estimates	Transformed estimates
θ_1	0.00	1.120	0.3985
θ_2	0.39	0.565	1.56
θ_3	0.42	0.489	2.08
θ_4	0.53	0.710	0.99
θ_5	1.97	0.477	2.19
θ_6	0.46	0.669	1.12

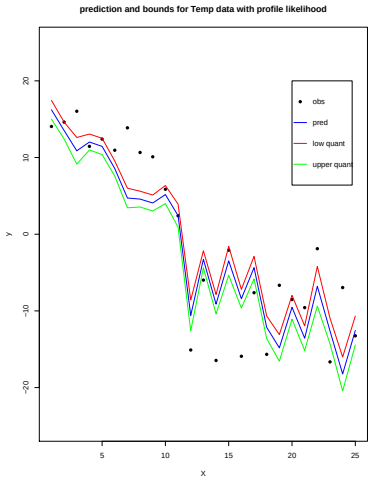
We observe from Table (2.1) that the estimated values obtained by `DiceKriging` analysis are different from the ones obtained by [65].

2.2.4 Analysis of Temperature data

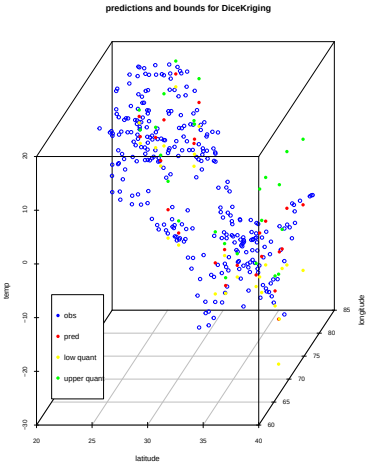
In order to study a practical application of the Sacks model in Section (2.2), we analyse temperature data from Pakistan Meteorological Department. The data comprises 100 years temperature forecasts spanning over 12 months for 5599 grid values of the input variables, latitude and longitude giving rise the total data points equal to $6,718,804$. A proportion of data is analysed taking into account the year 2010 and the temperature records for the month of January for the same year. There are 5599 data points for the chosen study and a random sample of size 300 is selected as training points whereby the model is validated for randomly selected 25 untried data points. The model employed for the estimation and prediction is Sacks Model described in Section (2.2). The analysis is done both with the help of `R` code developed for profile likelihood given in Equation(2.25) and with the use of package `DiceKriging` where $Matern(5/2)$ is used as a covariance function. The covariance parameter θ is estimated with the package `DiceKringing` and the same is employed in evaluating likelihood profile (2.25). Empirical mean square error of prediction using `R` code is 24.45 and the same is 5.414 for *DiceKriging*. Since the input is 2-dimensional grid, therefore, 3-d plots are given along with simple graphical presentation of the predictor and bounds for both the methodologies. It is pertinent to mention that figures below on the left are the plots of estimated and predicted temperature measures against the inputs latitude and longitude. Whereas the plots on the right shows the estimated and predicted values along with the bounds against the number of untried locations which is 25. The results are displayed in Figure (2.2)



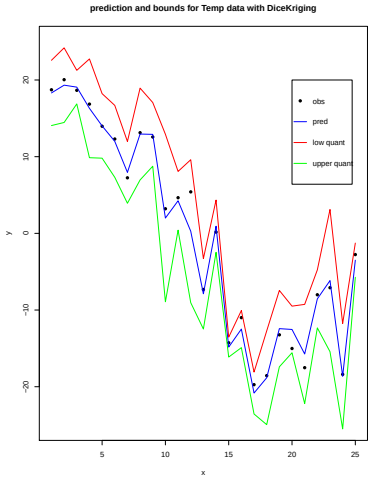
(a) prediction and bounds (LP) 3d plot



(b) prediction and bounds (LP)



(c) prediction and bounds (DiceKriging) 3d plot



(d) Prediction and bounds (DiceKriging)

Figure 2.2: Plots for emulator and bounds of Temp data

2.3 Bayesian version of Sacks Model

In Section (2.2) Sacks model is evaluated in detail from frequentist point of view. This section is devoted to analyse the same model within Bayesian framework. Bayesian models are often convenient to work with particularly in situations when the number of parameters to be estimated or the number of dimensions increase so that it is no more feasible to employ classical frequentist approach towards the optimal estimation of the unknown parameters. Since, it is observed

from the analysis of data for Sacks model that analytical solution for correlation parameter is cumbersome to obtain and numerical approximation becomes expensive when number of dimensions increase. Therefore, we adopt Bayesian approach to analyse Sacks model. The standard Bayesian technique for the Sacks model given in Equation (2.1) is described next.

In context of Bayesian framework, the unknown parameters are assumed to be the random quantities. Therefore, it infers that the unknown parameter is modelled by the random variable Θ which follows a probability distribution $p_\Theta(.)$, known as the prior distribution. The prior distribution elicits one's beliefs about the unknown parameters before observing the data. The observed data is assumed to have a distribution conditioned on the parameter Θ given by $p_{Y|\Theta}(y|.)$. Therefore the joint distribution of observed data and the parameter can be expressed as

$$p_{Y,\Theta}(y,.) = p_{Y|\Theta}(y|.) \times p_\Theta(.)$$

From joint distribution above, it is straightforward to define conditional distribution

$$p_{\Theta|Y}(. \mid y) = \frac{p_{Y,\Theta}(y,.)}{p_Y(y)}$$

which can be simplified as,

$$p_{\Theta|Y}(. \mid y) \propto p_{Y|\Theta}(y|.) \times p_\Theta(.) \tag{2.28}$$

that is posterior $\propto$ likelihood $\times$ prior where, $\propto$ accounts for the proportionality factor $p_Y(y)$ which is independent of the parameter. The left hand side of the relation (2.28) is the posterior distribution of parameters whereas the first term on right hand side is the likelihood of data given parameters and the second term is the joint prior density of unknown parameters. The posterior distribution represents the updated knowledge about parameters after observing the data. Once the posterior distribution is obtained for priors, one can easily find the posterior predictive distribution. Recall that the entire parameter vector for the model under study is $\phi = (\beta, \sigma^2, \theta)$, and let Θ be the joint random variable associated with ϕ, then the predictive distribution of the response $\hat{y}(x)$ at untried points denoted by y_0 is given by following expression:

$$p(y_0 \mid y) = \int p(y_0 \mid y) \times p(\phi \mid y), \tag{2.29}$$

where y_0 is the predicted response at an unobserved point and y is the observed response.The

expectation and covariance of the posterior predictive distribution is defined as

$$E(y_0 \mid y) = E_{\Theta|y}\left(E(y_0 \mid \phi, y)\right) \tag{2.30}$$

$$Cov(y_0 \mid y) = E_{\Theta|y}\left(Cov(y_0 \mid \phi, y)\right) + Cov_{\Theta|y}\left(E(y_0 \mid \phi, y)\right) \tag{2.31}$$

In this Bayesian setting, analytical form of the predictive distribution can be obtained with some effort, when number of parameters are not very large thus making the integral less complicated. However, when the unknown parameters increase with increase in dimension, the posterior distribution becomes hard to obtain in analytic form. Therefore, it is reasonable to use approximation methods such as Markov Chain Monte Carlo (MCMC) which allows to generate samples from posterior distribution without being perturbed by analytical form of the distribution. Nevertheless, the method allows to compute predictive mean and variance for the response at unknown tried points using Equation (2.30) and Equation (2.31).

2.3.1 Markov chain Monte Carlo (MCMC) and comparison with Maximum Likelihood

A simple introduction of Monte Carlo Markov Chains is given in this section. MCMC encompasses a set of algorithms to draw samples from a probability distribution. It helps to characterize the distribution even when the mathematical properties are not completely specified. MCMC is a combination of two properties namely, **Monte-Carlo** is the method to approximate the properties of the distribution from large number of samples drawn from that distribution. **Markov chain** borrows its name from Markov property which states that the probability of observing any given state depends only on the state one before and not on any other previous states. This leads to define the rule of sampling for approximating the target distribution. It infers that, in Markov chain, each sample generated is a state on which the very next sample selection depends on. In other words new samples do not depend on the previous samples but only the last one. A detailed exposition of MCMC can be found in [27]. We use the synthetic data to compare maximum likelihood method and Bayesian methodology using MCMC technique for making predictions for Sacks model expressed in Equation (2.1). The data is randomly generated with the Gaussian process function in `R` for true value of the scale parameter $\theta = 1.6$ for only one input x. First, maximum likelihood is employed similar to what is given in Equation (2.25) and θ is estimated in a similar fashion as described in Section (2.2.1). A random search is made among 100 different values of θ in the grid [0.08 90]. For each of these randomly selected values, likelihood is computed and we found $\hat{\theta}_{mle} = 2.8$. The

MCMC sampler was run with the following prior specifications:
$\beta \sim \mathcal{N}(\mu = 1, \sigma^2 = 2)$, $\sigma^2 \sim \Gamma(shape = 1, rate = 1)$ and $\theta \sim \beta \exp(shape = 3, rate = 6)$ The stochastic response variable Y follows normal distribution (recall from section 2.2) such that

$$Y \mid \phi \sim \mathcal{N}(F\beta, \sigma^2 R)$$

The posterior distribution defined in Equation(2.28) does not have a closed analytical form for the given choice of priors. Therefore, one of the MCMC algorithms termed `Metropolis Hasting` [31] is adopted to run the chain. The algorithm develop its roots by defining the target distribution $q(x)$ and proposal conditional distribution $g(x^* \mid x)$ where x^* is the new sample and x is the current sample from the Markov chain. The algorithm runs in the following sequence

Algorithm 1 Metropolis Hasting

Initialize the chain at some value x^0
Draw a uniform random number $u \sim U(0, 1)$
Generate a new test sample $x^* \sim g(x^* \mid x^i)$
if $u < min\left(1, \dfrac{q(x^*)g(x^i \mid x^*)}{q(x^i)g(x^* \mid x^i)}\right)$ **then**
    $x^{i+1} = x^*$
else $x^{i+1} = x^i$
    continue until desired number of runs
end if

The chain is run for 300000 iterations with the initial values of the parameters , $\beta_1 = 2$, $\beta_2 = 1$, $\sigma^2 = 1$ and $\theta = 2$ respectively. The estimated value of θ is given by the expected value of posterior distribution of θ, which is found to be 1.82. Moreover, predictions at untried and design points are also computed along with mean square error using MCMC estimates of the parameters. The following table and plots provide details of the results of MCMC estimation.

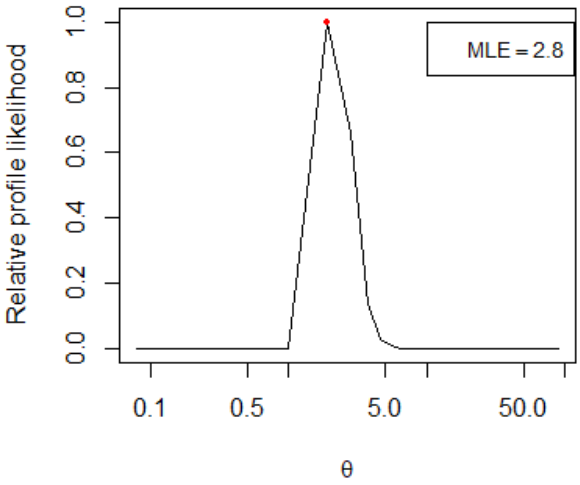


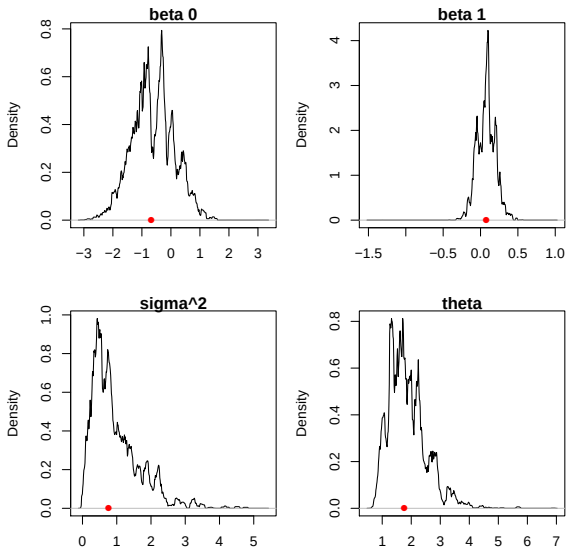
Figure 2.3: MLE for θ

Parameters	Mean	SD	SE
β_0	-0.5247	0.7010	0.0012
β_1	0.069	0.1188	0.0002
σ^2	0.9564	0.7107	0.00129
θ	1.8186	0.6524	0.00119

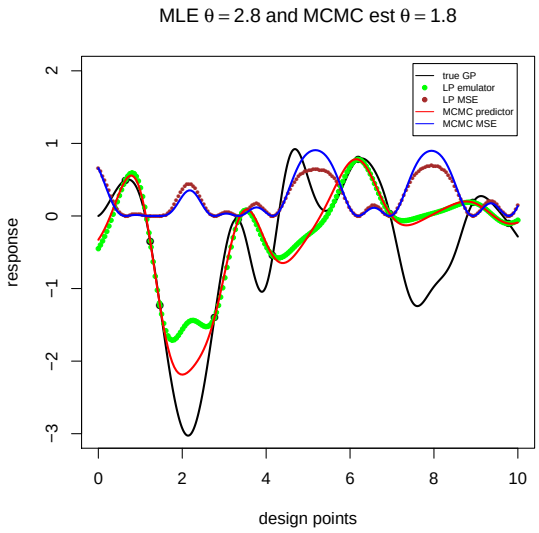
Table 2.2: MCMC estimation of parameters from Posterior Distribution

Parameters	2.5%	25%	50%	75%	97.5%
β_0	-1.8716	-0.9758	-0.50532	-0.0924	0.7673
β_1	-0.1738	0.001529	0.07182	0.14807	0.3014
σ^2	0.1325	0.42737	0.82268	1.23411	3.0167
θ	0.8765	1.3334	1.71405	2.2659	3.3608

Table 2.3: Quantiles for each parameter



(a) MCMC density plots



(b) MSE and Emulator plots for MCMC and ML estimation

Figure 2.4: comparison of MCMC and ML estimation

2.3.2 Comments

From comparison of maximum likelihood estimate and MCMC estimate of θ it is evident from Table (2.2) that MCMC estimate of $\theta = 1.8186$ is close to the true value of $\theta = 1.6$ as compared to MLE estimate of $\theta = 2.8$. Moreover the density plots Figure (2.4a) fulfil the assumption of normality, depicting MCMC random walk. Therefore, MCMC approach is convenient to employ particularly in situations when number of parameters increase with the increase in dimensions. However, implementation of MCMC needs careful consideration of the choice of priors and proposal distribution. Once, an intelligent choice is made based on some past knowledge or expert experience, one may use MCMC to obtain an approximate posterior distribution and hence predictions can be made at untried design points. We can see from Figure (2.4b) that response is predicted at untried points with small mean square error for both the methods. It leads to the conclusion that although MCMC estimate is closer to the true value compared to maximum likelihood method, however, no method outperforms the other in terms of mean square error of predictions.

2.4 Model 2 (Convex Combination of Gaussian Processes)

We review the second model in detail proposed by [29]. The prime difference of this model from the Sacks Model lies in the fact that the response $y(x)$ is assumed to be the realization of a stochastic process, $Y(x)$, which is a mixture of two independent Gaussian processes. The combination of more than one Gaussian process is useful when one wishes to characterize both global trends and finer details in the same model or when there is uncertainty regarding the family of correlation functions [29]. The model takes the following form

$$Y(x) = f(x)^T\beta + Z(x) + \epsilon(x) \tag{2.32}$$

where, $f(x)$ is a vector of known regression functions and β is a vector of unknown regression coefficients, $\epsilon(x)$ is random error, and $Z(x)$ is a convex combination of two Gaussian processes such that

$$Z(x) = pZ_1(x) + (1-p)Z_2(x), \qquad 0 \leq p \leq 1 \tag{2.33}$$

where, $Cov(Z_1(x), Z_1(w)) = \sigma^2 R_1(|h|^p; \theta_1)$ and $Cov(Z_2(x), Z_2(w)) = \sigma^2 R_2(|h|^p; \theta_2)$ where $h = (x - w)$ is the lag difference between any two design points, $R_i(h; \theta_i)$ is the correlation matrix constructed by the elements of $\varrho(h)$ details of which are elucidated in Section (2.2.1) and $\theta = (\theta_1, \theta_2)$ are correlation parameters associated with these two Gaussian processes.
The variance and covariance of $Z(x)$ in equation (2.33) can be evaluated in a simple fashion

$$Var(Z(x)) = p^2 Var(Z_1(x)) + (1-p)^2 Var(Z_2(x)) = (p^2 + (1-p)^2)\sigma^2, \tag{2.34}$$

and

$$\begin{aligned} Cov(Z(x_i), Z(x_j)) &= p^2 Cov(Z_1(x_i), Z_1(x_j)) + (1-p)^2 Cov(Z_2(x_i), Z_2(x_j)) \\ &= \sigma^2(p^2\varrho_1(x_i - x_j) + (1-p)^2\varrho_2(x_i - x_j). \end{aligned} \tag{2.35}$$

The correlation between $Z(x_i)$ and $Z(x_j)$ becomes

$$Corr(Z(x_i), Z(x_j)) = \frac{p^2\varrho_1(x_i - x_j) + (1-p)^2\varrho_2(x_i - x_j)}{(p^2 + (1-p)^2)} \tag{2.36}$$

From the above expressions

$$Cov(Z(x), Z(w)) = \sigma^2 \tilde{R}(|h|^p; \theta_1, \theta_2) \tag{2.37}$$

The matrix $\tilde{R}(h; \theta_1, \theta_2)$, or simply $\tilde{R}$ is formed by the weighted combination of two correlation matrices as explained

$$\tilde{R}_{p,\theta_1,\theta_2}(h) = \frac{p^2 R_1(h; \theta_1) + (1-p)^2 R_2(h; \theta_2)}{(p^2 + (1-p)^2)} \tag{2.38}$$

Simplifying

$$\begin{aligned} R_1 = R_1(h; \theta_1) = (\varrho_1(x_i - x_j))_{i,j} \quad 1 \leq i \leq n; 1 \leq j \leq n \\ R_2 = R_1(h; \theta_2) = (\varrho_2(x_i - x_j))_{i,j} \quad 1 \leq i \leq n; 1 \leq j \leq n \end{aligned} \tag{2.39}$$

The model analysis is done with the help of Bayesian approach and change of variables. Posterior distribution and predictions are obtained with the use of Metropolis-Hasting algorithm. We on the contrary try to follow the same developments explained in Section (2.2) for Convex Combination of Gaussian processes. We assume from the model analogy that the expressions developed for the predictor $\hat{y}$, mean square error of $\hat{y}$ and the maximum likelihood for β, σ^2 and $\theta = (\theta_1, \theta_2)$ takes the same form as that for Sack's model (2.1) with the only difference in correlation matrix, R and $\tilde{R}$. In other words, we simply replace R with $\tilde{R}$ and run some simulation studies.

2.4.1 Maximum Likelihood Estimator

Maximum likelihood estimator for the convex model is computed using the similar expression given in Equation(2.25) by replacing R with $\tilde{R}$ defined in Equation (2.38)

$$l(\beta, \sigma^2, \theta) = -\frac{1}{2}\left(n \ln(2\pi) + n \ln(\sigma^2) + \ln|\tilde{R}| + n \ln \tilde{C} + 1\right) \tag{2.40}$$

where,

$$\tilde{C} = Y^T \tilde{R}^{-1}(I - F(F^T \tilde{R}^{-1} F)^{-1} F^T)\tilde{R}^{-1} Y \tag{2.41}$$

and $\tilde{R}$ is given in Equation(2.38).

A data of size $N = 600$ is generated using two Gaussian processes, one with true value of $\theta_1 = 1.5$ and the other having true value of $\theta_2 = 5$. A sample of size $n = 10$ and 25 over

the region $[0,\ 10]$ are selected at random from combination of two Gaussian processes. These samples are used as design points $\mathbf{s_i}$, $1 \leq i \leq n$ with realization of response $y(\mathbf{s_i})$. Different values of θ_1 and θ_2 are explored in order to search for the likelihood estimator for the true parameters as:

1. A grid of 60 values of θ_1 over $[0.8,\ 4]$
2. A grid of 60 values of θ_2 over $[3,\ 8]$
3. $p_1 = 0.3$ and $p_2 = 0.7$

For every value of θ_1, we compute relative profile likelihood Equation (2.40) for each value of θ_2. The estimates of θ_1 and θ_2 are the ones that maximize the relative profile likelihood. The maximum likelihood estimates of θ_1 and θ_2 are found to be **1.7** and **4.7** respectively. The contour plot of the relative profile likelihood is given in Figure (2.5).

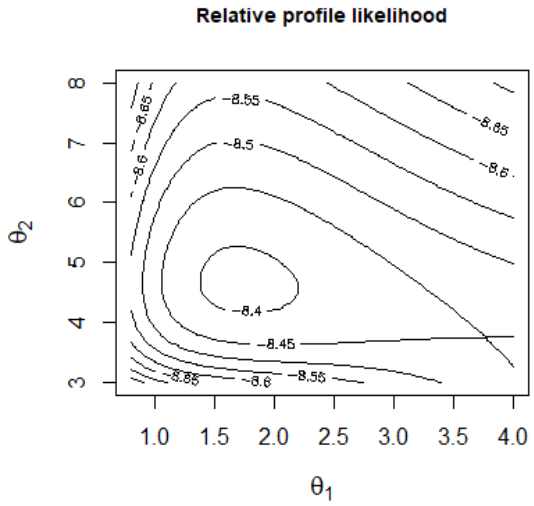
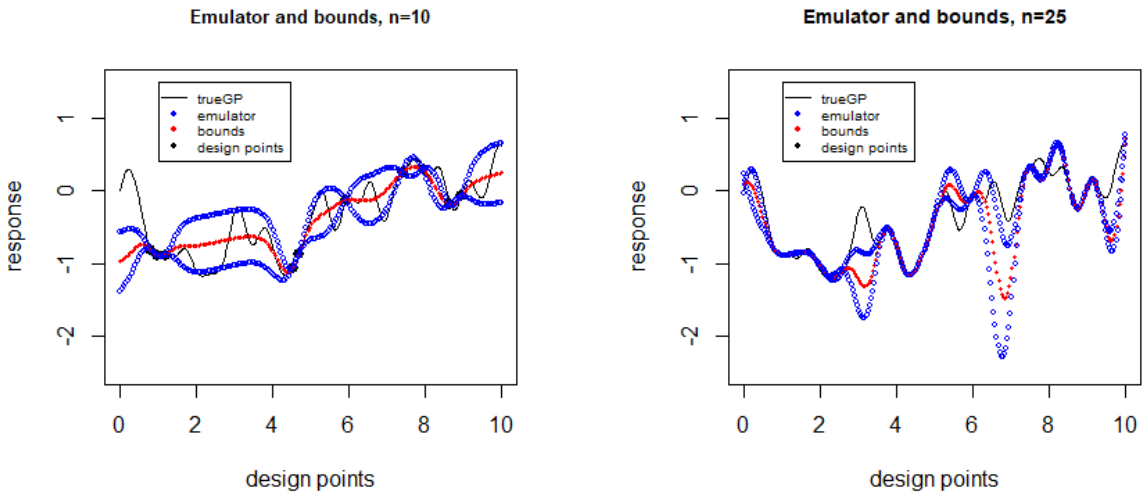


Figure 2.5: Contour Plot MLE for CGP $n = 10$

For the estimated values of $\theta_1 = 1.7$ and $\theta_2 = 4.7$, emulator in Equation (2.20) with $\tilde{R}$ is fit to the true data with $n = 10$ and $n = 25$ design points. The mean square error of predictions is computed from Equation (2.15) and the relative prediction bounds are computed for 200 untried input data points. The plots of true Gaussian process, emulator and mean squared prediction errors for $n = 10$ and $n = 25$ design points are presented in Figure (2.6).



(a) Emulator and MSE plot for n=10 (b) Emultor and MSE plot for n=25

Figure 2.6: Plots for emulator and bounds CGP

We observe that increasing the design points plays an important role in improved interpolation of true process and reducing the mean square error, as seen by confidence intervals. From Fig (2.6), it can be seen that by increasing sample size from $n = 10$ to $n = 25$, the emulator closely interpolates the true process and confidence bounds become narrower where new points lie close to the design points. However, for the regions outside design points the emulator performs poorly in terms of interpolation and prediction, resulting in wider confidence bounds.

2.4.2 Comparative study of Design Choices

We observed the emulator performance for random samples in Section (2.4.1). An important feature of Sacks emulator Equation(2.20) is that it provides exact predictions at design points and performs poorly if the unknown points do not lie close to the training data points. Therefore we want to explore further different choices of design points in order to investigate the behaviour of contour plots, maximum likelihood estimator of correlation parameters θ_1 and θ_2 and mean squared prediction error. In order to study different design choices, the data of size $N = 600$ is generated for each of two Gaussian processes with true parameters $\theta_1 = 1.5$ and $\theta_2 = 5$ respectively. A sample of size $n = 25$ is selected from the data combination of two Gaussian processes. Following are the different design choices considered for this study.

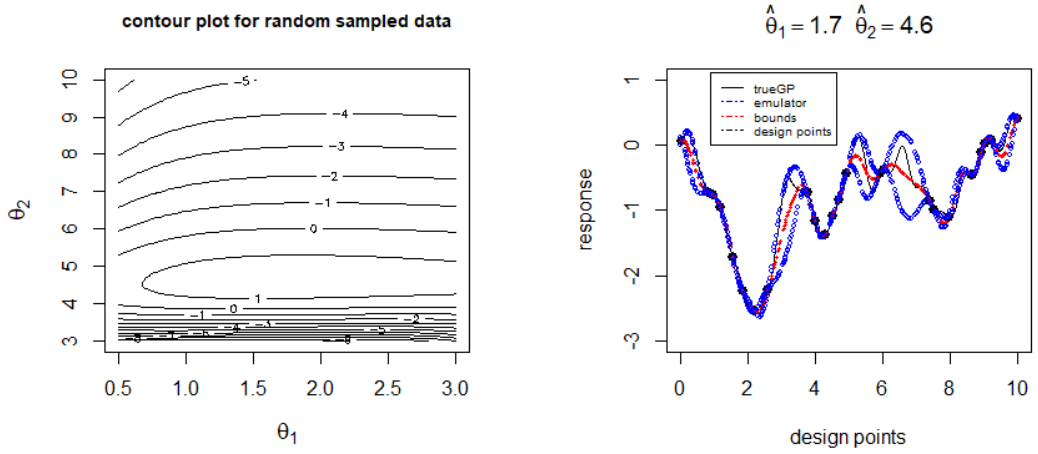
1. Even distribution of design points over entire data space

2. sample selection from two extreme ends

3. sample selection from clusters in three regions

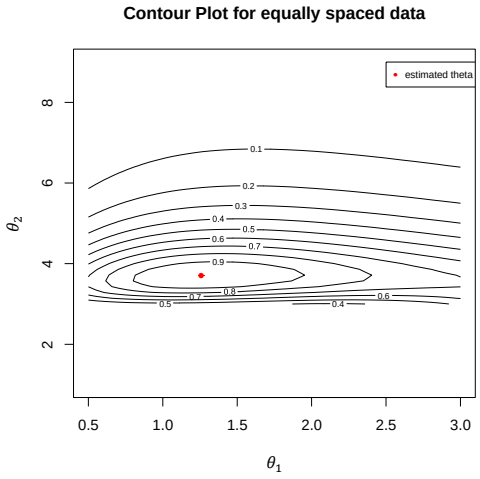
4. sample chosen by adding fixed constant to each sample point

The behaviour of the model at different design choices are evaluated and maximum likelihood estimates of the parameters are obtained. In addition, the predictions are made at 300 untried points for each of the design at estimated values of the parameters. The empirical errors are also computed as the difference between the true and predicted values of the response for each of these designs i.e. $\sum(y_i - \hat{y}_i)^2$. The true process with design points,their respective contour plots along with the emulator predictions and bounds are given in Figures (2.7, 2.8, 2.9, 2.10 and 2.11).

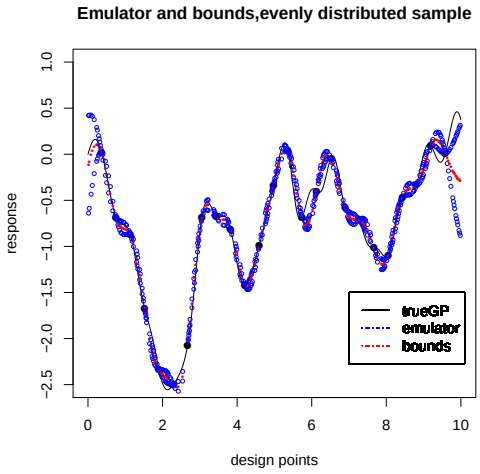


(a) Contour Plot (b) emulator and prediction bounds

Figure 2.7: Plots for a random design of $n = 25$ data points

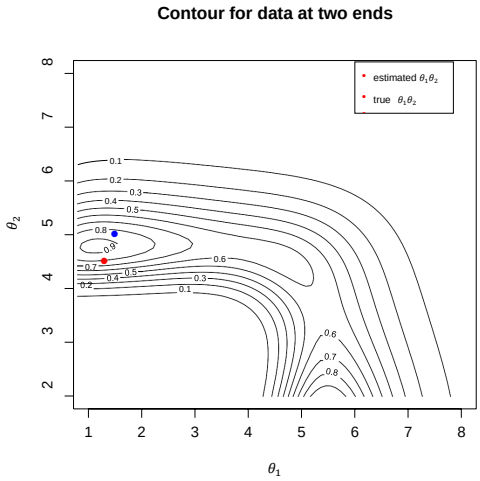


(a) Contour Plot

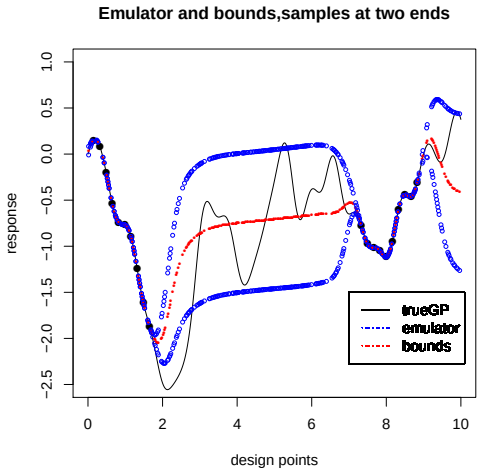


(b) emulator and prediction bounds

Figure 2.8: Plots for equally spaced design points

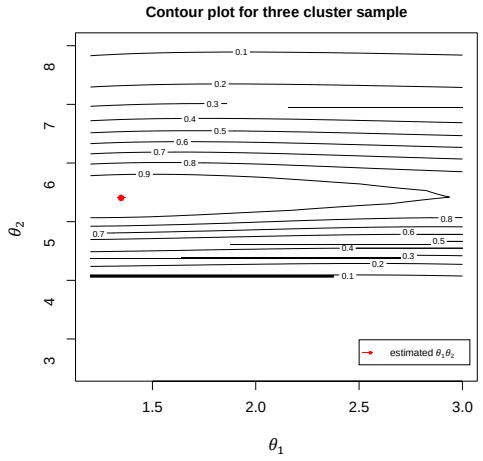


(a) Contour Plot

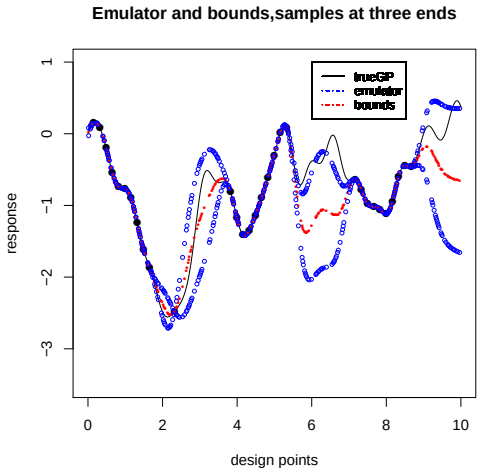


(b) emulator and bounds

Figure 2.9: Plots for design points at two extremes of design region

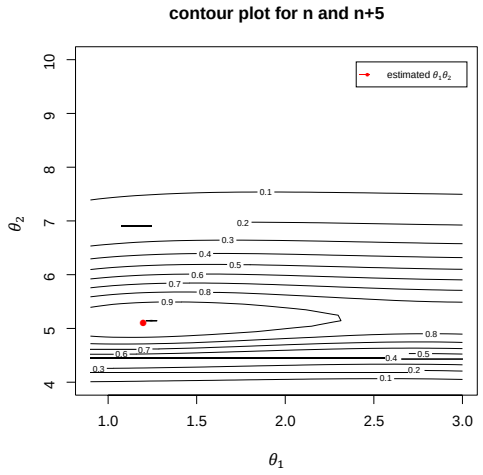


(a) Contour Plot

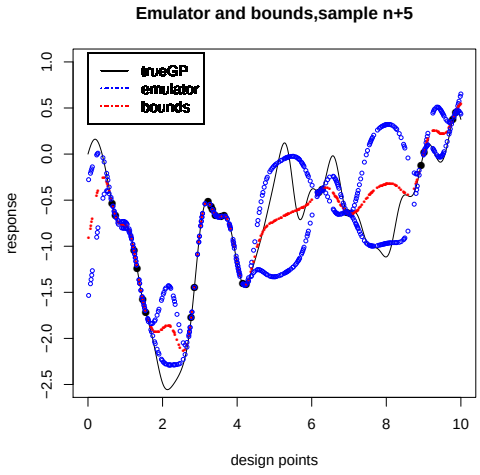


(b) emulator and prediction bounds

Figure 2.10: Plots for design points in three clusters



(a) Contour Plot



(b) emulator and bounds

Figure 2.11: Plots for random design points with a constant (5) added to each point

In addition to the development of different plots, the empirical errors are also measured. A summary of the mean square error along with the estimated values of the parameters for each of the sample designs are presented in Table (2.4).

design points	**MSE($\times$10)**	**MLE(θ_1)**	**MLE(θ_2)**
Random sample	1.095	1.7	4.6
equally spaced	0.198	1.26	3.7
two ends	1.029	1.2	4.77
three ends	3.004	1.34	5.4
Sample+5	1.218	1.24	5.24
True value		1.5	5

Table 2.4: MSE and MLE for design choices in Convex Combination model

We also compared Sacks model (Equation (2.1)) with that of Convex combination model (Equation (2.32)) and obtained the following results in Table(2.5)

MLE and MSE for different designs (Sacks model)		
design	$MLE(\theta)$	$MSE(\times 10)$
random	2.8	0.210
equally spaced	0.85	0.004
two ends	0.98	0.237
three ends	1.17	0.866
sample+5	0.97	0.039
True value	0.8	

Table 2.5: MSE and MLE for design choices in Sack's model

2.4.3 Comments

For the Convex combination model, the closest approximation to the true value of θ_1 is found for the design with design points in three clusters and that of θ_2 is more close for design where a constant value 5 is added to each of the chosen sample. Also, the random sample design provides the worst estimation of θ_1 and θ_2 . The mean square error is smallest for equally spaced samples and highest for samples chosen at three clusters. For Sacks model, equally spaced design provide closest approximation to the true value of θ and mean square error is also minimum for the same design. Different design choices may provide different results for different datasets, therefore, the results obtained for Convex Combination model and Sacks model cannot be generalized. However, the design with equally spaced data points outperform all other designs by providing minimum least squared prediction error for both the models as evident from Tables (2.4) and (2.5).

2.5 Calibration of Computer models: Linking computer modelling with Reality

Kennedy and O Hagan [34] proposed a calibrated model within Bayesian framework whereby, calibration refers to gain some understanding of the unknown inputs which are called calibration parameter. These are the inputs whose true values are unknown yet they play important part in defining the response. Consequently, the calibration model is characterized by two groups of inputs that are to be distinguished for computer models. One of these inputs comprises the fixed but unknown inputs, referred to as calibration inputs and are assumed to be fixed for all the observations. The other group of inputs are known variables whose values may be changed over the process. The proposed calibrated model is aimed to gain some knowledge about the unknown inputs with the help of observations from a physical process, particularly, when the underlying system being studied is approximated by complex mathematical models. In order to estimate the calibrated parameters and to make predictions about the true value of the real process based on physical observations, Bayesian approach has been adopted. Let the observation z_i be the realization of a random process Z at an indeterminate point x_i and can be modelled as

$$z_i = \zeta(x_i) + e_i, \quad \text{where} \quad \zeta(x_i) = \rho\eta(x_i, \theta) + \delta(x_i) \tag{2.42}$$

Here z_i is an observation of true value of the real process $\zeta(x_i)$ for the known value of input variable x_i and e_i is the observation error. The output from the computer code is $\eta(x_i, \theta)$ but the true values of θ are unknown therefore θ is replaced with t for the computer output such that $\eta(x, t) = \mathbf{y}$ denotes the output of the computer model for given values of $x = (x_1, x_2 \dots x_n)$ and $t = (t_1 \dots t_n)$. Here, t is the set of input to the computer model for true unknown calibration parameter $\theta = (\theta_1, \dots \theta_n)$ and $\delta(.)$ is a model inadequacy function, independent of the code output $\eta(x, t)$. Equation (2.42) is a link between reality and observation.The quantity ρ is the scaling parameter that accounts for systematic discrepancy between the reality and computer output and e_i is the observational error for ith observation given as $e_i \sim \mathcal{N}(0, \sigma^2)$. The further developments of this work comprise of finding the priors associated with the functions $\eta(x, \theta)$ and $\delta(x)$ followed by finding the posterior distribution of the calibrated parameter θ. Finally the predictions can be made at unknown points with the use of posterior predictive distribution.

The unknown functions $\eta(x, \theta)$ and $\delta(x)$ in Equation (2.42) are assumed to have priors represented by Gaussian processes. To begin with posterior distribution, complete data is needed that include the observed and the computer output and can be represented $\mathbf{d}^T = (\mathbf{z}^T, \mathbf{y}^T)$. Af-

ter evaluating the form of distribution of d , it is possible to write the the posterior distribution as a product of likelihood $d|\theta, \beta, \phi$ and prior distributions, where ϕ is vector of hyperparameters.

$$\pi(\theta, \beta, \phi|d) \propto \pi(d|(m_d(\theta), V_d(\theta))\pi(\theta)\pi(\phi) \tag{2.43}$$

The details of all the developments are given in Appendix (A). Once, hyper-parameters are estimated, posterior distribution of calibrated parameter is established, conditioned on these estimated parameters

$$\pi(\theta|\phi, d) \propto p(d|\phi, \theta)p(\phi)p(\theta)$$

In order to make inference about $z(x)$, the posterior distribution of $z(.)$ is conditioned on estimated parameters ϕ and the calibration parameter θ is acquired using numerical computation methods.

2.5.1 `SAVE PACKAGE` Results for Transistor data analysis

In order to implement the calibration model for the predictions of observed data in presence of calibration parameter, `SAVE` package developed by [54], is employed. The analysis is carried out in three steps, at the first stage an object of the class `SAVE` is created that uses the package `DiceKriging` which fit the emulator of the computer model and help specifying the prior of calibration parameter. The second stage creates an object `bayesfit` which pertains to the estimation of posterior distribution of field and bias precisions along with that of calibration parameter using prior information from the object `SAVE`. As a last step, validation of the untried configuration of controllable parameters is done, with the aid of object `validate`. It produces bias corrected and pure model predictions along with the corresponding tolerance bounds.

The data used in this analysis is the Circuit-Simulator data explained in Section (2.2.3). Among the six input variables of transistor data $\mathbf{x} \in \mathbb{R}^6$, the input x_6 is set as calibration parameter in this setting in order to evaluate the effectiveness of the `SAVE` package. The data is divided into field and model data, comprising first 20 data points as training set and the rest of the 12 data points as the test data for validation. The field data doesn't include the calibration parameter x_6 whereby the model data includes the calibration parameter. This is because the calibration variable is considered to be unknown in the observed data and estimated by the calibration model [34]. The results for the posterior distributions of the precisions and the calibration parameter as well as the bias corrected predictions together with the tolerance bounds are exhibited in Figures (2.12, 2.13, 2.14).

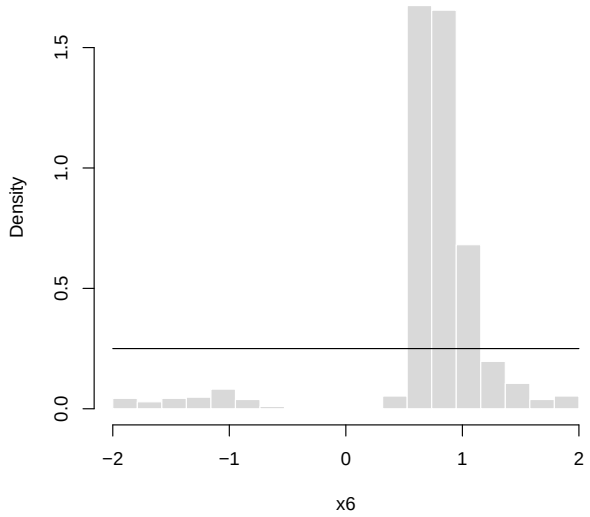


Figure 2.12: Posterior distribution of the calibration parameter for input x_6. The solid line corresponds to the prior used

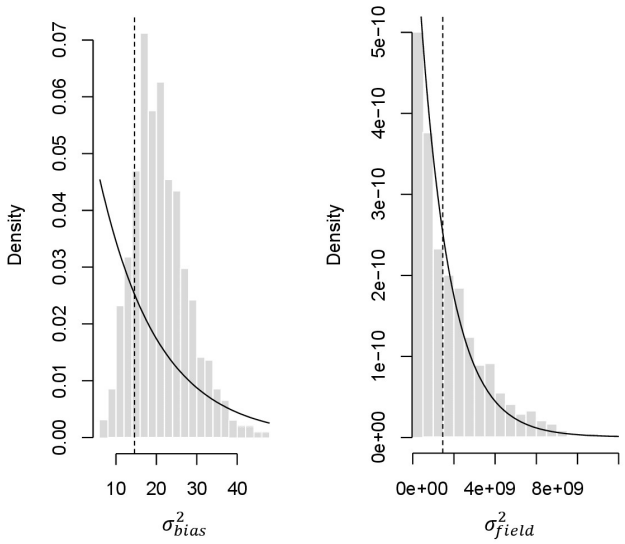


Figure 2.13: Posterior distribution of the bias and field precision. The dashed vertical line indicates the estimates and solid line represent the priors used

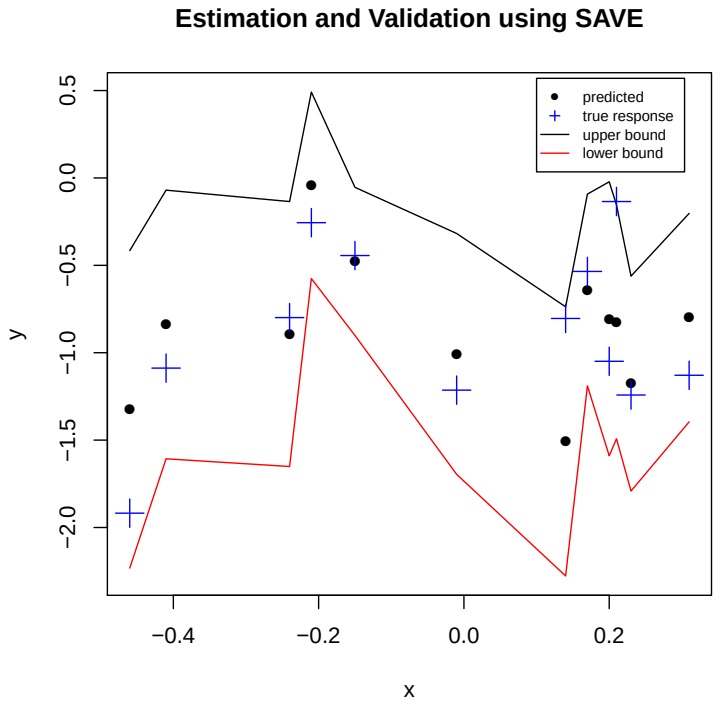


Figure 2.14: Bias corrected prediction plots. The solid lines represent 90% tolerance bounds whereby circles and + represent the observed and predicted field data respectively

It is evident from Figure (2.14) that with $SAVE$ package, the calibration model developed by [34] is able to capture information pertaining to calibrated parameter hence making predictions about the observed data with empirical mean square error to be 0.1387. The error may tend to increase, if true values of the calibrated parameter are not available, but only some expert guess.

2.6 NEURAL NETWORKS

In preceding sections we discussed in detail the DACE model and its variants for computer experiments. All these models are also termed as surrogate models attributed to their feature of providing an approximate to the true unknown function responsible to generate the response. One of the most widely adopted methodology in the realm of surrogate models in recent times is Neural networks. In this section we introduce neural networks and explain the simple framework of how they work. In Chapter (5) we will present comparative study of different models including neural networks for the choice of different designs. Although the accounts of

study of human brain dates back to thousands of years, the first step towards the concept of neural networks emerged in 1943 when [44] developed the first mathematical model of a neuron. In this model a single cell of the neural system processes the inputs that results in returning an output. An Artificial Neural Network (ANN) is a computational model and one of the most powerful algorithm that is propelled by the manner in which neural system in the human body processes data. Artificial Neural Networks are the new advancing avenues in the era of AI and encompasses a wide scope of exploration in context of computer models. Artificial Neural Networks have created a great deal of fervor in Machine Learning exploration and industry, because of numerous leap forward outcomes in computer vision, speech recognition and text processing. These artificial networks may be used for predictive modelling, adaptive control and applications where they can be trained via a dataset. The simple Artificial Neural Network called the Multi Layer Perceptron is explained with the help of simple neuron structure [87] and given in Figure (2.15).

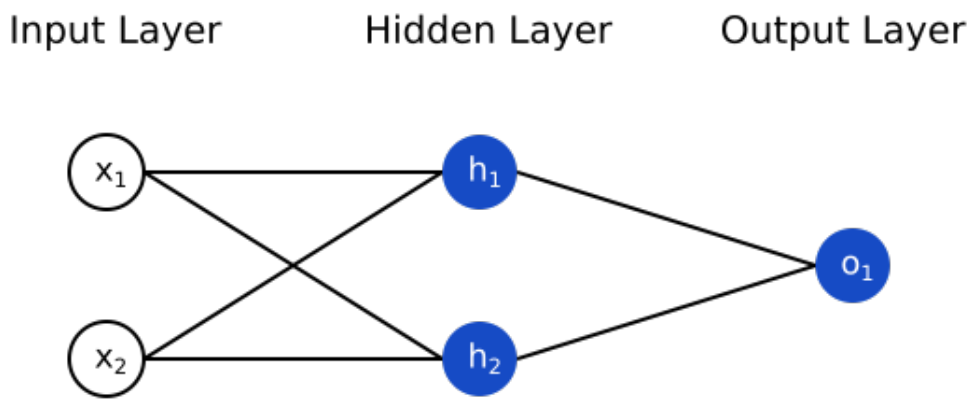


Figure 2.15: Neural Network Graphical presentation [87]

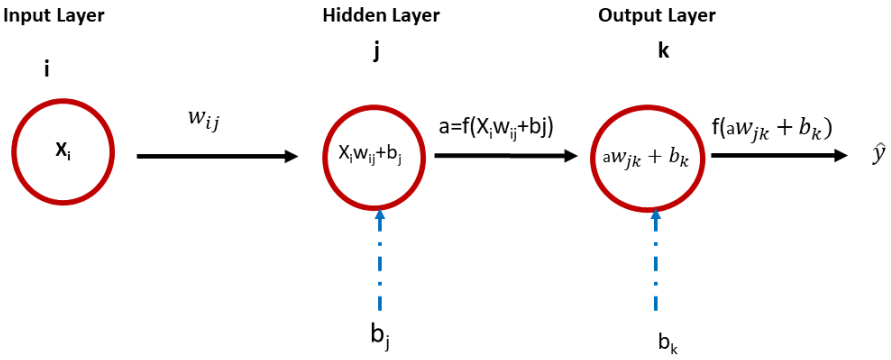


Figure 2.16: Neural Network Computations

Each circle in the figure (2.15) is called the node. All nodes are connected across layers but no two nodes of same layer are linked. Each node in the hidden layer is the computational site for processing the inputs. The hidden layer computations are depicted in figure (2.16). The subscripts i j and k represent the number of nodes in input layer, hidden layer and outer layer respectively. At a node of the hidden layer j, each input value x_i is multiplied by a corresponding weight w_{ij} and added to a bias b_j associated with jth hidden layer. The result is then passed through an activation function f generating the node's output. For the given structure of Neural Networks in figure (2.15) the input x_i will have one weight associated with it for one node of the hidden layer. Similarly for more than one nodes there will be different weight associated with each node. Between any two layers, weights will always form a matrix whose rows are equal to input nodes and columns are the output nodes. The hidden layer in matrix form can be expressed as $X = f(W^1 I + B^1)$, where W^1 is the matrix of weights from input nodes to hidden layer nodes, I is the vector of inputs and B is the vector of bias associated with each hidden node and f is the activation function. Similarly the final output of the network at the output node is obtained in a similar fashion. The most important question in Neural Network is: How do the link weights in a Neural Network get updated. In order to answer, one has to look into the difference between the target and the network output which is termed as error. This provokes a need to artifice a mathematical relationship between the errors and the weights which allow to minimize the errors and refine the network's weights thereof. One of the commonly used algorithms for this purpose is called Gradient Descent. Mathematical computations for figure (2.16) are explained in ythe following paragraph.

