Thomas J. Santner
Brian J. Williams
William I. Notz

The Design and Analysis of Computer Experiments

February 18, 2014

Springer

Use the template dedic.tex *together with the Springer document class SVMono for monograph-type books or SVMult for contributed volumes to style a quotation or a dedication at the very beginning of your book in the Springer layout*

Preface

Use the template *preface.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your preface in the Springer layout.

A preface is a book's preliminary statement, usually written by the *author or editor* of a work, which states its origin, scope, purpose, plan, and intended audience, and which sometimes includes afterthoughts and acknowledgments of assistance.

When written by a person other than the author, it is called a foreword. The preface or foreword is distinct from the introduction, which deals with the subject of the work.

Customarily *acknowledgments* are included as last part of the preface.

Place(s),
month year

Firstname Surname
Firstname Surname

Acknowledgements

Use the template *acknow.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) if you prefer to set your acknowledgement section as a separate chapter instead of including it as last part of your preface.

Contents

To Gail, Aparna, and Claudia

for their encouragement and patience

Preface

Use the template *preface.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your preface in the Springer layout.

A preface is a book's preliminary statement, usually written by the *author or editor* of a work, which states its origin, scope, purpose, plan, and intended audience, and which sometimes includes afterthoughts and acknowledgments of assistance.

When written by a person other than the author, it is called a foreword. The preface or foreword is distinct from the introduction, which deals with the subject of the work.

Customarily *acknowledgments* are included as last part of the preface.

Place(s), *Firstname Surname*
month year *Firstname Surname*

–

Chapter 1
Physical Experiments and Computer Experiments

1.1 Introduction

Historically, there has been extensive use of both physical experiments and, later, stochastic simulation experiments in order to determine the impact of input variables on outputs of scientific, engineering,or teachnological importance.

In the past 15 to 20 years, there has been an increasing use of computer codes to infer the effect of input variables on such outputs. To explain their genesis, suppose that a mathematical theory exists that relates the output of a complex physical process to a set of input variables, e.g., a set of differential equations. Secondly, suppose that a numerical method exists for accurately solving the mathematical system. Typical methods for solving complex mathematical systems include finite element (FE) and computational fluid dynamics (CFD) schemes. In other applications, code output are simply extremely sophisticated simulations run to the point that the simulation error is essentially zero. Assuming that sufficiently powerful computer hardware and software exists to implement the numerical method, one can treat the output of the resulting computer code as an experimental output corresponding the inputs to that code; the result is a *computer experiment* that produces a "response" corresponding to any given set of input variables.

This book describes methods for designing and analyzing research investigations that are conducted using computer codes that are used either alone or in addition to a physical experiment.

Historically, Statistical Science has been the scientific discipline that creates methodology for designing empirical research studies and analyzing data from them. The process of designing a study to answer a specific research question must first decide which variables are to be observed and the role that each plays, eg, as an explanatory variable or a response variable. Traditional methods of data collection include retrospective techniques such as cohort studies and the case-control studies used in epidemiology. The gold standard data collection method for establishing cause and effect relationships is the prospective designed experiment. *Agricultural field experiments* were one of the first types of designed experiments. Over time,

many other subject matter areas and modes of experimentation have been developed. For example, *controlled clinical trials* are used to compare medical therapies and *stochastic simulation experiments* are used extensively in operations research to compare the performance of (well) understood physical systems having stochastic components such as the flow of material through a job shop.

There are critical differences between data generated by a physical experiment and data generated by a computer code that dictate that different methods must be used to analyze the resulting data. Physical experiments measure a stochastic response corresponding to a set of (experimenter-determined) treatment input variables. Unfortunately, most physical experiments also involve nuisance input variables that may or may not be recognized and cause (some of the) variation in the experimental response. Faced with this reality, statisticians have developed a variety of techniques to increase the validity of treatment comparisons for physical experiments. One such method is *randomization*. Randomizing the order of applying the experimental treatments is done to prevent unrecognized nuisance variables from systematically affecting the response in such a way as to be confounded with treatment variables. Another technique to increase experimental validity is *blocking*. Blocking is used when there are recognized nuisance variables, such as different locations or time periods, for which the response is expected to behave differently, even in the absence of treatment variable effects. For example, yields from fields in dryer climates can be expected to be different from those in wetter climates and males may react differently to a certain medical therapy than females. A block is a group of experimental units that have been predetermined to be homogeneous. By applying the treatment in a symmetric way to blocks, comparisons can be made among the units that are as similar as possible, except for the treatment of interest. *Replication* is a third technique for increasing the validity of an experiment. Adequate replication means that an experiment is run on a sufficiently large scale to prevent the unavoidable "measurement" variation in the response from obscuring treatment differences.

In some cases computer experimentation is feasible when physical experimentation is not. For example, the number of input variables may be too large to consider performing a physical experiment, there may be ethical reasons why a physical experiment cannot be run, or it may simply be economically prohibitive to run an experiment on the scale required to gather sufficient information to answer a particular research question. However, we note that when using only computer experiments to determine the multivariate relationship between a set of inputs and a critical output based on an assumed model that has no empirical verification is an extrapolation and all extrapolations can be based on incorrect assumptions.

Nevertheless, the number of examples of scientific and technological developments that have been conducted using computer codes are many and growing. They have been used to predict climate and weather, the performance of integrated circuits, the behavior of controlled nuclear fusion devices, the properties of thermal energy storage devices, and the stresses in prosthetic devices. More detailed motivating examples will be provided in Section 1.2.

In contrast to classical physical experiments, a computer experiment yields a deterministic answer (at least up to "numerical noise") for a given set of input conditions; indeed, the code produces *identical* answers if run twice using the same set of inputs. Thus, using randomization to avoid potential confounding of treatment variables with unrecognized "noise" factors is irrelevant–only code inputs can affect the code output. Similarly, blocking the runs into groups that represent "more nearly alike" experimental units is also irrelevant. Indeed, none of the traditional principles of blocking, randomization, and replication are of use in solving the design and analysis problems associated with computer experiments. However, we still use the word "experiment" to describe such a code because the goal in both physical and computer experiments is to determine which treatment variables affect a given response and, for those that do, to quantify the details of the input-output relationship.

In addition to being deterministic, the codes used in some computer experiments can be time-consuming; in some finite element models, it would not be unusual for code to run for 12 hours or even considerably longer to produce a single response. Another feature of many computer experiments is that the number of input variables can be quite large–15 to 20 or more variables. One reason for the large number of inputs to some codes is that the codes allow the user to not only manipulate the control variables (the inputs that can be set by an engineer or scientist to control the process of interest) but also inputs that represent operating conditions and/or inputs that are part of the physics or biology model being studied but that are known only to the agreement of experts in the subject matter field. As examples of the latter two types of inputs, consider biomechanics problem of determining the strain that occurs at the bone–prosthesis boundary of a prosthetic knee. Of course, prosthesis geometry is one input that determines the strain. But this output also depends on the magnitude of the load (an operating condition input) and friction between the prosthetic joint and the bone (a model-based input). Other examples of problems with both engineering and other types of input variables are given in Section 2.1.

This book will discuss methods that can be used to design and analyze computer experiments that account for their special features. The remainder of this chapter will provide several motivating examples and an overview of the book.

1.2 Examples of Computer Models

This section sketches several scientific arenas where computer models are used. Our intent is to show the *breadth* of application of such models and to provide some feel for the types of inputs and outputs used by these models. The details of the mathematical models implemented by the computer code will not be given, but references to the source material are provided for interested readers.

*Example 1.1 (*A Physics-based Tower Model). Consider an experiment in which a ball is dropped, not thrown, from an initial height x (in meters). The experimenter measures the time (in seconds) from when the ball is dropped until it hits the ground,

say $y(x)$. A computational simulator of this experiment can be derived from Newton's Law with drag coefficient. Let $s(\tau)$ denote the position of the particle at time τ and let θ denote the coefficient of drag. While the value of θ in the physical experiment is unknown, assume that based on engineering experience, it is possible to specify a prior distribution, $\pi(\theta)$, of likely values for θ. Then Newton's law states

$$\frac{d^2 s(\tau)}{d\tau^2} = -1 - \theta\left(\frac{ds(\tau)}{d\tau}\right) \tag{1.2.1}$$

subject to the initial conditions $s(0) = x$ and $\frac{ds(\tau)}{d\tau}\,|_{x=0} = 0$. The first condition states the initial height of the ball is x and the second condition states that the ball is dropped rather than than thrown with some positive initial velocity. The computational model output $\eta(x, \theta)$ is the (smallest) zero of the equation

$$s(\tau) = 0\ . \tag{1.2.2}$$

Table 1.1 lists data from a set of computational solutions to this model. The data

x	θ	$\eta(x, \theta)$
0.042	0.125	1.6066
0.125	0.375	2.0309
0.000	0.583	1.6384
0.167	0.708	2.4639
0.083	0.875	2.2184
0.333	0.000	2.3094
0.208	0.208	2.1730
0.292	0.500	2.7144
0.375	0.792	3.4574
0.250	0.917	3.0891
0.583	0.0417	2.8543
0.417	0.292	2.8228
0.542	0.417	3.3511
0.500	0.625	3.6211
0.458	1.00	4.2778
0.708	0.167	3.2887
0.750	0.333	3.7132
0.667	0.542	3.9795
0.625	0.750	4.3771
0.792	0.833	5.3177
0.875	0.0833	3.4351
1.000	0.250	4.0449
0.833	0.458	4.2369
0.958	0.667	5.3178
0.917	0.958	6.3919

Table 1.1 Data from a set of computational simulations of the time for a ball to drop from a given height x when the drag coefficient is θ.

from the computational model is shown in Figure 1.1. Notice that the (x, θ) inputs 'cover' the input space, that the drop time increases in both x and θ.

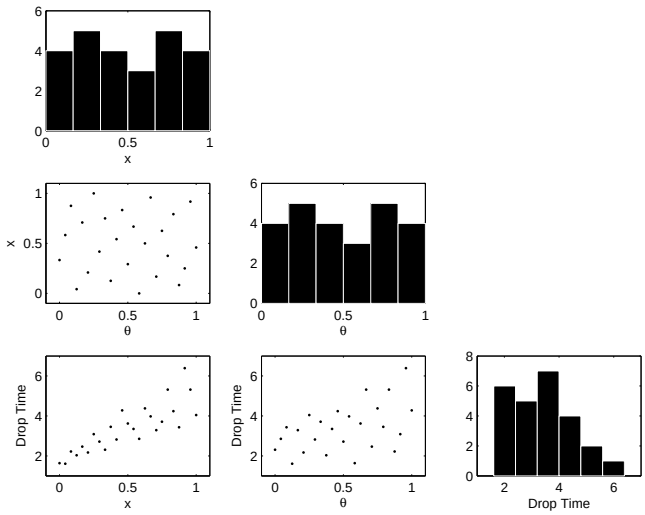


Fig. 1.1 Matrix scatterplot of Tower Data.

Corresponding to the computational simulations are the results from six physical experiments of timing a dropped ball from a given height

x	Drop Time (sec.)
0.0	1.5929
0.2	2.1770
0.4	2.8706
0.6	3.8330
0.8	4.5965
1.0	4.7509

Table 1.2 Observed times for a ball to drop a given height x.

The objectives from this set of experiments might be several fold: (1) to estimate the "true" drag coefficient (or provide a distribution of values consistent with the data), (2) to predict the Drop Time at an untested height, (3) to quantify the uncertainty in the predicted values in (2).

*Example 1.2 (*Stochastic Evaluation of Prosthetic Devices). *see Kevin paper in literature* In this example taken from Ong et al (2008), a three-dimensional 14,000-node finite element model of the pelvis was used to evaluate the inpact of biomechanical engineering design variables on the performance of an acetabular cup under a distri-

bution of patient loading environments and deviations from ideal surgical insertion parameters.

previously developed in detail

Design and environmental variables, their notation, ranges, and distributions 2 denotes the chi-square distribution with degrees of freedom; $N(2, ?)$ denotes the normal Gaussian distribution with mean and variance σ^2; $U(a, b)$ denotes the uniform distribution over the interval (a, b), $DU(a_1, \ldots, a_d)$ denotes the discrete uniform distribution over the values $a_1, \ldots, a_d$; $Tr(a, b)$ denotes the triangular distribution over (a, b) centered at $a + b/2$;

Table 1 Design and environmental variables, their notation, ranges, and distributions (χ^2_ν denotes the chi-square distribution with ν degrees of freedom; $N(\mu, \sigma^2)$ denotes the normal (Gaussian) distribution with mean μ and variance σ^2; $U(a,b)$ denotes the uniform distribution over the interval (a,b); $DU\{a_1,\ldots,a_d\}$ denotes the discrete uniform distribution over the values $\{a_1,\ldots,a_d\}$; $Tr(a,b)$ denotes the triangular distribution over (a,b) centered at $(a+b)/2$). D_r = 56 mm (targeted reamer diameter).

Parameter	Notation	Distribution	Range
		Engineering design variables	
Cup equatorial diameter (mm)	$D_{e,c}$	—	[56, 57, 58, 59]
Cup eccentricity (mm)	E_c	—	[0, 1, 2]
		Patient-dependent environmental variables	
Gait load magnitude (BW)	F_g	$0.0716\chi^2(47.1)$	[1.9, 5.5]
Gait load polar direction (deg)	θ_g	$N(34,(2.45)^2)$	[29.1, 38.9]
		Model uncertainty environmental variables	
Statistical density-modulus relation weight	W	Tr(0, 1)	[0, 1]
		Surgical environmental variables	
Cup penetration during insertion (mm)	p	$N(0.55,(0.29)^2)$	[0, 1.25]
Nominal reaming deviations at equator (mm)	$\delta_{r,a}$	$N(0.85\%D_r,(100\%\mu)^2)$	$[0.07, \mu+2\sigma]$
Nominal reaming deviations at pole (mm)	$\delta_{p,a}$	$N(-1.35\%D_r,(100\%\mu)^2)$	$[-1.75\%D_r, \min\{0.95\delta_{r,a}; \mu+2\sigma\}]$
Frequency of undulations (cross section)	ω_1	$U(4,9)$	[4, 9]
Frequency of undulations (transverse section)	ω_2	DU (4, 5, 6, 7)	[4, 5, 6, 7]
Peak amplitude of undulations (mm)	A_p	$U(0.85, 1.15)$	[0.85, 1.15]
Rate of decay of undulations	f	$U(0.85, 1.15)$	[0.85, 1.15]

*Example 1.3 (*Evolution of Fires in Enclosed Areas). Deterministic computer models are used in many areas of fire protection design including egress (exit) analysis. We describe one of the early "zone computer models" that is used to predict the fire conditions in an enclosed room. Cooper (1980) and Cooper and Stroup (1985) provided a mathematical model and its implementation in FORTRAN for describing the evolution of a fire in a single room with closed doors and windows that contains an object at some point below the ceiling that has been ignited. The room is assumed to contain a small leak at floor level to prevent the pressure from increasing in the room. The fire releases both energy and hot combustion by-products. The rate at which energy and the by-products are released is allowed to change with time. The by-products form a plume which rises towards the ceiling. As the plume rises, it draws in cool air, which decreases the plume's temperature and increases its volume flow rate. When the plume reaches the ceiling, it spreads out and forms a hot gas layer whose lower boundary descends with time. There is a relatively sharp inter-

face between the hot upper layer and the air in the lower part of the room, which in this model is considered to be at air temperature. The only interchange between the air in the lower part of the room and the hot upper layer is through the plume. The model used by these programs can therefore be described as a two "zone" model.

The Cooper and Stroup (1985) code is called ASET (Available Safe Egress Time). Walton (1985) implemented their model in BASIC, calling his computer code ASET-B; he intended his program to be used in the first generation of personal computers available at that time of its development. ASET-B is a compact, easy to run program that solves the same differential equations as ASET using a simpler numerical technique.

The *inputs* to ASET-B are

- the room ceiling height and the room floor area,
- the height of the burning object (fire source) above the floor,
- a heat loss fraction for the room (which depends on the insulation in the room, for example),
- a material-specific heat release rate, and
- the maximum time for the simulation.

The program *outputs* are the *temperature* of the hot smoke layer and its *distance* above the fire source as a function of time.

Since these early efforts, computer codes have been written to model wildfire evolution as well as fires in confined spaces. As typical examples of this work, we point to Lynn (1997) and Cooper (1997), respectively. The publications of the Building and Fire Research Laboratory of NIST can be found online at `http://fire.nist.gov/bfrlpubs/`. Finally, we mention the review article by Berk et al (2002) which describes statistical approaches for the evaluation of computer models for wildfires. Sahama and Diamond (2001) give a case study using the statistical methods introduced in Chapter 3 to analyze a set of 50 observations computed from the ASET-B model.

To provide a sense of the effect of each of these variables on the evolution of the fire, we *fixed* the heat release rate to correspond to fire material that constitutes a "semi-universal" fire; this heat release profile corresponds to a fire in a "fuel package consisting of a polyurethane mattress with sheets and fuels similar to wood cribs and polyurethane on pallets and commodities in paper cartons stacked on pallets." (Birk (1997)). Then we varied the remaining four factors using a "Sobol´ design" (Sobol´ designs are described in Section **??**). We computed the time, to the nearest second, for the fire to reach five feet above the burning fuel package, the fire source.

Scatterplots were constructed of each input versus the time required by hot smoke layer to reach five feet above the fire source. Only room area showed strong visual associations with the output; Figure 1.2 shows this scatterplot (see also Figure 3.10 for all four plots). This makes intuitive sense because more by-product is required to fill the top of a large room and hence, longer times are required until this layer reaches a point five feet above the fire source. The data from this example will be used later to illustrate several analysis methods.

*Example 1.4 (*Turbulent Mixing). BJW—references???

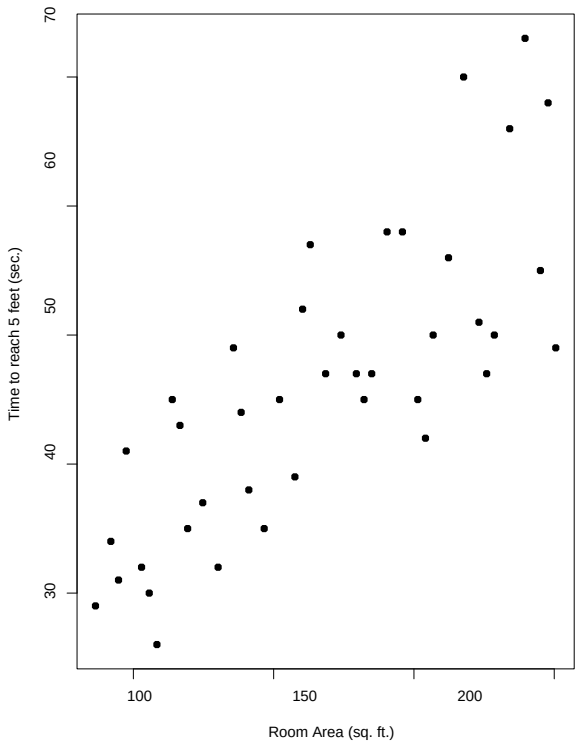


Fig. 1.2 Scatterplot of room area versus the time for the hot smoke layer to reach five feet above the fire source.

Example 1.5. "Structural performance is a function of both design and environmental variables. Hip resurfacing design variables include the stem-bone geometry and the extent of fixation, and environmental variables include variations in bone structure, bone modulus, and joint loading. Environmental variables vary from individual to individual as well as within the same individual over time, and can be treated stochastically to describe a population of interest."

see paper in literature

*Example 1.6 (*Qualitative Inputs).

Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. F. J. (2006), Building Surrogate Models Based on Detailed and Approximate Simulations,*ASME Journal of Mechanical Design*, **128**, 668–677.

*Example 1.7 (*Formation of Pockets in Sheet Metal). Montgomery and Truss (2001) discussed a computer model that determines the failure depth of symmetric rectangular pockets that are punched in automobile steel sheets; the failure depth is the

depth at which the sheet metal tears. Sheet metal, suitably formed in this manner, is used to fabricate many parts of automobiles. This application is but one of many examples of computer models used in the automotive industry.

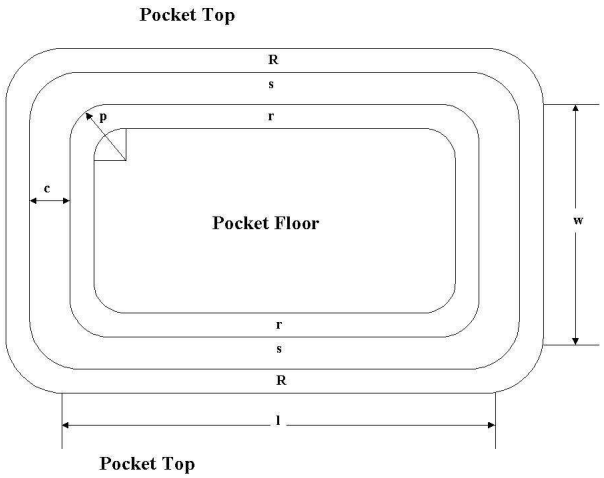


Fig. 1.3 Top view of the pocket formed by a punch and die operation. The floor of pocket is the innermost rectangle. The regions *R*, *s*, and *r* correspond to the similarly labeled regions in the side view.

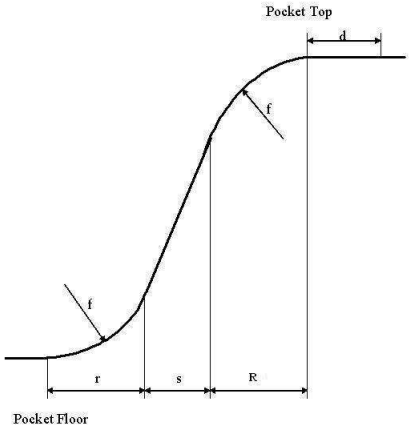


Fig. 1.4 Side view of part of a symmetric pocket formed by a punch and die operation. The angled side wall is created by the same fillet angle at the top by the die and at the bottom by the edge of the punch.

Rectangular pockets are formed in sheet metal by pressing the metal sheet with a punch which is a target shape into a conforming die. There are *six input variables* to the Montgomery and Truss (2001) code, all of which are engineering design variables. These variables can either be thought of as characteristics of the punch/die machine tool used to produce the pockets or, in most cases, as characteristics of the resulting pockets.

Five of the variables can easily be visualized in terms of the pocket geometry. In a top view of the pocket, Figure 1.3 illustrates the *length* l and the *width* w of the rectangular pocket (defined to omit the curved corner of the pocket). In a side view of the pocket, Figure 1.4 shows the *fillet radius* f, which is the radius of the circular path that the metal follows as it curves from the flat upper metal surface to the straight portion of the pocket wall; this region is denoted R in both the side and top views of the pocket. The same fillet radius is followed as the straight portion of the pocket wall curves in a circular manner to the pocket floor; this region is denoted by r in both views. Viewed from the top in Figure 1.3, the *clearance* is the horizontal distance c during which the angled side wall descends vertically to the pocket floor in Figure 1.4. In terms of the punch/die manufacturing tool, the clearance is the distance between the punch and the die when the punch is moved to its maximum depth within the die; the distance between the two tool components is constant. Lastly, the *punch plan view radius* p is illustrated in Figure 1.3. The *lock bead distance*, shown in Figure 1.4, is a distance d measured away from the pocket edge on the top metal surface; the machine tool does not allow stretching of the sheet metal beyond the distance d from the pocket edge.

To provide a sense of the (marginal) effect of each of these variables on the failure depth, we plotted the failure depth versus each of the six explanatory variables for the set of 234 runs analyzed by Montgomery and Truss (2001). Two of these scatterplots are shown in Figure 1.5; they are representative of the six marginal scatterplots. Five variables are only weakly related to failure depth and the panel in Figure 1.5 showing failure depth versus fillet radius is typical of these cases. One variable, clearance, shows a strong relationship with failure depth.

*Example 1.8 (***Other Examples).**

The purpose of this subsection is to sketch several applications of computer models that involve *large* numbers of input variables compared with the models described in Subsections 1.3–**??**. These examples will also serve to broaden the reader's appreciation of the many scientific and engineering applications of such models. Finally, as an additional source of motivating examples, we again remind the reader of Berk et al (2002), who report on a workshop that discussed computer models in four diverse areas: transportation flows, wildfire evolution, the spatio-temporal evolution of storms, and the spread of infectious diseases.

Booker et al (1997) describe a project to design an optimally shaped helicopter blade. While the thrust of their report concerned the development of an optimization algorithm that was used to minimize a function of the computer model outputs, their application is of interest because the engineering specification of the rotor required 31 design variables. Their specific objective function was a measure of the rotor vibration that combined the forces and moments on the rotor, these latter quantities

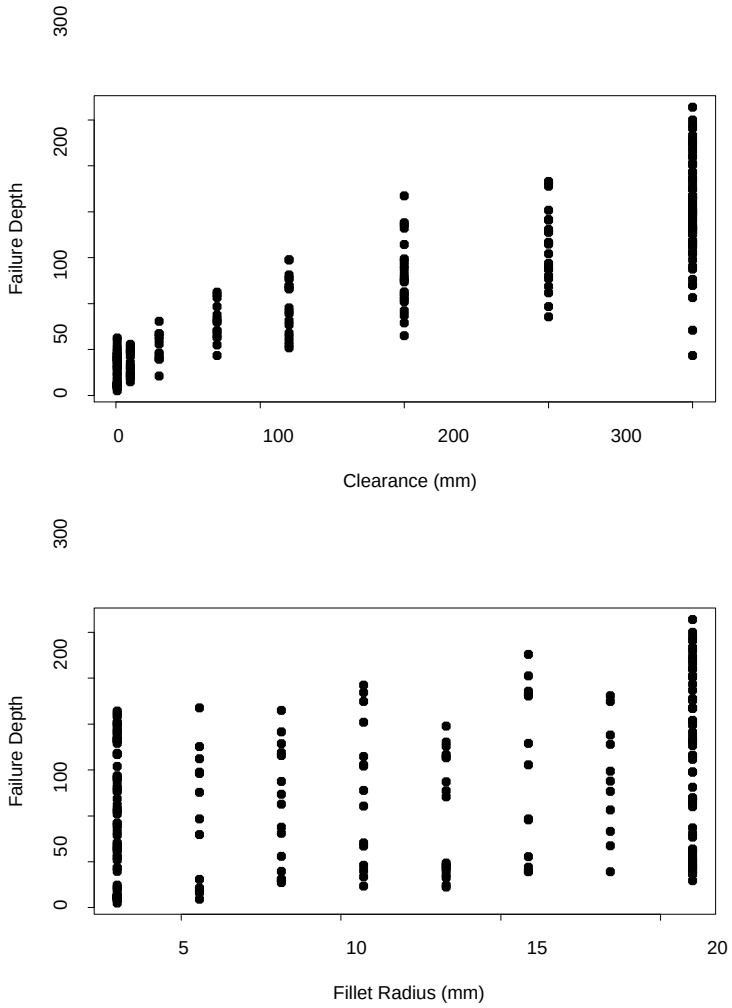


Fig. 1.5 Top panel—scatterplot of failure depth (millimeters) versus clearance for 234 runs of the computer code described in Subsection 1.3; Bottom panel—failure depth versus fillet radius for same data.

being calculated by the computer code. Each run of the computer code required very little time (10-15 minutes). However, the computer code provided a much less accurate solution of the mathematical equations that describe the forces and moments on the rotor than did the finite element code of Chang et al (2001) for their application. This circumstance raises the question of how a fast and slower, gold standard code for the same output can be combined. This issue will be addressed in Section **??**. Section **??** will provide other information about statistical approaches to combining information from multiple sources.

We end this section with a description of how computer codes have played an important role in public policy decision making. Lempert et al (2002) provide an example of such application. The objective is to contrast the effects of several national policies for curbing the effect of greenhouse gases based on an "integrative" model of the future that links the world economy to the population and to the state of the environment. The model they utilized, the so-called *Wonderland model*, quantifies the state of the future over a window of typically 100 years, using several measures, one of which is a "human development index." The model is integrative in that, for example, the pollution at a given time point depends on the user-specified innovation rate for the pollution abatement, the current population, the output per capita, environmental taxes, and other factors. The human development index is a weighted average of a discounted annual improvement of four quantities including, for example, the (net) output per capita. There are roughly 30 input variables. Different public policies can be specified by some of the inputs while the remaining input variables determine the different initial conditions and evolution rates.

1.3 Inputs and Outputs of Computer Experiments

The purpose of this section is to describe the types of inputs and outputs that occur in computer experiments. We frame this classification to the more familiar setting of physical experiments. Recall that the input variables (factors) to a physical experiment can be grouped as follows:

- Treatment inputs whose impact on the output response is of primary scientific interest,
- Blocking factors are variables that can represent the effect of different environmental conditions. Blocking factors can also describe natural environmental variation in the experimental units.
- Unrecognized factors whose presence can *bias* the observed input/output relationship unless randomization is used to to distribute their effect as "measurement error."

Furthermore, we view the output of a randomized physical experiment as being a noisy version of the true input/ouput relationship.

The inputs to computer experiments can similarly be classified according the role they play in the code with the exception that there are not "unrecognized" treatment factors that potentially cause bias. However, the output of computer experiments, being based on a mathematical model of the input/output relationship, *can exhibit* bias because either the mathematical model omits elements of the physics or biology that charactize this relationship and/or the numerical method used to implement the mathematical model lacks accuracy for some subdomain of the input space.

To fix ideas, we denote the output of the computer code by $\boldsymbol{x}$. There are four types of inputs that we distinguish, not all of which need be present in any application. The first type of input variable that we distinguish is a *control variable*. If

the output of the computer experiment is some performance measure of a product or process, then the control variables are those variables that can be set by an engineer or scientist to "control" the product or process. Some experimenters use the terms *engineering variables* or *manufacturing variables* rather than control variables. We use the generic notation $\boldsymbol{x}_c$ to denote control variables. Control variables are present in physical experiments as well as in many computer experiments.

need additional examples based on the final models that we use As examples of control variables, we mention the dimensions b and d of the bullet tip prosthesis illustrated in Figure **??** (see Subsection **??**). Another example is given by Box and Jones (1992) in the context of a hypothetical physical experiment to formulate ("design") the recipe for a cake. The goal was to determine the amounts of three baking variables to produce the best tasting cake: *flour*, *shortening*, and *egg*; hence, these are control variables. The physical experiment considered two additional variables that also affect the taste of the final product: the *time* at which the cake is baked and the *oven temperature*. Both of the latter variables are specified in the baking recipe on the cake box. However, not all bakers follow the box directions exactly and even if they attempt to follow them precisely, ovens can have true temperatures that differ from their nominal settings and timers can be systematically off or be unheard when they ring.

again, we need additional examples based on the final models that we use The variables, baking time and oven temperature, are examples of *environmental variables*, a second type of variable that can be present in both computer and physical experiments. In general, environmental variables affect the output $y(\cdot)$ but depend on the specific user or on the environment at the time the item is used. Environmental variables are sometimes called *noise variables*. We use the notation $\boldsymbol{x}_e$ to denote the vector of environmental variables for a given problem. In practice, we typically regard environmental variables as random with a distribution that is known or unknown. To emphasize situations where we regard the environmental variables as random, we use the notation $\boldsymbol{X}_e$. The hip prosthesis example of Chang et al (1999) illustrates a computer experiment with environmental variables (see Subsection **??**); both of their outputs depended on the *magnitude* and *direction* of the force exerted on the head of the prosthesis. These two variables were patient specific and depended on body mass and activity. They were treated as having a given distribution that was characteristic of a given population.

In addition to control and environmental variables, there is a third category of input variable that sometimes occurs. This third type of input variable describes the uncertainty in the mathematical modeling that relates other inputs to output(s). As an example, O'Hagan et al (1999) consider a model for metabolic processing of U^{235} that involves various rate constants for elementary conversion processes that must be known in order to specify the overall metabolic process. In some cases, such elementary rate constants may have values that are unknown or possibly there is a known (subjective) distribution that describes their values. We call these variables *model* variables and denote them by $\boldsymbol{x}_m$. In a classical statistical setting we would call model variables "model parameters" because we use the results of a physical

experiment, the ultimate reality, to estimate their values. Some authors call model variables "tuning parameters."

1.4 Objectives of Experimentation

- Prediction of the output of the *computer code* at a point, over a region, (eg., the entire input domain)
- Calibration of the output of a computer simulation model to physical experimental data
- Set tuning parameters based on physical experimental data
- Prediction of output at a given (set) of inputs based on the "true" input-ouput relationship x
- Assessment of the uncertainty in the predicted output (computer code; true input-output relationship)
- Prediction of functionals of computer code output–means, percentiles (incuding extrema), IQR, all inputs that produce a given value of the output.

1.4.1 Introduction

This chapter describes some common research objectives that when one is employing a computer experiment, either alone or in conjunction with a physical experiment. Initially we consider the case of a single real-valued output $y(\cdot)$ that is to be evaluated at input training sites $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. We let $\widehat{y}(\boldsymbol{x})$ denote a generic predictor of $y(\boldsymbol{x})$ and consider goals for two types of inputs. In the first setting, referred to as a *homogeneous-input*, all inputs of $\boldsymbol{x}$ are of the type, i.e., either control variables *or* environmental variables *or* model variables. In the second setting, referred to as a *mixed-input*, $\boldsymbol{x}$ contains at least two of the three different types of input variables: control, environmental, and model. Finally, in Subsection 1.4.4, we outline some typical goals when there are several outputs. In all cases there can be both "local" and "global" goals that may be of interest.

The following section describes several fundamental goals for computer experiments depending on which types of variables are present and the number of responses that the code produces. For example, if the code produces a single real-valued response that depends on control and environmental variables, then we use the notation $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ to emphasize that the propagation of uncertainty in the environmental variables $\boldsymbol{X}_e$ must be accounted for. In some cases there may be multiple computer codes that produce related responses $y_1(\cdot), \ldots, y_m(\cdot)$ which either represent competing responses or correspond to "better" and "worse" approximations to the response. For example, if there are multiple finite element analysis codes based on greater or fewer node/edge combinations to represent the *same* phenomenon, then one might hope to combine the responses to improve prediction. Another al-

ternative is that $y_1(\cdot)$ represents the primary object of interest while $y_2(\cdot), \ldots, y_m(\cdot)$ represent "related information"; for example, this would be the case if the code produced a response *and* vector of first partial derivatives. A third possibility is when the $y_i(\cdot)$ represent competing objectives; in this case, the goal might be to optimize one response subject to minimum performance standards on the remaining ones.

Following the description of experimental goals, we summarize the basic issues in modeling computer output. Then we will be prepared to begin Chapter 3 on the first of the two basic issues considered in this book, that of predicting $y(\cdot)$ at (a new) input $\boldsymbol{x}_0$ based on training data $(\boldsymbol{x}_1, y(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_n, y(\boldsymbol{x}_n))$. Chapter 4 will address the second issue, the design problem of choosing the input sites at which the computer model should be run.

1.4.2 Research Goals for Homogeneous-Input Codes

First, suppose that $\boldsymbol{x}$ consists exclusively of control variables, i.e., $\boldsymbol{x} = \boldsymbol{x}_c$. In this case one important objective is to predict $y(\boldsymbol{x})$ "well" for all $\boldsymbol{x}$ in some domain $\mathcal{X}$. There have been several criteria used to measure the quality of the prediction in an "overall" sense. One appealing intuitive basis for judging the predictor $\widehat{y}(\boldsymbol{x})$ is its *integrated squared error*

$$\int_{\mathcal{X}} [\widehat{y}(\boldsymbol{x}) - y(\boldsymbol{x})]^2 \, w(\boldsymbol{x}) \, d\boldsymbol{x}, \tag{1.4.1}$$

where $w(\boldsymbol{x})$ is a nonnegative weight function that quantifies the importance of each value in $\mathcal{X}$. For example, $w(\boldsymbol{x}) = 1$ weights all parts of $\mathcal{X}$ equally while $w(\boldsymbol{x}) = I_{\mathcal{A}}(\boldsymbol{x})$, the indicator function of the set $\mathcal{A} \subset \mathcal{X}$, ignores the complement of $\mathcal{A}$ and weights all points in $\mathcal{A}$ equally.

Unfortunately, (1.4.1) cannot be calculated because $y(\boldsymbol{x})$ is unknown. However, later in Chapter 5 we will replace $[\widehat{y}(\boldsymbol{x}) - y(\boldsymbol{x})]^2$ by a posterior mean squared value computed under a certain "prior" model for $y(\boldsymbol{x})$ and obtain a quantity that can be computed (see Section 5.2 for methods of designing computer experiments in such settings).

The problem of predicting $y(\cdot)$ well over a region can be thought of as a global objective. In contrast, more local goals focus on finding "interesting" parts of the input space $\mathcal{X}$. An example of such a goal is to identify (any) $\boldsymbol{x}$, where $y(\boldsymbol{x})$ equals some target value. Suppose

$$\mathcal{L}(t_0) = \{\boldsymbol{x} \in \mathcal{X} \mid y(\boldsymbol{x}) = t_0\}$$

denotes the "level set" of input values where $y(\cdot)$ attains a target value t_0. Then we wish to determine any input $\boldsymbol{x}$ where $y(\cdot)$ attains the target level, i.e., any $\boldsymbol{x} \in \mathcal{L}(t_0)$. Another example of a local goal is to find extreme values of $y(\cdot)$. Suppose

$$\mathcal{M} = \left\{\boldsymbol{x} \in \mathcal{X} \mid y(\boldsymbol{x}) \geq y(\boldsymbol{x}^\star) \text{ for all } \boldsymbol{x}^\star \in \mathcal{X}\right\} \equiv \arg\max y(\cdot)$$

is the set of all arguments that attain the global maximum of $y(\boldsymbol{x})$. Then an analog of the level set problem is to find a set of inputs that attain the overall maximum, i.e., to determine any $\boldsymbol{x} \in \mathcal{M}$. The problem of finding global optima of computer code output has been the subject of much investigation (**?**, Bernardo et al (1992), Mockus et al (1997), Jones et al (1998), **?**).

There is a large literature on homogeneous-input problems when $\boldsymbol{x}$ depends only on environmental variables. Perhaps the most frequently occurring application is when the environmental variables are random inputs with a known distribution and the goal is determine how the variability in the inputs is transmitted through the computer code. In this case we write $\boldsymbol{x} = \boldsymbol{X}_e$ using upper case notation to emphasize that the inputs are to be treated as random variables and the goal is that of finding the distribution of $y(\boldsymbol{X}_e)$. This problem is sometimes called uncertainty analysis (**?**, **?**, Helton (1993), O'Hagan and Haylock (1997), and O'Hagan et al (1999) are examples of such papers). Also in this spirit, McKay et al (1979) introduced the class of Latin hypercube designs for choosing the training sites $\boldsymbol{X}_e$ at which to evaluate the code when the problem is to predict the *mean* of the $y(\boldsymbol{X}_e)$ distribution, $E\{y(\boldsymbol{X}_e)\}$. The theoretical study of Latin hypercube designs has established a host of asymptotic and empirical properties of estimators based on them (**?**, **?**, **?**, **?**, **?**) and enhancements of such designs (**?**, **?**, **?**, **?**, **?**).

The third possibility for homogeneous-input is when $y(\cdot)$ depends only on *model* variables, $\boldsymbol{x} = \boldsymbol{x}_m$. Typically in such a case, the computer code is meant to describe the output of a physical experiment but the mathematical modeling of the phenomenon involves *unknown* parameters, often unknown rate or physical constants. In this situation the most frequently discussed objective in the computer experiments literature is that of *calibration*. Calibration is possible when the results of a physical experiment are available whose response is the physical phenomenon that the computer code is meant to model. The goal is to choose the model variables $\boldsymbol{x}_m$ so that the computer output best matches the output from the physical experiment (examples are **?**, **?**, **?**, and the references therein).

1.4.3 Research Goals for Mixed-Inputs

Mixed-inputs can arise from any combination of control, environmental, and model variables. We focus on what is arguably the most interesting of these cases, that of $\boldsymbol{x}$ consisting of both control and environmental variables. In the problems described below, the environmental variables will be assumed to have a known distribution, i.e., $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{X}_e)$ where $\boldsymbol{X}_e$ has a known distribution. There are related problems for other mixed-input cases.

In this case, for each $\boldsymbol{x}_c$, $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ is a random variable with a distribution that is induced by the distribution of $\boldsymbol{X}_e$. The $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ distribution can change as $\boldsymbol{x}_c$ changes. As discussed above for the homogeneous-input case $\boldsymbol{x} = \boldsymbol{X}_e$, attention is typically focused on some specific aspect of this induced distribution. For example, recall the study of Chang et al (1999) for designing a hip prosthesis that was in-

troduced in Section 2.1. In their situation, $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ was the maximum strain at the bone-implant interface; it depended on the engineering variables, $\boldsymbol{x}_c$, that specified the geometry of the device, and on the environmental variables $\boldsymbol{x}_e$, consisting of the force applied to the hip joint and the angle at which it is applied. Chang et al (1999) considered the problem of finding engineering designs $\boldsymbol{x}_c$ that minimized the *mean strain* where the mean is taken with respect to the environmental variables. Of course, this is equivalent to maximizing the negative of the mean strain and for definiteness, we describe all optimization problems below as those of finding maxima of mean functions.

To describe this, and related goals, in a formal fashion, let

$$\mu(\boldsymbol{x}_c) = E\{y(\boldsymbol{x}_c, \boldsymbol{X}_e)\} \tag{1.4.2}$$

denote the mean of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ with respect to the distribution of $\underset{\sim}{\mathrm{X}}_e$. Similarly define (implicitly) the upper alpha quantile of the distribution of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$, denoted by $\xi^\alpha = \xi^\alpha(\boldsymbol{x}_c)$, as

$$P\{y(\boldsymbol{x}_c, \boldsymbol{X}_e) \geq \xi^\alpha\} = \alpha$$

(assuming for simplicity that there is a unique such upper α quantile). For example, the notation $\xi^{.5}(\boldsymbol{x}_c)$ denotes the *median* of the distribution of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$, which is a natural competitor of the mean, $\mu(\boldsymbol{x}_c)$, when $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ has a skewed distribution.

With this setup, it is possible to describe analogs for $\mu(\boldsymbol{x}_c)$, of the three goals stated above for $y(\boldsymbol{x}_c)$. If the distribution of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ is skewed, the objectives described below might better be stated in terms of the median $\xi^{.5}(\boldsymbol{x}_c)$. Let $\widehat{\mu}(\boldsymbol{x}_c)$ denote a generic predictor of $\mu(\boldsymbol{x}_c)$. The analog of predicting $y(\cdot)$ well over its domain is to predict $\mu(\boldsymbol{x}_c)$ well over the control variable domain in the sense of minimizing

$$\int [\mu(\boldsymbol{x}_c) - \widehat{\mu}(\boldsymbol{x}_c)]^2 \; w(\boldsymbol{x}_c)\, d\boldsymbol{x}_c. \tag{1.4.3}$$

To solve this problem, one must not only choose a particular predictor $\widehat{\mu}(\boldsymbol{x}_c)$ of $\mu(\boldsymbol{x}_c)$, but also the set of input training sites $(\boldsymbol{x}_c, \boldsymbol{x}_e)$ on which to base the predictor. As in the case of (1.4.1), the criterion (1.4.3) cannot be computed, but a Bayesian analog that has a computable mean will be introduced in Chapter 5.