The input node comprises the inputs of the network represented as X_i , Z^{hidden} represent the weighted sum of the inputs and bias and a^{hidden} symbolizes the output of the hidden layer after applying the activation function. Also, W^1 and W^2 are the weights associated from inputs to hidden and from hidden to output layers respectively including the bias weights.

$$\begin{aligned} XW^1 &= Z^{(hidden)} \\ a^{hidden} &= f(Z^{hidden}) \end{aligned} \tag{2.44}$$

The final output of the network is given

$$\begin{aligned} Z^{output} &= a^{hidden}W^{(2)} \\ \hat{Y} &= f(Z^{output}) \end{aligned} \tag{2.45}$$

We want to investigate the change in errors by change in weights or specifically, the interest lies in minimizing the error function. For the sake of brevity identity function is used as activation function and the Z^{hidden} is replaced by Z. The error function is defined as

$$\begin{aligned} L &= (Y - \hat{Y})^T(Y - \hat{Y}) \\ L &= (Y - ZW^{(2)})^T(Y - ZW^{(2)}) \end{aligned} \tag{2.46}$$

The gradient of the error function with respect to the change in weights is the crux of the Neural network that defines the updating of weights so as to minimize the errors which is also called backpropagation. The first gradient is with respect to the change in weights $W^{(2)}$ and the updated weights are given

$$\begin{aligned} \nabla L_1 &= \frac{\partial L}{\partial W^{(2)}} \\ W^{(2)}_{upd} &= W^{(2)} - \alpha \nabla L_1 \end{aligned} \tag{2.47}$$

$W^{(2)}_{upd}$ is the updated weight matrix and α is the learning rate that monitors the amount to which the network weights are adjusted with respect to the loss gradient or in other terms it is the strength of change. Too low value of α leads to very slow gradient descent while a large value may miss the minimum. Therefore, it is important to decide upon the value of learning rate in order to obtain convergence. After updating weights for $W^{(2)}$ the gardient is needed to

update the weights $W^{(1)}$

$$\nabla L_2 = \frac{\partial L}{\partial W^{(1)}}$$
$$W^{(1)}_{upd} = W^{(1)} - \alpha \nabla L_2 \tag{2.48}$$

2.6.1 Simple Example

The framework of neural networks explained above is elaborated with the help of a simple example. For this study only one node is considered at input, hidden and output layer in order to make the network as simple as possible. The neural network is employed using `Neuralnet` package in `R`. In addition, the neural network is also build with the aid of `R` codes in order to get a deeper understanding of the network functioning. The data assumed is given -2,-1.5,0,1.5,2 The graphical output of the `Neuralnet` package is given in Figure (2.17).

x	y
-1	-2
-0.5	-1.5
0	0
0.5	1.5
1	-5

Table 2.6: Training data for Neural Networks

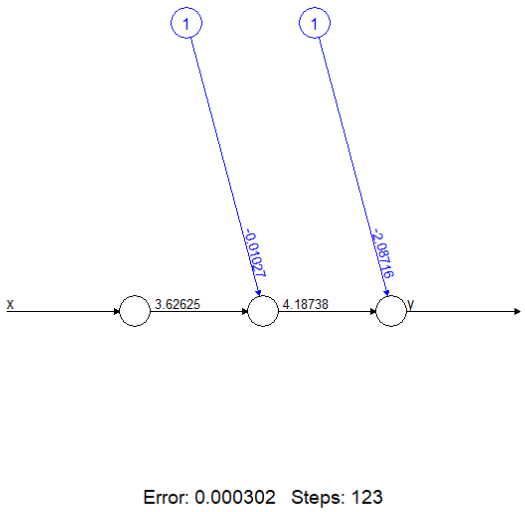


Figure 2.17: Neuralnet Results

The black lines in Figure (2.17) show the connections between each layer and the the numerical values on black lines are the weights associated with each neuron. The bias term added to each layer is given by blue lines. The Steps in Figure (2.17) shows that the network converged after 123 steps after achieving the minimum error of 3.02×10^{-4}.

2.6.2 TEMPERATURE DATA ANALYSIS USING NEURAL NETWORK

The temperature data described in Section (2.2.4) is analysed with the aid of neural network. The analysis is done using package `neuralnet` in `R`. A random sample of 100 data points for the year 2010 is used as a training set for neural network and a random sample of size 100 is used as test data. The temperature of January for the year 2010, is regressed on latitude and longitude assuming linear relationship in Neural Network. The predictions thus obtained are compared with the target observations and mean squared prediction error is computed as $\frac{\sum(y_i - \hat{y}_i)^2}{100}$. The resulting plot of Neural Network with associated weights and biases is given in Figure (2.18).

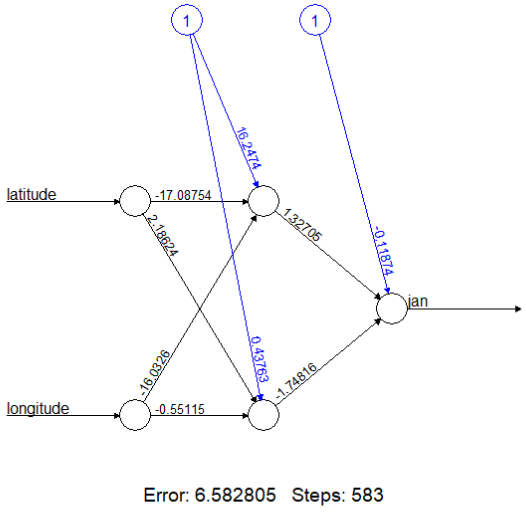


Figure 2.18: Neural Network for Temperature data

2.6.3 Comments

The first two nodes in Figure (2.18) represent the inputs which are latitude and longitude respectively. The next layer is the hidden layer with two nodes and the last layer with one node is the network output. Each directed arrow represents the updated weight from one node

of the layer to each node of the next layer. The blue arrows show the bias associated with each node. We noticed that the empirical mean square error of predictions for temperature data using neural network is 0.162 and 0.081 using *DiceKriging*. We train the neural network with only one hidden layer in this example. The performance of neural network is expected to improve if we increase the number of layers. The predictions at untried points using DACE model and Neural networks against the observed temperature is presented in Figure (2.19).

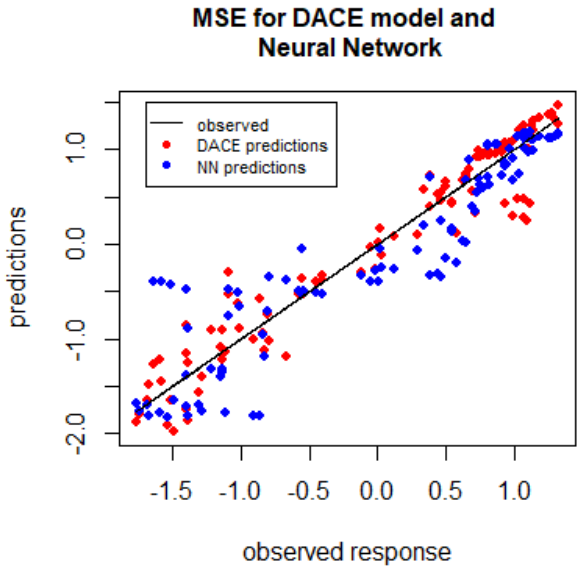


Figure 2.19: MSE for Temp data using DACE model and Neural Network

Chapter 3

Smooth Ridge Model

In Statistics, smoothness of a data set bears great importance when the purpose is to approximate the underlying function in an attempt to capture the meaningful patterns in data while removing noise and random phenomena. Smooth interpolation is a powerful concept in the modern age of data science and machine learning. Several variants of smooth models are available in literature. A correspondence of Bayesian estimation and smoothing splines is accounted in [36], while variational approach to splines can be found in [46]. A significant contribution in the realm of smoothing splines is made in the recent times by [59], where different modelling approaches to the identification of smoothing spline ANOVA models are discussed, namely: Classical, State-Dependent Regression (SDR) which is a non parametric approach based on recursive filtering and smoothing estimation and Adaptive Component Selection and Selection Operator (ACOSSO); a new regularization method for simultaneous model fitting and variable selection in non-parametric regression models in the framework of smoothing spline ANOVA. These methods are compared among themselves and with Design and Analysis of Computer Experiments (DACE) in terms of computational cost and out of samples prediction. It is concluded with the help of different examples that no single method outperforms in all situations but the performance is based on the underlying model. An advanced adaptation of this literature comprises Smooth supersaturated models introduced by [4]. The two subjects of interest in this paper include: Algebraic theory that helps in choosing the extended basis and the ways to construct statistical models based on smooth polynomial interpolators. It is also shown that extended model basis followed by optimizing a measure of smoothness provides a mean to construct optimal interpolators that outperforms the kriging-based method used in computer experiments. The significance of such models is recognized in signal processing by [41].

This chapter is aimed at introducing the methodology to propose a new predictive model

named Smooth Ridge (SR) model along with the review of underlying theory which provides basis for such development. The chapter is comprised of following sections. Section (3.1) provides a brief overview of simple regression model, ridge regression and supersaturated model along with the notations employed. The methodology to construct Smooth Ridge model is explained in Section (3.2). The variance comparison of Smooth Ridge estimator with that of the regression estimate is given in Section(3.3). A detailed account of the theoretical results for Mean Square Error of the parameter is presented in Section (3.4) followed by the emulator development and mean square error of Smooth Ridge model furnished in Section (3.5). The preliminary comparisons with simulations of synthetic data are dispensed in Section (3.6) followed by real life application of COVID in Section (3.7). Sensitivity Analysis for Smooth Ridge model is provided in Section (3.8) followed by the Sensitivity Indices in Section (3.9). A Bayesian development of Smooth Ridge model is introduced in Section (3.10). The chapter is concluded with conclusions in Section (3.11).

3.1 Review of models

Before stating the elements of proposed model it is important to recapitulate some models that provide a framework to formulate the foundations of Smooth Ridge model. For a design model matrix F and response variable Y, a linear relationship is commonly assumed between Y and the covariates $\mathbf{x}$ given in matrix form $Y = F\beta + \epsilon$, where $\beta = (\beta_1, \ldots \beta_k)$ is vector of regression parameter. The $\epsilon = (\epsilon_1, \ldots \epsilon_n)$ is the random error and attributed to unexplained part of the response not explained by $F\beta$. The random error is assumed to follow $\epsilon \sim \mathcal{N}(0, \sigma^2 I_n)$. The randomness of ϵ implies the randomness of Y which is distributed as multivariate normal $Y \sim \mathcal{N}(F\beta, \sigma^2 I_n)$.
The maximum likelihood estimator of the regression parameter is the standard formula

$$\hat{\beta} = (F^T F)^{-1} F^T Y \tag{3.1}$$

The expectation and variance-covariance matrix of the estimator $\hat{\beta}$ are

$$E(\hat{\beta}) = \beta \quad \text{and} \quad Var(\hat{\beta}) = \sigma^2 (F^T F)^{-1}$$

The above computations are possible only if $(F^T F)^{-1}$ is well defined i.e. $(F^T F)$ is invertible. In other words there is no collinearity among the covariates. Here covariates are the columns of F and collinearity implies that two or more covariates are linearly related. This is a common

occurring problem in high dimensional data resulting in increased uncertainty that reflects in the larger error of regression estimates attributed to the collinear covariates. In terms of dimensions, collinearity associated with $(n \times k)$ design matrix such that $n < k$ infers that the $(k \times k)$ matrix $F^T F$ has rank less than k. This phenomenon leads to singularity problem, which implies that the design model matrix has zero determinant hence non-invertible. Explicitly, if $k > n$, the unknown regression coefficients $\hat{\beta}$ of length k cannot be estimated uniquely with system of n linear equations.

One of the solution to overcome the problem of singular matrix $F^T F$ and hence obtain estimates of regression parameters β is ridge regression estimator introduced by [33]. The ridge estimator is defined as

$$\hat{\beta}_\lambda = (F^T F + \lambda I_k)^{-1} F^T Y \tag{3.2}$$

where, $\lambda \in [0, \infty)$ is ridge parameter also known as regularization parameter. Equation (3.2) provides a well defined estimator of β even in the presence of singularity problem of $F^T F$. The equation (3.2) suggests the dependence of $\hat{\beta}_\lambda$ on λ. The set of all ridge estimates $\hat{\beta}_\lambda$ is termed the *ridge trace*. For limiting case of λ, we have:

$$\lim_{\lambda \to \infty} \hat{\beta}_\lambda = 0 \quad \text{and} \quad \lim_{\lambda \to 0} \hat{\beta}_\lambda = \hat{\beta}$$

It is important to note that $\hat{\beta}_\lambda$ is a biased estimator of β since $E(\hat{\beta}_\lambda) = (F^T F + \lambda I)^{-1} F^T F \beta$, however, it can be shown that there exist value of λ such that the mean square error of ridge estimator $\hat{\beta}_\lambda$ is less than that of standard regression estimator $\hat{\beta}$ [24].

3.1.1 Smooth Supersaturated model (SSM)

The Smooth Ridge model borrows its elements from Ridge regression model that helps finding the estimate of SR estimator. In addition, the methodology employed in development of Smooth Ridge model is based on the theory of smooth supersaturated model explained by [4], that provides a bases of smooth interpolation therefore a brief overview of smooth supersaturated model is revisited.

For a set of factors $\mathbf{x} = (x_1, x_2, \ldots x_d)$ and a set of positive integers $\alpha = (\alpha_1, \alpha_2, \ldots \alpha_d)$, a monomial is defined as $x^\alpha = x_1^{\alpha_1}, \ldots x_d^{\alpha_d}$ wherein the linear combination of monomials form a polynomial. The algebraic theory in [55] states that a saturated model is the one such that, for an indeterminate point $\mathbf{x}$ in χ there exists a saturated non-singular monomial basis $B_M = \mathbf{x}^\alpha, \alpha \in M$. Here, M is some set of distinct index vector α, such that the size of basis is equal to the design size n i.e. $|M| = n$ and the design model matrix $F = (\mathbf{x}^\alpha_{x \in \mathscr{D}, \alpha \in M})$ is non-

singular. Such basis are called good saturated basis attributed to the fact that they provide exact interpolation for a given set of observations $\mathbf{y} = (y(x_1), \ldots y(x_n))$. Supersaturated models are built on supersaturated basis which is an extension of saturated basis where the design $\mathscr{D}$ is fixed but the model takes a larger set of monomials i.e. $|M| > n$. The following definition from [4] establishes basic objects for our development.

Definition 3.1. *(1) A finite set of monomials B is called a hierarchical basis if, for any monomial x^α in B, all its divisors are in B*
(2) Given a design $\mathscr{D}$ of size n, a good supersaturated basis is a basis $B_M = \{\mathbf{x}^\alpha, \alpha \in M\}$ with $|M| = N > n$ such that there is a hierarchical non-singular sub-basis of size n.

A basis that satisfies the definition above is termed a good basis. Strictly speaking, a basis has to be a hierarchical basis before it is inspected for identifiability. Consider the following simple example for a clear understanding. The bold notation is avoided for the sake of simplicity in the examples and the model development.

Example 3.1. *Consider a single factor $d = 1$ and $M = \{1, 2, 3, 4\}$. We have $|M| = 4$ and $B_M = \{x,\ x^2,\ x^3,\ x^4\}$. Let there be $n = 3$ design points, and an attempt is to find hierarchical non-singular basis of size n. Evaluating the basis B_M at design points results into:*

design	basis			
	x	x^2	x^3	x^4
-1	-1	1	-1	1
0	0	0	0	0
1	1	1	1	1

There are four possible sub-basis of size 3 from the set B_M which needs to be checked to fulfil the definition of good basis.

sub-basis				identifiable
1	x	x^2	x^3	No
2	x	x^2	x^4	No
3	x	x^3	x^4	No
4	x^2	x^3	x^4	No

Table 3.1: Sub-basis for $d = 1$ at $n = 3$ points

Since, none of the sub-basis for the given example is identifiable, therefore the given basis does not provide any good sub-basis.

Now consider another example with $d = 2$ *factors with* $n = 3$ *design points. Let* $M = \{(0,0),(0,1),(1,0),(1,1)\}$ *so that* $B_M = \{1, x_1, x_2, x_1x_2\}$. *The evaluation of* B_M *at design points* $(-1,-1),(0,1)$ *and* $(1,0)$ *gives the following table:*

design		basis			
		1	x_1	x_2	x_1x_2
-1	-1	1	-1	-1	1
0	1	1	0	1	0
1	0	1	1	1	0

The four possible sub-basis of size 3 *and the respective hierarchical as well as identifiable conditions are given as follows*

sub-basis			hierarchical	identifiable
1	x_1	x_2	Y	Y
1	x_1	x_1x_2	N	Y
1	x_2	x_1x_2	N	N
x_1	x_2	x_1x_2	N	Y

Table 3.2: sub-basis for $d = 2$ at $n = 3$ points

By definition (3.1), for a basis to be a good basis requires at least one hierarchical sub-basis. It can be seen from the table above that there is one hierarchical sub-basis of size $n = 3$ *when* $|M| = 4 > n$, *that is also non-singular. Therefore the basis* $B_M = \{1, x_1, x_2, x_1x_2\}$ *is a good basis*

3.1.2 Comment

In the above example with $d = 1$ we have one factor, had $M = \{0,1,2,3,4\}$, there would be good basis. Given the definition of supersaturated basis,the development of smooth interpolator can be achieved in the following fashion. For an indeterminate point $\mathbf{x}$ and real valued observations $\mathbf{y}$ let B_M be a good supersaturated basis, then the polynomial model in the basis is given by

$$y(x) = \sum_{\alpha \in M} \beta_\alpha x^\alpha \tag{3.3}$$

In order to make computations numerically stable and efficient, orthogonal polynomials instead of pure monomials are employed for the construction of basis and hence design model matrix. For an integration region $\Omega = [-1,\ 1]^d$, Legendre polynomials serves as the suitable choice of orthogonal polynomials with respect to uniform measure. The model in Equation (3.3) can then be rewritten as

$$y(x) = \sum_{\alpha \in M} \beta_\alpha L_\alpha(x) \tag{3.4}$$

where, $L_\alpha(x) = \prod_{i=1}^{k} L_{\alpha_i}(x_i)$. For one factor $d = 1$ and design $\mathscr{D}$, consider $M = (0, 1, 2, 3)$, then the Legendre polynomials are $L_0(x) = 1, L_1(x) = x, L_2(x) = (3x^2 - 1)/2, L_3(x) = (5x^3 - 3x)/2$.

We next characterize the polynomial model in terms of its roughness,following [4]. For a desired region $\chi \subset \mathbb{R}^d$ and an indeterminate $x \in \chi$ the Hessian of $y(x)$ is defined as

$$H(y(x)) = \{\frac{\partial^2 y(x)}{\partial x_i \partial x_j}\}$$

and the average roughness measure is given

$$\psi(y(x)) = \int_\chi ||H(y(x))||^2 dx \tag{3.5}$$

where, $||H(y(x))||^2 = trace(H(y(x))^2)$.

We propose smooth supersaturated model to be the one where the coefficients β_α in Equation (3.3) are selected to minimize the measure of roughness, hence named *smooth* proposed by [4]. The vector of model terms can be written as $f(x) = (x^\alpha : \alpha \in M)^T$ so that Equation (3.3) can be expressed in vector form $y(x) = f(x)^T \beta$. For any indeterminate point x denote $f^{(ij)} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ from which it follows that the measure of roughness in Equation (3.5) is

$$\psi(y(x)) = \beta^T K \beta, \tag{3.6}$$

where

$$K = \int_\chi \left(\sum_{i,j=1}^{d} f^{(ij)} f^{(ij)^T} \right) dx \tag{3.7}$$

It is pertinent to mention here that the matrix K may not be have full rank. It is because the constant and any non-linear model terms will have zero entries in K whereas the higher order basis will give positive entries to the matrix. The supersaturated model introduced by [4] is $\hat{y}(x) = \sum_{\alpha \in M} \hat{\theta}_\alpha x^\alpha$ where the coefficients $\hat{\theta}_\alpha$ are selected by minimizing the roughness

and solving the constrained minimization problem i.e.

$$\min_{\theta} \psi(y(x)) \quad \text{such that} \quad y_i = \hat{y}(x^i) \quad \text{for} \quad i = 1 \dots n. \tag{3.8}$$

The coefficient estimate thus obtained is $\hat{\theta} = (X^T X + K(1-P)K)^{-1} X^T Y$ where $P = (X^T (X^T X)^{-1} X)$.

Example 3.2. *We give a simple example showing the construction of K. For $d = 2$ factors, $M = \{(0,0),(1,0),(0,1),(2,0),(0,3)\}$ and $B_M = \{1, x_1, x_2, x_1^2, x_2^3\}$. the vector of model functions is*

$$f(x)^T = \begin{pmatrix} 1 & x_1 & x_1^2 & x_2 & x_2^3 \end{pmatrix}$$

The computations below follow from $f^{(ij)} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$, i.e. $f^{(12)}(x) = \frac{\partial^2 f(x)}{\partial x_1 \partial x_2}$. Therefore,

$$f^{(11)}(x)^T = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 \end{pmatrix},$$

$$f^{(12)}(x)^T = f^{(21)}(x) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \textit{and}$$

$$f^{(22)}(x)^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 6x \end{pmatrix}$$

Consider the polynomial model $y(x) = f(x)^T \beta$ with $\beta = \left(\beta_{(00)}, \beta_{(10)}, \beta_{(01)}, \beta_{(20)}, \beta_{(03)}\right)$. Therefore, the Hessian becomes

$$H = \begin{pmatrix} \beta^T f^{(11)} & \beta^T f^{(12)} \\ \beta^T f^{(21)} & \beta^T f^{(22)} \end{pmatrix}$$

$$||H||^2 = \begin{pmatrix} 4\beta_{(20)}^2 & 0 \\ 0 & 36\beta_{(03)} x_2^2 \end{pmatrix}$$

Therefore, $||H(y(x))||^2 = trace(H(y(x)))^2 = 4\beta_{(20)}^2 + 36\beta_{(03)} x_2^2$

Following the development above, one of the vector multiplication, $f^{(11)}(f^{(11)})^T$ for solving summation $\left(\sum_{i,j=1}^{d} f^{(ij)} f^{(ij)^T}\right)$ in Equation (3.7) is elaborated as

$$f^{11}(f^{11})^T = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 2 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

summing up all such matrices to find $\left(\sum_{i,j=1}^{d} f^{(ij)} f^{(ij)^T}\right)$ *results into following*

$$\left(\sum_{i,j=1}^{d} f^{(ij)} f^{(ij)^T}\right) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 36x_2^2 \end{pmatrix}$$

integrating the sum over the region $[0\ \ 1]$ *the matrix* K *in Equation (3.7) is*

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{pmatrix}$$

3.2 Smooth Ridge Model (SRM)

We propose a flexible model based on ridge regression in Equation (3.2) and measure of roughness K in Equation (3.7). The methodology pertaining to the new model is elaborated as follows:

For a given design $\mathscr{D}$, real valued observations $\mathbf{y}$ and good supersaturated basis B_M the polynomial regression model named *Smooth Ridge (SR) Model* in the basis is given in matrix form

$$Y = F\beta + e \tag{3.9}$$

where, β is a vector of the coefficients associated with the supersaturated basis. An important feature of supersaturated model is that the number of polynomials exceeds the number of design points i.e. $k > n$. For $f(x) = (x^\alpha :\ \alpha \in M)^T$, where M is a list of multi-indices such that $|M| = k$, the resulting $(n \times k)$ design model matrix is of the form $F = (f(x))^T$.

Consider a simple example where, $d = 2$, $M = \{(0,0),(1,0),(0,1),(1,1),(2,0),(0,2)\}$, with $k = 6$, then the vector of model terms can be expressed as $f(x) = (1\ \ x_1\ \ x_2\ \ x_1x_2\ \ x_1^2\ \ x_2^2)^T$ where, $f_1(\mathbf{x}) = 1, f_2(\mathbf{x}) = x_1 \ldots, f_6(\mathbf{x}) = x_2^2$ and the design model matrix is constructed with the model terms evaluated at n design points such that $F = (f_1(\mathbf{x})\ f_2(\mathbf{x})\ \ldots f_k(\mathbf{x}))$ The error e of the model in Equation (3.9) is assumed to follow the usual multivariate normal distribution in standard regression. Recall that for given data, least square estimation for

standard regression model is based on minimizing error sum of squares denoted by $SSE = e^T e$. A smooth regularization version of SSE is proposed in the following manner

$$\begin{aligned} L &= e^T e + \lambda \beta^T K \beta \\ L &= (Y - F\beta)^T (Y - F\beta) + \lambda \beta^T K \beta \end{aligned} \tag{3.10}$$

where λ is the regularization parameter and K is the matrix of size $k \times k$ given in Equation (3.7). Since, K is the measure of average roughness based on second derivative therefore the coefficients β are chosen to minimize L in Equation (3.10). This is done by expanding Equation (3.10), taking the first derivative of L and equating to zero

$$\begin{aligned} L &= Y^T Y - 2\beta^T F^T Y + \beta^T F^T F \beta + \lambda \beta^T K \beta \quad \text{so that} \\ \frac{\partial L}{\partial \beta} &= -2F^T Y + 2F^T F \beta + 2\lambda K \beta \end{aligned}$$

and equating the derivative to zero

$$\tilde{\beta}_\lambda = (F^T F + \lambda K)^{-1} F^T Y \tag{3.11}$$

The estimator $\tilde{\beta}_\lambda$ of β minimizes the linear combination of error sum of squares SSE and the measure of roughness K, hence named *Smooth Ridge(SR)* estimator. The resulting model is termed as *Smooth Ridge (SR) model*. The expectation and variance of the estimator $\tilde{\beta}_\lambda$ can be computed as

$$E(\tilde{\beta}_\lambda) = E\left((F^T F + \lambda K)^{-1} F^T Y\right) = (F^T F + \lambda K)^{-1} F^T E(Y)$$

As we know that $E(Y) = F\beta$, therefore

$$E(\tilde{\beta}_\lambda) = (F^T F + \lambda K)^{-1} F^T F \beta \tag{3.12}$$

Equation (3.11) can be simplified by writing $A = (F^T F + \lambda K)^{-1} F^T$ hence, $\tilde{\beta}_\lambda$ can be written as a linear combination of Y such that $\tilde{\beta}_\lambda = AY$. The variance of $\tilde{\beta}_\lambda$ can then be evaluated as

$$Var(\tilde{\beta}_\lambda) = Var(AY) = A\sigma^2 A^T \tag{3.13}$$

It can be observed from Equation (3.12) that $E(\tilde{\beta}_\lambda) \neq \beta$ therefore $\tilde{\beta}_\lambda$ is a biased estimator of model parameters β.

We now look into the limiting behaviour of the Smooth Ridge estimator $\tilde{\beta}_\lambda$ when $\lambda = \infty$

with the help of simple example here. Let a design $\mathcal{D} = \{-1, 0.5, 1\}$ of size $n = 3$ with $k = 5$ Legendre terms to compute design model matrix F. We do not need the response y as we compute the weights $(F^T F + \lambda K)^{-1} F^T$ given in Equation (3.11) to examine the coefficients. Recall that OLS estimation for the given example is possible for only two linear model terms when $n = 3$. The plot of the weights for each of the five coefficients associated with the five model terms are given in Figure (3.1) against different values of λ.

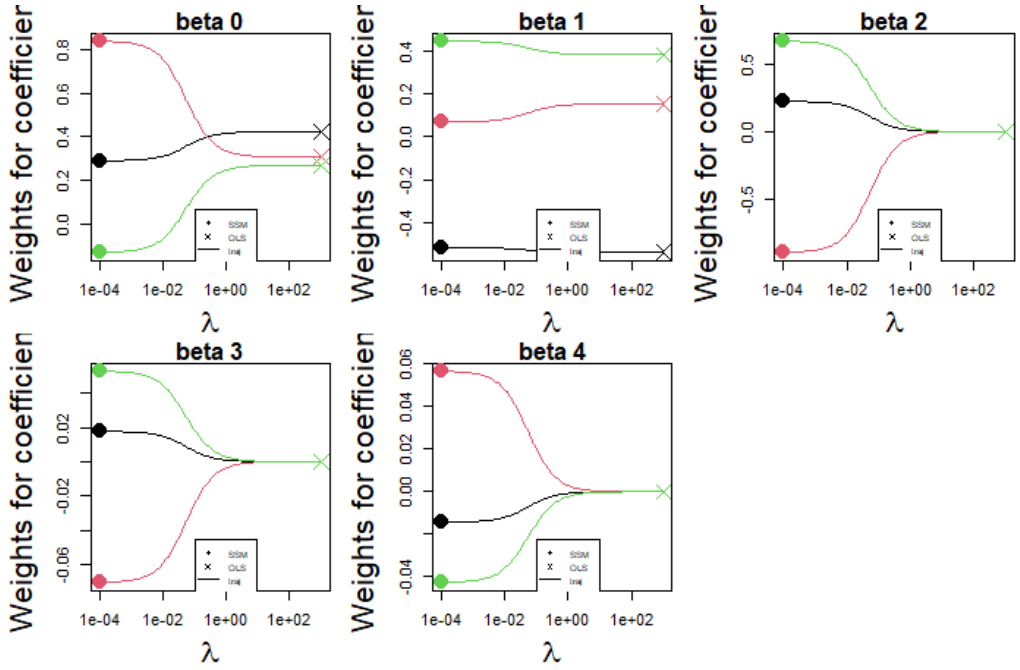


Figure 3.1: Coefficients for $\lambda \to \infty$

In Figure (3.1) there are five plots corresponding to five model terms which are rows of the weight matrix $(F^T F + \lambda K)^{-1} F^T$ and the three lines in each plot correspond to the three observations in the data. The solid dot in the Figure (3.1) is the coefficient weight for Smooth Ridge model and the star is the coefficient weight for OLS. We can see that the coefficients of β_2 and β_3 shrink to zero as λ increases. Therefore we deduce from this example that Smooth Ridge coefficients $\tilde{\beta}_\lambda$ converge to those of the OLS for linear model with intercept and linear terms only when $\lambda \to \infty$. We also observe an interesting feature of Smooth Ridge model when $\lambda = 0$. For the design model matrix that is full rank, Smooth Ridge model reduces to Regression model for $\lambda = 0$.

In the development of Smooth Ridge model we have a dual problem given in Equation (3.10), the error sum of squares $||Y - F\beta||_2^2$ and the curvature $\beta^T K \beta$, where λ determines the trade-off between these two. We want to look at the solutions to the Smooth Ridge regularized

regression problem which can be achieved by looking at the optimal trade-off curve [7] given in Figure (3.2). The coordinates of the plane X-Y are given by the two criteria $||Y - F\beta||_2^2$ and $\lambda\beta^T K\beta$ respectively. The shaded area is the set of achievable values ($||Y - F\beta||_2^2$, $\beta^T K\beta$). The blue dot is the coordinate referring to Smooth supersaturated model which minimizes the curvature $\beta^T K\beta$ given the constraint $Y = F\beta$. Therefore, the error sum of squares $||Y - F\beta||_2^2$ is zero. Similarly, the red dot is the point where curvature is zero and the dual criteria is reduced to error sum of squares only. The border of this area is termed the Pareto front, and the optimal trade-off curve is achieved by the solution of Equation (3.10).

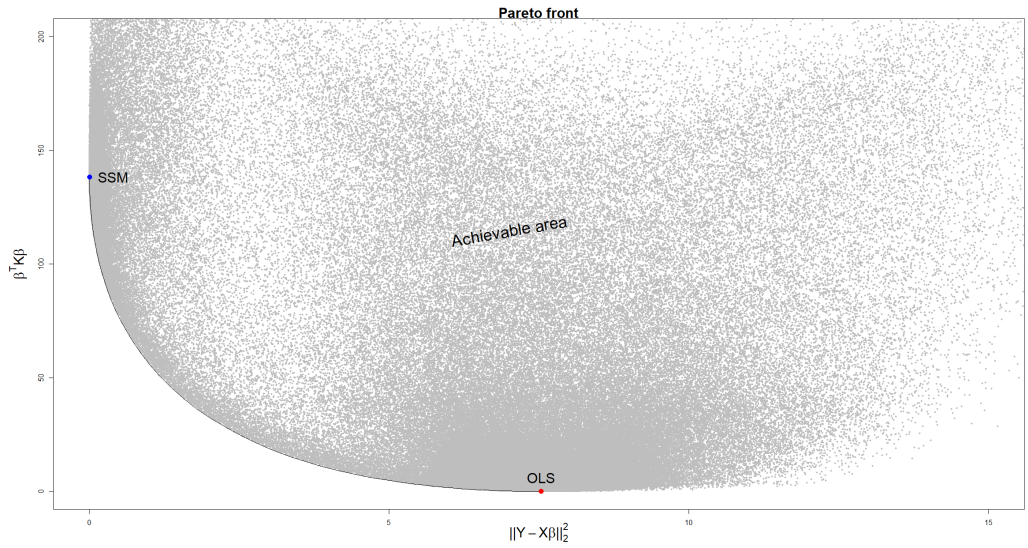


Figure 3.2: Plot of error sum of squares against curvature

3.3 Variance comparison of Smooth Ridge estimator with that of OLS estimator (Full rank design model matrix)

We are interested to compare the mean square error of smooth ridge estimator with that of linear regression estimator. A detailed theoretical comparison is provided with linear regression model and numerical results are given in the subsequent sections. To begin with, certain assumptions are to be made in order to build the theoretical results which are computationally feasible. First of which requires that $F^T F$ is invertible and the other demands K to be positive definite and invertible matrix. The explicit expression of Smooth ridge estimator is now developed with the help of a linear operator introduced by [79] for the purpose of development

of second moment of ridge regression estimator. The linear operator for smooth ridge model is defined as $U_\lambda = (F^TF + \lambda K)^{-1}F^TF$ such that $U_\lambda\hat{\beta}$ simplifies to $\tilde{\beta}_\lambda$, that is

$$U_\lambda\hat{\beta} = (F^TF + \lambda K)^{-1}F^TF(F^TF)^{-1}F^TY = \tilde{\beta}_\lambda \tag{3.14}$$

The regression estimator $\hat{\beta}$ is transformed to smooth ridge estimator $\tilde{\beta}_\lambda$ by using the linear operator U_λ. It is straightforward now from Equation (3.14) to simplify the expression of variance of $\tilde{\beta}_\lambda$.

$$V(\tilde{\beta}_\lambda) = U_\lambda V(\hat{\beta})U_\lambda^T = \sigma^2(F^TF + \lambda K)^{-1}(F^TF)^{-1}\left((F^TF + \lambda K)^{-1}\right)^T = \sigma^2AA^T \tag{3.15}$$

The result is obtained by employing the variance expression of regression estimator $Var(\hat{\beta}) = \sigma^2(F^TF)^{-1}$. With an explicit expression of $Var(\tilde{\beta}_\lambda)$, it is possible to compare it with $Var(\hat{\beta})$. The following development provides such a comparison.

$$\begin{aligned}
Var(\hat{\beta}) - Var(\tilde{\beta}_\lambda) &= \sigma^2[(F^TF)^{-1}) - U_\lambda(F^TF)^{-1}U_\lambda^T] \\
&= \sigma^2[U_\lambda U_\lambda^{-1}(F^TF)^{-1}(U_\lambda^{-1})^T(U_\lambda)^T - U_\lambda(F^TF)^{-1}U_\lambda^T] \\
&= \sigma^2U_\lambda[\underbrace{U_\lambda^{-1}(F^TF)^{-1}(U_\lambda^{-1})^T}_{Q_1} - (F^TF)^{-1}]U_\lambda^T
\end{aligned}$$

Since, $U_\lambda^{-1} = (F^TF)^{-1}(F^TF + \lambda K)$, thus solving for Q_1 gives the following

$$Var(\hat{\beta}) - Var(\tilde{\beta}_\lambda) = \sigma^2U_\lambda[2\lambda(F^TF)^{-1}K(F^TF)^{-1} + \lambda^2(F^TF)^{-1}K(F^TF)^{-1}K(F^TF)^{-1}]U_\lambda^T$$

Expanding U_λ, U_λ^T and simplifiying further provides the final expression for the difference

$$Var(\hat{\beta}) - Var(\tilde{\beta}_\lambda) = (F^TF + \lambda K)^{-1}[\underbrace{2\sigma^2\lambda K + \sigma^2\lambda^2K(F^TF)^{-1}K^T}_{Q_2}][(F^TF + \lambda K)^{-1}]^T \tag{3.16}$$

We next define positive(semi-) definite matrix and the resulting sum and product of the positive(semi-) definite matrices as follows

Definition 3.2.

1. *A symmetric $n \times n$ real matrix M is said to be positive(semi-) definite if for any non-zero column vector z of n real values, the scalar z^TMz is strictly positive/ (positive or 0).*

$$\begin{aligned}
M \text{ positive definite} \quad &\iff \quad z^TMz > 0 \text{ for all } z \in \mathbb{R}^n \setminus \mathbf{0} \\
M \text{ positive(semi-)definite} \quad &\iff \quad z^TMz \geq 0 \quad \text{for all } z \in \mathbb{R}^n
\end{aligned}$$

2. *The sum of two positive(semi-) definite matrices M and G is positive(semi-) definite matrix i.e.*

$$z^T(M+G)z = z^TMz + z^TGz \geq 0 \quad \forall z \in \mathbb{R}^n$$

3. *For symmetric real positive(semi-) definite matrices M and G the product M^TGM is positive definite and can be verified by*

$$z^tM^tGMz = b^tGb \geq 0 \quad \text{with} \quad b = Mz \quad \forall z \in \mathbb{R}^n$$

4. *If the difference of of two positive(semi-) definite matrices M and G is positive(semi-) definite matrix, it is represented as $M - G \succcurlyeq 0$ or $G \preccurlyeq M$*

The definition of positive(semi-) definite matrix in (3.2) helps to make conclusion about the variance difference evaluated in Equation(3.16). Since $(F^TF + \lambda K)^{-1}$ is square, invertible and positive(semi-) definite matrix and the sum (Q_2) in Equation(3.16) is positive(semi-) definite, therefore the difference of variances is positive(semi-) definite leading to the conclusion that variance of regression estimator $\hat{\beta}$ is greater than or equal to that of smooth ridge estimator which we write as

$$Var(\hat{\beta}) \succcurlyeq Var(\tilde{\beta}_\lambda)$$

The variance comparison provides a good reason to look into the mean square errors of the two estimators. For this purpose we first need to compute mean square error of $\tilde{\beta}_\lambda$ and $\hat{\beta}$. It can be readily observed that the mean square error of $\hat{\beta}$ is equal to variance attributed to the fact that bias$(\hat{\beta}) = 0$. Mean Square Error (MSE) of $\tilde{\beta}_\lambda$ is defined as

$$MSE(\tilde{\beta}_\lambda) = E\left((U_\lambda\hat{\beta} - \beta)^T(U_\lambda\hat{\beta} - \beta)\right) = E(\hat{\beta}^TU_\lambda^TU_\lambda\hat{\beta}) - E(\beta^TU_\lambda\hat{\beta}) - E(\hat{\beta}^TU_\lambda\beta) + E(\beta^T\beta)$$

adding and subtracting the terms $(\beta^TU_\lambda^TU_\lambda\hat{\beta})$ and $(\hat{\beta}^TU_\lambda^TU_\lambda\beta)$ and applying expectation yields:

$$\begin{aligned} MSE(\tilde{\beta}_\lambda) &= E[(\hat{\beta} - \beta)^TU_\lambda^TU_\lambda(\hat{\beta} - \beta)] + \beta^TU_\lambda^TU_\lambda\beta - \beta^TU_\lambda\beta - \beta^TU_\lambda^T\beta + \beta^T\beta \\ &= E[(\hat{\beta} - \beta)^TU_\lambda^TU_\lambda(\hat{\beta} - \beta)] + \beta^T(U_\lambda - I)^T(U_\lambda - I)\beta \quad \text{so that} \\ MSE(\tilde{\beta}_\lambda) &= \sigma^2 tr(U_\lambda(F^TF)^{-1}U_\lambda^T) + \beta^T(U_\lambda - I)^T(U_\lambda - I)\beta \end{aligned} \tag{3.17}$$

The above expression of MSE is obtained by using the results

$$\hat{\beta} \sim \mathcal{N}(\beta,\, \sigma^2(X^TX)^{-1}) \quad \text{and} \quad E(e^TAe) = tr(A\Sigma_e) + \mu_e^TA\mu_e \tag{3.18}$$

for quadratic form of multivariate normal distribution,where e is replaced by $\hat{\beta}$.

The foundation for MSE comparison is built on Theorem(2) of [77] which states that there exists $\lambda > 0$ such that mean square error of ridge regression estimator $\hat{\beta}_\lambda$ is less than the mean square error of simple regression estimator $\hat{\beta}$ or $MSE(\hat{\beta}_\lambda) \prec MSE(\hat{\beta})$. The following theorem is the smooth ridge version of the above statement, provided $(F^T F)$ is invertible

Theorem 3.1. *There exists $\lambda > 0$ such that $MSE(\tilde{\beta}_\lambda) \prec MSE(\hat{\beta})$*

Proof. Let $M(\lambda K)$ and $MSE_{(\lambda=0)}$ denote $MSE(\tilde{\beta}_\lambda)$ and $MSE(\hat{\beta})$ respectively, where $M(\lambda K)$ is given in Equation(3.17) and $MSE_{(\lambda=0)} = \sigma^2(F^T F)^{-1}$. Then

$$MSE_{(\lambda=0)} - M(\lambda K) = \sigma^2(F^T F)^{-1} - \sigma^2(U_\lambda(F^T F)^{-1}U_\lambda^T) - \beta^T(U_\lambda - I)^T(U_\lambda - I)\beta$$

Using the simplification of Equation(3.16), the difference can be simplified to

$$\begin{aligned} MSE_{(\lambda=0)} - M(\lambda K) = (F^T F + \lambda K)^{-1}\left(2\sigma^2\lambda K + \sigma^2\lambda^2 K(F^T F)^{-1}K^T\right)\left((F^T F + \lambda K)^{-1}\right)^T \\ - \lambda^2(F^T F + \lambda K)^{-1}K\beta\beta^T K^T\left((F^T F + \lambda K)^{-1}\right)^T \end{aligned}$$

which is simplified to

$$MSE_{(\lambda=0)} - M(\lambda K) = (F^T F + \lambda K)^{-1}\left(\underbrace{2\lambda\sigma^2 K + \sigma^2\lambda^2 K(F^T F)^{-1}K^T - \lambda^2 K\beta\beta^T K^T}_{Q_3}\right)\left((F^T F + \lambda K)^{-1}\right)^T \tag{3.19}$$

In summary mean square error of $\tilde{\beta}_\lambda$ is less than that of $\hat{\beta}$ if the difference in Equation (3.19) is positive(semi-) definite which is true if (Q_3) is positive(semi-) definite or

$$2\lambda\sigma^2 K + \lambda^2 K(F^T F)^{-1}K^T - \lambda^2 K\beta\beta^T K^T \succcurlyeq 0$$

It is straightforward to conclude that $\lambda^2 K(F^T F)^{-1}K^T$ is positive(semi-) definite. Consequently, It is sufficient to demonstrate that the inequality holds for

$$\frac{2}{\lambda}\sigma^2 K - K\beta\beta^T K^T \succcurlyeq 0$$

. From Theorem 3.2 introduced by [24], we are able to find a condition on λ such that the

inequality holds.

$$\frac{2}{\lambda}\sigma^2 K - K\beta\beta^T K^T \succcurlyeq 0$$
$$\lambda \preceq (\beta^T K^T (2\sigma^2 K)^{-1} K\beta) \tag{3.20}$$

which completes the proof □

3.4 MSE of smooth ridge estimator

Our prime interest is to perform MSE comparisons for Smooth Ridge model in the case where $F^T F$ is less than full rank. This is the case when number of model terms k exceed the number of design points n. Such a generalization of comparison for ridge and regression estimator is brought into light by [24], where $n < k$ results into condition when $F^T F$ becomes non-invertible. The following theorem by [24] provides a basis to derive condition where mean square error of Smooth Ridge estimator is less than that of Regression estimator.

Theorem 3.2 (Farebrother Theorem)**.** *For $n \times n$ positive definite matrix A; let b be $n \times 1$ non zero matrix and let d be a positive scalar. Then $dA - bb^T$ is (semi-) positive definite if and only if $b^T A^{-1} b \leq d$. We have a series of implications*

Proof. $dA - bb^T \succ 0$
$dc^T Ac > c^T bb^T c$ for an $n \times 1$ non-zero column vector c

$\frac{d}{b^T A^{-1} b} > \frac{(c^T b)^2}{c^T Ac \times b^T A^{-1} b}$ (dividing $b^T A^{-1} b$ on both sides and rearranging)

$\frac{d}{b^T A^{-1} b} > corr^2(A^{-\frac{1}{2}} b, A^{\frac{1}{2}} c)$

$\frac{d}{b^T A^{-1} b} > 1$ since, correlation is maximum at ± 1.

$d > b^T A^{-1} b$

The result is attributed to the fact that the inequality holds true in general if it is true at the maximum which concludes our proof. All the implications are reversible. □

We consider a simple example to examine the results given in Theorem(3.2).

Example 3.3. *Let* $A = \begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix}$ *and* $b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$. *Then,*

$$b^T A^{-1} b = \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} -1 & \frac{3}{2} \\ \frac{3}{2} & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 1 \tag{3.21}$$

which will be used in further comparisons. We consider two cases; $d > b^T A^{-1} b$ *and* $d < b^T A^{-1} b$.

- ***case 1:*** $d = \frac{1}{2}$ *which is less than 1. In this case*

$$dA - bb^T = \begin{pmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & -3 \end{pmatrix}$$

In order to check if $dA - bb^T \prec 0$ *let* c *be the vector of non-zero values* $c = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$. *It can be seen that* $c^T(dA - bb^T)c = -56$ *which is not positive. Hence,* $(dA - bb^T)$ *is not positive definite for* $d < b^T A^{-1} b$.

- ***case 2:*** $d = 2$ *which is greater than 1. In this case*

$$dA - bb^T = \begin{pmatrix} 3 & 4 \\ 4 & 0 \end{pmatrix}$$

and for $c = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$ *It can be seen that* $c^T(dA - bb^T)c$ *is positive supporting the result that* $(dA - bb^T)$ *is positive definite for* $d > b^T A^{-1} b$.

The Farebrother theorem helps in development of Smooth Ridge version of the mean square error comparison and given in the following theorem.

Theorem 3.3. *For an* $n \times k$ *design matrix* F *where,* $n < k$, *there exists a constant* λ *satisfying the inequality* $\lambda < ((K\beta)^T (UU^T K^T)^{-1} K\beta)^{-1}$ *such that* $MSE(\tilde{\beta}_\lambda) \preccurlyeq MSE(\hat{\beta})$ *given the condition that* $\left(2UU^T K^T + KU\Lambda^{-1} U^T K^T\right) - K\beta\beta^T K^T$ *is positive(semi-)definite.*

Proof. Let $F^T F$ be less than full rank matrix and K be the matrix defined in Equation (3.7), then mean square error of smooth ridge estimator $\tilde{\beta}_\lambda$ is

$$MSE(\tilde{\beta}_\lambda) = V(\tilde{\beta}_\lambda) + (bias(\tilde{\beta}_\lambda))^2$$

where

$$V(\tilde{\beta}_\lambda) = \sigma^2 (F^T F + \lambda K)^{-1} (F^T F)[(F^T F + \lambda K)^{-1}]^T$$

and

$$bias(\tilde{\beta}_\lambda) = E(\tilde{\beta}_\lambda) - \beta = (F^TF + \lambda K)^{-1}(F^TF)\beta - \beta = -\lambda(F^TF + \lambda K)^{-1}K\beta$$

Therefore, the $MSE(\tilde{\beta}_\lambda)$ can be expressed as

$$MSE(\tilde{\beta}_\lambda) = (F^TF + \lambda K)^{-1}\left[\sigma^2(F^TF) + \lambda^2 K\beta\beta^T K^T\right][(F^TF + \lambda K)^{-1}]^T \tag{3.22}$$

Recall from Equation (3.11) that $\tilde{\beta}_\lambda = (F^TF + \lambda K)^{-1}F^TY$. The design matrix F is $n \times k$ matrix of polynomial functions, Y is $n \times 1$ vector of observations, and F has rank $m < k$. We consider singular value decomposition of matrix F such that $F = U\Sigma V^T$ where U and V are the matrices containing left and right singular vectors of F respectively with Σ being matrix containing non-zero singular values. Then the matrix F^TF can be decomposed as

$$F^TF = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U^T \\ V^T \end{bmatrix} = U_m \Lambda U_m^T \tag{3.23}$$

which is the result of Karhunen-Loeve decomposition [30] where U_m and V_m are $k \times m$ and $m \times k$ sub matrices associated with non-zero singular values and Λ is an $m \times m$ diagonal matrix and $[U_m \;\; V_m]$ is a $k \times k$ orthonormal matrix. The smooth ridge estimator and regression estimators can therefore be represented using Karhunen-Loeve decomposition. For simplicity, U is substituted for U_m for development of following results.

$$\begin{aligned} \tilde{\beta}_\lambda &= (F^TF + \lambda K)^{-1}F^TY \\ &= (U\Lambda U^T + \lambda K)^{-1}F^TY \\ \hat{\beta} &= (F^TF)^- F^TY \\ \hat{\beta} &= (U\Lambda^{-1}U^T)F^TY \end{aligned} \tag{3.24}$$

where, $(F^TF)^{-1} = U\Lambda^{-1}U^T$ is Moore-Penrose generalized inverse of F^TF. Let $MSE_{(\lambda=0)}$ and MSE_λ be the mean square errors of $\hat{\beta}$ and mean square error of $\tilde{\beta}_\lambda$ in Equation (3.22). We re-write the mean square errors using the decomposition in (3.23).

$$\begin{aligned} MSE_{(\lambda=0)} &= \sigma^2 U\Lambda^{-1}U^T \\ MSE_\lambda &= (U\Lambda U^T + \lambda K)^{-1}[\sigma^2 U\Lambda U^T + \lambda^2 K\beta\beta^T K^T](U\Lambda U^T + \lambda K)^{-1} \end{aligned} \tag{3.25}$$

Multiplying both sides of $MSE_{(\lambda=0)}$ in Equation (3.25), $(U\Lambda U^T + \lambda K)^{-1}(U\Lambda U^T + \lambda K)$ and

using the fact that $U^TU = I$, simplifies to following

$$MSE_{(\lambda=0)} = (U\Lambda U^T + \lambda K)^{-1}(U\Lambda U^T + \lambda K)(\sigma^2 U\Lambda^{-1}U^T)(U\Lambda U^T + \lambda K)^T((U\Lambda U^T + \lambda K)^T)^{-1}$$
$$MSE_{(\lambda=0)} = (U\Lambda U^T + \lambda K)^{-1}\sigma^2\left(U\Lambda U^T + 2\lambda UU^TK^T + \lambda^2 KU\Lambda^{-1}U^TK^T\right)(U\Lambda U^T + \lambda K)^{-1} \tag{3.26}$$

An expression for the difference between two mean square errors be developed as

$$MSE_{(\lambda=0)} - MSE_\lambda = \lambda^2(U\Lambda U^T + \lambda K)^{-1}\left[\sigma^2\left(\underbrace{\frac{2}{\lambda}UU^TK^T + KU\Lambda^{-1}U^TK^T}_{\mathcal{A}}\right) - \underbrace{K\beta\beta^TK^T}_{\mathcal{B}}\right] \times (U\Lambda U^T + \lambda K)^{-1} \tag{3.27}$$

Given that the difference $\mathcal{A} - \mathcal{B} \succcurlyeq 0$,we use Theorem (3.2) to prove our result. For the sake of simplicity it is sufficient to show that

$$\frac{2}{\lambda}UU^TK^T - K\beta\beta^TK^T \succcurlyeq 0$$

We use Farebrother theorem (3.2) to show that the above condition is satisfied if and only if

$$\frac{1}{\lambda} > (K\beta)^T(UU^TK^T)^{-1}K\beta$$

equivalently,

$$\lambda < ((K\beta)^T(UU^TK^T)^{-1}K\beta)^{-1}$$

□

which concludes the proof. Therefore, the result implies that for the constant λ which satisfies the inequality $\lambda < ((K\beta)^T(UU^TK^T)^{-1}K\beta)^{-1}$, we infer that $MSE(\tilde{\beta}_\lambda) \preccurlyeq MSE(\hat{\beta})$.

The following theorem by [15] helps to conclude that $\mathcal{A}$ is positive (semi-) definite.

Theorem 3.4 (Courant Fischer min-max Theorem)**.** *For an $n \times n$ symmetric matrix A with eigenvalues $\lambda_1(X) \leq \lambda_2(X) \leq \cdots \leq \lambda_k(X) \cdots \leq \lambda_n(X)$*

$$\lambda_1 \leq \frac{x^TAx}{x^Tx} \leq \lambda_n \quad \forall x \in \mathbf{C}^n\backslash\{0\}$$

One of the properties of this theorem is described in [23] which shows that eigen values of the sum of two positive(semi-) definite (PSD) matrices is PSD. Explicitly,

Lemma 1. *Let A and B be two symmetric matrices. Then*

$$\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A+B) \leq \lambda_k(A) + \lambda_n(B)$$

Theorem (3.4) and definitions (3.2), provide sufficient evidence to conclude that $\mathcal{A}$ is PSD. However, there are unknown quantities β and σ^2 in the expression of mean square error difference in Equation(3.27), therefore some simple numerical examples are studied to find if λ can be identified such that the difference of mean square error of two estimators is Positive (semi-) definite.