The parallel of the problem of finding a control variable that maximizes $y(\boldsymbol{x}_c)$ is that of determining an $\boldsymbol{x}_c$ that maximizes the mean output $\mu(\boldsymbol{x}_c)$, i.e., finding an $\boldsymbol{x}_c^M$ that satisfies

$$\mu(\boldsymbol{x}_c^M) = \max_{\boldsymbol{x}_c} \mu(\boldsymbol{x}_c). \tag{1.4.4}$$

Similarly, a parallel to the problem of finding $\boldsymbol{x}_c$ to attain target $y(\cdot)$ values is straightforward to formulate for $\mu(\boldsymbol{x}_c)$.

Additional challenges occur in those applications when the distribution of $\boldsymbol{X}_e$ is not known precisely. To illustrate the consequences of such a situation, suppose that $\boldsymbol{x}_c^M$ maximizes $E_{G^N}\{y(\boldsymbol{x}_c, \boldsymbol{X}_e)\}$ for a given *nominal* $\boldsymbol{X}_e$ distribution, G^N. Now suppose, instead, that $G \neq G^N$ is the true $\boldsymbol{X}_e$ distribution. If

$$E_G\{y(\boldsymbol{x}_c^M, \boldsymbol{X}_e)\} \ll \max_{\boldsymbol{x}_c} E_G\{y(\boldsymbol{x}_c, \boldsymbol{X}_e)\}, \tag{1.4.5}$$

then $\boldsymbol{x}_c^M$ is substantially inferior to any $\boldsymbol{x}_c^\star$ that achieves the maximum in the right-hand side of (1.4.5). From this perspective, a control variable $\boldsymbol{x}_c$ can be thought of as being "robust" against misspecification of the $\boldsymbol{X}_e$ distribution if $\boldsymbol{x}_c$ comes close to maximizing the mean over the nominal $\boldsymbol{X}_e$ distribution and $\boldsymbol{x}_c$ is never far from achieving the maximum on the right-hand side of (1.4.5) for alternative $\boldsymbol{X}_e$ distributions, G. There are several formal methods of defining a robust $\boldsymbol{x}_c$ that heuristically embody this idea.

The classical method of defining a robust $\boldsymbol{x}_c$ is by a minimax approach (**?**). Given a set $\mathcal{G}$ of possible environmental variable distributions (that includes a "central," nominal distribution G^N), let

$$\mu(\boldsymbol{x}_c, G) = E_G\{y(\boldsymbol{x}_c, \boldsymbol{X}_e)\}$$

denote the mean of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ when $\boldsymbol{X}_e$ has distribution $G \in \mathcal{G}$. Then

$$\min_{G\in\mathcal{G}} \mu(\boldsymbol{x}_c, G)$$

is the smallest mean value for $y(\boldsymbol{x}_c, \cdot)$ that is possible when $\boldsymbol{X}_e$ distributions come from $\mathcal{G}$. We say $\boldsymbol{x}_c^{\mathcal{G}}$ is a $\mathcal{G}$-robust design provided

$$\min_{G\in\mathcal{G}} \mu(\boldsymbol{x}_c^{\mathcal{G}}, G) = \max_{\boldsymbol{x}_c} \min_{G\in\mathcal{G}} \mu(\boldsymbol{x}_c, G).$$

Philosophically, $\mathcal{G}$-robust designs can be criticized because they are *pessimistic*; $\boldsymbol{x}_c^{\mathcal{G}}$ maximizes a worst-case scenario for the mean of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$. In addition, one is faced with the challenge of specifying a meaningful $\mathcal{G}$. Finally, there can be substantial computational problems determining $\mathcal{G}$-robust designs.

An alternative definition, Bayesian in spirit, assumes that it is possible to place a distribution $\pi(\cdot)$ on the $G \in \mathcal{G}$ where $\mathcal{G}$ is the known set of environmental distributions. In the most straightforward case, the distributions in $\mathcal{G}$ can be characterized by a finite vector of parameters $\boldsymbol{\theta}$. Suppose that $\pi(\cdot)$ is a prior density over the $\boldsymbol{\theta}$ values. We define $\boldsymbol{x}_c^{\pi}$ to be $\pi(\cdot)$-robust provided

$$\int \mu(\boldsymbol{x}_c^{\pi}, \boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta} = \max_{\boldsymbol{x}_c} \int \mu(\boldsymbol{x}_c, \boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}.$$

A critique of $\pi(\cdot)$-robust designs is that, in addition to the difficulty of specifying a meaningful $\mathcal{G}$, one must also determine a prior $\pi(\cdot)$. However, $\pi(\cdot)$-robust designs are typically easier to compute than $\mathcal{G}$-robust designs.

A third, more heuristic definition of a robust $\boldsymbol{x}_c$ requires only a nominal $\boldsymbol{X}_e$ distribution, and neither a class $\mathcal{G}$ of alternative distributions nor a prior $\pi(\cdot)$ need be specified. This last definition is based on the following observation. Suppose that for a given $\boldsymbol{x}_c$, $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is (fairly) "flat" in $\boldsymbol{x}_e$; then the mean of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ will "tend" to be independent of the choice of $\boldsymbol{X}_e$ distribution. Assuming that we desire the mean

$\mu(\boldsymbol{x}_c)$ of $y(\cdot)$ under the nominal distribution to be large, a robust $\boldsymbol{x}_c$ maximizes $\mu(\boldsymbol{x}_c)$ among those $\boldsymbol{x}_c$ for which $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is flat. We call such an $\boldsymbol{x}_c$ a M-robust design. To define this notion formally, suppose that each component of $\boldsymbol{X}_e$ has a bounded support; the $\boldsymbol{X}_e$ has support on a bounded hyper-rectangle, say $\mathsf{X}_i[a_i, b_i]$. Let

$$\sigma^2(\boldsymbol{x}_c) = \frac{1}{\Pi_i(b_i - a_i)} \int y^2(\boldsymbol{x}_c, \boldsymbol{x}_e)\, d\boldsymbol{x}_e - \left(\frac{1}{\Pi_i(b_i - a_i)} \int y(\boldsymbol{x}_c, \boldsymbol{x}_e)\, d\boldsymbol{x}_e \right)^2$$

be the "variance" of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ with respect to a uniform distribution on $\boldsymbol{X}_e$. We define $\boldsymbol{x}_c^M$ to be M-robust provided $\boldsymbol{x}_c^M$ maximizes

$$\begin{gathered} \mu(\boldsymbol{x}_c) \\ \text{subject to} \\ \sigma^2(\boldsymbol{x}_c) \le B. \end{gathered}$$

Here B is an absolute bound on the variability of $y(\boldsymbol{x}_c, \cdot)$. An alternative, natural constraint is

$$\sigma^2(\boldsymbol{x}_c) \le \max_{\boldsymbol{x}_c^\star \in X_c} \sigma^2(\boldsymbol{x}_c^\star) \times B,$$

where B is now a relative bound that is < 1. Because $B < 1$, this second formulation has the theoretical advantage that the feasible region is always nonempty whereas in the former specification one may *desire* that the variance be no greater than a certain bound B, but there need not exist control variables $\boldsymbol{x}_c$ that achieve this target value. Using the relative constraint has the computational disadvantage that the maximum variance must be determined. Alternatively, and perhaps more in keeping with the quality control concept of having a "target" mean, we define $\boldsymbol{x}_c^V$ to be *V-robust* if it minimizes $\sigma^2(\boldsymbol{x}_c)$ subject to a constraint on $\mu(\boldsymbol{x}_c)$. **?** discuss the sequential design of computer experiments to find M-robust and V-robust choices of control variables.

1.4.4 Experiments with Multiple Outputs

To fix ideas, suppose that $y_1(\cdot), \ldots, y_m(\cdot)$ are the computed outputs. There are at least three different settings that lead to such a situation. First, the outputs can represent multiple codes for the same quantity; for example, **?** study multiple codes that represent coarser and finer finite element descriptions for the same response.

A second setting that leads to multiple outputs is when the $y_i(\cdot)$ are *competing* responses from *different* codes; in prosthesis design we desire to maximize the strain at the prosthesis–bone interface so that bone resorption does not occur and simultaneously minimize (or at least bound) the side to side "toggling" of the implant. The two objectives, maximizing strain and minimizing toggling, represent competing design goals. A third setting that leads to multiple outputs is when a single code produces $y_i(\cdot)$ that are related to one another. As an example, **?** and **?** consider the prediction of $y(\boldsymbol{x})$ for codes that produce $y(\cdot)$ *and* all its first partial derivatives for each input site $\boldsymbol{x}$. Thus we regard $y_1(\boldsymbol{x}) = y(\boldsymbol{x})$, the original output, and $y_2(\boldsymbol{x}), \ldots,$

$y_m(\boldsymbol{x})$ as the values of the partial derivatives of $y(\boldsymbol{x})$ with respect to each component of $\boldsymbol{x}$. These derivatives provide auxiliary information that permits more precise prediction of $y(\boldsymbol{x})$ than that based on $y(\cdot)$ alone.

The modeling of multiple $y_i(\cdot)$ depends on which scenario above holds, as do the possible scientific or engineering objectives. For example, when $y_2(\boldsymbol{x}), \ldots, y_m(\boldsymbol{x})$ represent auxiliary information about $y_1(\boldsymbol{x})$, the goal might simply be to use the additional information to better predict $y_1(\cdot)$. To continue the example introduced in the previous paragraph, **?** and **?** show how to model the output from codes that produce a response $y(\cdot)$ and the partial derivatives of $y(\cdot)$. They then use these models to derive (empirical) best linear unbiased predictors of $y(\cdot)$ at new sites $\boldsymbol{x}_0$ based on all the responses. See Section 3.5 for a discussion of modeling multiple responses.

Now consider the scenario where $\boldsymbol{x} = \boldsymbol{x}_c$, $y_1(\cdot)$ is the response of primary interest, and $y_2(\cdot), \ldots, y_m(\cdot)$ are competing objectives. Then we can define a feasible region of $\boldsymbol{x}_c$ values by requiring minimal performance standards for $y_2(\boldsymbol{x}_c), \ldots, y_m(\boldsymbol{x}_c)$. Formally, an analog of the problem of minimizing $y(\cdot)$ is

$$\begin{aligned} &\text{minimize} \quad y_1(\boldsymbol{x}_c) \\ &\text{subject to} \\ &y_2(\boldsymbol{x}_c) \geq M_2 \\ &\vdots \\ &y_m(\boldsymbol{x}_c) \geq M_m. \end{aligned}$$

Here M_i is the lower bound on the performance of $y_i(\cdot)$ that is acceptable. If in addition to control variables, $\boldsymbol{x}$ also contains environmental variables, then we can replace each $y_i(\boldsymbol{x}_c)$ above with $\mu_i(\boldsymbol{x}_c) = E\{y_i(\boldsymbol{x}_c, \boldsymbol{X}_e)\}$. In cases where $\boldsymbol{x} = \boldsymbol{x}_e$, a typical objective is to find the joint distribution of $(y_1(\boldsymbol{X}_e), \ldots, y_m(\boldsymbol{X}_e))$ or, even simpler, that of predicting the mean vector $(E\{y_1(\boldsymbol{X}_e)\}, \ldots, E\{y_m(\boldsymbol{X}_e)\})$.

Lastly, if the $y_i(\cdot)$ represent the outputs of *different* codes of varying accuracy for the same response, then a typical goal is to combine information from the various outputs to better predict the true response. Specification of this goal depends on identifying the "true" response; we postpone a discussion of this idea until we discuss modeling multiple response output in Section 3.5.

1.5 Organization of the Book

The remainder of the book is organized as follows. Chapter 2 outlines the conceptual framework for thinking about the design and analysis of computer experiments. This includes a classification of the types of input variables that can affect the output of a computer code, a summary of research goals when conducting a computer experiment, and an introduction to Gaussian random field models as a description of the output from a computer experiment. Using the Gaussian random model, Chapter 3 introduces methods that can be used for predicting the output of computer codes based on training data, plus methods of assessing the uncertainty in these predictions. The chapter compares the prediction methods and presents our recommenda-

tions concerning their use. Chapter **??** introduces several additional topics including the use of predictive distributions and prediction based on multiple outputs. Chapter 4 and Chapter 5 are concerned with experimental design, i.e., the selection of the input sites at which to run code. Chapter 4 begins this discussion by considering space-filling designs, meaning designs that spread observations evenly throughout the input region. Among the designs examined are those based on simple random sampling, those based on stratified random sampling, Latin hypercube designs, orthogonal arrays, distance-based designs, uniform designs, and designs that combine multiple criteria. Sobol´ designs, grid, and lattice designs are briefly mentioned at the end of the chapter. Chapter 5 considers designs based on statistical criteria such as maximum entropy and mean squared error of prediction. This chapter also considers sequential strategies for designing computer experiments when the goal is to optimize the output. Finally, Chapter **??** discusses some issues of validation of computer experiments using physical and other experiments as well as sensitivity analysis.

PErK software allows readers to fit most of the models discussed in this book. PErK, written in C and using the freely available GSL C software library, can be obtained at either

```
http://www.stat.ohio.edu/~comp_exp
http://www.springer-ny.com/
```

Appendix **??** describes the syntax used by PErK and provides examples of its use to fit a variety of models.

We have added Notes sections at the end of the chapters that describe developments beyond the basic ones given in this book. We recognize that the limitation in implementing many of the procedures we describe in this text is the availability of certain tabled constants. Thus, in addition to providing tables to implement the procedures, we also provide a number of FORTRAN programs to supplement our tables. In addition, we describe other public domain programs valuable for implementing certain selection, screening and simultaneous confidence interval procedures and state how to obtain them.

Chapter 2
Stochastic Models for Computer Output

2.1 Introduction

Recall from Chapter 1 that we will let input $\boldsymbol{x}$ denote a generic input to our computer experiment and $y(\boldsymbol{x})$ denote the associated output. The purpose of this chapter is to introduce several classes of random function models for $y(\boldsymbol{x})$ that will serve as the fundamental building blocks of the interpolators, experimental designs, calibration and tuning methodologies that will be introduced in later chapters. Some readers will regard our viewpoint about such processes as being Bayesian; however, computer experiments represent a highly nonparametric setting and careful eliciting a prior for the output of a black box code is much more difficult than, say, eliciting the prior for the mean output of a regression or the rate of change of the mean corresponding to a unit change in a regression function (see Oakley (2002) and Reese et al (2000) for advice and case-studies about the formation of prior distributions). Thus other readers may find it more intuitive to think about the process model assumption as an extension of the more familiar regression model.

Our viewpoint in this discussion is Bayesian because this approach is philosophically more satisfying in, for example, its interpretation of estimated standard errors for predictors. From the Bayesian viewpoint, such quantities refer to model uncertainty (informed by the training data). However our approach is not dogmatic; we *do* attempt to control the characteristics of the functions produced by our priors, but *do not* rigidly believe them. Instead, our goal is to choose flexible priors that are capable of producing many shapes for $y(\cdot)$ and then let the Bayesian machinery allow the data to direct the details of the prediction process. We also note that computer experiments are not alone in their use of Bayesian prediction methodology to analyze high-dimensional, highly correlated data. Many other scientific fields produce such data, albeit usually with measurement error. The statistical analyses used in geostatistics (Matheron (1963), Journel and Huijbregts (1979)), environmental statistics and disease mapping (Ripley (1981), Cressie (1993)), global optimization (Mockus et al (1997)), and statistical learning (Hastie et al (2001)) are based on the

Bayesian philosophy. Hence many of the methodologies discussed in their literatures are also relevant here.

In the following we let $\mathcal{X}$ denote the domain or input space for the unknown output function $y(\cdot)$; $y(\cdot)$ is regarded to be a draw from a random function ("stochastic process" or simply "process") which is denoted by $Y(\cdot)$. With the exception of the essentially philosophical statements in the following paragraph, we will adopt a pragmatic viewpoint in discussing stochastic process models rather than a measure theoretic one. Conceptually, a random function should be thought of as a mapping from elements of a sample space of outcomes, say Ω, to a given set of functions, just as random variables are mappings from a set Ω of elementary outcomes to the real numbers. It will occasionally add clarity to our discussion to make this explicit by writing $y(\boldsymbol{x}) = Y(\boldsymbol{x}, \omega)$ to be a *particular* function from $\mathcal{X}$ to $\mathbb{R}^1$, where $\omega \in \Omega$ is a specific element in the sample space. Sometimes we refer to $y(\cdot, \omega)$ as a *draw* from the random function $Y(\cdot)$ or as a *sample path* (in $\mathcal{X}$) of the random function. The introduction of the underlying sample space Ω helps clarify ideas when discussing the smoothness properties of functions drawn from $Y(\cdot)$.

We begin this discussion with a simple example to illustrate a random mechanism for generating functions $y(\cdot)$.

Example 2.1. Suppose that we generate $y(x)$ on $[-1, +1]$ by the mechanism

$$Y(x) = b_0 + b_1 x + b_2 x^2, \tag{2.1.1}$$

where b_0, b_1, and b_2 are independent with $b_i \sim N(0, \sigma_i^2)$ for $i = 1, 2, 3$. Functions drawn from $Y(x)$ are simple to visualize. Every realization $y(\cdot)$ is a quadratic equation ($P\{b_2 = 0\} = 0$) that is symmetric about an axis other than the y-axis (symmetry about the y-axis occurs if and only if $b_1 = 0$ and $P\{b_1 = 0\} = 0$). The quadratic is convex with probability $1/2$ and it is concave with probability $1/2$ (because $P\{b_2 > 0\} = 1/2 = P\{b_2 < 0\}$). Figure 2.1 illustrates ten outcomes from this random function when $\sigma_0^2 = \sigma_1^2 = \sigma_2^2 = 1.0$.

For any $x \in [-1, +1]$ the draws from (2.1.1) have mean zero, i.e.,

$$\begin{aligned} E\{Y(x)\} &= E\{b_0 + b_1 x + b_2 x^2\} \\ &= E\{b_0\} + E\{b_1\} \times x + E\{b_2\} \times x^2 \\ &= 0 + 0 \times x + 0 \times x^2 = 0. \end{aligned} \tag{2.1.2}$$

Equation (2.1.2) says that for any x, the mean of $Y(x)$ is *zero* over many drawings of the coefficients (b_0, b_1, b_2); this is true because each regression coefficient is independent and centered at the origin so that each regression term is positive and negative with probability 1/2 and thus their sum, $Y(x)$, is also positive and negative with probability 1/2.

For any $x \in [-1, +1]$ the pointwise variance of $Y(x)$ is

$$\begin{aligned} \mathrm{Var}\{Y(x)\} &= E\left\{\left(b_0 + b_1 x + b_2 x^2\right)\left(b_0 + b_1 x + b_2 x^2\right)\right\} \\ &= \sigma_0^2 + \sigma_1^2 x^2 + \sigma_2^2 x^4 \geq 0. \end{aligned}$$

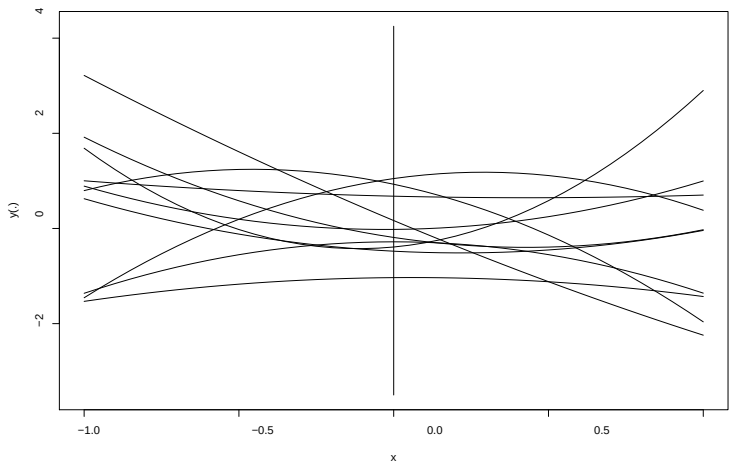


Fig. 2.1 Ten draws from the random function $Y(x) = b_0 + b_1 x + b_2 x^2$ on $[-1, +1]$, where b_0, b_1, and b_2 are independent and identically $N(0, 1.0)$ distributed.

The values of $Y(x_1)$ and $Y(x_2)$ at $x_1, x_2 \in [-1, +1]$ are related, as can be seen from

$$\begin{aligned} \text{Cov}\{Y(x_1), Y(x_2)\} &= E\left\{\left(b_0 + b_1 x_1 + b_2 x_1^2\right)\left(b_0 + b_1 x_2 + b_2 x_2^2\right)\right\} \\ &= \sigma_0^2 + \sigma_1^2 x_1 x_2 + \sigma_2^2 x_1^2 x_2^2. \end{aligned} \tag{2.1.3}$$

This covariance can be positive or negative. The sign of the covariance of $Y(x_1)$ and $Y(x_2)$ can intuitively be explained as follows. The covariance formula (2.1.3) is clearly positive for any x_1 and x_2 when both are positive or both are negative. Intuitively this is true because over many drawings of (b_0, b_1, b_2), x_1 and x_2 both tend to be on the same side of the axis of symmetry of the quadratic and thus $Y(x_1)$ and $Y(x_2)$ increase or decrease together. The covariance formula *can* be negative if x_1 and x_2 are on the *opposite* sides of the origin *and* σ_1^2 dominates σ_0^2 and σ_2^2 (algebraically, the middle term in (2.1.3) is negative and can exceed the sum of the other two terms). Intuitively, one circumstance where this occurs is if σ_0^2 is small (meaning the curves tend to go "near" $(0, 0)$), *and* σ_2^2 is small (the curves tend to be linear near the origin), *and* σ_1^2 is large; in this case, the draws fluctuate between those with large positive slopes and those with large negative slopes, implying that $Y(x_1)$ and $Y(x_2)$ tend to have the opposite sign over the draws.

Because linear combinations of a fixed set of independent normal random variables have the normal distribution, the simple model (2.1.1) for $Y(\cdot)$ satisfies: for each $L > 1$ and any choice of $x_1, \ldots, x_L \in \mathcal{X}$, the vector $(Y(x_1), \ldots, Y(x_L))$ is multivariate normally distributed. (See Appendix B for a review of the multivariate normal distribution.) The $y(\cdot)$ realizations have several limitations from the viewpoint

of computer experiments. First, the model can *only* produce quadratic draws. Second, the multivariate normal distribution of $(Y(x_1), \ldots, Y(x_L))$ is *degenerate* when $L \geq 4$. In the development below we wish to derive more flexible random functions that retain the computational advantage that $(Y(x_1), \ldots, Y(x_L))$ has the multivariate normal distribution. ■

Many computer experiments that produce either *multiple* outputs, $\boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), \ldots, y_m(\boldsymbol{x}))^\top$, or *functional* output, $y(\boldsymbol{x}, t)$, where $t \in \mathcal{T}$. As an example of the former, consider a code that computes not only $y(\boldsymbol{x})$ but also each of the partial derivatives of $y(\boldsymbol{x})$. In this case, $\boldsymbol{y}(\boldsymbol{x}) = (y(\boldsymbol{x}), \partial y(\boldsymbol{x})/\partial x_1, \ldots, \partial y(\boldsymbol{x})/\partial x_d)$. The usual approach to analyzing functional data is represent the functions by a set of basis functions, thus reducing reducing the functional output to multivariate data. Hence, Section 2.3 will describe random vector function models to handle settings that produce such output. In the general multiple output case, we view the random mechanism as associating a vector valued function, $\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{Y}(\boldsymbol{x}, \omega)$, with each elementary outcome $\omega \in \Omega$. Codes that produce multiple outcomes were introduced in Section **??**; their modeling will be considered in Subsection **??**; applications of such models will be provided in Subsections **??** and **??**.

2.2 Models Real-Valued Output

2.2.1 The stationary GP model

add discussion of hierarchical stationary GPs (as a prelude to the discussion of hierarchical Bayesian estimation)

2.2.2 Non-stationary Model 1: Regression + stationary GP model

used in later chapters as the basis for plug-in EBLUP predictors of various types; as a building block of fully Bayesian predictors; Blind kryging

2.2.3 Non-stationary Model 2: Regression + var(x) × stationary GP model ??

2.2.4 Treed GP model

2.2.5 Composite GP (Convolution) Models

Following the description of experimental goals, we summarize the basic issues in modeling computer output. Then we will be prepared to begin Chapter 3 on the first of the two basic issues considered in this book, that of predicting $y(\cdot)$ at (a new) input $\boldsymbol{x}_0$ based on training data $(\boldsymbol{x}_1, y(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_n, y(\boldsymbol{x}_n))$. Chapter 4 will address the second issue, the design problem of choosing the input sites at which the computer model should be run.

2.3 Models for Output having Mixed Qualitative and Quantitative Inputs

to be worked into a discussion of mixed input case

- **We describe a physical process by a mathematical model implemented with code on a computer. This code is sometimes referred to as a *simulator.***
- **The code produces deterministic outputs.**
- **The inputs include factors that are believed to affect the responses.**
- **We use the code to explore or experiment with the physical process, i.e., we try different inputs in order to assess their effect on the outputs. We call this a *computer experiment.***
- **The code runs slowly. One run may take a day or longer. Thus, we can only observe (experiment with) the code a small number of times. The choice of the inputs at which to observe the code must be done carefully. Monte Carlo methods that require many runs of the code are not feasible.**
- **To augment the limited number of runs of the code, we fit a statistical model (predictor) to the runs and use the statistical model to predict the code at unobserved inputs. This statistical predictor is sometimes called an *emulator.***
- **We assume that the code produces deterministic output**

  $$y(\boldsymbol{x}, t)$$

  that depend on a set of quantitative input variables

  $$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top$$

  and a qualitative variable having T levels, here indexed by t.

- **If there are $Q > 1$ qualitative variables, with the $q - th$ qualitative variable having T_q levels, we assume that the $T = \prod_{q=1}^{Q} T_q$ possible combinations of levels are indexed by a single symbol taking on values from 1 to T (lexicographically ordered). Unfortunately, this suppresses the inherent factorial structure, and we will return to this later.**

- **We also assume the quantitative input variables are restricted to some subset**

$$\mathcal{X} \subset \mathbb{R}^d$$

and that we observe the code at n points

$$(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \ldots, (\boldsymbol{x}_n, t_n)$$

- **A popular statistical model is the so-called Gaussian stochastic process (GaSP) model.**
- **As implemented in the literature, if the output of the simulator is assumed to depend only on quantitative inputs $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top$, we view our observation $y(\boldsymbol{x})$ as a realization of the random function**

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

where the β_j are unknown (regression) parameters, the f_j are known (regression) functions, and $Z(\boldsymbol{x})$ is a mean zero, second-order stationary Gaussian process with variance σ_Z^2

- **In addition, we assume**

$$\mathbf{cov}(Z(\boldsymbol{x}_i), Z(\boldsymbol{x}_j)) = \sigma_Z^2 R(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

where $R(\cdot, \cdot)$ is a valid correlation function, namely
- $R(\boldsymbol{x}_i, \boldsymbol{x}_i) = 1$
- **For any finite set of inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ the $n \times n$ matrix R whose i, jth entry is $R(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive definite.**

- **There are many possible choices for the correlation function $R(\cdot, \cdot)$. A very popular choice is the Gaussian correlation function**

$$R(\boldsymbol{x}_i, \boldsymbol{x}_j \mid \boldsymbol{\theta}) = \prod_{k=1}^{d} e^{-\theta_k (x_{i,k} - x_{j,k})^2},$$

where $\boldsymbol{x}_l = (x_{l,1}, x_{l,2}, \ldots, x_{l,d})^\top$ for $l = i, j$ and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d)^\top$ are unknown parameters (often referred to as the correlation parameters) with $\theta_i \geq 0$ for all i.
- **For purposes of this talk, assume the correlation function is the Gaussian correlation function unless otherwise stated.**

- **If $Z(\boldsymbol{x})$ is a mean zero, second-order stationary Gaussian process with variance σ_Z^2 and Gaussian correlation function, it has realizations that (w.p.1) are infinitely differentiable and shape determined by σ_Z^2 and the correlation parameters.**
- **For example, if $d = 1$, the realizations "oscillate" with amplitude determined by σ_Z^2 and frequency determined by the correlation parameter θ. Larger values of θ produce higher frequency oscillations (a less smooth looking curve) and thus some people refer to θ as a "roughness" parameter.**

- **(Insert figure)**

- **One can view the GaSP model as specifying the global (large-sale) trend and local variation around the trend.**

$$Y(\boldsymbol{x}) = \underbrace{\sum_{j=1}^{J} \beta_j f_j(\boldsymbol{x})}_{\text{regression or global trend}} + \underbrace{Z(\boldsymbol{x})}_{\text{local variation or trend}}$$

- **For this model, one can use the best linear unbiased predictor (BLUP), $\hat{Y}(\boldsymbol{x})$, as an emulator. The BLUP assumes that σ_Z^2 and $\boldsymbol{\theta}$ are known and estimates the β_j by generalized least-squares. If one substitutes estimates for σ_Z^2 and $\boldsymbol{\theta}$ into the BLUP, the resulting predictor is sometimes referred to as the empirical best linear unbiased estimator (EBLUP). Estimates might be based on maximum likelihood, restricted maximum likelihood, Bayes, or even cross-validation. Another alternative to the BLUP is to adopt a fully Bayes approach.**
- **Some people refer to using the BLUP or EBLUP as the kriging predictor.**
  - **One interesting characteristic of both the BLUP and EBLUP is that they interpolate the data. This is viewed as a desirable property in deterministic computer experiments.**
  - **Once we observe the simulator at a particular input, we know that further runs at the same input will produce the same output. Thus, it seems reasonable to use predictors that are interpolators.**

- **Because the EBLUP is an interpolator (regardless of the form of the regression part of the model) , and because realizations of $Z(\boldsymbol{x})$ can take on a wide variety of shapes, it is not uncommon to assume that the regression portion of the model is very simple. Many papers assume that the regression part is simply constant and use the constant mean model**

$$Y(\boldsymbol{x}) = \beta + Z(\boldsymbol{x})$$

- **Another option is to assume that the regression portion is just a linear trend.**
- **Note, another interpretation of the GaSP model is to think of it as specifying a prior on the the output as a function of $\boldsymbol{x}$.**

- **The GaSP model and kriging (possibly Bayesian) have become a standard method for developing a statistical predictor or emulator.**
- **Problem: Typical correlation functions (such as the Gaussian) assume that all the inputs are quantitative.**
- **Question: How can one incorporate qualitative variables into the GaSP model?**
- **Several methods have been proposed.**

- **We model $y(\boldsymbol{x}, t)$ as**

$$Y(\boldsymbol{x}, t) = \sum_{j=1}^{J} \beta_j f_j(\boldsymbol{x}, t) + Z_t(\boldsymbol{x})$$

where all terms are as before and $Z_t(\boldsymbol{x})$ is a mean zero, second-order stationary Gaussian process with variance $\sigma^2_{Z,t}$
- **In what follows I will assume a constant means model of the form**

$$Y(\boldsymbol{x}, t) = \beta_t + Z_t(\boldsymbol{x})$$

- **In this model, one often interprets $y(\boldsymbol{x}, t)$ as determining t different curves or response surfaces, indexed by t.**

- **We can fit separate GaSP models to each response surface. But perhaps if the response surfaces are similar, we can build predictors for each surface that "borrow" information from the other surfaces. This is similar to what one does in multiple regression by using indicator variables to represent different response surfaces.**
- **For example, one can use indicator variables to write a single multiple regression model representing several lines. This single model has more degrees of freedom for error than fitting separate lines. This comes at the expense of having to assume the error variance is the same for each line.**

- **What happens if we simply add indicator variables to our model and act as though they are quantitative?**
  For $1 \leq t \leq$ T define

$$I_t(i) = \begin{cases} 1 \text{ if } i = t \\ 0 \text{ otherwise} \end{cases}$$

- **In this case the Gaussian correlation function becomes**

$$\begin{aligned} R((\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2)) &= \prod_{l=1}^{T} e^{-\gamma_l (I_l(t_1) - I_l(t_2))^2} \times \prod_{k=1}^{d} e^{-\theta_k (x_{1,k} - x_{2,k})^2} \\ &= e^{-\gamma_{t_1}} \times e^{-\gamma_{t_2}} \times \prod_{k=1}^{d} e^{-\theta_k (x_{1,k} - x_{2,k})^2} \\ &= \tau_{t_1} \tau_{t_2} \times \prod_{k=1}^{d} e^{-\theta_k (x_{1,k} - x_{2,k})^2} \end{aligned}$$

where $\tau_{t_j} = e^{-\gamma_{t_j}}$. Notice $0 < \tau_{t_j} < 1$.

- **Suppose $T = 4$ with response surfaces 1 and 2 highly correlated, response surfaces 3 and 4 highly correlated, but response surfaces 1 and 3 essentially uncorrelated. If 1 and 2 are highly correlated we would expect $\tau_1\tau_2$ to be close to 1, and hence both τ_1 and τ_2 must be close to 1. Similarly, if 3 and 4 are highly correlated we would expect $\tau_3\tau_4$ to be close to 1, and hence both τ_3 and τ_4 must be close to 1. However, if 1 and 3 are essentially uncorrelated, we would expect $\tau_1\tau_3$ to be close to 0. But this is impossible if both are close to 1. (Of course, this assumes we can interpret the τ_i as correlations between response surfaces.)**

- **This shows that if we "naively" use indicator variables to represent qualitative variables (as we would do in standard regression) we impose a certain structure on "between response surfaces" correlations.**
  Thus, simply using indicator variables to represent qualitative variables (at least in the correlation function) does not work well.

- **So one has to model the qualitative variables more carefully, at least in terms of the correlation structure.**

- **One approach (Kennedy and O'Hagan 2000) assumes one has a collection of multi-fidelity computer simulations, each involving the same quantitative factors, and that these mulit-fidelity simulations can be modeled collectively by a single computer model with a common set of quantitative factors and a qualitative factor to describe the different accuracy of the simulations.**

- **This approach implicitly assumes that as the level of the qualitative factor changes (increases) the fidelity increases. Thus, it may not be appropriate for the more general case of incorporating a qualitative variable.**

- **Another approach (Han, et al. 2009) assumes**

  - **the $Z_t(\cdot)$ are mutually independent**
  - **Corr$(Z_t(\boldsymbol{x}_1), Z_t(\boldsymbol{x}_2)) = \prod_{k=1}^{d} e^{-\theta_{t,k}(x_{i,k}-x_{j,k})^2} = \prod_{k=1}^{d} \rho_{t,k}^{(x_{1,k}-x_{2,k})^2}$**
  - **certain priors on the β_t, the σ_t^2, and the $\rho_{t,k} = e^{-\theta_{t,k}}$.**

  and through the priors attempts to capture "similarities" between between the response surfaces $y(\boldsymbol{x}, t)$.

- **Yet another approach (Qian et al. 2008 and Zhou et al. 2010) assumes**

  - **Corr$(Z_{t_1}(\boldsymbol{x}_1), Z_{t_2}(\boldsymbol{x}_2)) = \tau_{t_1,t_2} \prod_{i=1}^{d} e^{-\theta_i(x_{1,i}-x_{2,i})^2}$ where τ_{t_1,t_2} is the cross correlation between the response surfaces corresponding to "categories" t_1 and t_2 of the qualitative variable.**
  - **$T \times T$ matrix $\boldsymbol{\tau} = \{\tau_{r,s}\}$ is a positive definite matrix with unit diagonal elements (this guarantees that the matrix of correlation whose i, jth entry is Corr$(Z_{t_i}(\boldsymbol{x}_i), Z_{t_j}(\boldsymbol{x}_j))$ is a valid correlation matrix) .**
  - **$\sigma_t^2 = \sigma_Z^2$ for all t.**

- **A hypersphere decomposition is used to model τ in Zhou et al. (2010), and this parameterization is quite useful for fitting the model.**

- **Several special cases of this model are mentioned by Qian et al. (2008). Each reduces the number of parameters one needs to estimate to fit the model.**
- **One case assumes all $\tau_{t_i,t_j} = \tau$ for $i \neq j$. This is sometimes referred to as the exchangeable model.**
- **Another (see McMillian et al. 1999) assumes**

$$\tau_{t_i,t_j} = e^{-(\theta_i+\theta_j)} I[i \neq j]$$

where θ_i and θ_j are positive and $I[i \neq j]$ is the indicator function.

- **Note that we encountered the McMillan et al. (1999) structure previously when we simply used indicator variables to represent the qualitative variable. We saw that this had undesirable properties.**

- **A third special case is a Kronecker product structure. Suppose we have J qualitative variables, and the jth qualitative variable has T_j levels. Let $\boldsymbol{t} = (t_1, t_2, \ldots, t_J)^\top$ denote the vector of qualitative variable levels for an input that has qualitative variable j at level t_j for $j = 1, \ldots, J$. A legitimate correlation function is**

$$\mathbf{Corr}(Z_{\boldsymbol{t}_1}(\boldsymbol{x}_1), Z_{\boldsymbol{t}_2}(\boldsymbol{x}_2)) = \prod_{j=1}^{J} \tau_{j,t_{1,j},t_{2,j}} \prod_{i=1}^{d} e^{-\theta_i (x_{1,i}-x_{2,i})^2}$$

where τ_j, the $T_j \times T_j$ matrix with r, sth entry $\tau_{j,r,s}$, is positive definite with unit diagonal entries. This corresponds to taking $\tau = \tau_1 \otimes \cdots \otimes \tau_J$, where $\otimes$ is the Kronecker product.

- **This special case reduces the number of τ_{t_i,t_j} parameters in the model. It also "imposes" a sort of multiplicative main effects structure on the τ_{t_i,t_j}, and hence takes into account the factorial structure.**

- **Qian et al. (2008) consider additional forms for τ and for the τ_{t_i,t_j} that assume the levels of the qualitative factor can be organized into similar groups and that allow for ordinal qualitative factors.**

- **The flexibility of the formulation in Qian et al. (2008) makes their model attractive and it has appeared in several papers. As a consequence, I will take a closer look at it.**
  But first, some comments.

- **The model used in Qian et al. (2008) and Zhou et al. (2010) makes some implicit assumptions about the different response surfaces determined by t. First, in the correlation structure**

$$\mathbf{Corr}(Z_{t_1}(\boldsymbol{x}_1), Z_{t_2}(\boldsymbol{x}_2)) = \tau_{t_1,t_2} \prod_{k=1}^{d} e^{-\theta_k (x_{1,k}-x_{2,k})^2}$$

the correlation parameters θ_k and the process variance σ_Z^2 are the same for all values of t. This implies that the "shape" of the local variation as a function of the quantitative variables is the same.

- **It is possible to use indicator variables in the Gaussian correlation function to generate the correlation structure in Qian et al. (2008). One way, is as follows.**
- **For** $1 \leq p \leq$ **T define**

$$I_p(i) = \begin{cases} 1 \textbf{ if } p = i \\ 0 \textbf{ otherwise} \end{cases}$$

and for $1 \leq p, q \leq T - 1$

$$W_{p,q}(i) = \begin{cases} I_p(i) + I_q(i) \textbf{ if } p \neq q \\ I_p(i) \qquad\quad \textbf{if } p = q \end{cases}$$

- $$\textbf{Corr}(Z_{t_1}(\boldsymbol{x}_1), Z_{t_2}(\boldsymbol{x}_2)) = \prod_{p,q=1}^{T-1} e^{-\gamma_{p,q}(W_{p,q}(t_1)-W_{p,q}(t_2))^2} \prod_{k=1}^{d} e^{-\theta_k(x_{1,k}-x_{2,k})^2}$$

- **One can show with some algebra, assuming $\gamma_{p,q} = \gamma_{q,p}$ and for $\tau_{i,j} > 0$, that for $i \neq j, i < T, j < T$**

$$-ln(\tau_{i,j}) = \gamma_{i,i} + \gamma_{j,j} - 4\gamma_{i,j} + 2 \sum_{q=1,q\neq i}^{T-1} \gamma_{i,q} + 2 \sum_{q=1,q\neq j}^{T-1} \gamma_{j,q}$$

and for $i \neq j, i = T, j < T$

$$-ln(\tau_{T,j}) = \gamma_{j,j} + 2 \sum_{q=1,q\neq j}^{T-1} \gamma_{j,q}$$

and for $i \neq j, i < T, j = T$

$$-ln(\tau_{T,j}) = \gamma_{i,i} + 2 \sum_{q=1,q\neq i}^{T-1} \gamma_{i,q}$$

- **Also for $i \neq j, i < T, j < T$**

$$\gamma_{i,j} = \frac{1}{4}(ln(\tau_{i,j}) - ln(\tau_{T,j}) - ln(\tau i, T)$$

and for $i < T$

$$\gamma_{i,i} = -\frac{1}{2} \sum\nolimits_q = 1, q \neq i^T ln(\tau_{i,q}) + \frac{1}{2} \sum\nolimits_q = 1, q \neq i^{T-1} ln(\tau_{T,q})$$

- **Thus for $\tau_{i,j} > 0$ there is a one-to-one correspondence between the $\tau_{i,j}, i \neq j$ and the $\gamma_{p,q}, p < m, q < m$ in the sense that given the $\tau_{i,j}$ we can determine the corresponding $\gamma_{p,q}$ and vice-versa.**
  This formulation with the variables $W_{p,q}(\cdot)$ allows us to use standard software for fitting the Gaussian correlation to estimate the $\gamma_{p,q}$ and θ_k and then obtain the $\tau_{i,j}$ in the Qian, Wu, Wu (2008) model, assuming all $\tau_{i,j} > 0$.

- **Another way to reformulate the Qian et al. (2008) model so that the correlation structure looks like one determined by the Gaussian correlation function is the following.**
- **In Kennedy and O'Hagan (2001), the Gaussian correlation function is expressed in the more general form**

$$R(\boldsymbol{x}_i, \boldsymbol{x}_j \mid \boldsymbol{\Omega}) = e^{-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{\Omega} (\boldsymbol{x}_i - \boldsymbol{x}_j)},$$

  where $\boldsymbol{\Omega}$ is an unknown $d \times d$ positive definite symmetric matrix whose i, jth entry is $\omega_{i,j}$. This reduces to the form presented earlier if $\boldsymbol{\Omega}$ is a diagonal matrix.

- **As before, for $1 \leq p \leq$ T define**

$$I_p(i) = \begin{cases} 1 \text{ if } p = i \\ 0 \text{ otherwise} \end{cases}$$

  Let

$$\boldsymbol{I}(i) = (I_1(i), I_2(i), \ldots, I_T(i))^\top$$

-

$$\begin{aligned} \mathbf{Corr}(Z_{t_1}(\boldsymbol{x}_1), Z_{t_2}(\boldsymbol{x}_2)) = {} & e^{-(\boldsymbol{I}(t_1) - \boldsymbol{I}(t_2))^\top \boldsymbol{\Omega} (\boldsymbol{I}(t_1) - \boldsymbol{I}(t_2))} \\ & \times e^{-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \mathbf{diag}(\theta_1, \ldots, \theta_d)(\boldsymbol{x}_i - \boldsymbol{x}_j)} \end{aligned}$$

- **In this formulation, one can show for $i \neq j$ and assuming all $\tau_{i,j} > 0$,**

$$-ln(\tau_{i,j}) = \omega_{i,i} + \omega_{j,j} - 2\omega_{i,j}$$

- **Notice that this reduces to the case $\tau_{i,j} = \tau$ when $\boldsymbol{\Omega}$ is a multiple of the identity matrix, and it reduces to the McMillan et al. (1999) model when $\boldsymbol{\Omega}$ is a diagonal matrix.**
  One can define separate $T_j \times T_j$ matrices $\boldsymbol{\Omega}_j$ and obtain the Kronecker product formulation of Qian et al. (2008), with each $\boldsymbol{\Omega}_j$ corresponding to the $\boldsymbol{\tau}_j$.

- **Another way to incorporate qualitative variables, inspired by how one can characterize the multivariate normal distribution, is as follows. Let $N_1((\boldsymbol{x}), N_2((\boldsymbol{x}), \ldots, N_S((\boldsymbol{x})$ be S independent, identically distributed mean zero, second-order stationary Gaussian processes with variance σ_Z^2. Assume each satisfies**

$$\mathbf{Corr}(N_i(\boldsymbol{x}_1), N_i(\boldsymbol{x}_2)) = \prod_{k=1}^{d} e^{-\theta_i(x_{1,i}-x_{2,i})^2}$$

- **Assume for** $1 \le t \le T$

$$Z_t(\boldsymbol{x}) = \sum_{i=1}^{S} a_{i,t} N_i(\boldsymbol{x})$$

- **Then**

$$(Z_1(\boldsymbol{x}), \ldots, Z_T(\boldsymbol{x}))^\top = \boldsymbol{A}(N_1(\boldsymbol{x}, \ldots, N_S(\boldsymbol{x}))^\top$$

  where A **is the** $T \times S$ **matrix with** $a_{i,j}$ **as its** i, jth **entry.**

- **This yields the Qian et al. (2008) model provided** $\tau = AA^\top$**. Qian, et al. (2008) use this representation to prove that** τ **must be a positive definite symmetric matrix with unit diagonal entries for the correlation structure in their model to be valid.**
  But there is much more that can be done with this representation. Much of what I now say is inspired by multivariate normal methods.

- **The exchangeable model can be represented by**

$$Z_i(\boldsymbol{x}) = \sqrt{\tau} N_1(\boldsymbol{x}) + \sqrt{1-\tau} N_{i+1}(\boldsymbol{x}),$$

  This indicates that the $Y(\boldsymbol{x}, i)$ **are composed of a common overall trend (the** $N_1(\boldsymbol{x})$ **term) and independent realizations of a "treatment effect" trend (the** $N_{i+1}(\boldsymbol{x})$ **terms). Both the overall trend and treatment effect trends are of the same magnitude for each** $Y(\boldsymbol{x}, i)$**.**

- **The McMillan et al. (1999) model can be represented by**

$$Y(\boldsymbol{x}, i) = \tau_i N_1(\boldsymbol{x}) + \sqrt{1-\tau_i^2} N_{i+1}(\boldsymbol{x}),$$

  This is similar to the exchangeable model, except the overall trend and treatment effect trends can be of different magnitudes for each $Y(\boldsymbol{x}, i)$**.**

- **Both the exchangeable model and the McMillan et al. (1999) model could be interpreted as having a one-way ANOVA structure or a constant mean (trend) structure plus noise.**

- **To represent the Kronecker product structure, let**

$$N^j(\boldsymbol{x}) = (N_1^j(\boldsymbol{x}), \ldots, N_{T_j}^j(\boldsymbol{x}))^\top$$

  where the $N_i^j(x)$ **are independent, identically distributed mean zero, second-order stationary Gaussian processes with variance** σ_Z^2**. Let** A_j **be the** $T_j \times T_j$

matrix satsfying $\tau_j = A_j A_j^\top$. Then in our general formulation $A(N_1(x), \ldots, N_S(x))^\top$ becomes

$$(\otimes_{j=1}^{J} A_j)(\otimes_{j=1}^{J} N^j(x)).$$

- **One can impose a factorial structure on the $Y(x, i)$. For example, suppose we believe the $Y(x, i)$ are determined by two factors F and G with f and g levels, respectively. Suppose $Y(x, i)$ corresponds to F at level ϕ and G at level γ.**

$$Y(x, i) \propto a_i^\mu N^\mu(x) + a_i^F N_\phi^F(x) + a_i^G N_\gamma^G(x),$$

  where $N^\mu(x)$ is an overall mean effect (trend), $N_\phi^F(x)$ is the effect of level ϕ of F, and $N_\gamma^G(x)$ the effect of level γ of G. This looks like a main effects model.

- **An alternative is to include a small "error" effect $N_i^\epsilon(x)$ which gives**

$$Y(x, i) \propto a_i^\mu N^\mu(x) + a_i^F N_\phi^F(x) + a_i^G N_\gamma^G(x) + a_i^\epsilon N_i^\epsilon(x)$$

- **One might think of a_i^ϵ as small relative to a_i^μ, a_i^F, and a_i^G**

- **If one does not require A to satisfy $\tau = AA^\top$, then the formulation**

$$(Y(x, 1), \ldots, Y(x, T))^\top = A(N_1(x, \ldots, N_S(x))^\top$$

- **Allows the $Y(x, i)$ to have different variances.**

- **Another application of this representation of the Qian et al. (2008) model is a kind of factor analysis. Estimate the matrix τ and find a parsimonious matrix A so that $\tau = AA^\top$. The form of A may suggest some sort of factorial structure, or perhaps the $Y(x, i)$ depend mostly on a relatively small number of the $N_j(x)$.**

  - **Comments on fitting models and examples to be added. One challenge is similar to what one encounters in factor analysis. In this representation of the Qian et al. (2008) model the matrix A is only determined up to multiplication by an orthogonal matrix.**

- **I did not say much about the Han et al. (2009) model. But this model allows the $Y(x, i)$ to have different variances and different correlation parameters. Can one choose priors to reflect special structure, such as factorial structure?**

- **Are there better ways to incorporate qualitative variables into the models that are popular in computer experiments?**
  How about treed processes?
  People that are not "locked into" the GaSP model may have fresh ideas.

- **Viewing $(Y(\boldsymbol{x}, 1), \ldots, Y(\boldsymbol{x}, T))^\top$ as multivariate may indicate that one should consider multivariate stochastic processes. There is some literature on multivariate stochastic processes. The use of multivariate stochastic processes in the spatial literature sometimes appears in papers discussing "co-regionalization."**

- **What sorts of experimental designs should one use to fit models such as Qian et al. (2008)? Should the same or very similar $\boldsymbol{x}$ be observed on each $Y(\boldsymbol{x}, i)$? Should different $\boldsymbol{x}$ be observed to better borrow information from those $Y(\boldsymbol{x}, i)$ that are similar? Perhaps a mixture of same and different $\boldsymbol{x}$? What about a sequential procedure that identifies the $\boldsymbol{x}$ and i for which the EBLUP $\hat{Y}(\boldsymbol{x}, i)$ has the largest prediction error, and then takes the next observation on curve i at $\boldsymbol{x}$?**
- **Other strategies?**

- **Do we really need to worry about qualitative variables in computer experiments? Are qualitative variables used in simulations, or is there a latent quantitative variable present? For example, types of metal used in a product could be viewed as qualitative, but in a simulation perhaps some quantitative property of the metal is all that is used.**

- **Much of what I have said is work-in-progress with my Ph.D. student, Yulei Zhang. One of the great things about this conference is that it is a wonderful forum for presenting such work and getting critical feedback.**

- **Treed processes as an alternative.**
- **Emulators are interpolators, so how much do we need to worry about nonstationarity?**
- **Computational issues with many factors with many levels. Is screening needed?**

Some References

1. **Han, G., Santner, T. J., Notz, W. I., and Bartel, D. L. (2009). Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables. *Technometrics* 51 (3), 278-288.**
2. **Kennedy, M. C. and O'Hagan, A. (2000). Predicting the Output From a Complex Computer Code When Fast Approximations are Available. *Biometrika* 87, 1-13.**
3. **Kennedy, M. C. and O'Hagan, A. (2001). Bayesian Calibration of Computer Models (with discussion). *Journal of the Royal Statistical Society B* 63, 425-464.**
4. **McMillan, N. J., Sacks, J., Welch, W. J., and Gao, F. (1999). Analysis of Protein Activity Data by Gaussian Stochastic Process Models. *Journal of Biopharmeceutical Statistics* 9, 145-160.**
5. **Qian, Z., Seepersad, C., Joseph, R., Allen, J. and Wu, C. F. J. (2008). Building Surrogate Models with Detailed and Approximate Simulations. *ASME Journal of Mechanical Design* 128, 668-677.**

6. Qian, P. Z. G. and Wu, C. F. J. (2008). Bayesian Hierarchical Modeling for Integrating Low-accuracy and High-accuracy Experiments. ***Technometrics*** **50, 192-204.**
7. Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008). Gaussian Process Models for Computer Experiments With Qualitative and Quantitative Factors. ***Technometrics*** **50, 383-396.**
8. Zhou, Q., Qian, P. Z. G., Wu, H., and Zhou, S. (2010). A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors. ***Technical Report*** **University of Wisconsin.**

2.4 Models for Multivariate and Functional Computer Output

2.4.1 Reducing Functional Data to Multivariate Data

2.4.2 Constructive Models

- AR spatial process models
- AR with shifts and rescaling

2.4.3 Separable Models (Conti and O'Hagan)

2.4.4 Basis Representations of Multivariate Output

- PCA
- kernel regression

$$\begin{aligned}\mathrm{Cor}\{Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)\} &= \frac{\mathrm{Cov}\{Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)\}}{\sqrt{\mathrm{Var}\{Z(\boldsymbol{x}_1)\} \times \mathrm{Var}\{Z(\boldsymbol{x}_2)\}}} \\ &= \frac{C(\boldsymbol{x}_1 - \boldsymbol{x}_2)}{\sigma_z^2} = R(\boldsymbol{x}_1 - \boldsymbol{x}_2).\end{aligned}$$

What properties must valid covariance and correlation functions possess? Assuming that $Z(\boldsymbol{x})$ is nondegenerate, then $C(\mathbf{0})$ $(= \sigma_z^2) > 0$ while $R(\mathbf{0}) = 1$. Because $\mathrm{Cov}\{Y(\boldsymbol{x}+\boldsymbol{h}), Y(\boldsymbol{x})\} = \mathrm{Cov}\{Y(\boldsymbol{x}), Y(\boldsymbol{x}+\boldsymbol{h})\}$, the covariance and correlation functions of stationary GRFs must be *symmetric about the origin*, i.e.,

$$C(\boldsymbol{h}) = C(-\boldsymbol{h}) \text{ and } R(\boldsymbol{h}) = R(-\boldsymbol{h}).$$

Both $C(\cdot)$ and $R(\cdot)$ must be *positive semidefinite* functions; stated in terms of $C(\cdot)$, this means that for any $L \geq 1$, and any real numbers $w_1, \ldots, w_L$, and any inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ in $\mathcal{X}$,

$$\sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j C(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq 0. \tag{2.4.1}$$

The sum (2.4.1) must be nonnegative because the left-hand side is the variance of $\sum_{i=1}^{L} w_i Y(\boldsymbol{x}_i)$. The covariance function $C(\cdot)$ is *positive definite* provided > 0 holds in (2.4.1) for every $(w_1, \ldots, w_L) \neq \mathbf{0}$ (any $L \geq 1$ and any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ in $\mathcal{X}$).

While every covariance function must satisfy the symmetry and positive semidefinite properties above, these properties do not offer a convenient method for generating valid covariance functions. Rather, what is of greater importance is a characterization of the class of covariance functions because this would allow us to generate valid covariance functions. While a general study of how to determine the form of valid stationary covariance functions is beyond the scope of this book, one answer to this question is relatively simple to state, and we do so next.

As a prelude to identifying this class of covariance functions (and as an introduction to the topic of smoothness which is taken up again in Subsection 2.4.5), we introduce the concept of mean square (MS) continuity. Mean square properties describe the average performance of the sample paths. For purposes of stating the definitions of MS properties, there is nothing to be gained by restricting attention to GRFs and so we consider general random functions $Y(\cdot)$.

Definition Suppose $Y(\cdot)$ is a stationary process on $\mathcal{X}$ that has finite second moments. We say that $Y(\cdot)$ is *MS continuous* at the point $\boldsymbol{x}_0 \in \mathcal{X}$ provided

$$\lim_{\boldsymbol{x} \to \boldsymbol{x}_0} E\left\{(Y(\boldsymbol{x}) - Y(\boldsymbol{x}_0))^2\right\} = 0.$$

The process is *MS continuous on* $\mathcal{X}$ provided it is MS continuous at every $\boldsymbol{x}_0 \in \mathcal{X}$.

Suppose $C_Y(\cdot)$ is the covariance function of the stationary process $Y(\cdot)$, then

$$E\left\{(Y(\boldsymbol{x}) - Y(\boldsymbol{x}_0))^2\right\} = 2\left(C_Y(\mathbf{0}) - C_Y(\boldsymbol{x} - \boldsymbol{x}_0)\right). \tag{2.4.2}$$

The right-hand formula shows that $Y(\cdot)$ is MS continuous at $\boldsymbol{x}_0$ provided $C_Y(\cdot)$ is continuous at the origin—in fact, $Y(\cdot)$ is MS continuous at *every* $\boldsymbol{x}_0 \in \mathcal{X}$ provided $C_Y(\cdot)$ is continuous at the origin. Stated in terms of the correlation function, $C_Y(\boldsymbol{h}) \to C_Y(\mathbf{0}) = \sigma_z^2$ as $\boldsymbol{h} \to \mathbf{0}$ is equivalent to

$$R_Y(\boldsymbol{h}) = C_Y(\boldsymbol{h})/\sigma_z^2 \to 1.0 \text{ as } \boldsymbol{h} \to \mathbf{0}.$$

Continuing our discussion of general random functions $Y(\cdot)$, Bochner (1955) proved that the covariance function of every stationary, MS continuous random function $Y(\cdot)$ on $\mathbb{R}^d$, can be written in the form

$$C_Y(\boldsymbol{h}) = \int_{\mathbb{R}^d} \cos(\boldsymbol{h}^\top \boldsymbol{w})\, dG(\boldsymbol{w}), \tag{2.4.3}$$

where $G(\cdot)$ is positive finite symmetric measure on $\mathbb{R}^d$. In particular, this characterization must hold for the special case of stationary GRFs. (See also the discussions in Cramér and Leadbetter (1967) on page 126, Adler (1981) on page 25, Cressie (1993) on page 84, or Stein (1999) on page 22-25.)

The process variance corresponding to $C_Y(\cdot)$ having the form (2.4.3) is

$$C_Y(\mathbf{0}) = \int_{\mathbb{R}^d} dG(\boldsymbol{w}) < +\infty$$

which is finite because G is a bounded measure on $\mathbb{R}^d$; $F(\cdot) = G(\cdot)/C_Y(\mathbf{0})$ is a symmetric probability distribution, called the *spectral distribution*, corresponding to $C_Y(\cdot)$. The function

$$R_Y(\boldsymbol{h}) = \int_{\mathbb{R}^d} \cos(\boldsymbol{h}^\top \boldsymbol{w})\, dF(\boldsymbol{w}) \tag{2.4.4}$$

is the correlation function corresponding to the spectral distribution $F(\cdot)$. If $F(\cdot)$ has a density $f(\cdot)$, then $f(\cdot)$ is called the *spectral density* corresponding to $R_Y(\cdot)$. In this case

$$R_Y(\boldsymbol{h}) = \int_{\mathbb{R}^d} \cos(\boldsymbol{h}^\top \boldsymbol{w}) f(\boldsymbol{w})\, d\boldsymbol{w}. \tag{2.4.5}$$

The right-hand side of (2.4.5) gives us a method to produce valid correlation functions (and covariance functions)—choose a symmetric density $f(\cdot)$ and evaluate the integral (2.4.5).

Example 2.2. This first example shows how (2.4.5) can be used to generate valid correlation functions from probability density functions that are symmetric about the origin. Consider the one-dimensional case. Perhaps the simplest choice of one-dimensional density is the uniform density over a symmetric interval which we take to be $(-1/\theta, +1/\theta)$ for a given $\theta > 0$. Thus the spectral density is

$$f(w) = \begin{cases} \theta/2, & -1/\theta < w < 1/\theta \\ 0, & \text{otherwise} \end{cases}$$

and the corresponding correlation function is

$$R(h) = \int_{-1/\theta}^{+1/\theta} \frac{\theta}{2} \cos(hw)\, dw = \begin{cases} \frac{\sin(h/\theta)}{h/\theta}, & h \neq 0 \\ 1, & h = 0 \end{cases}.$$

This correlation has scale parameter θ; Figure 2.2 shows that $R(h)$ can model both positive and negative correlations. ■

Any function $R_Y(\cdot)$ of the form (2.4.4) must satisfy $R_Y(\mathbf{0}) = 1$, must be continuous at $\boldsymbol{h} = \mathbf{0}$, must be symmetric about $\boldsymbol{h} = \mathbf{0}$, and must be positive semidefinite. The

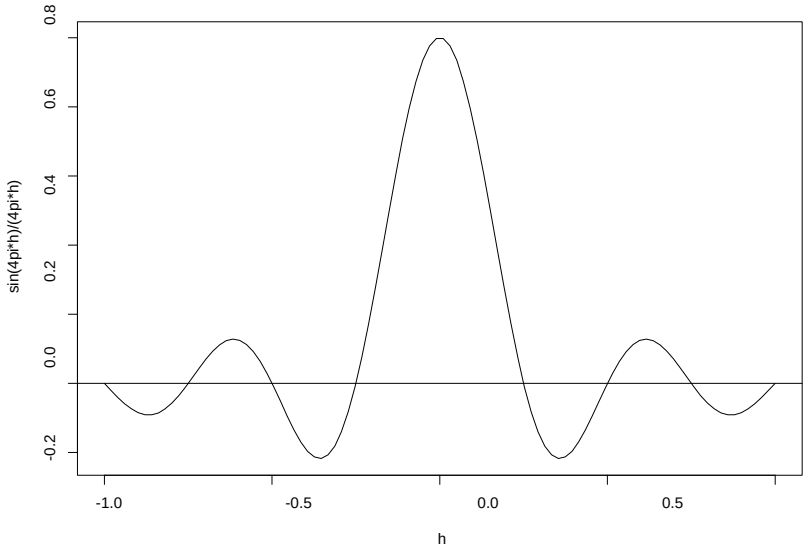


Fig. 2.2 The correlation function $R(h) = \sin(h/\theta)/(h/\theta)$ for $\theta = 1/4\pi$ over h in $[-1, +1]$.

first consequence holds because

$$R_Y(\mathbf{0}) = \int_{\mathbb{R}^d} \cos(\mathbf{0}^\top \boldsymbol{w})\, dF(\boldsymbol{w}) = \int_{\mathbb{R}^d} 1\, dF(\boldsymbol{w}) = 1,$$

where the third equality in the above is true because $F(\cdot)$ is a probability distribution. Continuity follows by an application of the dominated convergence theorem; notice that from the argument following (2.4.2), continuity of $R_Y(\boldsymbol{h})$ at the origin insures that the corresponding process is MS continuous. Symmetry holds because $\cos(-x) = \cos(x)$ for all real x. Positive semidefinite is true because for any $L \geq 1$, any real numbers $w_1, \ldots, w_L$, and any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ we have

$$\sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j R_Y(\boldsymbol{x}_i - \boldsymbol{x}_j)$$

$$
\begin{aligned}
&= \int_{\mathbb{R}^d} \sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \cos(\boldsymbol{x}_i^\top \boldsymbol{w} - \boldsymbol{x}_j^\top \boldsymbol{w})\, dF(\boldsymbol{w}) \\
&= \int_{\mathbb{R}^d} \sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \left\{ \cos(\boldsymbol{x}_i^\top \boldsymbol{w}) \cos(\boldsymbol{x}_j^\top \boldsymbol{w}) \right. \\
&\qquad \left. + \sin(\boldsymbol{x}_i^\top \boldsymbol{w}) \sin(\boldsymbol{x}_j^\top \boldsymbol{w}) \right\} dF(\boldsymbol{w}) \\
&= \int_{\mathbb{R}^d} \left\{ \left(\sum_{i=1}^{L} w_i \cos(\boldsymbol{x}_i^\top \boldsymbol{w}) \right)^2 + \left(\sum_{i=1}^{L} w_i \sin(\boldsymbol{x}_i^\top \boldsymbol{w}) \right)^2 \right\} dF(\boldsymbol{w}) \\
&\geq 0.
\end{aligned}
$$

Continuity, symmetry, and positive semidefiniteness also hold for any covariance function $C_Y(\cdot)$ of form (2.4.3).

We conclude by mentioning several additional tools that are extremely useful for "building" covariance and correlation functions given a basic set of such functions. Suppose that $C_1(\cdot)$ and $C_2(\cdot)$ are valid covariance functions. Then their sum and product,

$$C_1(\cdot) + C_2(\cdot) \text{ and } C_1(\cdot) \times C_2(\cdot),$$

are also valid covariance functions. The sum, $C_1(\cdot) + C_2(\cdot)$, is the covariance of two independent processes, one with covariance function $C_1(\cdot)$ and the other with covariance function $C_2(\cdot)$. Similarly, $C_1(\cdot) \times C_2(\cdot)$ is the covariance function of the product of two independent zero-mean GRFs with covariances $C_1(\cdot)$ and $C_2(\cdot)$, respectively.

The product of two valid correlation functions, $R_1(\cdot)$ and $R_2(\cdot)$, is a valid correlation function, but their sum is not (notice that $R_1(\mathbf{0}) + R_2(\mathbf{0}) = 2$, which is not possible for a correlation function). Correlation functions that are the products of one-dimensional marginal correlation functions are sometimes called *separable* correlation functions (not to be confused with the earlier use of the term separable).

We now introduce two widely–used families of correlation functions that have been used in the literature to specify stationary Gaussian stochastic processes (see also Journel and Huijbregts (1978), Mitchell et al (1990), Cressie (1993), Vecchia (1988), and Stein (1999)).

Example 2.3. Another familiar choice of a symmetric density that can be used as a spectral density is the normal density. To give a simple form for the resulting correlation function, take the spectral density to be $N(0, 2/\theta^2)$ for $\theta > 0$. Calculation gives

$$
\begin{aligned}
R(h) &= \int_{-\infty}^{+\infty} \cos(hw) \frac{\theta}{\sqrt{2\pi}\sqrt{2}} \exp\{-w^2\theta^2/4\}\, dw \\
&= \exp\left\{-(h/\theta)^2\right\}. \qquad (2.4.6)
\end{aligned}
$$

This correlation is sometimes called the *Gaussian correlation function* because of its form but the reader should realize that the name is, perhaps, a misnomer. The Gaussian correlation function is a special case of the more general family of corre-

lations called the power exponential correlation family. This family is far and away the most popular family of correlation models in the computer experiments literature. The one-dimensional GRF $Z(x)$ on $x \in \mathbb{R}$ has *power exponential* correlation function provided

$$R(h) = \exp\{-|h/\theta|^p\} \quad \text{for } h \in \mathbb{R}, \tag{2.4.7}$$

where $\theta > 0$, and $0 < p \leq 2$. In addition to the Gaussian subfamily, the case $p = 1$,

$$R(h) = \exp\{-(|h|/\theta)\}$$

is well-studied. The GRF corresponding to this correlation function is known as the Ornstein-Uhlenbeck process.

For later reference, we note that every power exponential correlation function, $0 < p \leq 2$, is continuous at the origin, and none, except the Gaussian $p = 2$, is differentiable at the origin. In fact, the Gaussian correlation function is infinitely differentiable at the origin.

From the fact that products of correlation functions are also correlation functions,

$$R(\boldsymbol{h}) = \exp\left\{-\sum_{j=1}^{d} |h_j/\theta_j|^{p_j}\right\} \tag{2.4.8}$$

is a d-dimensional separable version of the power exponential correlation function, as is the special case of the product Gaussian family

$$R(\boldsymbol{h}) = \exp\left\{-\sum_{j=1}^{d} (h_j/\theta_j)^2\right\}$$

which has dimension–specific scale parameters. ■

Example 2.4. Suppose that $Z(x)$ is a one-dimensional GRF on $x \in \mathbb{R}$ with correlation function

$$R(h|\theta) = \begin{cases} 1 - 6\left(\frac{h}{\theta}\right)^2 + 6\left(\frac{|h|}{\theta}\right)^3, & |h| \leq \theta/2 \\ 2\left(1 - \frac{|h|}{\theta}\right)^3, & \theta/2 < |h| \leq \theta \\ 0, & \theta < |h| \end{cases}, \tag{2.4.9}$$

where $0 < \theta$ and $h \in \mathbb{R}$. The function $R(h|\theta)$ has two continuous derivatives at $h = 0$ and also at the change point $h = \theta/2$ (see the right column of Figure 2.6). $R(h|\theta)$ assigns zero correlation to inputs x_1 and x_2 that are sufficiently far apart ($|x_1 - x_2| > \theta$). Formally, the spectral density that produces (2.4.9) is proportional to

$$\frac{1}{w^4\theta^3}\{72 - 96\cos(w\theta/2) + 24\cos(w\theta)\}.$$

Anticipating Section 3.2 on prediction in computer experiments, the use of (2.4.9) leads to cubic spline interpolating predictors. As in the previous example, we note that

$$R(\boldsymbol{h}|\boldsymbol{\theta}) = \prod_{j=1}^{d} R(h_j|\theta_j)$$

for $\boldsymbol{h} \in \mathbb{R}^d$ is a correlation function that allows each input dimension to have its own scale and thus dimension specific rate at which $Z(\cdot)$ values become uncorrelated. Other one-dimensional cubic correlation functions can be found in Mitchell et al (1990) and Currin et al (1991). ■

Example 2.5. Another useful compactly supported correlation function is the Bohman correlation function (Gneiting (2002)). The Bohman correlation in one-dimension is defined to be

$$R(h|\theta) = \begin{cases} \left(1 - \frac{|h|}{\theta}\right)\cos\left(\frac{\pi|h|}{\theta}\right) + \frac{1}{\pi}\sin\left(\frac{\pi|h|}{\theta}\right), & |h| < \theta; \\ 0, & |h| \geq \theta \end{cases} \tag{2.4.10}$$

where $\theta > 0$. ■

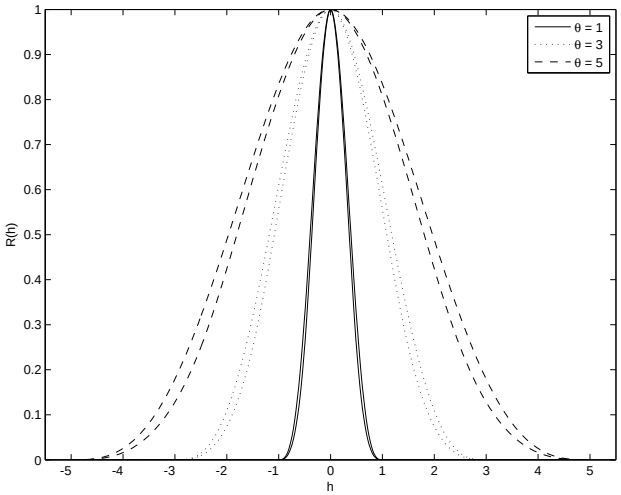


Fig. 2.3 Comparison of the Cubic and Bohman correlation functions for common θ.

Because they can result in zeroes in (many) off-diagonal elements of the correlation matrix of the training data, compactly supported correlation functions have been used the basis for prediction in such cases (see however, Kaufman et al (2011)). However, for the same θ these two functions are nearly identical as Figure 2.3 shows; as of the writing of this book there is not a great deal of difference that has been found in predictions when using the two functions.

2.4.5 Using the Correlation Function to Specify a GRF with Given Smoothness Properties

In practice we reduce the choice of a GRF to that of a covariance (or correlation) function whose realizations have desired prior smoothness characteristics. Hence we now turn attention to describing the relationship between the smoothness properties of a stationary GRF, $Z(\cdot)$, and the properties of its covariance function, $C(\cdot)$. To describe this relationship for general processes would require substantial space. By restricting attention to stationary GRFs we can provide a relatively concise overview. See Adler (1990), Abrahamsen (1997), or Stein (1999) for a discussion of these ideas for more general processes and for additional detail concerning the Gaussian process case.

There are several different types of "continuity" and "differentiability" that a process can possess. The definitions differ in their ease of application and the technical simplicity with which they are established. Given a particular property such as continuity at a point or differentiability over an interval, we would like to know that draws from a given random function model $Z(\cdot)$ have that property with probability one. For example, if Q is a property of interest, say continuity at the point $\boldsymbol{x}_0$, then we desire

$$P\{\omega : Z(\cdot, \omega) \text{ has property } Q\} = 1.$$

We term this *almost sure behavior* of the sample paths.

Subsection **??** introduced the widely-used concept of MS continuity. We saw an instance of the general fact that MS properties are relatively simple to prove, although they are not of direct interest in describing sample paths. Below we show that a slight strengthening of the conditions under which MS continuity holds guarantees almost sure continuity.

Recall that in Subsection **??** we stated that any stationary random function $Z(\cdot)$ on $\mathcal{X}$ having finite second moments is MS continuous on $\mathcal{X}$ provided that its correlation function is continuous at the origin, i.e., $R(\boldsymbol{h}) \to 1$ as $\boldsymbol{h} \to \boldsymbol{0}$. GRFs with either the cubic (2.4.9) or the power exponential (2.4.7) correlation functions are examples of such random functions.

Adler (1981) (page 60) shows that for the sample paths of stationary GRFs to be almost surely continuous, one need only add a condition requiring that $R(\boldsymbol{h})$ converge to unity sufficiently fast. For example, a consequence of his Theorem 3.4.1 is that, if $Z(\cdot)$ is a stationary GRF with correlation function $R(\cdot)$ that satisfies

$$1 - R(\boldsymbol{h}) \leq \frac{c}{|\log(\|\boldsymbol{h}\|_2)|^{1+\epsilon}} \quad \text{for all} \quad \|\boldsymbol{h}\|_2 < \delta \tag{2.4.11}$$

for some $c > 0$, some $\epsilon > 0$, and some $\delta < 1$, then $Z(\cdot)$ has almost surely continuous sample paths. MS continuity requires that $(1 - R(\boldsymbol{h})) \to 0$ as $\boldsymbol{h} \to 0$; the factor $|\log(\|\boldsymbol{h}\|_2)|^{1+\epsilon} \to +\infty$ as $\boldsymbol{h} \to 0$. Thus (2.4.11) holds provided that $1-R(\boldsymbol{h})$ converges to zero at least as fast as $|\log(\|\boldsymbol{h}\|_2)|^{1+\epsilon}$ diverges to $+\infty$. The product

$$[1 - R(\boldsymbol{h})] \times \left|\log(\|\boldsymbol{h}\|_2)\right|^{1+\epsilon}$$

is bounded for most correlation functions used in practice. In particular this is true for any power exponential correlation function with $0 < p \leq 2$. One can also use the spectral distribution to give sufficient conditions for almost sure continuity of sample paths. The standard conditions are stated in terms of the finiteness of the moments of the spectral distribution. For example, see Theorem 3.4.3 of Adler (1981) or Sections 9.3 and 9.5 of Cramér and Leadbetter (1967).

Conditions for almost sure continuity of the sample paths of nonstationary GRFs, $Z(\cdot)$, can be similarly expressed in terms of the rate at which

$$E\left\{|Z(\boldsymbol{x}_1) - Z(\boldsymbol{x}_2)|^2\right\}$$

converges to zero as $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \to 0$ (Adler (1981), Theorem 3.4.1).

As for continuity, a concept of mean square differentiability can be defined that describes the mean difference of the usual tangent slopes of a given process and a limiting "derivative process." Instead, here we directly discuss the parallel to almost sure continuity. Consider the individual sample draws $z(\boldsymbol{x}) = Z(\boldsymbol{x}, \omega)$, $\mathcal{X} \subset \mathbb{R}^d$, corresponding to specific outcomes $\omega \in \Omega$. Suppose that the j^{th} partial derivative of $Z(\boldsymbol{x}, \omega)$ exists for $j = 1, \ldots d$ and $\boldsymbol{x} \in \mathcal{X}$, i.e.,

$$\nabla_j Z(\boldsymbol{x}, \omega) = \lim_{\delta \to 0} \frac{Z(\boldsymbol{x} + \boldsymbol{e}_j \delta, \omega) - Z(\boldsymbol{x}, \omega)}{\delta}$$

exists where $\boldsymbol{e}_j$ denotes the unit vector in the j^{th} direction. Let

$$\boldsymbol{\nabla} Z(\boldsymbol{x}, \omega) = (\nabla_1 Z(\boldsymbol{x}, \omega), \ldots, \nabla_d Z(\boldsymbol{x}, \omega))$$

denote the vector of partial derivatives of $Z(\boldsymbol{x}, \omega)$, sometimes called the gradient of $Z(\boldsymbol{x}, \omega)$. We will state conditions on the covariance (correlation) function that guarantee that the sample paths are almost surely differentiable. The situation for higher order derivatives can be described in a similar manner, sample pathwise, for each ω.

As motivation for the condition given below, we observe the following heuristic calculation that gives the covariance of the derivative of $Z(\cdot)$. Fix $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in $\mathcal{X}$, then

$$\begin{aligned}
&\operatorname{Cov}\Big(\tfrac{1}{\delta_1}(Z(\boldsymbol{x}_1 + \boldsymbol{e}_j\delta_1) - Z(\boldsymbol{x}_1)) \,,\, \tfrac{1}{\delta_2}(Z(\boldsymbol{x}_2 + \boldsymbol{e}_j\delta_2) - Z(\boldsymbol{x}_2))\Big) \\
&\qquad = \frac{1}{\delta_1\delta_2}\Big\{C(\boldsymbol{x}_1 - \boldsymbol{x}_2 + \boldsymbol{e}_j(\delta_1 - \delta_2)) - C(\boldsymbol{x}_1 - \boldsymbol{x}_2 + \boldsymbol{e}_j\delta_1) \\
&\qquad\quad - C(\boldsymbol{x}_1 - \boldsymbol{x}_2 - \boldsymbol{e}_j\delta_2) + C(\boldsymbol{x}_1 - \boldsymbol{x}_2)\Big\} \\
&\qquad \to - \left.\frac{\partial^2 C(\boldsymbol{h})}{\partial h_j^2}\right|_{\boldsymbol{h} = \boldsymbol{x}_1 - \boldsymbol{x}_2}
\end{aligned} \tag{2.4.12}$$

as $\delta_1, \delta_2 \to 0$ when the second partial derivative of $C(\cdot)$ exists. These calculations motivate the fact that the covariance function of the partial derivatives of $Z(\cdot)$, if they

exist, are given by the partial derivatives of $C(\boldsymbol{h})$. Thus it should come as no surprise that to assure that a given Gaussian random field has, almost surely, differentiable draws, the conditions required are on the partial derivatives of the covariance function.

Formally, suppose

$$C_j^{(2)}(\boldsymbol{h}) \equiv \frac{\partial^2 C(\boldsymbol{h})}{\partial h_j^2}$$

exists and is continuous with $C_j^{(2)}(\mathbf{0}) \neq 0$; let $R_j^{(2)}(\boldsymbol{h}) \equiv C_j^{(2)}(\boldsymbol{h})/C_j^{(2)}(\mathbf{0})$ be the normalized version of $C_j^{(2)}(\cdot)$. Then almost surely $Z(\cdot)$ has j^{th} partial differentiable sample path, denoted $\nabla_j Z(\boldsymbol{x})$, provided $R_j^{(2)}(\cdot)$ satisfies (2.4.11). In this case $-C_j^{(2)}(\boldsymbol{h})$ is the covariance function and $R_j^{(2)}(\boldsymbol{h})$ is the correlation function of $\nabla_j Z(\boldsymbol{x})$.

Higher order $Z(\cdot)$ derivatives can be iteratively developed in the same way, although a more sophisticated notation must be introduced to describe the higher-order partial derivatives required of $C(\cdot)$. Conditions for nonstationary $Z(\cdot)$ can be determined from almost sure continuity conditions for nonstationary $Z(\cdot)$ (Adler (1981), Chapter 3).

We complete this section by illustrating the effects of changing the covariance parameters on the draws of several stationary GRFs that were introduced earlier and on one important additional family, the Matérn correlation function. In each case, the plot was obtained by linearly joining draws from an appropriate 20 or 40 dimensional multivariate normal distribution; hence the figures give the spirit, if not the detail, of the sample paths from the associated process. The interested reader can gain additional feel for stationary Gaussian processes by using the software of Kozintsev (1999) or Kozintsev and Kedem (2000) for generating two-dimensional Gaussian random fields (see the URL

`http://www.math.umd.edu/~bnk/CLIP/clip.gauss.htm`

for details).

Example 2.4. **(Continued–power exponential correlation function)** Figures 2.4 and 2.5 show the marginal effects of changing the shape parameter p and the scale parameter θ on the function draws from GRFs over $[0, 1]$ having the power exponential correlation function (2.4.7). *These figures, and those that illustrate the other GRFs that are discussed below, connect 20 points drawn from a multivariate normal distribution having the desired covariance matrix and so illustrate the spirit of the function draws, if not their fine detail.*

For powers $p < 2$, the sample paths are theoretically nondifferentiable and this can be seen in the bottom two panels of Figure 2.4. The sample paths for $p = 2.0$ are infinitely differentiable; the draws in the top panel of Figure 2.4 are very near the process mean of zero for $\theta = 1.0$. As shown in Figure 2.5, the number of local maxima and minima in sample paths is controlled by the scale parameter when $p = 2.0$. Figure 2.5 shows that as the scale parameter θ *decreases*, the correlations for each fixed pair of inputs decreases and the sample paths have increasing numbers of local maxima. This is true because the process exhibits less dependence for

"near-by" x and thus "wiggles" more like white noise, the case of uncorrelated $Z(\boldsymbol{x})$. As θ *increases*, the correlation for each pair of inputs increases and, as the correlation approaches unity, the draws become more nearly the constant zero, the process mean. In Figure 2.5 the most extreme case of this phenomenon is shown in the top panel where $(p, \theta) = (2.0, 0.50)$. ■

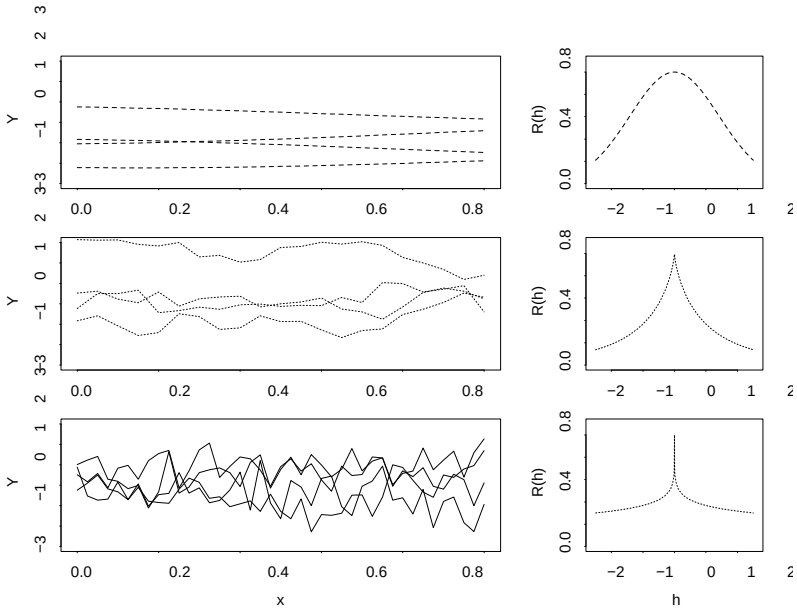


Fig. 2.4 The Effect of Varying the Power on the Sample Paths of a GRF with a Power Exponential Correlation Function. Four draws from a zero mean, unit variance GRF with the exponential correlation (2.4.7) having fixed $\theta \equiv 1.0$ with $p = 2.0$ (dashed lines), $p = 0.75$ (dotted lines), and $p = 0.20$ (solid lines).

Example 2.5. **(Continued–cubic correlation function)** Recall that the cubic correlation (and covariance) function (2.4.9) is twice continuously differentiable. Thus draws from a GRF with this correlation structure will be continuous and differentiable. Figure 2.6 shows draws from this process for different θ. As the scale parameter θ *decreases*, the domain where $R(h) = 0$ increases and hence the paths become more like white noise, i.e., having independent and identically distributed Gaussian components. As θ *increases*, the paths tend to become flatter with fewer local maxima and minima. ■

Example 2.6. The Matérn correlation function was introduced by Matérn in his thesis (Matérn (1960) or see the reprint Matérn (1986) and Vecchia (1988) for related work). This model has been used especially to describe the spatial and temporal

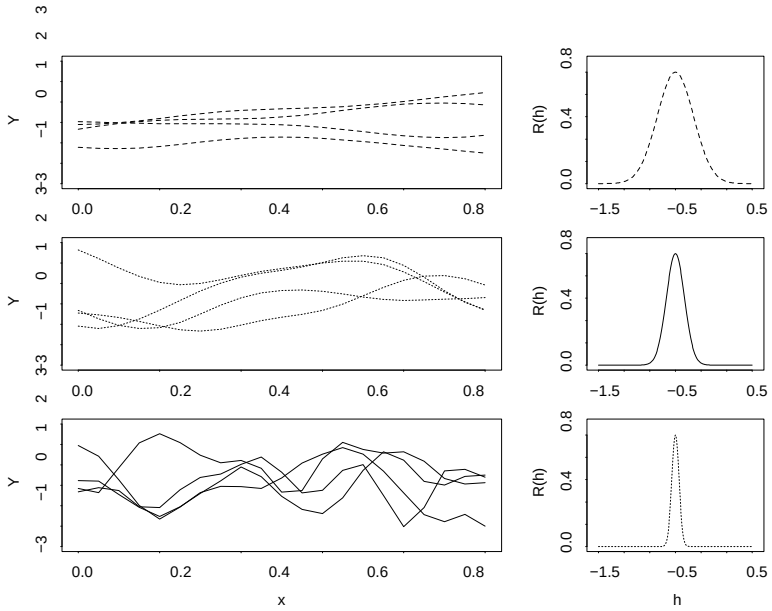


Fig. 2.5 The Effect of Varying the Scale Parameter on the Sample Paths of a GRF with a Power Exponential Correlation Function. Four draws from a zero mean, unit variance GRF with the exponential correlation function (2.4.6) (having fixed $p = 2.0$) for $\theta = 0.50$ (dashed lines), $\theta = 0.25$ (dotted lines), and $\theta = 0.10$ (solid lines).

variability in environmental data (see Rodríguez-Iturbe and Mejía (1974), Handcock and Stein (1993), Handcock and Wallis (1994), and especially Stein (1999)).