Example 3.4. *Upto this point we have used monomial bases. However we are interested in using Legendre polynomials which are linear combinations of monomials. Legendre basis are orthogonal basis and obeys the properties of good basis given in Def (3.1). For one dimension $d = 1$ design $\mathscr{D}$, an example as simple as with only $n = 2$ design points $(-1, 1)$ and $M_L = (0, 1, 2)$ is considered, then the Legendre polynomials are $L_0(x) = 1, L_1(x) = x$ and $L_2(x) = (3x^2 - 1)/2$ to build design model matrix* F. *The vector of unknown coefficient parameters of the model is $\beta = (\beta_0\ \beta_1\ \beta_2)^T$. We compute the difference of mean square error as follows: The matrix K for this example is*

$$K = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 18 \end{pmatrix} \quad \textit{so that}$$

$$K\beta\beta^T K^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 18^2\beta_2^2 \end{pmatrix}$$

The matrix U from kerhunen-Loeve decomposition is

$$U = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \textit{and} \quad KUU^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 18 & 0 & 18 \end{pmatrix}$$

Recall from Equation (3.23) that U is the sub-matrix associated with non-singular values of the singular value decomposition of F^TF. The difference $\frac{2}{\lambda}UU^TK^T - K\beta\beta^TK^T$ is

$$\frac{2}{\lambda}KUU^T - K\beta\beta^TK^T = \frac{2}{\lambda}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 18 & 0 & 18 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 18^2\beta_2^2 \end{pmatrix} = \frac{2}{\lambda}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 18 & 0 & 18(1 - 18\beta_2^2) \end{pmatrix}$$

For the difference to be positive(semi-) definite, the condition on λ *is*

$$\frac{2(18)}{\lambda} - 18^2\beta_2^2 > 0$$

which follows that mean square error difference is positive(semi-) definite for $\lambda < \frac{1}{9\beta_2^2}$*. Explicitly, this example is the numerical application of Theorm (3.3). The condition on* λ *helps us to infer that* $MSE(\tilde{\beta}_\lambda) \preccurlyeq MSE(\hat{\beta})$ *when inequality holds true.*

Example 3.5. *The above example is extended by adding one more model term such that* $k = 4$ *with* $n = 2$ *design points* $(x_1 = -1, x_2 = 1)$ *and* $M_L = (0, 1, 2, 3)$ *is considered, then the Legendre polynomials are* $L_0(x) = 1, L_1(x) = x$ *,* $L_2(x) = (3x^2 - 1)/2$ *and* $L_3(x) = (5x^3 - 3x)/2$ *to build design model matrix* F*. Then,* $\beta = (\beta_0\ \beta_1\ \beta_2,\ \beta_3)^T$ *is a vector of unknown coefficient parameters of the model. The computations for difference of mean square error are given in a similar fashion as above: The matrix* K *for this example is*

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 150 \end{pmatrix} \quad \textit{so that}$$

$$K\beta\beta^T K^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 324\beta_2^2 & 2700\beta_2\beta_3 \\ 0 & 0 & 2700\beta_2\beta_3 & 22500\beta_3^2 \end{pmatrix}$$

The matrix U *from kerhunen-Loeve decomposition is*

$$U = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & 1 \end{pmatrix} \quad \textit{and} \quad KUU^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 18 & 0 & 18 & 0 \\ 0 & 150 & 0 & 150 \end{pmatrix}$$

The difference $\frac{2}{\lambda}UU^T K^T - K\beta\beta^T K^T$ *is computed as*

$$\frac{2}{\lambda}KUU^T - K\beta\beta^T K^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{36}{\lambda} & 0 & \frac{36}{\lambda} - 324\beta_2^2 & -2700\beta_2\beta_3 \\ 0 & \frac{300}{\lambda} & -2700\beta_2\beta_3 & -22500\beta 3^2 \end{pmatrix}$$

In order to see if the matrix of difference is SPD, eigen values need to be evaluated for this matrix. Therefore the eigen values of the difference matrix above are $\mu_1 = \mu_2 = 0$,

$$\mu_3 = \frac{-162\beta_2^2\lambda - 11250\beta_3^2\lambda + 168 + 6\sqrt{729\beta_2^4\lambda^2 + 101250\beta_2^2\beta_3^2\lambda^2 + 3515625\lambda_3^4\lambda^2 + 1188\beta_2^2\lambda - 82500\beta_3^2\lambda +}}{\lambda}$$

and

$$\mu_4 = \frac{-162\beta_2^2\lambda - 11250\beta_3^2\lambda + 168 - 6\sqrt{729\beta_2^4\lambda^2 + 101250\beta_2^2\beta_3^2\lambda^2 + 3515625\beta_3^4\lambda^2 + 1188\beta_2^2\lambda - 82500\beta 3^2\lambda -}}{\lambda}$$

Let, $\beta_2 = 1$, $\beta_3 = 1$ *then straightforward calculations lead to conclude that* μ_3 *and* μ_4 *are non zero and positive for* $\lambda \geq 84$. *Therefore, for this example with* $n = 2$, $k = 4$, $\beta_2 = 1$, $\beta_3 = 1$ *and* $\lambda = 84$, *the eigenvalues are* $\mu_1 = \mu_2 = \mu_4 = 0$ *and* $\mu_3 = 5400$. *We notice that in the given example eigenvalues of the difference between mean square error matrix are not all positive for* $\lambda < 84$, *the non-zero eigenvalues are positive for* $\lambda > 84$ *and non-zero eigenvalues are either positive or zero at* $\lambda = 84$. *Therefore, a value of* λ *can be found such that the mean square error of ridge estimator is less than or equal to that of smooth regression estimator.*

Example 3.6. *We study one more example to observe how increasing the number of data points may effect the computation of eigenvalues of the difference matrix. We use* $k = 4$ *model terms and* $n = 3$ *design points* $x = (-1, 0, 1)$. *Let* $M = (0, 1, 2, 3)$, *then the Legendre polynomials are* $L_0(x) = 1, L_1(x) = x$, $L_2(x) = (3x^2 - 1)/2$ *and* $L_3(x) = (5x^3 - 3x)/2$ *to build design model matrix* F *and* $\beta = (\beta_0\ \beta_1\ \beta_2,\ \beta_3)^T$ *is a vector of unknown coefficient parameters of the model. The computations for difference of mean square error are given in a similar fashion as above: The matrix* K *for this example is the same as in previous example because* $k = 4$ *in both examples.*

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 150 \end{pmatrix} \quad \textit{so that}$$

$$K\beta\beta^T K^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 324\beta_2^2 & 2700\beta_2\beta_3 \\ 0 & 0 & 2700\beta_2\beta_3 & 22500\beta_3^2 \end{pmatrix}$$

The matrix U from kerhunen-Loeve decomposition is

$$U = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad and \quad KUU^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 36 & 0 \\ 0 & 150 & 0 & 150 \end{pmatrix}$$

The difference $\frac{2}{\lambda}UU^TK^T - K\beta\beta^TK^T$ *is computed as*

$$\frac{2}{\lambda}KUU^T - K\beta\beta^TK^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{72}{\lambda} - 324\beta_2^2 & -2700\beta_2\beta_3 \\ 0 & \frac{300}{\lambda} & -2700\beta_2\beta_3 & -22500\beta 3^2 \end{pmatrix}$$

In order to see if the matrix of difference is SPD, eigen values need to be evaluated for this matrix. Therefore the eigen values are $\mu_1 = \mu_2 = 0$,

$$\mu_3 = \frac{-162\beta_2^2\lambda - 11250\beta_3^2\lambda + 168 + 6\sqrt{729\beta_2^4\lambda^2 + 101250\beta_2^2\beta_3^2\lambda^2 + 3515625\lambda_3^4\lambda^2 + 1026\beta_2^2\lambda - 82500\beta_3^2\lambda +}}{\lambda}$$

and

$$\frac{-162\beta_2^2\lambda - 11250\beta_3^2\lambda + 168 - 6\sqrt{729\beta_2^4\lambda^2 + 101250\beta_2^2\beta_3^2\lambda^2 + 3515625\beta_3^4\lambda^2 + 1026\beta_2^2\lambda - 82500\beta 3^2\lambda + 484}}{\lambda}$$

Let, $\beta_2 = 1$, $\beta_3 = 1$ *then it is simple to evaluate that eigen values of the difference matrix above are all non-negative for* $\lambda \geq 80$. *Therefore, in this example for* $n = 3$, $k = 4$, $\beta_2 = 1$, $\beta_3 = 1$ *and* $\lambda = 80$ *the non-zero eigenvalues are positive,* $\mu_1 = \mu_2 = 0$, $\mu_3 = 6809.11$ *and* $\mu_4 = 126.88$. *We observe that the eigenvalues of the difference between mean square error matrix are not all positive for* $\lambda < 80$, *the non-zero eigenvalues are positive for* $\lambda \geq 80$. *Therefore, value of* λ *can be found such that the mean square error of smooth ridge estimator is less than or equal to that of regression estimator.*

The example given above, however, cannot be generalized to any number of design points and polynomials. We observe that increasing the regression parameters yield redundant equations for which reason a unique condition on λ cannot be obtained in general. However, the comparisons of models along with prediction mean square computation are provided in Section (3.5), where unknown regression parameters are replaced by their respective estimates using Smooth Ridge model.

3.5 Emulator and MSE results for smooth ridge model

Sections (3.2) and (3.4) provide a good reason to believe that the Smooth ridge estimator is efficient in terms of smaller variance and mean square error. This can further be examined in terms of prediction capabilities of the proposed model which requires to develop a predictive model and the associated prediction error. The polynomial model given in Equation (3.9) based on the estimator in Equation (3.11) at a new point x takes the form

$$\hat{y}_{\tilde{\beta}_\lambda} = f(x)^T \tilde{\beta}_\lambda \tag{3.28}$$

Recall from Equation (3.11) that $\tilde{\beta}_\lambda = (F^T F + \lambda K)^{-1} F^T Y$ which can be simplified as $\tilde{\beta}_\lambda = AY$, where, $A = (F^T F + \lambda K)^{-1} F^T$ which follows that we can write Equation (3.28) as $\hat{y}_{\tilde{\beta}_\lambda} = f(x)^T AY$ or simply $\hat{y}(x) = WY$ where $W = f(x)^T A$. Then we have the following theorem about the prediction mean square error for Smooth Ridge model.

Theorem 3.5. *The mean square error of the prediction $\hat{y}_{\tilde{\beta}_\lambda}$ at a new point x is given as*

$$MSE(\hat{y}_{\tilde{\beta}_\lambda}) = W\left(F\beta\beta^T F^T + \sigma^2\right) W^T - \sigma^2 \left(2W\mathbb{1} - I\right) - \left(2WF - f(x)^T\right) \beta\beta^T f(x)$$

where

$$\mathbb{1} = \begin{cases} 1, & \text{if } \epsilon_i = e \\ 0, & \text{otherwise} \end{cases}$$

Proof. The mean square error of $\hat{y}_{\tilde{\beta}_\lambda}$ by definition is

$$\begin{aligned} MSE(\hat{y}_{\tilde{\beta}_\lambda}) &= E\big(WY - Y(x)\big)^2 \\ MSE(\hat{y}_{\tilde{\beta}_\lambda}) &= E\big(WYY^T W^T - 2WYY(x) + Y(x)^2\big) \end{aligned} \tag{3.29}$$

The expectations in Equation (3.29) are evaluated as

$$\begin{aligned} E(WYY^T W^T) &= W(F\beta\beta^T F^T + \sigma^2)W^T \\ E(YY(x)) &= E(F\beta + \epsilon)(f(x)^T\beta + e) \\ E(YY(x)) &= F\beta\beta^T f(x) + E(\epsilon e) \end{aligned} \tag{3.30}$$

ϵ is a vector of order $n \times 1$ where e is a scalar. $E(\epsilon e) = \sigma^2$ for $\epsilon_i = e$ and 0 otherwise. Therefore,

$$E(YY(x)) = F\beta\beta^T f(x) + \sigma^2 \mathbb{1} \tag{3.31}$$

and finally expectation of $Y(x)^2$ is

$$E(Y(x))^2 = \sigma^2 + f(x)^T \beta\beta^T f(x) \tag{3.32}$$

Equations (3.30, 3.31 and 3.32), completes the proof for mean square error of $\hat{y}(x)$ i.e.

$$MSE(\hat{y}_{\tilde{\beta}_\lambda}) = WF\beta\beta^T F^T W^T + \sigma^2 WW^T - 2\sigma^2 W\mathbb{1} - 2WF\beta\beta^T f(x) + \sigma^2 + f(x)^T \beta\beta^T f(x) \tag{3.33}$$

□

3.5.1 Smooth Ridge predictor with Gaussian Emulator

Revisit the emulator in Equation (2.20), for sacks model (2.1)

$$\hat{y}(x) = f(x)^T \hat{\beta} + r(x)^T R^{-1}(Y - F\hat{\beta}) \tag{3.34}$$

An attempt here is to build an emulator for the Smooth Ridge model to make predictions at untried data points. This is done by introducing a modification in Equation (3.34) that suggests to replace $\hat{\beta}$ given in Equation (3.34) with $\tilde{\beta}_\lambda$ given in Equation (3.11). The resulting smooth predictor takes the form

$$\hat{y}_\lambda(x) = f(x)^T \tilde{\beta}_\lambda + r(x)^T R^{-1}(Y - F\tilde{\beta}_\lambda) \tag{3.35}$$

where $r(x)$ and R represent the quantities explained in section 2.2.1. Expanding $\tilde{\beta}_\lambda$ given in Equation(3.11), the emulator $\hat{y}_\lambda(x)$ is expressed as

$$\hat{y}_\lambda(x) = f(x)^T (F^T F + \lambda K)^{-1} F^T Y + r(x)^T R^{-1}(Y - F(F^T F + \lambda K)^{-1} F^T Y)$$

$\hat{y}(x)$ can be written as a linear combination of Y

$$\hat{y}_\lambda(x) = f(x)^T AY + r(x)^T R^{-1}(Y - FAY) = \left(f(x)^T A + r(x)^T R^{-1}(I - FA)\right) Y$$
$$\hat{y}_\lambda(x) = \left(f(x)^T A + r(x)^T R^{-1}(I - FA)\right) Y = CY \tag{3.36}$$

where $A = (F^TF + \lambda K)^{-1}F^T$ as defined in Section (3.2) and $C = f(x)^TA + r(x)^TR^{-1}(I - FA)$. We replace $\hat{y}_\lambda(x)$ to the random Gaussian variable $\hat{Y}_\lambda$, therefore $\hat{Y}_\lambda \sim \mathcal{N}(CF\beta, \sigma^2 CRC^T)$ as $Y \sim \mathcal{N}(F\beta, \sigma^2 R)$.

3.5.2 Mean Square Error of predictions for Smooth Ridge emulator

In this section we aim to develop an expression for mean square error of $\hat{y}_\lambda(x)$ following the development of Smooth Ridge emulator Equation (3.36) in Section (3.5.1). The following theorem elucidates the construction of mean square prediction error of Smooth Ridge emulator

Theorem 3.6. *Mean square error of the emulator $\hat{y}_\lambda(x)$ is*

$$MSE(\hat{y}_\lambda(x)) = CF\beta\beta^TF^TC^T + \sigma^2CRC^T - 2\sigma^2Cr(x) - 2CF\beta\beta^Tf(x) + \sigma^2 + f(x)^T\beta\beta^Tf(x)$$

Proof. We begin with the definition of mean square error which is the expectation of the squared deviations

$$MSE(\hat{y}_\lambda(x)) = E\big(CY - Y(x)\big)^2 = E\big(CYY^TC^T - 2CYY(x) + Y(x)^2\big) \tag{3.37}$$

The expectations in Equation(3.37) are evaluated in the following manner

$$E(CYY^TC^T) = CE(YY^T)C^T = C(F\beta\beta^TF^T + \sigma^2R)C^T \tag{3.38}$$

Recall that $Y = F\beta + Z$ with $E(Y) = F\beta$ and $cov(Y) = \sigma^2R$. Similarly,

$$E(CYY(x)) = Cov(CY, Y(x)) + E(CY)E(Y(x)) = \sigma^2Cr(x) + CF\beta\beta^Tf(x) \tag{3.39}$$

and finally for the last term of Equation (3.38)

$$E(Y(x))^2 = \sigma^2 + f(x)^T\beta\beta^Tf(x) \tag{3.40}$$

From Equations (3.38, 3.39 and 3.40),the mean square error of $\hat{y}(x)$ in Equation (3.37) is

$$MSE(\hat{y}_\lambda(x)) = CF\beta\beta^TF^TC^T + \sigma^2CRC^T - 2\sigma^2Cr(x) - 2CF\beta\beta^Tf(x) + \sigma^2 + f(x)^T\beta\beta^Tf(x) \tag{3.41}$$

which completes the proof. □

It is pertinent to mention here that $MSE(\hat{y}_\lambda(x))$ is zero at design points which leads to the result, $\hat{y}_\lambda(x) = y(x)$ at a design point s_j. This can be supported by taking into account that

$r(x)^T R^{-1} F$ reduces to $f(s_j)$ at design point s_j, since $r(s_j)^T R^{-1}$ becomes an indicator vector and chooses the row corresponding to s_j in the matrix F. Therefore, C in Equation(3.41)simplifies to identity I. Applying these results, MSE in Equation (3.41) becomes,

$$MSE(\hat{y}(s_j)) = f(s_j)^T \beta\beta^T f(s_j) + \sigma^2 - 2\sigma^2 - 2f(s_j)\beta\beta^T f(s_j)^T + \sigma^2 + f(s_j)^T \beta\beta^T f(s_j)^T$$

whose terms cancel to 0

.

3.6 Comparisons

In this section a brief account of comparisons of Smooth Ridge model with that of Sacks model in Section (2.1) is furnished with the aid of simulated data for one, two and three dimensions. A simple comparison is also evaluated under the assumption that true function generating the response is not Gaussian, therefore, the kriging estimation is not expected to perform well. Finally, a real life application of COVID data is employed to compare the predictive capabilities of models under study, where response variable is the probability of being infected by aerosole transmission of the virus. In order to make computations numerically stable and efficient, orthogonal polynomials instead of pure monomials are employed.

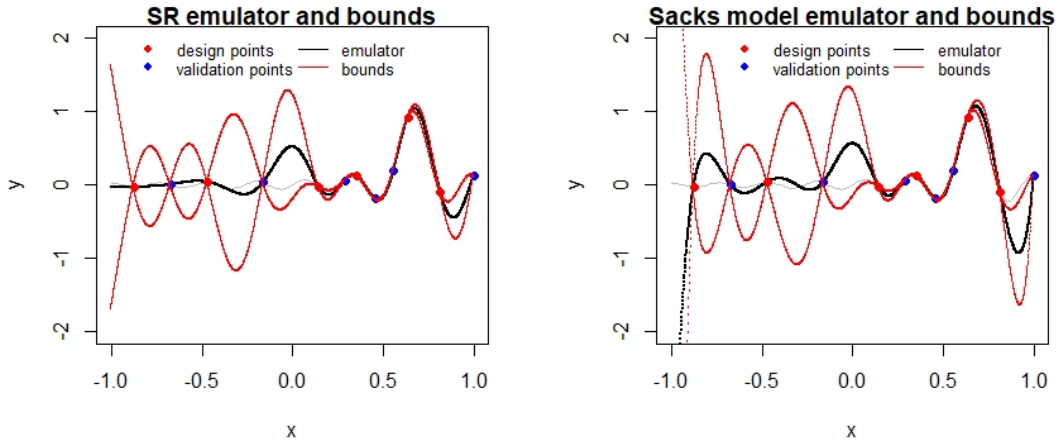
3.6.1 One dimension Example: comparison with Sacks model

In order to evaluate the performance of proposed model in terms of emulator given in Equation (3.35) and mean square error in Equation (3.41), we proceed in the following manner. For one dimension study, data is uniformly generated to build design model matrix F, using $k = 8$ Legendre Polynomial terms with $M = (0, 1, 2, 3, 4, 5, 6, 7)$. The following simulator is used to compute the response values $y(x)$ which is well known sinc function [85].

$$y(x) = \frac{sin(\frac{15\pi}{2}x - \frac{10\pi}{2})}{\frac{15\pi}{2}x - \frac{10\pi}{2}}$$

The six design points $n = 6$ for estimation of $\tilde{\beta}$ and six additional data points for validation are generated using uniform distribution. All 12 data points are then used for the computation of bias correction factor in Equation (3.35). The first step in the estimation of $\tilde{\beta}$ is to estimate λ which is done by employing the criterion of minimizing empirical mean square error, $\lambda = argmin \sum (Y - F\tilde{\beta})^2$ for a grid of $10,000$ values for λ in the region of $[10^{-5},\ 10]$. For estimated value of regularization parameter λ, the coefficient vector $\tilde{\beta}$ is computed thereof. For given

choice of design points, Legendre polynomial terms and minimization criterion the estimated value of λ is found to be $\hat{\lambda} = 0.0093$. For the corresponding coefficient vector $\tilde{\beta}$ the emulator given in Equation 3.35 and bounds for mean square error in Equation (3.41) are computed for 1500 untried points. The plots of emulator and bounds for Smooth Ridge model and Sacks model using Equations (2.20, 2.8) for comparison purpose are displayed in Figure (3.3)



(a) SR model Emulator and bounds (b) Sacks Emulator with bounds

Figure 3.3: Emulation and bounds Plots for SR and Sacks model

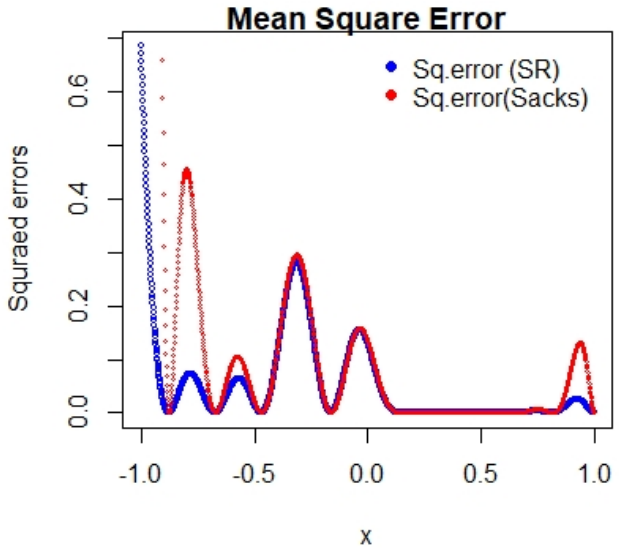


Figure 3.4: Squared prediction error for SR and Sacks model

It is noticeable from Figure (3.3) that prediction bounds for sacks model are wider at some points and the argument is supported by investigating into the mean square error plots.

In order to further verify the results a simulation study with 100 iterations for the same example is carried out by randomly generating design points and validation data using uniform distribution. Given below are the box plots for mean square error of both the models under study along with the average mean square errors.

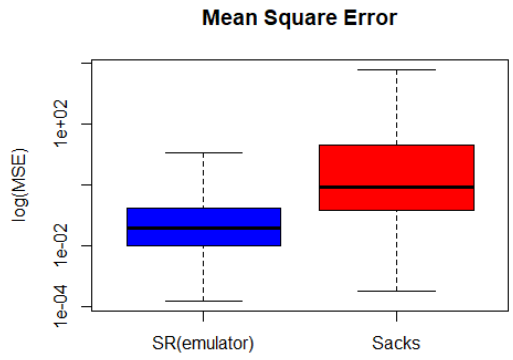


Figure 3.5: Box Plots of MSE for 1-D

Average Mean Square Error	
SR model	0.5774505
Sacks model	302.18

Table 3.3: Average MSE for two models

3.6.2 Bi- dimension Example: comparison with Sacks model

We now investigate further the performance of proposed model, is made by taking into account 2-dimension data. The comparison of SR model is made with Sacks model using bi-dimensional function. The design $\mathscr{D}$ is composed of uniformly generated 15 data points over the design region $[-1,\ 1]^2$. The data generated is divide into training and testing parts in $2:1$ with the aid of cross validation folds. The design model matrix F is $n \times k$ matrix constructed with $k = 12$ Legendre polynomial terms and $n = 10$ design points. The simulator is the bivariate function responsible to generate the response [4].

$$y(x_1, x_2) = sin((x_1 - 0.5)^2 + (x_2 - 0.5)^2 + 7x_1(x_2 - 0.5))$$

Mean square prediction errors for the given example is computed for 5 untried points for each of

- 200 random data set using uniform distribution
- 200 random data set with Latin Hypercube Sampling (LHS)

The graphical presentation of box plots for mean square error of the models under study along with the average mean square error for Uniform and Latin Hypercube samples are given in Figures (3.6, 3.7) and Tables (3.4, 3.5) respectively.

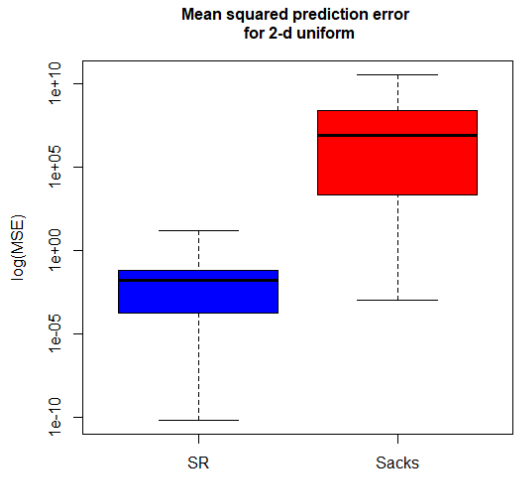


Figure 3.6: Box Plots of MSE 2D Uniform

Average Mean Square Error	
SR model	1.506×10^{-1}
Sacks model	1.02×10^{9}

Table 3.4: Average MSE 2D Uniform

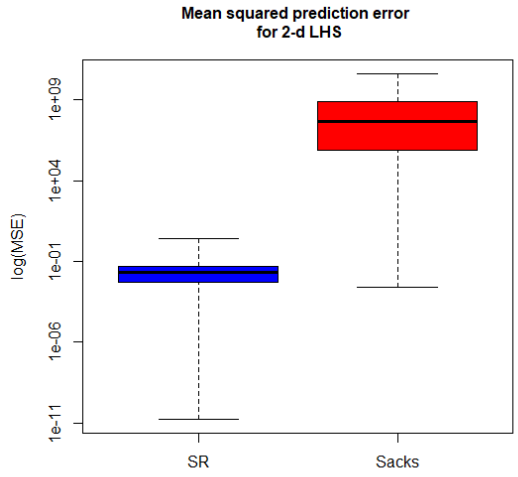


Figure 3.7: Box Plots of MSE 2D LHS

Average Mean Square Error	
SR model	6.548×10^{-2}
Sacks model	1.064×10^{9}

Table 3.5: Average MSE 2D LHS

It is evident from the Figures (3.6, 3.7) and Tables (3.4, 3.5) displayed above that average prediction error of emulator built on $\tilde{\beta}_\lambda$ is significantly less than that of Sacks model emulator

for given choice of design points and design model matrix. Also, the mean square error of Sacks model are skewed to the right indicating an increasing trend in the prediction errors. The results are consistent for both uniform and Latin hypercube samples.

3.6.3 Three dimension Example

In order to explore the performance of proposed model in higher dimension a known $3-d$ simulator is interpolated with the following details. A random Latin hypercube sample of size 45 is generated for $d = 3$ over the design region $[0,\ 1]^3$ which constitute design points for this example. The data generated is divided into training and testing parts in the ratio of $2:1$ with the aid of cross validation folds. The design model matrix is constructed with $n = 30$ design points and $k = 35$ polynomial terms with full basis of degree 4. Followed the examples in $1-d$ and $2-d$, the design points are used to build emulator for the models given in Equations (2.20) and (3.35) whereas the 15 testing data points serve to chose λ such that $\hat{\lambda} = argmin \sum(Y - F\tilde{\beta})$. The $3-d$ function [22] chosen as simulator from Bingham library is a function for comparison of computer experiments and fit well in to the current study. $y(x) = 100(e^{\frac{-2}{x_1^{1.75}}} + e^{\frac{-2}{x_2^{1.5}}} + e^{\frac{-2}{x_3^{1.25}}})$.

Mean square error for predictions at 15 new points is simulated 200 times for design points of size $n = 30$. The box plots for the two compared models are exhibited in Figure (3.8) and mean squared prediction error is given in Table (3.6).

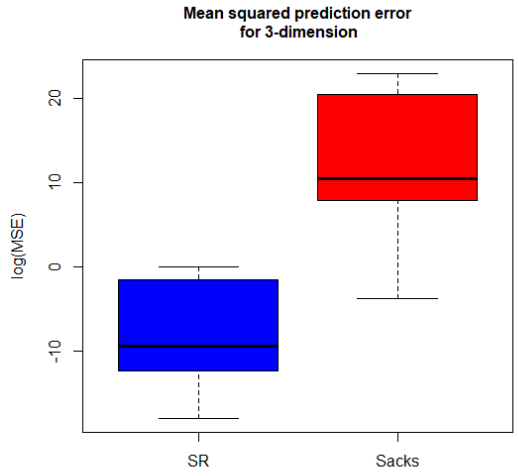


Figure 3.8: Box Plots of MSE for 3-D

Average Mean Square Error	
SR model	9.95×10^{-2}
Sacks model	5.199×10^{8}

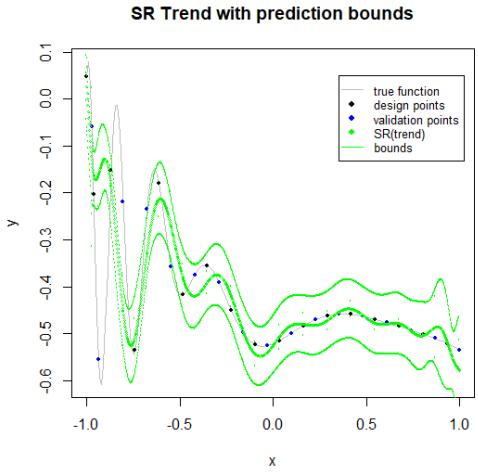
Table 3.6: Average MSE for SR and Sacks model

3.6.4 Non Gaussian function

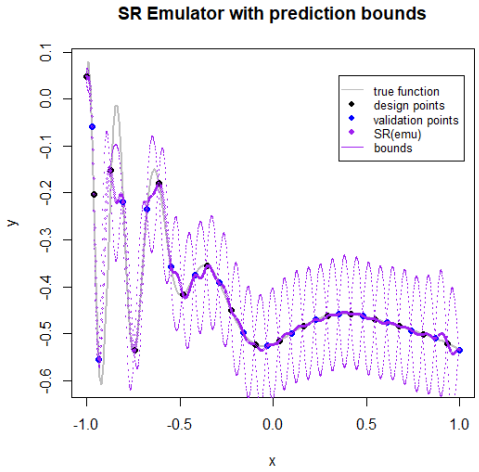
The examples presented in the preceding subsections pertain to the cases where the response $Y(x)$ follows Gaussian process, hence the random variable $Z(x)$ in model (2.1) plays an important part in the emulation resulting in improved predictions at untried inputs by taking into account the correlation matrix R given in Equation (2.6). We are interested to look into the performance of proposed model in such a scenario where the function generating the response $Y(x)$ does not follow Gaussian process. It is expected that predictive model in Equation (3.28) will outperform the estimators developed in Equations (2.20) and (3.35) respectively. The study is carried out as follows. A random sample of size $n = 34$ for $d = 1$ is generated with uniform distribution over the region $[-1,\ 1]$. These $n = 34$ data points serve as design points for the emulator based on Smooth Ridge model with Gaussian emulator given in Equation (3.35) and Sacks model emulator given in Equation (2.20). However, for the computation of $\tilde{\beta}_\lambda$ to estimate trend in Equation (3.28), $n = 34$ sample points are divided into two halves. 17 design points are employed to construct design model matrix F with 18 Legendre polynomial terms while the remaining 17 sample points act as validation data for computing validation error in order to choose λ such that $\hat{\lambda} = argmin \sum (Y - F\tilde{\beta})^2$. The estimated value of λ by minimizing the validation error for this given example is found to be 4.866×10^{-6}. The simulator to act as a response function for the purpose of this study is chosen such that it doesn't satisfy the properties of Gaussian process which is given

$$y(x) = -0.5 * (sin(40 * (x - 0.85)^4) * cos(2.5 * (x - 0.95)) + 0.5 * (x - 0.9) + 1)$$

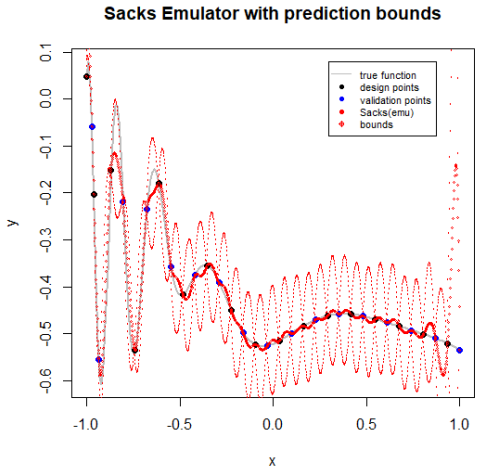
The Figures (3.9) present a detailed graphical account of true function, design points, validation data points, trend based on $\tilde{\beta}_\lambda$, Smooth Ridge emulator and Sacks emulator along with the corresponding bounds based on the mean square error developed in Equations (3.33), (3.41) and (2.8) for 1500 new points.



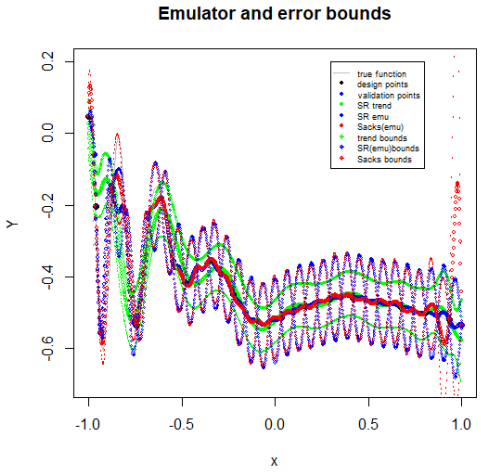
(a) SR Trend with bounds



(b) SR emulator with bounds



(c) Sacks emulator with bounds



(d) emulator and bounds for three models

Figure 3.9: emulator plots for three models

It is apparent from Fig (3.9) that Smooth Ridge emulator and Sacks emulator exhibit wider prediction bounds than those of the Smooth Ridge trend, attributed to larger mean square errors. In order to verify further the validation of results presented in Fig (3.9), we simulated the given example 100 times with random generation of data using uniform distribution. The box plots of 100 mean square errors of predictions at 1500 new points for each of the three models is given in Figure (3.10). The average of mean square errors for the models to be compared are presented in the Table (3.7).

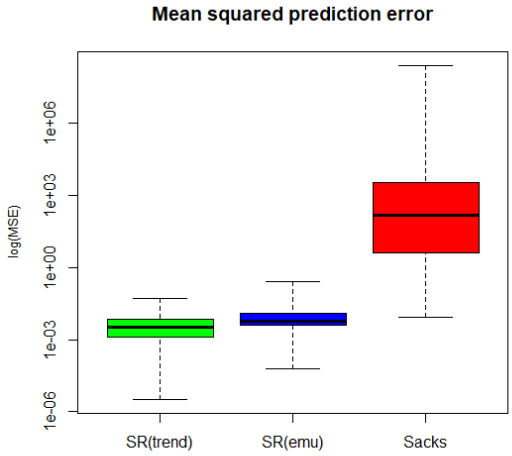


Figure 3.10: Box Plots of MSE for non Gaussian Function

Average Mean Square Error	
SR model trend	6.67×10^{-3}
SR with Gaussian	1.722×10^{-2}
Sacks model	3.76×10^{6}

Table 3.7: Average MSE for three models

It is evident from the box plots and average mean square errors table that the minimum least square error is accounted for the predictive model (3.28) followed by the emulators given in Equation (3.35) and (2.20) respectively. These results allow us to conclude that the proposed model is simplified compared to the use of Kriging model particularly, when the underlying function does not satisfy the conditions of gaussian process in terms of correlation. Moreover, Kriging model is known to be computationally expensive with the increase in the number of observations. The proposed model on the contrary is relatively less expensive where we can avoid computing correlation matrix R.

3.7 COVID-19 transmission model data

This study is about the random generation and analysis of COVID transmission data. This work is motivated by the estimator named COVID19-Aerosol transmission estimator developed by Prof. Jose L Jimenez, Dept. of Chem. and CIRES, Univ. of Colorado-Boulder. It is well known that COVID-19 disease is caused by the SARS-CoV-2 virus, which spreads among people in several ways. According to research studies, World Health Organization (WHO) has summarised the causes of spread of the virus [52].

> The virus can spread from an infected person's mouth or nose in small liquid particles when they cough, sneeze, speak, sing or breathe. These particles range from

> larger respiratory droplets to smaller aerosols. Current evidence suggests that the virus spreads mainly between people who are in close contact with each other, typically within 1 metre (short-range). A person can be infected when aerosols or droplets containing the virus are inhaled or come directly into contact with the eyes, nose, or mouth. The virus can also spread in poorly ventilated and/or crowded indoor settings, where people tend to spend longer periods of time. This is because aerosols remain suspended in the air or travel farther than 1 metre (long-range). People may also become infected by touching surfaces that have been contaminated by the virus when touching their eyes, nose or mouth without cleaning their hands.

For similar reasons, COVID19-Aerosol Transmission Estimator is developed by Prof.Jose L Jimenez to estimate the propagation of COVID-19 by aerosol transmission. The study is comprised of several variables, all kept fixed to find estimate of probability of being infected due to aerosol transmission of COVID virus. The proposed model, estimates the propagation of COVID-19 by aerosol transmission only. The model is based on a standard model of aerosol disease transmission proposed by [62]. The parameters and different factors are included as per recent literature on COVID-19. In the development of estimator, different scenarios are discussed and probability of infection is estimated based on a fixed value of each factor involved, thus the model employed is deterministic model and no randomization is incorporated. For this reason, the propagation of COVID-19 by aerosol transmission is chosen as an application of Smooth Ridge model. The objectives include

- generate data set for model fitting
- Fit models to the given set of data
- comparison of models in terms of predictive capabilities

For the purpose of data generation a fixed scenario(classroom) is considered with the fixed and variable inputs and presented in the following table. The four variable inputs namely vantilation, decayrate, deposition and quanta rate take a grid of values, references of which are borrowed from the COVID19-Aerosol Transmission Estimator and [47]. The tabular description of both type of inputs are enacted below:

Fixed inputs	
Length	7.6 m
Width	6.1 m
Area	47 m^2
Volume	142 m^3
Temperature	20 C
pressure	0.95 atm
Duration	1 h
Repitition	20
Number	10
infected	1
immune	0
susceptible	9

Variable input	
vantilation	0.4-5
Decayrate(h1)	0- 0.63
Deposition(h1)	0.24 -1.5
Qunta rate	4 - 134
BR	0.65-1.38

Table 3.8: Fixed and variable Inputs for `Aerosol Transmission Estimation`

In addition, few important inputs for computing the response include: Death rate, Hospital rate and Probability of infective. Among, variable inputs, breathing rate is chosen to have fixed value 0.8 estimated from *Miller et al.(2020)* [47], for someone occasionally talking. Some important quantities computed from the input given in Tables (3.8) are; First order loss (FOL), Net Emission Rate (NER), Average Quanta Concentration (AQC) and Quanta Inhaled per person (QIP).Explicitly;

$$
\begin{aligned}
FOL &= Vantilation + decayrateofthevirus + depositiontosurfaces \\
NER &= Quantarate \times maskefficiency \times Numberofinfectedpersons \\
AQC &= 1 - exp(-FOL \times durationofevent) \times \left(1 - \frac{1}{FOL \times duration}\right) \times \frac{NER}{FOL \times Volume} \\
QIP &= AQC \times Breathingrate \times duration \times (1 - maskefficiency \times peoplewearingmask)
\end{aligned}
\tag{3.42}
$$

Response variable y is Probability of infection given as $y = 1 - e^{-QIP}$ developed from Wells-Riley infection model [62].

3.7.1 Data set up and simulation

The following set up is employed for simulation study. A random sample of 100 values for each of the parameters: ventilation, Decay-rate, Deposition to surfaces and Quanta rate is generated using Latin hypercube sampling. The vector of response *Probability of infection* is obtained with the help of model defined above. The data set with 100 values of four inputs and respective response vector is then divided into two equal halves as training and validation data. The training data set is further divided into two halves, wherein 25 points serve to construct design model matrix for Smooth Ridge model and to estimate the model parameter $\tilde{\beta}_\lambda$ while, all of 50 training data points are used to construct design model matrix for Sacks model. We use a model M with $|M| = 25$ terms and employ Legendre polynomials $L_\alpha(x) = \prod_{i=1}^{4} L_{\alpha_i}(x_i)$ where $\alpha \in M$. The simulation study is carried out with random generation of 200 Latin hypercube samples of size 100. Four different models namely; SR model, SR model with Gaussian emulator, Sacks model, Ridge model and Ridge model with Gaussian emulator are compared in terms of prediction mean square error. Box plots of mean square prediction errors for four models are given in Figure (3.11)

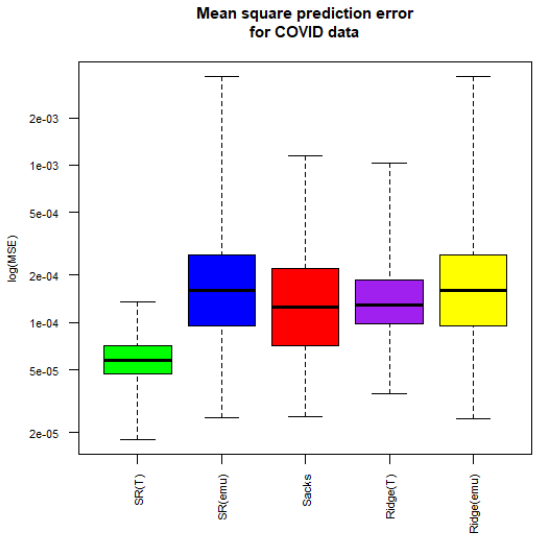


Figure 3.11: Boxplots of mean square prediction error for COVID data

3.7.2 Comments

The data for aerosol transmission of COVID-19 is randomly generated after identifying the input variables. The existing model including emulator used for computer experiments, are fit

to the data which are then compared with the proposed smooth ridge model. The first boxplot labelled *SR(T)* is for mean square error of trend part $F\beta$ of the proposed model without the Gaussian emulator. The second box plot is for `SR` model with Gaussian involved. Same is for the last two plots that are for ridge regression with and without Gaussian random variable. The boxplot coloured "red" presents the results for Sacks model emulator.

- It is apparent from the boxplots in Figure (3.11) that Smooth Ridge model with trend only exhibit least mean square prediction error followed by ridge regression trend.
- Moreover, the comparative modelling approach to COVID data for the estimation of infection probability provides a statistical framework to predict probability of hospitalization, probability of death and eventually the number of new COVID cases arising for a given scenario.

3.8 Comparison with Splines

The splines are functions widely employed in data smoothing and interpolation. A spline is a piecewise polynomial function of a specified degree that usually satisfies boundary conditions.[82]. In what follows a spline $f(x)$ is a univariate function defined over a closed interval $[u, v]$ that we will describe. The spline f is defined piecewise when the interval $[u, v]$ is partitioned into m disjoint sub-intervals [**?**] so that $[u, v] = [t_0, t_1) \cup \ldots [t_{m-2}, t_{m-1}] \cup [t_{m-1}, t_m]$, i.e. $t_0 = u$ and $t_k = v$. A polynomial P_i can be defined for each of the subintervals $P_i : [t_i, t_{i+1}]$. The points $t_0, t_1 \ldots t_m$ are called knots and the vector comprising knots $(t_0, \ldots, t_m)$ is called knot vector. The spline is said to be of a given degree v if the polynomials $P_1, P_2 \ldots$ have all the same degree. We briefly discuss and compare two types of spline functions called cubic splines and penalized splines.
The function f can be represented as basis splines for fixed knots m and degree v

$$f(x) = \sum_{i=1}^{m+v+1} \beta_i B_i(x)$$

where, β_i are spline coefficients and B_i are the basis functions. The basis function adds smoothness to the spline fitting. There are many variants of basis splines [20], for example, splines characterized by the choice of basis function, the choice of extended knots and the special conditions imposed on splines. One of the most commonly used splines is cubic splines which is a piecewise cubic polynomials having continuous first and second derivative [13].

3.8.1 Comparison of SR,SSM and Spline with R example

In this section we compare two of smooth methodologies namely Smooth Ridge and Smooth Supersaturated model with Splines. In both cases we want to evaluate the convergence of two methods namely splines and SR as a function of λ and design model term. From the theoretical view point, the convergence of minimal roughness of SSM to that of splines has been studied in [4] and we contribute with a numerical simulation.

We consider a univariate setting in which a sample of size n is randomly selected from equally spaced data points over the domain $[-1, 1]$. For the construction of design model matrix F, we consider six different size of model basis (see Table 3.9). The true function for the response vector chosen is $y(x) = (6x - 2)^2 sin(12x - 4)$ [71]. For the Smooth Ridge model eight different values of λ are selected; $\lambda = \{0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 5, 10\}$, to investigate the behaviour of SR model relative to Splines. For each combination of number of model terms k and λ, we computed the absolute difference of predictions between SSM-Splines and SR-splines. We then plot the average difference against for all the values of λ that we considered. For univariate case, SSM has limitation oevr the choice of design terms therefore for $n = \{40, 50\}$, we only plot the difference of Smooth Ridge and Splines.

relationship between design points and basis	
n	model basis **k**
10	{15,21,27,33,39}
25	{35,41,47,53,59}
40	{45,49, 53, 57, 61}
50	{55, 57, 59, 61, 63}

Table 3.9: Choice of design points and model basis

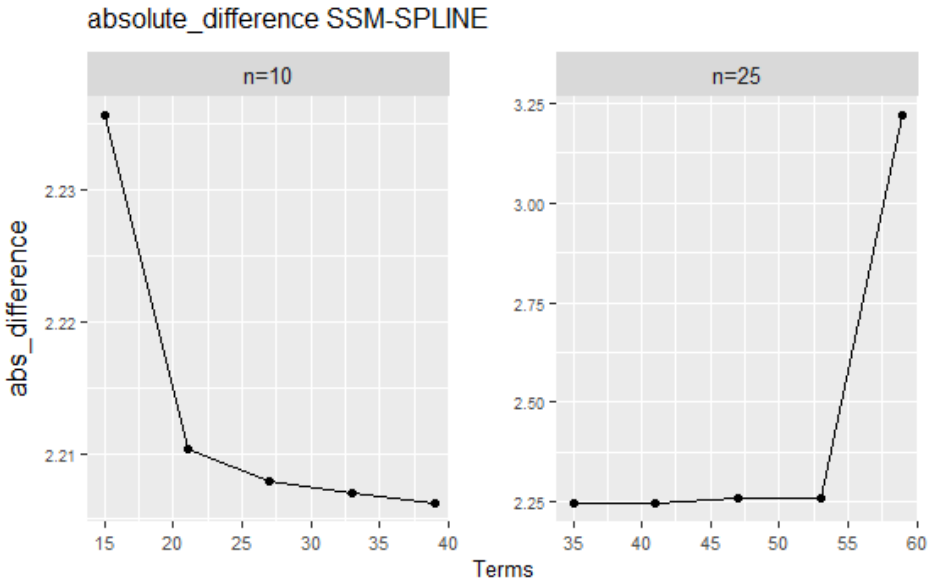


Figure 3.12: Difference of prediction mean (Smooth-Supersaturated model,Cubic Spline)

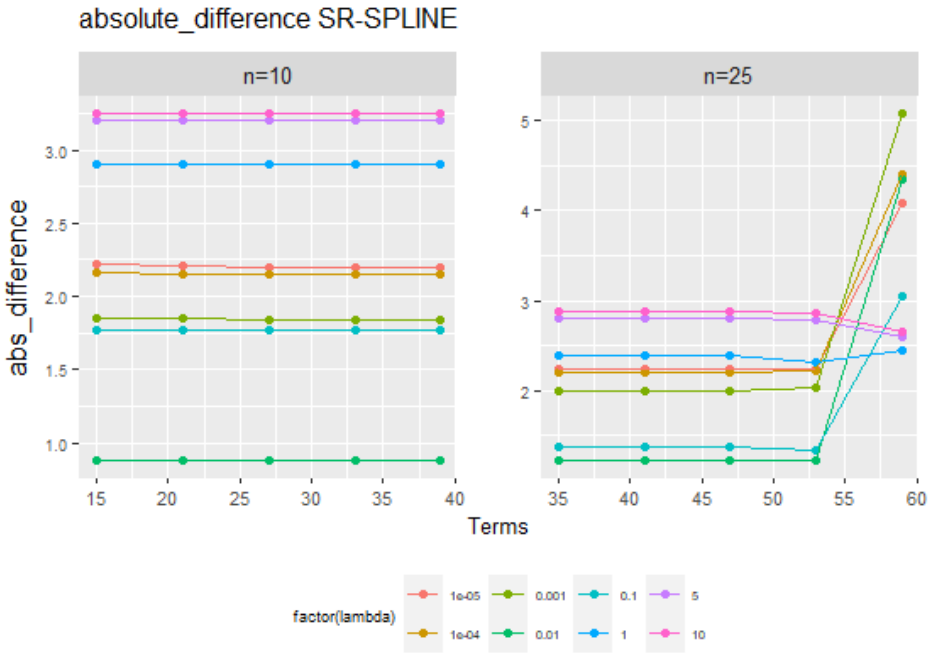


Figure 3.13: Difference of prediction mean (Smooth-Ridge model,Cubic Spline)

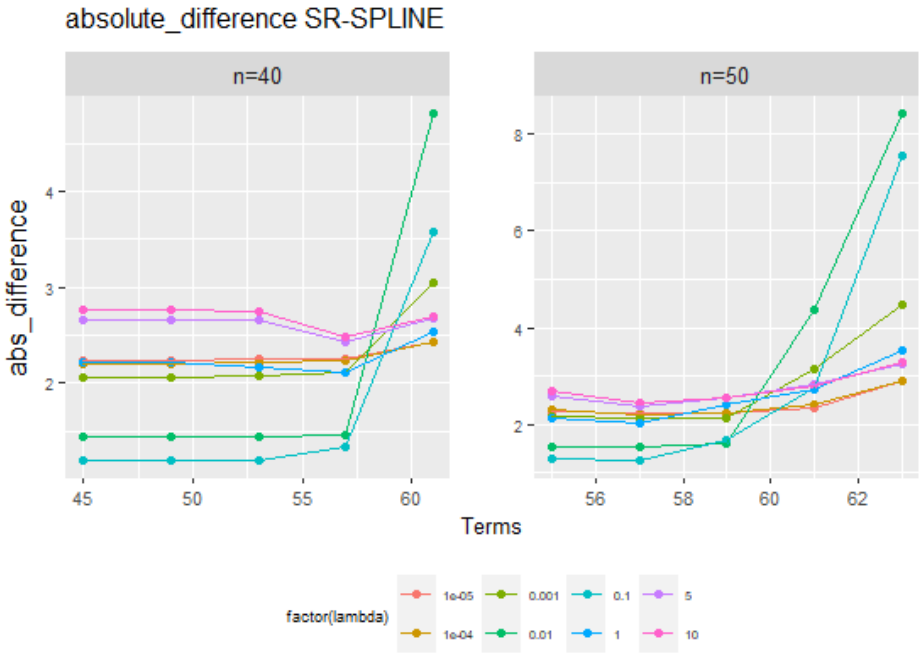


Figure 3.14: Difference of prediction mean (Smooth-Ridge model,Cubic Spline, $n = \{40, 50\}$)

3.8.2 Comments

From Figure (3.12), we can identify the change in difference between SSM and cubic splines as the model terms increase. It is to emphasise here that λ does not play any role in this comparison. It is apparent from the figure that for smalll sample size $n = 10$, there is a sharp decrease in the difference when model terms increases from $k = 15$ to $k = 20$. There is a gradual decrease of the difference with closer agreement between the methodolgies for $n > 20$. For larger sample size $n = 25$, the difference between SSM and Splines remain steady for model terms upto $k = 55$ with a drastic increase for $n > 55$. Figure (3.13) shows the difference between Smooth Ridge and Spline for $n = 10$ and $n = 25$. It is apparent from the difference plots that they are independent of the choice of model terms and depend. This can be seen from the fact that the for each value of λ the difference is constant across different choice of model terms. We identify a value of lambda $\lambda = 0.01$ such that the absolute difefrence of predictions between SSR and Splines is minimum. For a lrager sample size $n = 25$, we witness a similar behaviour of the differences as for $n = 10$ when $k < 55$. A sharp increase is evident for $k > 55$ and $\lambda < 1$. On the contrary it is interesting to observe that for $\lambda = \{5, 10\}$, the difference steadily decreases for $k > 55$. For $n = \{40, 50\}$, we can observe the similar behaviour as that of $n = 25$ with the most steep slope when $\lambda = \{0.01, 0.1\}$ and $k > 55, k > 58$ respectively. In

summary, for each of the chosen sample size n, the difference of predictions between Smooth Ridge and Splines remain steady upto certain choice of model term k but changes with the change in λ. the difference takes a sudden jump for certain λ and choice of model terms.