From the viewpoint of the spectral representation, the Matérn correlation function arises by choosing the t distribution as the spectral density. Given $\nu > 0$ and $\theta > 0$, use of the t density

$$f(w) = \frac{\Gamma(\nu + 1/2)}{\Gamma(\nu)\sqrt{\pi}} \left(\frac{4\nu}{\theta^2}\right)^{\nu} \frac{1}{\left(w^2 + \frac{4\nu}{\theta^2}\right)^{\nu+1/2}}$$

in spectral correlation formula (2.4.4) gives the two parameter correlation family

$$R(h) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{2\sqrt{\nu}\,|h|}{\theta}\right)^{\nu} K_{\nu}\left(\frac{2\sqrt{\nu}\,|h|}{\theta}\right), \tag{2.4.13}$$

where $K_{\nu}(\cdot)$ is the modified Bessel function of order ν. As is usual in the literature, we refer to (2.4.13) as the Matérn correlation function. The parameter θ is clearly a scale parameter for this family. The modified Bessel function arises as the solution of a certain class of ordinary differential equations (Kreyszig (1999)). In general, $K_{\nu}(t)$ is defined in terms of an infinite power series in t; when ν equals a half integer, i.e., $\nu = n + 1/2$ for $n \in \{0, 1, 2, \ldots\}$, then $K_{n+1/2}(\cdot)$ can be expressed as the finite

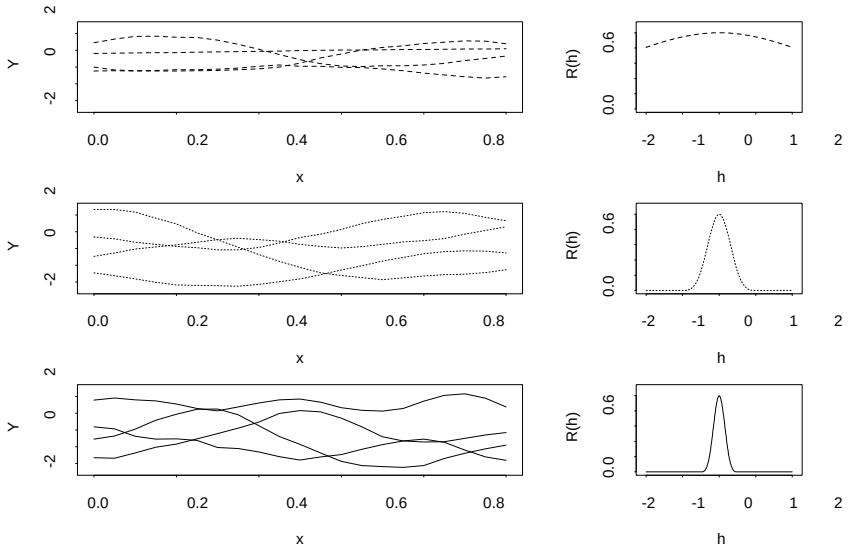


Fig. 2.6 The Effect of Varying the Scale Parameter on the Sample Paths of a GRF with a Cubic Correlation Function. Four draws from a zero mean, unit variance GRF with the cubic correlation function (2.4.9) for $\theta = 0.5$ (solid lines), $\theta = 1.0$ (dotted lines), and $\theta = 10.0$ (dashed lines). The corresponding correlation function is plotted to the right of each set of sample paths.

sum

$$K_{n+1/2}(t) = e^{-t}\sqrt{\frac{\pi}{2t}}\sum_{k=0}^{n}\frac{(n+k)!}{k!\,(n-k)!}\frac{1}{(2t)^k}.$$

The corresponding Matérn correlation function (2.4.13) is

$$e^{-2\sqrt{\nu}|h|/\theta}\left\{b_0\left(\frac{|h|}{\theta}\right)^n + b_1\left(\frac{|h|}{\theta}\right)^{n-1} + b_2\left(\frac{|h|}{\theta}\right)^{n-2} + \cdots + b_n\right\},$$

where the coefficients are given by

$$b_j = \frac{\sqrt{\pi}\,\nu^{(n-j)/2}}{4^j\Gamma(\nu)}\,\frac{(n+j)!}{j!\,(n-j)!}$$

for $j = 0, 1, \ldots$ where $\nu = n + 1/2$; the b_j depend on ν but not θ. For example, when $n = 0$ ($\nu = 1/2$),

$$K_{1/2}(t) = \sqrt{\pi}e^{-t}/\sqrt{2t} \text{ and so } R(h) = e^{-\sqrt{2}|h|/\theta},$$

which is a special case of the power exponential correlation function with $p = 1$ that was introduced earlier. Similarly, $R(h) \to e^{-(h/\theta)^2}$ as $\nu \to \infty$ so that this class of correlations includes the Gaussian correlation function in the limit.

The smoothness of functions drawn from a GRF with Matérn correlation depends on ν. Let $\lceil \nu \rceil$ denote the integer ceiling of ν, i.e., the smallest integer that is greater than or equal to ν. For example, $\lceil 3.2 \rceil = 4$ and $\lceil 3 \rceil = 3$. Then functions drawn from a GRF having the Matérn correlation have almost surely continuously differentiable sample draws of order $(\lceil \nu \rceil - 1)$. Thus we refer to ν as the smoothness parameter of the Matérn family (see Cramér and Leadbetter (1967)).

Products of the one-dimensional Matérn correlation function can be useful for modeling d-dimensional input responses. In this case, the family might include dimension specific scale parameters and a common smoothness parameter,

$$R(\boldsymbol{h}) = \prod_{i=1}^{d} \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i} \right)^{\nu} K_\nu \left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i} \right),$$

or dimension specific scale and smoothness parameters.

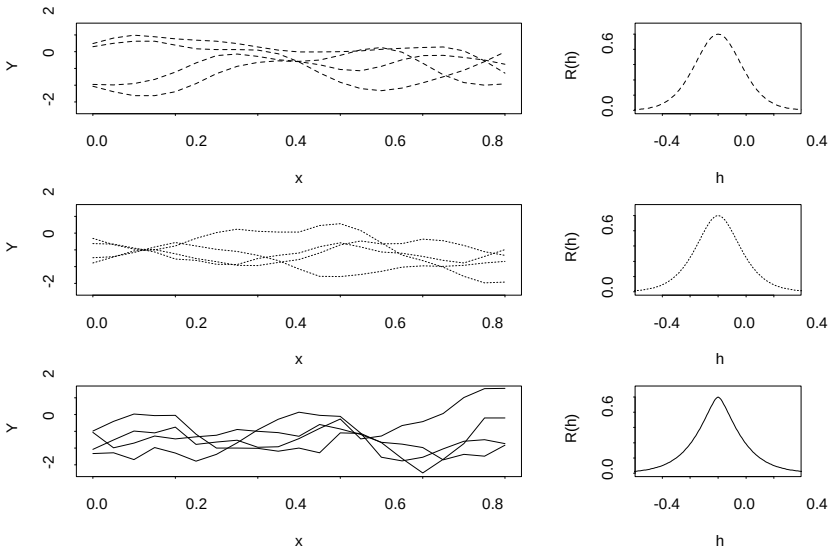


Fig. 2.7 The Effect of Varying the ν Parameter on the Sample Paths of a GRF with Matérn Correlation Function. Four draws from a zero mean, unit variance GRF with the Matérn correlation function (2.4.13) (having fixed $\theta = 0.25$) for $\nu = 1$ (solid lines), $\nu = 2.5$ (dotted lines), and $\nu = 5$ (dashed lines).

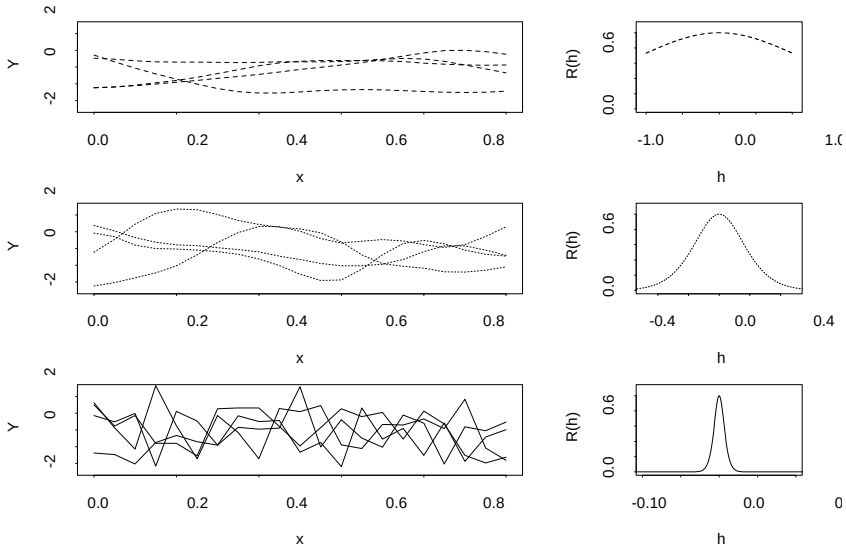


Fig. 2.8 The Effect of Varying the Scale Parameter on the Sample Paths of a GRF with Matérn Correlation Function. Four draws from a zero mean, unit variance GRF with the Matérn correlation function (2.4.13) (having fixed $\nu = 4$) for $\theta = 0.01$ (solid lines), $\theta = 0.25$ (dotted lines), and $\theta = 2.0$ (dashed lines).

We conclude by displaying sets of function draws from one-dimensional GRFs on $[0, 1]$ having different Matérn correlation functions to illustrate the effect of changing the scale and shape parameters.

Figure 2.7 fixes the scale parameter at $\theta = 0.25$ and varies $\nu \in \{1, 2.5, 5\}$. The draws clearly show the increase in smoothness as ν increases. As a practical matter, it is difficult for most observers to distinguish sample paths having 3 or 4 continuous derivatives from those that are infinitely differentiable. In contrast, Figure 2.8 fixes the smoothness parameter at $\nu = 4$ and varies $\theta \in \{0.01, 0.25, 2.0\}$. For fixed ν and $0 < h < 1.0$, the scaled range of $|h|/\theta$ varies substantially for different θ; $|h|/\theta$ ranges from 0.0 to 100 for $\theta = 0.01$ while this ratio only varies over 0.0 to 0.5 for $\theta = 2.0$. Notice that we use different h ranges for plotting $R(h)$ in Figure 2.8 to better illustrate the character of the correlation function near the origin. As θ decreases, the correlation function of any two fixed points decreases (to zero) and hence the sample paths "look" more like white noise. Thus the bottom panel of this figure plots a process with many more local maxima and minima than does the top panel. ■

2.4.6 Hierarchical Gaussian Random Field Models

While the examples above can provide guidance about the choice of a specific GRF prior for $y(\cdot)$, it will often be the case that the user will not be prepared to specify every detail of the GRF prior. For example, it will often be difficult to specify the correlation function of the GRF. A flexible alternative to the complete specification of a GRF is to use a *hierarchical* GRF prior model for $Y(\cdot)$. To describe this model, suppose that

$$Y(\boldsymbol{x}) = \sum_{j=1}^{p} f_j(\boldsymbol{x})\beta_j + Z(\boldsymbol{x}) = \boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{\beta} + Z(\boldsymbol{x}),$$

where $Z(\cdot)$ is a Gaussian random field with zero mean, variance σ_z^2, and correlation function $R(\cdot \,|\, \boldsymbol{\psi})$. Here $R(\cdot \,|\, \boldsymbol{\psi})$ denotes a parametric family of correlation functions. In a hierarchical model some (or all) of $\boldsymbol{\beta}$, σ_z^2, and $\boldsymbol{\psi}$ are not specified but rather a 2^{nd} stage distribution that describes expert opinion about the relative likelihood of the parameter values is specified.

To be specific, suppose it desired to place a 2^{nd} stage prior on all three parameters $\boldsymbol{\beta}, \sigma_z^2$, and $\boldsymbol{\psi}$. Sometimes this task is facilitated because the prior $[\boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\psi}]$ can be expressed in "pieces." Suppose that it is reasonable to assume that large scale location parameters $\boldsymbol{\beta}$ and the small scale variance, σ_z^2, are independent of the correlation parameters, $\boldsymbol{\psi}$. This means that

$$[\boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\psi}] = [\boldsymbol{\beta}, \sigma_z^2] \times [\boldsymbol{\psi}] = [\boldsymbol{\beta} \,|\, \sigma_z^2] \times [\sigma_z^2] \times [\boldsymbol{\psi}] \,.$$

The second equality is true because $[\boldsymbol{\beta}, \sigma_z^2] = [\boldsymbol{\beta} \,|\, \sigma_z^2] \times [\sigma_z^2]$ always holds. Thus the overall prior can be determined from these three pieces, which is often easier to do.

One complication with hierarchical models is that even when $[\boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\psi}]$ can be specified, it will usually be the case that the $Y(\boldsymbol{x})$ posterior cannot be expressed in closed form. Subsection 3.3.2.4 discusses the problem of computing the posterior mean in the context of various "empirical best linear unbiased predictors." See especially the discussion of "posterior mode empirical best linear unbiased predictors" beginning on page 73.

As an example, suppose that the input $\boldsymbol{x}$ is d-dimensional and that $R(\cdot \,|\, \boldsymbol{\psi})$ has the product Matérn correlation function

$$R(\boldsymbol{h} \,|\, \boldsymbol{\psi}) = \prod_{i=1}^{d} \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i} \right)^{\nu} K_\nu \left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i} \right) \tag{2.4.14}$$

with unknown common smoothness parameter and dimension-specific scale parameters; thus $\boldsymbol{\psi} = (\theta_1, \ldots, \theta_d, \nu)$. Consider specification of prior $[\boldsymbol{\psi} = (\theta_1, \ldots, \theta_d, \nu)]$. Suppose that any ν, $2 \le \nu \le 50$ is equally likely, which implies that the number of derivatives in each dimension is equally likely to range from 1 to 49. Given ν, 2^{nd} stage priors can be placed on each scale parameter by soliciting expert opinion about likelihood of correlation values between $Y(\boldsymbol{x}_1)$ and $Y(\boldsymbol{x}_2)$ where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ differ in exactly one coordinate direction. See Oakley (2002) for details and a case

study. There are other examples of the construction of 2^{nd} stage prior distributions for parameters, mostly in the environmental literature. For example, Handcock and Wallis (1994) build a prior distribution for correlation parameters in their space-time model of the mean temperature of a region of the northern United States.

The references in the previous paragraph describe what might be thought of as "informative" 2^{nd} stage priors. Again returning to the Matérn correlation function (2.4.14), it may be difficult to choose even the means and variances of the smoothness parameter and the scale parameters for specific dimensions, much less the $[\boldsymbol{\psi}]$ joint distribution. In such cases it is tempting to develop and use so-called "non-informative" 2^{nd} stage priors, which give "equal" weight to all the legitimate parameter values. The reader should be warned that there is not always agreement in the statistical community about what constitutes a non-informative prior, even for parameters having finite ranges. Furthermore not every choice of a non-informative 2^{nd} stage prior dovetails with the 1^{st} stage model to produce a legitimate prior for $y(\cdot)$ (see the important paper by Berger et al (2001)). More will said about non-informative 2^{nd} stage priors in Subsection 3.3.2.4 on page 73, which discusses "posterior mode empirical best linear unbiased predictors." Such predictors assume that a hierarchical GRF model is specified having parametric correlation function $R(\cdot\,|\,\boldsymbol{\psi})$ with unknown $\boldsymbol{\psi}$.

A third possible choice for a 2^{nd} stage parameter prior is a "conjugate" prior. Conjugate priors lead to closed-form posterior calculations, and are sometimes reasonable. Subsection 4.1.2 discusses conjugate and non-informative 2^{nd} stage $[\boldsymbol{\beta}]$ distributions (with σ_z^2 and $\boldsymbol{\psi}$ known). Subsection 4.1.3 gives the analogous conjugate and non-informative 2^{nd} stage $[\boldsymbol{\beta}, \sigma_z^2]$ distributions (with $\boldsymbol{\psi}$ known). These two subsections give closed-form expressions for the posterior of $Y(\boldsymbol{x})$ given the data.

2.5 Chapter Notes

There are many sources that provide detailed theoretical discussions of random functions, particularly the Gaussian random functions introduced in Subsections **??**–2.4.5 (**?**, **?**, **?**, and **?**, for example). It is not our purpose to present a complete account of the theory. Rather, we desire to give an overview of these models, to describe the relationship between the "correlation function" of stationary Gaussian random functions and the smoothness properties of its realizations $y(\boldsymbol{x})$, and to develop intuition about this relationship through a series of examples.

Chapter 3
Predicting Output from Computer Experiments

3.1 Introduction

This chapter discusses techniques for predicting the output of a computer model based on training data. A naíve view of this problem might regard it as being *point estimation* of a *fixed population quantity*. In contrast, *prediction* is the problem of providing a point guess of the realization of a *random variable*. The reason why prediction is the relevant methodology for the computer experiment application will be discussed in Section 3.2.

Knowing how to predict computer output is a prerequisite for answering most practical research questions involving computer experiments. A partial list of such problems is given in Section **??**. As an example, Section 5.3 will present a method for sequentially designing a computer experiment to find input conditions that maximize the computer output; this sequential design method uses the prediction methods developed in this chapter.

To introduce ideas, we initially consider the generic problem of predicting an arbitrary random variable Y_0 based on data $\boldsymbol{Y}^n = (Y_1, \ldots, Y_n)^\top$. When Y_0 and $\boldsymbol{Y}^n$ are dependent random quantities, it seems intuitive that $\boldsymbol{Y}^n$ contains information about Y_0. Hence it should be no surprise that the Y_0 predictors we introduce in this chapter depend on the joint distribution of Y_0 and $\boldsymbol{Y}^n$. Section 3.2 describes several Y_0 predictors based on various optimality and convenience criteria.

Section 3.3 applies these methods to predict $Y(\boldsymbol{x}_0)$, the computer output at $\boldsymbol{x}_0$, based on training data $(\boldsymbol{x}_i, Y(\boldsymbol{x}_i))$, $1 \leq i \leq n$. The reader who is familiar with regression methodology might think of using a flexible regression model as a predictor to solve this problem. For example, a cubic model in $\boldsymbol{x}$ might be considered a reasonably adaptable regression model. As an alternative, Section 3.3 applies the predictors developed in Section 3.2 to the computer experiment problem. A simulation study of the small-sample mean squared prediction error of cubic regression with the alternative predictors developed in Section 3.3 shows that cubic regression is vastly inferior to any of the Section 3.3 predictors.

3.2 Prediction Basics

3.2.1 Classes of Predictors

Consider a general setup in which we wish to predict a random variable Y_0 based on training data $\boldsymbol{Y}^n = (Y_1, \ldots, Y_n)^\top$. Let $\widehat{Y}_0 = \widehat{Y}_0(\boldsymbol{Y}^n)$ denote a generic predictor of the random variable Y_0 based on $\boldsymbol{Y}^n$. This book is concerned with three classes of Y_0 predictors:

- Predictors: $\widehat{Y}_0 = \widehat{Y}_0(\boldsymbol{Y}^n)$ having *unrestricted functional form* in $\boldsymbol{Y}^n$
- *Linear predictors* (LP): $\widehat{Y}_0 = a_0 + \sum_{i=1}^n a_i Y_i = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$, where $\boldsymbol{a}^\top = (a_1, \ldots, a_n)$
- *Linear unbiased predictors* (LUP): Linear predictors $\widehat{Y}_0 = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ that have the additional property of "unbiased" with respect to a given family $\mathcal{F}$ of (joint) distributions for $(Y_0, \boldsymbol{Y}^n)$

Definition The predictor $\widehat{Y}_0 = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ is *unbiased* for Y_0 with respect to the class of distributions $\mathcal{F}$ provided

$$E_F\left\{\widehat{Y}_0\right\} = E_F\left\{Y_0\right\} \tag{3.2.1}$$

for all $F \in \mathcal{F}$, where $E_F\{\cdot\}$ denotes expectation under the $F(\cdot)$ distribution for $(Y_0, \boldsymbol{Y}^n)$.

In (3.2.1), we emphasize that the unbiasedness of the predictor $\widehat{Y}_0$ depends on the class $\mathcal{F}$. Ordinarily we will suppress the subscript F in the notation $E_F\{\cdot\}$ when this distribution is clear from the context. As the size of the family $\mathcal{F}$ *increases*, the requirement (3.2.1) must hold for additional F and thus the set of LPs that are unbiased with respect to $\mathcal{F}$ cannot increase (and usually decreases).

Example 3.1. **(The form of LUPs for a location parameter model)** Suppose

$$Y_i = \beta_0 + \epsilon_i \tag{3.2.2}$$

for $0 \le i \le n$, where the $\{\epsilon_i\}$ are uncorrelated with mean zero and variance $\sigma_\epsilon^2 > 0$. This is the model for $Y_0, \ldots, Y_n$ that specifies the first two moments of $(Y_0, \boldsymbol{Y}^n)$. An alternate statement of this model is that $Y_0, \ldots, Y_n$ are uncorrelated with mean β_0 and positive variance. Suppose that $\mathcal{F}$ is defined to be the set of those distributions (3.2.2) where β_0 is a *given nonzero* value, but $\sigma_\epsilon^2 > 0$ is positive but otherwise unknown. Any

$$\widehat{Y}_0 = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$$

is an LP of Y_0.

Which of these LPs are unbiased? By definition, the linear predictor $\widehat{Y}_0$ is unbiased with respect to $\mathcal{F}$ provided

$$E\left\{\widehat{Y}_0\right\} = E\left\{a_0 + \sum_{i=1}^{n} a_i Y_i\right\} = a_0 + \beta_0 \sum_{i=1}^{n} a_i$$
$$\stackrel{\text{set}}{=} E\{Y_0\} = \beta_0 \tag{3.2.3}$$

for all $\sigma_\epsilon^2 > 0$. Because (3.2.3) is independent of σ_ϵ^2, the definition of unbiased is automatically true for $F \in \mathcal{F}$ as long as $(a_0, \boldsymbol{a})$ satisfies

$$a_0 + \beta_0 \sum_{i=1}^{n} a_i = \beta_0 \tag{3.2.4}$$

for the given β_0. One class of solutions of (3.2.4) is $a_0 = \beta_0$ and any $\boldsymbol{a}$ satisfying $\sum_{i=1}^n a_i = 0$; any such solution gives the (data independent) predictor $\widehat{Y}_0 = \beta_0$. Other LUPs result by choosing $a_0 = 0$ and any $\boldsymbol{a}$ for which $\sum_{i=1}^n a_i = 1$; in particular, the sample mean of $Y_1, \ldots, Y_n$ is the LUP of Y_0 corresponding to $a_1 = \cdots = a_n = 1/n$ (and $a_0 = 0$).

Now consider the (enlarged) family $\mathcal{F}$ of moment models corresponding to (3.2.2) where β_0 is an *unspecified* real number *and* σ_ϵ^2 is positive but unknown. In this case the linear predictor $\widehat{Y}_0$ is unbiased with respect to $\mathcal{F}$ provided (3.2.4) holds for all $\beta_0 \in \mathbb{R}$ and for all $\sigma_\epsilon^2 > 0$. This condition holds if and only if $a_0 = 0$ and $\sum_{i=1}^n a_i = 1$ (the necessity of the first follows by considering the case $\beta_0 = 0$ and the second from $\beta_0 \neq 0$). Thus the sample mean of the training data is an LUP of β_0 but no constant predictor is an LUP. This example illustrates the general fact that every LUP with respect to $\mathcal{F}$ is also an LUP with respect to subfamilies of $\mathcal{F}$. ■

3.2.2 Best MSPE Predictors

Historically, the most widely used prediction criterion is the *mean squared prediction error* (MSPE). When $F(\cdot)$ is the joint distribution of $(Y_0, \boldsymbol{Y}^n)$, the MSPE of $\widehat{Y}_0 = \widehat{Y}_0(\boldsymbol{Y}^n)$ is

$$\text{MSPE}(\widehat{Y}_0, F) \equiv E_F\{(\widehat{Y}_0 - Y_0)^2\}. \tag{3.2.5}$$

Definition The predictor $\widehat{Y}_0$ of Y_0 is a *minimum MSPE predictor* at F provided

$$\text{MSPE}(\widehat{Y}_0, F) \leq \text{MSPE}(Y_0^\star, F) \tag{3.2.6}$$

for any alternative predictor $Y_0^\star$.

Minimum MSPE predictors are also called *best MSPE predictors*. Predictors of practical importance will simultaneously minimize the MSPE for *many* distributions F.

The fundamental theorem of prediction shows that the conditional mean of Y_0 given $\boldsymbol{Y}^n$ is the minimum MSPE predictor of Y_0 based on $\boldsymbol{Y}^n$.

Theorem 3.1. Suppose that $(Y_0, \boldsymbol{Y}^n)$ has a joint distribution F for which the conditional mean of Y_0 given $\boldsymbol{Y}^n$ exists. Then

$$\widehat{Y}_0 = E\{Y_0 \mid \boldsymbol{Y}^n\}$$

is the best MSPE predictor of Y_0.

Proof: Fix an arbitrary unbiased predictor $Y_0^\star = Y_0^\star(\boldsymbol{Y}^n)$,

$$\begin{aligned}
\mathrm{MSPE}(Y_0^\star, F) &= E_F\{(Y_0^\star - Y_0)^2\} \\
&= E_F\left\{(Y_0^\star - \widehat{Y}_0 + \widehat{Y}_0 - Y_0)^2\right\} \\
&= E_F\left\{(Y_0^\star - \widehat{Y}_0)^2\right\} + \mathrm{MSPE}(\widehat{Y}_0, F) \\
&\quad + 2E_F\left\{(Y_0^\star - \widehat{Y}_0)(\widehat{Y}_0 - Y_0)\right\} \\
&\geq \mathrm{MSPE}\left(\widehat{Y}_0, F\right) \\
&\quad + 2E_F\left\{\left(Y_0^\star - \widehat{Y}_0\right)\left(\widehat{Y}_0 - Y_0\right)\right\} \qquad (3.2.7) \\
&= \mathrm{MSPE}\left(\widehat{Y}_0, F\right),
\end{aligned}$$

where the final equality holds because

$$\begin{aligned}
E_F\left\{\left(Y_0^\star - \widehat{Y}_0\right)\left(\widehat{Y}_0 - Y_0\right)\right\} &= E_F\left\{\left(Y_0^\star - \widehat{Y}_0\right) E_F\left\{\left(\widehat{Y}_0 - Y_0\right) \mid \boldsymbol{Y}^n\right\}\right\} \\
&= E_F\left\{\left(Y_0^\star - \widehat{Y}_0\right)\left(\widehat{Y}_0 - E_F\{Y_0 \mid \boldsymbol{Y}^n\}\right)\right\} \\
&= E_F\left\{\left(Y_0^\star - \widehat{Y}_0\right) \times 0\right\} \\
&= 0. \quad \square
\end{aligned}$$

There are two interesting properties of the best MSPE predictor that can be seen from the proof of Theorem 3.1. The first is that the conditional mean $\widehat{Y}_0$ is essentially the unique best MSPE predictor in many cases that arise in practice. This is because $\mathrm{MSPE}(\widehat{Y}_0, F)$ and $\mathrm{MSPE}(Y_0^\star, F)$ are equal if and only if equality holds in (3.2.7), which occurs when $\widehat{Y}_0 = Y_0^\star$ almost everywhere. The second is that $\widehat{Y}_0 = E\{Y_0 \mid \boldsymbol{Y}^n\}$ must be *unbiased* with respect to the model F for $(Y_0, \boldsymbol{Y}^n)$ because

$$E\{\widehat{Y}_0\} = E\{E\{Y_0 \mid \boldsymbol{Y}^n\}\} = E\{Y_0\}.$$

Example 3.1. **(Continued–best MSPE predictors)** Consider finding the minimum MSPE predictor $\widehat{Y}_0 = E\{Y_0 \mid \boldsymbol{Y}^n\}$ when the components of $(Y_0, \boldsymbol{Y}^n)$ are not merely uncorrelated but are independent $N(\beta_0, \sigma_\epsilon^2)$ random variables. By the independence of $Y_0, Y_1, \ldots, Y_n$, the conditional distribution $[Y_0 \mid \boldsymbol{Y}^n]$ is simply the $N(\beta_0, \sigma_\epsilon^2)$ marginal distribution of Y_0. In particular,

$$\widehat{Y}_0 = E\{Y_0 \mid \boldsymbol{Y}^n\} = \beta_0$$

is the best MSPE predictor. Notice that this minimum MSPE predictor changes with β_0 and thus is specific to this particular $(Y_0, \boldsymbol{Y}^n)$ joint distribution.

Now consider a more interesting two-stage model for the distribution of $(Y_0, \boldsymbol{Y}^n)$. Assume that σ_ϵ^2 is known and that the distribution specified in the previous paragraph is the first-stage (conditional) distribution of $(Y_0, \boldsymbol{Y}^n)$ given β_0, denoted by $[Y_0, \boldsymbol{Y}^n \,|\, \beta_0]$. Combine this first-stage distribution with the non-informative second-stage distribution

$$[\beta_0] \propto 1$$

for β_0. While improper priors need not produce proper posterior distributions, in this case one can show

$$[Y_0, \boldsymbol{Y}^n] = \int [Y_0, \boldsymbol{Y}^n \,|\, \beta_0]\, [\beta_0]\, d\beta_0$$

gives a *proper* joint distribution of $(Y_0, \boldsymbol{Y}^n)$. Using this $(Y_0, \boldsymbol{Y}^n)$ distribution, the conditional distribution

$$[Y_0 \,|\, \boldsymbol{Y}^n = \boldsymbol{Y}^n] \sim N_1\left[\overline{\boldsymbol{Y}}_n, \sigma_\epsilon^2\left(1 + \frac{1}{n}\right)\right]$$

can be calculated where $\overline{\boldsymbol{Y}}_n = (\sum_{i=1}^n y_i)/n$ is the sample mean of the training data. It follows that, for this two-stage model, the minimum MSPE predictor of Y_0 is $\widehat{Y_0} = (\sum_{i=1}^n Y_i)/n$. ■

Example 3.2. **(More best MSPE predictors)** Consider the regression model developed in Chapter 2 in which

$$Y_i \equiv Y(\boldsymbol{x}_i) = \sum_{j=1}^{p} f_j(\boldsymbol{x}_i)\beta_j + Z(\boldsymbol{x}_i) = \boldsymbol{f}^\top(\boldsymbol{x}_i)\boldsymbol{\beta} + Z(\boldsymbol{x}_i) \tag{3.2.8}$$

for $0 \le i \le n$, where the $\{f_j(\cdot)\}$ are known regression functions, $\boldsymbol{\beta}$ is a *given* nonzero $p \times 1$ vector, and $Z(\boldsymbol{x})$ is a zero mean stationary Gaussian process with dependence specified by the covariance

$$\mathrm{Cov}\{Z(\boldsymbol{x}_i), Z(\boldsymbol{x}_j)\} = \sigma_z^2 R(\boldsymbol{x}_i - \boldsymbol{x}_j)$$

for some *known* correlation function $R(\cdot)$ (see Section **??**). Then the joint distribution of $Y_0 = Y(\boldsymbol{x}_0)$ and $\boldsymbol{Y}^n = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n))^\top$ is the multivariate normal distribution

$$\begin{pmatrix} Y_0 \\ \boldsymbol{Y}^n \end{pmatrix} \sim N_{1+n}\left[\begin{pmatrix} \boldsymbol{f}_0^\top \\ \boldsymbol{F} \end{pmatrix}\boldsymbol{\beta}, \sigma_z^2 \begin{pmatrix} 1 & \boldsymbol{r}_0^\top \\ \boldsymbol{r}_0 & \boldsymbol{R} \end{pmatrix}\right], \tag{3.2.9}$$

where $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$ is the $p \times 1$ vector of regressors at $\boldsymbol{x}_0$, $\boldsymbol{F}$ is the $n \times p$ matrix of regressors having $(i, j)th$ element $f_j(\boldsymbol{x}_i)$ for $1 \le i \le n, 1 \le j \le p, \boldsymbol{\beta}$ is a $p \times 1$ vector of unknown regression coefficients, and the $n \times 1$ vector $\boldsymbol{r}_0 = (R(\boldsymbol{x}_0 - \boldsymbol{x}_1), \ldots, R(\boldsymbol{x}_0 - \boldsymbol{x}_n))^\top$ and $n \times n$ matrix $\boldsymbol{R} = (R(\boldsymbol{x}_i - \boldsymbol{x}_j))$ are defined in terms of the correlation function $R(\cdot)$. Assuming that the design matrix $\boldsymbol{F}$ is of full column rank p and that $\boldsymbol{R}$ is positive definite, Theorems 3.1 and B.2 show that

$$\widehat{Y_0} = E\{Y_0 \mid \boldsymbol{Y}^n\} = \boldsymbol{f}_0^\top \boldsymbol{\beta} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\beta}) \tag{3.2.10}$$

is the best MSPE predictor of Y_0.

The class of distributions $\mathcal{F}$ for which (3.2.10) is the minimum MSPE predictor is again embarrassingly small. In addition to $\widehat{Y_0}$ depending on the multivariate normality of $(Y_0, \boldsymbol{Y}^n)$, it also depends on *both* $\boldsymbol{\beta}$ and the specific correlation function $R(\cdot)$. Thus the best MSPE predictor changes when either $\boldsymbol{\beta}$ or $R(\cdot)$ changes, however, $\widehat{Y_0}$ is the same for all $\sigma_z^2 > 0$.

As a final illustration, consider finding the minimum MSPE predictor of $Y(\boldsymbol{x}_0)$ based the following two-stage model for the regression data $(\boldsymbol{x}_i, Y(\boldsymbol{x}_i))$, $0 \le i \le n$. Suppose that (3.2.9) specifies the conditional distribution of $(Y_0, \boldsymbol{Y}^n)$ given $\boldsymbol{\beta}$ as the first stage of a two-stage model. (Assuming σ_z^2 is known, say, although this is not needed). The second stage of the model puts an arbitrary prior on $(\boldsymbol{\beta}, \sigma_z^2)$. The best MSPE predictor of Y_0 is

$$\begin{aligned}\widehat{Y_0} = E\{Y_0 \mid \boldsymbol{Y}^n\} &= E\{E\{Y_0 \mid \boldsymbol{Y}^n, \boldsymbol{\beta}\} \mid \boldsymbol{Y}^n\} \\ &= E\left\{\boldsymbol{f}_0^\top \boldsymbol{\beta} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\beta}) \,\middle|\, \boldsymbol{Y}^n\right\}\end{aligned}$$

and the last expectation is with respect to the conditional distribution of $\boldsymbol{\beta}$ given $\boldsymbol{Y}^n$. Thus

$$\widehat{Y_0} = \boldsymbol{f}_0^\top E\{\boldsymbol{\beta} \mid \boldsymbol{Y}^n\} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}E\{\boldsymbol{\beta}|\boldsymbol{Y}^n\}) \tag{3.2.11}$$

is the minimum MSPE predictor of Y_0 for *any* two-stage model whose first stage is given by (3.2.9) and has arbitrary second stage $\boldsymbol{\beta}$ prior for which $E\{\boldsymbol{\beta}|\boldsymbol{Y}^n\}$ exists.

Of course, the explicit formula for $E\{\boldsymbol{\beta}|\boldsymbol{Y}^n\}$, and hence $\widehat{Y_0}$, depends on the $\boldsymbol{\beta}$ prior. For example, when $\boldsymbol{\beta}$ has the non-informative prior, $[\boldsymbol{\beta}] \propto 1$, the conditional distribution $[\boldsymbol{\beta}|\boldsymbol{Y}^n]$ can be derived by observing

$$\begin{aligned}[\boldsymbol{\beta}|\boldsymbol{Y}^n = \boldsymbol{y}^n] &\propto [\boldsymbol{y}^n|\boldsymbol{\beta}]\,[\boldsymbol{\beta}] \\ &\propto \exp\left\{-\frac{1}{2\sigma_z^2}(\boldsymbol{y}^n - \boldsymbol{F}\boldsymbol{\beta})^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^n - \boldsymbol{F}\boldsymbol{\beta})\right\} \times 1 \\ &\propto \exp\left\{-\frac{1}{2\sigma_z^2}\left(\boldsymbol{\beta}^\top \boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F}\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top \boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{y}^n\right)\right\} \\ &= \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{A}^{-1}\boldsymbol{\beta} + \boldsymbol{\nu}^\top \boldsymbol{\beta}\right\}, \text{ say,}\end{aligned}$$

where $\boldsymbol{A}^{-1} = \boldsymbol{F}^\top(\sigma_z^2\boldsymbol{R})^{-1}\boldsymbol{F}$ and $\boldsymbol{\nu} = \boldsymbol{F}^\top(\sigma_z^2\boldsymbol{R})^{-1}\boldsymbol{y}^n$. Notice that $\text{rank}(\boldsymbol{A}) = p$ under the continuing assumption that $\boldsymbol{F}$ has full column rank p. Applying (B.1.2) of Appendix B gives

$$[\boldsymbol{\beta}|\boldsymbol{Y}^n] \sim N_p\left[(\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{Y}^n, \sigma_z^2(\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\right]$$

because the σ_z^2 terms cancel in the expression for the mean of $\boldsymbol{\beta}|\boldsymbol{Y}^n$. Thus the best MSPE predictor of Y_0 under this two-stage model is

$$\widehat{Y_0} = \boldsymbol{f}_0^\top \widehat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right), \tag{3.2.12}$$

where $\widehat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{Y}^n$. ■

There are at least three useful ways of thinking about the predictor (3.2.12). The first way is to regard (3.2.12) as the sum of the regression predictor of $\boldsymbol{f}_0^\top \widehat{\boldsymbol{\beta}}$ plus the "correction" $\boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right)$. The second way of viewing (3.2.12) is as a function of the training data $\boldsymbol{Y}^n$; this viewpoint is important for describing the statistical properties of $\widehat{Y_0}$. The final method of examining formula (3.2.12) is as a function of $\boldsymbol{x}_0$, the point at which the prediction is to be made. The remainder of this subsection and Example 3.3 considers the nature of the correction in $\widehat{Y_0}$. We will return to the latter two methods of thinking about $\widehat{Y_0}$ in Section 3.3.

The correction term in (3.2.12) is $\boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right)$, which is a linear combination of the residuals $\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}$ based on the model (3.2.8) with prediction point specific coefficients, i.e.,

$$\boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right) = \sum_{i=1}^{n} c_i(\boldsymbol{x}_0) \left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right)_i, \tag{3.2.13}$$

where the weight $c_i(\boldsymbol{x}_0)$ is the i^{th} element of $\boldsymbol{R}^{-1}\boldsymbol{r}_0$ and $\left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right)_i$ is the i^{th} residual based on the fitted model.

Example 3.3. To illustrate the regression and correction terms in the predictor (3.2.12), suppose the true unknown curve is the one-dimensional dampened cosine function

$$y(x) = e^{-1.4x} \cos(7\pi x/2)$$

over $0 \le x \le 1$ (see the top panel in Figure 3.1). We use a seven point training data set (also shown in Figure 3.1). The training data locations x_i were determined by selecting x_1 at random in the interval $[0\,,\, 1/7]$ and then adding $i/7$ to x_1 for $1 \le i \le 6$ to obtain six additional points. These next six x_i are equally spaced and located in the intervals $[1/7,\, 2/7], \ldots, [6/7,\, 1]$. The choice of a design of the computer experiment will be discussed in Chapters 4 and 5.

Consider prediction of $y(\cdot)$ based on the stationary stochastic Gaussian process

$$Y(x) = \beta_0 + Z(x),$$

where $Z(\cdot)$ has zero mean, variance σ_z^2, (Gaussian) correlation function

$$R(h) = e^{-136.1 \times h^2}.$$

Here $\boldsymbol{F} = \mathbf{1}_7$ is a 7×1 column vector of ones and, by (3.2.13), the predictor (3.2.12) is

$$\widehat{Y}(x_0) = \widehat{\beta_0} + \sum_{i=1}^{7} c_i(x_0) \left(Y_i - \widehat{\beta_0} \right)$$

when viewed as a function of x_0 where $\{x_i\}_{i=1}^7$ are the training data and $(Y_i - \widehat{\beta_0})$ is the i^{th} residual from fitting the constant model. In this case, the regression predictor is $\widehat{\beta_0}$.

Consider specifically the prediction of $y(x_0)$ at $x_0 = 0.55$. The seven residuals $Y_i - \widehat{\beta_0}$ and their associated weights $c_i(0.55)$, $1 \le i \le 7$, are plotted in Figure 3.1. Notice that (1) the weights can be positive or negative and (2) the correction to the

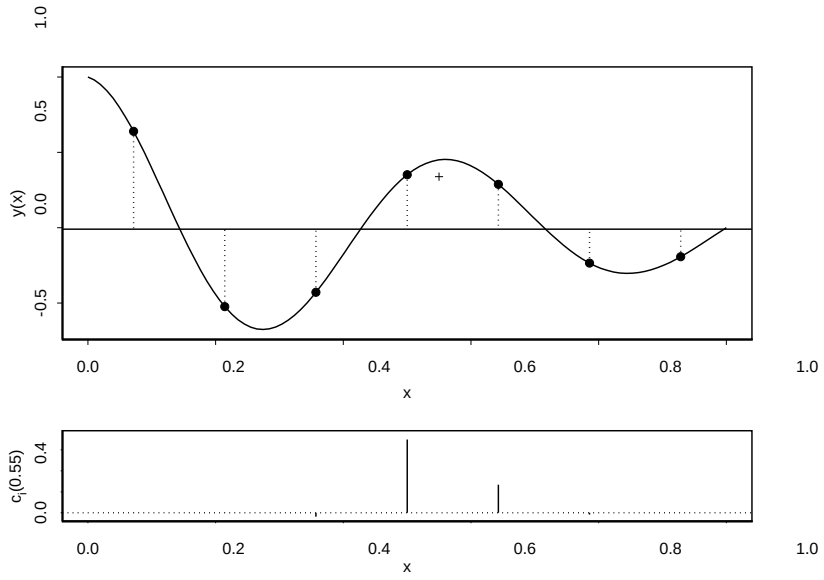


Fig. 3.1 The top panel shows the true curve $y(x) = \exp\{-1.4x\} \times \cos(3.5\pi x)$ (solid line); the seven point input training data (dots); the BLUP at $x_0 = 0.55$ (cross); and the residuals, $Y_i - \widehat{\beta_0}$, (vertical dotted lines). The bottom panel plots the *weight* at each training data point as a line segment of length $|c_i(0.55)|$ from the origin with negative $c_i(0.55)$ plotted downward and positive $c_i(0.55)$ plotted upward.

regression $\widehat{\beta_0}$ is based primarily on the residuals at the two training sites nearest to $x_0 = 0.55$; in fact, the three weights for the three training data points that are furthest from $x_0 = 0.55$ are indistinguishable from zero. ■

Returning to the general discussion of the correction $\boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}})$, we show that this term forces the predictor to *interpolate* the training data. To see why this is the case, suppose that $\boldsymbol{x}_0 = \boldsymbol{x}_i$ for some fixed i, $1 \le i \le n$. Then $\boldsymbol{f}_0 = \boldsymbol{f}^\top(\boldsymbol{x}_i)$ and

$$\boldsymbol{r}_0^\top = (\, R(\boldsymbol{x}_i - \boldsymbol{x}_1), R(\boldsymbol{x}_i - \boldsymbol{x}_2), \ldots, R(\boldsymbol{x}_i - \boldsymbol{x}_n)\,)$$

which is the i^{th} row of $\boldsymbol{R}$. Thus $\boldsymbol{R}^{-1}\boldsymbol{r}_0 = (0, \ldots, 0, 1, 0, \ldots, 0)^\top = \boldsymbol{e}_i$, the i^{th} unit vector, because this product is the i^{th} column of $\boldsymbol{R}^{-1}\boldsymbol{R} = \boldsymbol{I}_n$, the $n \times n$ identity

matrix. Hence

$$\boldsymbol{r}_0^\top \boldsymbol{R}^{-1}\left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right) = \boldsymbol{e}_i^\top \left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right) = Y_i - \boldsymbol{f}^\top(\boldsymbol{x}_i)\widehat{\boldsymbol{\beta}}$$

and so

$$\widehat{Y}(\boldsymbol{x}_0) = \boldsymbol{f}^\top(\boldsymbol{x}_i)\widehat{\boldsymbol{\beta}} + \left(Y_i - \boldsymbol{f}^\top(\boldsymbol{x}_i)\widehat{\boldsymbol{\beta}}\right) = Y_i\,.$$

Although we focus on the case of nonzero dependence in this book, we note that the argument above shows that for regression data with white noise (independent) measurement errors added to the mean of each observation, i.e., when the $Y_1, \ldots, Y_n$ are independent with Y_i having mean $\boldsymbol{f}^\top(\boldsymbol{x}_i)\boldsymbol{\beta}$, then $\boldsymbol{r}_0 = (0, \ldots, 0)^\top$. In this case the best MSPE predictor, expression (3.2.12), reduces to $\widehat{Y}(\boldsymbol{x}_0) = \boldsymbol{f}^\top(\boldsymbol{x}_0)\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{x}_0 \neq \boldsymbol{x}_i$ where $\widehat{\boldsymbol{\beta}}$ is the ordinary least squares estimator of the mean of $Y(\boldsymbol{x}_0)$ and $\widehat{Y}(\boldsymbol{x}_0) = Y_i$ for $\boldsymbol{x}_0 = \boldsymbol{x}_i$. Thus the best MSPE predictor *interpolates* the data but has discontinuities at each of the data points.

Expression (3.2.12) is the basis for most predictors used in computer experiments. The next subsection shows that (3.2.12) has additional optimality properties that help explain its popularity. Before beginning this topic, we present a final example to show that best MSPE predictors need *not* be a linear function of the training data lest the previous (Gaussian model) examples, where the predictors are all linear in the data, suggest otherwise to the reader.

Example 3.4. Suppose that (Y_0, Y_1) has the joint distribution given by the density

$$f(y_0, y_1) = \begin{cases} 1/y_1^2\,, \; 0 < y_1 < 1, 0 < y_0 < y_1^2 \\ 0\,, \quad \text{otherwise.} \end{cases}$$

Then it is straightforward to calculate that the conditional distribution of Y_0 given $Y_1 = y_1$ is uniform over the interval $(0, y_1^2)$. Hence the best MSPE predictor of Y_0 is the center of this interval, i.e., $\widehat{Y}_0 = E\{Y_0|Y_1\} = Y_1^2/2$ which is nonlinear in Y_1. In contrast, the minimum MSPE *linear unbiased* predictor of Y_0 is that $a_0 + a_1 Y_1$ which minimizes $E\{(a_0 + a_1 Y_1 - Y_0)^2\}$ among those (a_0, a_1) that satisfy the unbiasedness requirement $E\{a_0 + a_1 Y_1\} = E\{Y_0\}$. Unbiasedness leads to the restriction

$$a_0 + a_1 \frac{1}{2} = \frac{1}{6} \quad \text{or} \quad a_0 = \frac{1}{6} - a_1 \frac{1}{2}.$$

Applying calculus to minimize the MSPE

$$E\left\{\left(\left(\frac{1}{6} - a_1\frac{1}{2}\right) + a_1 Y_1 - Y_0\right)^2\right\}$$

(expressed in terms of a_1) shows that $a_1 = 1/2$ (and $a_0 = 1/6 - a_1/2 = -1/12$), i.e., $\widehat{Y}_0^L = -\frac{1}{12} + \frac{1}{2}Y_1$ is the minimum MSPE linear unbiased predictor of Y_0.

As Figure 3.2 shows, the predictors $\widehat{Y}_0$ and $\widehat{Y}_0^L$ are very close over their $(0, 1)$ domain. The MSPE of $\widehat{Y}_0$ is obtained from

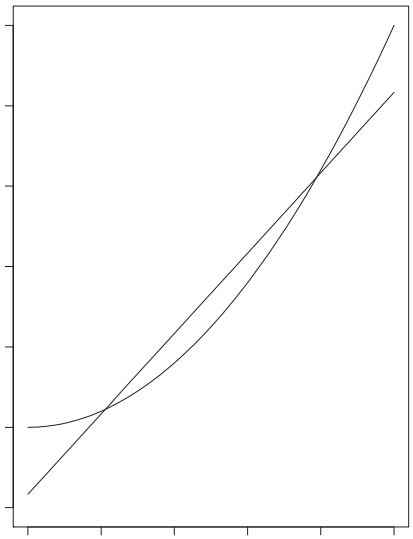


Fig. 3.2 The predictors $\widehat{Y}_0$ and $\widehat{Y}_0^L$ based on $y_1 \in (0, 1)$

$$\begin{aligned} E\left\{\left(Y_0 - Y_1^2/2\right)^2\right\} &= E\left\{E\left\{(Y_0 - Y_1^2/2)^2 \middle| Y_1\right\}\right\} \\ &= E\left\{\mathrm{Var}\{Y_0|Y_1\}\right\} \\ &= E\left\{Y_1^2/12\right\} \\ &= 1/60 \cong 0.01667. \end{aligned} \tag{3.2.14}$$

The inner term $Y_1^2/12$ in (3.2.14) is the variance of the uniform distribution over $(0, y_1^2)$. A similar calculation gives the MSPE of $\widehat{Y}_0^L$ to be 0.01806 which is greater than the MSPE of the unconstrained predictor, as theory dictates, but the difference is small, as Figure 3.2 suggests. ■

3.2.3 Best Linear Unbiased MSPE Predictors

As we have seen, minimum MSPE predictors depend in detail on the joint distribution of the training data and Y_0; this criterion typically leads to optimality within a very restricted class of competing predictors. In an attempt to find predictors that are optimal for a broader class of models, we focus on the two simpler types of predictors that were introduced in Example 3.4. Firstly we consider the class of Y_0

predictors that are linear in $\boldsymbol{Y}^n$, and secondly the class of predictors that are *both* linear and unbiased for Y_0.

The predictor $\widehat{Y}_0$ is a *minimum MSPE linear predictor* of Y_0 at F provided $\widehat{Y}_0$ is linear and

$$\text{MSPE}(\widehat{Y}_0, F) \leq \text{MSPE}(Y_0^\star, F) \tag{3.2.15}$$

for any other linear predictor $Y_0^\star$. Minimum MSPE linear predictors are sometimes called *best linear predictors* (BLPs).

Restricting further the class of predictors to those that are both linear and unbiased, one can again seek optimal MSPE predictors. To apply such a strategy, one must first determine which linear predictors are unbiased. Recall that unbiasedness is determined with respect to a family $\mathcal{F}$ of distributions. In the computer experiment literature, the emphasis is on finding a linear predictor $\widehat{Y}_0 = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ that is unbiased with respect to every F in some family of distributions $\mathcal{F}$ and simultaneously minimizes the MSPE at F in the same family $\mathcal{F}$. Given $\mathcal{F}$, a predictor $\widehat{Y}_0 = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ which is unbiased for $\mathcal{F}$ that satisfies (3.2.15) for $F \in \mathcal{F}$ is said to be *minimum MSPE linear unbiased* or simply a *best linear unbiased predictor* (BLUP).

Example 3.5. Consider best linear unbiased prediction for the nonparametric location parameter model (3.2.2) introduced in Example 3.1 where β_0 is *fixed*, i.e., for the family of distributions $\mathcal{F} = \mathcal{F}(\beta_0)$. Recall that $\widehat{Y}_0 = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ is unbiased provided $a_0 + \beta_0 \sum_{i=1}^n a_i = \beta_0$. The MSPE of the unbiased predictor $\widehat{Y}_0 = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ is

$$\begin{aligned} E\left\{\left(a_0 + \sum_{i=1}^n a_i Y_i - Y_0\right)^2\right\} &= E\left\{\left(a_0 + \sum_{i=1}^n a_i(\beta_0 + \epsilon_i) - \beta_0 - \epsilon_0\right)^2\right\} \\ &= \left(a_0 + \beta_0 \sum_{i=1}^n a_i - \beta_0\right)^2 \\ &\quad + \sigma_\epsilon^2 \times \sum_{i=1}^n a_i^2 + \sigma_\epsilon^2 \\ &= \sigma_\epsilon^2 \times \left(1 + \sum_{i=1}^n a_i^2\right) \qquad (3.2.16) \\ &\geq \sigma_\epsilon^2 \qquad (3.2.17) \end{aligned}$$

Equality holds in (3.2.16) because $\widehat{Y}_0$ is unbiased and equality occurs in (3.2.17) if and only if $a_0 = \beta_0$ and $a_1 = \ldots = a_n = 0$, which shows that

$$\widehat{Y}_0 = \beta_0$$

is the unique BLUP for model $\mathcal{F}$. For this example, as for previous examples that determined various types of best MSPE predictors, the BLUP depends heavily on $\mathcal{F}$.

Now consider the BLUP with respect to the enlarged model $\mathcal{F} = \mathcal{F}(\mathbb{R})$ where β_0 is an unknown real number and $\sigma_\epsilon^2 > 0$. For this $\mathcal{F}$, recall that every unbiased $\widehat{Y_0} = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ must satisfy $a_0 = 0$ and $\sum_{i=1}^n a_i = 1$. The MSPE of $\widehat{Y_0}$ is

$$E\left\{\left(\sum_{i=1}^n a_i Y_i - Y_0\right)^2\right\} = \left(\beta_0 \sum_{i=1}^n a_i - \beta_0\right)^2 + \sigma_\epsilon^2 \times \sum_{i=1}^n a_i^2 + \sigma_\epsilon^2$$

$$= 0 + \sigma_\epsilon^2 \times \left(1 + \sum_{i=1}^n a_i^2\right) \tag{3.2.18}$$

$$\geq \sigma_\epsilon^2(1 + 1/n), \tag{3.2.19}$$

where equality holds in (3.2.18) because $\sum_{i=1}^n a_i = 1$ and the minimum in (3.2.19) is calculated by observing that $\sum_{i=1}^n a_i^2$ is minimized subject to $\sum_{i=1}^n a_i = 1$ when $a_i = 1/n$ for $1 \leq i \leq n$. This tells us that the sample mean $\widehat{Y_0} = \frac{1}{n}\sum_{i=1}^n Y_i$ is the best linear unbiased predictor of Y_0 for the enlarged $\mathcal{F}$. The formula (3.2.19) for its MSPE is familiar from regression; σ_ϵ^2/n is the variance of the sample mean $\frac{1}{n}\sum_{i=1}^n Y_i$ while the "extra" σ_ϵ^2 accounts for the additional variability of Y_0. ■

Example 3.6. **(BLUP for a measurement error model)** Suppose that

$$Y_i \equiv Y(\boldsymbol{x}_i) = \sum_{j=1}^p f_j(\boldsymbol{x}_i)\beta_j + \epsilon_i = \boldsymbol{f}^\top(\boldsymbol{x}_i)\boldsymbol{\beta} + \epsilon_i$$

for $0 \leq i \leq n$, where the $\{f_j\}$ are known regression functions, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ is unknown, and the measurement errors $\{\epsilon_i\}$ are uncorrelated with common mean zero and common variance σ_ϵ^2. Consider the BLUP of $Y_0 = Y(\boldsymbol{x}_0)$ for the moment model $\mathcal{F}$, where $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\sigma_\epsilon^2 > 0$ but both are otherwise unknown. The predictor $\widehat{Y_0} = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ is unbiased with respect to $\mathcal{F}$ provided

$$E\left\{a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n\right\} = a_0 + \boldsymbol{a}^\top \boldsymbol{F}\boldsymbol{\beta} \stackrel{\text{set}}{=} E\{Y_0\} = \boldsymbol{f}_0^\top \boldsymbol{\beta}$$

for all $(\boldsymbol{\beta}, \sigma_\epsilon^2)$, where $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$. This is equivalent to

$$a_0 = 0 \quad \text{and} \quad \boldsymbol{F}^\top \boldsymbol{a} = \boldsymbol{f}_0. \tag{3.2.20}$$

In the Chapter Notes, Subsection 3.6, we show that the BLUP of Y_0 is

$$\widehat{Y_0} = \boldsymbol{f}_0^\top \widehat{\boldsymbol{\beta}}, \tag{3.2.21}$$

where $\widehat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top \boldsymbol{F})^{-1} \boldsymbol{F}^\top \boldsymbol{Y}^n$ is the ordinary least squares estimator of $\boldsymbol{\beta}$ and that the BLUP is unique. ■

In the next section, we turn specifically to the problem of prediction for computer experiments. We begin our discussion with the Gaussian stochastic process model introduced in Section 2.3 and then derive predictors of $Y(\boldsymbol{x}_0)$ when $\boldsymbol{\beta}$ is unknown, first when the correlation function is *known* and then when it is *unknown*.

3.3 Empirical Best Linear Unbiased Prediction

3.3.1 Introduction

Many types of analyses of computer experiment output are facilitated by having available easily-computed approximations to the computer code. Such approximations are often called *surrogates* in the global optimization literature (Booker et al (1999)) and *simulators* in the engineering literature (Bernardo et al (1992)). Neural networks, splines, and predictors based on Gaussian process models are some of the approximation methods that have been used to form predictors for the output from computer experiments. We emphasize the latter for three reasons: the assumptions that lead to these predictors are explicitly stated; several familiar predictors, including linear and cubic splines, are special cases; and such predictors use data-dependent scaling of each input dimension.

The basis for most practical predictors is the Gaussian random function model introduced in Section 2.3. Recall that this model regards the deterministic computer output $y(\cdot)$ as the realization of the random function

$$Y(\boldsymbol{x}) = \sum_{j=1}^{p} f_j(\boldsymbol{x})\beta_j + Z(\boldsymbol{x}) = \boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{\beta} + Z(\boldsymbol{x}), \tag{3.3.1}$$

where $f_1(\cdot), \ldots, f_p(\cdot)$ are known regression functions, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ is a vector of unknown regression coefficients, and $Z(\cdot)$ is a stationary Gaussian process on $\mathcal{X}$ having zero mean, variance σ_z^2, and correlation function $R(\cdot)$.

Suppose that the training data consists of the computer output at the input sites $\boldsymbol{x}_1$, $\ldots$, $\boldsymbol{x}_n$ and that $y(\boldsymbol{x}_0)$ is to be predicted. The model (3.3.1) implies that $Y_0 = Y(\boldsymbol{x}_0)$ and $\boldsymbol{Y}^n = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n))^\top$ has the multivariate normal distribution

$$\begin{pmatrix} Y_0 \\ \boldsymbol{Y}^n \end{pmatrix} \sim N_{1+n}\left[\begin{pmatrix} \boldsymbol{f}_0^\top \\ \boldsymbol{F} \end{pmatrix}\boldsymbol{\beta}, \sigma_z^2 \begin{pmatrix} 1 & \boldsymbol{r}_0^\top \\ \boldsymbol{r}_0 & \boldsymbol{R} \end{pmatrix}\right], \tag{3.3.2}$$

where $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$ is the $p \times 1$ vector of regression functions for $Y(\boldsymbol{x}_0)$, $\boldsymbol{F} = (f_j(\boldsymbol{x}_i))$ is the $n \times p$ matrix of regression functions for the training data, $\boldsymbol{r}_0 = (R(\boldsymbol{x}_0 - \boldsymbol{x}_1), \ldots, R(\boldsymbol{x}_0 - \boldsymbol{x}_n))^\top$ is the $n \times 1$ vector of correlations of $\boldsymbol{Y}^n$ with $Y(\boldsymbol{x}_0)$, and $\boldsymbol{R} = (R(\boldsymbol{x}_i - \boldsymbol{x}_j))$ is the $n \times n$ matrix of correlations among the $\boldsymbol{Y}^n$. The parameters $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\sigma_z^2 > 0$ are *unknown*.

The following discussion applies the development of Section 3.2 to find the BLUP of $Y(\boldsymbol{x}_0)$ under the following enlargement of the normal theory model (3.3.2).

Drop the Gaussian assumption in (3.3.2) to make the model a nonparametric, moment model based on an arbitrary second-order stationary process. It is assumed that $(Y_0, \boldsymbol{Y}^n)$ satisfy

$$\begin{pmatrix} Y_0 \\ \boldsymbol{Y}^n \end{pmatrix} \sim \left[\begin{pmatrix} \boldsymbol{f}_0^\top \\ \boldsymbol{F} \end{pmatrix} \boldsymbol{\beta},\ \sigma_z^2 \begin{pmatrix} 1 & \boldsymbol{r}_0^\top \\ \boldsymbol{r}_0 & \boldsymbol{R} \end{pmatrix} \right], \tag{3.3.3}$$

where $\boldsymbol{\beta}$ and $\sigma_z^2 > 0$ are unknown.

Begin by considering the conceptual case in which the correlation function $R(\cdot)$ is *known* (and hence $\boldsymbol{r}_0$ and $\boldsymbol{R}$ are also known). In this case Subsection 3.6 of the Chapter Notes shows that

$$\widehat{Y}(\boldsymbol{x}_0) = \widehat{Y}_0 \equiv \boldsymbol{f}_0^\top \widehat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}), \tag{3.3.4}$$

is the BLUP of $Y(\boldsymbol{x}_0)$ with respect to the family of distributions (3.3.3). In (3.3.4) $\widehat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{Y}^n$ is the generalized least squares estimator of $\boldsymbol{\beta}$. Of course, both $\boldsymbol{r}_0$ and $\boldsymbol{R}$ are determined when the correlation function $R(\cdot)$ is known. However both $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\sigma_z^2 > 0$ are unknown.

In Section 3.2 we proved that (3.3.4) is the best MSPE predictor among all predictors under a two-stage model whose first stage was the Gaussian model (3.3.2). Here we *increase* the size of the model class by not specifying the first two moments of $Y(\boldsymbol{x})$, i.e., $\boldsymbol{\beta}$ and σ_z^2 are unknown in (3.3.3), but *restrict* the class of predictors to *linear predictors* that are unbiased with respect to any model of the form (3.3.3).

We complete this discussion by stating three properties of the predictor (3.3.4) under (3.3.3). First, $\widehat{Y}_0$ interpolates the training data $(\boldsymbol{x}_i, Y(\boldsymbol{x}_i))$ for $1 \le i \le n$ (Section 3.2). Second, (3.3.4) is a LUP of $Y(\boldsymbol{x}_0)$; as shown below, this fact permits the straightforward derivation of its variance optimality. Linearity follows by substituting $\widehat{\boldsymbol{\beta}}$ into $\widehat{Y}_0$ yielding

$$\begin{aligned} \widehat{Y}_0 &= \Big[\boldsymbol{f}_0^\top (\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \\ &\quad + \ \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} (\boldsymbol{I}_n - \boldsymbol{F}(\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1}) \Big] \boldsymbol{Y}^n \\ &\equiv \boldsymbol{a}_*^\top \boldsymbol{Y}^n, \end{aligned} \tag{3.3.5}$$

where (3.3.5) defines $\boldsymbol{a}_*$. Unbiasedness with respect to (3.3.3) holds because for any $\boldsymbol{\beta} \in \mathbb{R}^p$ and every $\sigma_z^2 > 0$,

$$\begin{aligned} E\{\widehat{Y}_0\} &= \boldsymbol{a}_*^\top E\{\boldsymbol{Y}^n\} \\ &= \boldsymbol{a}_*^\top \boldsymbol{F} \boldsymbol{\beta} \\ &= [\boldsymbol{f}_0^\top \boldsymbol{I}_n + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} (\boldsymbol{F} - \boldsymbol{F}\boldsymbol{I}_n)] \boldsymbol{\beta} \\ &= \boldsymbol{f}_0^\top \boldsymbol{\beta} \\ &= E\{Y(\boldsymbol{x}_0)\}, \end{aligned} \tag{3.3.6}$$

where (3.3.6) holds by substituting for $\boldsymbol{a}_*$ in (3.3.5).

Third, we determine the behavior of $\widehat{Y}_0 = \widehat{Y}(\boldsymbol{x}_0)$ as a function of $\boldsymbol{x}_0$. This can be easily discerned because (3.3.4) depends on $\boldsymbol{x}_0$ only through the $n \times 1$ vector $\boldsymbol{r}_0 = \boldsymbol{r}(\boldsymbol{x}_0) = (R(\boldsymbol{x}_0 - \boldsymbol{x}_1), \ldots, R(\boldsymbol{x}_0 - \boldsymbol{x}_n))^\top$ and $\boldsymbol{f}(\boldsymbol{x}_0)$. Hence

$$\widehat{Y}(\boldsymbol{x}_0) = \sum_{j=1}^{p} \widehat{\beta}_j f_j(\boldsymbol{x}_0) + \sum_{i=1}^{n} d_i R(\boldsymbol{x}_0 - \boldsymbol{x}_i), \tag{3.3.7}$$

where $\boldsymbol{d} = (d_1, \ldots, d_n)^\top = \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}})$. In the special case $Y(\boldsymbol{x}) = \beta_0 + Z(\boldsymbol{x})$, $\widehat{Y}(\boldsymbol{x}_0)$ depends on $\boldsymbol{x}_0$ only through $R(\boldsymbol{x}_0 - \boldsymbol{x}_i)$. The "smoothness" characteristics of $\widehat{Y}(\boldsymbol{x}_0)$ are inherited from those of $R(\cdot)$. For $\boldsymbol{x}_0$ "near" any $\boldsymbol{x}_i$ (more precisely, in the limit as $\boldsymbol{x}_0$ approaches $\boldsymbol{x}_i$), the behavior of $\widehat{Y}(\boldsymbol{x}_0)$ depends on that of $R(\cdot)$ at the origin.

Example 3.7. As in Example 3.3, suppose that

$$f(x) = e^{-1.4x} \cos(7\pi x/2),$$

a dampened cosine curve over $0 \le x \le 1$, is the true output function. Figure 3.3 shows $f(x)$ as a solid line and the set of $n = 7$ points that we previously introduced as training data. Example 3.3 emphasized the role of the residuals in interpreting the BLUP (3.3.4) of $Y(x_0)$. We complete this example by re-examining the BLUP, this time as a function of x_0. Because the known correlation function for this example is

$$R(h) = e^{-136.1 \times h^2},$$

we have

$$\widehat{Y}(x_0) = \sum_{i=1}^{7} d_i \exp\{-136.1(x_i - x_0)^2\}, \tag{3.3.8}$$

where $\{x_i\}_{i=1}^7$ are inputs for the training data and $(d_1, \ldots, d_7)$ are calculated from the expression following (3.3.7). Figure 3.3 shows that $\widehat{Y}(x_0)$ does interpolate the training data. Because each exponential component of (3.3.8) is infinitely differentiable in x_0, $\widehat{Y}(x_0)$ is also infinitely differentiable in x_0. ■

3.3.2 Prediction When the Correlation Function is Unknown

The basic strategy is to predict $y(\boldsymbol{x}_0)$ by

$$\widehat{Y}(\boldsymbol{x}_0) = \widehat{Y}_0 \equiv \boldsymbol{f}_0^\top \widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{r}}_0^\top \widehat{\boldsymbol{R}}^{-1} \left(\boldsymbol{Y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right), \tag{3.3.9}$$

where $\widehat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top \widehat{\boldsymbol{R}}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^\top \widehat{\boldsymbol{R}}^{-1} \boldsymbol{Y}^n$ and the estimates $\widehat{\boldsymbol{R}}$ and $\widehat{\boldsymbol{r}}_0$ are determined from an *estimator* of the correlation function $R(\cdot)$. Such predictors are called *empirical best linear unbiased predictors* (EBLUPs) of $Y(\boldsymbol{x}_0)$, despite the fact that they are

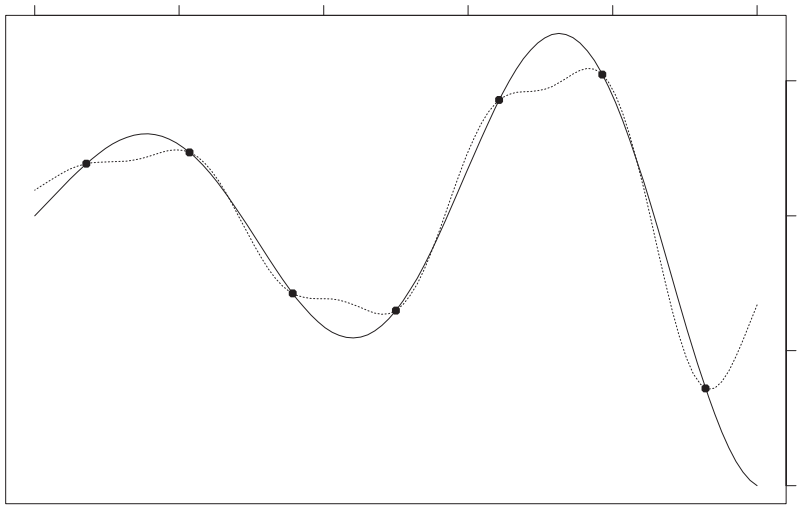


Fig. 3.3 True curve $y(x) = \exp\{-1.4x\} \times \cos(3.5\pi x)$ (solid line), a seven point input design (dots), and the BLUP $\widehat{Y}(x_0)$ for $x_0 \in [0, 1.0]$ (dotted line)

typically no longer linear in the training data $\boldsymbol{Y}^n$ ($\widehat{\boldsymbol{R}} = \widehat{\boldsymbol{R}}(\boldsymbol{Y}^n)$ and $\widehat{\boldsymbol{r}}_0 = \widehat{\boldsymbol{r}}_0(\boldsymbol{Y}^n)$ are usually highly nonlinear in $\boldsymbol{Y}^n$) nor need they be unbiased for $Y(\boldsymbol{x}_0)$ (although see Kackar and Harville (1984)). Different EBLUPs correspond to different estimators of $R(\cdot)$.

Virtually all estimators of the correlation function that have appeared in the literature assume that $R(\cdot) = R(\cdot|\boldsymbol{\psi})$, where $\boldsymbol{\psi}$ is a finite vector of parameters. As an example, the exponential correlation function

$$R(\boldsymbol{h}|\boldsymbol{\psi}) = \exp\left\{-\sum_{j=1}^{d} |h_j/\theta_j|^{p_j}\right\}$$

has d scale parameters $\theta_1, \ldots, \theta_d$ and d power parameters $p_1, \ldots, p_d$ so that $\boldsymbol{\psi} = (\theta_1, \ldots, \theta_d, p_1, \ldots, p_d)$ contains $2 \times d$ components. In this case, the correlation matrix $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{\psi})$ depends on $\boldsymbol{\psi}$ as does the vector of correlations $\boldsymbol{r}_0 = \boldsymbol{r}_0(\boldsymbol{\psi})$. We describe four methods of estimating $\boldsymbol{\psi}$ that lead to four different EBLUPs. All except the "cross-validation" estimator of $\boldsymbol{\psi}$ assume that the training data have the Gaussian conditional distribution

$$[\boldsymbol{Y}^n|\boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\psi}] \sim N_n\left[\boldsymbol{F}\boldsymbol{\beta}, \sigma_z^2\boldsymbol{R}\right]. \tag{3.3.10}$$

Furthermore, the Bayes predictor assumes that prior information is available concerning model parameters.

We focus on the estimation of the correlation parameters $\boldsymbol{\psi}$ and not the process variance, σ_z^2. This is because the predictor $\widehat{Y}_0$ depends only on $\boldsymbol{\psi}$ and is independent of σ_z^2. However, in Section **??** (for example, formula (4.1.12)), we will see that σ_z^2 is required to estimate the posterior variance of the predictor at each new training site $\boldsymbol{x}_0$. Except for cross validation, all the methods we present below can be used to estimate σ_z^2.

3.3.2.1 Maximum Likelihood EBLUPs

Perhaps the most popular choice of $\boldsymbol{\psi}$ estimator is the maximum likelihood estimate (MLE). Using the multivariate normal assumption, the log likelihood is (up to an additive constant)

$$\ell(\boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\psi}) = -\frac{1}{2}\left[n \log \sigma_z^2 + \log(\det(\boldsymbol{R})) + (\boldsymbol{y}^n - \boldsymbol{F}\boldsymbol{\beta})^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^n - \boldsymbol{F}\boldsymbol{\beta})/\sigma_z^2\right], \tag{3.3.11}$$

where $\det(\boldsymbol{R})$ denotes the determinant of $\boldsymbol{R}$. Given $\boldsymbol{\psi}$, the MLE of $\boldsymbol{\beta}$ is its generalized least squares estimate

$$\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(\boldsymbol{\psi}) = \left(\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{y}^n \tag{3.3.12}$$

and the MLE of σ_z^2 is

$$\widehat{\sigma_z^2} = \widehat{\sigma_z^2}(\boldsymbol{\psi}) = \frac{1}{n}\left(\boldsymbol{y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)^\top \boldsymbol{R}^{-1}\left(\boldsymbol{y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right). \tag{3.3.13}$$

Substituting these values into Equation (3.3.11), we obtain that the maximum of (3.3.11) over $\boldsymbol{\beta}$ and σ_z^2 is

$$\ell(\widehat{\boldsymbol{\beta}}, \widehat{\sigma_z^2}, \boldsymbol{\psi}) = -\frac{1}{2}\left[n \log \widehat{\sigma_z^2}(\boldsymbol{\psi}) + \log(\det(\boldsymbol{R}(\boldsymbol{\psi}))) + n\right],$$

which depends on $\boldsymbol{\psi}$ alone. Thus the MLE chooses $\widehat{\boldsymbol{\psi}}$ to minimize

$$n \log \widehat{\sigma_z^2}(\boldsymbol{\psi}) + \log\left(\det\left(\boldsymbol{R}(\boldsymbol{\psi})\right)\right), \tag{3.3.14}$$

where $\widehat{\sigma_z^2}$ is defined by (3.3.13). The predictor corresponding to $\widehat{\boldsymbol{\psi}}$ is called an MLE-EBLUP of $Y(\boldsymbol{x}_0)$.

For the Gaussian stochastic process model, the SAS procedure PROC Mixed and program `GaSP` (Gaussian Stochastic Process, Welch et al (1992)) can calculate MLEs of the parameters for the product power exponential correlation function. The program `PErK` (Parametric EmpiRical Kriging, Williams (2001)) can calculate the MLEs of the parameters for both the product power exponential and product Matérn correlation functions.

3.3.2.2 Restricted Maximum Likelihood EBLUPs

Again assume that $R(\cdot)$ (and hence $\boldsymbol{R}$ and $\boldsymbol{r}_0$) depends on an unknown finite vector of parameters $\boldsymbol{\psi}$. Restricted (residual) maximum likelihood estimation (REML) of variance and covariance parameters was introduced by Patterson and Thompson (1971) as a method of determining less biased estimates of such parameters than maximum likelihood estimation (see also Patterson and Thompson (1974)). Some authors use the term *marginal maximum likelihood estimates* for the same concept.

The REML estimator of $\boldsymbol{\psi}$ maximizes the likelihood of a maximal set of linearly independent combinations of the $\boldsymbol{Y}^n$ where each linear combination is orthogonal to $\boldsymbol{F\beta}$, the mean vector of $\boldsymbol{Y}^n$. Assuming that $\boldsymbol{F}$ is of full column rank p, this method corresponds to choosing an $(n-p)\times n$ matrix $\boldsymbol{C}$ of full row rank $n-p$ that satisfies $\boldsymbol{CF} = \boldsymbol{0}$, and the REML estimator of $\boldsymbol{\psi}$ is the maximizer of the likelihood of the transformed "data"

$$\boldsymbol{W} \equiv \boldsymbol{CY}^n \sim N\left[\boldsymbol{CF\beta} = \boldsymbol{0}, \sigma_Z^2 \boldsymbol{CR}(\boldsymbol{\psi})\boldsymbol{C}^\top\right]. \tag{3.3.15}$$

Notice that $\boldsymbol{W}$ contains p fewer "observations" than $\boldsymbol{Y}^n$ but $\boldsymbol{W}$ has the advantage that these data contain none of the unknown parameters $\boldsymbol{\beta}$.

As an example, consider the simplest linear model setting, that of independent and identically distributed $N(\beta_0, \sigma^2)$ observations $Y_1, \ldots, Y_n$. In this case, $p = 1$. The MLE of σ^2 based on the $Y_1, \ldots, Y_n$ is $\sum_{i=1}^n (Y_i - \overline{Y})^2/n$, which is a (downward) biased estimator of σ^2. One set of linear combinations having the orthogonality property $\boldsymbol{CF} = \boldsymbol{0}$ is obtained as follows. Let $\overline{Y}$ be the mean of $Y_1, \ldots, Y_n$. The linear combinations $W_1 = Y_1 - \overline{Y}, \ldots, W_{n-1} = Y_{n-1} - \overline{Y}$ each have mean zero and correspond to multiplying $\boldsymbol{Y}^n$ by an easily described $(n-1)\times n$ matrix $\boldsymbol{C}$ having full row rank $n-1$. Maximizing the likelihood based on $W_1, \ldots, W_{n-1}$ and expressing the result in terms of $Y_1, \ldots, Y_n$ gives

$$\sum_{i=1}^n (Y_i - \overline{Y})^2/(n-1)\,.$$

The $n-1$ divisor in the error sum of squares produces an unbiased estimator of σ_Z^2.

Returning to the general case, it can be shown that the REML estimator of $\boldsymbol{\psi}$ is independent of the choice of linear combinations used to construct $\boldsymbol{W}^n$ subject to the number of columns of $\boldsymbol{C}$ being maximal in the sense of $\boldsymbol{C}$ having rank $n-p$ (Harville (1974), Harville (1977)). With some algebra it can be shown that the REML estimator of σ_Z^2 is

$$\widetilde{\sigma_Z^2} = \frac{n}{n-p}\widehat{\sigma_Z^2} = \frac{1}{n-p}\left(\boldsymbol{y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)^\top \boldsymbol{R}^{-1}\left(\boldsymbol{y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right),$$

where $\widehat{\sigma_Z^2}$ is the MLE, formula (3.3.13), of σ_Z^2 and the REML estimator of $\boldsymbol{\psi}$ is the minimizer of

$$(n-p)\log\widetilde{\sigma_Z^2} + \log(\det(\boldsymbol{R}(\boldsymbol{\psi}))). \tag{3.3.16}$$

3.3.2.3 Cross-Validation EBLUPs

Cross-validation is a popular method for choosing model parameters in parametric model settings. Important early references describing cross-validation are Allen (1974), Stone (1974), and Stone (1977); Hastie et al (2001) summarize recent applications.

We again assume that the correlation function is parametric with $R(\cdot) = R(\cdot|\boldsymbol{\psi})$ so that $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{\psi})$ and $\boldsymbol{r}_0 = \boldsymbol{r}_0(\boldsymbol{\psi})$. For $i = 1, \ldots, n$ let $\widehat{Y}_{-i}(\boldsymbol{\psi})$ denote the predictor (3.3.9) of $y(\boldsymbol{x}_i)$ when $\boldsymbol{\psi}$ is the true correlation parameter based on all the data *except* $(\boldsymbol{x}_i, y(\boldsymbol{x}_i))$. The cross-validated estimator of $\boldsymbol{\psi}$ minimizes the empirical mean squared prediction error

$$\text{XV-PE}(\boldsymbol{\psi}) = \sum_{i=1}^{n} (\widehat{Y}_{-i}(\boldsymbol{\psi}) - y(\boldsymbol{x}_i))^2. \tag{3.3.17}$$

More general forms of the cross-validation criterion have been proposed by Golub et al (1979) and Wahba (1980).

3.3.2.4 Posterior Mode EBLUPs

The motivation and form of the posterior mode EBLUP is as follows. Recall that the minimum MSPE predictor of $Y(\boldsymbol{x}_0)$ is $E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n\}$ (Theorem 3.1). As described in Subsection 2.4.6, in fully Bayesian settings where a prior is available for $(\boldsymbol{\beta}, \sigma_Z^2, \boldsymbol{\psi})$, this conditional mean can be very difficult to compute. To explain why, suppose conditionally given $(\boldsymbol{\beta}, \sigma_Z^2, \boldsymbol{\psi})$ that $\boldsymbol{Y}^n$ is from a GRF and that a prior is available for $[\boldsymbol{\beta}, \sigma_Z^2, \boldsymbol{\psi}]$. The minimum MSPE predictor is by

$$E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n\} = E\{E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n, \boldsymbol{\psi}\}|\boldsymbol{Y}^n\}, \tag{3.3.18}$$

where the inner expectation on the right-hand side of (3.3.18) is regarded as a function of $\boldsymbol{\psi}$ and the outer expectation is with respect to the (marginal) posterior distribution $[\boldsymbol{\psi}|\boldsymbol{Y}^n]$.

The inner expectation (3.3.18) can be calculated by

$$E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n, \boldsymbol{\psi}\} = E\left\{(Y(\boldsymbol{x}_0), \boldsymbol{\beta}, \sigma_Z^2)|\boldsymbol{Y}^n, \boldsymbol{\psi}\right\},$$

which assumes that the conditional $[(\boldsymbol{\beta}, \sigma_Z^2)|\boldsymbol{Y}^n, \boldsymbol{\psi}]$ is available and the integration over $(\boldsymbol{\beta}, \sigma_Z^2)$ has been performed. Subsection 4.1.3 gives several examples of closed-form expressions for $E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n, \boldsymbol{\psi}\}$. Even where it can be evaluated in closed form, this integrand is a very complicated function of $\boldsymbol{\psi}$. For example, $E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n, \boldsymbol{\psi}\}$ involves the determinant of the correlation matrix as one of several terms in the examples of Subsection 4.1.3.

Even if $E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n, \boldsymbol{\psi}\}$ is known, the density of $[\boldsymbol{\psi}|\boldsymbol{Y}^n]$ generally cannot be expressed in closed form. One simple-minded but nevertheless attractive approximation to the right-hand side of (3.3.18) is

$$E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n, \widehat{\boldsymbol{\psi}}\}, \tag{3.3.19}$$

where $\widehat{\boldsymbol{\psi}}$ is the posterior mode of $[\boldsymbol{\psi}|\boldsymbol{Y}^n]$. The posterior mode of $\boldsymbol{\psi}$ is that $\widehat{\boldsymbol{\psi}}$ that maximizes

$$[\boldsymbol{\psi}|\boldsymbol{Y}^n] = \frac{[\boldsymbol{Y}^n|\boldsymbol{\psi}][\boldsymbol{\psi}]}{[\boldsymbol{Y}^n]} \propto [\boldsymbol{Y}^n|\boldsymbol{\psi}][\boldsymbol{\psi}].$$

This approximation is based on the (greatly) simplifying assumption that $[\boldsymbol{\psi}|\boldsymbol{Y}^n]$ is degenerate with mass located at its mode (Gibbs (1997)). Equation (3.3.19) is the definition of the *posterior mode* EBLUP.

While the predictor (3.3.19) uses the correlation parameter that seems "most likely" as judged by the posterior, the choice of prior for the correlation function parameters is nontrivial. Subsection 2.4.6 discusses hierarchical models for the $[\boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\psi}]$ prior.

The harried but Bayesian inclined user may wish to compute the posterior mode $\widehat{\boldsymbol{\psi}}$ based on a non-informative prior for $[\boldsymbol{\psi}]$, as alluded to in Subsection 2.4.6. Neal (2003) describes a Monte Carlo algorithm called "slice sampling" that can sample from this posterior distribution in certain cases (see Robert and Casella (1999) for a general introduction to Markov Chain Monte Carlo algorithms). The reader should be warned that correlation parameters are one instance where improper priors can lead to improper posteriors. Berger et al (2001) proves that, for isotropic correlation functions, many of the improper priors suggested in the literature yield improper posteriors. It also proposes an improper prior for default use whose posterior is proper.

The program `PErK` can compute the posterior mode EBLUP when the prior satisfies

$$[\boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\psi}] = [\boldsymbol{\beta}, \sigma_z^2][\boldsymbol{\psi}],$$

where $[\boldsymbol{\beta}, \sigma_z^2]$ is the proper prior (1) in Table 4.1 and the prior options for $[\boldsymbol{\psi}]$ are specified in Appendix **??**.

3.4 A Simulation Comparison of EBLUPs

Which EBLUP should be used? We summarize the results of a simulation study that compares the small-sample predictive accuracy of seven predictors. This material is drawn, in part, from the thesis of Lehman (2002).

In brief, this study was conducted as follows. The seven predictors were cubic regression, with coefficients estimated by ordinary least squares, and six EBLUPs; the six EBLUPS were formed by estimating the parameters of two different parametric correlation functions using three different methods. Each prediction method was applied to 200 randomly drawn surfaces on $[0, 1]^2$. The 200 surfaces consist of 50 surfaces generated using each of four stochastic mechanisms. The first group of surfaces was meant to be "near-cubic" while the last three groups of surfaces were generated using the *krigifier* of Trosset (1999) and Trosset and Padula (2000). In

addition to studying the predictor and the type of surface being predicted, this study examines the effect of the choice of training sites and the size of the training data set that is used to form the predictor. Three designs were used: a maximin distance Latin hypercube design, a Sobol´ sequence, and a D-optimal design. Two sample sizes were selected for each design criterion: $n = 11$ (a "small" sample size roughly corresponding to 5 observations per input dimension) and $n = 16$ (a "large" sample size). Larger n produced extremely small empirical mean squared prediction errors for all designs and did not adequately differentiate them.

The response in this study was the *empirical root mean squared prediction error* (ERMSPE). For each predictor, each true surface, and each set of training data (determined by the design and sample size), the estimates of the correlation parameters and the ERMSPE

$$\text{ERMSPE} = \sqrt{\frac{1}{625}\sum_{i=1}^{625}\left(y(\boldsymbol{x}_i) - \widehat{y}(\boldsymbol{x}_i)\right)^2}$$

were computed on a grid of $25^2 = 625$ equispaced points on $[0, 1]^2$. Here $y(\boldsymbol{x})$ is the true function value and $\widehat{y}(\boldsymbol{x})$ is the predicted value. Our *primary criterion* for judging each prediction method was its *prediction accuracy*, as measured by the ERMSPE over the 200 surfaces. Concerning correlation parameter estimates, we merely note that the poorer prediction methods consistently had more biased and highly variable estimates.

We expect, for example, the cubic regression predictor to do well when the training data are selected according to a D-optimal design and the true surface is near cubic. This study does have limitations that we will note below. However, it does provide some guidance about the relative predictive accuracy of these popular prediction methods. Before discussing the results, we describe each of the factors affecting the ERMSPE in more detail.

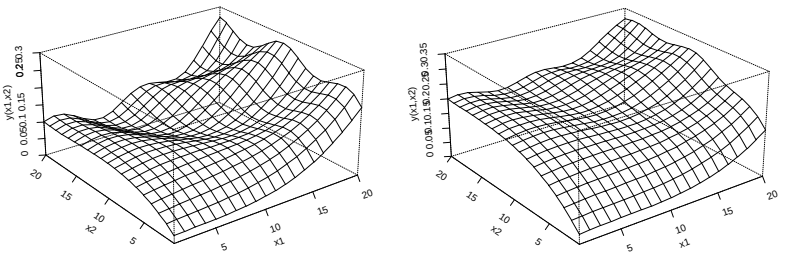


Fig. 3.4 Two of the 50 random $y(x_1, x_2)$ surfaces generated by the near-cubic process (3.4.1)

The 50 near-cubic surfaces were chosen to be of the form

$$y(x_1, x_2) = x_1^3/3 - (R_1 + S_1)\, x_1^2/2 + (R_1 S_1)\, x_1 + x_2^3/3 - (R_2 + S_2)\, x_2^2/2 + (R_2 S_2) x_2 + A \sin\left(\frac{2\pi x_1 x_2}{S}\right), \quad (3.4.1)$$

where the model coefficients (R_1, S_1), (R_2, S_2), and (A, S) were selected randomly. The eight coefficients were taken to be mutually independent; R_1 and S_1 were distributed uniformly over the interval $(0, 0.5)$ (denoted $U(0, 0.5)$), R_2 and S_2 were $U(.5, 1.0)$, A was $U(0, .05)$, and S was $U(.04, 1.0)$. The small amplitude coefficient of the sin(·) term, A, assured that there were only minor deviations from the cubic model. Two functions $y(\cdot)$ drawn using this stochastic mechanism are displayed in Figure 3.4.