3.9 comparison of Smooth Splines with Smooth Ridge

Choosing the right number of knots in splines has been an active area of research in splines literature to avoid under or over fitting. Much of the research in this domain is motivated by the pioneer work of smoothing splines by [61] where a penalty is introduced to control the amount of smoothing. In smoothing splines the function $f(x)$ can be written in vector form as as $f(x) = B(x)\beta^p$, and the coefficient vector is selected by minimizing

$$\min_{\beta^p} ||y - B\beta^p||^2 + \lambda_p(\beta^p)^T H \beta^p \tag{3.43}$$

where $B(x)$ is a vector of spline basis, β^p is the parameter vector and H is the penalized matrix built with the derivatives of the splines bases. The Equation (3.43) is prone to $n \times n$ matrix inversion problem since the spline basis B grows with the order of the size of the sample [28]. To this end a reformulation of basis function is made, where $B(x)$ is high dimensional fixed basis function in contrast to smoothing splines referred as Penalized spline [64]. In this section we compare the Smooth Ridge model to that of penalized splines owing to the similarity of the penalty employed in both methodologies. It is relevant to mention here that the penalty λ_p and λ for smooth splines and Smooth Ridge model respectively are not similar. λ_p is the smoothing parameter that serves as trade-off between the data fitting and having linear model with the splines basis as basis function. λ represents the penalty parameter which is trade-off between the linear model and the curvature with supersaturated basis as basis function. With the help of simple univariate example, we are interested to investigate if for given estimated λ_p we can find a value of λ such that the prediction behaviour of the two methodologies become closer.

A univariate random sample of size n is selected with equally spaced data points over the design region $[0,\ 1]$ which constitute design points for this example. The design model matrix F is constructed with n design points and different choice of model terms k given in Table (3.10). The true function for the response is $y(x) = (6x-2)^2 sin(12x-4)$ [71]. Different values of λ for Smooth Ridge model are $\lambda = \{10^{-12}, 10^{-10}, 10^{-8}, 10^{-6}, 10^{-4}, 0.02, 0.04, 0.06, 0.08, 0.1\}$ when $n = \{10, 25\}$ and $\lambda = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1\}$ for $n = \{40, 50\}$. For each combination of design model term k and λ absolute difference of predictions is

computed pairwise between SR and penalized splines at equally spaced 1000 new data points. The average absolute difference against each of the model terms is plotted for each of the values of λ. It is relevant to mention here that the choice of penalty parameter λ_p for penalized splines is fixed at the estimated value 8.7×10^{-18}. For model fitting with penalized splines, smooth splines function `ss` is employed in `R` and predictions at new points are obtained from `predict` function.

relationship between design points and basis	
n	model basis **k**
10	{12, 17, 22, 27, 32, 37}
25	{27, 32, 37, 42, 47, 52}
40	{42 ,46, 50, 54, 58, 62}
50	{52,54,56,58,60,62}

Table 3.10: Choice of design points and model basis

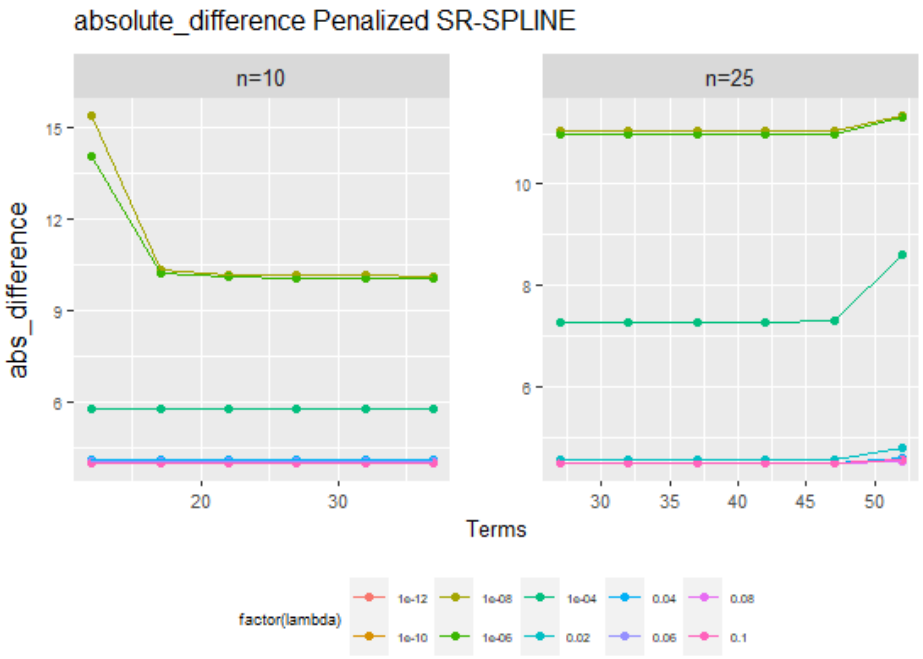


Figure 3.15: Difference of prediction mean (Smooth-Ridge model,Smooth-Spline $n = \{10, 25\}$)

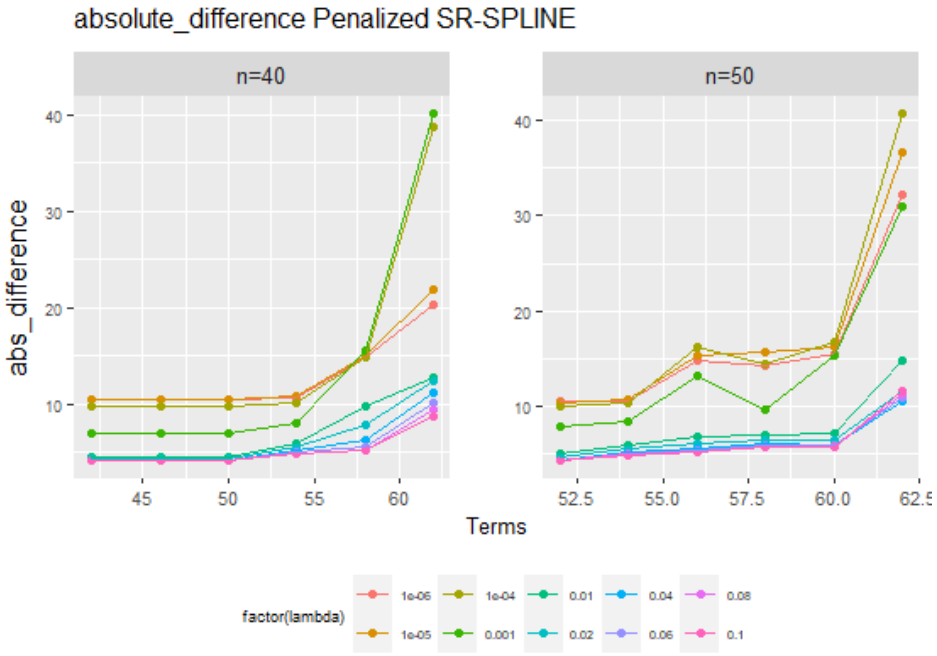


Figure 3.16: Difference of prediction mean (Smooth-Ridge model,Smooth-Spline)

3.9.1 Comments

We can readily infer from Figure (3.15) that the average absolute prediction difference for $n = 10$ is constant for any choice of model terms across all values of λ except $\lambda = \{10^{-8}, 10^{-6}\}$ where the difference is constant for $k > 17$. The minimum difference is observed at $\lambda = 0.1$. The similar constant pattern is noticeable for $n = 25$ across all model terms for every value of λ except $\lambda = 10^{-4}$ where we can see a sudden increase in difference for $k > 47$ and $\lambda = \{10^{-8}, 10^{-6}\}$ where a gradual upward trend appears for $k > 47$.

The absolute average difference for $n = 40$ and $n = 50$ is plotted in Figure (3.16). For $n = 40$ we can observe that the difference (absolute average) between Smooth ridge and Splines remain constant for $k = 42$ to $k = 54$ model terms and all values of λ with the minimum difference observed at $\lambda = 0.1$. The difference starts drifting upwards for $k \geq 55$ with the highest spike at $\lambda = 10^{-4}$ and 10^{-3}. For $n = 50$ there is constant trend in difference for $\lambda > 0.001$ with steady increase at $60 \leq k \leq 65$. For $\lambda < 0.001$ the curve keeps wiggling until a sharp jump at $k = 60$. The minimum difference is achieved at $\lambda = 0.1$.

In summary, for a given choice of supersaturated and spline basis, a vlaue of λ can be found such that the prediction behaviour of the two approaches converge. This domain has a potential to be explored as a future research.

3.10 Sensitivity Analysis

Sensitivity Analysis (SA) provides a significant tool to address the exploration of computer experiments, the fundamental goal of which is to explore computer code. Sensitivity Analysis investigates how uncertainty in a model's output can be attributed to various sources of uncertainty in the model input [67]. In other words, Sensitivity Analysis is used to identify the input variables that contribute the most to an output behaviour, the less or non-influential inputs, or to ascertain some interaction effects within the model. In general, output values from complex computer models can depend on a high number of input parameters and physical variables. In practical situations, not all of the input variables may always be known with high precision. Some of these input parameters and variables may be unknown, unspecified, or defined with a large imprecision range. For example, variables that describe field conditions, and variables that cannot be directly measured and include unknown or partially known model parameters. Moreover, the relationship between output and input is not always fully understood. Therefore, investigating computer code experiments remains a significant challenge in this context. A good model is characterized by the confidence evaluation which necessitates, first, quantifying the uncertainty in any model results (uncertainty analysis) and second, evaluation of each input's contribution to the output uncertainty (sensitivity analysis). The Sensitivity analysis process entails the computation and analysis of the importance indices of the input variables with respect to a given quantity of interest in the model output. Importance measures of each uncertain input variable on the response variability provide a deeper understanding of the modelling in order to reduce the response uncertainties in the most effective way. For instance, putting more efforts on knowledge of influential inputs will reduce their uncertainties. The underlying goals for SA are model calibration, model validation and assisting with the decision making process.

The initial approach to Sensitivity Analysis is known as the local approach which allows to study the impact of small input perturbations on the model output. This is the deterministic approach that consists of calculating the partial derivatives of the model at a specific point of the input variable space [66], [10], [2]. Local approaches find their applications to tackling uncertainty analysis and Sensitivity analysis in problems related to environmental systems such as climatology, oceanography and hydrology some of which can be found in [11], [58] , [49] and [84]. In contrast to local SA, which emphasises the change in model output with the small variations around a specific value of model inputs, Global sensitivity analysis takes into account the entire domain of input parameter variations [81]. A detailed account of global sensitivity is given in [68].

One of the variant of global sensitivity analysis is variance-based sensitivity analysis is known as Sobol' method or Sobol' indices [73]. In this framework the variation in model output is decomposed into fractions which are attributed to the model inputs. The variation is expressed in terms of percentages which are direct measure of uncertainty. Sobol' indices are appealing because:(1) this is global method that explores entire input domain for sensitivity measures,(2) they perform well for non-linear response and no assumption of linearity is needed and (3) can account for interaction effects in non-additive systems. In this section, the method to compute Sobol' indices is detailed which will then be employed within the framework of Legendre polynomials for the construction of Smooth Ridge model, in Chapter (4).

We now review the basic material following [73]. For the computation of sensitivity indices, any model may be viewed as a blackbox function defined as $\mathbf{y} = f(\mathbf{x})$, where $\mathbf{x}$ is a vector of d model inputs $(x_1, x_2, \dots x_d)$, and $\mathbf{y}$ is a univariate model output. Furthermore, the inputs are assumed to be independently and uniformly distributed within the unit hypercube, $\mathbf{x}_i \in [0,\ 1]$ for $i = 1, 2, \dots, d$. Since, any input space can be transformed onto the unit hypercube, therefore no loss of generality is incurred. The function f defined as $f : [0,\ 1]^d \to \mathbb{R}^d$ can be decomposed as,

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i<j}^{d} f_{ij}(x_i, x_j) + \ldots + f_{1,2,\ldots,d}(x_1, x_2, \ldots, x_d) \tag{3.44}$$

where f_0 is a constant, $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ are functions of x_i and x_i, x_j respectively, same goes for the higher order interactions. The quantity f_0 is the average of $f(x)$ over the unit cube i.e.

$$\begin{aligned} f_0 &= \int_0^1 f(\mathbf{x}) d\mathbf{x} \\ &= \int_0^1 \ldots \int_0^1 f(x_1, x_2, \ldots, x_d) dx_1, \ldots, dx_d \end{aligned} \tag{3.45}$$

The ith main effect is defined as

$$f_i(x_i) = \int_0^1 \ldots \int_0^1 f(x_1, x_2, \ldots, x_d) dx_1, \ldots, dx_{i-1} dx_{i+1}, \ldots, dx_d - f_0 \tag{3.46}$$

The two way interaction effect is computed as

$$f_{i,j}(x_i, x_j) = \int_0^1 \ldots \int_0^1 f(\mathbf{x}) \prod_{k \neq i,j} dx_k - f(x_i) - f(x_j) - f_0 \tag{3.47}$$

The three way interaction effect is computed in a similar fashion

$$f_{i,j,k}(x_i, x_j, x_k) = \int_0^1 \dots \int_0^1 f(\mathbf{x}) \prod_{l \neq i,j,k} dx_l - f_i(x_i) - f_j(x_j) - f_k(x_k) - f_{i,j}(x_i, x_j) \\ - f_{j,k}(x_j, x_k) - f_{i,k}(x_i, x_k) - f_0$$

From the above equations, it can be seen that $f_i(x_i)$ is the effect of x_i only, while integrating out the rest of the effects termed as main effect of x_i. Similarly, the interaction $f_{i,j}(x_i, x_j)$ is the effect of varying both x_i and x_j simultaneously. The higher order interaction effects follow the same analogy. It is vital to mention here that the definition of main and interaction effects are driven by the condition of orthogonality [74] which states that.

$$\int_0^1 f_{i_1,i_2,\dots i_s(x_{i_1},x_{i_2}}, \dots, x_{i_s}) dx_k = 0 \text{ for } k = i_1, \dots, i_s \tag{3.48}$$

In order to decompose the total variation into fractions, the variance of $\mathbf{y}$ is defined as:

$$D = \int_0^1 f(\mathbf{x})^2 d\mathbf{x} - f_0^2 \tag{3.49}$$

which can be decomposed into its constituent fractions followed by the function decomposition above such that,

$$D = \sum_{i=1}^{d} D_i + \sum_{i<j}^{d} D_{ij} + \dots + D_{1,2,\dots,d} \tag{3.50}$$

$$\text{where,} \quad D_i = \int_0^1 f_i(x_i)^2 dx_i \quad \text{and} \quad D_{ij} = \iint_0^1 f_{i,j}(x_i, x_j)^2 dx_i dx_j. \tag{3.51}$$

The rest of the variance decomposition follows the same formulation.

3.11 Sensitivity Indices

The sensitivity indices indicate the percentage contribution of the effects (indicated by the decomposition of variance or percentage of variation proportional to total variation. Therefore, the first order sensitivity index or main effect index is the direct variance based measure of sensitivity denoted by S_i. In the same manner, S_{ij} are second order interaction indices and so

on. The sensitivity indices are computed as

$$S_i = \frac{D_i}{D} \quad \text{and} \quad S_{ij} = \frac{D_{ij}}{D} \tag{3.52}$$

The higher order interaction indices can be computed analogously by dividing the relevant terms in variance decomposition by the total variance.

3.11.1 Example with Legendre Polynomials

The examples used for the comparison of models in preceding sections of Chapter(3) were constructed on Legendre polynomials which provided the formulation of $f(\mathbf{x})$. In this example, sensitivity indices are computed with the use of Legendre polynomials denoted by $P_n(x)$ and defined over the unit cube $[0,\ 1]$. It is pertinent to mention that Legendre polynomials employed in the preceding sections had the domain over $[-1,\ 1]$ denoted by $L_\alpha(x)$, therefore, in order to perform sensitivity analysis for the similar examples, Legendre polynomials will be transformed to $[0,\ 1]$ as follows

$$P_n(x) = L_n(2x-1)$$

Let $\mathbf{x} = (x_1, x_2)$ be the input vector with two input variables. In order to construct Legendre polynomials, the set of basis indices chosen is $M = \{(0,0),(1,0),(0,1),(1,1)\}$. The general form of the model $y(\mathbf{x}) = f(\mathbf{x})$ can be expressed as

$$f(\mathbf{x}) = \theta_{00}P_0(x_1)P_0(x_2) + \theta_{10}P_1(x_1)P_0(x_2) + \theta_{01}P_0(x_1)P_1(x_2) + \theta_{11}P_1(x_1)P_1(x_2) \tag{3.53}$$

The function $f(\mathbf{x})$ is decomposed similar to what is given in Equation (3.44). Therefore,the constant f_0, main effects $f(x_i)$ and interaction $f(x_i, x_j)$ are computed using the expressions given in Equations (3.45, 3.46 and 3.47).

$$f_0 = \theta_{00}$$

$$f_1(x_1) = \int_0^1 f(\mathbf{x})dx_2 - f_0 = \theta_{00} + \theta_{10}L_1(2x_1-1) + 0 + 0 - \theta_{00}$$

$$f_1(x_1) = \theta_{10}L_1(2x_1-1) \tag{3.54}$$

$$\text{Similarly,} \quad f_2(x_2) = \theta_{01}L_1(2x_2-1) \tag{3.55}$$

$$\text{and,} \quad f_{12}(x_1,x_2) = \theta_{11}L_1(2x_1-1)L_1(2x_2-1) \tag{3.56}$$

The variances for this example can be decomposed similar to what is explained in Equation (3.50). Therefore, the contribution of marginal variations and variation due to interaction to the total variations is computed in the equations below.

$$D_1 = \int_0^1 f_1(x_1)^2 = \int_0^1 \left(\theta_{10} L_1(2x_1 - 1)\right)^2 dx_1$$

applying substitution and simplifying, the variance of the main effect x_1 and x_2 is given as,

$$D_1 = \frac{\theta_{10}^2}{3} \tag{3.57}$$

$$D_2 = \frac{\theta_{01}^2}{3} \tag{3.58}$$

The interaction terms variation can be computed as

$$\begin{aligned} D_{12} &= \int_0^1 \int_0^1 f_{12}(x_1, x_2)^2 dx_1 dx_2 \\ &= \theta_{11}^2 \int_0^1 L_1^2(2x_1 - 1) dx_1 \int_0^1 L_1^2(2x_2 - 1) dx_2 \\ D_{12} &= \frac{\theta_{11}^2}{9} \end{aligned} \tag{3.59}$$

From Equations (3.57),(3.58) and (3.59), the total variance is straightforward to obtain

$$D = D_1 + D_2 + D_3 \tag{3.60}$$

$$D = \frac{3\theta_{10}^2 + 3\theta_{01}^2 + \theta_{11}^2}{9} \tag{3.61}$$

The sensitivity indices are now easy to compute with the aid of expressions given in Equation (3.52). The computations above provide a mechanism to formulate the general form of the variances to avoid solving integrals for a new problem or choice of basis function. For some set of distinct index vector α, let $\alpha_k^{(i)}$ represent the (k^{th}) index for (i^{th}) input x_i such that $\alpha_k \in \alpha$. Then, the variances associated with main effects and interactions can be expressed in general notation as

$$D_i = \sum_{k \neq 0} \frac{\theta^2_{\alpha_k^{(i)}}}{2\alpha_k^{(i)} + 1} \quad \text{and} \quad D_{ij} = \sum_{k \neq 0} \frac{\theta^2_{(\alpha_k^{(i)} \alpha_k^{(j)})}}{(2\alpha_k^{(i)} + 1)(2\alpha_k^{(j)} + 1)} \tag{3.62}$$

3.12 A gentle introduction of Bayesian formulation for Smooth Ridge model

An introduction of Bayesian methodology is described in Section (2.3). In this section we provide a gentle introduction of Bayesian regression in the context of Smooth Ridge model. A sharp distinction between the frequentist and Bayesian is that the former treats the parameters as platonic quantities which assume some fixed value and needs to be estimated from the data. On the other hand the Bayesian treats the parameters as random quantities and formalizes one's believes about the parameters in the form of probability distribution. The schism between the frequentist and Bayesian centres on the concept of probability [79]. A detailed account of the Bayesian formulation for Ridge regression is accounted in [79] which provides a direction to develop Bayesian version of Smooth Ridge model.

Recall that the Smooth ridge model has two parameters β and σ^2. The commonly chosen priors of β, and σ^2 within the context of ridge regression are $\beta|\sigma^2 \sim \mathcal{N}(0, \sigma^2\lambda^{-1}I)$ and $\sigma^2 \sim \mathcal{IG}(\alpha_0, \gamma_0)$ [79]. We formulate the priors for Smooth Ridge model similar to that of Ridge regression. We assume that the smoothness matrix K is invertible, for example for a non-hierarchical model. Then the priors for β, and σ^2 are $\beta|\sigma^2 \sim \mathcal{N}(0, \sigma^2\lambda^{-1}K^{-1})$ and $\sigma^2 \sim \mathcal{IG}(\alpha_0\, \gamma_0)$. The joint posterior distribution of β, and σ^2 for the given choice of priors and likelihood of the Smooth ridge model is given as:

$$\begin{aligned}\Pi_{\beta,\sigma^2}(\beta, \sigma^2|y, x) &\propto f_y(y|x, \beta, \sigma^2) \times f_\beta(\beta|\sigma^2) \times f_{\sigma^2}(\sigma^2) \\ &\propto \sigma^{-n} exp\left[-\frac{1}{2}\sigma^{-2}(Y - F\beta)^T(Y - F\beta)\right] \times \sigma^{-k} exp\left[-\frac{1}{2}\sigma^{-2}\lambda\beta^T K\beta\right] \\ &\quad \times (\sigma^2)^{-\alpha_0-1} exp\left[-\frac{1}{2}\sigma^{-2}\gamma_0\right] \qquad (3.63)\end{aligned}$$

The terms inside the exponent containing β can be solved and simplify to the following:

$$(Y - F\beta)^T(Y - F\beta) + \lambda\beta^T K\beta = Y^TY - \beta^T(F^TF + \lambda K)\tilde{\beta}_\lambda - \tilde{\beta}_\lambda^T(F^TF + \lambda K)\beta + \beta^T(F^TF + \lambda K)\beta$$

adding and subtracting $\tilde{\beta}_\lambda^T(F^TF + \lambda K)\tilde{\beta}_\lambda$ we get,

$$\begin{aligned}(Y - F\beta)^T(Y - F\beta) + \lambda\beta^T K\beta &= Y^TY - Y^TF(F^TF + \lambda K)^{-1}F^TY \\ &\quad + (\beta - \tilde{\beta}_\lambda)^T(F^TF + \lambda K)(\beta - \tilde{\beta}_\lambda) \qquad (3.64)\end{aligned}$$

substituting the result from Equation (3.64) in Equation (3.63) we can write the posterior

distribution as

$$\begin{aligned}\Pi_{\beta,\sigma^2}(\beta,\sigma^2|y,x) &\propto \Pi_\beta(\beta|\sigma^2,y,x)\times\Pi(\sigma^2|y,x)\\ &\propto \sigma^{-k}exp\left[-\frac{1}{2}\sigma^{-2}(\beta-\tilde{\beta}_\lambda)^T(F^TF+\lambda K)(\beta-\tilde{\beta}_\lambda)\right]\\ &\quad\times\sigma^{-n}exp\left[-\frac{1}{2}\sigma^{-2}(Y^TY-Y^TF(F^TF+\lambda K)^{-1}X^TY)\right]\\ &\quad\times(\sigma^2)^{-\alpha_0-1}exp\left[-\frac{1}{2}\sigma^{-2}\gamma_0\right]\end{aligned}\tag{3.65}$$

It is straightforward to see that the posterior distribution of β is

$$\Pi_\beta(\beta|\sigma^2,y,x)\propto exp\left[-\frac{1}{2}\sigma^{-2}(\beta-\tilde{\beta}_\lambda)^T(F^TF+\lambda K)(\beta-\tilde{\beta}_\lambda)\right]\tag{3.66}$$

The closed form of the posterior distribution of σ^2 can be obtained with little effort which is inverse gamma distribution with parameters α_0' and γ_0' where, $\alpha_0'=\alpha_0+1$ and $\gamma_0'=\gamma_0+\frac{1}{2}\left[Y^TY-Y^TF(F^TF+\lambda K)^{-1}X^TY\right]$ respectively.

3.12.1 Empirical Bayes estimate of λ

Empirical Bayes is branch of Bayesian Statistics in which the prior distribution is not fully specified and hyper-parameters are found empirically from data in hand [79]. We illustrate the method of empirical Bayes to find the estimate of hyper-parameter λ which suggest to obtain marginal posterior by integrating out the model parameters from the distribution. Our interest

here lies in the estimation of λ which can be obtained as:

$$\begin{aligned}
\hat{\lambda}_{em} &= \arg\max_{\lambda} \int_0^{\infty} \int_{\mathbb{R}} \Pi_{\beta,\sigma^2}(\beta, \sigma^2|y, x) d\beta d\sigma^2 \\
&= \int_0^{\infty} \int_{\mathbb{R}} \sigma^{-k} exp\left[-\frac{1}{2}\sigma^{-2}(\beta - \tilde{\beta}_{\lambda})^T (X^T X + \lambda K)(\beta - \tilde{\beta}_{\lambda})\right] \\
&\quad \times \sigma^{-n} exp\left[-\frac{1}{2}\sigma^{-2}(Y^T Y - Y^T X (X^T X + \lambda K)^{-1} X^T Y)\right] \\
&\quad \times (\sigma^2)^{-\alpha_0 - 1} exp\left[-\frac{1}{2}\sigma^{-2}\gamma_0\right] d\beta d\sigma^2 \\
\hat{\lambda}_{em} &= \arg\max_{\lambda} \int_0^{\infty} \sigma^{-n} exp\left[-\frac{1}{2}\sigma^{-2}(Y^T Y - Y^T X (X^T X + \lambda K)^{-1} X^T Y)\right] \\
&\quad \times |F^T F + \lambda K|^{-\frac{1}{2}} (\sigma^2)^{-\alpha_0 - 1} exp\left[-\frac{1}{2}\sigma^{-2}\gamma_0\right] d\sigma^2 \\
\hat{\lambda}_{em} &= \arg\max_{\lambda} |F^T F + \lambda K|^{-\frac{1}{2}} (\gamma_0')^{-(\alpha_0 + n/2)} \qquad (3.67)
\end{aligned}$$

where, $\gamma_0' = \gamma_0 + \frac{1}{2}[Y^T Y - Y^T F (X^T F + \lambda K)^{-1} F^T Y]$. For the empirical estimate of λ we can find empirical estimate of β.

3.13 Conclusion

In this chapter, Smooth Ridge model is proposed which is a fine blend of smooth supersaturated model and ridge model where former provides a method of smooth interpolation and later takes care of the singularity problem in design model matrix. The prime purpose of the model is to provide improved predictions for out-of-sample unknown points along with quantification of prediction errors. Theoretical results are provided to derive the condition for which mean square error of SR emulator is less than that of mean square error of Sacks model. The results are supported with the help of examples using simulated data where SR model is compared with Sacks model for Design and Analysis of Computer Experiments (DACE). The comparison study is carried out for $1-d$, $2-d$ and $3-d$ data. It is apparent from the respective plots and tables that proposed model outperforms sacks model in all the examples studied. The performance of proposed model is also evaluated for the case when kriging is not supported by the observed data in which case the SR model with trend only provides efficient predictions at unknown points with least mean square error compared to SR model with kriging and Sacks model for DACE. Moreover, the real life example is studied for aerosol transmission of COVID-19 and Smooth Ridge model is fit to the data generated and comparative study is carried out

with ridge model and DACE models.The Smooth Ridge model outperformed well known and widely employed DACE emulator. These results highlight two features of the model:

1. Simple model
2. Computationally less expensive

The proposed SR model provides a new direction in the realm of smooth interpolation for computer experiments. It is shown that Ridge regression and standard regression models are special cases of SR model which makes it more flexible. Moreover, the model bears inherent features of smoothness along with efficient interpolation irrespective of the singularity problem of design matrix. In addition, the development of Bayesian methodology for Smooth Ridge model is introduced that provides a framework for future research direction.

Chapter 4

Comparisons

The methodology developed in chapter (3) needs to be evaluated scrupulously before making final conclusions. For this reason, detailed comparisons are carried out, in order to assess the performance of proposed Smooth Ridge (SR) model against the contemporary models in literature, namely: Ridge regression, Smooth Supersaturated (SSM) model and DACE model. A brief account of each of these models is delineated in the preceding chapters (2,3). It is pertinent to mention here that development of Smooth Ridge model is built primarily on two key factors: supersaturated basis and K matrix which is constructed by defining a measure of roughness. For low dimension data, it is easy to increase the number of basis that give rise to a model with higher order polynomials resulting into smoother models. However, as the dimensions of input data increase, the number of basis to be included in the model becomes cumbersome. One of the reasons is the increase in the sparsity of K matrix corresponding to linear model terms, the other being computational expense. Therefore, it is pivotal to explore the proposed model flexibility and performance evaluation for different number of dimensions and basis. The rest of the chapter is composed as follows. Section (4.1) introduces the overview of simulations and methodology underlying the comparisons. The methodology to build Smooth Ridge model for fix number of design points n against different choices of model terms k is described in Section (4.2). For fixed number of model terms k and different choices of design points n, the simulation methodology is explained in Section (4.3). The comparisons for fixed design points and fixed model terms are repeated for a simulator which is not supported by polynomials and are accounted in Section (4.4) and (4.5). COVID Data study as an application of Smooth Ridge (SR) model elucidated in chapter (3) is re-accounted for different choices of model terms k in Section (4.6). The comparison study with temperature data is furnished in Section (4.7). The chapter is concluded with the introduction and methodology of Sensitivity

analysis for Smooth Ridge model in Section (4.8).

4.1 An overview of the simulations

In this section a comprehensive detail is provided in order to build the computer simulations for the comparisons of Smooth Ridge model with other models under study against each of the number of input dimensions d chosen. The building blocks for these comparisons encompass the following:

1. dimensions of input data (d)
2. simulator for computer simulation y
3. number of model terms (basis) k
4. number of design points n
5. number of computer runs

Five different dimensions, $d = \{2, 5, 10, 15, 20\}$, are chosen for this study. For each of the dimension, simulations are run for two scenarios:

1. fix number of design points n
2. fix number of model terms k

For the fixed number of design points different choices of basis are explored to decide upon the model terms, subsequently evaluating the model performance. Similarly, for the fixed number of model terms k, various choices of design points n are considered in order to adopt an holistic approach for reaching the conclusions. The data comprising both the design points (training data) and validation (test data) are first generated with Latin hypercube sampling. The data obtained is then converted to $[-1,\ 1]^d$ for the purpose of employing Legendre polynomials to construct the model terms giving rise to the design model matrix F. The data is divided in the ratio of $2:1$ to serve as training and testing data. The cross validation method is implemented to create data folds whereby two CV folds are randomly chosen to set out the design points n and the remaining CV fold to act as test data. The method is repeated iteratively over all possible choices of CV folds in such a way that each of three CV folds act as training data. Different dimensions of input data for the comparsions are chosen therefore the true function or the simulator $y(x)$ responsible for generating the response is to be the one that can be

employed for any number of dimensions. For this purpose two of such functions are chosen to run the computer simulations and making conclusions based on the comparison results namely Bratley et al.(1992) and Levy function. Further details on these functions are enacted in the forthcoming sections. Following the choice of number of dimensions, simulator and scenarios for models' comparisons, the predictors for the Smooth Ridge, Ridge, Smooth Supersaturated and Design and Analysis of Computer Experiments (DACE) models are devised and the predictions for the test data are obtained. The predictions are compared in the following manner:

1. with trend part of the model $F\beta^*$ only
2. with the Gaussian emulator $\hat{y}(x) = f(x)^T\beta^* + r(x)^T R^{-1}(y - F\beta^*)$

where β^* is generic notation to denote the estimated β vector of parameters for the respective model employed. The different form of β^* for each model attributes to difference in the estimation method. A recount of the estimated β^* for each of the model is summarized as:

1. Smooth Ridge
$$\tilde{\beta}_\lambda = (F^T F + \lambda K)^{-1} F^T Y$$
2. Ridge model
$$\hat{\beta}_\lambda = (F^T F + \lambda I)^{-1} F^T Y$$
3. DACE model
$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} Y$$

The DACE model referred as Sacks model in the rest of the chapter is implemented in two ways: one in its original form named Sacks Linear while the other version is obtained by the use of Legendre basis and termed as Sacks Polynomial. For estimation of covariance parameter θ in Sacks linear and polynomial model, the package `DiceKriging` is used. The regularization parameter λ of Ridge and Smooth Ridge model is estimated by choosing the value of λ that minimizes the empirical mean square error i.e. $\lambda = argmin \sum (y(x) - f(x)^T\beta^*)^2$ for a grid of $10,000$ values for λ over the range of $[10^{-5},\ 10]$. For the estimated value of λ parameter vector β is then estimated for Ridge and Smooth Ridge models. Relative efficiencies are computed in terms of empirical mean square error of predictions.

$$MSE = \sum (y(x_i) - \hat{y}(x_i))^2 \tag{4.1}$$
$$RE = \frac{MSE(model)}{MSE(SR)} \tag{4.2}$$

where, $y(x_i)$ and $\hat{y}(x_i)$ are true and predicted response and *model* symbolizes the models to be compared with Smooth Ridge model. It is relevant to mention that supersaturated model(SSM) cannot be implemented for the case when $n \geq k$ owing to the fact that the methodology is valid only for supersaturated basis or when $n < k$. The methodology to run the computer simulations for the aforementioned models is delineated for each of the functions in the following sections. It is pertinent to mention here that SSM methodology is limited to the condition $n < k$, therefore no graphical presentation is obtained for $n \geq k$. A summary of different cases to be employed is given in the following table for a quick reference.

dimension	**fixed n**	**fixed k**	**test data points**
2	10	50	5
5	20	56	10
10	30	66	15
15	40	136	20
20	50	231	25

Table 4.1: Summary of different scenarios for computer simulations

4.2 Simulation Results for Bratley function: Fixed n

This section is devoted to elaborate the simulation results for the two cases given in Table (4.1) when the simulator chosen is Bratley et al.(1992) function from Derek Bingham library [8]. The function takes the form

$$y(x) = \sum_{i=1}^{d} (-1)^i \prod_{j=1}^{i} x_j \tag{4.3}$$

The input domain of the function for each variable x_i is $[0,\ 1]$ for $i = 1, \ldots d$. For the fixed size of design points n against each of the input dimensions d (Table(4.1)) different choices of basis to construct model terms and hence the design model matrix are given in Table (4.2).

relationship between design points and basis			
dimension	supersaturated basis $\mathbf{n} < \mathbf{k}$	saturated basis $\mathbf{n} = \mathbf{k}$	$\mathbf{n} > \mathbf{k}$
2	{21,36,55,66,70,80,90,100}	10	6
5	{21,30,45,56,70,80,90,126}	20	15
10	{40,50,66,80,90,100,286}	30	25
15	{50,60,80,90,100,136}	40	{25,30}
20	{60,80,90,100,231}	50	{30,40}

Table 4.2: Choice of basis for **n** fixed

In Table (4.2) $k = \{6, 10, 21, 55, 66\}$ are full basis of order $2, 3, 7, 9$ and 10 for $d = 2$. For $d = 5$, $\{k = 21, 56, 126\}$ are full basis of order $2, 3$ and 4. For $d = 10$, $k = 66, 286$ are full basis of order 2 and 3 whereas $k = 136$ and 231 are full basis of order 2 for $d = 15$ and $d = 20$ respectively. The computer simulations are run for 300 iterations. A detailed graphical account of relative efficiencies with Gaussian emulator is elicited for $d = 5$ and $d = 20$ in Figures (4.1, 4.2). The plots for $d = \{2, 10, 15\}$ are given in Appendix (A.4).

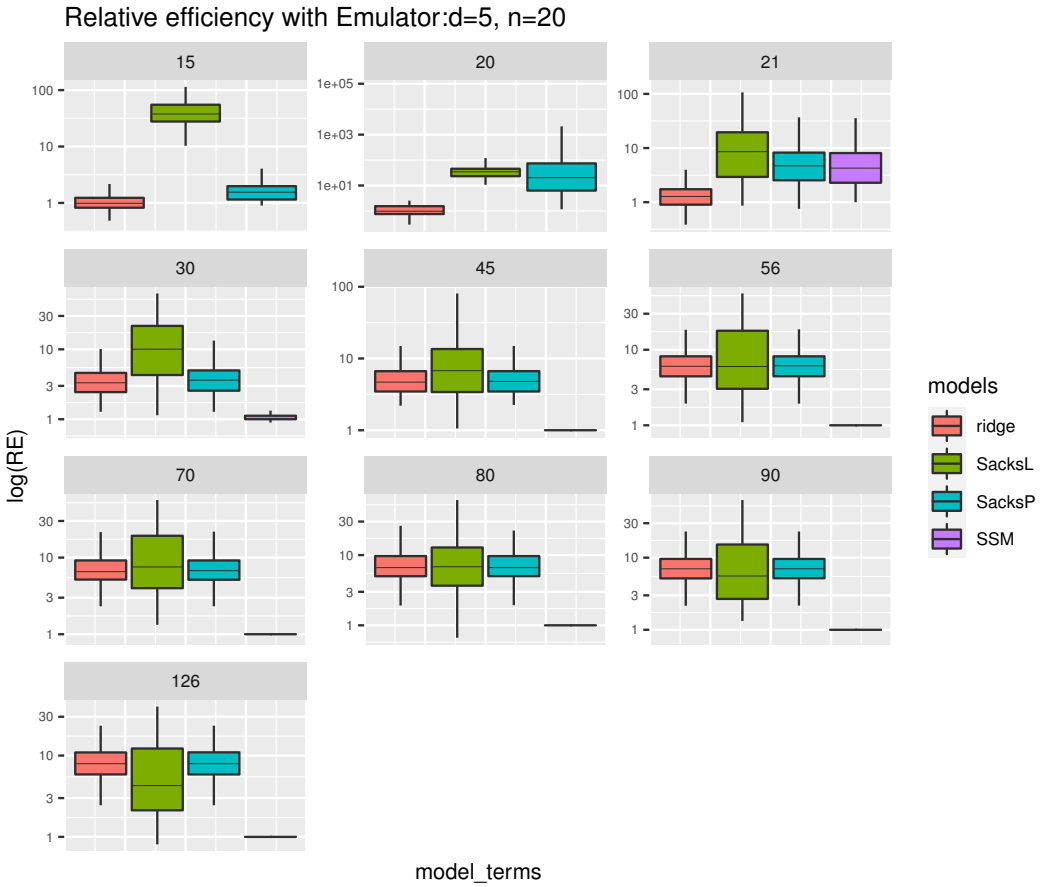


Figure 4.1: RE for d=5, n=20

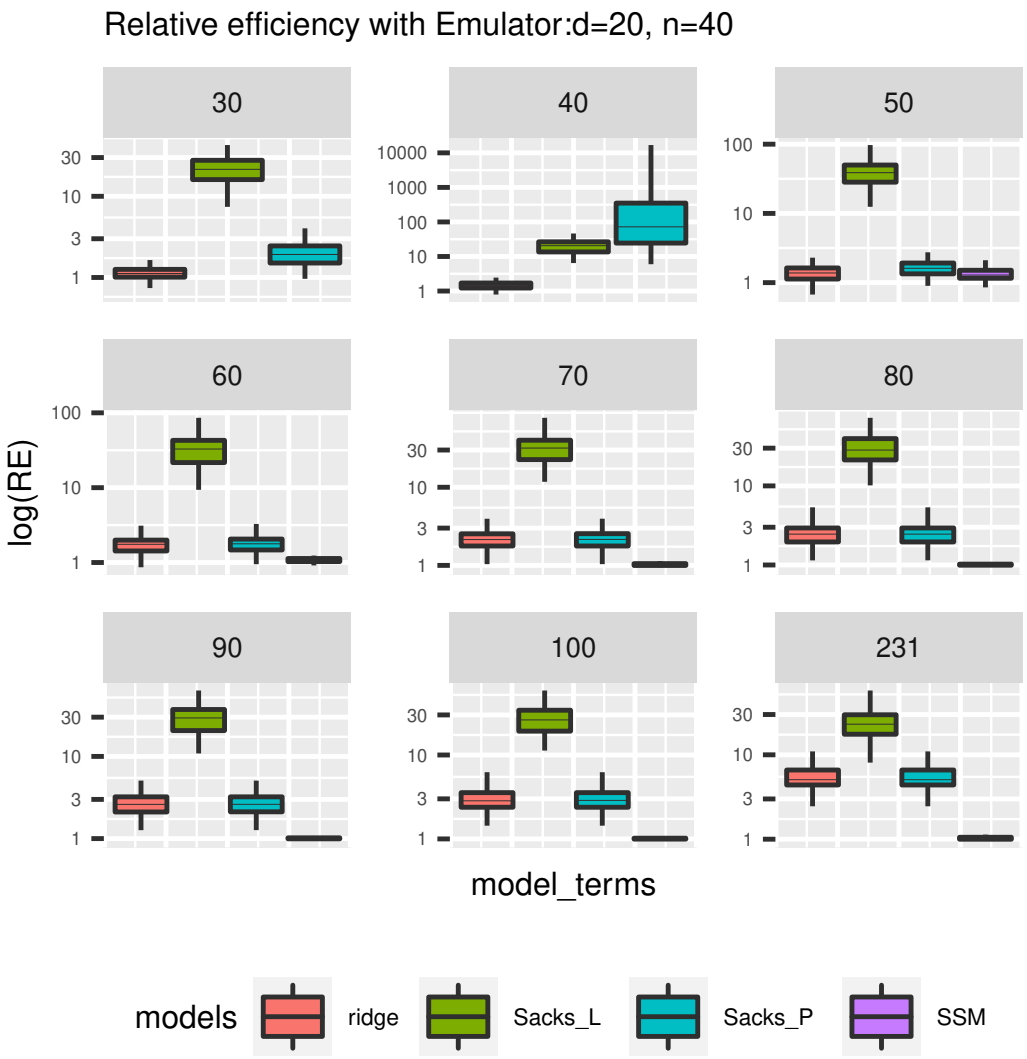


Figure 4.2: RE for d=20, n=40

4.2.1 Comments

For comparison of relative efficiency of Smooth Ridge model with the contemporary models when Bratley function is employed, we obtained RE plots for $d = \{2, 5, 10, 15, 20\}$. We discuss two of the cases here, one with less dimension $d = 5$ and the other with large number of dimension $d = 20$. We observe from Figure (4.1) that for $n > k$ Ridge regression performs equally well to that of Smooth ridge model whereas Sacks model with polynomial terms exhibit a close efficiency to that of Smooth ridge model. For $n = k$ Smooth ridge outperforms all models. For the case where $n < k$ we can infer that as k increases or in other words the larger

the k is from n the Smooth supersaturated model coincides with the Smooth ridge model in terms of the relative efficiency of prediction mean squared error. On the contrary all other models show less efficiency relative to Smooth ridge model.

We now look into the performance of Smooth ridge model when $d = 20$ Figure (4.2). The boxplots show the similar pattern as for $d = 5$ i.e. Ridge regression performs closely to that of Smooth ridge model when $n \geq k$. In contrast when k becomes large, the relative efficiency of Smooth supersaturated model to that of Smooth ridge model becomes equal to 1. It is interesting to note that Ridge model and Supersaturated models performs well relative to other models in different cases however none of these models outperforms Smooth Ridge model significantly.

4.3 Bratley Function: Fixed basis k

In this section a compendious study is carried out in order to build the computer simulations for the comparisons of Smooth Ridge model with contemporary models for a fixed number of model terms k against each of the input dimensions d chosen. These comparisons are done for the true function given in Equation (4.3). The choice of design points for fixed number of basis is made according to the input dimension of the underlying problem. The estimation of parameters are underpinned by the same details explained in Section (4.1). An exposition to build computer simulations for fixed k is provided for each of the dimensions (details in Table(4.1)). For fixed number of basis k, the choice of design points for each dimension are summarized in the Table (4.3).

relationship between design points and basis			
dimension	supersaturated basis $\mathbf{n < k}$	saturated basis $\mathbf{n = k}$	$\mathbf{n > k}$
2	$n = \{10, 20, 30, 40\}$	50	$n = \{60, 70, 80, 90$
5	$n = \{20, 30, 40, 50\}$	56	$n = \{60, 70, 80, 90$
10	$n = \{20, 30, 40, 50, 60\}$	66	$n = \{70, 80, 90,$
15	$n = \{20, 30, 40, 50, 60, 70, 80, 90, 100\}$	136	140
20	$n = \{20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200\}$	231	251

Table 4.3: Choice of design points for **k** fixed

The graphical account of the relative efficiencies computed from Equation (4.2) are given for $d = 2$ and $d = 15$ in Figures (4.3, 4.4). The plots for $d = \{5, 10, 20\}$ are dispensed in Appendix (A.4.1).