The last three groups of surfaces were generated using the *krigifier* of Trosset (1999) and Trosset and Padula (2000), which was proposed as a device for generating random "true" functions. In brief, each surface was the BLUP-type interpolator

$$y(x_1, x_2) = \widehat{\beta} + \boldsymbol{r}(x_1, x_2)^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^{144} - \mathbf{1}_{144}\widehat{\beta}) \quad (3.4.2)$$

for $0 < x_1, x_2 < 1$, where $\boldsymbol{Y}^{144}$ was a 144×1 vector that was drawn from a Gaussian stochastic process at a 12×12 equispaced grid of points on $[0, 1]^2$. The components of $y(\cdot)$ were $\widehat{\beta} = (\mathbf{1}_{144}^\top \boldsymbol{R}^{-1} \mathbf{1}_{144})^{-1}\, \mathbf{1}_{144}^\top\, \boldsymbol{R}^{-1}\, \boldsymbol{Y}^{144}$, $\boldsymbol{r}(x_1, x_2) = \boldsymbol{r}(\boldsymbol{x})$ is the 144×1 vector with i^{th} component $R(\boldsymbol{x}_i - \boldsymbol{x})$, and $\boldsymbol{R}$ is the 144×144 matrix with $(i, j)^{th}$ element $R(\boldsymbol{x}_i - \boldsymbol{x}_j)$. The correlation function $R(\cdot)$ was specified to be of Matérn form; each set of 144 points was drawn from a Gaussian stochastic process that had mean 100, variance 1, and the Matérn correlation function

$$R(h_1, h_2) = \prod_{i=1}^{2} \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i}\right)^{\nu} K_\nu\left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i}\right) \quad (3.4.3)$$

with $\theta_1 = 1/\sqrt{8}$ and $\theta_2 = 1/\sqrt{15}$, where $K_\nu(\cdot)$ is the modified Bessel function of order ν (see Section 2.3). Recall that the "smoothness" of a given draw $y(\boldsymbol{x})$ is determined by the smoothness of the correlation function of the Gaussian stochastic process. Fifty of the draws came from the process with $\nu = 5$, 50 came from the process with $\nu = 10$, and 50 came from the process with $\nu = 50$, giving a total of 150 surfaces. The $\nu = 50$ correlation function effectively corresponds to the product exponential correlation function

$$R(h_1, h_2) = e^{-\theta_1 \times |h_1|^2} \times e^{-\theta_2 \times |h_2|^2} \quad (3.4.4)$$

with $\theta_1 = 8$ and $\theta_2 = 15$. The smoothness of the surfaces drawn increased with ν. Figure 3.5 displays two of the true test surfaces drawn using (3.4.2).

The output for each surface was evaluated for each of the 6 ($= 3 \times 2$) design × sample size combinations. One design we used selected the $\{\boldsymbol{x}_i\}$ to be a maximin

distance LHD on $[0,1]^2$ (see page **??**). The second design took the $\{\boldsymbol{x}_i\}$ to be D-optimal (Section 5.1) with respect to the cubic model

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \beta_6 x_1^2 x_2 + \beta_7 x_1 x_2^2 + \beta_8 x_1^3 + \beta_9 x_2^3. \quad (3.4.5)$$

With 10 coefficients, we selected $n = 11$ so that the D-optimal design would have 1 degree of freedom to estimate error. The third design chose the $\{\boldsymbol{x}_i\}$ to be a Sobol´ sequence of length 11 or 16 (Subsection **??**).

The first predictor considered in this study was the ordinary least squares regression predictor based on the cubic model (3.4.5). The remaining six predictors were EBLUPs of the form (3.3.9) that were constructed using the MLE, REML, and XV covariance parameter estimation methods with either the power exponential or Matérn parametric correlation families.

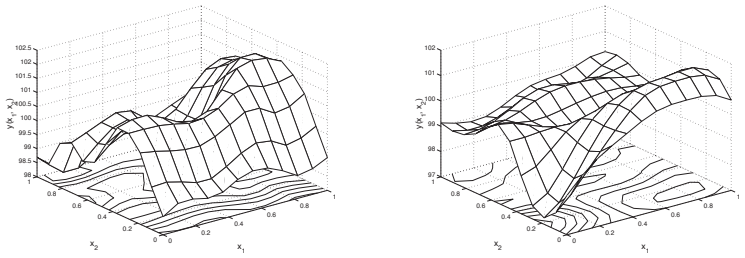


Fig. 3.5 Two of the 150 random $y(x_1, x_2)$ surfaces drawn from the krigifier defined by the correlation function (3.4.3); the left panel draw used $\nu = 5$ and right panel draw used $\nu = 50$

While formal statistical mechanisms can be used to analyze these data, the primary conclusions can be seen from the following figures. Figure 3.6 is a comparative plot of the distributions of the ERMSPEs for the seven predictors conditioned on n but ignoring the training data design and the stochastic mechanism used to generate the true surface. Thus each of the box and whisker plots is constructed from 600 points (3 designs $\times$ 4 stochastic mechanisms $\times$ 50 surfaces/stochastic mechanism). It is clear that cross-validation EBLUPs are inferior to those that use MLE or REML to estimate the covariance parameters. The same is true of cubic regression. Also the

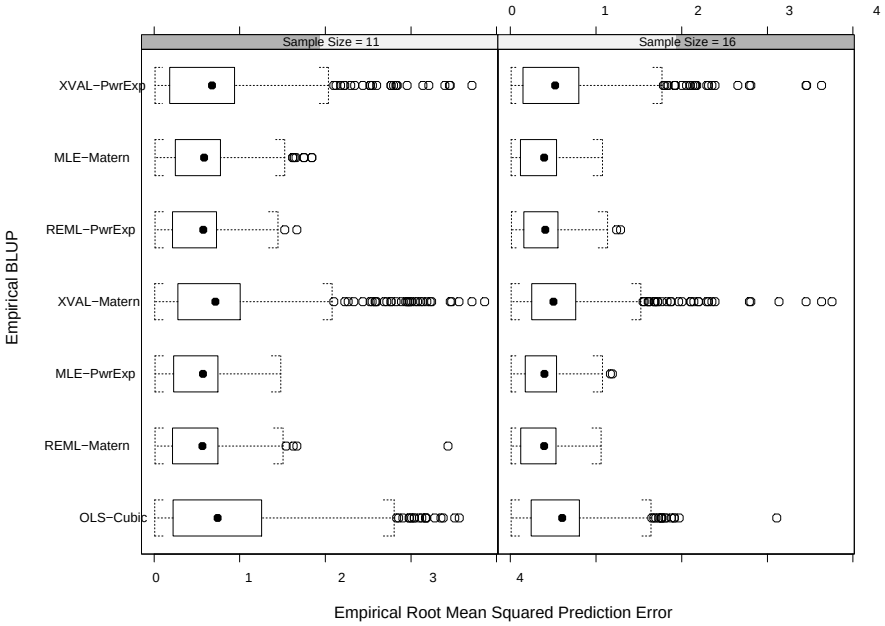


Fig. 3.6 Distribution of ERMSPE over 625 equispaced grid of points in $[0, 1]^2$ for the seven predictors conditioned on sample size (ignoring training data design and stochastic mechanism generating the true surface). Two large ERMSPE values by cross-validation EBLUPs were omitted from the plot.

$n = 11$ ERMSPEs tend to be larger than those for $n = 16$. In a figure not shown, when prediction is restricted to the near-cubic surfaces, the cubic regression predictor performs comparably to the kriging predictors. However, when the true $y(\cdot)$ is not near-cubic, the cubic regression predictor performs substantially worse than the kriging predictors.

Restricting attention to the four MLE- and REML- EBLUPs, Figures 3.7 and 3.8 separately examine graphically the distribution of the ERMSPE for each n, each type of design, and each class of true surface. The conclusions are similar in both figures. First, for near-cubic surfaces, any choice of training design and any choice of EBLUP is comparable and estimates the surface well. Second, use of the D-optimal design gives inferior prediction compared with the maximin distance LHD or the Sobol´ design. Third, outliers with slightly poorer performance occur more often with the Sobol´ design than with the maximin distance LHD.

Surface-by-surface comparisons of the ERMSPE of these predictors were made for the MLE-EBLUP versus the REML-EBLUP predictors and for the four differ-

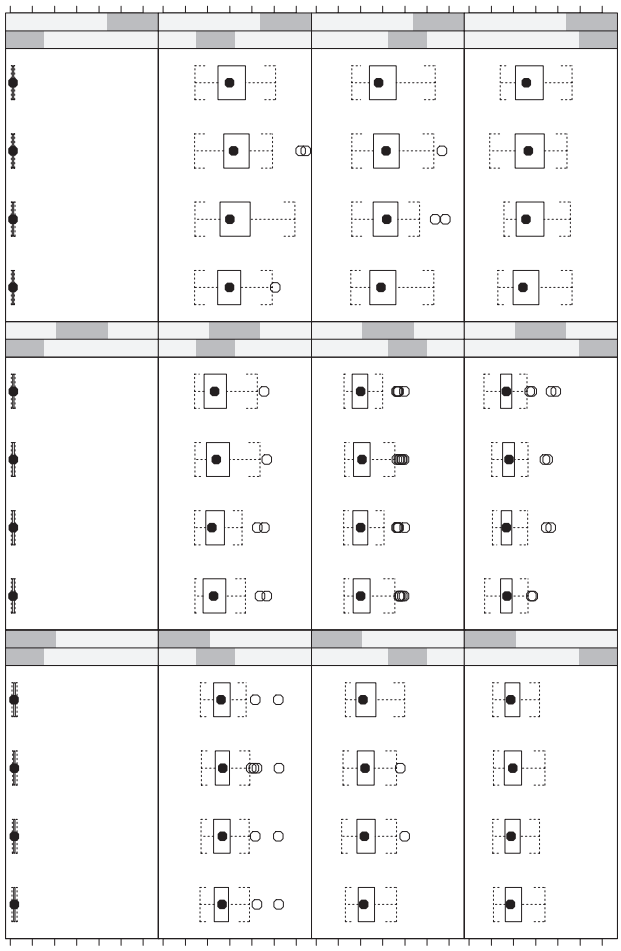


Fig. 3.7 Distribution of ERMSPE over 625 equispaced grid of points in $[0, 1]^2$ when $n = 16$ conditioned on training data design and type of true surface for the the MLE- and REML- EBLUPs based on either the power exponential and the Matérn correlation functions.

ent true surface groups. These plots showed little difference between the predictive ability of the two methods.

Before stating our primary conclusions, we wish to note several limitations of this particular empirical study that may effect our recommendations in some settings.

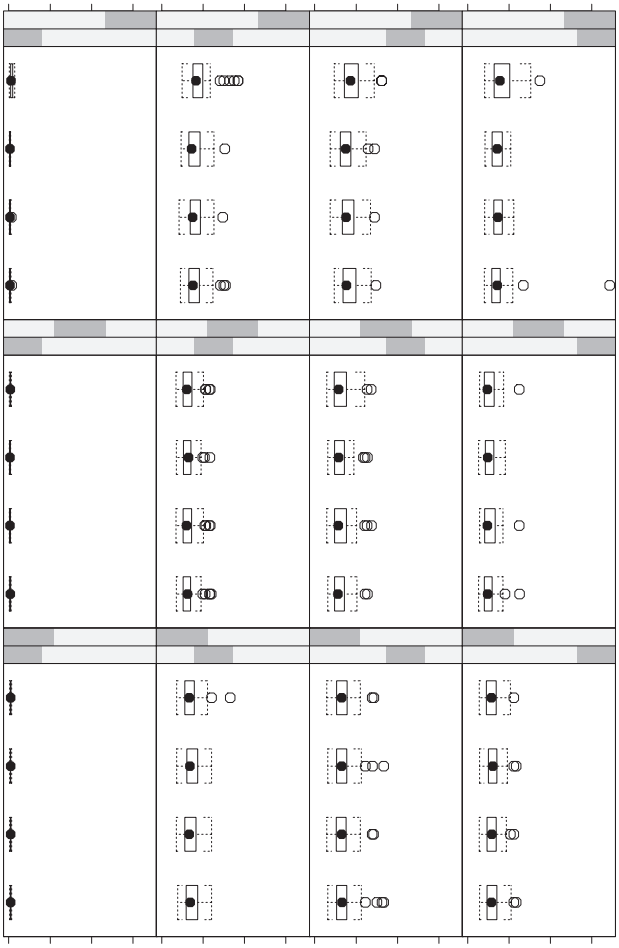


Fig. 3.8 Distribution of ERMSPE over 625 equispaced grid of points in $[0, 1]^2$ when $n = 11$ conditioned on training data design and type of true surface for the MLE- and REML- EBLUPs based on either the power exponential and the Matérn correlation functions.

First, all of the true surfaces are rather smooth, with the roughest corresponding to the krigifier with $\nu = 5$. Second, the dimension of the input in this study is $d = 2$, a rather low value. Third, none of the krigifier surfaces has a nonstationary trend term.

It would be desirable to enhance the ranges of all three of these factors to broaden the applicability of our recommendation.

There are several additional caveats that should be kept in mind regarding our recommendations. This section makes recommendations based on the prediction accuracy of several predictors. Among the other important products of the prediction process are prediction bounds based on the plug-in estimates of prediction variability that will be introduced in Section **??**. The accuracy of such prediction intervals will be addressed in more detail in Section **??**. Our assessment of the empirical coverage of the corresponding intervals does not change the basic recommendations given below. Another issue is that we did not explicitly assess how small an initial sample size can be used to provide "reasonably" accurate prediction surfaces; for sequential designs (in addition to arising in high-dimensional, high-cost codes), such problems occur in Subsection **??** where the sequential design of a computer experiment is presented to find the global optimum of $y(\cdot)$. Certainly five observations per dimension appears to be adequate based on this limited study.

Recommendation *We recommend use of either the REML-EBLUP or MLE-EBLUP based on the power exponential correlation family. The Matérn correlation family produces similar ERMSPEs as the power exponential correlation family but is more computationally expensive. Maximin distance LHDs produce good predictors with Sobol´ designs a close second. D-optimal designs should be not be used to generate training data.*

Example 3.8. We illustrate the use of PErK to fit the REML empirical BLUP that is described earlier in this section. Recall the data introduced in Section 1.2 on the time for a fire to reach five feet above a fire source located in a room of a given *room height* and *room area*. In addition to room geometry, this output time is also dependent on the inputs: *heat loss fraction*, a measure of how well the room retains heat, and the *height of the fire source* above the room floor. Figure 3.9 plots each of the six two-dimensional projections of the 40 input points generated by a Sobol´ design. As noted in Chapter 4, Sobol´ designs provide points that have a greater range of inter-point distances than do the maximin distance Latin hypercube designs (see Example **??**). This may allow better estimation of correlation parameters if the predictions are required at a set of points of varying distances from the training data.

Figure 3.10 displays scatterplots of each of the four input variables versus the time for a fire to reach five feet above the fire source, this output denoted by $y(\boldsymbol{x})$. Of these inputs, only *room area* appears to have a strong relationship with response time.

We desire to predict $y(\cdot)$ on a regular 320 point grid consisting of $4 \times 4 \times 4 \times 5$ equally spaced points over the ranges of the variables: heat loss fraction, room height, fire height, and room area, respectively. Our predictor is an EBLUP based on the Gaussian Stochastic Process with Matérn correlation function

$$R(\boldsymbol{h}) = \prod_{i=1}^{4} \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i} \right)^{\nu} K_{\nu}\left(\frac{2\sqrt{\nu}\,|h_i|}{\theta_i} \right) \tag{3.4.6}$$

with unknown correlation parameter $\boldsymbol{\psi} = (\theta_1, \ldots, \theta_4, \nu)$. Recall that for large ν, the i^{th} component correlation function of this product converges to the Gaussian correlation

$$e^{-(h_i/\theta_i)^2}. \tag{3.4.7}$$

The PErK code used to fit the REML estimates of the correlation parameters and to predict the output is specified as follows.

PErK job to determine REML estimator, Example 3.8

```
CorrelationFamily = Matern
CorrelationType = 1
RandomError = No
Tries = 5
LogLikelihoodTolerance = 1.e-5
SimplexTolerance = 1.e-5
RandomNumberSeed = 26841
CorrelationEstimation = REML
CrossValidate
# Input stage
Ranges < ranges
X < sobol-40.x
Y < sobol-40.y
XPred < grid-320.x
YTrue < grid-320.y
RegressionModel < reg.constant
# Output stage
Summary > reml.summary
RegressionModel > reml.beta
StochasticProcessModel > reml.corpar
Predictions > reml.ypred
```

The inputs to this PErK run are two files that contain the $\boldsymbol{x}$ and $y(\boldsymbol{x})$ data; these are the 40×4 file `sobol-40.x` and the 40×1 file `sobol-40.y`, respectively. The output file `reml.summary`, below, contains a summary of the quality of the fit including the root mean squared prediction error

$$\sqrt{\frac{1}{320}\sum_{i=1}^{320}\left(y(\boldsymbol{x}_i) - \widehat{y}(\boldsymbol{x}_i)\right)^2} = 0.46891$$

for the 320 point test grid and a similar root mean squared prediction error for the cross-validated predictions of the 40 point training data.

reml.summary output file for Example 3.8

```
Number of Design Sites = 40
Correlation Family = Matern I
Random Error = No
Stochastic Process Variance = 7386.59136
Restricted Log Likelihood = -4.65190
Number of Restricted Log Likelihood Evaluations = 5594

Condition Number of Cholesky Square Root
of Correlation Matrix = 1252.53494

Cross Validation RMSPE = 0.40972
Cross Validation MAD = 0.74330
Case = 29

Number of Prediction Sites = 320
Prediction RMSPE = 0.46891
Prediction MAD = 1.52242
Case = 260
```

The 320 predictions and their estimated standard errors under the Gaussian Stochastic Process model are listed in `reml.ypred`. Figure 3.11 plots the predicted versus the true times for the 320 points. The predicted values are close to the true values throughout the input variable space.

The estimated constant of the process is contained in the `reml.beta` output file.

reml.beta output file for Example 3.8

```
The estimates of the linear model parameters are:

Parameter Beta
1         23.58701
```

The estimated correlation parameters are given in the file `reml.corpar`.

reml.corpar output file for Example 3.8

```
Correlation Family = Matern I

REML estimates of the correlation range parameters are:

Case Range
1         12.34622
2         6.88973
3         9.52963
4         7.08559

The REML of the correlation smoothness parameter is:
1.67681
```

The REML estimate of ν is 1.68 while the REMLs of the scale parameters $\theta_1, \ldots, \theta_4$ range from 6.89 to 12.35. ■

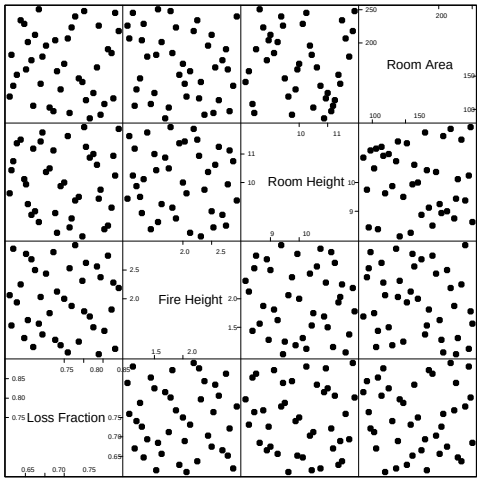


Fig. 3.9 Scatterplot matrix of the 40 input points used in Example 3.8.

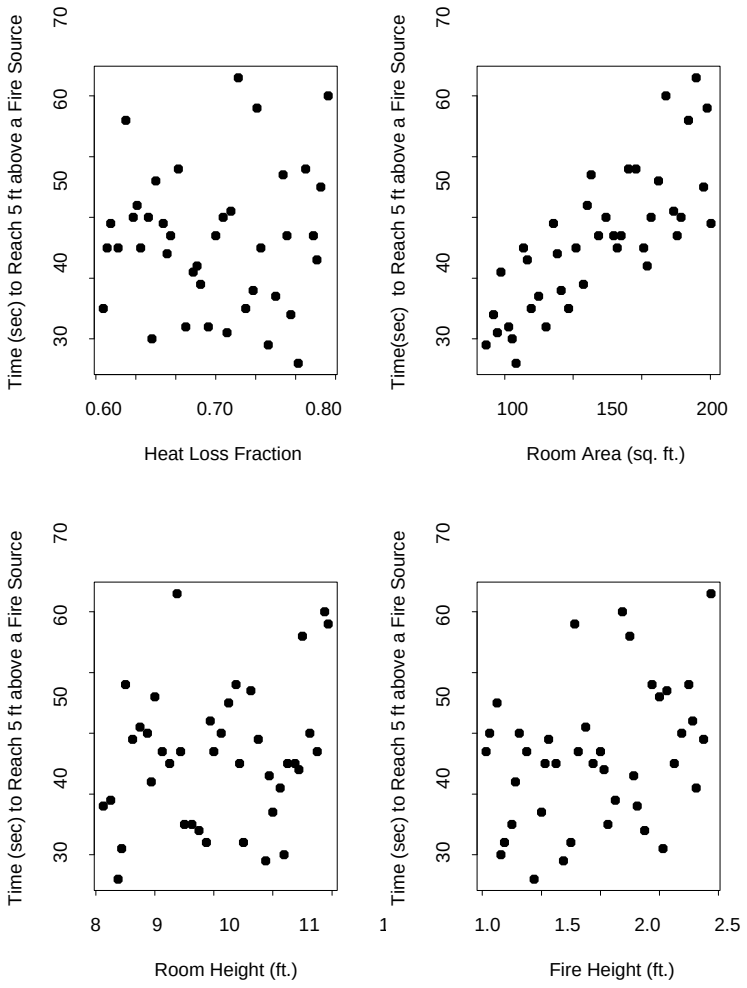


Fig. 3.10 Scatterplots of the time for a fire to reach five feet above a fire source versus each of the inputs: (1) room height, (2) room area, (3) heat loss fraction, and (4) height of the fire source above the floor, using the data from Example 3.8.

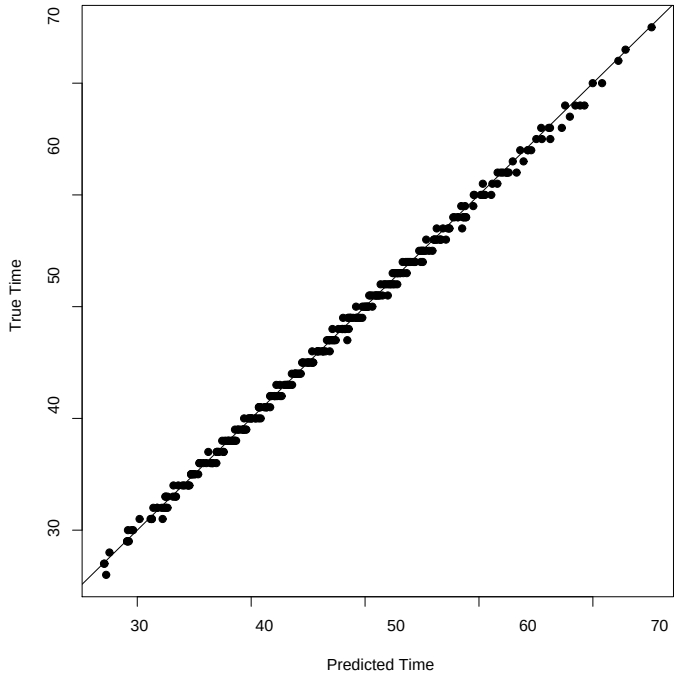


Fig. 3.11 Scatterplot of the true versus predicted times to reach five feet above a fire source for the equispaced grid of 320 points used in Example 3.8.

3.5 Prediction for Multivariate Output Simulators

s

3.6 Chapter Notes

3.6.1 Proof That (3.2.21) Is a BLUP (page 66)

The predictor $\widehat{Y_0}$ is linear because

$$\widehat{Y_0} = \boldsymbol{f}_0^\top (\boldsymbol{F}^\top \boldsymbol{F})^{-1} \boldsymbol{F}^\top \boldsymbol{Y}^n = \boldsymbol{a}_\star^\top \boldsymbol{Y}^n \,. \tag{3.6.1}$$

Furthermore $\widehat{Y_0}$ is unbiased because

$$\boldsymbol{F}^\top \boldsymbol{a}_\star = \boldsymbol{F}^\top \boldsymbol{F} (\boldsymbol{F}^\top \boldsymbol{F})^{-1} \boldsymbol{f}_0 = \boldsymbol{f}_0.$$

To see that (3.6.1) minimizes the MSPE pick any $\boldsymbol{a}$ for which $\boldsymbol{a}^\top \boldsymbol{Y}^n$ is unbiased, i.e., any $\boldsymbol{a}$ for which $\boldsymbol{F}^\top \boldsymbol{a} = \boldsymbol{f}_0$, and fix any $(\boldsymbol{\beta}, \sigma_\epsilon^2)$. Then the MSPE for this moment model is

$$\begin{aligned}
E\left\{\left(\boldsymbol{a}^\top \boldsymbol{Y}^n - Y_0\right)^2\right\} &= E\left\{\left(\boldsymbol{a}^\top (\boldsymbol{F}\boldsymbol{\beta} + \boldsymbol{\epsilon}^n) - \boldsymbol{f}_0^\top \boldsymbol{\beta} - \epsilon_0\right)^2\right\} \\
&= E\left\{\left(\boldsymbol{\beta}^\top (\boldsymbol{F}^\top \boldsymbol{a} - \boldsymbol{f}_0) + \sum_{i=1}^n a_i \epsilon_i - \epsilon_0\right)^2\right\} \\
&= E\left\{\left(\sum_{i=1}^n a_i \epsilon_i - \epsilon_0\right)^2\right\} \qquad (3.6.2) \\
&= \sigma_\epsilon^2 \left(\sum_{i=1}^n a_i^2 + 1\right) = \sigma_\epsilon^2 \left(\boldsymbol{a}^\top \boldsymbol{a} + 1\right). \qquad (3.6.3)
\end{aligned}$$

Equality holds in (3.6.2) because $\boldsymbol{a}$ satisfies the unbiasedness condition (3.2.20) and equality holds in (3.6.3) because the measurement errors are uncorrelated. This shows that the BLUP corresponds to that choice of $\boldsymbol{a}$ that minimizes $\boldsymbol{a}^\top \boldsymbol{a}$ subject to $\boldsymbol{F}^\top \boldsymbol{a} = \boldsymbol{f}_0$. But for any such $\boldsymbol{a}$,

$$\begin{aligned}
\boldsymbol{a}^\top \boldsymbol{a} &= (\boldsymbol{a} - \boldsymbol{a}_\star + \boldsymbol{a}_\star)^\top (\boldsymbol{a} - \boldsymbol{a}_\star + \boldsymbol{a}_\star) \\
&= (\boldsymbol{a} - \boldsymbol{a}_\star)^\top (\boldsymbol{a} - \boldsymbol{a}_\star) + \boldsymbol{a}_\star^\top \boldsymbol{a}_\star \\
&\quad + 2\,(\boldsymbol{a} - \boldsymbol{a}_\star)^\top \boldsymbol{a}_\star \\
&= (\boldsymbol{a} - \boldsymbol{a}_\star)^\top (\boldsymbol{a} - \boldsymbol{a}_\star) + \boldsymbol{a}_\star^\top \boldsymbol{a}_\star \qquad (3.6.4) \\
&\geq \boldsymbol{a}_\star^\top \boldsymbol{a}_\star, \qquad (3.6.5)
\end{aligned}$$

where (3.6.1) defines $\boldsymbol{a}_\star$. Equality holds in (3.6.4) because the cross product is zero when $\boldsymbol{F}^\top \boldsymbol{a} = \boldsymbol{f}_0$:

$$\begin{aligned} \boldsymbol{a} - \boldsymbol{a}_\star = \boldsymbol{a} - \boldsymbol{F}\left(\boldsymbol{F}^\top \boldsymbol{F}\right)^{-1} \boldsymbol{f}_0 &= \boldsymbol{a} - \boldsymbol{F}\left(\boldsymbol{F}^\top \boldsymbol{F}\right)^{-1} \boldsymbol{F}^\top \boldsymbol{a} \\ &= \left(\boldsymbol{I} - \boldsymbol{F}\left(\boldsymbol{F}^\top \boldsymbol{F}\right)^{-1} \boldsymbol{F}^\top\right) \boldsymbol{a} \end{aligned}$$

which implies

$$\begin{aligned} (\boldsymbol{a} - \boldsymbol{a}_\star)^\top \boldsymbol{a}_\star &= \boldsymbol{a}^\top \left(\boldsymbol{I} - \boldsymbol{F}\left(\boldsymbol{F}^\top \boldsymbol{F}\right)^{-1} \boldsymbol{F}^\top\right) \times \left(\boldsymbol{F}\left(\boldsymbol{F}^\top \boldsymbol{F}\right)^{-1} \boldsymbol{f}_0\right) \\ &= \boldsymbol{a}^\top \left(\boldsymbol{a}_\star - \boldsymbol{F}\left(\boldsymbol{F}^\top \boldsymbol{F}\right)^{-1} \left(\boldsymbol{F}^\top \boldsymbol{F}\right)\left(\boldsymbol{F}^\top \boldsymbol{F}\right)^{-1} \boldsymbol{f}_0\right) \\ &= 0\,. \end{aligned}$$

Furthermore this argument shows that the BLUP is unique because equality holds in (3.6.5) if and only if $\boldsymbol{a} = \boldsymbol{a}_\star$. □

3.6.2 Proof That (3.3.4) Is a BLUP (page 68)

This proof is more complicated than its measurement error counterpart studied in Example 3.6 of Section 3.2. However, part of the argument used in Example 3.6 can be retained here. The class of LUPs of $Y(\boldsymbol{x}_0)$ with respect to (3.3.3) depends only on the first moment of $(Y_0, \boldsymbol{Y}^n)$ and hence is the same as for Example 3.6. The predictor $\widehat{Y}(\boldsymbol{x}_0) = a_0 + \boldsymbol{a}^\top \boldsymbol{Y}^n$ is unbiased for $Y(\boldsymbol{x}_0)$ provided

$$a_0 = 0 \qquad \text{and} \qquad \boldsymbol{F}^\top \boldsymbol{a} = \boldsymbol{f}_0. \tag{3.6.6}$$

Now fix any LUP of $Y(\boldsymbol{x}_0)$, say $\boldsymbol{a}^\top \boldsymbol{Y}^n$. Let $\boldsymbol{Z}^n = (Z(\boldsymbol{x}_1), \ldots, Z(\boldsymbol{x}_n))^\top$ and $Z_0 = Z(\boldsymbol{x}_0)$ be the corresponding stochastic process parts of $\boldsymbol{Y}^n$ and $Y(\boldsymbol{x}_0)$ in (3.3.1), respectively. For fixed $\boldsymbol{\beta}$ and σ_z^2, the MSPE of $\boldsymbol{a}^\top \boldsymbol{Y}^n$ is

$$\begin{aligned} E\{(\boldsymbol{a}^\top \boldsymbol{Y}^n - Y_0)^2\} &= E\{(\boldsymbol{a}^\top (\boldsymbol{F}\boldsymbol{\beta} + \boldsymbol{Z}^n) - (\boldsymbol{f}_0^\top \boldsymbol{\beta} + Z_0))^2\} \\ &= E\{((\boldsymbol{a}^\top \boldsymbol{F} - \boldsymbol{f}_0^\top)\boldsymbol{\beta} + \boldsymbol{a}^\top \boldsymbol{Z}^n - Z_0)^2\} \\ &= E\{\boldsymbol{a}^\top \boldsymbol{Z}^n (\boldsymbol{Z}^n)^\top \boldsymbol{a} \\ &\quad - 2\boldsymbol{a}^\top \boldsymbol{Z}^n Z_0 + Z_0^2\} \qquad (3.6.7) \\ &= \sigma_z^2 \boldsymbol{a}^\top \boldsymbol{R}\boldsymbol{a} - 2\sigma_z^2 \boldsymbol{a}^\top \boldsymbol{r}_0 + \sigma_z^2 \\ &= \sigma_z^2 \left(\boldsymbol{a}^\top \boldsymbol{R}\boldsymbol{a} - 2\boldsymbol{a}^\top \boldsymbol{r}_0 + 1\right), \qquad (3.6.8) \end{aligned}$$

where (3.6.7) follows from (3.6.6). Thus the BLUP chooses $\boldsymbol{a}$ to minimize

$$\boldsymbol{a}^\top \boldsymbol{R}\boldsymbol{a} - 2\boldsymbol{a}^\top \boldsymbol{r}_0 \tag{3.6.9}$$

subject to

$$\boldsymbol{F}^\top \boldsymbol{a} = \boldsymbol{f}_0\,. \tag{3.6.10}$$

The method of Lagrange multipliers can be used to minimize the quadratic objective function (3.6.9) subject to linear constraints (3.6.10). We find $(\boldsymbol{a}, \boldsymbol{\lambda}) \in \mathbb{R}^{n+p}$ to minimize

$$\boldsymbol{a}^\top \boldsymbol{R}\boldsymbol{a} - 2\boldsymbol{a}^\top \boldsymbol{r}_0 + 2\boldsymbol{\lambda}^\top (\boldsymbol{F}^\top \boldsymbol{a} - \boldsymbol{f}_0)\,. \tag{3.6.11}$$

Calculating the gradient of (3.6.11) with respect to $(\boldsymbol{a}, \boldsymbol{\lambda})$ and setting it equal to the zero vector gives the system of equations

$$\begin{aligned} \boldsymbol{F}^\top \boldsymbol{a} - \boldsymbol{f}_0 &= \boldsymbol{0} \\ \boldsymbol{R}\boldsymbol{a} - \boldsymbol{r}_0 + \boldsymbol{F}\boldsymbol{\lambda} &= \boldsymbol{0} \end{aligned}$$

or

$$\begin{pmatrix} 0 & \boldsymbol{F}^\top \\ \boldsymbol{F} & \boldsymbol{R} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_0 \\ \boldsymbol{r}_0 \end{pmatrix},$$

which implies

$$\begin{aligned} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{pmatrix} &= \begin{pmatrix} 0 & \boldsymbol{F}^\top \\ \boldsymbol{F} & \boldsymbol{R} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{f}_0 \\ \boldsymbol{r}_0 \end{pmatrix} \\ &= \begin{pmatrix} -\boldsymbol{Q} & \boldsymbol{Q}\boldsymbol{F}^\top \boldsymbol{R}^{-1} \\ \boldsymbol{R}^{-1}\boldsymbol{F}\boldsymbol{Q} & \boldsymbol{R}^{-1} - \boldsymbol{R}^{-1}\boldsymbol{F}\boldsymbol{Q}\boldsymbol{F}^\top \boldsymbol{R}^{-1} \end{pmatrix} \times \begin{pmatrix} \boldsymbol{f}_0 \\ \boldsymbol{r}_0 \end{pmatrix}, \end{aligned}$$

where $\boldsymbol{Q} = (\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F})^{-1}$. After a small amount of algebra, the $\boldsymbol{a}$ solution gives (3.3.4) as the BLUP for the family (3.3.3). □

3.6.3 Implementation Issues

The calculation of either the MLE or the REML of the correlation parameters requires the repeated evaluation of the determinant and inverse of the $n \times n$ matrix $\boldsymbol{R}$. The Cholesky decomposition provides the most numerically stable method of calculating these quantities (Harville (1997)). Nevertheless, the repeated evaluation of these quantities is the most time consuming aspect of algorithms that sequentially add data. As an example, Williams et al (2000) report the times to maximize the REML likelihood which is required during the execution of their global optimization algorithm. In a six-dimensional input case, they fit the Matérn correlation function with a *single* shape parameter and separate range parameters for each input (a seven-dimensional $\boldsymbol{\psi}$ correlation parameter). When 50 training points were used, their optimization of the $\boldsymbol{\psi}$ likelihood (3.3.16) required 2,140 seconds of Sun Ultra 5 CPU time and this optimization required 4,230 seconds of CPU time for 82 training points. Fitting the power exponential model was faster with 1,105 seconds of CPU time required for the 50 point case and 3,100 seconds of CPU time for the 82 point case. Indeed, applications that require a sequence of correlation parameter estimates

for increasing n often re-estimate these parameters only periodically, for example, when every fifth point is added to the design. A more rational plan is to re-estimate the correlation parameters more often for small n and more frequently for large n. For sufficiently large n, these estimators become intractable to calculate.

The dimension of the optimization problems required to find MLEs and REMLs can be large. For example, in a product exponential model with 20 input variables, each having unknown scale and power parameters, $\boldsymbol{\psi}$ is 40-dimensional. Such high-dimensional likelihood surfaces tend to have many local maxima, making global optimization difficult.

A variety of algorithms have been successfully used to determine MLEs and REMLs of correlation parameters. Among these are the Nelder-Mead simplex algorithm (Nelder and Mead (1965)), branch and bound algorithms (Jones et al (1998)), and stochastic global optimization algorithms (Rinnooy Kan and Timmer (1984)). As noted above, the primary feature of a successful algorithm is that it must be capable of handling many local maxima in order to find a global maximum. There has been limited head-to-head comparison of the efficiency of these algorithms in finding optima.

As an example, to address high-dimensional MLE and REML parameter estimation problems, Welch et al (1992) proposed using a dimensionality reduction scheme to perform a series of presumably simpler optimizations. The idea is to make tractable the high-dimensional $\boldsymbol{\psi}$ minimization in (3.3.14) or (3.3.16) by constraining the number of free parameters allowed in the minimization; only "important" input variables are allowed to possess their own unconstrained correlation parameters. They illustrate the method for the Gaussian correlation family (2.4.6) where $\boldsymbol{\psi} = (\theta_1, \ldots, \theta_d)$.

First, each of the d input variables must be scaled to have the *same range*. At each stage of the process, let C denote the indices of the variables having *common* values of the correlation parameters for that step and let $C \setminus \{j\}$ denote the set difference of C and $\{j\}$. In the following meta-code, *S0* is an initialization step while *S1* and *S2* are induction steps.

S0 Set $C = \{1, 2, \ldots, d\}$, i.e., $\psi_1 = \cdots = \psi_d = \psi$. Maximize (3.3.14) or (3.3.16) as a function of ψ and denote the resulting log likelihood by ℓ_0.

S1 For each $j \in C$, maximize (3.3.14) or (3.3.16) under the constraint that variables ψ_h with $h \in C \setminus \{j\}$ have a common value and ψ_j varies freely. Denote the result by ℓ_j.

S2 Let $j^{\max}$ denote the variable producing the largest increase in $\ell_j - \ell_0$ for $j \in C$.

S3 If $\ell_{j^{\max}} - \ell_0$ represents a "significant" increase in the log likelihood as judged by a stopping criterion, then update C to be $C \setminus j^{\max}$, ℓ_0 to be $\ell_{j^{\max}}$, and *fix* $\psi_{j^{\max}}$ at its value estimated in *S1*. Continue the next iteration at Step *S1*. Otherwise, stop the algorithm and estimate the correlation parameters to be the values produced by the previous iteration.

Variations are, of course, possible in this alg. For example, two-dimensional optimizations are used in every cycle of *S1* because all $\psi_{j^{\max}}$ estimated in previous cycles are fixed in subsequent ones. Instead, *S1* could allow the ψ_j values previ-

ously estimated to vary freely along with the next individual ψ_j to be estimated. Of course, the number of variables in the maximization would increase at each step of the algorithm.

3.6.4 Alternate Predictors

This chapter has focused on the use of empirical best linear unbiased prediction, also known as empirical kriging prediction in the geostatistics literature. Empirical kriging methodology becomes numerically unstable when the size of the training sample, n, is large because the predictor (3.3.9) requires the inversion of an $n \times n$ matrix, which can be near-singular for certain correlation functions and choices of inputs $\boldsymbol{x}$. While numerous authors have written code to make empirical kriging more efficient (see An and Owen (2001) for some analysis of the computational burden), there is a point beyond which empirical kriging cannot be used. Hence several other approaches have been investigated in the literature that are computationally simpler than empirical kriging. We mention two of these methods.

One method of prediction that leads to computationally simpler predictors is to use the Gaussian random field model with a "Markov random field" model for the dependence structure. The special structure of the resulting correlation matrix allows for its analytic inversion and the usual empirical kriging predictor (3.3.9) has a simple form. See Cressie (1993), page 364, for a summary of the properties of MRF-based predictors and for additional references.

An and Owen (2001) described a predictor that they dubbed "quasi-regression." Their method exploits the use of an orthogonal basis function system to relate the inputs to the computer output. These methods are extremely computationally efficient and a wide variety of basis systems can be used.

Chapter 4
Bayesian Prediction of Computer Simulation Output

4.1 Predictive Distributions

this version is used only as a place holder

4.1.1 Introduction

A predictive distribution for the random variable Y_0 is meant to capture all the information about Y_0 that is contained in $\boldsymbol{Y}^n = (Y_1, \ldots, Y_n)^\top$. Of course, knowing $\boldsymbol{Y}^n$ does not completely specify Y_0 but $\boldsymbol{Y}^n$ does provide a probability distribution of more likely and less likely values for Y_0 that is called the *predictive distribution* of Y_0 given $\boldsymbol{Y}^n$. This section derives predictive distributions useful for computer output based on two hierarchical models for $[Y_0, \boldsymbol{Y}^n]$. Section 3.5 considers prediction in the case of multiple response models, as described in Section 2.3.

Formally, the predictive distribution of Y_0 based on $\boldsymbol{Y}^n$ is defined to be the conditional distribution of Y_0 given $\boldsymbol{Y}^n$, which is denoted by $[Y_0 \mid \boldsymbol{Y}^n]$. The mean of the $[Y_0 \mid \boldsymbol{Y}^n]$ distribution arose earlier, in Equation (3.3.4) of Section 3.2, where we showed that $E\{Y_0 \mid \boldsymbol{Y}^n\}$ is the best MSPE predictor of Y_0.

This section derives predictive distributions for the output of computer experiments under two hierarchical models for $[Y_0, \boldsymbol{Y}^n]$. In the computer experiment application, $Y_0 = Y(\boldsymbol{x}_0)$ and $\boldsymbol{Y}^n$ is training data $(Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n))$. As one application of the predictive distribution, we place prediction bounds on the best MSPE predictor $\widehat{Y}(\boldsymbol{x}_0) = E\{Y_0 \mid \boldsymbol{Y}^n\}$.

Both families of hierarchical models have two stages with the first-stage for $Y(\cdot)$ based on the regression plus stationary Gaussian process model introduced in Chapter 2. This leads to the first-stage conditional distribution

$$\begin{pmatrix} Y_0 \\ \boldsymbol{Y}^n \end{pmatrix} \Bigg| \ \textit{parameters} \sim N_{1+n}\left[\begin{pmatrix} \boldsymbol{f}_0^\top \\ \boldsymbol{F} \end{pmatrix}\boldsymbol{\beta},\ \sigma_z^2 \begin{pmatrix} 1 & \boldsymbol{r}_0^\top \\ \boldsymbol{r}_0 & \boldsymbol{R} \end{pmatrix}\right] \tag{4.1.1}$$

for $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$. Here $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$ is a known $p \times 1$ vector of regression functions for Y_0; $\boldsymbol{F} = (f_j(\boldsymbol{x}_i))$ is a known $n \times p$ matrix of regression functions for the training data; $\boldsymbol{\beta}$ is an unknown $p \times 1$ vector of regression coefficients; $\boldsymbol{R} = (R(\boldsymbol{x}_i - \boldsymbol{x}_j))$ is a known $n \times n$ matrix of the correlations among the training data $\boldsymbol{Y}^n$; $\boldsymbol{r}_0 = (R(\boldsymbol{x}_i - \boldsymbol{x}_0))$ is a known $n \times 1$ vector of the correlations of Y_0 with $\boldsymbol{Y}^n$.