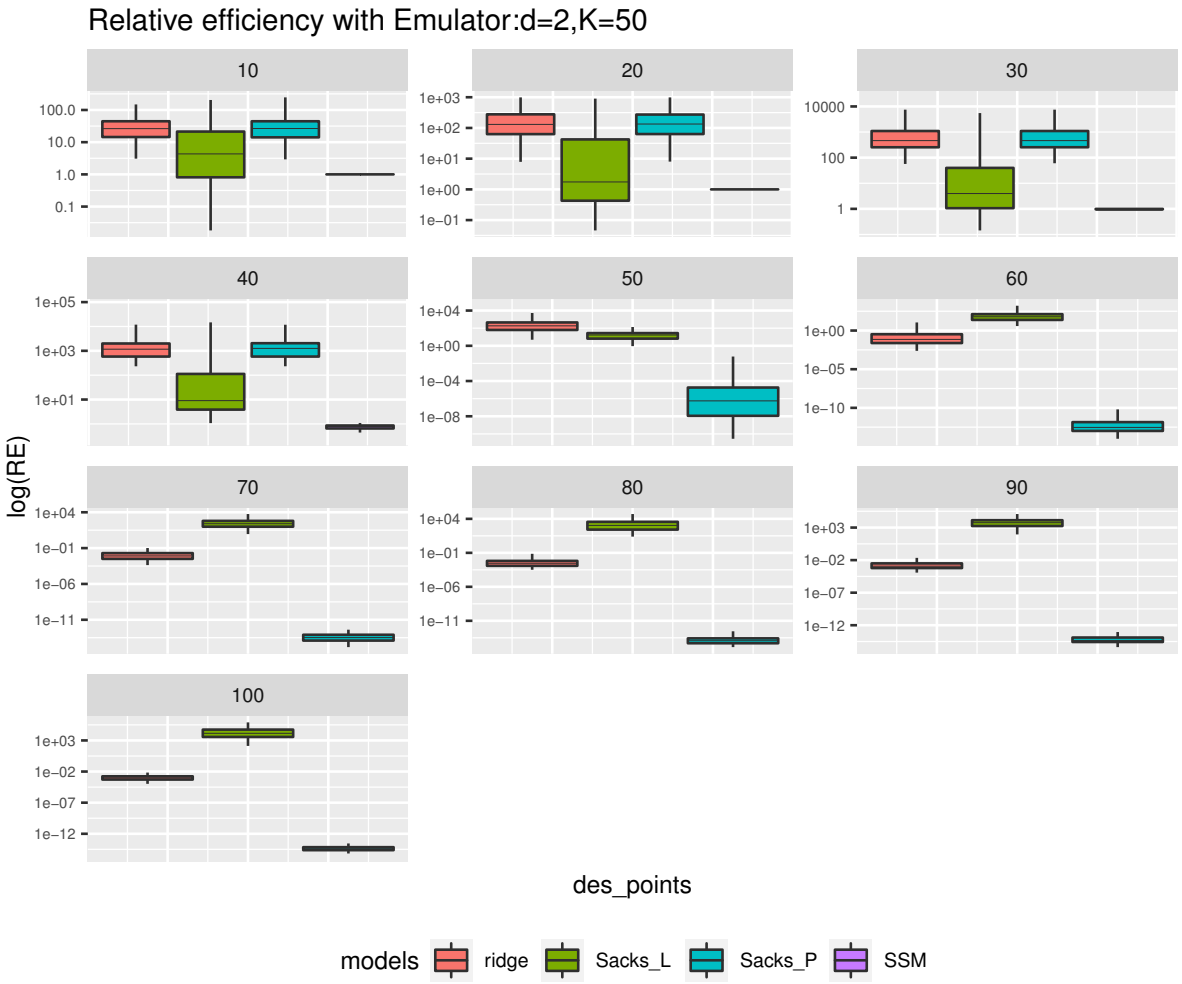


Figure 4.3: RE for d=2, k=50

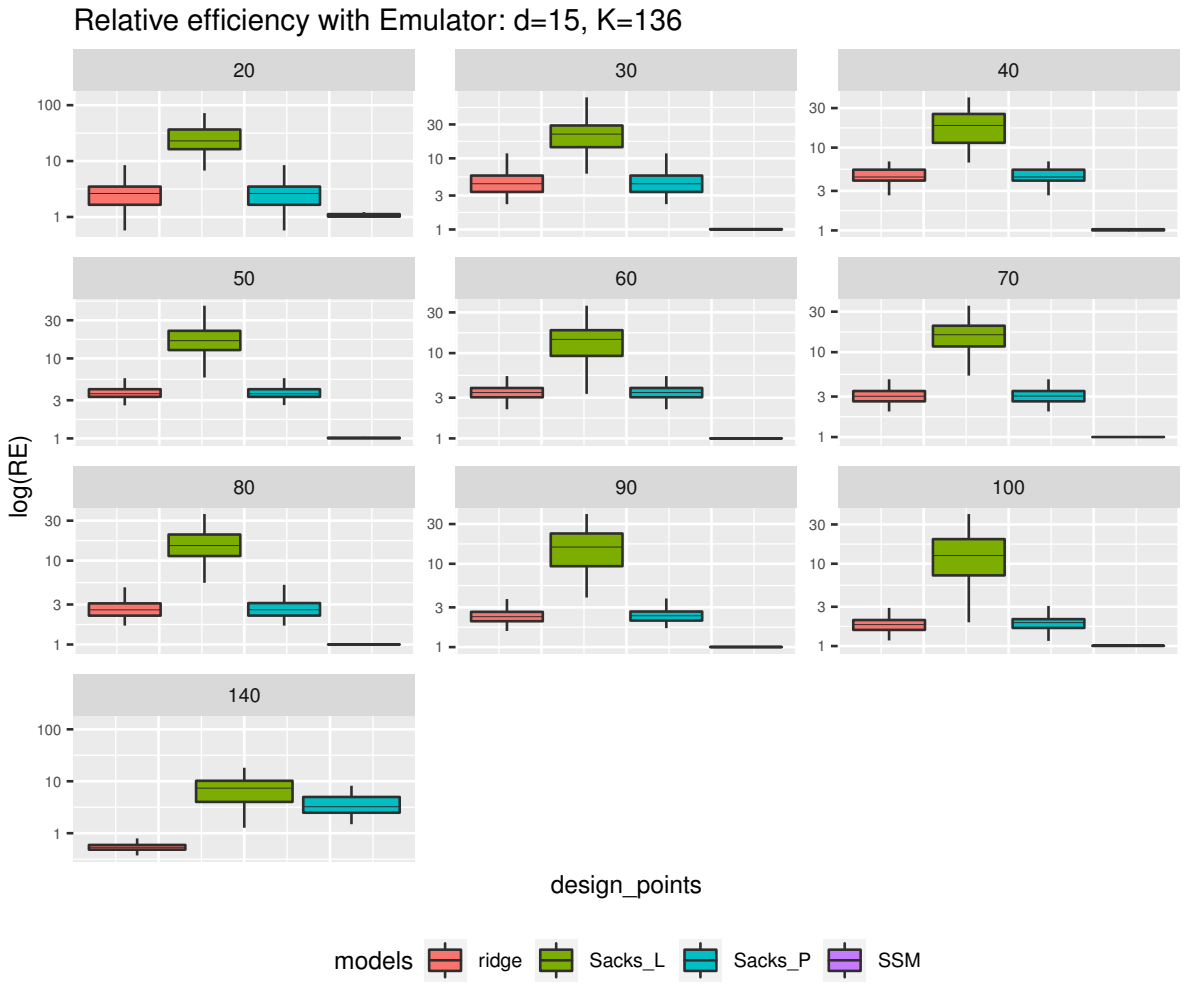


Figure 4.4: RE for d=15, k= 136

4.3.1 Comments

We discuss two of the results here for $d = 2$ and $d = 15$ when k is fixed and n is varying. From Figure(4.3) we can readily observe that for less number of dimensions Sacks model with polynomial terms outperforms Smooth Ridge model when $n \geq k$. In contrast Smooth supersaturated model performs similar to that of Smooth ridge model when $n < k$. It is important to recall here that Sacks with polynomial terms is different from Sacks model of DACE where linear terms are considered. Also the construction of trend in Sacks (polynomial) is based on Legendre polynomials similar to what is employed in Smooth Ridge model.

We now look at the plot for high dimension $d = 15$ in Figure (4.4) to draw some conclusions. For $n < k$ where $k = 136$ it is apparent that relative efficiency of Smooth supersaturated model

is similar to that of Smooth ridge model. Furthermore, ridge regression exhibits a better performance relative to all other models when $n < k$. It is important to mention here that Smooth Ridge model has an advantage over Super saturated model attributed to the fact that Smooth Ridge can be implemented for any choice of n and k.

4.4 LEVY Function: Fixed n

The comparative study for fixed design points n and fixed basis size k explicated in Sections (4.2) and (4.3) respectively is developed utilizing the Bratley function (4.3) as true function for the response variable. However, the Bratley function is supported by the polynomials which may cause the Smooth Ridge model to exhibit an improved performance when compared with other models. Nevertheless, the same model terms are used for all models under study, which should not cause any significant difference in performance measures solely on the choice of simulator. In order to further evaluate the performance of Smooth Ridge model relative to Ridge, SSM and Sacks models, an effort is made to use a function that is supported for all dimensions yet not favoured by the polynomials. One of the functions chosen to replace Bratley function is Levy function [39]. The function takes the form

$$f(x) = sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2[1 + 10sin^2(\pi w_i + 1)] + (w_d - 1)^2[1 + sin^2(2\pi w_d)] \qquad (4.4)$$

where,

$$w_i = 1 + \frac{x_i - 1}{4}, \quad \text{for all} \quad \text{i =1,. . . ,d.}$$

The function is evaluated for each variable x_i over the hypercube $[-10,\ 10]$ for all $i = 1, \ldots d$. In order to obtain predictions for untried data points for all models, comparison study is carried out in a similar fashion explained in preceding sections i.e.

- fix number of design points n
- fix number of basis k

This section narrates the details of computer simulations for fixed number of design points n. The simulation details follow from Section (4.1). The only difference is that the response vector y is computed by transforming the input variables over the hypercube $[-10\ 10]^d$ in order to use Levy simulator. The fixed number of design points n considered against each dimension

is appended in Table(4.1).The choice of basis against fixed design points for each dimension is explicated in Table (4.2) and the same is followed for Levy function. Number of simulation runs are 300 and prediction mean square error and relative efficiency are computed in a similar fashion given in Equation (4.1). The graphical description of relative efficiency for each of the model and model terms for $d = 2$ and $d = 15$ are depicted in Figures (4.5, 4.6) respectively. The plots for $d = \{5, 10, 20\}$ are appended in Appendix (A.5).

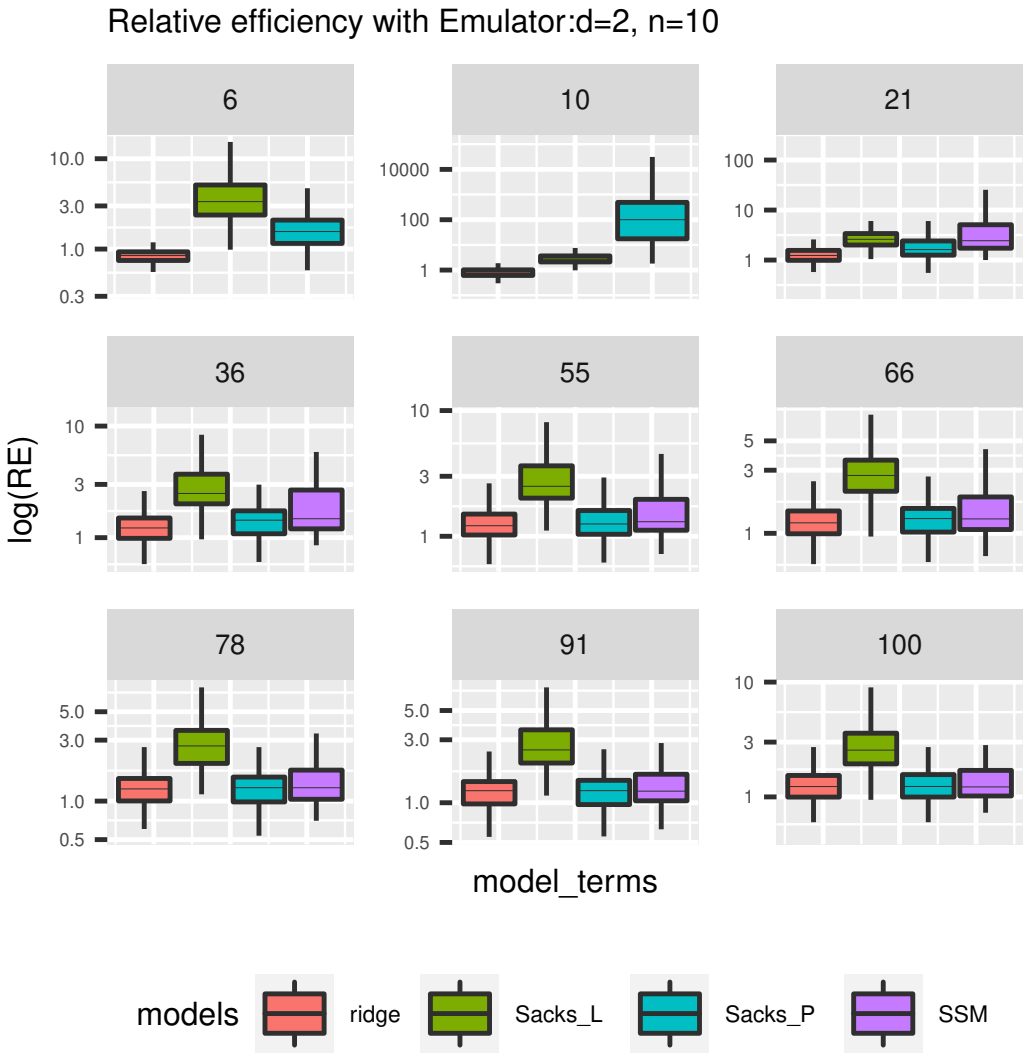


Figure 4.5: RE for d=2, n= 10 (Levy)

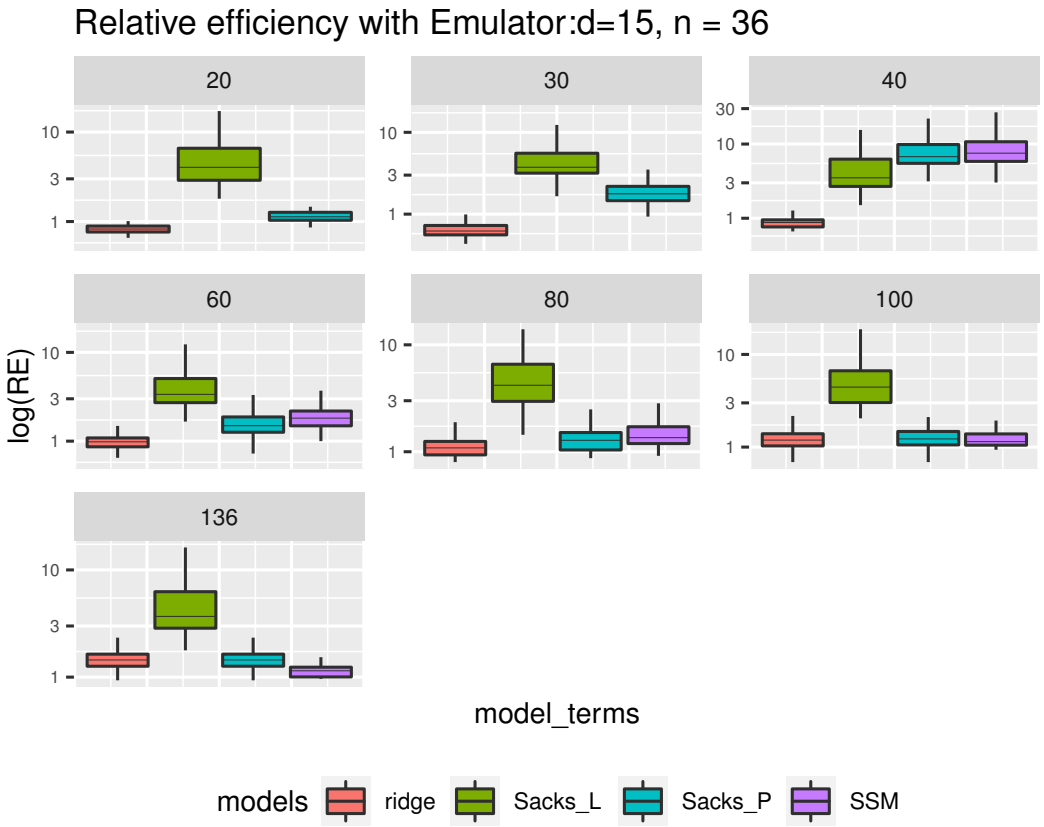


Figure 4.6: RE for d=15, n= 36 (Levy)

4.4.1 Comments

We highlight two cases for $d = 2$ and $d = 15$ respectively. For $d = 2$ we can infer from Figure (4.5) that ridge model performs slightly better than smooth ridge when $n > k$. The Smooth Ridge model outperforms the rest of the models as k becomes greater than n.

For $d = 15$, when $n \geq k$ for example $k = (6, 10)$ in Figure (4.6) the relative efficiency of Smooth ridge and Ridge model is close to 1 and outperforms Sacks model with linear and polynomial terms. For $n < k$ Smooth ridge outperforms other models with Ridge model as a close competitor. We also observe that Sacks model (polynomial) and Smooth supersaturated model perform very similar to each other relative to Smooth ridge model. Also, ridge model remains very close to Smooth ridge model in terms of prediction performance. Explicitly, Ridge model outperforms Smooth ridge model for $n > k$ which remains the case for $k = 40$ when there are 4 more terms than the design points. For $k = 60$ Ridge model and Smooth ridge model perform relatively similar to each other. For $k = (80, 100, 136)$ Smooth ridge model

outperforms all other models though the Smooth saturated model becomes closer to Smooth ridge model when difference between the number of terms k and the design points n increases.

4.5 LEVY Function: Fixed k

Section (4.4) is devoted to illustrate the comparison study of Smooth Ridge model with the contemporary models for fixed design size n when Levy function is employed. In this section, the comparisons are done for fixed number of basis k against different choices of design points n using the Levy function. The number of basis are selected taking into account the number of dimensions undertaken (Table 4.1) and the suitable choice of design points is made according to the details furnished in Table(4.3). The details on the computer simulation are followed from Section (4.1) which are not re-accounted for the sake of brevity. The relative efficiency (RE) in Equation (4.2) is evaluated for each of the model against Smooth Ridge model and resulting graphical description for $d = 2$ and $d = 15$ is furnished in Figures (4.7, 4.8). The plots for $d = \{5, 10\}$ are given in Appendix (A.5.1).

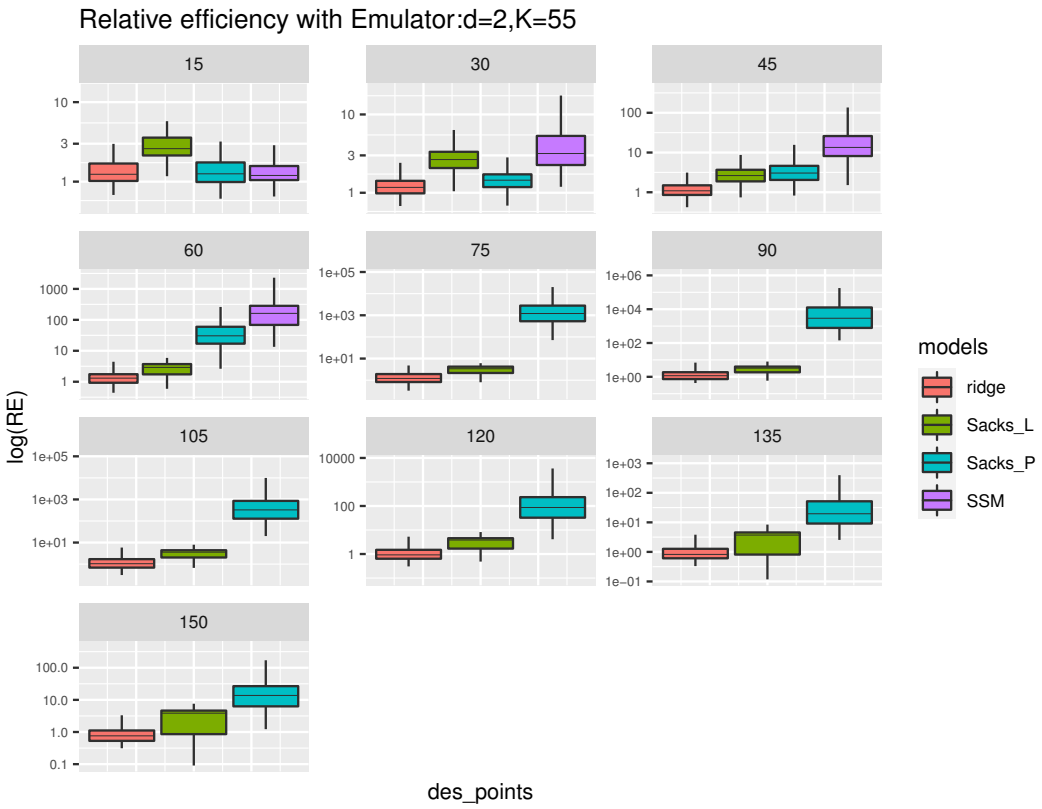


Figure 4.7: RE for d=2, k=55 (Levy)

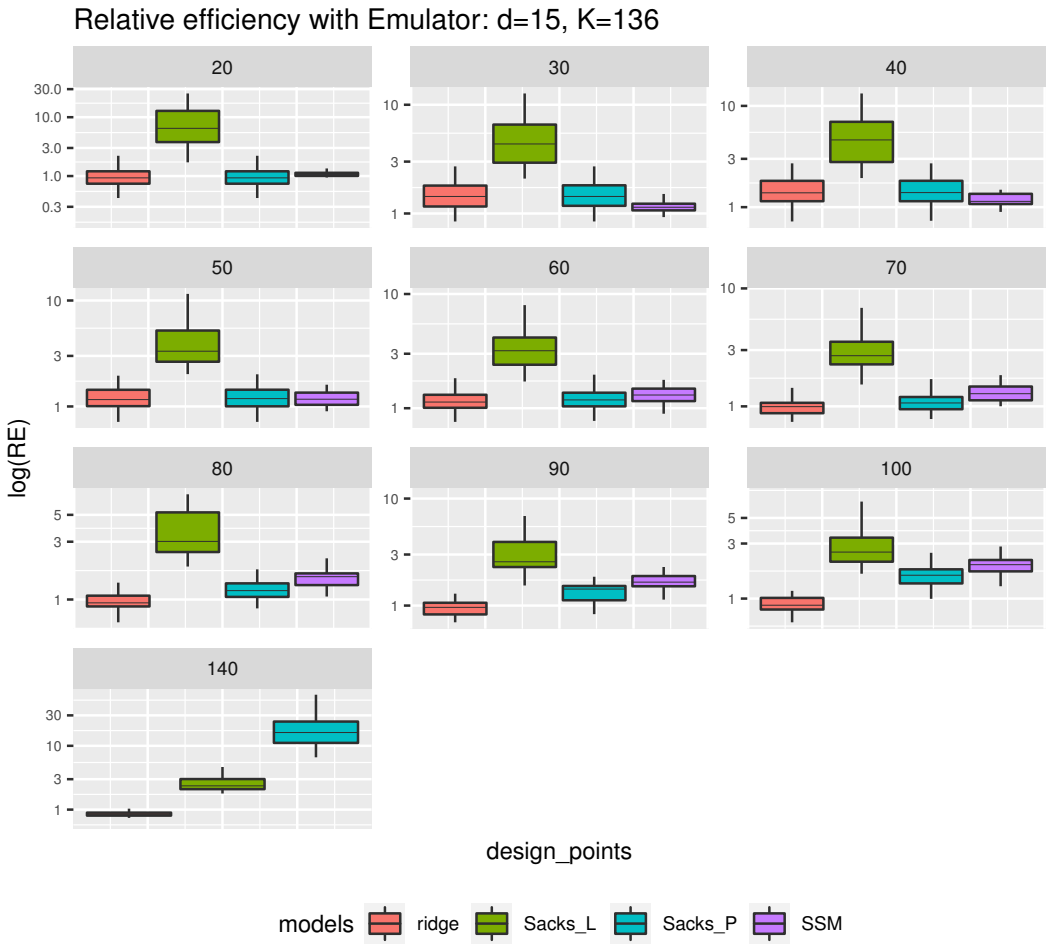


Figure 4.8: RE for d=15, k=136 (Levy)

4.5.1 Comments

We now interpret the relative efficiency plots for $2-d$ and $15-d$ comparisons presented in Figures (4.7, 4.8). It is apparent from figure (4.7) that smooth ridge model beats all other models when $n < k$. As n increases the relative efficiency of ridge regression becomes closer to 1. It is noticeable that Sacks model (linear) performance is very close to that of ridge model relative to smooth ridge model and the agreement gets closer as n increases.

For $d = 15$ in Figure (A.14) we observe that the instances where difference between the number of design points and model terms is large, smooth supersaturated model performance coincides with that of Smooth Ridge model which outperforms all other models. In contrast, when n increases for example $n \geq 60$ ridge model and smooth ridge model becomes closer in terms of relative efficiency leading the other models.

4.5.2 Summary of the conclusions

We summarise the conclusions drawn from the comparisons for Bratley and Levy functions. In summary when $d = 2$, relative efficiency of Sacks model (polynomial) and smooth ridge model is close to 1 for $n > k$ and $n < k$ respectively. For $d = 15$ relative efficiency of Ridge regression model and smooth saturated model is close to 1 for $n > k$ and $n < k$ respectively. We infer from these comparisons that Smooth ridge model remains the most efficient model in all cases.

4.6 COVID data modelling: Comparisons

An application of Smooth Ridge model is sketched in Chapter (3) for the estimation of aerosol transmission of COVID-19 virus. In this section, detailed comparisons are performed for the same application to get insights into the predictive capabilities of all models under study in a real life application.

In Chapter (3), the inputs for Aerosol Transmission Estimator are categorised as fixed and variables respectively. In this section, the comparisons are done with the same set of fixed inputs where a classroom scenario is considered. The four variable input take a grid of values, references of which are borrowed from the COVID19-Aerosol Transmission Estimator and [47]. The tabular description of both type of inputs are given in Table (3.8). The simulator is the response variable i.e. probability of infection given as $y = 1 - e^{-QIP}$ [62].

4.6.1 Data setup for comparisons

In this section the methodology underlying the development of computer simulations for COVID data is elucidated. These simulations are formulated to compare the performance of Smooth Ridge model with Ridge, Smooth Supersaturated model, DACE model (Sacks(linear)) and Sacks(polynomial). For a given set of input $x \in \mathbb{R}^4$, design model matrix F, response vector $\mathbf{y}$ and regression parameters β, the comparisons are done for predictions at untried points, taking into account the following as predicted response

- trend part (estimator) of the model $F\beta^*$ only
- the Gaussian emulator $\hat{y}(x) = f(x)^T\beta^* + r(x)^T R^{-1}(y - F\beta^*)$

where β^* symbolizes the estimated vector of regression parameters pertaining to the model employed enlisted in Section (4.1). The four variable inputs variables namely: ventilation, decay rate of virus, deposition to surfaces and quanta rate are randomly generated with Latin

hypercube sampling. These data are transformed into the respective grid of reference values for each of the inputs, presented in Table(3.8). Legendre polynomials are evaluated on the input data over the hypercube $[-1\ 1]^4$ giving rise to design model matrix F.The models are fit to training data and predictions are made on the test data with the parameter estimates obtained from training data. The allocation of data into training and test sets is accomplished with the help of cross validation method in the ratio of $2:1$ as described in earlier sections of this chapter. The comparisons for the above explained details, are done for two cases

- fixed design size n
- fixed basis size k

The number of basis for the composition of Legendre Polynomials for fixed design points and various choices of basis for a fixed design are given in the table below

relationship between design points and basis			
	supersaturated basis $\mathbf{n < k}$	saturated basis $\mathbf{n = k}$	$\mathbf{n > k}$
fixed design size $n = 30$	$k = \{35, 45, 56, 70, 85, 95, 126\}$	30	20
fixed basis $k = 70$	$n = \{20, 30, 40, 50, 60\}$	70	$n = \{90, 100\}$

Table 4.4: Choice of design points and basis for COVID data study

4.6.2 Computer Simulations for fixed n

There are four input for this study so that $x \in \mathbb{R}^4$. The design size fixed at $n = 30$, is used as training data, whereas 15 data points are taken as test data. The data generated is iterated 100 times and at each iteration the split of data into test and training sets is repeated three times for 3 cross validation folds. Figure 4.9 depicts the graphical narration of the relative efficiency of prediction mean square error for the models under study relative to Smooth Ridge model.

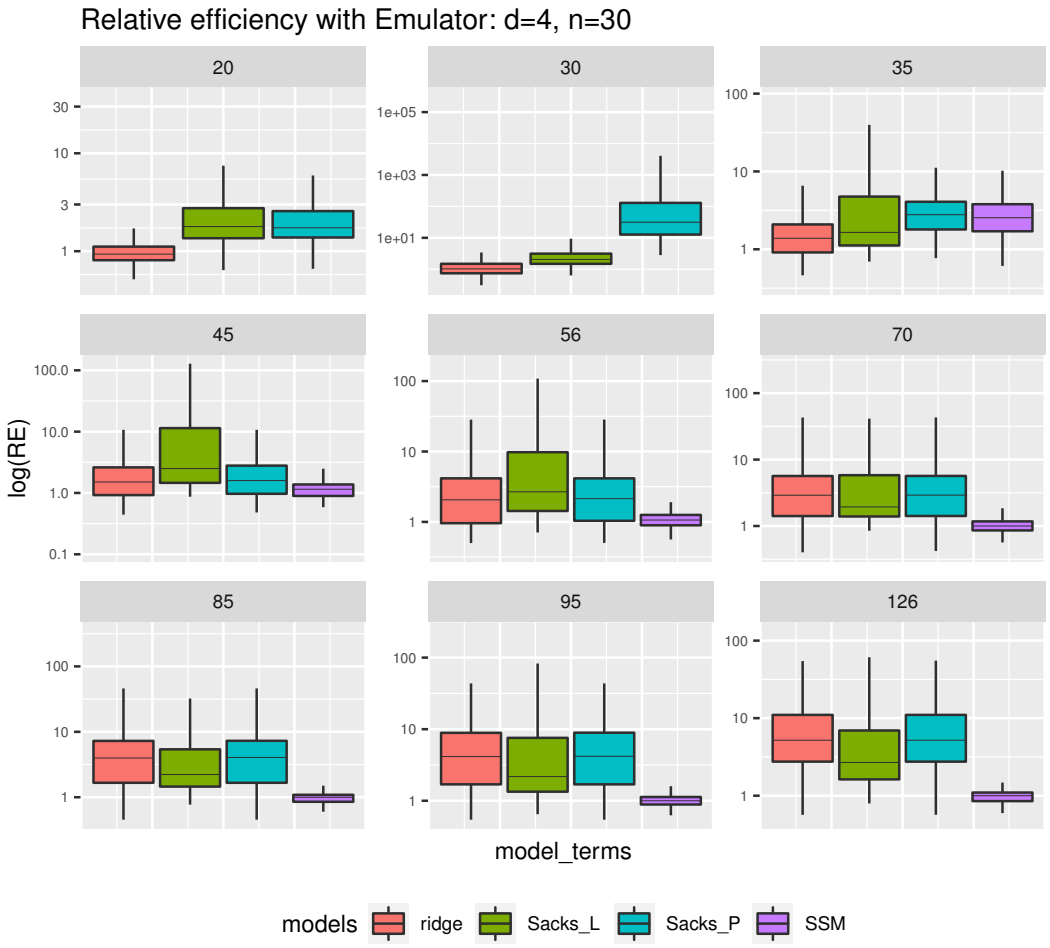


Figure 4.9: RE for COVID data ($n = 30$)

4.6.3 Computer simulations for fixed k

Contrary to the comparisons for fixed design size, different sizes of design n are considered for fixed number of basis k. Full basis of order 4 is considered giving $k = 70$. A random data of a given size n and $\frac{n}{3}$ is divided into training and test sets with CV folds.The process is repeated over 100 iterations and in each iteration, the data split is repeated for all three possible CV folds, whereby 2 folds act as training and 1 fold as test data.Various choices of design size are provided in Table (4.4). The graphical presentation of the relative efficiency of prediction mean square error for the models under study relative to Smooth Ridge model are given in Figure (4.10).

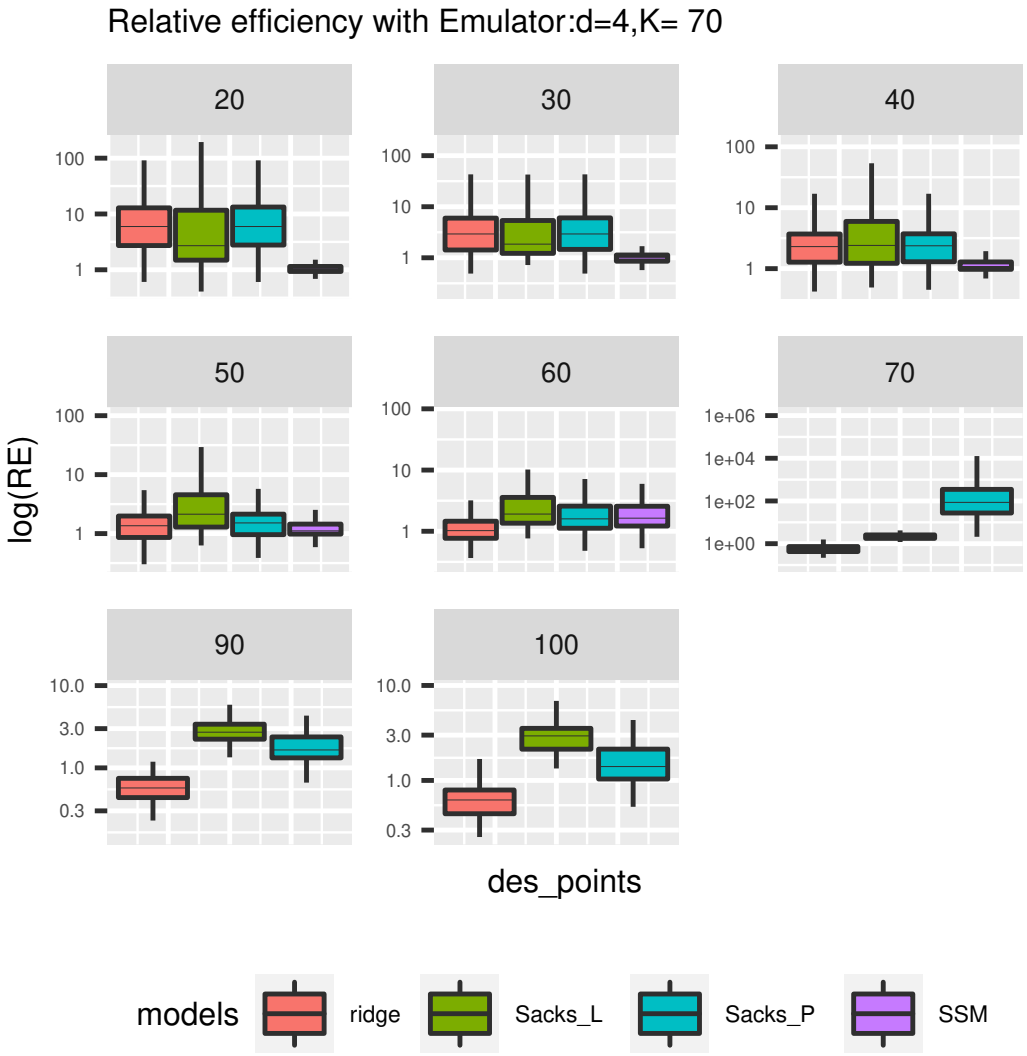


Figure 4.10: RE for COVID data ($k = 70$)

4.6.4 Comments

From Figures (4.9, 4.10) we can see the similar pattern in both the plots. For the cases $n < k$ when difference between n and k increases, the agreement between smooth supersaturated model and smooth ridge model becomes closer. Also, the two models supersedes the other models in terms of predictive performance. On the other hand when $n \geq k$ ridge regression leads in predictive performance as n is further away form k yet close to smooth ridge model. For small difference between n and k smooth ridge and ridge models perform equally well as compared to other models.

4.7 Comparisons for Temperature data

In Section (2.2.4), a comprehensive introduction of temperature data is accounted which is recapitulated in this section in order to assess the comparative performance of the models under study. The data comprises of 100 year temperature forecasts segmented for each of the 12-months. The input variable is $2-d$ grid of latitude and longitude making a total of 503910 data points. Among the enormous data set, the small part of data is considered for the study similar to that given in Section (2.2.4). In order to examine the predictive capabilities of models under study, one month temperature data is considered for one year over the entire grid of input variables, latitude and longitude. The resulting data contains 5599 data points. It is relevant to mention here that Smooth supersaturated model requires $n < k$, which explicitly states that number of design points should be less than the model terms k, which is cumbersome to fulfil in presence of large data set. Partly because, a small sample do not capture the true behaviour of the large data in an effective manner and partly due to the computational limitations of `ssm` package in `R`. The comparison details follow the ones set out in the preceding sections. Consequently, the results of relative efficiencies with Gaussian emulator are graphically extended here and can be found in Figures (4.11, 4.12).

4.7.1 Computer simulations for n fixed and k fixed

For temperature data the two input variables are latitude and longitude.Therefore, for $x \in \mathbb{R}^2$, the design points are fixed at $n = 56$. Since, the size of temperature data 5599 is very large therefore, It is of interest to delve into the models' behaviour when number of design points are increased while keeping the model terms fixed.The model terms are fixed to be $k = 100$ so as to allow more number of design points to be chosen for the comparisons.The following table presents the details on the relationship between design points and model terms.

relationship between design points and basis			
	supersaturated basis $\mathbf{n < k}$	saturated basis $\mathbf{n = k}$	$\mathbf{n > k}$
fixed design size $n = 56$	$k = \{21, 36, 45, 50\}$	56	$k = \{66, 78, 91, 100\}$
fixed basis $k = 100$	$n = \{28, 37, 45, 62\}$	100	$n = \{150, 200, 250, 300, 400, 500\}$

Table 4.5: Choice of design points and basis for Temperature data

where, $k = \{36, 45, 66, 78, 91\}$ are full basis of order $7, 8, 10, 11$ and 12. Box-plots of relative efficiency for fixed design and fixed model terms are furnished in Figures (4.11, 4.12).

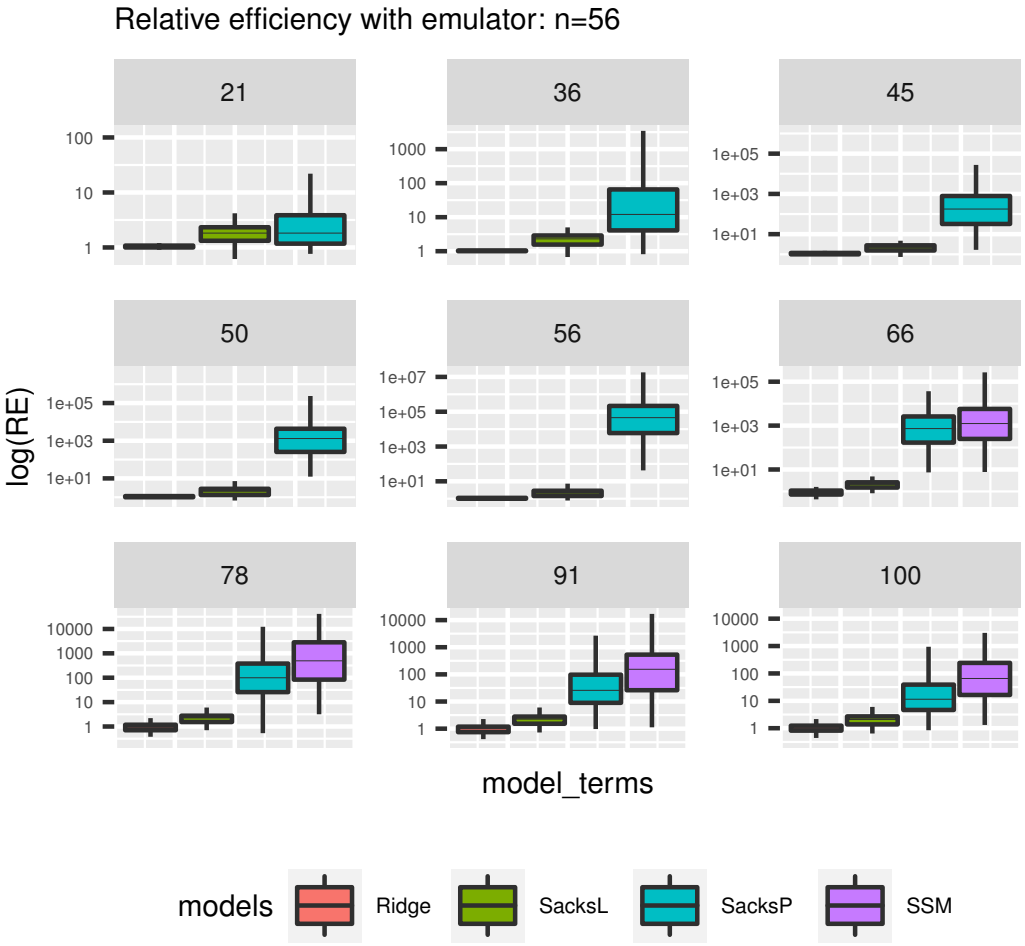


Figure 4.11: RE for Temp data: fixed n

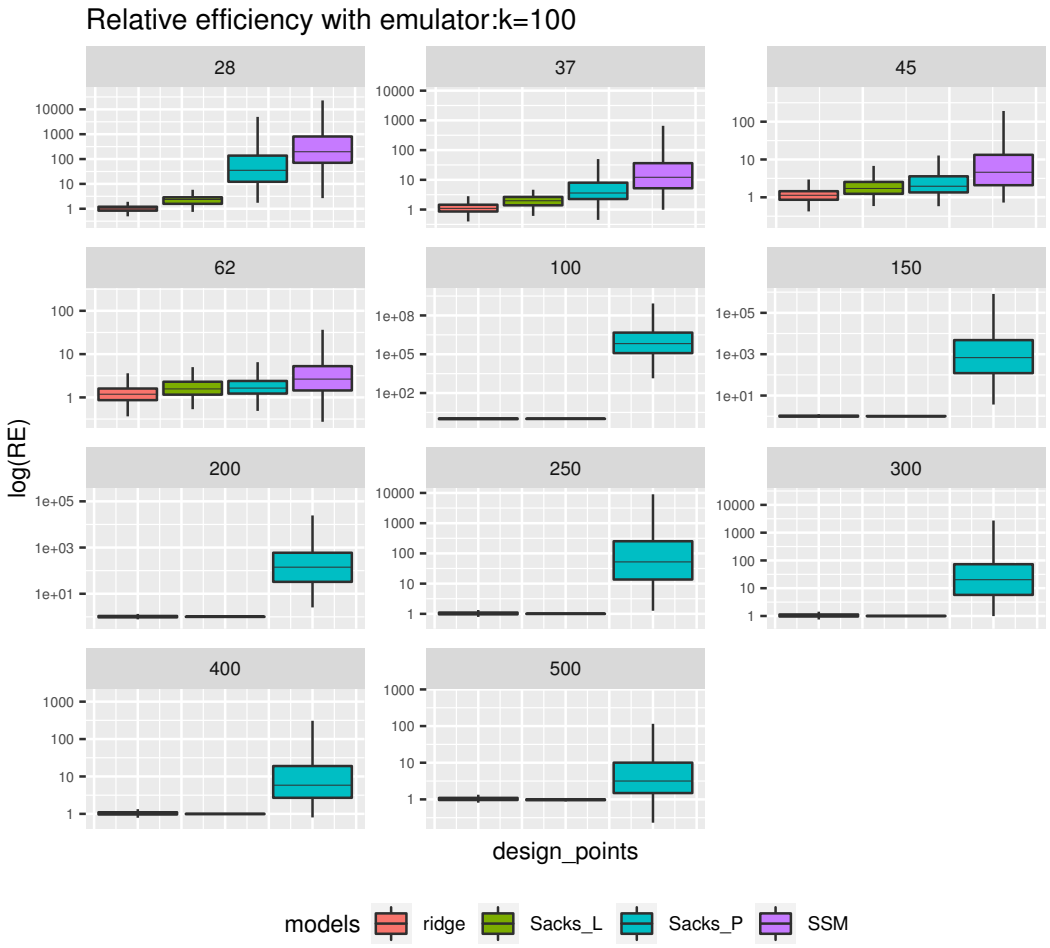


Figure 4.12: RE for Temp data: fixed k

4.7.2 Comments

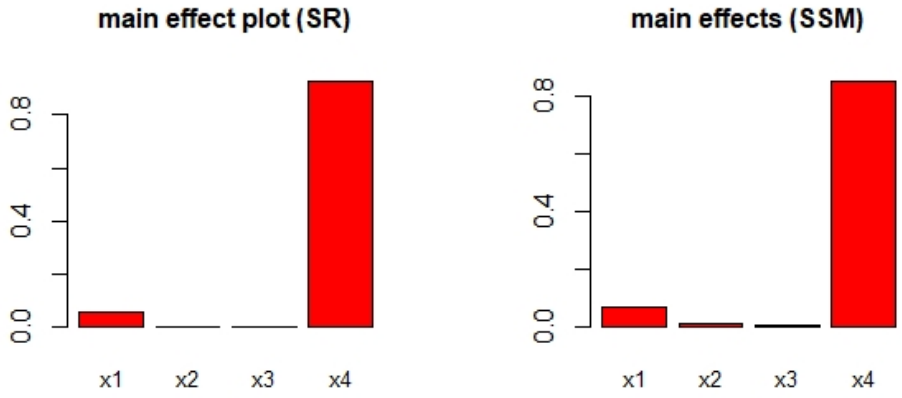
For temperature data it is pertinent to mention that there are only two input variables therefore $d = 2$. From Figures (4.11, 4.12) it can be readily observed that for $n > k$, the relative efficiency of Sacks model (linear) and Ridge model is relatively close to each other and as n becomes large Figure(4.12) the relative efficiency of both the models is equal to 1. This suggest that Smooth ridge model, Ridge model and Sacks(linear) model exhibit similar predictive outcomes for large number of data points and less model terms k. The relative efficiency of ridge regression is close to 1 when $n < k$.

4.8 Senistivity Indices: Comparisons of Smooth Ridge and SSM

A comprehensive introduction of sensitivity analysis is catered in Section (3.10). Following the methodological development of Sobol' sensitivity indices in Section (3.10), this section is devoted to the computation of sensitivity indices for comparison purposes by employing Smooth Ridge model and Smooth Supersaturated model. Two aforementioned real-life problems namely COVID data study and Temperature data are revisited for the computation of sensitivity indices. The details pertaining to COVID data and Temperature data study are furnished in Sections (4.6) and (4.7) respectively and therefore not re-accounted for the comparison of sensitivity analysis.

4.8.1 Sensitivity Analysis for COVID data

The probability of catching the disease through aerosol transmission of SARS-CoV-2 virus is modelled in Section (4.6). In order to study and compare sensitivity indices, that quantifies the relative contribution of each of the four variables and their interactions, sensitivity analysis enacted in Section (3.10) is carried out. The choice of design points n and number of model terms k are kept fixed for the study. Consequently, 45 data points are generated with Latin Hypercube Sampling whereby 30 data points are fixed as design points and remaining 15 points serve as test data. The model terms are constructed with 35 Legendre basis. Sensitivity indices are computed for Smooth Ridge model and Smooth Supersaturated model with respect to the Equation(3.52) in Chapter (3). The difference in the variance computation for both models lies in the estimated parameter vector β. For COVID data, there are four variables resulting into 14 sensitivity indices, 4 corresponding to main effects, 6 double interaction effects and 4 three order interactions. The plots and tables of the sensitivity indices for Smooth Ridge model and Supersaturated models are given in Figure (4.13) and Tables (4.6, 4.7).



(a) Sobol' Indices for main effects: SR (b) Sobol' Indices for main effects: SSM

Figure 4.13: Sobol' Indices for main effects

Sobol' Indices for main effects: SR (%)	
x_1	5.94
x_2	0.13
x_3	0.18
x_4	92.58

Table 4.6: main effects (SR)

Sobol' Indices for main effects: SSM(%)	
x_1	7.21
x_2	1.08
x_3	0.71
x_4	85.25

Table 4.7: main effects (SSM)

Figure 4.14: order 2 interactions for SR and SSM

Sobol' Indices for interaction effects: SR(%)	
x_1x_2	0.0058
x_1x_3	0.334
x_1x_4	0.29
x_2x_3	0.0174
x_2x_4	0.037
x_3x_4	0.1196

Table 4.8: order 2 interaction effects (SR)

Sobol' Indices for interaction effects: SSM(%)	
x_1x_2	0.0072
x_1x_3	0.14
x_1x_4	0.073
x_2x_3	0.032
x_2x_4	0.0069
x_3x_4	0.021

Table 4.9: main effects (SSM)

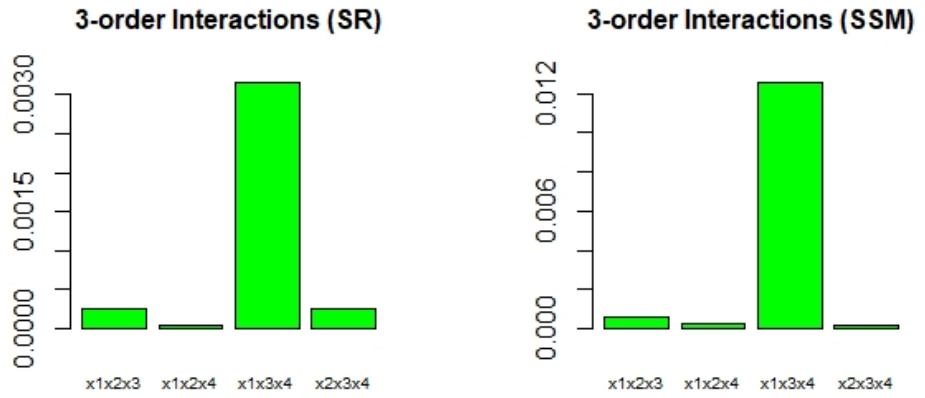


Figure 4.15: order 2 interactions for SR and SSM

Sobol' Indices for order-3 interactions: SR(%)	
$x_1x_2x_3$	0.0256
$x_1x_2x_4$	0.042
$x_1x_3x_4$	0.315
$x_2x_3x_4$	0.0026

Table 4.10: order 2 interaction effects (SR)

Sobol' Indices for order-3 interactions: SSM(%)	
$x_1x_2x_3$	0.0038
$x_1x_2x_4$	0.0019
$x_1x_3x_4$	0.0809
$x_2x_3x_4$	0.00137

Table 4.11: main effects (SSM)

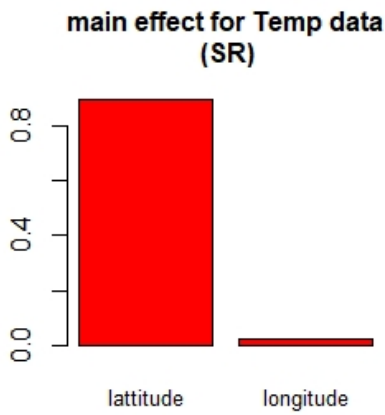
4.8.2 Comments

From Figures (4.13a, 4.13b), we observe that x_4 or quanta rate is the most influential variable. The main effect quanta rate contributes 92% to the total variation in Smooth Ridge model and 85% in Smooth supersaturated model. The sensitivity index for two order interactions is

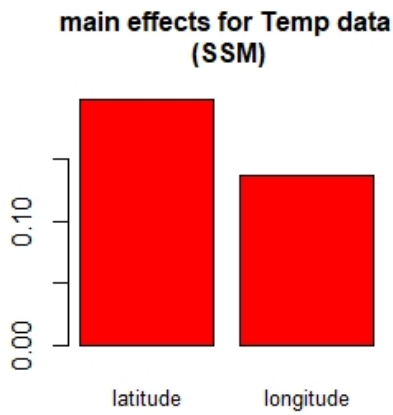
highest for x_1x_3 or ventilation $\times$ decayrate in Figures (4.14) for Smooth Ridge and Smooth Supersaturated models respectively. Similarly for three order interactions, the most contribution to total variation is attributed to $x_1x_3x_4$ or ventilation$\times$decayrate$\times$quanta rate from Figure (4.15) for both models.

4.8.3 Sensitivity Analysis for Temperature Data

The comparison study of Smooth Ridge model with the other models for temperature data, is expounded in Section (4.7). The similar data set is considered for employing sensitivity analysis for Smooth Ridge model and Supersaturated model. However, unlike in Section (4.7), both the number of design points n and model terms k are kept fixed in order to evaluate sensitivity indices. For a fixed design of 63 grid values of input variables, latitude and longitude, $k = 100$ model terms are chosen. The unknown parameter vector β is estimated with the respective methodologies of Smooth Ridge and Supersaturated models, accounted in chapter(3). Since,the temperature data comprises two inputs therefore there are two main indices and one interaction effect. The resulting plots are given in Figure (4.16) and Table (4.12) for both the models.



(a) Main effects: SSR



(b) Main effects: SSM

Figure 4.16: main effects for temperature data

Sobol' Indices for Temperature data (%)		
	SR	SSM
lat	89.34	19.80
long	2.60	13.65
lat*long	8.064	66.56

Table 4.12: Sobol' Indices for Temperature data

4.8.4 Comments

From Figure (4.16) and Table (4.12) we infer that latitude has highest contribution (89%) to total variation in Smooth ridge model and 19% in Smooth supersaturated model. There are only two inputs for this data therefore we have only one 2-order interaction effect which contributes 8% and 66% to the total variation for Smooth ridge and Smooth supersaturated models.

Chapter 5

Designs for computer models

Computer experiments have been gaining a paramount attention for last few decades and an intensive amount of research effort has been dedicated to select inputs at which to compute the output of a computer experiment. The experimental region is the region corresponding to the values of the inputs over which one wishes to study or model the response. A specific set of values of the inputs is said to be a point in the region. Hence, an experimental design is the selection of points in the experimental region at which the response is to be computed. Typically the computer experiments differ from physical ones in that the given set of inputs yield equal response in different runs. In computer experiments the functional form that describes the relationship between the inputs and the response is unknown. Therefore, uncertainty is attributed to the approximation of the exact relationship. The discrepancy between the response from computer code and model prediction is referred as error or model bias. Taking into account the problem of model bias, following are the principles of design selection [69].

1. For a given set of inputs, designs should not take more than one observation with the assumption that computer code remains unchanged over time. In other words, the replications do not allow to estimate variability because the response remains unchanged.

2. Because the true relation between the response and the inputs is unknown, designs should be flexible to allow fit different models. Explicitly, when one wants to explore different models for the underlying problem they are interested to look for, the the designs should be flexible to be employed in all models. In addition, design should exhibit information about entire experimental region.

5.1 Space-filling designs

Suppose that the key features of true model are equally likely in all parts of experimental region then it is reasonable to use designs that spread the points evenly across the region. Different ways of defining the evenly spread points lead to different types of designs, all of which are referred as *space-filling* designs [69]. The space-filling designs are likely to be employed when interest lies in the prediction accuracy because interpolators are used as predictors. Since, the prediction error at any input is a function of its distance relative to the design points, therefore the designs which are not space filling may yield poor predictions. It is shown in [80] that no design will outperform certain space-filling designs in terms of the rate at which the maximum of the mean-squared prediction error decreases as sample size increases. There are different strategies to chose designs which spread evenly throughout the experimental region. Some of the design methods are given below.

5.1.1 Simple random sample design

One of the method is to superimpose a regular grid on the experimental region. For example, consider the experimental region as a unit square $[0,1]\times[0,1]$. In order to observe response at 25 evenly spaced points the following grid of points $\{0.1,0.3,0.5,0.7,0.9\}\times\{0.1,0.3,0.5,0.7,0.9\}$ is considered. One of the possibilities to achieve such a design is to use simple random sampling. However, the numbers need to be recorded to finite decimal places so as to regard the number of points between 0 and 1 as finite. Also, the simple random sample may not be appropriate to use for small sample size because in high-dimensional experimental regions the sample is prone to clustering and unable to capture entire portions of the region.

5.1.2 Stratified sample designs

In high-dimensional experimental regions with small samples, the sample is prone to exhibit some clustering and fail to distribute points evenly across the experimental region [69]. An alternative approach is to use Stratified random sampling which suggests to divide the experimental region to n equally spaced strata and selecting one point from each stratum. The stratified sampling allows flexibility in selecting a design by varying the size and position of the strata and specifying different distributions for sampling. This can be useful if some portions of the experimental region are of greater interest than others. On the other hand if the interest lies in the even spread of sample points, equal spacing of strata and sampling according to a uniform distribution appears to be a viable choice.

5.1.3 Uniform designs

Uniform designs have been widely adopted for computer experiments. These designs are constructed by assuming a uniform distribution over the design region. Uniform designs draw observations at a set of points that minimize some discrepancy measure. One of the most well known measure of discrepancy is the Kolmogorov-Smirnov statistic. Now suppose that output is expected to depend on few of the inputs or factors, then the interest lies in that the points are evenly spread onto these factors across the projection of experimental region. It is cumbersome to guarantee such a design with simple random sampling and Stratified sampling.

5.1.4 Latin Hypercube Design

Latin Hypercube Design introduced by [45] is the solution to the problem described above. This method chooses the samples to ensure that the points are evenly spread over the values of each variable and not only over the entire unit square. For the unit square $[0,1]^2$ experimental region, each axis $[0,1]$ is divided into n equally spaced intervals $[0, \frac{1}{n}, \ldots, \frac{(n-1)}{n}, 1]$ which partition the unit square into n^2 cells of equal size. These cells are filled with integers $1, 2, \ldots n$ in such a way that each integer appears exactly once in each row and column giving rise to Latin Hypercube Square arrangement. Then one integer is selected followed by selecting a point at random from the cells containing that integer. Thus the sample of n points selected is Latin Hypercube Design of size n. Latin Hypercube Designs are used greatly in computer experiments most recently by [34].

5.1.5 Sobol' Sequences and low discrepency

Another class of space filling designs are the ones that are employed in numerical integration. One of such designs is Sobol' Sequence designs [72]. These are sequential designs which are easy to generate. In terms of even spread Latin Hypercube designs always surpasses Sobol' Sequences. The Sobol' designs on the other hand exhibit a great variety of distances between pairs of points (inter-point distances). Therefore, in situations where inter-point distances provide information to improve prediction errors, Sobol' sequences are preferred over Latin Hypercube Designs. Another useful property of Sobol' sequence is the computation of sequential designs, for example, a longer design can be obtained by adding more points to a short design in Sobol'. On the contrary, Latin Hypercube Design needs to be recomputed if more points to be added. However the properties of designs for numerical integration are derived for large numbers of observations hence their application for computer experiments with small

number of observations are not very clear. In order to get better insights of the design features that are important for parameter estimations in computer experiments, analytical methods are very difficult to apply. Consequently, substantial amount of numerical studies are useful for understanding the design performance for different models. Sobol' sequences are generated by binary digits. Use of other bases lead to generalizations such as digital nets. These sequences are studied because they are believed to have low discrepancy values see [51] and [32].

5.2 Prediction capabilities of models and designs

In this section, a comparative study is carried out where different design strategies introduced above in Subsections (5.1.3, 5.1.4, 5.1.5) are employed to the models explicated in Chapter(2) namely Sacks model in Section (2.2), Neural Networks in Section (2.6), and Linear regression. The choice of sampling design to select design points depends primarily on the objective of the experiment and other phenomena discussed in the paragraphs above. This study exhibits a general behaviour of the models performance under different design strategies. The aim is to evaluate which design is appropriate for which model. In other words models' prediction capabilities are evaluated against different designs. Recall the notations where d-dimensional vector of input variables is $\mathbf{x} = (x_j | j = 1, 2 \dots d) \in \mathbb{R}^d$. Let χ be the experimental region scaled to the unit hypercube $[0,1]^d$ and consider a design $\mathscr{D}$ be the set of n points taken from χ such that each point $x_i \in [0,1]^d$. The 2-Dimension and 3-Dimension input data is considered for this study.

The simulator $y(\mathbf{x})$ to obtain the response vector is the function from [88] and given as:

$$y(x) = \frac{10^d}{2}(2\pi^{\frac{-d}{2}} e^{-0.5||10(x-1/3)||_2^2} + 2\pi^{\frac{-d}{2}} e^{-0.5||10(x-2/3)||_2^2}.$$

For the purpose of emulation/prediction, the function is evaluated on $x_i \in [0,1]$, for all $i = 1, \dots, d$ which makes it flexible to be used for any number of dimensions for the input variable. Ten design points are generated for each of three sampling designs: Uniform, Sobol' and Latin Hypercube. Five of the design points are chosen as training data whereas the rest of the five design points serve as test data. Each of the three models are evaluated for each of the three designs are employed. A detailed account of box plots for the mean square prediction error is presented in Table(5.1) both for 2-dimension and 3-dimension data. For 2-dimension inputs, the box plots for each design against all the three models are depicted in the Figure (5.1). The graphical results for 3-dimension study are given in Appendix (A.3).

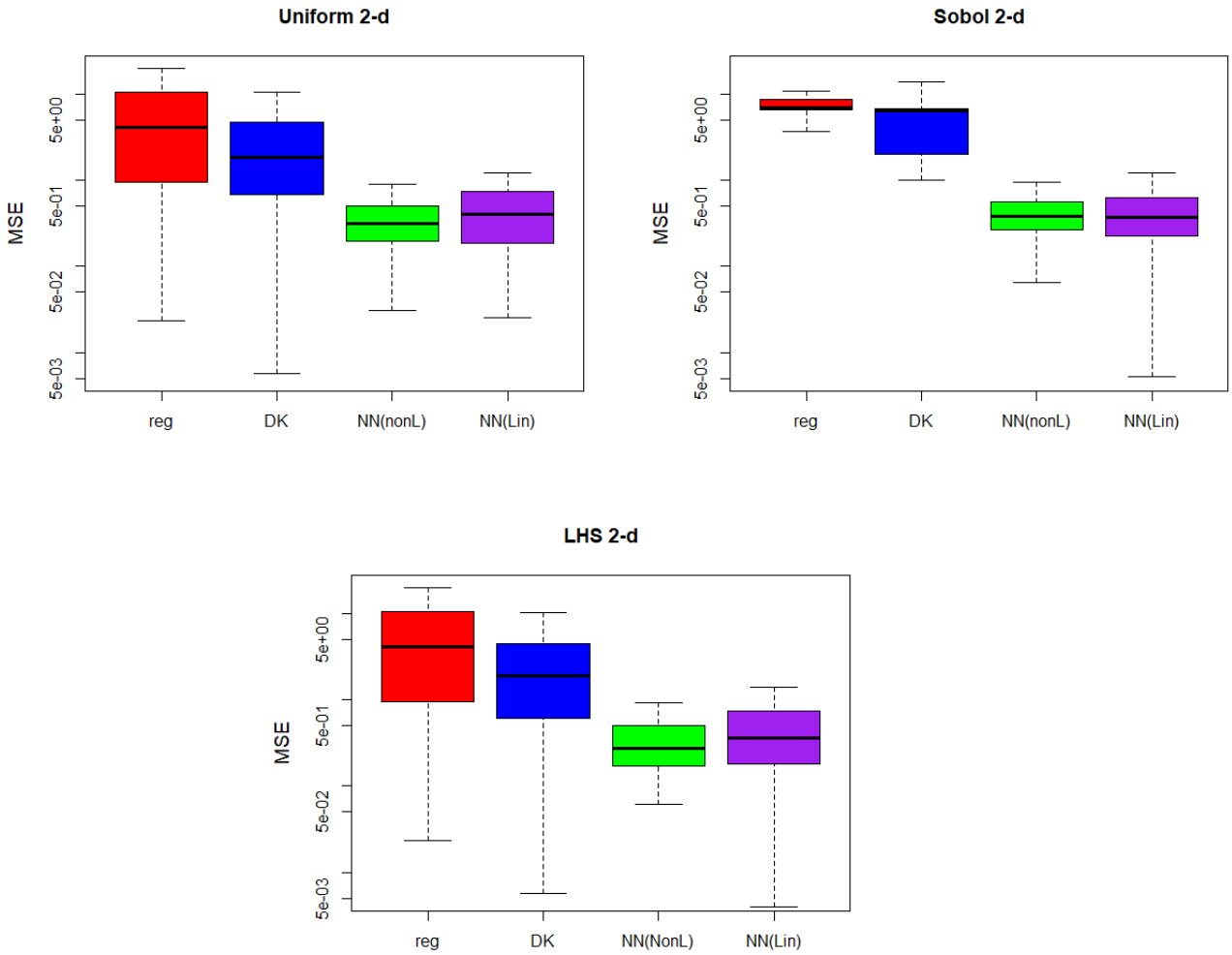


Figure 5.1: Box plots of mean square error of each design against different models (2-d)

models	designs		
	Uniform	Sobol"	Latin Hypercube
Regression	9.25	9.13	9.25
DACE	3.29	5.28	3.22
NN (non-Lin)	0.41	0.45	0.36
NN(Lin)	1.40	4.98	1.37

Table 5.1: MSE of all models against all designs chosen (2-d)

5.2.1 Results for the design plots

We observe from Table (5.1) that mean square error is highest with the use of all three designs for regression model. For Sacks model (DACE) Latin Hypercube design gives minimum mean square error followed by Uniform which gives mean square error close to that of LHS. The highest mean square error is obtained by the use of Sobol' in DACE model. The similar pattern can be seen for Neural Network when a non linear activation function is used. Among all the models, Neural Network provides minimum mean square error for all the designs employed.

5.3 Criterion-based Optimal Designs

Optimal designs are a class of experimental designs that are optimal with respect to some statistical criterion. The development of the concept of optimal designs dates back to 1916 and credited to [70] who introduced optimal designs for polynomial regressions. The idea was further extended by [35] who introduced the concept of probability measure on design space. An optimal design primarily requires specification of an appropriate statistical model and a suitable criterion. The choice of an optimal criterion relies on the objective of the experimental study. Most of the optimality criteria in literature for the optimal designs are developed based on some functions of the information matrix. Therefore, optimality is highly dependent on the model employed, hence it is vital to benchmark the design performance for different models. Cornell [17] comments that "A design that is optimal for a given model using one of the criteria is usually near-optimal for the same model with respect to other criteria". Some of the widely adopted optimality criteria are accounted in this section. As mentioned above almost all optimality criteria depend on the model employed, therefore, it is reasonable to define a general model prior to explain designs. Recall from Chapter (2) that for an $n \times 1$ response vector $\mathbf{y}$, an $n \times k$ design model matrix F and a vector of regression coefficients β of order $k \times 1$, the linear regression model in matrix notation can be written as $\mathbf{y} = F\beta + \epsilon$. For the regression model, the estimate of regression parameters β, variance of the estimator $\hat{\beta}$, predictor $\hat{y}(x)$ and variance of the prediction $V(\hat{y}(x))$ are discussed in Section (3.1). The design problem consist of selecting $\mathscr{D}$ from design space $\chi \subset \mathbb{R}^d$ such that the points $[x_i]_{i \in 1 \ldots n}$ satisfy some optimal criterion, giving rise to an optimal design. A brief account of some of the optimal criteria are delineated in this section. We use optimal criteria in Sections (5.3.1, 5.3.2, 5.3.3) as basis to explore designs for Smooth Ridge model in the later part of this chapter.

5.3.1 G-optimality

The first ever criterion based design approach was introduced by Smith [70]. The methodology was based on selecting a design that minimizes the maximum prediction variance

$$\min_{x_1,\ldots,x_n} \max_{x \in \mathcal{X}} Var(\hat{y}_x) \tag{5.1}$$

For every choice of design we compute maximum prediction variance for unknown points. Among all designs the G-optimal is the one for which the maximum prediction variance is minimum. The same design was given the name G-optimality few decades later by Kiefer and Wolfowitz [35], who showed the equivalence of G- and D- optimality in their celebrated theorem that we survey in Section (5.4).

5.3.2 D-optimality

A criterion for design that focusses on the variance of the parameter estimates is proposed by Wald [83] which was later named D-optimality by [35]. For statistical models for example regression model, the Fisher Information matrix is the inverse of the variance-Covariance matrix of the estimator. D-optimality criterion is to maximize the determinant of Information matrix $|X^T X]$ or equivalently minimizing the variance $|X^T X|^{-1}$. By definition,

$$\min_{x_1,\ldots,x_n} |X^T X|^{-1}$$

D-optimality has attracted a lot of research interest, such as the sequential Wynn-Fedorov approach [86], Locally D-optimal designs for exponential regression [21] and notable work in the book [1].

5.3.3 A-optimality

The criterion seeks the design to minimize the trace of the inverse of the information matrix which results in minimizing the average variance of regression coefficients' estimates. Explicitly,

$$\min_{x_1,\ldots,x_n} trace(X^T X)^{-1}$$

5.3.4 C-Optimality

C-optimality is a special case of A-optimality, when the interest lies in estimating a linear function of the parameters of interest $c^T\theta$. Therefore, the aim is to minimize $var(c^T\hat{\theta})$ and the optimal design is the one such that

$$\min_{x_1,\ldots,x_n} trace(c^T(X^TX)^{-1}c)$$

5.3.5 E-optimality

A special case of C optimality is E-optimality where the objective is to minimize the maximum variance of $c^T\hat{\theta}$ for all c conditioned on $\|c\| = 1$. Hence, an E-optimal design is the one that

$$\min_{x_1,\ldots,x_n} \max_{\|c\|=1} c^T(X^TX)^{-1}c$$

which is equivalent to minimize the maximum eigen value

$$\min_{x_1,\ldots,x_n} \lambda_{max}(X^TX)^{-1}$$

A detailed account of optimal designs within the bayesian context and the bayes counterparts of above mentioned designs can be found in [12]. D-optimal and G-optimal designs received the most attention in the literature one of the reasons attributed to the General Equivalence Theorem provided by [35] which established the equivalence of G and D optimality under certain conditions. A brief overview of General Equivalence Theorem is given in Section (5.4).

5.4 General Equivalence Theorem

We provide a gentle introduction of General Equivalence Theorem because we believe that there is a scope of developing it for the Smooth Ridge model. It is important to mention here that the notations used in this section is different from the rest of the material in this thesis. Before stating the General Equivalence Theorem it is necessary to distinguish between exact theory and approximate theory. For a problem of maximizing function that involves integers, calculus rules cannot be applied. A standard approach is to extend the function definition to real numbers and find the maximum using calculus techniques. It can then be argued that the maximum over integers occurs at an adjacent integer. The analogous design problem differentiates the exact (integers in this example) from the approximate theory (real numbers).

The following property holds for all the design criteria described above

$$\varphi(aX^TX) = \text{constant} \times \varphi(X^TX)$$

which implies that a design that maximizes $\varphi(aX^TX)$ also maximizes $\varphi(X^TX)$. For an n-point design with n_i observations at x_i such that $\sum n_i = n$, let ξ be the probability measure on the design space χ defined such that

$$\xi(x_i) = \begin{cases} 0 & \text{if there are no observations at } x_i \\ \frac{n_i}{n} & \text{if there are } n_i \text{ observations at } x_i \end{cases}$$

In context of the measure ξ, an exact design on χ refers to discrete n-point design such that ξ takes on values as multiple of $\frac{1}{n}$. The idea can be extended to an approximate design so that the design measure ξ satisfies

$$\int_{\chi} \xi(dx) = 1$$

Let $\mathcal{M}(\xi) = \frac{1}{n}X^TX$ be the moment matrix for an exact design. Then for an approximate design, (ij) element of matrix $\mathcal{M}(\xi)$ can be expressed as

$$m_{ij}(\xi) = \int_{\chi} x_i x_j \xi(dx)$$

Similarly, a normalized generalization of $V(\hat{y}_x)$ is

$$d(x, \xi) = x(\mathcal{M}(\xi))^{-1}x^T$$

which for the exact design becomes

$$d(x, \xi) = nx(X^TX)x^T$$

Following the notations and description above for exact and approximate designs, D-optimality and G-optimality can be re-defined as

- The design ξ^* is D-optimal if and only if $\mathcal{M}(\xi^*)$ is nonsingular and

$$\max_{\xi} |\mathcal{M}(\xi)| = |\mathcal{M}(\xi^*)| \tag{5.2}$$

- The design ξ^* is G-optimal if and only if

$$\min_{\xi} \max_{x \in \mathcal{X}} d(x, \xi) = \max_{x \in \mathcal{X}} d(x, \xi^*) \tag{5.3}$$

Note that D-optimality is essentially a parameter estimation criterion whereas G-optimality is a response estimation criterion.

Theorem 5.1. *The Equivalence Theorem states that D- and G- design criteria are identical when the design is expressed as a measure on $\mathcal{X}$. It can be shown that a sufficient condition for the design ξ^* to satisfy the G-optimality criterion is*

$$\max_{x \in \mathcal{X}} d(x, \xi^*) = k$$

where k is the number of parameters in regression model. The equivalence of D- and G-optimality established in the General Equivalence Theorem [35] is given as: If φ is concave on the space of design information matrices, and differentiable at $\mathcal{M}(\xi^*)$, then the following are equivalent

1. *The measure ξ^* is φ-optimal*
2. *The Frechet derivative*

$$F_{\varphi}(\mathcal{M}(\xi^*), x^T x) \leq 0 \quad \text{for all} \quad x \in \mathcal{X}$$

3. *The following equality holds*

$$\max_{x \in \mathcal{X}} F_{\varphi}(\mathcal{M}(\xi^*), x^T x) = \min_{\xi} \max_{x \in \mathcal{X}} F_{\varphi}(\mathcal{M}(\xi), x^T x)$$

The sufficient condition for G-optimality can be used to verify whether or not a specific design is D-optimal.

5.5 D- and A-optimal designs for Smooth Ridge Model

In this section, criterion based designs are explored for Smooth Ridge model introduced in Chapter (3) within the classical framework of optimal designs. First, the expressions for D-optimality and A-optimality are developed for Smooth Ridge model. Then, a simple numerical study is done to find the optimal designs followed by the analytical results. One of the reasons

for choosing the D-optimality and A-optimality criteria is that it does not depend on the response $y(x)$ hence independent of the function generating the response. Nevertheless, the criteria depend on the model and the nature of the estimator thereof. Re-visit the Smooth Ridge estimator $\tilde{\beta}_\lambda$ given in Equation (3.11) and the variance of the estimator in Equation (3.13).

$$\tilde{\beta}_\lambda = (F^T F + \lambda K)^{-1} F^T Y, \quad Var(\tilde{\beta}_\lambda) = A\sigma^2 A^T \quad \text{where} \quad A = (F^T F + \lambda K)^{-1} F^T.$$

The D-optimality criterion for Smooth Ridge model is the maximization of the determinant of $(AA^T)^{-1}$ or equivalently minimizing the determinant of (AA^T) i.e. minimizing the variance of $\tilde{\beta}_\lambda$. Following the notations from Section (5.4) and Section (2.1) and defining $\mathcal{M} = AA^T$, we can now define D-optimal design for Smooth Ridge model.

Definition 5.1. *A design $\mathscr{D}^*$ is D-optimal if and only if $\mathcal{M}(\mathscr{D})$ is nonsingular and*

$$\min_{\mathscr{D}} |\mathcal{M}(\mathscr{D})| = |\mathcal{M}(\mathscr{D}^*)| \tag{5.4}$$

5.5.1 D-optimal design with two design points

To begin with, a simple example is considered with two design points x_1 and x_2 and following analytical results are established.

Theorem 5.2. *Consider Smooth Ridge model with k Legendre terms and design $\mathscr{D} \in [-1, 1]$ of size n such that $k > n$. The design $\mathscr{D}$ is constructed with two design points x_1 and x_2 such that for a design of size n each design point is replicated r_1 and r_2 times respectively so that $n = r_1 + r_2$. Define the D-optimality as the product of the non-zero eigen values of the matrix $\mathcal{M} = AA^T$, then the D-optimal design is given by ± 1.*

Proof. The D-optimality criterion is

$$\frac{1}{(r_1 r_2)(x_1 - x_2)^2}$$

which is minimized when the distance between the points is maximal. Let $\Delta = x_1 - x_2$, then it is straightforward to conclude that the minimum of $1/\Delta^2$ can be obtained at ± 1, at which $\Delta = 2$ and the D-optimal criterion is $\frac{1}{4r_1 r_2}$ □

Corollary 5.2.1. *For a symmetrical design with coordinates $\{x_1, -x_1\}$ and replicates r_1 and r_2, the D-optimality criterion simplifies to $\frac{1}{4r_1 r_2 x_1^2}$. The minimum is achieved at ± 1 giving the criterion to be $\frac{1}{4r_1 r_2}$.*

The following are some of the properties deduced from the analytical results:

1. The D-optimal design is independent of the choice of model terms k.
2. The D-optimal design is independent of the the choice of λ.
3. The D-optimal design favours the linear model which is not a desired property beacause the Smooth Ridge Model attempts to capture the non-linear relationship.

5.5.2 Numerical Results for D-optimality: Two design points with replications

The analytical results in Theorem (5.2) are investigated with the help of following numerical study. A design region χ over the unit hypercube $[-1, 1]$ is explored by selecting two design points x_1, x_2, with replications $r_1 = 3$ and $r_2 = 2$ respectively resulting into a design of size $n = 5$. The simulations are designed to chose 90% designs of the form $\mathscr{D} = \{x_1, x_1, x_1, x_2, x_2\}$ and 10% of the symmetrical design $\mathscr{D} = \{-x_1, -x_1, -x_1, x_1, x_1\}$ to evaluate the results explicit in Theorem(5.2) and Corollary(5.2.1) respectively. The simulations are run for $50,000$ designs over $[-1, 1]$ against each of the 3 values of $\lambda = \{0.001, 0.1, 5\}$. The D-optimality criterion for Smooth Ridge model given in Equation(5.4) is evaluated for each design against each value of λ. It is cruicial to notice here that $\mathcal{M} = AA^T$ is less than full rank because of the condition $n < k$, therefore D-optimality is computed as the product of non-zero eigen values of the matrix $\mathcal{M}$.

The graphical representation of the designs and the resulting D-optimality measures are given in Figure (5.2). For every value of λ, one of the $50,000$ designs that meets the minimum D-optimality criterion is chosen and the value of the D-optimal measure is recorded. This results into 3 optimal designs against each of the 3 different values of the regularization parameter λ and are furnished in the Table (5.2). In each plot we can see the footprint of $50,000$ designs against the D-optimality criterion for a given value of λ.

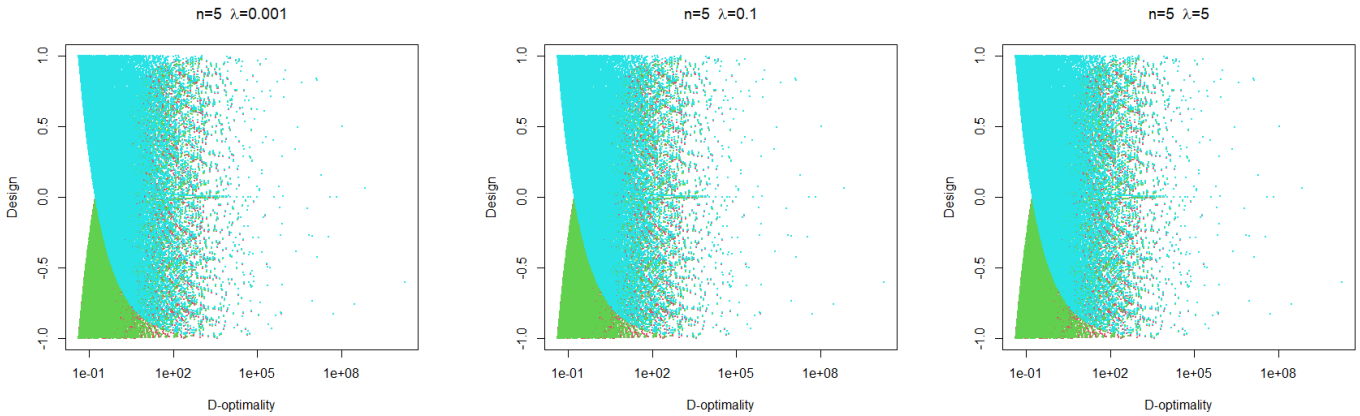


Figure 5.2: Designs against D-optimality criterion $n = 5$

	D-optimal designs $n = 5$					
λ	x_1	x_2	x_3	x_4	x_5	D-opt
0.001	-0.9999218	-0.9999218	-0.9999218	0.9999218	0.9999218	0. 04167319
0.1	-0.9999218	-0.9999218	-0.9999218	0.9999218	0.9999218	0.04167319
5	-0.9999218	-0.9999218	-0.9999218	0.9999218	0.9999218	0.04167319

Table 5.2: D-optimal designs and min D-optimality $n = 5$

It can be observed from Table (5.2) that for every value of λ, the optimal design is ± 1 for each value of λ and the minimum measure of D-optimality is constant approximately equal to 0.042. The results are consistent with Theorem (5.2) and Corollary (5.2.1).

5.5.3 A-optimality for two design points

The A-optimality criterion for Smooth Ridge model is the minimization of the trace of determinant of (AA^T). A-optimality can be defined as

Definition 5.2. *A design $\mathscr{D}^*$ is A-optimal if and only if $\mathcal{M}(\mathscr{D})$ is nonsingular and*

$$\min_{\mathscr{D}} |\mathcal{M}(\mathscr{D})| = tr\,|\mathcal{M}(\mathscr{D}^*)| \tag{5.5}$$

Analytical results for A-optimality are established similar to that of D-optimality when two design points x_1 and x_2 are replicated to get a sample of size n which are explicated in the following theorem.

Theorem 5.3. *Consider Smooth Ridge model with k Legendre terms and design* $\mathscr{D} \in [-1, 1]$ *of size n such that* $k > n$*. The design* $\mathscr{D}$ *is constructed with two design points* x_1 *and* x_2*, where each design point is replicated* r_1 *and* r_2 *times respectively giving rise to a design of size n. Define the A-optimality as the sum of eigen values of the matrix* $\mathcal{M} = AA^T$*, then the optimal design is achieved at* ± 1*.*

Proof. The A-optimality criterion is

$$\frac{r_1 x_1^2 + r_2 x_2^2 + n}{r_1 r_2 (x_1 - x_2)^2}$$

which is minimized when the distance between the points in the denominator is maximal. Let $\Delta = x_1 - x_2$ then for given r_1 and r_2 the minimum of $\frac{r_1 x_1^2 + r_2 x_2^2 + n}{r_1 r_2 \Delta^2}$ is obtained when $x_1 = -1$, $x_2 = 1$ at which $\Delta = 2$ and the criterion is $\frac{r_1 + r_2 + n}{4 r_1 r_2}$. Note that the choice of $x_1 = 1$, $x_2 = -1$ is equivalent. Also note that $x_1 = x_2$ leads to a single design point which is not considered here. □

Corollary 5.3.1. *For a symmetrical design* $\mathscr{D} = \{-x_1, x_1\}$ *the A-optimality criterion becomes* $\frac{n(x_1^2+1)}{4 r_1 r_2 x_1^2}$ *which is minimized at* ± 1 *and the minimum value of the criterion is reached at* $\frac{n}{2 r_1 r_2}$*.*

5.5.4 Numerical Results: A-optimality with two design points and replication

The numerical study in Section (5.5.2) is revisited to search A-optimal designs. Note that the search of optimal design in this section is performed over the same candidate set of designs as in Section (5.5.2). Each of the 50000 designs are evaluated against every value of $\lambda = \{0.001, 0.1, 5\}$ where 90% of the designs comprise two design points $\{x_1, x_2\}$ and 10% of the designs are composed of the design points $\{-x_1, x_1\}$ that are replicated $r_1 = 3$ and $r_2 = 2$ respectively. The plots of the designs against the A-optimality criterion and tabular representation of optimal designs along with the minimum measure of the criterion itself are displayed in Figure (5.3) and Table (5.3) respectively.

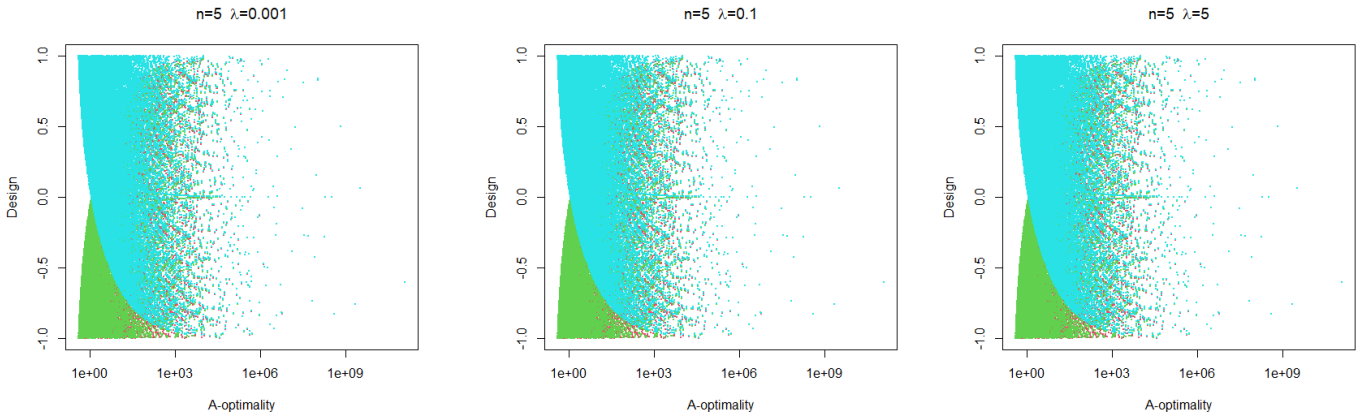


Figure 5.3: Designs against A-optimality criterion $n = 5$

	A-optimal designs $n = 5$					
λ	x_1	x_2	x_3	x_4	x_5	D-opt
0.001	–0.9999218	-0.9999218	-0.9999218	0.9999218	0.9999218	0.4166993
0.1	-0.9999218	-0.9999218	-0.9999218	0.9999218	0.9999218	0.4166993
5	-0.9999218	-0.9999218	-0.9999218	0.9999218	0.9999218	0.4166993

Table 5.3: D-optimal designs and min D-optimality $n = 5$

It can be readily observed from Table (5.3) that optimal designs are independent of the choice of λ and are close to ± 1 which verify the results in Theorem (5.3) and corollary(5.3.1). The Tables(5.2,5.3) provide an intution that A-optimal designs coincide with that of D-optimal designs for the given choice of design points. In addition minimum A-optimality measure is equal to 0.42 for all values of λ.

5.6 D- and A-optimal designs with distinct design points for Ridge regression: SVD approach

The numerical results followed by the analytical conclusions in Section (5.5) are developed for two design points with replications. It is observed that two design points lead to a linear model which is not a desired property for Smooth Ridge model for the reason that the model is developed in an attempt to capture non-linear relationship. Also, it is not useful in practical applications where we have more design points. As a result, it is inevitable to explore other sampling options with more than two distinct design points. The analytical results are very

cumbersome to develop for the optimal design of Smooth Ridge model for any choice of design points. This is attributed to the complex nature of the Smooth Ridge model. Therefore, in order to build an intuition of the optimal designs of Smooth Ridge model, we first attempt to develop some results of optimal designs for ridge regression model when $K = I$ in Smooth Ridge model. We aim to obtain this objective with the help of singular value decomposition of the design model matrix F. Let the singular value decomposition of $(n \times k)$-dimensional design model matrix F be

$$F = UDV^T$$

where, U is $(n \times n)$-dimensional matrix with columns of left singular vectors, D is $(n \times n)$ diagonal matrix with singular values and V is $(k \times n)$ dimensional matrix with columns of right singular vectors. U and V are orthogonal such that $UU^T = I = VV^T$. The ridge estimator can be expressed in terms of singular value decomposition of F.

$$\hat{\beta}_\lambda = (F^TF + \lambda I)^{-1}F^TY = V(D^2 + \lambda I)^{-1}DU^TY \tag{5.6}$$

It is straightforward to extend the singular value decomposition and find the variance of ridge estimator.

$$V(\hat{\beta}_\lambda) = \sigma^2\left((F^TF + \lambda I)^{-1}F^TF(F^TF + \lambda I)^{-1}\right) = \sigma^2\left(V(D^2 + \lambda I)^{-1}D^2(D^2 + \lambda I)^{-1}V^T\right)$$

If we ignore σ^2 and simplify the variance of ridge estimator, we have

$$V(\hat{\beta}_\lambda) = \sum_{i=1}^{p} \frac{d_i^2}{(d_i^2 + \lambda)^2} v_i v_i^T, \tag{5.7}$$

which is the result obtained after [30].

For a given λ and a design model matrix F, we inspect the singular values d_i for high-dimensionality. When $n < k$, $d_i = 0$ for $j = n+1, \ldots, k$ which implies that $V(D^2 + \lambda I)^{-1}D^2(D^2 + \lambda I)^{-1}V^T = 0$ and hence the variance is determined by the first n columns of V. This fact provides a motivation to find an analytical result for D-optimality and A-optimality. The eigen values of the variance of $\hat{\beta}_\lambda$ in Equation (5.7) are computed by Karhunen-Loeve decomposition and can be written as:

$$(D^2 + \lambda I)^{-1}D^2(D^2 + \lambda I)^{-1} = diag\left(\frac{d_1^2}{(d_1 + \lambda)^2}, \ldots, \frac{d_n^2}{(d_n + \lambda)^2}\right). \tag{5.8}$$

Using equation (5.8) $\mathcal{M}$ can now be written as $\mathcal{M} = V(D^2 + \lambda I)^{-1}D^2(D^2 + \lambda I)^{-1}V^T$ and $|\mathcal{M}|$

becomes the product of the eigen values of $\mathcal{M}$ given as $\prod_{i=1}^{n}\left(\frac{d_i^2}{(d_i^2+\lambda)^2}\right)$. It is vital to mention here that for rank deficient matrix AA^T, $|\mathcal{M}|$ can be written as the product of non-zero eigen values of $\mathcal{M}$ hence, we name the optimal design as pseudo D-optimal design. However, for the sake of simplicity we refer the pseudo D-optimal design as D-optimal design for the rest of the chapter. Therefore the D-optimal design for singular value decomposition of ridge regression model is defined as

Definition 5.3. *A design $\mathscr{D}^*$ is D-optimal if*

$$\min_{\mathscr{D}} |\mathcal{M}(\mathscr{D})| = \min_{\mathscr{D}} \left(\prod_{i=1}^{n} \left(\frac{d_i^2}{(d_i^2+\lambda)^2} \right) \right) = |\mathcal{M}(\mathscr{D}^*)| \tag{5.9}$$

Therefore an optimal design for ridge regression is the one that minimizes the product of non-zero singular values of $\mathcal{M}$.

Pseudo A-optimal design or simply A-optimal design for singular value decomposition of ridge regression model can be defined in a similar way.

Definition 5.4. *A design $\mathscr{D}^*$ is A-optimal if it minimizes the trace $|\mathcal{M}(\mathscr{D})|$ which is equivalent to minimize the sum of eigen values of $\mathcal{M}$ given by $\sum_{i=1}^{n}\left(\frac{d_i^2}{(d_i^2+\lambda)^2}\right)$.*

$$\min_{\mathscr{D}} |\mathcal{M}(\mathscr{D})| = min_{\mathscr{D}} \left(\sum_{i=1}^{n} \left(\frac{d_i^2}{(d_i^2+\lambda)^2} \right) \right) = tr\,|\mathcal{M}(\mathscr{D}^*)| \tag{5.10}$$

The D and A optimality criteria for ridge regression reveal some important properties.

1. For a fixed design the product $\prod_{i=1}^{n}\left(\frac{d_i^2}{(d_i^2+\lambda)^2}\right)$ is minimized for large values of λ. This implies that no bounds can be specified for λ which is strictly monotone in a way that for $\lambda_i < \lambda_j$, $|\mathcal{M}(\lambda_i)| > |\mathcal{M}(\lambda_j)|$.

2. For $\lambda \to \infty$, $|\mathcal{M}(\mathscr{D})| \to 0$ therefore the product is asymptotic i.e. the minimum value of objective criterion is acheived when $\lambda \to \infty$.

3. For a given λ the product $\prod_{i=1}^{n}\left(\frac{1}{(d_i+\frac{\lambda}{d_i})^2}\right)$ is minimum when the denominator term $(d_i+\frac{\lambda}{d_i^2})$ is maximum. Therefore, for a given λ the design with large singular values d_i will be the D-optimal one.

Similar properties hold for A-optimality criterion.

5.7 SVD Optimal Designs for Smooth ridge model

The results developed for D and A- optimal designs for ridge model provides an intuition and framework to find optimal designs for Smooth ridge model. The analytical results are difficult to obtain because of the matrix K. We refer to singular value decomposition of design model matrix F defined in Equation 5.6. The smooth ridge parameter $\tilde{\beta}_\lambda$ and variance of smooth ridge parameter is given as

$$\tilde{\beta}_\lambda = (F^T F + \lambda K)^{-1} F^T Y = (V D^2 V^T + \lambda K)^{-1} V D U^T Y \tag{5.11}$$

The variance of smooth ridge estimator after singular value decomposition can be presented as

$$\frac{V(\hat{\beta}_\lambda)}{\sigma^2} = (F^T F + \lambda K)^{-1} F^T F (F^T F + \lambda K)^{-1} = (V D^2 V^T + \lambda K)^{-1} V D^2 V^T (V D^2 V^T + \lambda K)^{-1} \tag{5.12}$$

It is observed that Equation(5.12) cannot be simplified further. Hence, analytical form of the product and sum of the eigen values of the $V(\hat{\beta}_\lambda)$ is not possible to achieve. In other words D and A optimal criteria are hard to derive analytically in the form of singular values of design model matrix F. However, we believe that for a given matrix K in smooth ridge model, the D and A-optimality criteria bear the same properties as that of ridge regression model explained in Section (5.6).

5.7.1 Numerical Results: D- and A- optimality for Smooth Ridge model

We adopt numerical approach to find D and A- optimal designs in a similar fashion i.e. the designs that minimize $|\mathcal{M}(\mathscr{D})|$ and $tr\,|\mathcal{M}(\mathscr{D})|$ respectively where $\mathcal{M}(\mathscr{D}) = \big(V D^2 V^T + \lambda K)^{-1} V D^2 V^T (V D^2 V^T$
In order to run the simulations two sampling designs namely symmetric and uniform are chosen and evaluated to find D-optimal and A-optimal designs. Symmetric and uniform designs of size $n = \{5, 6, 7, 8, 10\}$ are selected from design region χ over the hypercube $[-1, 1]$ with model terms $k = \{6, 7, 8, 9, 11\}$ respectively. Therefore a symmetric and uniform design of size $n = 5$ is $\mathscr{D} = \{-x_1, -x_2, 0, x_2, x_1\}$ and $\mathscr{D} = \{x_1, x_2, x_3, x_4, x_5\}$ respectively. The simulations are designed so that 5% of the designs are symmetric and the remaining 95% of designs, are in general position. For each sample size n, $50,000$ designs over $[-1, 1]$ are obtained for each of the 3 values of $\lambda = \{0.001, 0.1, 5\}$. Among $50,000$ designs the D-optimal and A-optimal designs are

the ones which satisfy the respective criteria. This results into three D and A-optimal designs against each of the three values of the regularization parameter λ. The graphical description of all the $50,000$ designs against the D-optimality criteria for the given λ is displayed for $n = 5$ and $n = 6$ in Figures (5.4, 5.5). The rest of the plots are given in Appendix (A.6). The design that satisfies the D-optimality criteria for each of three values of λ are given in Tables (5.4, 5.5) respectively.

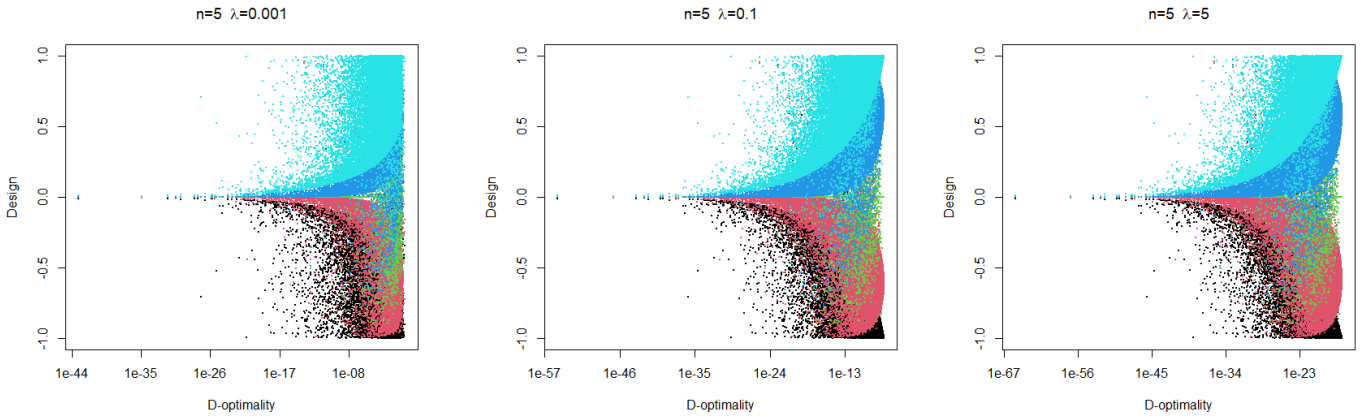


Figure 5.4: Designs against D-optimality criterion $n = 5$

	D-optimal designs $n = 5$					
λ	x_1	x_2	x_3	x_4	x_5	D-opt ($\times 10^{44}$)
0.001	-0.0131647	-0.000000347	0	0.000000347	0.01316467	8.292517
0.1	-0.0131647	-0.000000347	0	0.000000347	0.01316467	7.88×10^{-12}
5	-0.0131647	-0.000000347	0	0.000000347	0.01316467	4.797×10^{-22}

Table 5.4: D-optimal designs and min-optimality $n = 5$

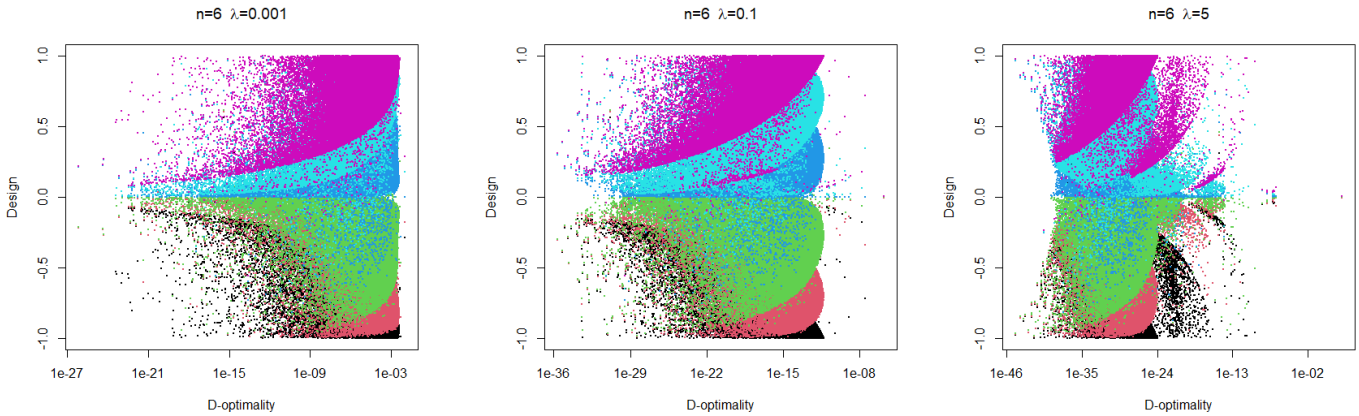


Figure 5.5: Designs against D-optimality criterion $n = 6$

	D-optimal designs $n = 6$						
λ	x_1	x_2	x_3	x_4	x_5	x_6	D-opt ($\times\ 10^{27}$)
0.001	-0.2188916	-0.2180156	-0.213669648	0.213669648	0.2180156	0.2188916	6.815
0.1	-0.3083092	-0.3061263	-0.297516100	0.297516100	0.3061263	0.3083092	2.3166×10^{-9}
5	-0.9774893	-0.9703251	-0.967225006	0.967225006	0.9703251	0.9774893	2.033×10^{-18}

Table 5.5: D-optimal designs and min-optimality $n = 6$

5.7.2 Results of D-optimal design

In Figures (5.4, 5.5) all $50,000$ designs are plotted against the D-optimality measures for each of the three values of $\lambda = \{0.001, 0.1, 5\}$. For $n = 5$, 0 is included as an inherent design point in the symmetric design and it is evident from Figure(5.4) that D-optimal designs are lying close to 0 and the optimal designs remain same for all values of λ. For $n = 6$ in Figure (5.5) 0 is not inherent to the symmetric design and the optimal design points lie on top of each other. Also,the optimal designs move away from 0 towards the design points -1 and 1 with increasing λ. It is noticeable from Tables (5.4, 5.5) that D-optimality decreases as values of λ increases which is one of the conclusion deduced from optimality results obtained for Ridge regression in Section (5.6).

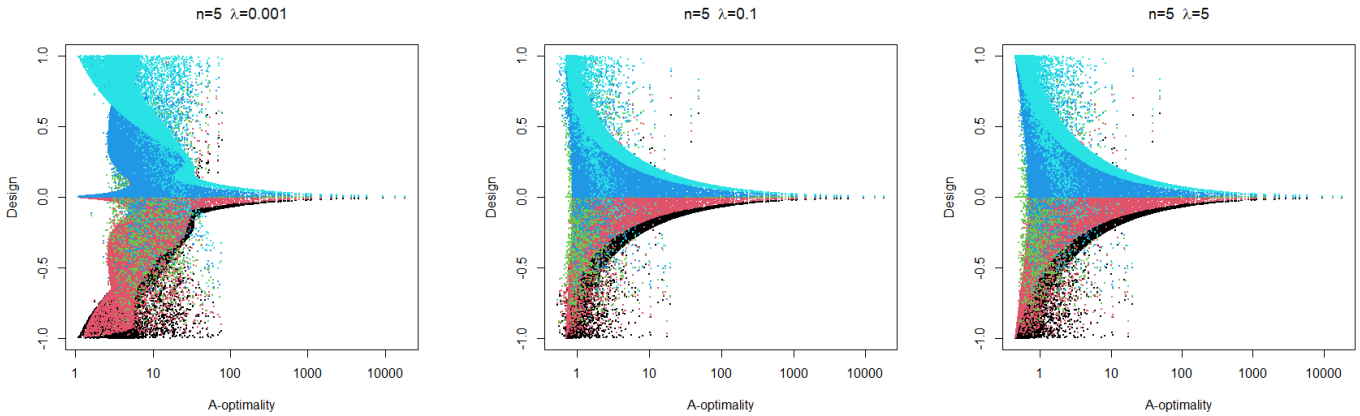


Figure 5.6: Designs against A-optimality criterion $n = 5$

	A-optimal designs $n = 5$					
λ	x_1	x_2	x_3	x_4	x_5	A-opt
0.001	-0.99721622	-0.00007477	0	0.00007477	0.99721622	1.101637
0.1	-0.76376333	-0.7559421	-0.7331361	0.8205765	0.94029286	0.5409493
5	-0.99768684	-0.9947400	0	-0.9947400	0.99768684	0.4524402

Table 5.6: A-optimal designs and min A-optimality $n = 5$

For A-optimality the designs for $n = 5$ and $n = 6$ are plotted in Figures (5.6, 5.7). The design that satisfies the A- optimality criteria for each of three values of λ are given in Tables (5.6, 5.7) respectively.

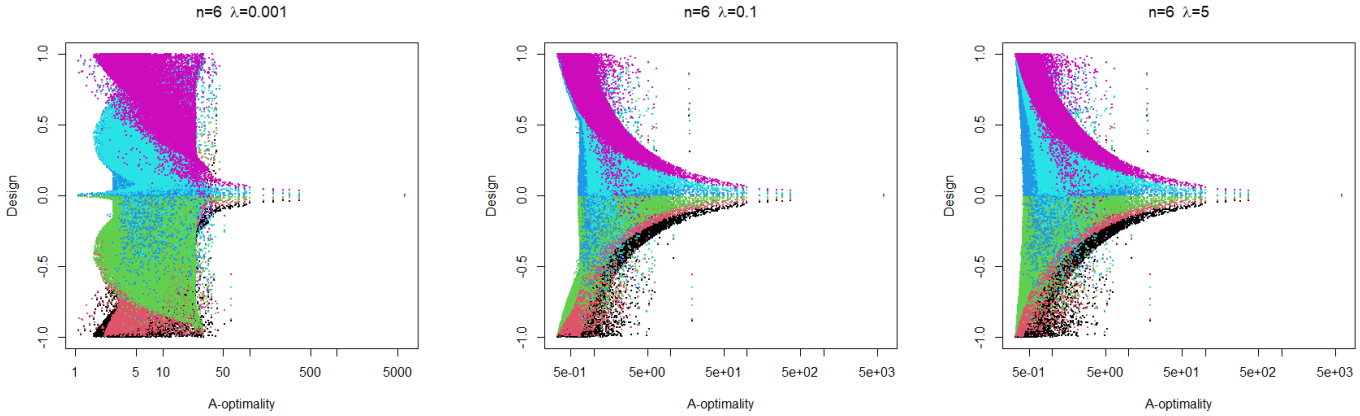


Figure 5.7: Designs against A-optimality criterion $n = 6$

	A-optimal designs $n = 6$						
λ	x_1	x_2	x_3	x_4	x_5	x_6	A-opt
0.001	-0.9567726	-0.9495472	-0.003632956	0. 003632956	0.9495472	0.9567726	1.085739
0.1	-0.9998098	-0.9932017	-0.984863805	0.984863805	0.9932017	0.9998098	0.3370789
5	0.9998098	-0.9932017	-0.984863805	0.984863805	0.9932017	0.9998098	0.3358132

Table 5.7: A-optimal designs and min A-optimality $n = 6$

5.7.3 Results of A-optimal design

Figures (5.6,5.7) depict all design plots against the respective A-optimal criterion for $n = 5$ and $n = 6$. The A-optimal design among $50,000$ designs for each value of $\lambda = \{0.001, 0.1, 5\}$ are presented in Tables (5.6,5.7) for $n = 5$ and $n = 6$ respectively. It is apparent that the optimal design is such that the two design points are chosen very close to 0 when 0 is inherent design point for $n = 5$ and the remaining two points are at the boundary of $[-1, 1]$ for $\lambda = 0.001$. The optimal design moves away from zero for large values of λ. From Table (5.7) it is obvious that for $n = 6$ the optimal design points lie very close to each other in pairs for example the first two design points lie close to -1 followed by the next two points close to 0 and the last two points are close to 1 for $\lambda = 0.001$ whereas half of the design points lie close to -1 and the rest of the design points are close to 1 for large λ. It is noticeable that D-optimality and A-optimality measures tend to be smaller with large values of λ for all n which is consistent with the result obtained for ridge regression.