We emphasize that the models discussed in Subsections 4.1.2 and 4.1.3 assume that the correlation structure is *known*, in this case $\boldsymbol{R}$ and $\boldsymbol{r}_0$ are known. The first model assumes that $\boldsymbol{\beta}$ is unknown and σ_z^2 is known so that $\boldsymbol{\beta}$ is the conditioning *parameter* in (4.1.1) while the second model assumes that $(\boldsymbol{\beta}, \sigma_z^2)$ is unknown so that $(\boldsymbol{\beta}, \sigma_z^2)$ is the conditioning *parameter*. The predictive distributions corresponding to these two models are stated in Theorems 4.1 and 4.2. This section will focus on the interpretation and application of these two theorems. Sketches of their proofs will be deferred until Section **??**.

The assumption of known correlation structure is dropped in Subsection 4.1.4. There we consider the frequently occurring case that the correlation function is parametric with a form that is known, up to a vector of (unknown) *parameters*.

4.1.2 Predictive Distributions When σ_z^2, R, and $\boldsymbol{r}_0$ Are Known

The following theorem specifies the predictive distribution of Y_0 for two different choices of second-stage priors for $\boldsymbol{\beta}$. The first, a normal prior, can be regarded as an informative choice while the second can be thought of as non-informative. The non-informative prior is formally obtained by letting the variance $\tau^2 \to \infty$ in the normal one.

Theorem 4.1. Suppose $(Y_0, \boldsymbol{Y}^n)$ follows a two-stage model with known σ_z^2 in which

$$[(Y_0, \boldsymbol{Y}^n)\,|\boldsymbol{\beta}] \sim N_{1+n}\left[\begin{pmatrix} \boldsymbol{f}_0^\top \\ \boldsymbol{F} \end{pmatrix}\boldsymbol{\beta},\ \sigma_z^2 \begin{pmatrix} 1 & \boldsymbol{r}_0^\top \\ \boldsymbol{r}_0 & \boldsymbol{R} \end{pmatrix}\right].$$

(a) If

$$\boldsymbol{\beta} \sim N_p\left[\boldsymbol{b}_0, \tau^2 \boldsymbol{V}_0\right], \tag{4.1.2}$$

where $\boldsymbol{b}_0$, $\boldsymbol{V}_0$, and τ^2 are known, then Y_0 has the predictive distribution

$$[Y_0 \mid \boldsymbol{Y}^n = \boldsymbol{y}^n] \sim N_1\left[\mu_{0|n}, \sigma_{0|n}^2\right], \tag{4.1.3}$$

where

$$\mu_{0|n} = \mu_{0|n}(\boldsymbol{x}_0) = \boldsymbol{f}_0^\top \boldsymbol{\mu}_{\boldsymbol{\beta}|n} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}\left(\boldsymbol{y}^n - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{\beta}|n}\right), \tag{4.1.4}$$

for

$$\boldsymbol{\mu}_{\boldsymbol{\beta}|n} = \left(\frac{\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F}}{\sigma_Z^2} + \frac{\boldsymbol{V}_0^{-1}}{\tau^2}\right)^{-1}\left(\frac{\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{y}^n}{\sigma_Z^2} + \frac{\boldsymbol{V}_0^{-1}\boldsymbol{b}_0}{\tau^2}\right), \tag{4.1.5}$$

and

$$\sigma_{0|n}^2 = \sigma_{0|n}^2(\boldsymbol{x}_0) = \sigma_Z^2\left\{1 - (\boldsymbol{f}_0^\top, \boldsymbol{r}_0^\top)\begin{bmatrix} -\frac{\sigma_Z^2}{\tau^2}\boldsymbol{V}_0^{-1} & \boldsymbol{F}^\top \\ \boldsymbol{F} & \boldsymbol{R}\end{bmatrix}^{-1}\begin{pmatrix}\boldsymbol{f}_0 \\ \boldsymbol{r}_0\end{pmatrix}\right\}. \tag{4.1.6}$$

(b) If

$$[\boldsymbol{\beta}] \sim 1 \tag{4.1.7}$$

on $\mathbb{R}^p$, then Y_0 has the predictive distribution

$$[Y_0 \mid \boldsymbol{Y}^n = \boldsymbol{y}^n] \sim N_1\left[\mu_{0|n}, \sigma_{0|n}^2\right], \tag{4.1.8}$$

where $\mu_{0|n} = \mu_{0|n}(\boldsymbol{x}_0)$ and $\sigma_{0|n}^2 = \sigma_{0|n}^2(\boldsymbol{x}_0)$ are given by (4.1.4) and (4.1.6), respectively, with the substitution $\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{y}^n$ for $\boldsymbol{\mu}_{\boldsymbol{\beta}|n}$ in (4.1.4) and the $p \times p$ zero matrix for $-\frac{\sigma_Z^2}{\tau^2}\boldsymbol{V}_0^{-1}$ in (4.1.6).

There are several interesting features of (a) and (b) that aid in their interpretation. First, there is a "continuity" in the priors and posteriors as $\tau^2 \to \infty$. Concerning the priors, we had earlier observed that the non-informative prior (4.1.7) is the limit of the normal prior (4.1.2) as $\tau^2 \to \infty$. On the posterior side and paralleling this prior convergence, it can be calculated that the posterior mean $\mu_{0|n}(\boldsymbol{x}_0)$ and the posterior variance $\sigma_{0|n}^2(\boldsymbol{x}_0)$ in (4.1.8) for the non-informative prior are the limits, as $\tau^2 \to \infty$ of posterior mean and variance for the informative normal prior, which are (4.1.4) and (4.1.6), respectively. A second interesting feature is that while the prior (4.1.7) is intuitively non-informative, it is *not* a proper distribution. Nevertheless, the corresponding predictive distribution is proper. Indeed, we saw in (3.2.12) that $\widehat{\boldsymbol{\beta}}$ is the posterior mean of $\boldsymbol{\beta}$ given the data $\boldsymbol{Y}^n = \boldsymbol{y}^n$ for this two-stage model (and is the generalized least squares estimator of $\boldsymbol{\beta}$ from a frequentist viewpoint). Lastly, recall that in Subsection 3.3.2 we used a conditioning argument to derive the formula

$$\boldsymbol{f}_0^\top \widehat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}\left(\boldsymbol{y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)$$

in (4.1.8) as the predictive mean for the non-informative prior. This same type of conditioning can be applied to derive posterior mean (4.1.4)–(4.1.5) for the normal prior (4.1.2) for $\boldsymbol{\beta}$.

To understand the implications of Theorem 4.1, we examine some properties of the mean and variance of the predictive distribution (4.1.3). Both $\mu_{0|n}(\boldsymbol{x}_0)$ and $\sigma_{0|n}^2(\boldsymbol{x}_0)$ depend on $\boldsymbol{x}_0$ only through the regression functions $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$ and the correlation vector $\boldsymbol{r}_0 = \boldsymbol{r}(\boldsymbol{x}_0)$. Focusing first on $\mu_{0|n}(\boldsymbol{x}_0)$, a little algebra shows that $\mu_{0|n}(\boldsymbol{x}_0)$ is linear in $\boldsymbol{Y}^n$ and, with additional calculation, that it is an unbiased predictor of $Y(\boldsymbol{x}_0)$, i.e., $\mu_{0|n}(\boldsymbol{x}_0)$ is a linear unbiased predictor of $Y(\boldsymbol{x}_0)$.

Second, the continuity and other smoothness properties of $\mu_{0|n}(\boldsymbol{x}_0)$ are inherited from those of the correlation function $R(\cdot)$ and the regressors $\{f_j(\cdot)\}_{j=1}^p$ because

$$\mu_{0|n}(\boldsymbol{x}_0) = \sum_{j=1}^{p} f_j(\boldsymbol{x}_0)\mu_{\boldsymbol{\beta}|n,j} + \sum_{i=1}^{n} d_i R(\boldsymbol{x}_0 - \boldsymbol{x}_i),$$

where $\mu_{\boldsymbol{\beta}|n,j}$ is the j^{th} element of $\boldsymbol{\mu}_{\boldsymbol{\beta}|n}$. Previously, Subsection **??** had observed a parallel behavior for the BLUP (3.3.4), which is exactly the predictive mean $\mu_{0|n}(\boldsymbol{x}_0)$ in part (b) of Theorem 4.1.

Third, $\mu_{0|n}(\boldsymbol{x}_0)$ depends on the parameters σ_z^2 and τ^2 only through their ratio. This is because

$$\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{\beta}|n} &= \left(\frac{\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{F}}{\sigma_z^2} + \frac{\boldsymbol{V}_0^{-1}}{\tau^2}\right)^{-1}\left(\frac{\boldsymbol{F}\boldsymbol{R}^{-1}\boldsymbol{y}_n}{\sigma_z^2} + \frac{\boldsymbol{V}_0^{-1}\boldsymbol{b}_0}{\tau^2}\right) \\
&= (\sigma_z^2)\left(\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{F} + \frac{\sigma_z^2}{\tau^2}\boldsymbol{V}_0^{-1}\right)^{-1} \\
&\qquad \times \ (\sigma_z^2)^{-1}\left(\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{y}^n + \frac{\sigma_z^2}{\tau^2}\boldsymbol{V}_0^{-1}\boldsymbol{b}_0\right) \\
&= \left(\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{F} + \frac{\sigma_z^2}{\tau^2}\boldsymbol{V}_0^{-1}\right)^{-1}\left(\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{y}^n + \frac{\sigma_z^2}{\tau^2}\boldsymbol{V}_0^{-1}\boldsymbol{b}_0\right).
\end{aligned}$$

.

Lastly, the mean predictors $\mu_{0|n}(\boldsymbol{x}_0)$ in Theorem 4.1 interpolate the training data. This is true because when $\boldsymbol{x}_0 = \boldsymbol{x}_i$ for some $i \in \{1, \ldots, n\}$, $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_i)$, and $\boldsymbol{r}_0^\top\boldsymbol{R}^{-1} = \boldsymbol{e}_i^\top$, the i^{th} unit vector. Thus

$$\begin{aligned}
\mu_{0|n}(\boldsymbol{x}_i) &= \boldsymbol{f}^\top(\boldsymbol{x}_i)\boldsymbol{\mu}_{\boldsymbol{\beta}|n} + \boldsymbol{r}_0^\top\boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{\beta}|n}) \\
&= \boldsymbol{f}^\top(\boldsymbol{x}_i)\boldsymbol{\mu}_{\boldsymbol{\beta}|n} + \boldsymbol{e}_i^\top(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{\beta}|n}) \\
&= \boldsymbol{f}^\top(\boldsymbol{x}_i)\boldsymbol{\mu}_{\boldsymbol{\beta}|n} + (Y_i - \boldsymbol{f}^\top(\boldsymbol{x}_i)\boldsymbol{\mu}_{\boldsymbol{\beta}|n}) \\
&= Y_i\,.
\end{aligned}$$

Example 4.1. This example illustrates the effect of various choices of the prior $[\boldsymbol{\beta}]$ on the mean of the predictive distribution which is stated in Theorem 4.1. We use the same true function

$$y(x) = e^{-1.4x}\cos(7\pi x/2)\,, \qquad 0 < x < 1,$$

and $n = 7$ point training data as in Examples 3.3 and 3.7. The predictive distribution of $Y(\boldsymbol{x}_0)$ is based on the two-stage model whose first stage is the stationary stochastic process

$$Y(x) \,|\, \beta_0 = \beta_0 + Z(x)\,, \qquad 0 < x < 1,$$

where $\beta_0 \in \mathbb{R}$ and $R(h) = \exp\{-136.1 \times h^2\}$.

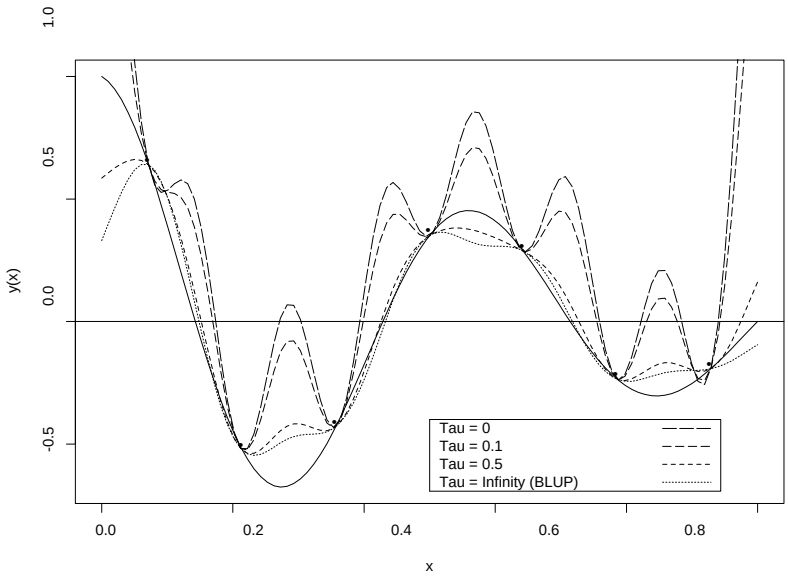


Fig. 4.1 The predictor $\mu_{0|n} = \mu_{\beta_0|n} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^n - \mathbf{1}_n \mu_{\beta_0|n})$ in (4.1.9) and (4.1.10) with $b_0 = 5$, $\sigma_z = .41$, and four choices of τ^2.

Suppose we take $\beta_0 \sim N(b_0, \tau^2 \times v_0^2)$ in part (a) of Theorem 4.1 and $v_0 = 1$ to guarantee identifiability of the prior variance. Both b_0 and τ^2 are assumed known. The mean of the posterior, Equation (4.1.4), is

$$\mu_{0|n} = \mu_{\beta_0|n} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \left(\boldsymbol{y}^n - \mathbf{1}_n \mu_{\beta_0|n} \right), \tag{4.1.9}$$

where $\mu_{\beta_0|n}$ is the posterior mean of β_0 given $\boldsymbol{Y}^n$ which is

$$\begin{aligned} \mu_{\beta_0|n} = \mu_{\beta_0|n}(b_0, \tau^2) &= \frac{(\mathbf{1}_n^\top \boldsymbol{R}^{-1} \boldsymbol{y}^n + b_0 \sigma_z^2/\tau^2)}{(\mathbf{1}_n^\top \boldsymbol{R}^{-1} \mathbf{1}_n + \sigma_z^2/\tau^2)} \\ &= \omega b_0 + (1-\omega)(\mathbf{1}_n^\top \boldsymbol{R}^{-1} \mathbf{1}_n^\top)^{-1} (\mathbf{1}_n^\top \boldsymbol{R}^{-1} \boldsymbol{y}^n) \\ &= \omega b_0 + (1-\omega)\widehat{\beta_0}, \end{aligned} \tag{4.1.10}$$

where $\omega = \sigma_z^2/[\tau^2 \mathbf{1}_n^\top \boldsymbol{R}^{-1} \mathbf{1}_n + \sigma_z^2] \in (0,1)$. In words, (4.1.10) can be interpreted as saying that the posterior mean of β_0 given $\boldsymbol{Y}^n$ is a convex combination of the MLE of β_0 and its prior mean, which are $\widehat{\beta_0}$, the generalized least squares estimator of β_0 and b_0, respectively. The behavior of the weight ω provides additional intuition about two extreme cases of $\mu_{\beta_0|n}$. When the prior certainty in b_0 increases in such a way that $\tau^2 \to 0$ for fixed process variance σ_z^2, then $\omega \to 1$ and $\mu_{0|n} \to b_0$, meaning that the predictor uses only the prior and ignores the data, which is reasonable for perfect prior information. Similarly, when the prior certainty in b_0 decreases in such

a way that $\tau^2 \to \infty$ for fixed process variance σ_z^2, then $\omega \to 0$ and $\mu_{0|n} \to \widehat{\beta_0}$ so the predictor uses only the data and ignores the prior, which is, again, intuitively reasonable when there is no prior information.

Figure 4.1 shows the effect of changing the prior on $\mu_{0|n}(x_0)$; remember that $\mu_{0|n}(x_0)$ depends not only on $\mu_{\beta_0|n}$ but also on the correction term $\boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^n - \mathbf{1}_n \mu_{\beta_0|n})$. The four predictors correspond to $b_0 = 5$, $\sigma_z = .41$, and four τ^2 values, with a fixed power exponential correlation function. *Smaller* τ^2 values produce predictors that have greater excursions from the data than do predictors having *greater* τ^2 values. In this case, the predictors having smaller τ^2 produce larger excursions from the true curve than does the BLUP (3.3.4) (which equals $\mu_{0|n}(x_0)$ with $\tau^2 = \infty$). This prior mean of β_0 was purposely taken to be the "large" value of $b_0 = 5.0$ which is not near the data to illustrate the effect of τ^2. Smaller τ^2 values correspond to being more certain about the prior and thus, the predictor pulls away from the data except when the training data pull it back. ■

Turning attention to the variance of the predictive distribution, $\sigma^2_{0|n}(\boldsymbol{x}_0)$, first observe that this quantity can be interpreted as the (unconditional) mean squared prediction error of $\mu_{0|n}(\boldsymbol{x}_0)$ because

$$\begin{aligned}
\text{MSPE}(\mu_{0|n}(\boldsymbol{x}_0)) &= E\left\{\left(Y(\boldsymbol{x}_0) - \mu_{0|n}(\boldsymbol{x}_0)\right)^2\right\} \\
&= E\left\{\left(Y(\boldsymbol{x}_0) - E\left\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n\right\}\right)^2\right\} \\
&= E\left\{E\left\{\left(Y(\boldsymbol{x}_0) - E\left\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n\right\}\right)^2 |\boldsymbol{Y}^n\right\}\right\} \\
&= E\left\{\sigma^2_{0|n}(\boldsymbol{x}_0)\right\} \\
&= \sigma^2_{0|n}(\boldsymbol{x}_0).
\end{aligned}$$

Thus $\sigma^2_{0|n}(\boldsymbol{x}_0)$ is the usual measure of precision of $\mu_{0|n}(\boldsymbol{x}_0)$.

The reader should be alert to the fact that $\sigma^2_{0|n}(\boldsymbol{x}_0)$ has a number of equivalent algebraic forms that are used in different papers and books (see Sacks et al (1989), Cressie (1993)). Using basic matrix manipulations and starting with (4.1.6), we obtain

$$\begin{aligned}
\sigma^2_{0|n} &= \sigma^2_z \left\{ 1 - (\boldsymbol{f}_0^\top, \boldsymbol{r}_0^\top) \begin{bmatrix} -\frac{\sigma^2_z}{\tau^2}\boldsymbol{V}_0^{-1} & \boldsymbol{F}^\top \\ \boldsymbol{F} & \boldsymbol{R} \end{bmatrix}^{-1} \begin{pmatrix} \boldsymbol{f}_0 \\ \boldsymbol{r}_0 \end{pmatrix} \right\} \\
&= \sigma^2_z \left\{ 1 - \left[-\boldsymbol{f}_0^\top \boldsymbol{Q}^{-1} \boldsymbol{f}_0 + 2\boldsymbol{f}_0^\top \boldsymbol{Q}^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0 \right.\right. \\
&\qquad \left.\left. + \ \boldsymbol{r}_0^\top \{ \boldsymbol{R}^{-1} - \boldsymbol{R}^{-1} \boldsymbol{F} \boldsymbol{Q}^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \} \boldsymbol{r}_0 \right] \right\} \qquad (4.1.11) \\
&= \sigma^2_z \{ 1 - \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0 + \boldsymbol{f}_0^\top \boldsymbol{Q}^{-1} \boldsymbol{f}_0 - 2\boldsymbol{f}_0^\top \boldsymbol{Q}^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0 \\
&\qquad + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \boldsymbol{F} \boldsymbol{Q}^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0 \} \\
&= \sigma^2_z \{ 1 - \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0 + \boldsymbol{h}^\top \boldsymbol{Q}^{-1} \boldsymbol{h} \}, \qquad (4.1.12)
\end{aligned}$$

where

$$\boldsymbol{Q} = \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F} + \frac{\sigma^2_z}{\tau^2} \boldsymbol{V}_0^{-1}, \qquad (4.1.13)$$

$\boldsymbol{h} = \boldsymbol{f}_0 - \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0$, and (4.1.11) follows from Lemma B.3. In particular, expression (4.1.12)

$$\sigma^2_{0|n} = \sigma^2_z \{ 1 - \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0 + \boldsymbol{h}^\top (\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F})^{-1} \boldsymbol{h} \}$$

is a frequently-used expression for the variance of the BLUP (3.3.4), i.e., for $\mu_{0|n}(\boldsymbol{x}_0)$ in Part (b) of Theorem 4.1. (See, for example, (5.3.15) of Cressie (1993).)

Intuitively, the variance of the posterior of $Y(\boldsymbol{x}_0)$ given $Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n)$ should be *zero* whenever $\boldsymbol{x}_0 = \boldsymbol{x}_i$ because we *know* exactly the response at each of the training data sites $\boldsymbol{x}_i$ and there is no measurement error term in our stochastic process model. To see that this *is* the case analytically, fix $\boldsymbol{x}_0 = \boldsymbol{x}_i$ for some $1 \le i \le n$, recall that $\boldsymbol{r}_0^\top \boldsymbol{R}^{-1} = \boldsymbol{e}_i^\top$, and observe that $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_i)$. From (4.1.12),

$$\begin{aligned}
\sigma^2_{0|n}(\boldsymbol{x}_i) &= \sigma^2_z \{ 1 - \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0 + (\boldsymbol{f}_0^\top - \boldsymbol{r}_0^\top \boldsymbol{R}^{-1} \boldsymbol{F}) \boldsymbol{Q}^{-1} (\boldsymbol{f}_0 - \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{r}_0) \} \\
&= \sigma^2_z \{ 1 - \boldsymbol{e}_i^\top \boldsymbol{r}(\boldsymbol{x}_i) + (\boldsymbol{f}^\top(\boldsymbol{x}_i) - \boldsymbol{e}_i^\top \boldsymbol{F}) \boldsymbol{Q}^{-1} (\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{F}^\top \boldsymbol{e}_i) \} \\
&= \sigma^2_z \{ 1 - 1 + (\boldsymbol{f}^\top(\boldsymbol{x}_i) - \boldsymbol{f}^\top(\boldsymbol{x}_i)) \boldsymbol{Q}^{-1} (\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{f}(\boldsymbol{x}_i)) \} \\
&= \sigma_z \{ 1 - 1 + 0 \} = 0
\end{aligned}$$

where $\boldsymbol{Q}$ is given in (4.1.13).

Perhaps the most important use of Theorem 4.1 is to provide pointwise predictive bands about the predictor $\mu_{0|n}(\boldsymbol{x}_0)$. The bands can be obtained by using the fact that

$$\frac{Y(\boldsymbol{x}_0) - \mu_{0|n}(\boldsymbol{x}_0)}{\sigma^2_{0|n}(\boldsymbol{x}_0)} \sim N(0, 1) \,.$$

This gives the posterior prediction interval

$$P\{ Y(\boldsymbol{x}_0) \in \mu_{0|n}(\boldsymbol{x}_0) \pm \sigma_{0|n}(\boldsymbol{x}_0) z^{\alpha/2} | \boldsymbol{Y}^n \} = 1 - \alpha,$$

where $z^{\alpha/2}$ is the upper $\alpha/2$ critical point of the standard normal distribution (see Appendix A). As a special case, if the input x_0 is real with limits $a < x_0 < b$, then $\mu_{0|n}(x_0) \pm \sigma_{0|n}(x_0)z^{\alpha/2}$ are pointwise $100(1-\alpha)\%$ prediction bands for $Y(x_0)$, $a < x_0 < b$. Below, we illustrate the prediction band calculation following the statement of the predictive distribution for our second hierarchical $(Y_0, \boldsymbol{Y}^n)$ model in Theorem 4.2.

4.1.3 Predictive Distributions When R and r_0 Are Known

Using the fact that $[\boldsymbol{\beta}, \sigma_z^2] = [\boldsymbol{\beta} \,|\, \sigma_z^2] \times [\sigma_z^2]$, Theorem 4.2 provides the (predictive) distribution of $Y(\boldsymbol{x}_0)$ given $\boldsymbol{Y}^n$ for four priors corresponding to informative and non-informative choices for each of the terms $[\boldsymbol{\beta} \,|\, \sigma_z^2]$ and $[\sigma_z^2]$, i.e., proper and improper distributions, respectively. These four combinations give rise to the simplest $[\boldsymbol{\beta}, \sigma_z^2]$ priors that are, with adjustments given in Subsection 4.1.4, useful in practical situations. In all cases, the posterior is a location shifted and scaled univariate t distribution having degrees of freedom that are enhanced when there is informative prior information for either $\boldsymbol{\beta}$ or σ_z^2 (see Appendix B.2 for a definition of the non-central t distribution, $T_1(\nu, \mu, \sigma)$).

The informative conditional $[\boldsymbol{\beta} \,|\, \sigma_z^2]$ choice is the multivariate normal distribution with known mean $\boldsymbol{b}_0$ and known correlation matrix $\boldsymbol{V}_0$; lacking more definitive information, $\boldsymbol{V}_0$ is often taken to be diagonal, if not simply the identity matrix. This model makes strong assumptions, for example, it says that, componentwise, $\boldsymbol{\beta}$ is equally likely to be less than or greater than $\boldsymbol{b}_0$. The non-informative $\boldsymbol{\beta}$ prior is the intuitive choice

$$\pi(\boldsymbol{\beta}) = 1.$$

Our informative prior for σ_z^2 is the distribution of a constant divided by a Chi-square random variable, i.e., we model $[\sigma_z^2]$ as having the density of the $c_0/\chi^2_{\nu_0}$ random variable. This density has prior mean and variance

$$\frac{c_0}{\nu_0 - 2}, \text{ for } \nu_0 > 2 \text{ and } \frac{2 \times c_0^2}{(\nu_0 - 2)^2(\nu_0 - 4)}, \text{ for } \nu_0 > 4,$$

which allows one to more easily assign the model parameters. The non-informative prior used below is "Jeffreys prior"

$$\pi(\sigma_z^2) = \frac{1}{\sigma_z^2}$$

(see Jeffreys (1961), who gives arguments for this choice). Table 4.1 lists the notation for each of these four combinations that is used in Theorem 4.2.

Theorem 4.2. Suppose $(Y_0, \boldsymbol{Y}^n)$ follows a two-stage model in which the conditional distribution $[(Y_0, \boldsymbol{Y}^n) \mid (\boldsymbol{\beta}, \sigma_z^2)]$ is given by (4.1) and $[(\boldsymbol{\beta}, \sigma_z^2)]$ has one of the priors corresponding to the four products (1)–(4) stated in Table 4.1. Then

	$[\sigma_z^2]$	
$[\boldsymbol{\beta} \mid \sigma_z^2]$	$c_0/\chi^2_{\nu_0}$	$1/\sigma_z^2$
$N(\boldsymbol{b}_0, \sigma_z^2 \boldsymbol{V}_0)$	(1)	(2)
1	(3)	(4)

Table 4.1 Labels of four $[\boldsymbol{\beta}, \sigma_z^2]$ priors corresponding to informative and non-informative choices for each of $[\boldsymbol{\beta} \mid \sigma_z^2]$ and $[\sigma_z^2]$.

$$[Y_0 \mid \boldsymbol{Y}^n] \sim T_1\left(\nu_i, \mu_i, \sigma_i^2\right), \tag{4.1.14}$$

where

$$\nu_i = \begin{cases} n + \nu_0, & i = (1) \\ n, & i = (2) \\ n - p + \nu_0, & i = (3) \\ n - p, & i = (4), \end{cases}$$

$$\mu_i = \mu_i(\boldsymbol{x}_0) = \begin{cases} \boldsymbol{f}_0^\top \boldsymbol{\mu}_{\boldsymbol{\beta}|n} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^n - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{\beta}|n}), & i = (1) \text{ or } (2) \\ \boldsymbol{f}_0^\top \widehat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^n - \boldsymbol{F}\widehat{\boldsymbol{\beta}}), & i = (3) \text{ or } (4) \end{cases}$$

with $\boldsymbol{\mu}_{\boldsymbol{\beta}|n} = \left(\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F} + \boldsymbol{V}_0^{-1}\right)^{-1} \left(\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{y}^n + \boldsymbol{V}_0^{-1} \boldsymbol{b}_0\right)$,
$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \left(\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{y}^n\right)$, and

$$\sigma_i^2 = \sigma_i^2(\boldsymbol{x}_0) = \frac{Q_i^2}{\nu_i} \left\{ 1 - (\boldsymbol{f}_0^\top, \boldsymbol{r}_0^\top) \begin{bmatrix} \boldsymbol{V}_i & \boldsymbol{F}^\top \\ \boldsymbol{F} & \boldsymbol{R} \end{bmatrix}^{-1} \begin{pmatrix} \boldsymbol{f}_0 \\ \boldsymbol{r}_0 \end{pmatrix} \right\} \tag{4.1.15}$$

for $i = (1), \ldots, (4)$, where

$$\boldsymbol{V}_i = \begin{cases} -\boldsymbol{V}_0^{-1}, & i = (1) \text{ or } (2) \\ \boldsymbol{0}, & i = (3) \text{ or } (4) \end{cases}$$

and

$$Q_i^2 = \begin{cases} c_0 + Q_2^2, & i = (1) \\ Q_4^2 + \left(\boldsymbol{b}_0 - \widehat{\boldsymbol{\beta}}\right)^\top \left(\boldsymbol{V}_0 + [\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F}]^{-1}\right)^{-1} \left(\boldsymbol{b}_0 - \widehat{\boldsymbol{\beta}}\right), & i = (2) \\ c_0 + Q_4^2, & i = (3) \\ \boldsymbol{y}^{n\top} \left[\boldsymbol{R}^{-1} - \boldsymbol{R}^{-1} \boldsymbol{F} \left(\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1}\right] \boldsymbol{y}^n, & i = (4). \end{cases}$$

The formulas above for the degrees of freedom, location shift, and scale factor in the predictive t distribution all have very intuitive interpretations. The base value for the degrees of freedom ν_i is $n - p$, which is augmented by p additional degrees of freedom when the prior for $\boldsymbol{\beta}$ is informative (cases (1) and (2)), and ν_0 additional degrees of freedom when the prior for σ_z^2 is informative (cases (1) and (3)). For example, the degrees of freedom for case (4), with both components non-informative, is $n - p$ with no additions; the degrees of freedom for (1), with both components

informative, is $n + \nu_0 = (n - p) + p + \nu_0$, corresponding to two incremental prior sources.

The location shift μ_i is precisely the same centering value as in Theorem 4.1 for the case of known σ_z^2, either (4.1.4) or (4.1.8), depending on whether the informative or non-informative choice of prior is made for $[\boldsymbol{\beta} \mid \sigma_z^2]$, respectively. In particular, the non-informative prior for $\boldsymbol{\beta}$ gives the BLUP (3.3.4).

The scale factor $\sigma_i^2(\boldsymbol{x}_0)$ in (4.1.15) is an estimate of the scale factor $\sigma_{0|n}^2(\boldsymbol{x}_0)$ in (4.1.6) of Theorem 4.1. The term in braces multiplying σ_z^2 in (4.1.6) is the same as the term in braces in (4.1.15) after observing that $\tau^2 = \sigma_z^2$ in Table 4.1. The remaining term in (4.1.15), Q_i^2/ν_i, is an estimate of σ_z^2 in (4.1.6). The quadratic form, Q_i^2, pools information about σ_z^2 from the conditional distribution of $\boldsymbol{Y}^n$ given σ_z^2 with information from the prior of σ_z^2 (when the latter is available). The scale factor $\sigma_i^2(\boldsymbol{x}_0)$ is zero when $\boldsymbol{x}_0$ is any of the training data points.

Theorem 4.2 is used to place pointwise prediction bands about $y(\boldsymbol{x}_0)$ by using the fact that, given $\boldsymbol{Y}^n$,

$$\frac{Y(\boldsymbol{x}_0) - \mu_i(\boldsymbol{x}_0)}{\sigma_i(\boldsymbol{x}_0)} \sim T_1(\nu_i, 0, 1)\,.$$

This gives

$$P\{Y(\boldsymbol{x}_0) \in \mu_i(\boldsymbol{x}_0) \pm \sigma_i(\boldsymbol{x}_0) t_{\nu_i}^{\alpha/2} | \boldsymbol{Y}^n\} = 1 - \alpha, \tag{4.1.16}$$

where $t_\nu^{\alpha/2}$ is the upper $\alpha/2$ critical point of the T_ν distribution (see Appendix A). When x_0 is real, $\mu_i(x_0) \pm \sigma_i(x_0)\ t_{\nu_i}^{\alpha/2}$ for $a < x_0 < b$ are pointwise $100(1 - \alpha)\%$ prediction bands for $Y(x_0)$ at each $a < x_0 < b$.

Example 4.1. **(Continued)** Figure 4.2 plots the prediction bands corresponding to the BLUP when the predictive distribution is specified by the non-informative prior $[\boldsymbol{\beta}, \sigma_z^2] \propto 1/\sigma_z^2$ in Theorem 4.2. Notice that these bands have *zero width* at each of the true data points, as noted earlier. Prediction bands for any informative prior specification also have zero width at each of the true data points. ■

4.1.4 Prediction Distributions When Correlation Parameters Are Unknown

Subsections 4.1.2 and 4.1.3 assumed that the correlations among the observations are *known*, i.e., $\boldsymbol{R}$ and $\boldsymbol{r}_0$ are known. Now we assume that $y(\cdot)$ has a hierarchical Gaussian random field prior with parametric correlation function $R(\cdot \mid \boldsymbol{\psi})$ having *unknown* vector of parameters $\boldsymbol{\psi}$ (as introduced in Subsection 2.4.6 and previously considered in Subsection 3.3.2 for predictors). To facilitate the discussion below, suppose that the mean and variance of the normal predictive distribution in (4.1.3) and (4.1.8) are denoted by $\mu_{0|n}(\boldsymbol{x}_0) = \mu_{0|n}(\boldsymbol{x}_0|\boldsymbol{\psi})$ and $\sigma_{0|n}^2(\boldsymbol{x}_0) = \sigma_{0|n}^2(\boldsymbol{x}_0|\boldsymbol{\psi})$, where $\boldsymbol{\psi}$ was known in these earlier sections. Similarly, recall that the location and scale parameters of the predictive t distributions in (4.1.14) are denoted by $\mu_i(\boldsymbol{x}_0) = \mu_i(\boldsymbol{x}_0|\boldsymbol{\psi})$ and $\sigma_i^2(\boldsymbol{x}_0) = \sigma_i^2(\boldsymbol{x}_0|\boldsymbol{\psi})$, for $i \in \{(1), (2), (3), (4)\}$.

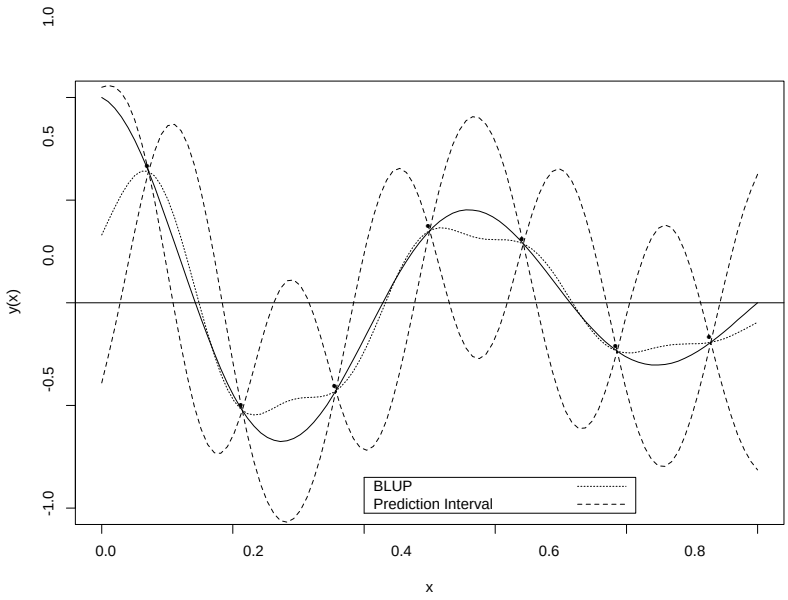


Fig. 4.2 The BLUP and corresponding pointwise 95% prediction interval limits for $y(x)$ based on the non-informative prior of Theorem 4.2.

We consider two issues. The first is the assessment of the standard error of the plug-in predictor $\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ of $Y_0(\boldsymbol{x}_0)$ that is derived from $\mu_{0|n}(\boldsymbol{x}_0|\boldsymbol{\psi})$ by substituting $\widehat{\boldsymbol{\psi}}$, which is an estimator of $\boldsymbol{\psi}$ that might be the MLE or REML. This question is implicitly stated from the frequentist viewpoint. The second issue is Bayesian; we describe the Bayesian approach to uncertainty in $\boldsymbol{\psi}$ which is to model it by a prior distribution.

When $\boldsymbol{\psi}$ is *known*, recall that $\sigma^2_{0|n}(\boldsymbol{x}_0|\boldsymbol{\psi})$ is the MSPE of $\mu_{0|n}(\boldsymbol{x}_0|\boldsymbol{\psi})$. This suggests estimating the MSPE of $\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ by the plug-in MSPE $\sigma^2_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$. The correct expression for the MSPE of $\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ is

$$\text{MSPE}(\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}}), \boldsymbol{\psi}) = E_{\boldsymbol{\psi}}\left\{\left(\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}}) - Y(\boldsymbol{x}_0)\right)^2\right\}. \tag{4.1.17}$$

Zimmerman and Cressie (1992) show that when the underlying surface is generated by a Gaussian random function,

$$\sigma^2_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}}) \leq \text{MSPE}(\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}}), \boldsymbol{\psi}) \tag{4.1.18}$$

under mild conditions so that $\sigma^2_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ *underestimates* the true variance of the plug-in predictor. The amount of the underestimate is most severe when the underlying Gaussian random function has weak correlation. Zimmerman and Cressie (1992) propose a correction to $\sigma^2_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ which provides a more nearly unbiased estimator

of MSPE($\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}}), \boldsymbol{\psi}$). Nevertheless, $\sigma^2_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ continues to be used for assessing the MSPE of $\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ because the amount by which it underestimates (4.1.17) has been shown to be asymptotically negligible for several models (for general linear models by Prasad and Rao (1990) and for time series models by Fuller and Hasza (1981)), and because of the lack of a compelling alternative that has demonstrably better small-sample properties.

An alternative viewpoint that accounts for uncertainty in $\boldsymbol{\psi}$ is to compute the mean squared prediction error based on the posterior distribution $[Y_0|\boldsymbol{Y}^n]$ (termed the "fully Bayesian approach" by some authors). We sketch how this is accomplished, at least in principle.

Assume that, in addition to $\boldsymbol{\beta}$ and σ^2_z, knowledge about $\boldsymbol{\psi}$ is summarized in a 2^{nd} stage $\boldsymbol{\psi}$ prior distribution. Often it will be reasonable to assume that the location and scale parameters, $\boldsymbol{\beta}$ and σ^2_z, respectively, are independent of the correlation information so that the prior for the ensemble $[\boldsymbol{\beta}, \sigma^2_z, \boldsymbol{\psi}]$ satisfies

$$[\boldsymbol{\beta}, \sigma^2_z, \boldsymbol{\psi}] = [\boldsymbol{\beta}, \sigma^2_z][\boldsymbol{\psi}].$$

For example, the non-informative prior of Theorem 4.2

$$[\boldsymbol{\beta}, \sigma^2_z] = \frac{1}{\sigma^2_z}$$

leads to the joint

$$[\boldsymbol{\beta}, \sigma^2_z, \boldsymbol{\psi}] = \frac{1}{\sigma^2_z}[\boldsymbol{\psi}].$$

In this case it is useful to regard the t posterior distributions that were stated in Theorem 4.2 as conditional on $\boldsymbol{\psi}$ and indicated by the notation $[Y_0|\boldsymbol{Y}^n, \boldsymbol{\psi}]$.

The required posterior distribution can be derived from

$$\begin{aligned}[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n] &= \int [Y(\boldsymbol{x}_0), \boldsymbol{\psi}|\boldsymbol{Y}^n]\, d\boldsymbol{\psi} \\ &= \int [Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n, \boldsymbol{\psi}]\, [\boldsymbol{\psi}|\boldsymbol{Y}^n]\, d\boldsymbol{\psi} \qquad (4.1.19)\end{aligned}$$

(see however the warning on page 74). The integration (4.1.19) can be prohibitive. For example, using the power exponential family with input-variable-specific scale and power parameters, the dimension of $\boldsymbol{\psi}$ is $2\times$ (number of inputs); $\boldsymbol{\psi}$ would be of dimension 12 for a six-dimensional input. Often, the posterior $[\boldsymbol{\psi}|\boldsymbol{Y}^n]$ can be obtained from

$$[\boldsymbol{\psi}|\boldsymbol{Y}^n] = \int \left[\boldsymbol{\beta}, \sigma^2_z, \boldsymbol{\psi}|\boldsymbol{Y}^n\right] d\boldsymbol{\beta}\; d\sigma^2_z, \qquad (4.1.20)$$

where the integrand in (4.1.20) is determined from

$$\left[\boldsymbol{\beta}, \sigma^2_z, \boldsymbol{\psi}|\boldsymbol{Y}^n\right] \propto \left[\boldsymbol{Y}^n|\boldsymbol{\beta}, \sigma^2_z, \boldsymbol{\psi}\right]\left[\boldsymbol{\beta}, \sigma^2_z, \boldsymbol{\psi}\right].$$

Equation (4.1.20) involves an integration of dimension equal to 1+ (the number of regressors), which is ordinarily less complicated than the integration (4.1.19) and can be carried out in closed form for "simple" priors.

In sum, one must both derive $[\boldsymbol{\psi}|\boldsymbol{Y}^n]$ *and* carry out the typically high dimensional integration (4.1.19) in order to compute the required posterior distribution. Once the posterior is available, the Bayesian alternatives to $\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ and $\sigma^2_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ are

$$E\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n\}$$

and

$$\mathrm{Var}\{Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n\}.$$

Both the predictor and the associated assessment of accuracy account for uncertainty in $\boldsymbol{\psi}$.

Handcock and Stein (1993) carried out the integrations in (4.1.19) for a specific two-input example using several regression models and isotropic correlation functions (power exponential and Matérn). As expected, they reported that for most cases that were studied, the Bayesian predictor and its standard errors gave wider confidence bands for the $Y(\boldsymbol{x}_0)$ than the plug-in predictors $\mu_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$ and $\sigma^2_{0|n}(\boldsymbol{x}_0|\widehat{\boldsymbol{\psi}})$. The plug-in predictor had particularly poor performance relative to the Bayes predictor when $\widehat{\boldsymbol{\psi}}$ was determined by an eye-fit to the variogram associated with the correlation function.

We assess the magnitude of the underestimate of the plug-in MSPE estimator, which is given on the left-hand side of (4.1.18), by calculating the achieved coverage of the pointwise prediction intervals (4.1.16) having a given nominal level. The simulation results below consider only the four top predictors from Section 3.3; these top performing predictors were the EBLUPs based on the power exponential and Matérn correlation functions using either REML or MLE to estimate the unknown correlation parameters. In addition, only training data corresponding to the LHD and Sobol´ designs were used because the D-optimal design (assuming the cubic model) tended to produce more biased predictions than the predictions based on training data using either the LHD or Sobol´ designs. For *each* of the 200 randomly selected surfaces on $[0, 1]^2$ that were described in the empirical study of Section 3.3, we computed the observed proportion of the 625 $\boldsymbol{x}_0$ points on $[0, 1]^2$ that were covered by the prediction interval (4.1.16) using $i = (4)$ (so that $\nu_i = n - 1$). This observed proportion was calculated for nominal 80%, 90%, 95%, and 99% prediction intervals. Figure 4.3 shows a trellis plot of a typical set of achieved coverages when $n = 16$, and the nominal coverage was 90%. Each box and whisker plot is based on the achieved coverages of the 50 randomly drawn surfaces from that combination of predictor, design, and surface.