5.8 Designs based on prediction variance: An exploratory analysis

In Chapter 4 we studied a detailed comparison of different models in the context of their prediction capabilities relative to that of Smooth ridge model. Therefore, it seems viable to search for the designs that minimize the prediction variance for Smooth ridge model $V(\tilde{y}_\lambda)$ in Equation (3.13). We aim to explore two of the optimal designs that address the prediction variance namely G-optimal design defined in Equation (5.1) and an I optimal design which is defined as,

Definition 5.5. *A design $\mathscr{D}$ is I-optimal if it minimizes the average prediction variance $\int V(\hat{y}(x))$ over the design region χ.*

In order to find I- and G-optimal designs for Smooth Ridge model we carry out an ex-

ploratory analysis with the help of simulation study in R. In order to run the simulations two sampling designs namely symmetric and uniform are selected from design region χ over the hypercube $[-1, 1]$. Designs of size $n = \{4, 5\}$ are chosen with model terms $k = \{5, 6\}$ respectively. Therefore a symmetric and uniform design is of the form $\mathscr{D} = \{-x_1, -x_2, 0, x_2, x_1\}$ and $\mathscr{D} = \{x_1, x_2, x_3, x_4, x_5\}$ for $n = 5$. Similarly, $\mathscr{D} = \{-x_1, -x_2, -x_3, x_3, x_2, x_1\}$ and $\mathscr{D} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ are the symmetric and uniform designs of size 6. The simulations are designed so that 95% of the designs are symmetric and the remaining 5% of designs, are in general position. For each sample size n, 5000 designs over $[-1, 1]$ are obtained for each of the 3 values of $\lambda = \{0.001, 0.1, 1\}$. Among 5000 designs the G- and I-optimal designs are the ones which satisfy the criteria given in Equation (5.1) and Def (5.5). This gives us three G- and I-optimal designs against each of the three values of the regularization parameter λ for a given design size. The graphical description of the designs against average prediction variance and a given λ is given in Figures (5.8, 5.9) for $n = 5$ and $n = 6$ respectively. Among 5000 designs, the I-optimal design for each value of λ can be found in Tables (5.8, 5.9).

Designs for average prediction variance and I-optimality $n = 5$

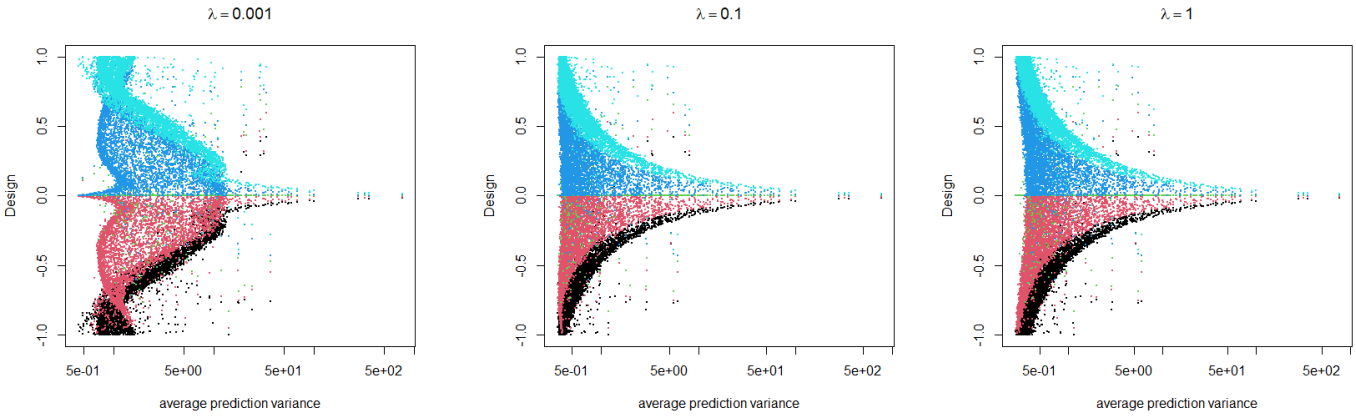


Figure 5.8: Designs for average prediction error n=5

	I-optimal designs $n = 5$					
λ	I-optimality	x_1	x_2	x_3	x_4	x_5
0.001	0.44044248	-0.9287	-0.0038	0	0.0038	0.9287
0.1	0.3530070	-0.8441	-0.7342	0	0.7342	0.844
1	0.2743626	-0.9765	-0.9642	0	0.9642	0.9765

Table 5.8: I-optimal designs and I-optimality $n = 5$

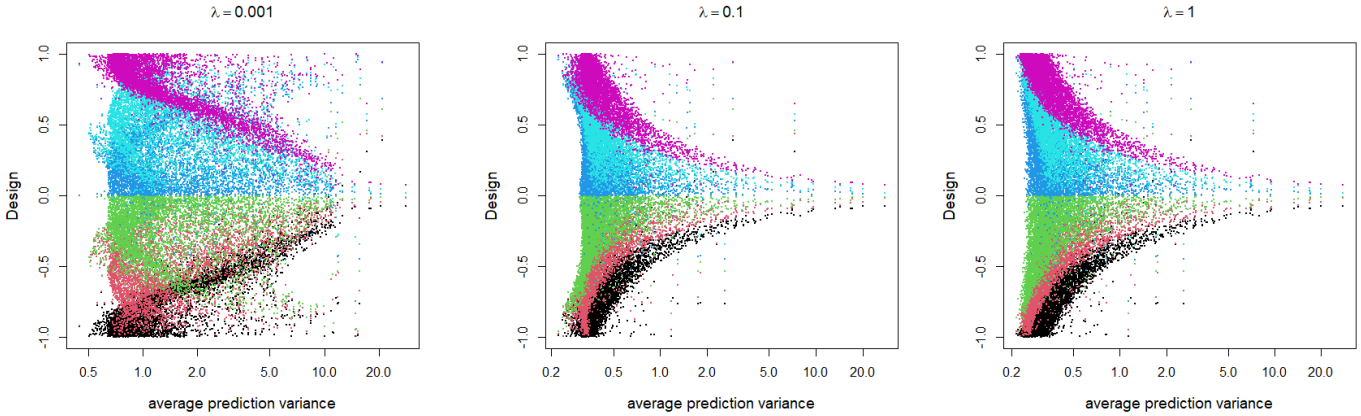


Figure 5.9: Designs for average prediction variance n=6

	I-optimal designs $n = 6$						
λ	I-optimality	x_1	x_2	x_3	x_4	x_5	
0.001	0.4495441	-0.9291	-0.9184	-0.0019	0.0019	0.9184	0.9291
0.1	0.223206	-0.9909	-0.9767	-0.9587	0.9587	0.9767	0.9909
1	0.2155392	-0.9766	-0.9643	-0.8621	0.8621	0.9643	0.9766

Table 5.9: I-optimal designs and I-optimality $n = 6$

The plots for 5000 designs are given in Figures (5.10, 5.11) for given values of λ and design size $n = \{5, 6\}$ respectively. The respective G-optimal designs along with the G-optimality measures for $n = 5$ and $n = 6$ are displayed in Tables (5.10, 5.11).

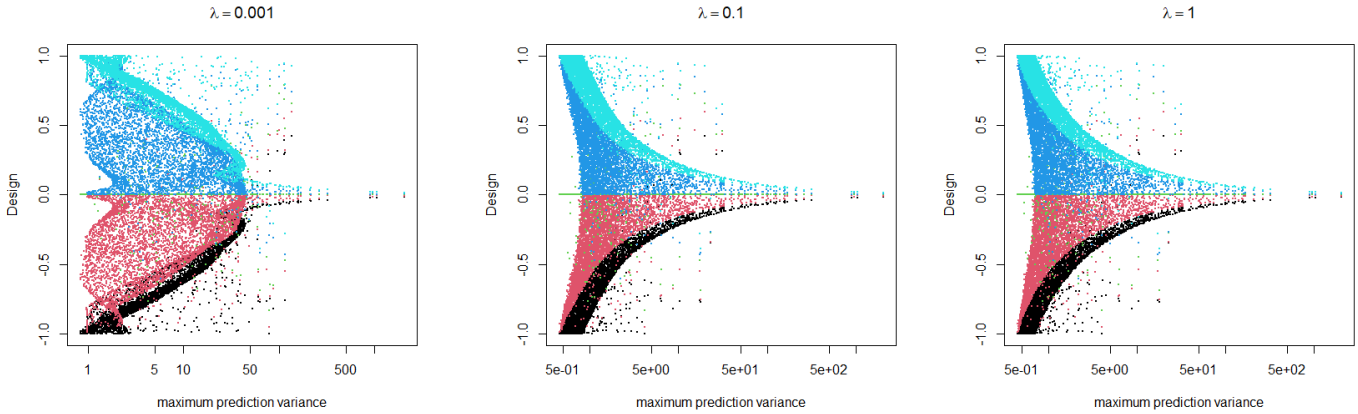


Figure 5.10: Designs for maximum prediction variance n=5

	G-optimal designs $n = 5$					
λ	G-optimality	x_1	x_2	x_3	x_4	x_5
0.001	0.8362186	-0.999	-0.6116	0	0.6116	0.999
0.1	0.4597038	-0.994	-0.9844	0	0.9844	0.994
1	0.4464979	-0.994	-0.9844	0	0.9844	0.994

Table 5.10: G-optimal designs and G-optimality $\mathbf{n = 5}$

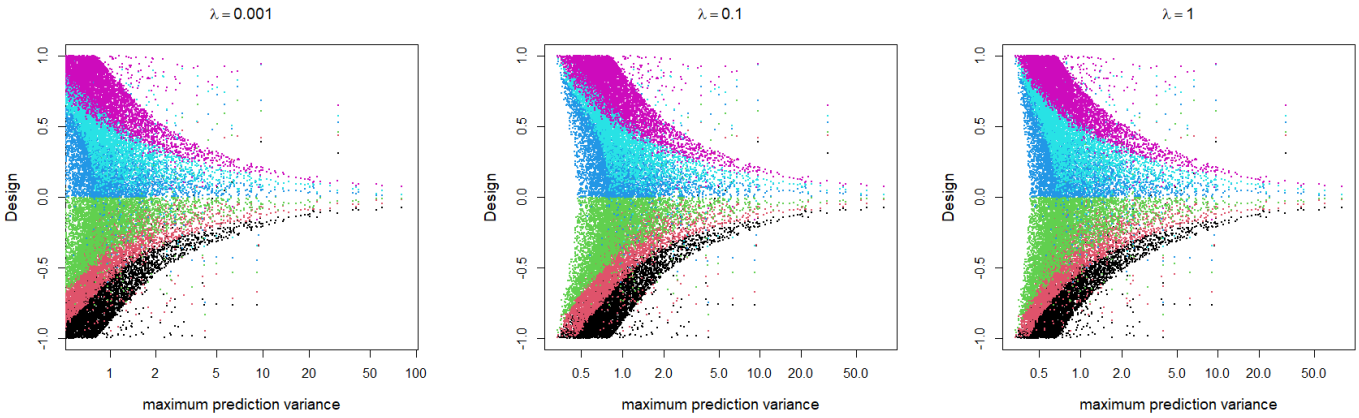


Figure 5.11: Designs for maximum prediction variance n=6

	G-optimal designs $n = 6$						
λ	G-optimality	x_1	x_2	x_3	x_4	x_5	x_6
0.001	0.6263141	-0.9291	-0.9184	-0.0019	0.0019	0.9184	0.9291
0.1	0.3363496	-0.9982	-0.9937	-0.9440	0.9440	0.9937	0.9982
1	0.3363428	-0.9982	-0.9937	-0.9440	0.9440	0.99379	0.9982

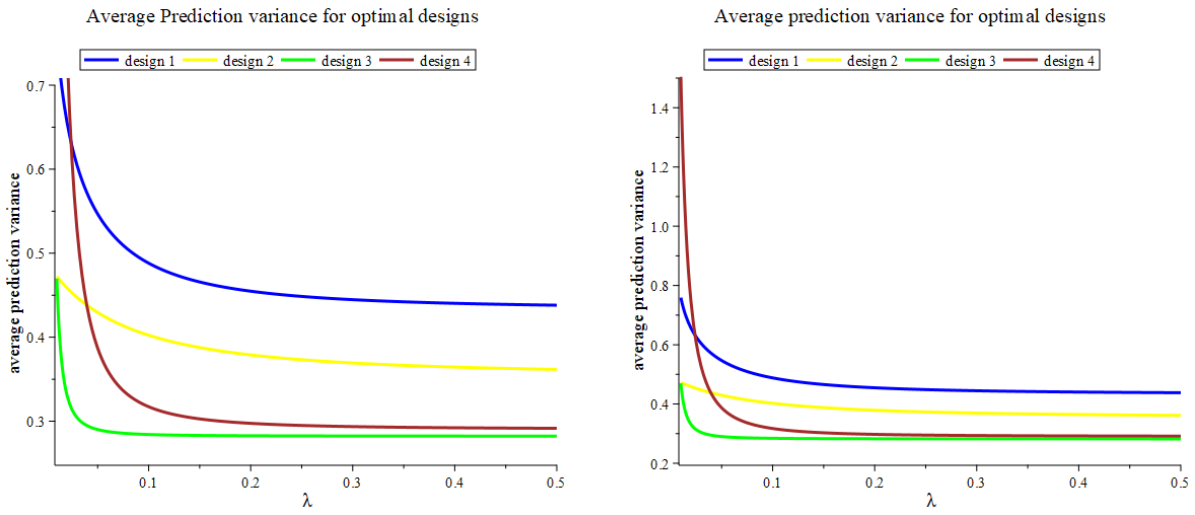
Table 5.11: G-optimal designs and G-optimality $n = 6$

5.8.1 Results of I- and G-optimal designs

We can see from Figures (5.8, 5.10, 5.9, 5.11) that the minimum of the average prediction variance and minimum of the maximum prediction variance is at the extreme left of the x-axis. Therefore the I-optimal design is the one that gives minimum prediction variance for 100 untried points over all the 5000 designs. Similarly, the G-optimal design is the one for which maximum variance is minimum over all 5000 designs. The I-optimal designs for three values of λ along with minimum prediction variance and the G-optimal designs along with minimum of the maximum prediction variance is given in Tables (5.8, 5.10, 5.9, 5.11). We observe a very interesting feature in G-optimal designs (Tables (5.10, 5.11)) that the optimal designs do not depend on λ when$\lambda \geq 0.1$ in the given example. Also, the I-optimality and G-optimality decreases with the increase in λ.

5.9 Analytical results for I-optimal design for a fixed design

In order to get some more insight into the I-optimal designs in terms of its dependence on λ, we develop the analytical form of the prediction variance of smooth ridge estimator given in Equation (3.13) with the use of `Maple`. It is important to mention that prediction variance is a function of unknown input x and λ. In order to find average prediction error, we integrate the prediction variance with respect to x over $[-1,\ 1]$ i.e. $\int_{-1}^{1} Var(\tilde{\beta}_\lambda)dx$ for each of the optimal designs in Tables (5.8, 5.9) which are labelled design2-design4 in the Figure (5.12). We add one more design in addition to the optimal designs which is $x = \{-1, -0.5, 0, 0.5, 1\}$ and $x = \{-1, -0.6, -0.2, 0.2, 0.6, 1.0\}$ for $n = 5$ and $n = 6$ respectively and is labelled as design1. We then plot the average prediction error for the optimal designs against different values of λ. The plots for $n = 5$ and $n = 6$ are exhibited in Figure (5.12).



(a) $n = 5$ (b) $n = 6$

Figure 5.12: Average prediction variance for optimal designs against λ

5.9.1 Maple results for I-optimal design

We can see from Figure (5.12) that for each of the four designs average prediction variance is asymptotic in λ. We observe that for $\lambda < 0.1$ the average prediction variance decreases with the increase in λ and the change is stable at $\lambda > 0.1$. Also, the average prediction variance is minimum for the designs $x = \{-0.8441, -0.7342, 0, 0.7342, 0.8441\}$ and $x = \{-0.9909, -0.9767, -0.9587, 0.9587, 0.9767, 0.9909\}$ for $n = 5$ and $n = 6$ respectively shown by green lines.

Chapter 6

Discussions and Recommendations

In this thesis we developed a methodology to build a smooth emulator for computer experiments. The framework of the model developed can be divided into two categories, a classical setting with the trend only and the Gaussian emulator. We investigated the predictive performance of the model in both frameworks. Explicitly, the estimator of the model is built on the foundations of a measure of smoothness, hence a cheaper alternative to exepensive emulators is proposed where Gaussian process is not employed. On the other hand, the standard Gaussian process is incorporated in the model to look into any improvement over existing Gaussian based emulators. In chapter 3 we devised analytical results to show that mean square error of the Smooth Ridge estimator improves over that of the standard regression estimator both for full rank and less than full rank design model matrix. We noticed that for less than full rank design model matrix, the mean square error of the model estimator is improved under certain conditions on the parameter λ. We verify our analytical results with the help of initial simulated studies for 1-D adn 2-D. In addition we chose a simulator that does not obey the Gaussian process in which case the Smooth Ridge model without Gaussian process performs the best among all the emulators. In order to explore the application of Smooth Ridge model to real life applications, we included two real life studies, one encompassing the transmission of COVID virus and other includes the simulated temperature data. In both the problems, our methodology outperformed the existing emulators.

In chapter 4 we carried out an extensive simulated study to compare the Smooth emulator with contemporary models. We first used Bratley function as a simulator to evaluate the performance of our proposed model for different choice of design points and for different choice of model terms. The comparisons are focussed on rank deficient design model matrix. We considered input variables from 2-dimensions to 20-dimensions. Since, the Smooth emulator

is constructed with Legendre polynomials and Bratley function is favourable to polynoimals therefore we searched for a simluator that is not supported by polynomials. For this reason we used Levy function as simulator and repeated all the comparisons. We concluded from our results that Smooth emulator outperforms DACE model (Gaussian emuator) in all cases.

In chapter 5 we developed a methodology to find analytical results for D- and A- optimality for two design points with replications. Though the replication is of no use in computer experiments, we explored the design choices in classical setting to develop a framework which can be extended to computer emulation. We incorporated Singular Value Decomposition to produce analytical results of D- and A- optimailty for Ridge regression. A detailed explorarory analysis is done for the optimal designs of Smooth emulator that provides some useful insights.

6.1 Limitations of Smooth Ridge model

We discuss here some of the limitations of the Smooth Ridge model.

1. For the matrix that is less than full rank design model matrix $MSE(\tilde{\beta}_\lambda) < MSE(\hat{\beta})$ under a condition on λ. In other words the inequality cannot be generalized for all values of λ but for a sufficient condition on λ.

2. It is difficult to employ the model when number of dimensions are increased. We employed up-to $D = 20$ dimensions for the comparisons so far. It is because the dimensions of the matrix K and design model matrix are increased which make them computationally expensive.

3. It is difficult to find analytical formulation of optimal designs due to numerical complexities. We achieved optimal designs for univariate and symmetric designs only.

6.2 Advantages of Smooth Ridge emulator

1. Smooth Ridge model provides a framework for computer model when number of design points are less than the model parameters. In other words design model matrix is less than full rank and usual Gaussian emulators cannot be used to model the computer experiments.

2. The model does not require the assumption of correlation structure among the response unlike most of the Gaussian emulators. Therefore, Smooth Ridge model can also be employed to the data which does not obey the properties of the Gaussian process.

3. Cheap alternative to expensive simulations

4. For the design model matrix that is full rank, simple Regression model becomes a special case of Smooth Ridge model when $\lambda = 0$.

5. For non-hierarchical models, there exist cases for which the proposed model becomes Ridge regression model for example, for the choice of basis for which the smoothness matrix becomes identity i.e. $K = I$.

6.3 Future recommendation

The methodology of smooth emulators proposed in this thesis opens new avenues to the modelling for computer experiments. In many real life applications with less number of observations and large number of potential input variables, this methodology is a stepping stone to build computer model. We have employed Legendre polynomials, smoothness measure and also combine the elements of Gaussian process. We also provide some insightful results from exploratory design study. Based on the current work we recommend some future research directions.

1. Search methodology to decide on the number of model terms for more than one factor. The number of supersaturated basis for one factor to be included in the model is given by [3]. This is achieved by the convergence of smoothness measure to that of splines. Therefore, a methodology can be built for more than one factor using convergence to spline as a starting point.

2. We achieved some optimal designs for the univariate and symmetric design points. This work provides a motivation to explore multivariate optimal designs but the search is limited to regular structures for example symmetric designs.

3. The Smooth Ridge model provides a good reason to employ LASSO methodology within the framework of smooth emulators. As an initial proposal we can write the smoothness matrix as $K = LU$ where L and U are lower and upper triangular respectively. The curvature $\beta^T K \beta$ can then be written as $\beta^T LU \beta$. We may use only one part of the matrix say L to construct Manhattan norm i.e. $||L\beta||_1$. The rest of the methodology needs to be developed that may produce some useful results.

4. The Bayesian methodology introduced in Chapter (3), Section (3.12) is not examined in detail to evaluate the model performance. Therefore, there is a potential to explore the

methodology with the use of methods like MCMC to evaluate the model performance against frequentist approach and contemorary emulators in future.

Appendix A

Appendices

A.1 Developments for Kennedy O Hagan model

A.1.1 Priors for unknown functions

The unknown functions $\eta(x, \theta)$ and $\delta(x)$ in equation(29) are assumed to have priors represented by Gaussian processes i.e.

$$\eta(x, t) \sim \mathcal{N}(m_1(x, \theta), c_1((x, \theta), (x', \theta'))\,.$$

$$\delta(x) \sim \mathcal{N}(m_2(x), c_2(x, x')\,.$$

where, $m_1(x, \theta) = h_1(x, \theta)^T \beta_1$ and $m_2(x) = h_2(x)^T \beta_2$. The covariance functions c_1 and c_2 takes any of the forms given in appendix. Also, $(\beta_1, \beta_2) \propto 1$
The covariance hyper-parameters for c_1 and c_2 denoted given by ψ and ρ, σ^2 are collectively denoted by ϕ. Therefore the complete parametric space include ϕ, $\beta = (\beta_1^T, \beta_2^T)$ and calibrated parameters θ i.e. (θ, β, ϕ)

A.1.2 Posterior Distribution

To begin with posterior distribution, we need the complete data that include the observed and the computer output and can be represented $d^T = (z^T, y^T)$, where

$$d|\theta, \beta, \phi \sim \mathcal{N}(m_d^\theta, V_d^\theta)\,.$$

In order to find the mean vector and variance matrix of the likelihood of complete data, following are some notations to be explained. $D_1 = (x_1^*, t_1) \dots (x_N^*, t_N)$ is the set of points where the code outputs y are available, $D_2 = (x_1 \dots x_n)$ is the set of points for the observations z of the real process and $D_2(\theta) = (x_1, \theta) \dots (x_n, \theta)$ is the set of variables augmented by the calibration parameters. In order to find the distribution of d we need to find the expressions for mean vector and variance matrix of joint distribution of y and z The expectation of y and z is given by

$$E(y) = E(\eta(x,t)) = H_1(D_1)\beta_1 \tag{A.1}$$

$$E(z) = E(\rho\eta(x_i,\theta) + \delta(x_i) + e_i) = \rho H_1 D_2(\theta)\beta_1 + H_2(D_2)\beta_2 \tag{A.2}$$

In order to find the variance matrix of z recall from 5.1, that

$$\eta(x,t) \sim \mathcal{N}(m_1(x,\theta), c_1((x,\theta),(x',\theta'))\,.$$

$$\delta(x) \sim \mathcal{N}(m_2(x), c_2(x,x')\,.$$

Hence, the variance matrix of z can be derived as

$$Var(z) = E(z^T z) - [E(z)]^T[E(z)] \tag{A.3}$$

$$\begin{aligned} E(z^T z) &= E[(\rho\eta(x,\theta) + \delta(x) + e(x))^T(\rho\eta(x,\theta) + \delta(x) + e(x)] \\ &= E[\rho^2\eta(x,\theta)^T\eta(x,\theta) + \delta(x)^T\delta(x) + e(x)^T e(x) + \eta(x,\theta)^T\delta(x) \\ &+ \rho\eta(x,\theta)^T e(x) + \rho\eta(x,\theta)\delta(x)^T + \delta(x)^T e(x)] \end{aligned} \tag{A.4}$$

$$\begin{aligned} (E(z))^T E(z) &= [E(\rho\eta(x,\theta) + \delta(x) + e(x))]^T[E(\rho\eta(x,\theta) + \delta(x) + e(x))] \\ &= \rho^2 E[\eta(x,\theta)^T\eta(x,\theta)] - E(\eta(x,\theta))^T E(\eta(x,\theta)) + E[\delta(x)^T\delta(x)] \\ &- [E(\delta(x))]^T E(\delta(x)) \end{aligned} \tag{A.5}$$

Replacing the terms in Equation (36), using Equation (37) and (38), the variance of z is found as

$$Var(z) = \rho^2(D_2(\theta)) + V_2(D_2) + \sigma^2 I \tag{A.6}$$

Once, the expressions for mean and variance for y and z are obtained, the distribution of d can be derived utilising the theory of joint Gaussian distribution and explicit expressions of its mean vector and variance matrix are as

$$m_d(\theta) = H(\theta)\beta \tag{A.7}$$

where,

$$H(\theta) = \begin{pmatrix} H_1(D_1) & 0 \\ \rho H_1(D_2(\theta)) & H_2(D_2) \end{pmatrix}$$

and

$$var(d) = V_d(\theta) = \begin{pmatrix} V_1(D_1) & \rho C_1(D_1, D_2(\theta))^T \\ \rho C_1(D_1, D_2(\theta)) & \lambda I_n + \rho^2 V_1(D_2(\theta)) + V_2(D_2) \end{pmatrix}$$

After evaluating the form of distribution of d , it is possible to write the posterior distribution as a product of likelihood $d|\theta, \beta, \phi$ and prior distributions

$$\pi(\theta, \beta, \phi|d) \propto \pi(d|(m_d(\theta), V_d(\theta))\pi(\theta)\pi(\phi) \tag{A.8}$$

In Equation (41) β can be integrated out by completing square in the following manner

$$\begin{aligned} f(d) &\propto (d - H(\theta)\beta)^T V_d(\theta)^{-1}(d - H(\theta)\beta) \\ &\propto d^T V_d(\theta)^{-1} d - 2d^T V_d(\theta)^{-1} H\beta + \beta^T H^T V_d(\theta)^{-1} H\beta \end{aligned} \tag{A.9}$$

differentiating Equation (42) with respect to β, yields $\hat{\beta}$

$$\begin{aligned} \hat{\beta} &= (H^T V^{-1} H)^{-1} H^T (V_d(\theta))^{-1} d \\ \hat{\beta} &= W(\theta) H^T (V_d(\theta))^{-1} d \end{aligned} \tag{A.10}$$

where, $W(\theta) = (H^T V^{-1} H)^{-1}$. In order to integrate out β from Equation (42), pdf of β can be expanded to complete square as follows

$$\begin{aligned} f(\beta) &\propto (\beta - \hat{\beta})^T W^{-1} (\beta - \hat{\beta}) \\ &\propto \beta^T W^{-1} \beta - 2\hat{\beta}^T W^{-1} \beta + \hat{\beta}^T W^{-1} \hat{\beta} \end{aligned} \tag{A.11}$$

adding and subtracting $2\hat{\beta}^T W^{-1} \beta + \hat{\beta}^T W^{-1} \hat{\beta}$ in equation (51) gives the following result

$$\begin{aligned} f(d) \propto\ & d^T V_d(\theta)^{-1} d - 2d^T V_d(\theta)^{-1} H\beta + 2\hat{\beta}^T W^{-1} \beta - \hat{\beta}^T W^{-1} \hat{\beta} \\ & + \beta^T H^T V_d(\theta)^{-1} H\beta - 2\hat{\beta}^T W^{-1} \beta + \hat{\beta}^T W^{-1} \hat{\beta} \end{aligned} \tag{A.12}$$

After integrating out β the posterior distribution can be defined and simplified as follows

$$\begin{aligned} & p(\theta, \phi|d) \propto p(\theta)p(\phi)p(d|\theta, \phi) \\ & p(\theta, \phi|d) \propto p(\theta)p(\phi)|V_d(\theta)|^{-\frac{1}{2}}|W(\theta)|^{\frac{1}{2}} \times \exp^{-\frac{1}{2}}[(d - H(\theta)\hat{\beta}(\theta))^T V_d(\theta)^{-1}(d - H(\theta)\hat{\beta}) \end{aligned} \tag{A.13}$$

A.1.3 Estimation of hyper-parameters

In order to evaluate the posterior distribution, it is imperative to find estimates of hyper-parameters,ϕ ,described in Section (6.2). The set of hyper-parameters ϕ are estimated in two steps. In first stage the code output y is used to estimate ψ_1 of $c_1((.,.),(.,.))$ i.e.

$$\begin{aligned} y|\beta_1,\psi_1 &\propto N(H_1(D_1)\beta_1, V_1(D_1)) \\ p(\beta_1,\psi_1|y) &\propto p(\psi_1)p(\beta_1)p(y|\beta_1,\psi_1) \end{aligned} \tag{A.14}$$

Integrating out β_1 from Equation (47), the resulting posterior distribution of ψ_1 takes the form

$$\begin{aligned} p(\psi_1|y) \propto p(\psi_1)|V_1(D_1)|^{-\frac{1}{2}}|W_1(D_1)|^{\frac{1}{2}} \\ \times \exp^{-\frac{1}{2}}[(y - H_1(D_1)\hat{\beta}_1(\theta)^T V_1(D_1)^{-1}(y - H_1(D_1)\hat{\beta}_1)] \end{aligned} \tag{A.15}$$

Once the stage 1 hyper-parameters are estimated, in stage two the observed data z is used to estimate ρ,λ and ψ_2 of $c_2(.,.)$ while keeping estimated ψ_1 fixed. The remaining set of estimated parameters can be estimated by maximizing $p(\rho,\sigma^2,\psi_2|d,\psi_1)$

$$p(\beta_2,\rho,\sigma^2,\psi_2|d,\psi_1) \propto p(\beta_2,\rho,\lambda,\psi_2) \times p(z|y,\beta_2,\phi) \tag{A.16}$$

The distribution of $z|y,\beta_2,\phi$ cannot be analytically obtained, however, the mean vector and variance matrix can be obtained using $z|y,\beta_2,\phi,\theta$ whereby $z|y,\beta_2,\phi$ can be approximated by a normal distribution with these moments. Using the approximation and integrating out β_2 from Equation (50), the posterior distribution takes the form

$$\begin{aligned} p(\rho,\sigma^2,\psi_2|d,\psi_1) \propto p(\rho,\lambda,\psi_2)|V|^{-\frac{1}{2}}|W_2|^{\frac{1}{2}} \times \exp^{-\frac{1}{2}}[(z - H_2(D_2)\hat{\beta}_2 \\ - \rho\hat{\eta}(D_2))^T V^{-1}(z - H_2(D_2)\hat{\beta}_2 - \rho\hat{\eta}(D_2))] \end{aligned} \tag{A.17}$$

where, $\hat{\beta}_2 = W_2 H_2(D_2) V_{-1}(z - \rho\hat{\eta}(D_2))$ and $W_2 = (H_2(D_2)^T V^{-1} H_2(D_2))^{-1}$

A.2 Covariance Functions for kriging models

The kriging metamodels depend heavily on the choice of the covariance functions K explained in Section (3). A crucial condition on K is to be chosen among the set of positive definite kernels in order to be admissible. Therefore, it is good practice to chose beforehand, a family of positive definite kernel for the covariance function and estimate the corresponding parameters

depending on the given data. In higher dimensions, one feasible way to get admissible covariance is by taking tensor products of 1-d admissible kernels. These kernels, termed separable, are used extensively in computer experiments literature. The current version of DiceKriging put forward by [63] provides covariance kernels that are built upon this model, up to a multiplicative constant $\sigma > 0$:

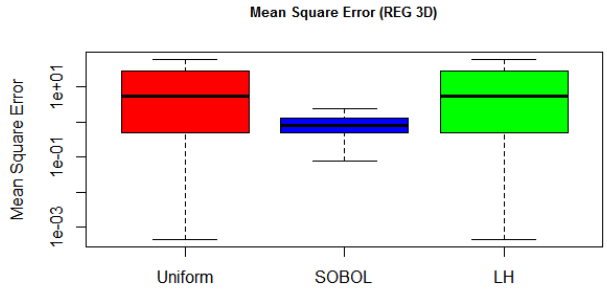
$$K(w,u) = \sigma^2 \prod_{j=1}^{d} k(h_j, \theta_j) \tag{A.18}$$

where, $h = (h_1, \ldots, h_d) := (w - u)$, θ_g is the covariance parameter called length scale and k is a 1-dimensional covariance kernel. Different formulas widely used for k are given in table(9)

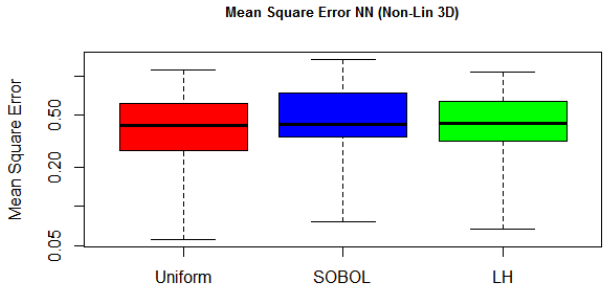
Gaussian:	$k(h) = exp\big(-\frac{h^2}{2\theta^2}\big)$
Matern $\nu = \frac{5}{2}$:	$k(h) = \big(1 + \frac{\sqrt{5}\lvert h\rvert}{\theta} + \frac{5h^2}{3\theta^2}\big)exp\big(-\frac{\sqrt{5}\lvert h\rvert}{\theta}\big)$
Matern $\nu = \frac{3}{2}$:	$k(h) = \big(1 + \frac{\sqrt{3}\lvert h\rvert}{\theta}exp\big(-\frac{\sqrt{3}\lvert h\rvert}{\theta}\big)$
Exponential:	$k(h) = exp\big(-\frac{\lvert h\rvert}{\theta}\big)$
Power-Exponential:	$k(h) = exp\big(-\big(\frac{\lvert h\rvert}{\theta}\big)\big)$

Table A.1: covariance kernels used in DiceKriging

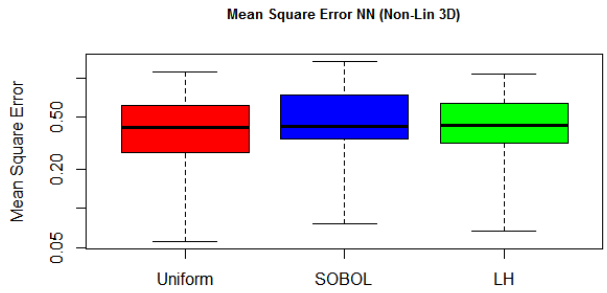
A.3 Box plots for Design study ($d = 3$) in Section (5.2)



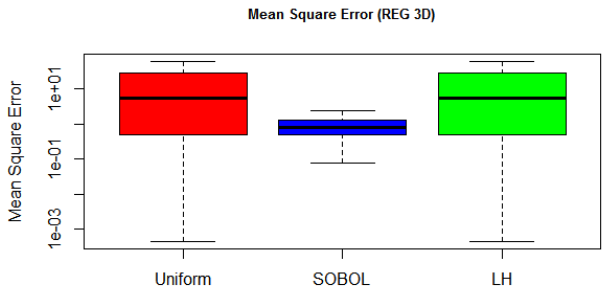
(a) Box plots for Dicekriging



(b) Box plots for Neural network (Non-Lin)

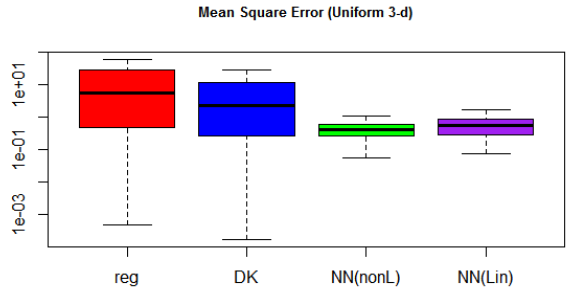


(c) Box Plots for Neural network (Lin)

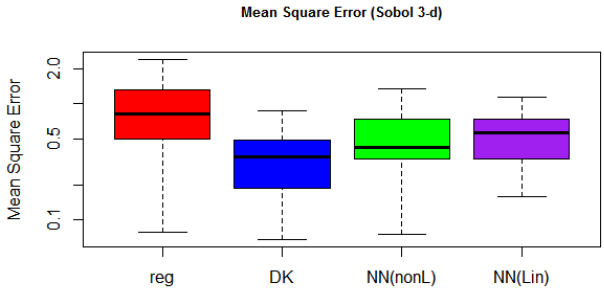


(d) Box Plots for linear regression

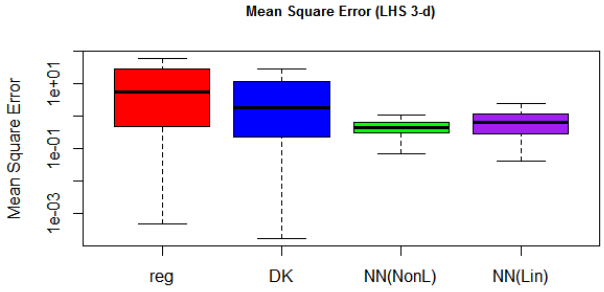
Figure A.1: Box plots of mean square error for different models (3-D)



(a) uniform samples



(b) Sobol' samples



(c) Latin hypercube sampling

Figure A.2: Box plots of mean square error for different designs (3-D)

designs	models			
	Regression	DACE	NN (non-Lin)	NN(Lin)
Latin hypercube	8.9470	4.2265	0.3930	0.6924
Sobol'	1.0206	5.0497	0.4275	0.7653
Uniform	9.2455	3.2877	0.4061	1.4035

Table A.2: MSE for different models against different designs

A.4 Relative efficiency plots for BRATLEY function (fixed n)

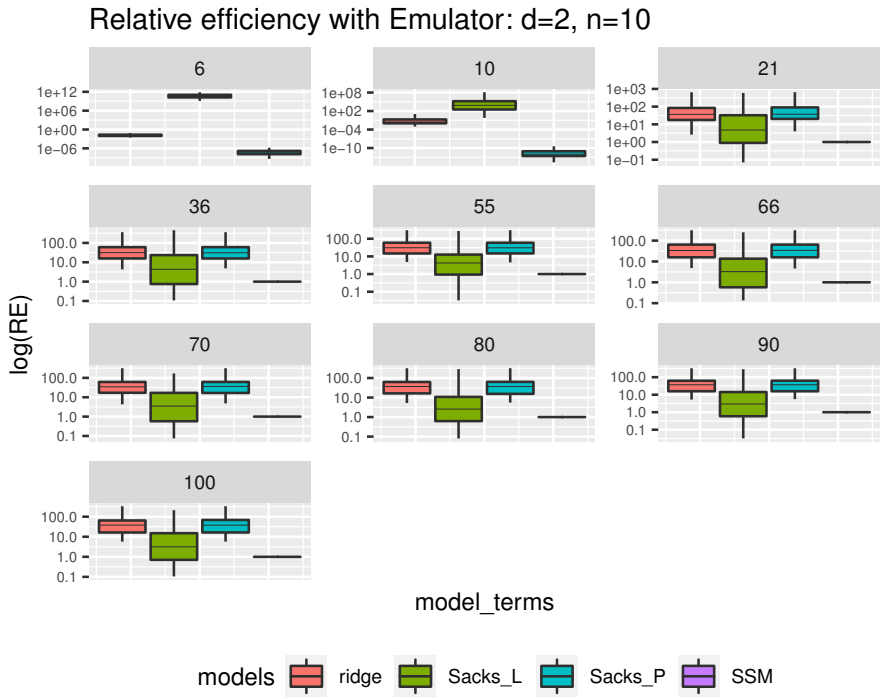


Figure A.3: RE for d=2, n=10

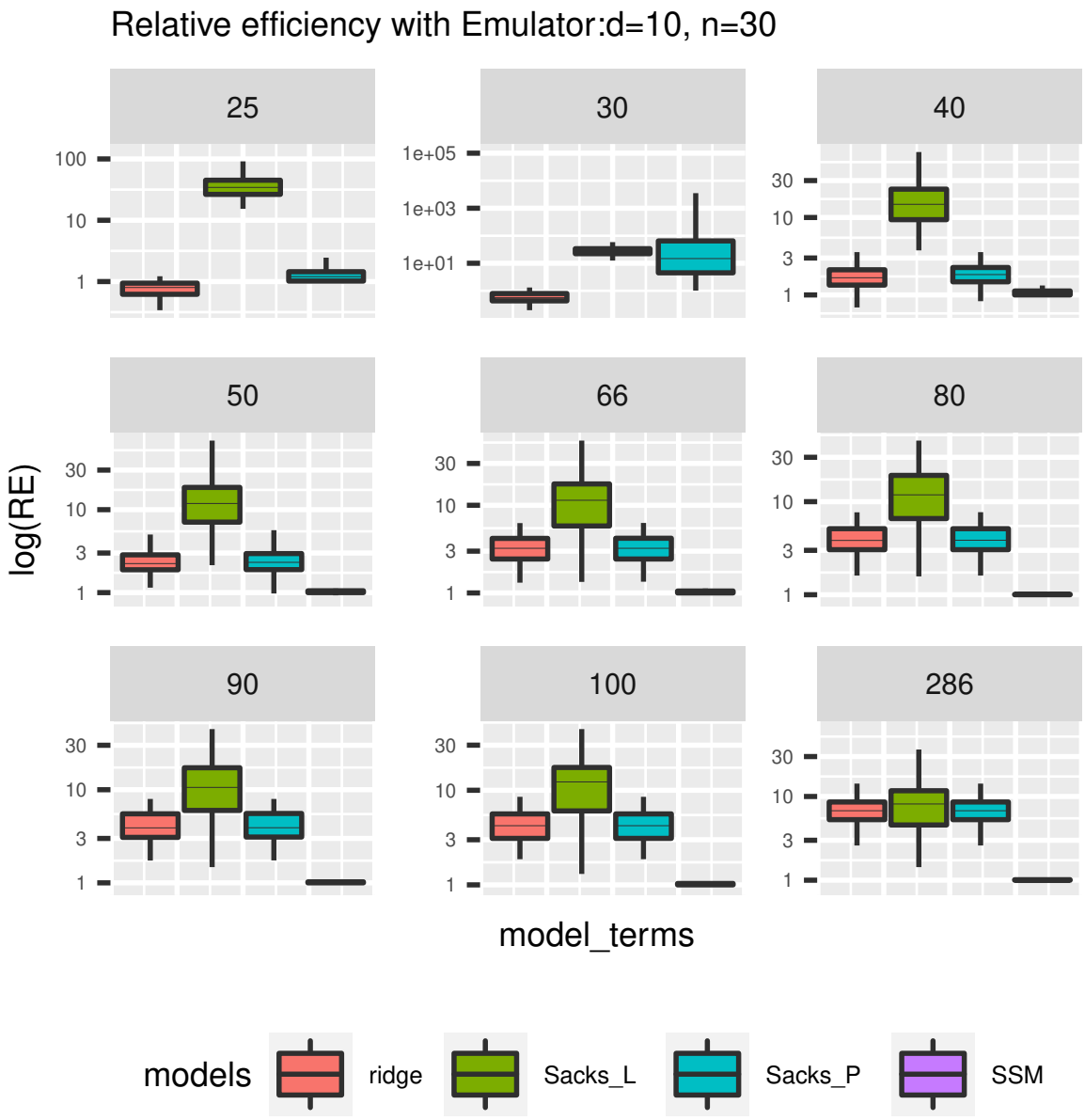


Figure A.4: RE for d=10, n=30

d=15

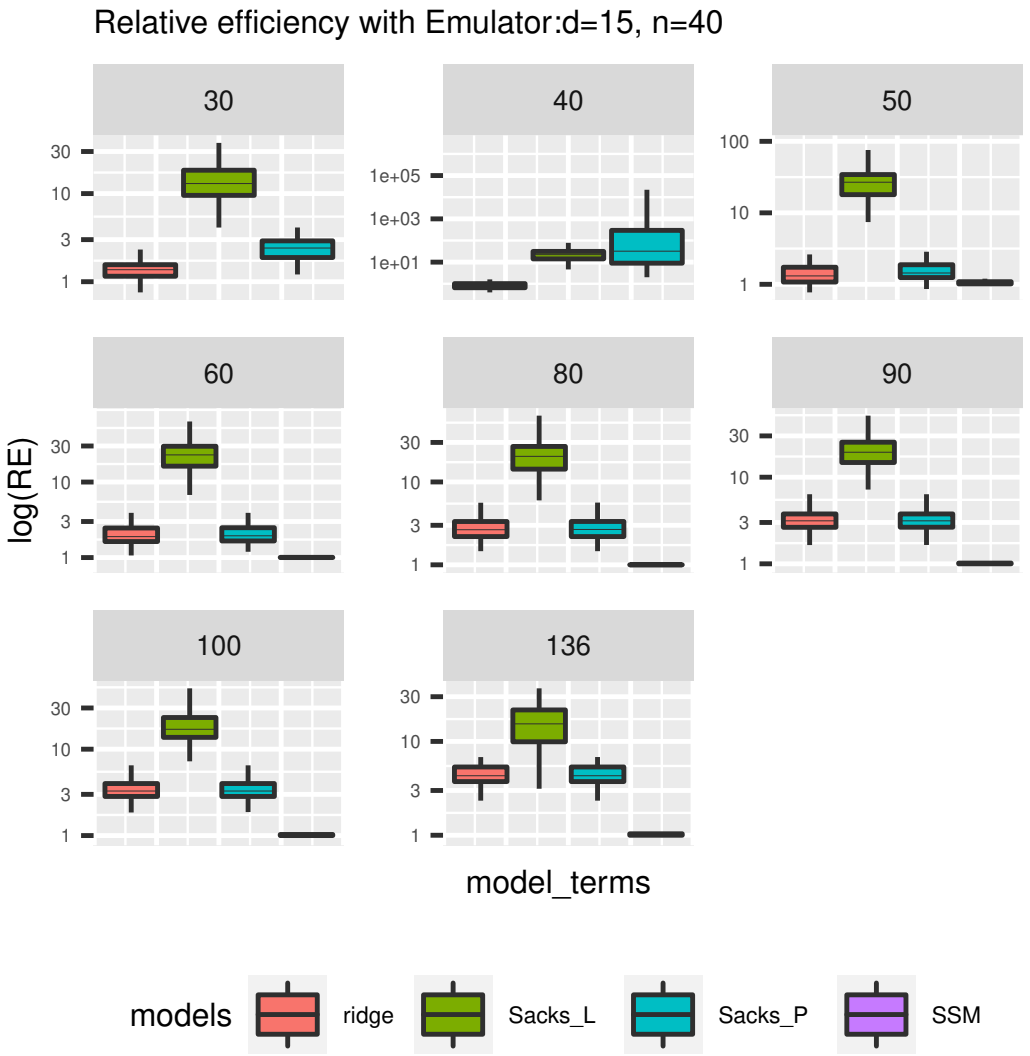


Figure A.5: RE for d=15, n=40

A.4.1 Relative efficiency plots for BRATLEY function (fixed k)

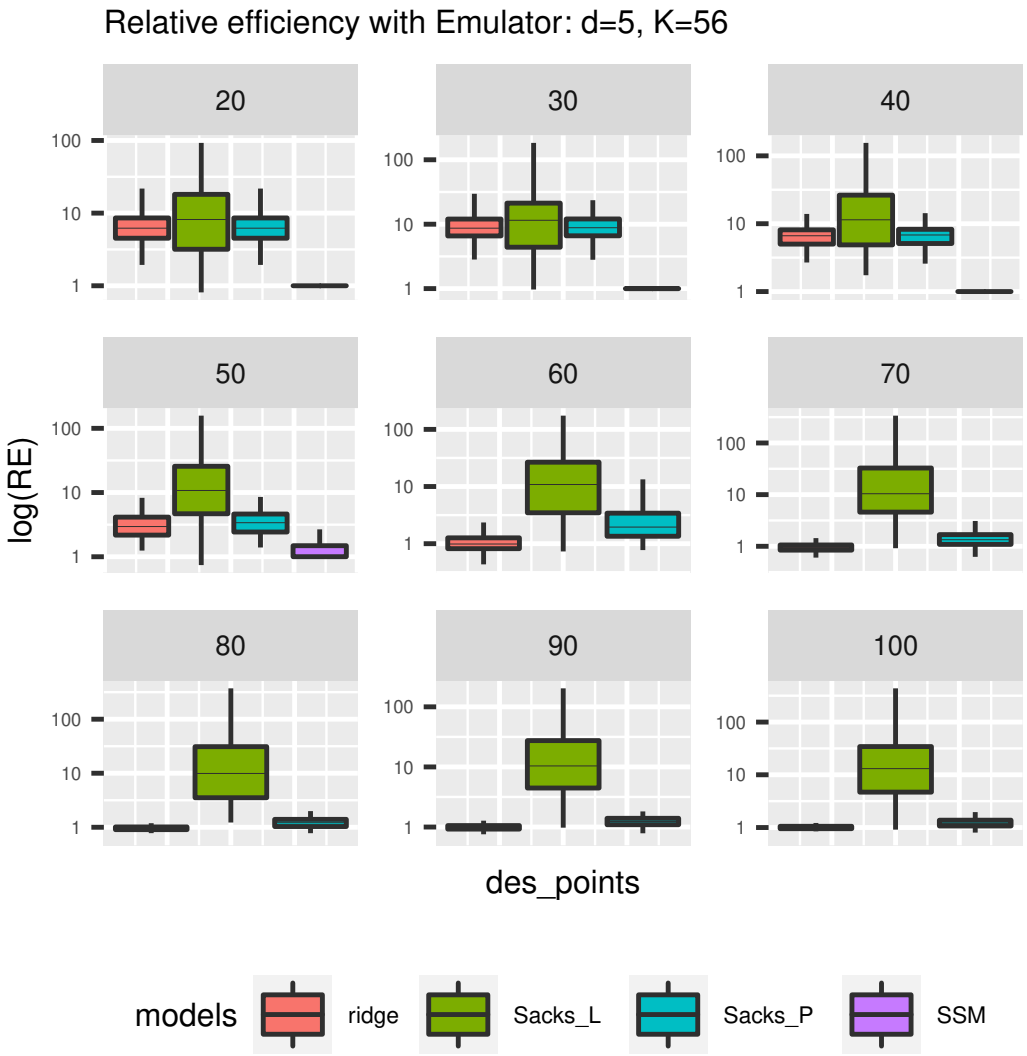


Figure A.6: RE for d=5, k=56

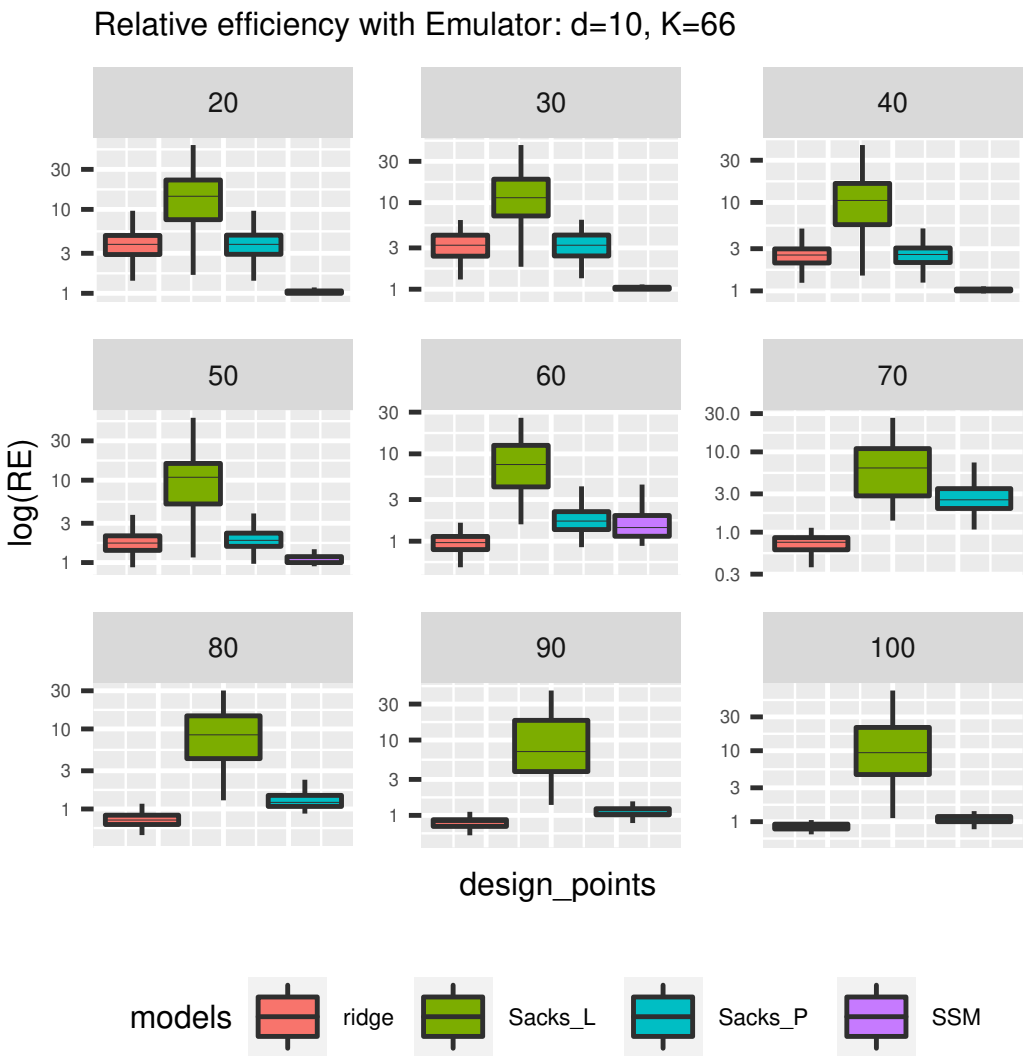


Figure A.7: RE for d=10, k=66

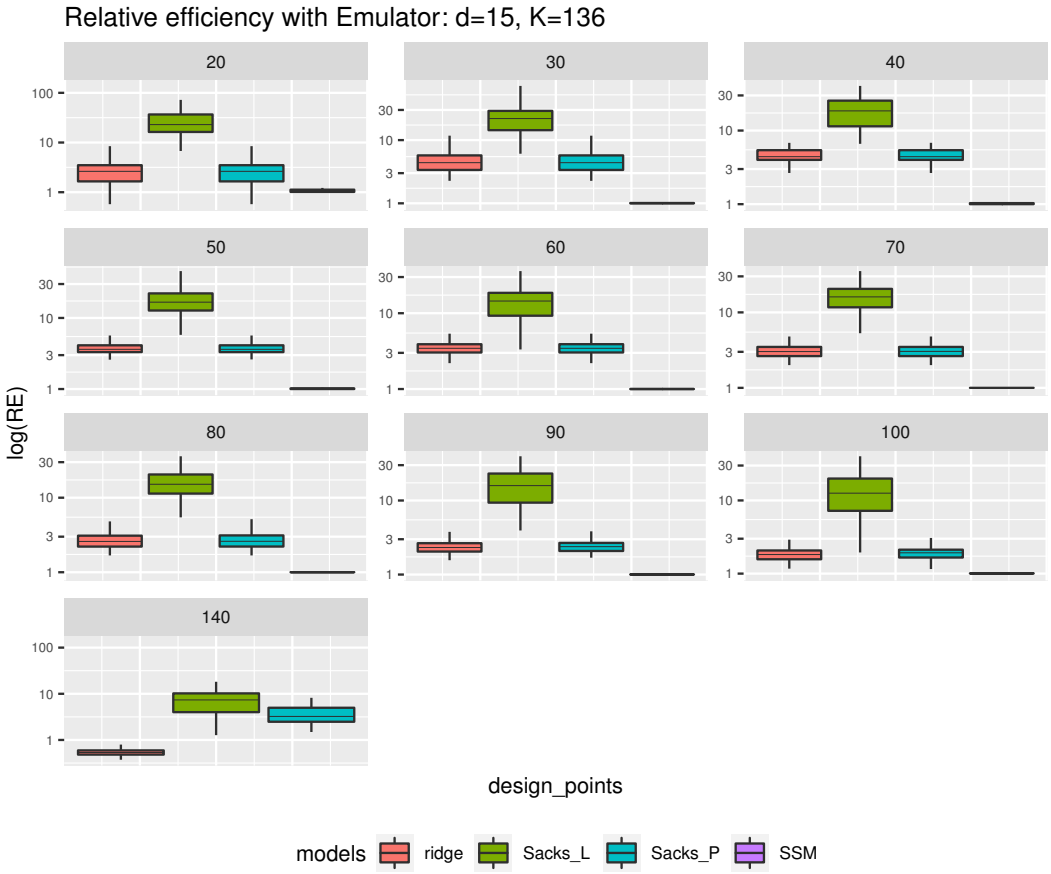


Figure A.8: RE for d=15, k= 136

A.5 Relative efficiency plots for Levy function (fixed n)

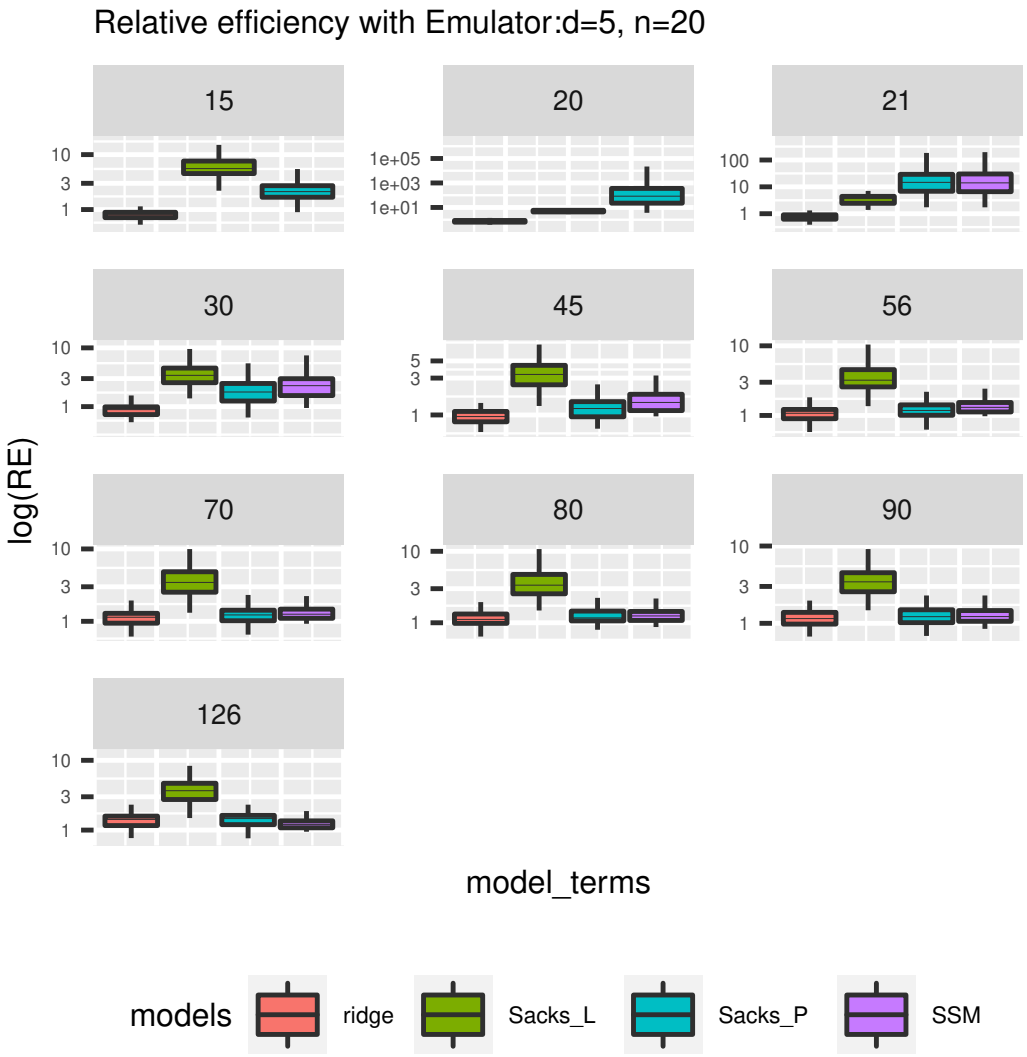


Figure A.9: RE for d=5, n= 20 (Levy)

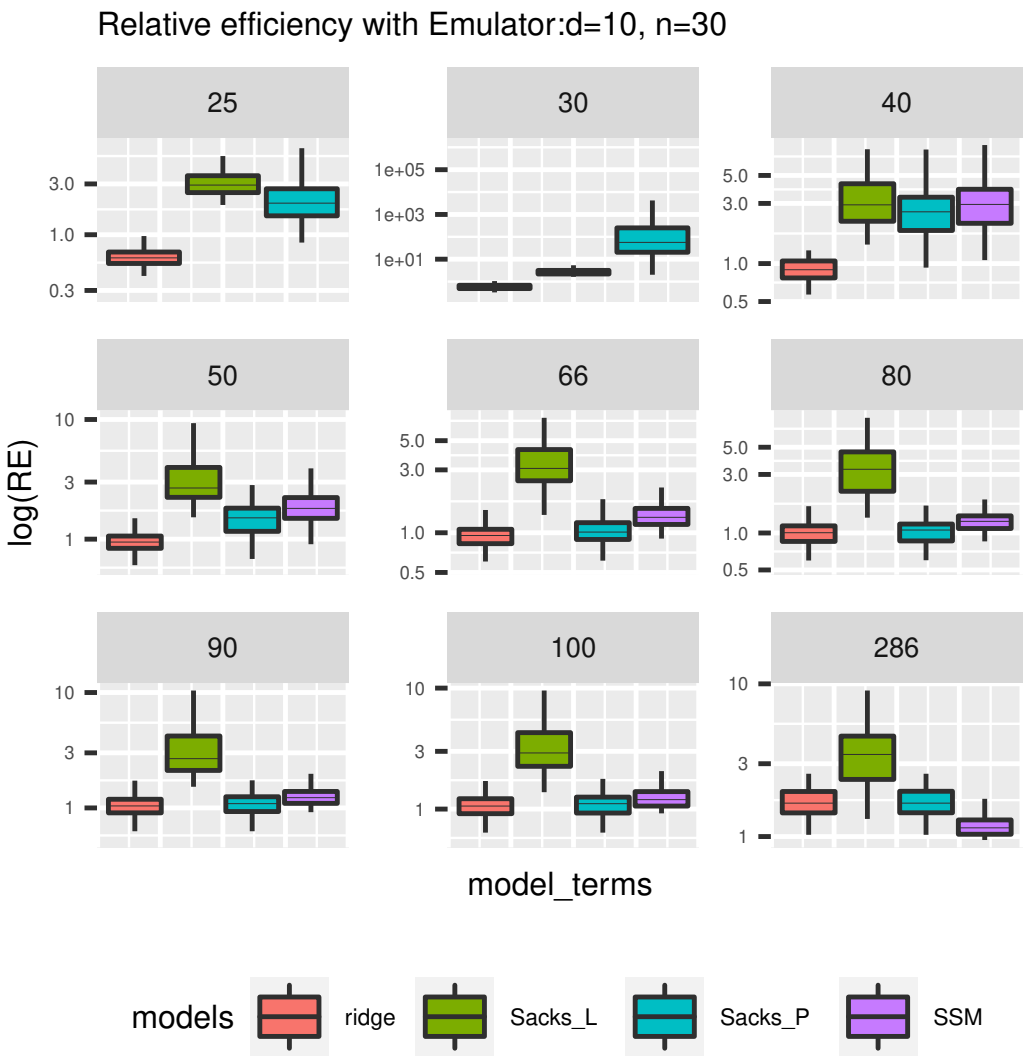


Figure A.10: RE for d=10, n= 30 (Levy)

A.5.1 Relative Efficiency plots for Levy function (fixed k)

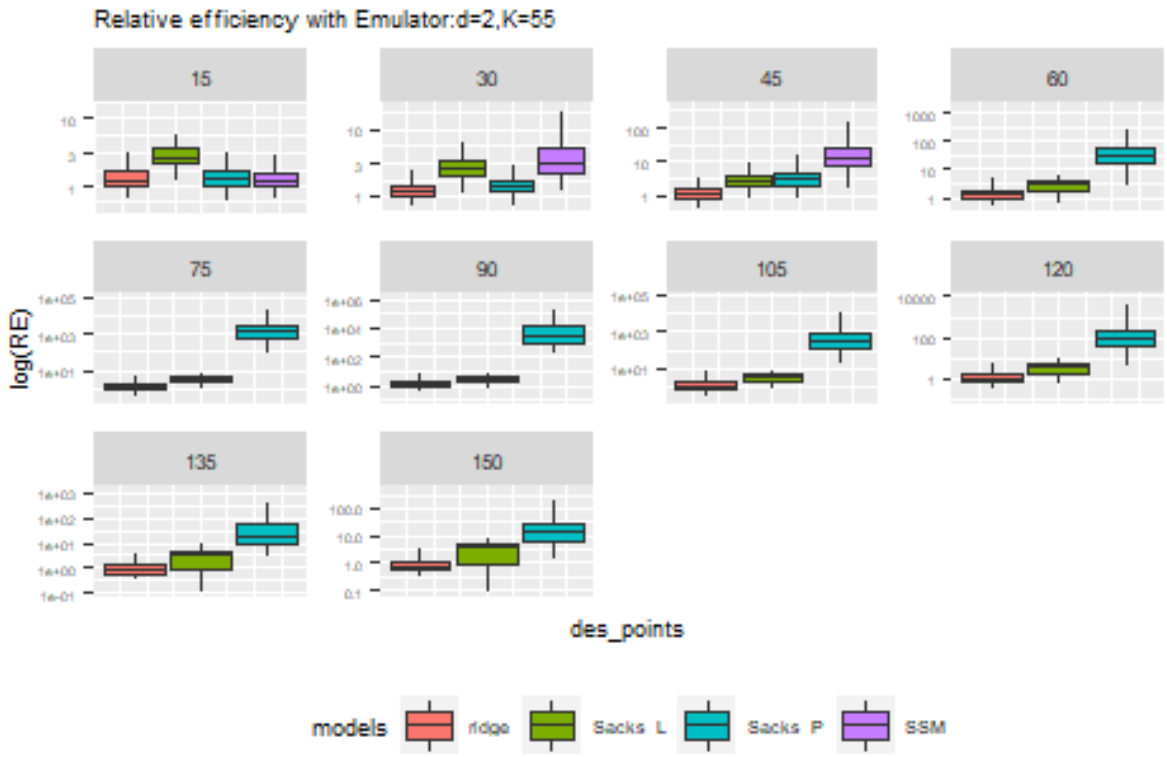


Figure A.11: RE for d=2, k=55 (Levy)

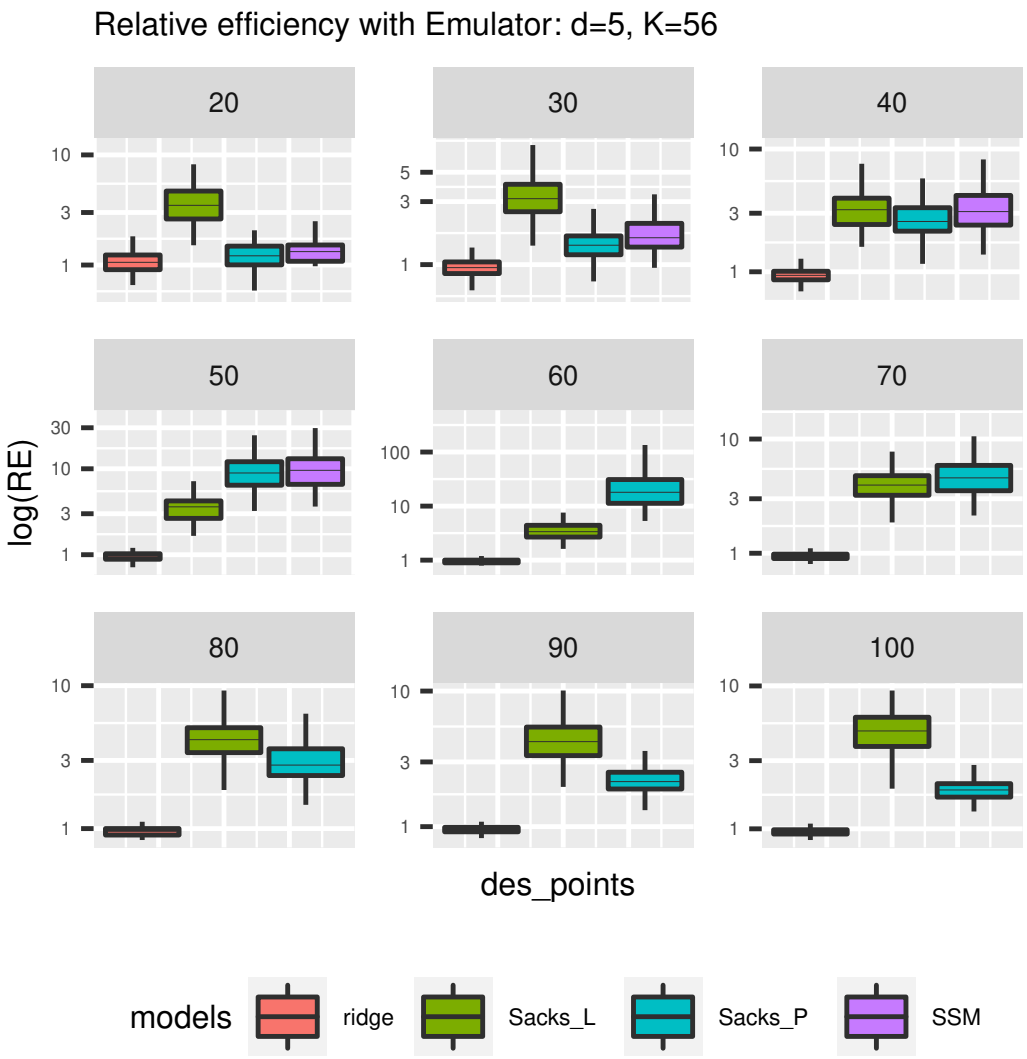


Figure A.12: RE for d=5, k=56 (Levy)

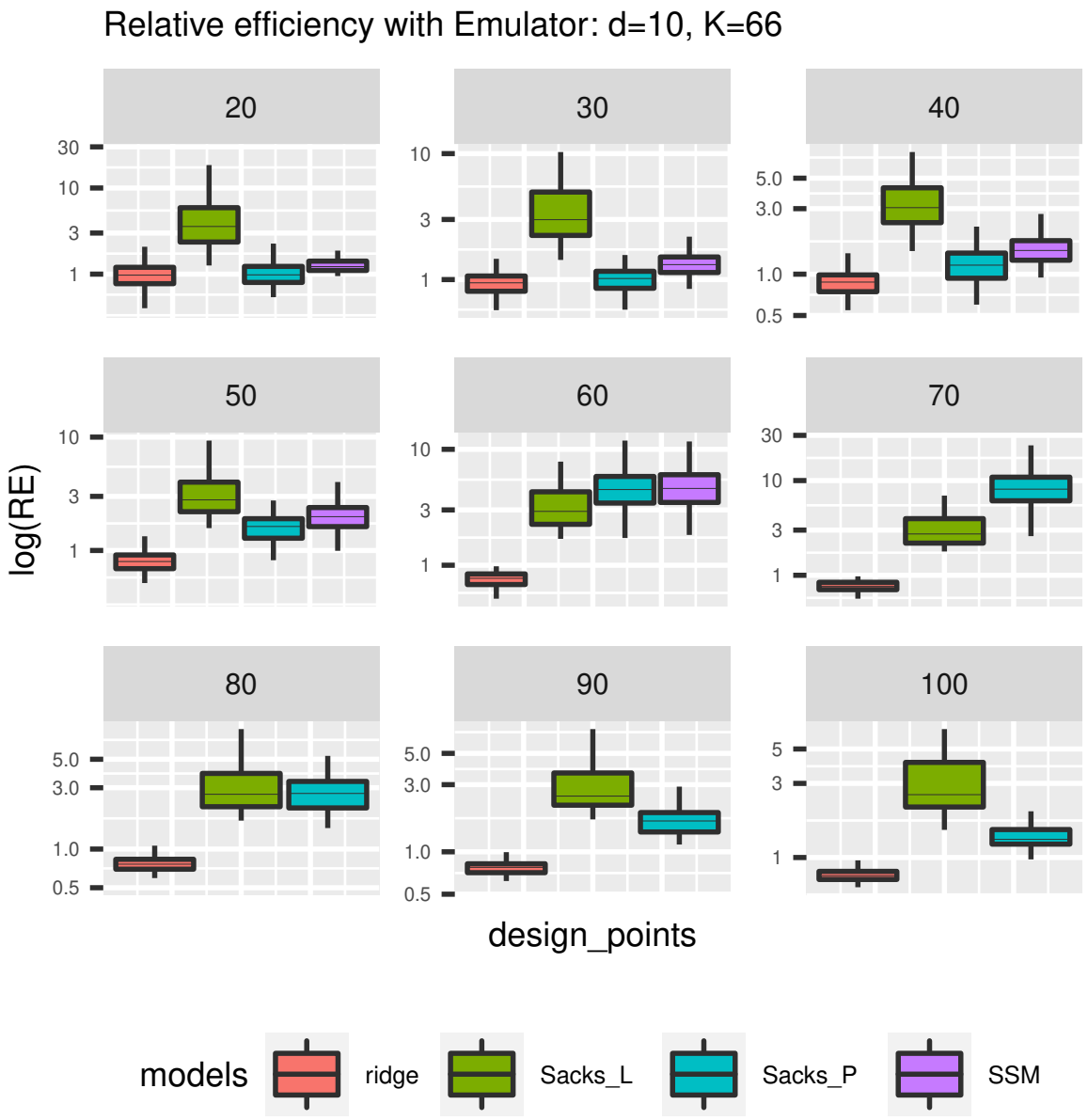


Figure A.13: RE for d=10, k=66 (Levy)

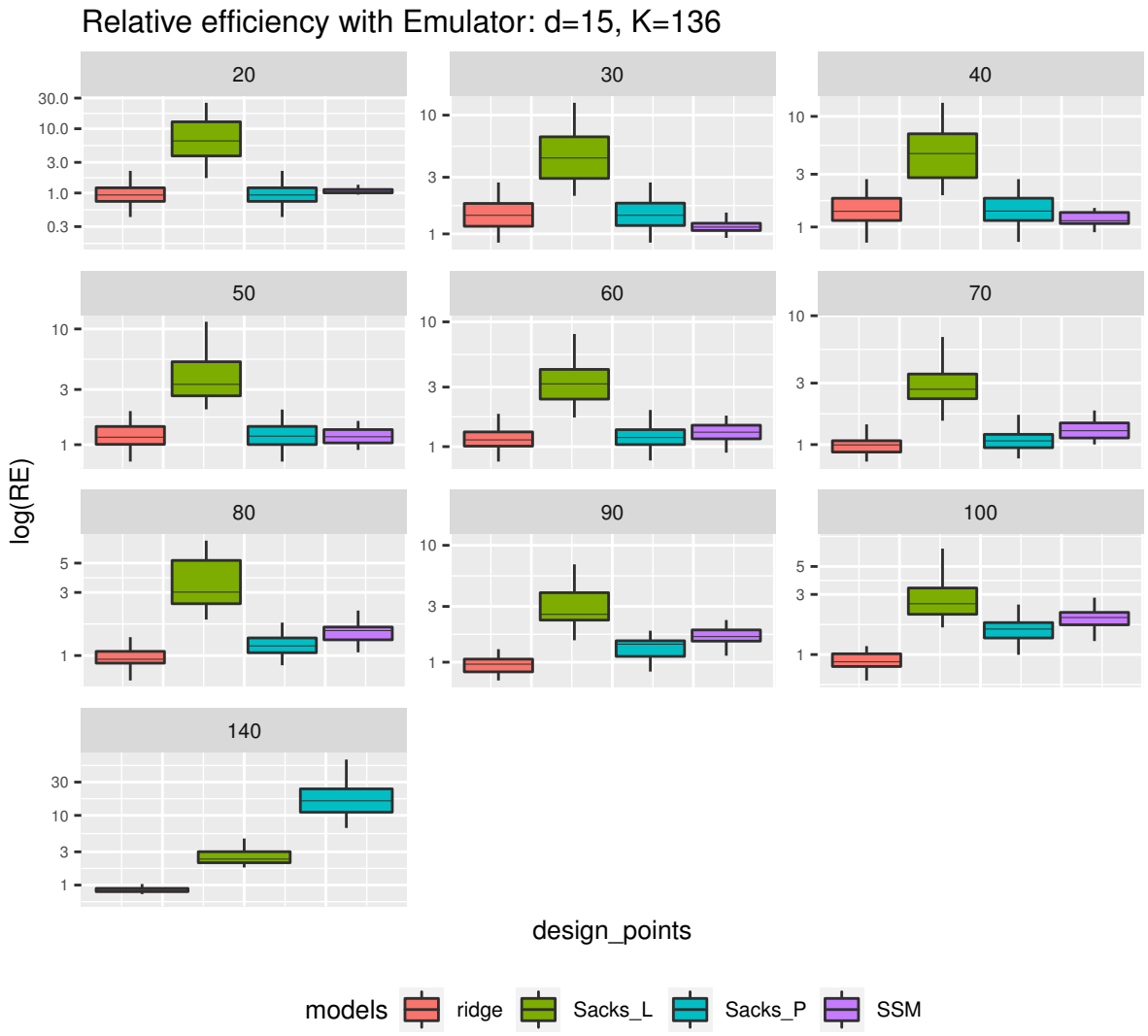


Figure A.14: RE for d=15, k=136 (Levy)

A.6 Plots for D- and A-optimality designs in Section (5.6)

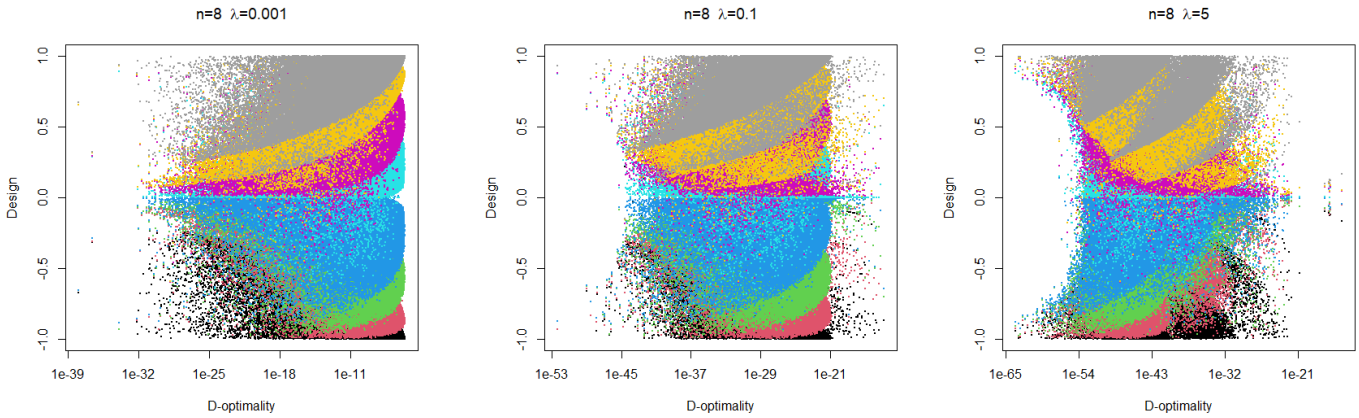


Figure A.15: Designs against D-optimality criterion $n = 8$

		D-optimal designs $n = 8$							
λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	D-opt ($\times 10^{38}$)
0.001	-0.6723	-0.65758	-0.6551	-0.6543	0.6543	0.6551	0.6575	0.6723	1.2652
0.1	-0.9327	-0.9322	-0. 8775	-0. 8767	0. 8767	0. 8775	0.9322	0.9327	4.7149×10^{-15}
1	0.9418	-0.9257	-0.9101	-0. 8786	0. 8786	0.9101	0.9257	0.9418	2.8595×10^{-26}

Table A.3: A-optimal designs and min-optimality $n = 8$

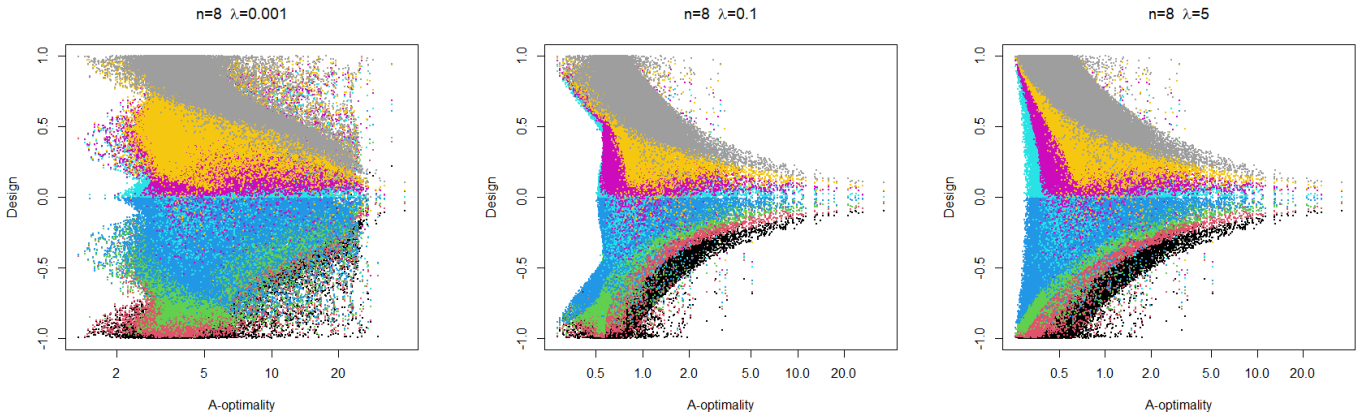


Figure A.16: Designs against A-optimality criterion $n = 8$

		A-optimal designs $n = 8$							
λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	A-opt
0.001	–0.9968	-0.4187	-0.4149	-0.4110	0.4110	0.4149	0.4187	0.9968	1.349
0.1	-0.9928	-0.9825	-0.9663	-0.9259	0.9259	0.9663	0.9825	0.9928	0.2845
1	-0.9928	-0.9825	-0.9663	-0.9259	0.92599	0.9663	0.9825	0.9928	0.2586

Table A.4: A-optimal designs and min-optimality $n = 8$

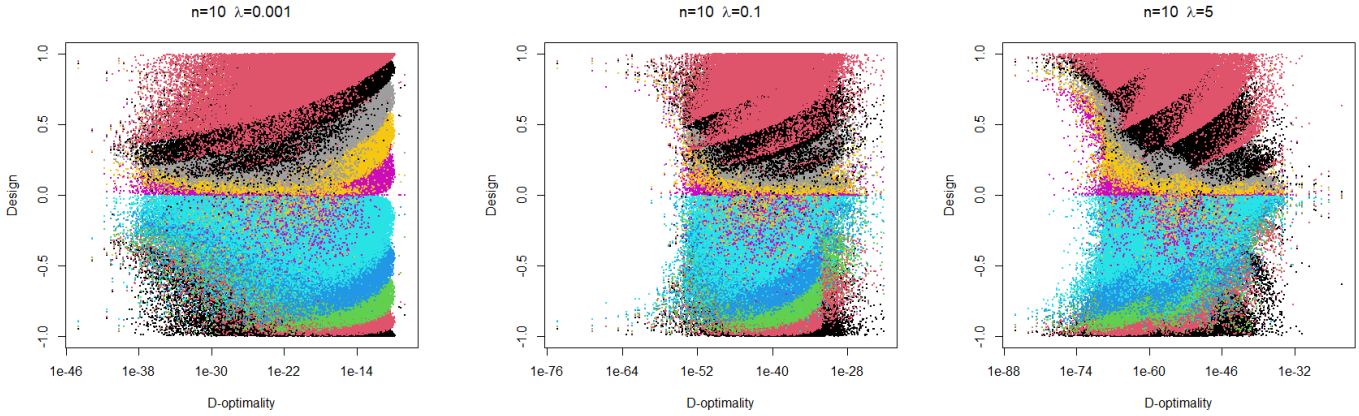


Figure A.17: Designs against D-optimality criterion $n = 10$

		D-optimal designs $n = 10$									
λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	D-opt ($\times 10^{45}$)
0.001	-0.948	-0.932	-0.930	-0.896	-0.894	0.894	0.896	0.930	0.932	0.948	2.40
0.1	-0.948	-0.932	-0.930	-0. 896	-0.894	0.894	0.896	0.930	0.932	0.948	4.9 $\times 10^{-30}$
1	-0.95	-0.94	-0.92	-0. 85	-0.84	0.84	0.85	0.92	0.94	0.95	1.8×10^{-41}

Table A.5: D-optimal designs and min-optimality $n = 10$

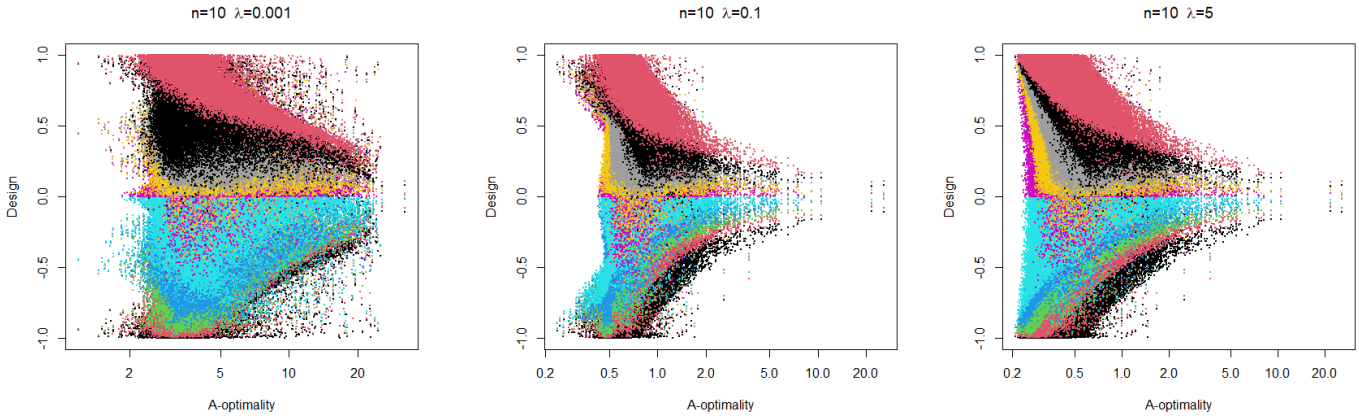


Figure A.18: Designs against A-optimality criterion $n = 10$

		A-optimal designs $n = 10$									
λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	A-opt)
0.001	-0.945	-0.939	-0.450	-0.450	-0.441	0.441	0.450	0.450	0.939	0.945	1.19
0.1	-0.912	-0.882	-0.868	-0.858	-0.858	0.858	0.858	0. 868	0. 882	0.9129	0.246
1	-0.967	-0.967	-0.9515	-0.929	-0.878	0.878	0.929	0.951	0.967	0.967	0.213

Table A.6: A-optimal designs and min-optimality $n = 10$

Bibliography

[1] Atkinson, A. C. and Donev, A. N. [1992], Optimum experimental designs, Technical report, Clarendon Press.

[2] Bailis, R., Ezzati, M. and Kammen, D. M. [2005], 'Mortality and greenhouse gas impacts of biomass and petroleum energy futures in africa', *Science* **308**(5718), 98–103.

[3] Bates, R. A., Curtis, P. R., Maruri-Aguilar, H. and Wynn, H. P. [2014], 'Optimal design for smooth supersaturated models', *Journal of Statistical Planning and Inference* **154**, 3–11.

[4] Bates, R. A., Maruri-Aguilar, H. and Wynn, H. P. [2014], 'Smooth supersaturated models', *Journal of Statistical Computation and Simulation* **84**(11), 2453–2464.

[5] Booker, A. J., Dennis, J., Frank, P. D., Serafini, D. B. and Torczon, V. [1998], Optimization using surrogate objectives on a helicopter test example, *in* 'Computational Methods for Optimal Design and Control', Springer, pp. 49–58.

[6] Box, G. E. P. and Wilson, K. B. [1951], 'On the experimental attainment of optimum conditions', *Journal of the Royal Statistical Society: Series B (Methodological)* **13**(1), 1–38.

[7] Boyd, S., Boyd, S. P. and Vandenberghe, L. [2004], *Convex optimization*, Cambridge university press.

[8] Bratley, P., Fox, B. L. and Niederreiter, H. [1992], 'Implementation and tests of low-discrepancy sequences', *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **2**(3), 195–213.

[9] Broomhead, D. S. and Lowe, D. [1988], 'Radial basis functions, multi-variable functional interpolation and adaptive networks', *Royal Signals and Radar Establishment Malvern (United Kingdom)* .

[10] Campbell, J. E., Carmichael, G. R., Chai, T., Mena-Carrasco, M., Tang, Y., Blake, D., Blake, N., Vay, S. A., Collatz, G. J., Baker, I. et al. [2008], 'Photosynthetic control of atmospheric carbonyl sulfide during the growing season', *Science* **322**(5904), 1085–1088.

[11] Cariboni, J., Gatelli, D., Liska, R. and Saltelli, A. [2007], 'The role of sensitivity analysis in ecological modelling', *Ecological modelling* **203**(1-2), 167–182.

[12] Chaloner, K. and Verdinelli, I. [1995], 'Bayesian experimental design: A review', *Statistical Science* pp. 273–304.

[13] Chambers, J. and Hastie, T. [1992], 'Statistical methods in s. pacific grove: Wadsworth & brooks'.

[14] Chang, P. B., Williams, B. J., Bhalla, K. S. B., Belknap, T. W., Santner, T. J., Notz, W. I. and Bartel, D. L. [2001], 'Design and analysis of robust total joint replacements: finite element model experiments with environmental variables', *Journal of Biomechanical Engineering* **123**(3), 239–246.

[15] contributors, W. [2021], 'Min-max theorem — Wikipedia, the free encyclopedia'. [Online; accessed 8-March-2021].

[16] Cooper, L. Y. [1982], 'A mathematical model for estimating available safe egress time in fires', *Fire and Materials* **6**(3-4), 135–144.

[17] Cornell, J. A. [2011], *Experiments with mixtures: designs, models, and the analysis of mixture data*, Vol. 403, John Wiley & Sons.

[18] Cortes, C. and Vapnik, V. [1995], 'Support-vector networks', *Machine learning* **20**(3), 273–297.

[19] Damianou, A. and Lawrence, N. [2013], Deep gaussian processes, *in* 'Artificial Intelligence and Statistics', pp. 207–215.

[20] De Boor, C. [2001], 'A practical guide to splines. revised edition. springer-verlag new york'.

[21] Dette, H., Melas, V. B. and Wong, W. K. [2006], 'Locally d-optimal designs for exponential regression models', *Statistica Sinica* pp. 789–803.

[22] Dette, H. and Pepelyshev, A. [2010], 'Generalized latin hypercube design for computer experiments', *Technometrics* **52**(4), 421–429.

[23] Exchange, M. S. [2016], 'Are the eigenvalues of the sum of two positive definite matrices increased?'.

[24] Farebrother, R. [1976], 'Further results on the mean square error of ridge regression', *Journal of the Royal Statistical Society. Series B (Methodological)* **38**(3), 248–250.

[25] Fermi, E., Pasta, P., Ulam, S. and Tsingou, M. [1955], Studies of the nonlinear problems, Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[26] Fisher, R. A. [1936], 'Design of experiments', *Br Med J* **1**(3923), 554–554.

[27] Gamerman, D. and Lopes, H. F. [2006], *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*, CRC press.

[28] Greiner, A. [2009], 'Estimating penalized spline regressions: Theory and application to economics', *Applied Economics Letters* **16**(18), 1831–1835.

[29] Harari, O. and Steinberg, D. M. [2014], 'Convex combination of gaussian processes for bayesian analysis of deterministic computer experiments', *Technometrics* **56**(4), 443–454.

[30] Hastie, T., Tibshirani, R., Friedman, J. H. and Friedman, J. H. [2009], *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2, Springer.

[31] Hastings, W. K. [1970], 'Monte carlo sampling methods using markov chains and their applications'.

[32] Hickernell, F. [1998], 'A generalized discrepancy and quadrature error bound', *Mathematics of computation* **67**(221), 299–322.

[33] Hoerl, A. E. and Kennard, R. W. [1970], 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67.

[34] Kennedy, M. C. and O'Hagan, A. [2001], 'Bayesian calibration of computer models', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(3), 425–464.

[35] Kieffer, J. [1959], 'Optimum experimental design', *Journal of Royal Statistical Society, Ser. B* **21**, 272–319.

[36] Kimeldorf, G. S. and Wahba, G. [1970], 'A correspondence between bayesian estimation on stochastic processes and smoothing by splines', *The Annals of Mathematical Statistics* **41**(2), 495–502.

[37] Krige, D. G. [1951], 'A statistical approach to some basic mine valuation problems on the witwatersrand', *Journal of the Southern African Institute of Mining and Metallurgy* **52**(6), 119–139.

[38] Kuhnt, S. and Steinberg, D. M. [2010], 'Design and analysis of computer experiments'.

[39] Laguna, M. and Marti, R. [2005], 'Experimental testing of advanced scatter search designs for global optimization of multimodal functions', *Journal of Global Optimization* **33**(2), 235–255.

[40] Lempert, R. J., Schlesinger, M. E., Bankes, S. C. and Andronova, N. G. [2000], 'The impacts of climate variability on near-term policy choices and the value of information', *Climatic Change* **45**(1), 129–161.

[41] Lin, Z., Xu, L. and Wu, Q. [2004], 'Applications of gröbner bases to signal and image processing: A survey', *Linear algebra and its applications* **391**, 169–202.

[42] Lu, T.-T. and Shiou, S.-H. [2002], 'Inverses of 2× 2 block matrices', *Computers & Mathematics with Applications* **43**(1-2), 119–129.

[43] Matheron, G. [1963], 'Principles of geostatistics', *Economic geology* **58**(8), 1246–1266.

[44] McCulloch, W. S. and Pitts, W. [1943], 'A logical calculus of the ideas immanent in nervous activity', *The bulletin of mathematical biophysics* **5**(4), 115–133.

[45] McKay, M. D., Beckman, R. J. and Conover, W. J. [1979], 'Comparison of three methods for selecting values of input variables in the analysis of output from a computer code', *Technometrics* **21**(2), 239–245.

[46] Micula, G. [2002], 'A variational approach to spline functions theory.', *General Mathematics* **10**(1-2), 21–50.

[47] Miller, S. L., Nazaroff, W. W., Jimenez, J. L., Boerstra, A., Buonanno, G., Dancer, S. J., Kurnitski, J., Marr, L. C., Morawska, L. and Noakes, C. [2021], 'Transmission of sars-cov-2 by inhalation of respiratory aerosol in the skagit valley chorale superspreading event', *Indoor air* **31**(2), 314–323.

[48] Montgomery, G. P. and Truss, L. T. [2001], Combining a statistical design of experiments with formability simulations to predict the formability of pockets in sheet metal parts, Technical report, SAE Technical Paper.

[49] Murphy, J. M., Sexton, D. M., Barnett, D. N., Jones, G. S., Webb, M. J., Collins, M. and Stainforth, D. A. [2004], 'Quantification of modelling uncertainties in a large ensemble of climate change simulations', *Nature* **430**(7001), 768–772.

[50] Myers, R. H., Montgomery, D. C. and Anderson-Cook, C. M. [2016], *Response surface methodology: process and product optimization using designed experiments*, John Wiley & Sons.

[51] Niederreiter, H. [1992], *Random number generation and quasi-Monte Carlo methods*, SIAM.

[52] Organization, W. H. [2021], 'Coronavirus disease (covid-19): How is it transmitted? 2020'.

[53] O'Hagan, A. [2006], 'Bayesian analysis of computer code outputs: A tutorial', *Reliability Engineering & System Safety* **91**(10-11), 1290–1300.

[54] Palomo, J., Paulo, R., García-Donato, G. et al. [2015], 'Save: an r package for the statistical analysis of computer models', *Journal of Statistical Software* **64**(13), 1–23.

[55] Pistone, G., Riccomagno, E. and Wynn, H. P. [2001], 'Algebraic statistics, volume 89 of monographs on statistics and applied probability'.

[56] Powell, M. J. D. [1977], 'Restart procedures for the conjugate gradient method', *Mathematical programming* **12**(1), 241–254.

[57] Quinonero-Candela, J., Rasmussen, C. E. and Williams, C. K. [2007], *Approximation methods for Gaussian process regression*, MIT Press.

[58] Rakovec, O., Hill, M. C., Clark, M., Weerts, A., Teuling, A. and Uijlenhoet, R. [2014], 'Distributed evaluation of local sensitivity analysis (delsa), with application to hydrologic models', *Water Resources Research* **50**(1), 409–426.

[59] Ratto, M. and Pagano, A. [2010], 'Using recursive algorithms for the efficient identification of smoothing spline anova models', *AStA Advances in Statistical Analysis* **94**(4), 367–388.

[60] Razavi, S., Tolson, B. A. and Burn, D. H. [2012], 'Review of surrogate modeling in water resources', *Water Resources Research* **48**(7).

[61] Reinsch, C. H. [1967], 'Smoothing by spline functions', *Numerische mathematik* **10**(3), 177–183.

[62] Riley, E., Murphy, G. and Riley, R. [1978], 'Airborne spread of measles in a suburban elementary school', *American journal of epidemiology* **107**(5), 421–432.

[63] Roustant, O., Ginsbourger, D. and Deville, Y. [2012], 'Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodelling and optimization', *Journal of Statistical Software* **51**(1), 54p.

[64] Ruppert, D., Wand, M. P. and Carroll, R. J. [2003], *Semiparametric regression*, number 12, Cambridge university press.

[65] Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. [1989], 'Design and analysis of computer experiments', *Statistical science* pp. 409–423.

[66] Saltelli, A. [2000], 'Sensitivity analysis; saltelli a, chan k, scott m, editors'.

[67] Saltelli, A. [2002], 'Making best use of model evaluations to compute sensitivity indices', *Computer physics communications* **145**(2), 280–297.

[68] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M. and Tarantola, S. [2008], *Global sensitivity analysis: the primer*, John Wiley & Sons.

[69] Santner, T. J., Williams, B. J., Notz, W. and Williams, B. J. [2003], *The design and analysis of computer experiments*, Vol. 1, Springer.

[70] Smith, K. [1918], 'On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations', *Biometrika* **12**(1/2), 1–85.

[71] Sobester, A., Forrester, A. and Keane, A. [2008], *Engineering design via surrogate modelling: a practical guide*, John Wiley & Sons.

[72] Sobol', I. M. [1967], 'On the distribution of points in a cube and the approximate evaluation of integrals', *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* **7**(4), 784–802.

[73] Sobol, I. M. [1993], 'Sensitivity analysis for non-linear mathematical models', *Mathematical modelling and computational experiment* **1**, 407–414.

[74] Sobol, I. M. [2001], 'Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates', *Mathematics and computers in simulation* **55**(1-3), 271–280.

[75] Sprott, D. A. [2008], *Statistical inference in science*, Springer Science & Business Media.

[76] Stein, M. L. [1999], *Interpolation of spatial data: some theory for kriging*, Springer Science & Business Media.

[77] Theobald, C. [1974], 'Generalizations of mean square error applied to ridge regression', *Journal of the Royal Statistical Society: Series B (Methodological)* **36**(1), 103–106.

[78] Tuo, R., Wu, C. J. et al. [2015], 'Efficient calibration for imperfect computer models', *The Annals of Statistics* **43**(6), 2331–2352.

[79] van Wieringen, W. N. [2015], 'Lecture notes on ridge regression', *arXiv preprint arXiv:1509.09169* .

[80] Vazquez, E. and Bect, J. [2011], 'Sequential search based on kriging: convergence analysis of some algorithms', *arXiv preprint arXiv:1111.3866* .

[81] Wagner, H. M. [1995], 'Global sensitivity analysis', *Operations Research* **43**(6), 948–969.

[82] Wahba, G. [1990], *Spline models for observational data*, SIAM.

[83] Wald, A. [1943], 'On the efficient design of statistical investigations', *The annals of mathematical statistics* **14**(2), 134–140.

[84] Willmott, C. J., Rowe, C. M. and Philpot, W. D. [1985], 'Small-scale climate maps: A sensitivity analysis of some common assumptions associated with grid-point interpolation and contouring', *The American Cartographer* **12**(1), 5–16.

[85] Woodward, P. M. and Davies, I. L. [1952], 'Information theory and inverse probability in telecommunication', *Proceedings of the IEE-Part III: Radio and Communication Engineering* **99**(58), 37–44.

[86] Wynn, H. P. [1970], 'The sequential generation of d-optimum experimental designs', *The Annals of Mathematical Statistics* **41**(5), 1655–1664.

[87] Zhao, V. [2019], 'Machine learning for beginners: An introduction to neural networks', https:// towardsdatascience.com/machine learning for beginners an introduction to neural networks d49f22d238f9.

[88] Zhou, Y. [1999], 'Adaptive importance sampling for integration.'.