The conclusions are as follows.

- Prediction intervals based on LHDs are slightly preferable to those based on Sobol´ designs, particularly for more irregular surfaces, i.e., surfaces with many local maxima and minima.

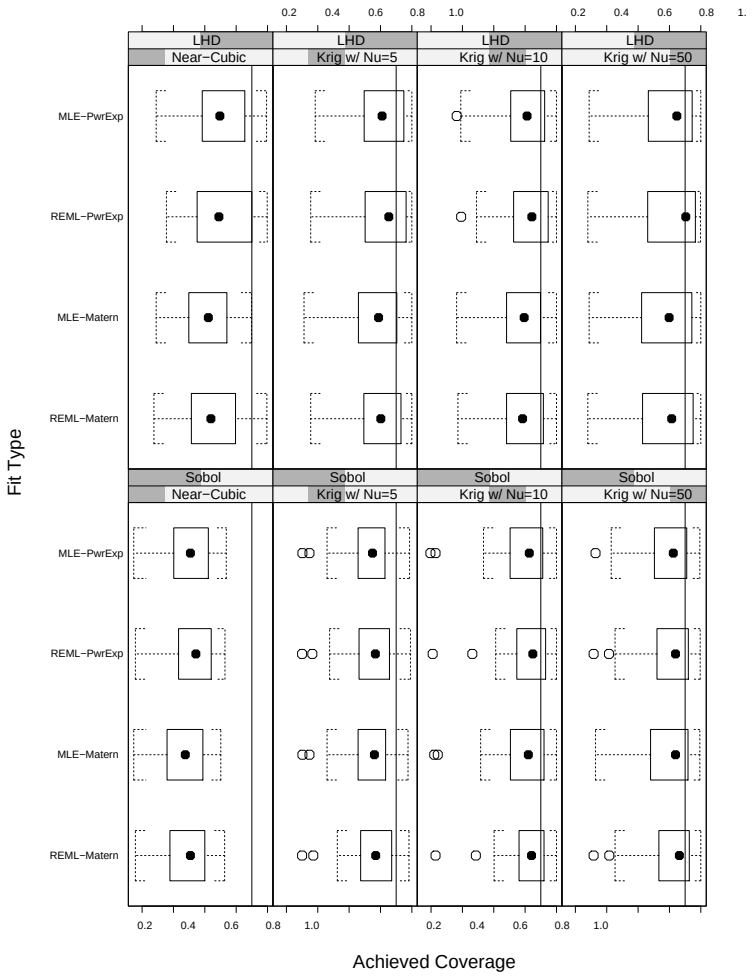


Fig. 4.3 Box and whisker plots of the fifty proportions of the 625 equispaced grid of points in $[0,1]^2$ that were covered by 90% nominal prediction intervals (4.1.16) classified by predictor, by experimental design, and type of surface.

- All four EBLUPs produced nearly equivalent coverages, for each combination of the experimental design and source of random surface.
- For krigifier surfaces, the shortfall in the median coverage is 10% to 15%.

Chapter 5
Space-Filling Designs for Computer Experiments

5.1 Introduction

In this chapter and the next, we discuss how to select inputs at which to compute the output of a computer experiment to achieve specific goals. The inputs we select constitute our "experimental design." We will sometimes refer to the inputs as "runs." The region corresponding to the values of the inputs over which we wish to study or model the response is the experimental region. A point in this region corresponds to a specific set of values of the inputs. Thus, an experimental design is a specification of points (runs) in the experimental region at which we wish to compute the response.

We begin by reviewing some of the basic principles of classical experimental design and then present an overview of some of the strategies that have been employed in computer experiments. For details concerning classical design see, for example, the books by Atkinson and Donev (1992), Box and Draper (1987), Dean and Voss (1999), Pukelsheim (1993), Silvey (1980), and Wu and Hamada (2000).

5.1.1 Some Basic Principles of Experimental Design

Suppose that we observe a response and wish to study how that response varies as we change a set of inputs. In physical experiments, there are a number of issues that make this problematic. First, the response may be affected by factors other than the inputs we have chosen to study. Unless we can completely control the effects of these additional factors, repeated observations at the same values of the inputs will vary as these additional factors vary. The effects of additional factors can either be unsystematic (random) or systematic. Unsystematic effects are usually referred to as random error, measurement error, or noise. Systematic effects are often referred to as bias. There are strategies for dealing with both noise and bias.

Replication and *blocking* are two techniques used to estimate and control the magnitude of random error. Replication (observing the response multiple times at the same set of inputs) allows one to directly estimate the magnitude and distribution of random error. Also, the sample means of replicated responses have smaller variances than the individual responses. Thus, the relation between these means and the inputs gives a clearer picture of the effects of the inputs because uncertainty from random error is reduced. In general, the more observations we have, the more information we have about the relation between the response and the inputs.

Blocking involves sorting experimental material into, or running the experiment in, relatively homogeneous sets called blocks. The corresponding analysis explores the relation between the response and the inputs within blocks, and then combines the results across blocks. Because of the homogeneity within a block, random error is less within a block than between blocks and the effects of the inputs more easily seen. There is an enormous body of literature on block designs, including both statistical and combinatorial issues. General discussions include John (1980), John (1987), Raghavarao (1971), or Street and Street (1987).

Bias is typically controlled by *randomization* and by exploring how the response changes as the inputs change. Randomization is accomplished by using a well-defined chance mechanism to assign the input values as well as any other factors that may affect the response and that are under the control of the experimenter, such as the order of experimentation, to experimental material. Factors assigned at random to experimental material will not systematically affect the response. By basing inferences on changes in the response as the input changes, bias effects "cancel," at least on average. For example, if a factor has the same effect on every response, subtraction (looking at changes or differences) removes the effect.

Replication, blocking, and randomization are basic principles of experimental design for controlling noise and bias. However, noise and bias are not the only problems that face experimenters. Another problem occurs when we are interested in studying the effects of several inputs simultaneously and the inputs themselves are highly correlated. This sometimes occurs in observational studies. If, for example, the observed values of two inputs are positively correlated so that they increase together simultaneously, then it is difficult to distinguish their effects on the response. Was it the increase in just one or some combination of both that produced the observed change in the response? This problem is sometimes referred to as collinearity. Orthogonal designs are used to overcome this problem. In an orthogonal design, the values of the inputs at which the response is observed are uncorrelated. An orthogonal design allows one to independently assess the effects of the different inputs. There is a large body of literature on finding orthogonal designs, generally in the context of factorial experiments. See, for example, Hedayat et al (1999).

Another problem that can be partly addressed (or at least detected) by careful choice of an experimental design, occurs when the assumptions we make about the nature of the relation between the response and the inputs (our statistical model) are incorrect. For example, suppose we assume that the relationship between the response and a single input is essentially linear when, in fact, it is highly nonlinear. Inferences based on the assumption that the relationship is linear will be incorrect.

It is important to be able to detect strong nonlinearities and we will need to observe the response with at least three different values of the input in order to do so. Error that arises because our assumed model is incorrect is sometimes referred to as model bias. Diagnostics, such as scatterplots and quantile plots, are used to detect model bias. The ability to detect model bias is improved by careful choice of an experimental design, for example, by observing the response at a wide variety of values of the inputs. We would like to select designs that will enable us to detect model inadequacies and lead to inferences that are relatively insensitive to model bias. This usually requires specifying both the model we intend to fit to our data as well as the form of an alternative model whose bias we wish to guard against; thus designs for model bias are selected to protect against certain types of bias. Box and Draper (1987) discuss this issue in more detail.

In addition to general principles, such as replication, blocking, randomization, orthogonality, and the ability to detect model bias, there exist very formal approaches to selecting an experimental design. The underlying principle is to consider the purpose of the experiment and the statistical model for the data and choose the design accordingly. If we can formulate the purpose of our experiment in terms of optimizing a particular quantity, we can then ask what inputs we should observe the response at to optimize this quantity. For example, if we are fitting a straight line to data, we might wish to select our design so as to give us the most precise (minimum variance) estimate of the slope. This approach to selection of an experimental design is often referred to as *optimal design*. See Atkinson and Donev (1992), Pukelsheim (1993), or Silvey (1980) for more on the theory of optimal design. In the context of the linear model, popular criteria involve minimizing some function of the covariance matrix of the least squares estimates of the parameters. Some common functions are the determinant of the covariance matrix (the generalized variance), the trace of the covariance matrix (the average variance), and the average of the variance of the predicted response over the experimental region. A design minimizing the first criterion is called *D-optimal*, a design minimizing the second is called *A-optimal*, and a design minimizing the third is called *I-optimal*. In many experiments, especially experiments with multiple objectives, it may not be clear how to formulate the experiment goal in terms of some quantity that can be optimized. Furthermore, even if we can formulate the problem in this way, finding the optimal design may be quite difficult.

In many experiments all the inputs at which we will observe the response are specified in advance. These are sometimes referred to as a single-stage or one-stage experimental designs. However, there are good reasons for running experiments in multiple stages. We agree with Box et al (1978) (page 303), who advocate the use of sequential or multi-stage designs.

"In exploring a functional relationship it might appear reasonable at first sight to adopt a comprehensive approach in which the entire range of every factor was investigated. The resulting design might contain all combinations of several levels of all factors. However, when runs can be made in successive groups, this is an inefficient way to organize experimental programs. The situation relates to the paradox

that the best time to design an experiment is after it is finished, the converse of which is that the worst time is at the beginning, when the least is known. If the entire experiment was designed at the outset, the following would have to be assumed known: (1) which variables were the most important, (2) over what ranges the variables should be studied, (3) in what metrics the variables and responses should be considered)e.g., linear, logarithmic, or reciprocal scales), and (4) what multivariable transformations should be made (perhaps the effects of variables x_1 and x_2 would be most simply expressed in terms of their ratio x_1/x_2 and their sum $x_1 + x_2$.

The experimenter is least able to answer such questions at the outset of an investigation but gradually becomes more able to do so as a program evolves.

All the above arguments point to the desirability of a sequence of moderately sized designs and reassessment of the results as each group of experiments becomes available."

One consideration in planning an experiment, which is sometimes overlooked, is whether to use a single-stage or a multi-stage design.

5.1.2 Design Strategies for Computer Experiments

Computer experiments, at least as we consider them here, differ from traditional physical experiments in that repeated observations at the same set of inputs yield identical responses. A single observation at a given set of inputs gives us perfect information about the response at that set of inputs, so replication is unnecessary. Uncertainty arises in computer experiments because we do not know the exact functional form of the relationship between the inputs and the response, although the response can be computed at any given input. Any functional models that we use to describe the relationship are only approximations. The discrepancy between the actual response produced by the computer code and the response we predict from the model we fit is our error. We referred to such error as model bias in the previous subsection.

Based on these observations, two principles for selecting designs in the types of computer experiments we consider are the following.

1. *Designs should not take more than one observation at any set of inputs. (But note that this principle assumes the computer code remains unchanged over time. When a design is run sequentially and the computer code is written and executed by a third party, it may be good policy to duplicate one of the design points in order to verify that the code has not been changed over the course of the experiment.)*
2. *Because we don't know the true relation between the response and inputs, designs should allow one to fit a variety of models and should provide information about all portions of the experimental region.*

If we believe that interesting features of the true model are just as likely to be in one part of the experimental region as another, if our goal is to be able to do prediction over the entire range of the inputs, and if we are running a single-stage experiment it is plausible to use designs that spread the points (inputs, runs) at which we observe the response evenly throughout the region. There are a number of ways to define what it means to spread points evenly throughout a region and these lead to various types of designs. We discuss a number of these in this chapter. Among the designs we will consider are designs based on selecting points in the experimental region by certain sampling methods; designs based on measures of distance between points that allow one to quantify how evenly spread out points are; designs based on measures of how close points are to being uniformly distributed throughout a region; and designs that are a hybrid of or variation on these designs. We will refer to all the designs in this chapter as *space-filling* or *exploratory* designs.

When runs of a computer experiment are expensive or time-consuming, and hence observing the response at a "large" number of inputs is not possible, what is a reasonable sample size that will allow us to fit the models described in Chapters 2–4? One rule of thumb suggested by Chapman et al (1994) and Jones et al (1998) is to use a sample size of $10d$ when the input space is of dimension d. However, because the "volume" of the design space increases as a power of d, $10d$ points becomes a very sparse sample as d increases. Obviously 10 points evenly spread over the unit interval are much more densely distributed than 100 points in the ten-dimensional unit cube. So is the $10d$ rule of thumb reasonable? Loeppky et al (2009) carefully investigate this issue and conclude that a sample size of $10d$ is a reasonable rule of thumb for an initial experiment when $d \leq 5$. When the response is sensitive to relatively few of the inputs, the rule is also reasonable for an initial experiment for d up to 20 or even larger. Loeppky et al (2009) also discuss diagnostics one can use to determine whether additional observations are needed (beyond those recommended by the $10d$ rule of thumb) and approximately how many might be needed to improve overall fit. They point out that one should always check the accuracy of the predictor fitted to the data and if it is poor, additional observations (perhaps many) may be needed.

The complexity of the input-output relationship has a direct bearing on the required sample size. Polynomial models provide some insight. The minimum number of points needed to uniquely determine a response surface of order m in d variables (all monomials of order m or less are included) is

$$\binom{m+d}{m}.$$

For a second-order response surface ($m = 2$), the $10d$ rule of thumb holds up to $d = 16$. For a third-order response surface, the $10d$ rule of thumb holds up to $d = 4$. For a fourth-order response surface, the $10d$ rule of thumb holds up to $d = 2$. Also, for an input-output relation such as $y = \sin(c\pi x), 0 \leq x \leq 1$ the $10d$ rule won't hold in one-dimension for large c, assuming one has no prior knowledge of the functional form of this relationship.

In practice does one encounter input-output relationships that produce very complicated response surfaces? Chen et al (2011) discuss a computer experiment concerning bistable laser diodes in which the two-dimensional response surface is quite rough over a portion of the design space and would require substantially more than 20 observations to accurately approximate.

Although not in the context of computer experiments, It is interesting to note that Box et al (1978) (page 304) recommend the following for multi-stage designs: "As a rough general rule, not more than one quarter of the experimental effort (budget) should be invested in a first design."

In practice we don't know the true model that describes the relation between the inputs and the response. However, if the models we fit to the data come from a sufficiently broad class, we may be willing to assume some model in this class is (to good approximation) "correct." In this case it is possible to formulate specific criteria for choosing a design and adopt an optimal design approach. Because the models considered in the previous chapters are remarkably flexible, this approach seems reasonable for these models. Thus, we discuss some criterion-based methods for selecting designs in Chapter **??**.

5.2 Designs Based on Methods for Selecting Random Samples

In the language of Section 1.3, the designs described in this section are used in cases when all inputs $\boldsymbol{x}$ are control variables as well as in cases when they are mixtures of control and environmental variables. However, most of these designs were originally motivated by their usefulness in applications where the inputs were all environmental variables; in this case we denote the inputs by $\boldsymbol{X}$ to emphasize their random nature. Let $y(\cdot)$ denote the output of the code. When the inputs are environmental variables, the most comprehensive objective would be to find the distribution of the random variable $Y = y(\boldsymbol{X})$ when $\boldsymbol{X}$ has a known distribution. If, as is often the case, this is deemed too difficult, the easier problem of determining some aspect of its distribution such as its mean $E\{Y\} = \mu$ or its variance is considered. Several of the designs introduced in this section, in particular the Latin hypercube design, were developed to solve the problem of estimating μ in such a setting. However, the reader should bear in mind that such designs are useful in more general input settings.

5.2.1 Designs Generated by Elementary Methods for Selecting Samples

Intuitively, we would like designs for computer experiments to be space-filling when prediction accuracy over the entire experimental region is of primary interest. The reason for this is that interpolators are used as predictors (e.g., the BLUP 3.3.4 or its Bayesian counterparts such as those that arise as the means of the predictive distributions derived in Section 3.3). Hence, the prediction error at any input site is a function of its location relative to the design points. Indeed, we saw, in Section **??**, that the prediction error is *zero* at each of the design points. For this reason, designs that are not space-filling, for example, designs that concentrate points on the boundary of the design space, can yield predictors that perform quite poorly in portions of the experimental region that are sparsely observed.

Deterministic strategies for selecting the values of the inputs at which to observe the response are to choose these values so they are spread evenly throughout or fill the experimental region. There are several methods that might be used to accomplish this, depending on what one means by "spreading points evenly" or "filling the experimental region."

A very simple strategy is to select points according to a regular grid pattern superimposed on the experimental region. For example, suppose the experimental region is the unit square $[0, 1] \times [0, 1]$. If we wish to observe the response at 25 evenly spaced points, we might consider the grid of points $\{0.1, 0.3, 0.5, 0.7, 0.9\} \times \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

There are several statistical strategies that one might adopt. One possibility is to select a simple random sample of points from the experimental region. In theory, there are infinitely many points between 0 and 1 and this makes selecting a simple random sample problematic. In practice, we only record numbers to a finite number of decimal places and thus, in practice, the number of points between 0 and 1 can be regarded as finite. Therefore, we can assume our experimental region consists of finitely many points and select a simple random sample of these.

Simple random sampling in computer experiments can be quite useful. If we sample the inputs according to some distribution (for example, a distribution describing how the values are distributed in a given population), we can get a sense of how the corresponding outputs are distributed and this can serve as the basis for inferences about the distribution of the output. However, for many purposes, other sampling schemes, such as stratified random sampling, are preferable to simple random sampling. Even if the goal is simply to guarantee that the inputs are evenly distributed over the experimental region, simple random sampling is not completely satisfactory, especially when the sample sizes are relatively small. With small samples in high-dimensional experimental regions, the sample will typically exhibit some clustering and fail to provide points in large portions of the region.

To improve the chances that inputs are spread "evenly" over the experimental region, we might use *stratified random sampling*. If we want a design consisting of n points, we would divide the experimental region into n strata, spread evenly

throughout the experimental region, and randomly select a single point from each. Varying the size and position of the strata, as well as sampling according to different distributions within the strata, allows considerable flexibility in selecting a design. This may be more or less useful, depending on the purpose of the computer experiment. For example, we may wish to explore some portions of the experimental region more thoroughly than others. However, if the goal is simply to select points that are spread evenly throughout the experimental region, spacing the strata evenly and sampling each according to a uniform distribution would seem the most natural choice.

If we expect the output to depend on only a few of the inputs (this is sometimes referred to as factor sparsity), then we might want to be sure that points are evenly spread across the projection of our experimental region onto these factors. A design that spreads points evenly throughout the full experimental region will not necessarily have this property. Alternatively, if we believe our model is well approximated by an additive model, a design that spreads points evenly across the range of each individual input (one-dimensional projection) might be desirable. For a sample of size n, it can be difficult to guarantee that a design has such projection properties, even with stratified sampling. Latin hypercube sampling, which we now discuss, is a way to generate designs that spread observations evenly over the range of each input separately.

5.2.2 Designs Generated by Latin Hypercube Sampling

Designs generated by Latin hypercube sampling are called Latin hypercube designs (LHD) throughout this book. We begin by introducing Latin hypercube (LH) sampling when the experimental region is the unit square $[0, 1]^2$. To obtain a design consisting of n points, divide each axis $[0, 1]$ into the n equally spaced intervals $[0, 1/n), \ldots, [(n-1)/n, 1]$. This partitions the unit square into n^2 cells of equal size. Now, fill these cells with the integers $1, 2, \ldots, n$ so as to form a Latin square, i.e., an arrangement in which each integer appears exactly once in each row and in each column of this grid of cells. Select one of the integers at random. In each of the n cells containing this integer, select a point at random. The resulting sample of n points are a LHD of size n (see Figure 5.2 for an example with $n = 5$). The method of choosing the sample ensures that points are spread evenly over the values of each input variable. Of course, such a LH sample could select points that are spread evenly along the diagonal of the square (see Figure 5.3). Although the points in such a sample have projections that are evenly spread out over the values of each input variable separately, we would not regard them as evenly spread out over the entire unit square.

We now describe a general procedure for obtaining a LH sample of size n from $\boldsymbol{X} = (X_1, \ldots, X_d)$ when $\boldsymbol{X}$ has independently distributed components. Stein (1987) discusses the implementation of LH sampling when $\boldsymbol{X}$ has dependent components, but we will not consider this case here.

In the independence case the idea is as follows. Suppose that a LH sample of size n is to be selected. The domain of each input variable is divided into n intervals. Each interval will be represented in the LH sample. The set of all possible Cartesian products of these intervals constitutes a partitioning of the d-dimensional sample space into n^d "cells." A set of n cells is chosen from the n^d population of cells in such a way that the projections of the centers of each of the cells onto each axis yield n distinct points on the axis; then a point is chosen at random in each selected cell.

In detail, we construct the LH sample as follows. For $k = 1, \ldots, d$, let $F_k(\cdot)$ denote the (marginal) distribution of X_k, the k^{th} component of $\boldsymbol{X}$ and, for simplicity, assume that X_k has support $[a_k, b_k]$. We divide the k^{th} axis into n parts, each of which has equal probability, $1/n$, under $F_k(\cdot)$. The division points for the k^{th} axis are

$$F_k^{-1}(\frac{1}{n}), \ldots, F_k^{-1}(\frac{n-1}{n}).$$

To choose n of the cells so created, let $\boldsymbol{\Pi} = (\Pi_{jk})$ be an $n \times d$ matrix having permutations of $\{1, 2, \ldots, n\}$ as columns which are randomly selected from the set of all possible permutations. Then the "upper-left hand" coordinates of the j^{th} cell in $\mathbb{R}^d$ are

$$F_k^{-1}(n^{-1}(\Pi_{jk} - 1)), \quad k = 1, \ldots, d,$$

with the convention $F_k^{-1}(0) = a_k$.

For $j = 1, \ldots, n$, let X_{jk}, $k = 1, \ldots, d$, denote the k^{th} component of the j^{th} vector, $\boldsymbol{X}_j$. Then we define the LH sample to have values

$$X_{jk} = F_k^{-1}(\frac{1}{n}(\Pi_{jk} - 1 + U_{jk})),$$

where the $\{U_{jk}\}$ are independent and identically distributed $U[0, 1]$ deviates, for $j = 1, \ldots, n$ and $k = 1, \ldots, d$. In sum, the j^{th} row of $\boldsymbol{\Pi}$ identifies the cell that $\boldsymbol{X}_j$ is sampled from, while the corresponding (independently generated) uniform deviates determine the location of $\boldsymbol{X}_j$ within the sampled cell.

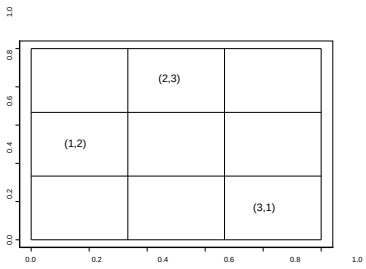


Fig. 5.1 Cells selected by the Latin hypercube sample (1,2), (2,3), and (3,1)

Example 5.1. Suppose $\boldsymbol{X} = (X_1, X_2)$ is uniformly distributed over $[0,1] \times [0,1]$ so that $F_k^{-1}(w) = w, 0 < w < 1$. To obtain a LH sample of size $n = 3$, we compute

$$X_{jk} = \frac{1}{3}\left(\Pi_{jk} - 1 + U_{jk}\right), \quad j = 1,2,3;\ k = 1,2.$$

The actual sample depends on the specific choice of $\boldsymbol{\Pi}$ and the $\{U_{jk}\}$.

To envision the pattern of the LH sample, divide the unit interval in each dimension into [0,1/3), [1/3,2/3), and [2/3,1], yielding a partition of $[0,1] \times [0,1]$ into nine squares (cells) of equal area. In the LH sample, each of these subintervals will be represented exactly once *in each dimension*. For simplicity of discussion, suppose we label these subintervals as 1, 2, and 3 in the order given above. One possible LHD would involve points randomly sampled from the (1,1), (2,3), and (3,2) squares and another possible design from the (1,2), (2,3), and (3,1) squares. Figure 5.1 plots the cells selected by the second design. These two selections correspond to the permutations

$$\boldsymbol{\Pi} = \begin{pmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 2 \end{pmatrix} \text{ and } \boldsymbol{\Pi} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 1 \end{pmatrix}. \tag{5.2.1}$$

Note that in each dimension, each subinterval appears exactly once. Because each subinterval is of length $1/3$, the addition of $U_{jk}/3$ to the left-hand boundary of the selected subinterval serves merely to pick a specific point in it. ■

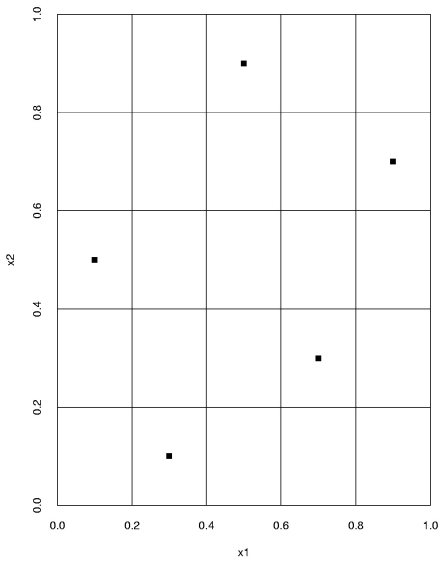
In the computer experiment setting, the input variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ are not regarded as random for purposes of experimental design. As in Example 5.1, suppose that each input variable has been scaled to have domain [0,1]. Denoting the k^{th} component of $\boldsymbol{x}_j$ by x_{jk} for $k = 1, \ldots, d$, suppose that we obtain a LHD from a given $\boldsymbol{\Pi}$ as follows:

$$x_{jk} = \frac{\Pi_{jk} - 0.5}{n}, \quad j = 1, \ldots, n;\ k = 1, \ldots, d\,.$$

This corresponds to taking $U_{jk} = 0.5$ for each $j = 1, \ldots, n$ and $k = 1, \ldots, d$ rather than as a sample from a $U[0,1]$ distribution. The "cells" are now identified with all d-dimensional Cartesian products of the intervals $\{(0, \frac{1}{n}], (\frac{1}{n}, \frac{2}{n}], \ldots, (1 - \frac{1}{n}, 1]\}$, and each $\boldsymbol{x}_j$ is sampled from the *center* of the cell indicated by the j^{th} row of $\boldsymbol{\Pi}$. An example of a LHD for $n = 5$ and $d = 2$ is given in Figure 5.2 with its associated $\boldsymbol{\Pi}$ matrix.

As mentioned previously, LHDs need not be space-filling over the full experimental region. To illustrate this point, consider the LHD for $n = 5$ and $d = 2$ that is shown in Figure 5.3, which one might *not* view as space-filling. One consequence of computing responses at this set of inputs is that we would expect a predictor fitted using this design to generally perform well only for $x_1 \approx x_2$. For example, consider the deterministic function

$$y(x_1, x_2) = \frac{x_1}{1 + x_2}, \quad \mathcal{X} = [0,1] \times [0,1]\,.$$



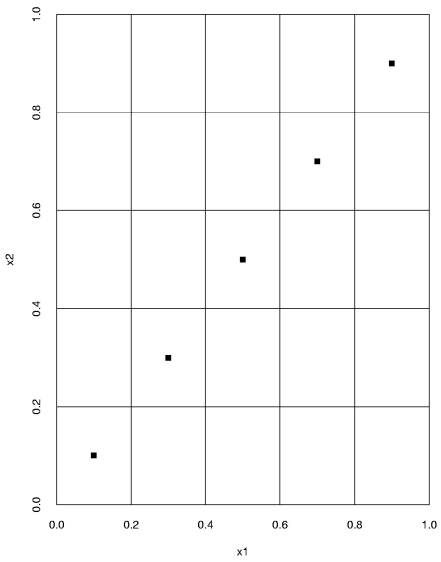
$$\boldsymbol{\Pi} = \begin{pmatrix} 3 & 5 \\ 4 & 2 \\ 2 & 1 \\ 1 & 3 \\ 5 & 4 \end{pmatrix}$$

Fig. 5.2 A space-filling Latin hypercube design and the corresponding permuation $\boldsymbol{\Pi}$.

The MLE-EBLUP was fitted to the observed responses using the training data for both of the designs shown in Figures 5.2 and 5.3 (Section 3.3). The predictor was based on the stochastic process

$$Y(x_1, x_2) = \beta_0 + Z(x_1, x_2),$$

where $Z(\cdot)$ is a zero mean Gaussian stochastic process with unknown process variance and product power exponential correlation function (2.4.7).



$$\boldsymbol{\Pi} = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \end{pmatrix}$$

Fig. 5.3 A non space-filling Latin hypercube design

The prediction error $|y(x_1, x_2) - \widehat{Y}(x_1, x_2)|$ was calculated on a grid of 100 equally-spaced (x_1, x_2) points for each design. Figure 5.4 plots a comparison of the prediction errors for the two designs where the symbol "1" ("0") indicates that the prediction error for the design of Figure 5.3 is larger (smaller) than the prediction error for the design of Figure 5.2. The space-filling design of Figure 5.2 clearly yields a better predictor over most of the design space except for the diagonal where the LHD in Figure 5.3 collects most of its data.

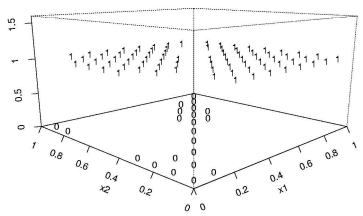


Fig. 5.4 Comparison of two LHDs. The plotting symbol "1" ("0") at location (x_1, x_2) means that the design in Figure 5.2 had lower (higher) mean squared prediction error than the design in Figure 5.3.

It is apparent from this discussion that although all LHDs possess desirable *marginal* properties, only a subset of these designs are truly "space-filling." Subsection 5.3.3 will discuss design criteria that Welch (1985) has successfully applied to select space-filling LHDs for use in computer experiments.

LHDs have been used extensively in the computer experiments literature; see, for example, Welch et al (1992) and Bernardo et al (1992). Other examples include Kennedy and O'Hagan (2001), Butler (2001), and Craig et al (2001). Because of their widespread use, it is worth examining in some detail the properties of LHDs in the setting, where all inputs are all environmental variables.

Designs based on LH sampling were introduced by McKay et al (1979) as a competitor to simple random sampling and stratified sampling when estimating the mean, variance, or distribution function of an output random variable. Stein (1987) and Owen (1992b) established additional large sample properties of LH sampling for estimating the mean $E\{Y\}$. We now look carefully at some of the results in these papers. This will provide greater insight into the actual properties of LHDs. We then reconsider their use in computer experiments.

5.2.3 Properties of Sampling-Based Designs

Suppose that a random vector of inputs $\boldsymbol{X} = (X_1, \ldots, X_d)$ to the computer output $y(\cdot)$ is distributed according to the known joint distribution $F(\cdot)$ over the experimental region $\mathcal{X} \equiv [a_i, b_i]^d \subset \mathbb{R}^d$. Based on a sample $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ from the distribution $F(\cdot)$, we are interested in estimating the mean of $g(Y)$ where $Y = y(\boldsymbol{X})$ and $g(\cdot)$ is a known function of the real-valued argument. This mean is

$$\mu = \mathrm{E}\{g(Y)\} = \int_{\mathcal{X}} g(y(\boldsymbol{x}))f(\boldsymbol{x})d\boldsymbol{x}.$$

We consider the properties of the naive moment estimator

$$T = T(y(\boldsymbol{X}_1), \ldots, y(\boldsymbol{X}_n)) = \frac{1}{n}\sum_{j=1}^{n} g(y(\boldsymbol{X}_j))$$

when $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ are either a simple random sample, a stratified random sample, or a Latin hypercube sample. In this derivation of the properties of T, we assume that the coordinates of $\boldsymbol{X}$ are independent, each with density $f(\cdot)$. Let

$$\sigma^2 = \text{Var}\{g(Y)\}.$$

For clarity denote the estimator T by T_R when simple random sampling is used and by T_L when LH sampling is used. McKay et al (1979) show the following.

Theorem 5.1. If $y(x_1, \ldots, x_d)$ is monotonic in each of its arguments, and $g(w)$ is a monotonic function of $w \in \mathbb{R}$, then $\text{Var}\{T_L\} \leq \text{Var}\{T_R\}$.

At this point a few cautions are in order. First, these results show only that for estimating the expected value of $g(Y)$ over the experimental region, designs based on proportional sampling are better than those based on simple random sampling, and, under certain conditions, LHDs are better than those based on simple random sampling. Designs based on LH sampling need not always be better than designs based on simple random sampling nor do we know if designs based on LH sampling are better than other types of designs, such as stratified sampling. Note, however, that the formulas derived in Section **??** and Section **??** do allow one to compare designs based on LH and stratified proportional sampling.

Second, in most computer experiments we do not know the relation between the output $y(\boldsymbol{x})$ and the component inputs $x_1, \ldots, x_d$. It is unlikely that we would be willing to assume this relationship is monotonic. And if we make such an assumption, the conditions on $g(\cdot)$ given in the above theorem imply that the extrema of $g(\cdot)$ are on the boundary of the experimental region. If, as is often the case, we are interested in finding the extrema of $g(\cdot)$ and we know the extrema are on the boundary of the experimental region, we would want to take observations near or on the boundary rather than using a LHD.

Third, the above properties are relevant if we are interested in estimating the expected value of $g(Y)$ over the experimental region. To illustrate, let $I\{E\}$ denote the indicator function as E (1 or 0, as E is true or false) and y_{fixed} be a given point in $\mathbb{R}$. Then setting $g(y) = y$ yields the mean of Y over the experimental region while setting $g(y) = I\{y \leq y_{fixed}\}$ produces the cumulative distribution function of Y at y_{fixed}. However, finding the expected value of $g(Y)$ over the experimental region is not usually the goal in computer experiments. More typically, our goal is to fit a model that approximates $g(\cdot)$ over the experimental region or to determine the points in the experimental region that are extrema of $g(\cdot)$. Thus, although LHDs are quite popular in computer experiments, the above results do not indicate whether they have good properties in many of the situations for which computer experiments are conducted. Better justification comes from the results we now describe.

Additional properties of sample means based on Latin hypercube samples have been established by Stein (1987) and Owen (1992b). For simplicity, we take $g(y) = y$ *for the remainder of this section* and use $\overline{Y} = \frac{1}{n}\sum_{j=1}^{n} y(\boldsymbol{X}_j)$ to estimate $\int_{\mathcal{X}} y(\boldsymbol{x}) dF(\boldsymbol{x})$. Let $F_i(\cdot)$ denote the marginal distribution of X_i, the i^{th} coordinate of $\boldsymbol{X}$. As above, we assume the coordinates of $\boldsymbol{X}$ are independent so

$$F(\boldsymbol{x}) = \prod_{i=1}^{d} F_i(x_i).$$

For $1 \le j \le d$, let $\boldsymbol{X}_{-j}$ denote $\boldsymbol{X}$ omitting X_j,

$$F_{-j}(\boldsymbol{x}_{-j}) = \prod_{i=1, i\neq j}^{d} F_i(x_i)$$

the distribution function of $\boldsymbol{X}_{-j}$, $\boldsymbol{x}_{-j}$ the corresponding argument extracted from $\boldsymbol{x}$, and $\mathcal{X}_{-j}$ denote the support of $F_{-j}(\cdot)$. Assuming $\int_{\mathcal{X}} y^2(\boldsymbol{x}) dF(\boldsymbol{x}) < \infty$, we decompose $y(\boldsymbol{x})$ as follows. Define

$$\mu = \int_{\mathcal{X}} y(\boldsymbol{x})\, dF(\boldsymbol{x}) \quad \text{and} \quad \alpha_j(x_j) = \int_{\mathcal{X}_{-j}} [y(\boldsymbol{x}) - \mu]\; dF_{-j}(\boldsymbol{x}_{-j}).$$

Then μ is the overall mean, the $\{\alpha_j(x_j)\}$ are the "main effect" functions corresponding to the coordinates of $\boldsymbol{x}$, and $r(\boldsymbol{x}) = y(\boldsymbol{x}) - \mu - \sum_{i=1}^{d} \alpha_i(x_i)$ is the residual (from additivity) of $y(\boldsymbol{x})$. These quantities are continuous analogs of an "analysis of variance" decomposition of $y(\boldsymbol{x})$. Further reason for this designation is the fact that

$$\int_{a_j}^{b_j} \alpha_j(x_j)\, dF_i(x_j) = 0 \quad \text{and} \quad \int_{\mathcal{X}_{-j}} r(\boldsymbol{x})\, dF_{-j}(\boldsymbol{x}_{-j}) = 0$$

for any x_j and all j.

Stein (1987) shows that for large samples, $\mathrm{Var}\left\{\overline{Y}\right\}$ is smaller under LH sampling than simple random sampling unless all main effect functions are 0. To be precise, Stein (1987) proves the following expansions for the variance of $\overline{Y}$ under the two sampling schemes.

Theorem 5.2. As $n \to \infty$, under Latin hypercube sampling and simple random sampling we have

$$\mathrm{Var}_{LHS}\left\{\overline{Y}\right\} = \frac{1}{n}\int_{\mathcal{X}} r^2(\boldsymbol{x}) dF(\boldsymbol{x}) + o(n^{-1}) \quad \text{and}$$

$$\mathrm{Var}_{SRS}\left\{\overline{Y}\right\} = \frac{1}{n}\int_{\mathcal{X}} r^2(\boldsymbol{x}) dF(\boldsymbol{x}) + \frac{1}{n}\sum_{i=1}^{d}\int_{a_i}^{b_i} \alpha_i^2(x_i) dF_i(x_i) + o(n^{-1}),$$

respectively.

The implication of this expansion is that, unless all $\alpha_j(\cdot)$ are identically 0, in the limit, LH sampling has a smaller variance than simple random sampling.

Further, not only can the variance of Y be estimated but also the normality of $\overline{Y}$ can be established. For simplicity, we assume $\mathcal{X} = [0,1]^d$ and that $F(\cdot)$ is uniform. More general cases can often be reduced to this setting by appropriate transformations. Owen (1992b) shows that $\overline{Y}$ computed from inputs based on LH sampling is

approximately normally distributed for large samples. This can be used as the basis for statistical inference about μ. Owen (1992b) proves the following.

Theorem 5.3. If $y(\boldsymbol{x})$ is bounded, then under LH sampling, $\sqrt{n}(\overline{Y} - \mu)$ tends in distribution to $N\left(0, \int_{\mathcal{X}} r^2(\boldsymbol{x}) d\boldsymbol{x}\right)$ as $n \to \infty$.

Owen (1992b) also provides estimators of the asymptotic variance

$$\int_{\mathcal{X}} r^2(\boldsymbol{x})\, d\boldsymbol{x}$$

to facilitate application of these results to computer experiments.

Section **??** of the Chapter Notes describes the use of LHDs in a generalization of these constant mean results to a regression setting, which has potential for use in computer experiments.

5.2.4 Extensions of Latin Hypercube Designs

There are several ways in which LHDs have been extended. Randomized orthogonal arrays are one such extension. An *orthogonal array O on s symbols of strength t* is an $n \times d$ ($p \geq t$) matrix whose entries are the s symbols arranged so that in every $n \times t$ submatrix of O, all of the s^t possible rows appear the same number λ of times; obviously $n = \lambda s^t$. We denote such an orthogonal array by $OA(n, d, \lambda, s, t)$. For additional discussion regarding orthogonal arrays see Raghavarao (1971) or Wu and Hamada (2000).

Owen (1992a) describes a procedure for generating n point space-filling designs in p dimensions from the columns of an $n \times p$ orthogonal array. The resulting designs are called *randomized orthogonal arrays*. If one plots the points of a randomized orthogonal array generated from an orthogonal array of strength t, in t or fewer of the coordinates, the result will be a regular grid. For details of concerning randomized orthogonal arrays, see Owen (1992a). Example 5.3 illustrates the method in 3 dimensions based on an orthogonal array of strength 2.

Example 5.2. A simple example of a randomized orthogonal array is the following. Suppose we take $n = 3$, $s = 3$, $t = 1$, and $\lambda = 1$. An orthogonal array on three symbols of strength $t = 1$ is the 3×2 matrix, both of whose columns are the integers 1, 2, 3.

$$\begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{pmatrix}$$

From this, a randomized orthogonal array is generated by following the procedure described in Example 5.1 used to generate the design displayed in Figure 5.1 (permute the second column of the above orthogonal array using the second permutation in (5.2.1)). The resulting design is a LHD and in this example, the projections into

each of the two dimensions (inputs) are uniform. Notice that, in general, an orthogonal array on s symbols of strength one with $n = s$ and $\lambda = 1$ is the $n \times p$ matrix, all of whose columns are the integers $1, 2, \ldots, s$. By following the procedure described in Subsection 5.2.2, one can generate a randomized orthogonal array which, in fact, is a LHD in d dimensions. ■

Example 5.3. Another example of a randomized orthogonal array is the following. Suppose we take $n = 9$, $s = 3$, $t = 2$, and $\lambda = 1$. An orthogonal array on three symbols of strength two is the 9×3 matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 3 \\ 3 & 2 & 1 \\ 3 & 3 & 2 \end{pmatrix}$$

To construct a randomized orthogonal array, we use this 9×3 matrix. Divide the unit cube $[0, 1] \times [0, 1] \times [0, 1]$ into a $3 \times 3 \times 3$ grid of 27 cells (cubes). Let (1,1,1) denote the cell (cube) $[0, \frac{1}{3}] \times [0, \frac{1}{3}] \times [0, \frac{1}{3}]$, (1,1,2) denote the cell $[0, \frac{1}{3}] \times [0, \frac{1}{3}] \times [\frac{1}{3}, \frac{2}{3}]$, (1,1,3) denote the cell $[0, \frac{1}{3}] \times [0, \frac{1}{3}] \times [\frac{2}{3}, 1]$, ..., and (3,3,3) the cell $[\frac{2}{3}, 1] \times [\frac{2}{3}, 1] \times [\frac{2}{3}, 1]$. Each row of the above 9×3 matrix correponds to one of these 27 cells. The point in the center of the nine cells determined by the rows of the matrix yields a nine point randomized orthogonal array. Projected onto and two-dimensional subspace, the design looks like a regular 3×3 grid. Instead of selecting the points in the centers of the nine cells, one could select a point a random from each of these cells. The resulting projections onto two-dimensional subspaces would not be a regular grid, but would be evenly spaced in each of the two-dimensional subspaces. ■

Although randomized orthogonal arrays extend the projection properties of LHDs to more than one dimension, they have the drawback that they only exist for certain values of n, namely for $n = \lambda s^t$, and only for certain values of d. Also, because $n = \lambda s^t$, only for relatively small values of s and t will the designs be practical for use in computer experiments in which individual observations are time-consuming to obtain and hence for which n must be small. See Tang (1993) and Tang (1994) for additional information about the use of randomized orthogonal arrays in computer experiments.

Cascading LHDs are another extension of LHDs. Cascading LHDs are introduced in Handcock (1991) and can be described as follows. Generate a LHD. At each point of this design, consider a small region around the point. In this small region, generate a second LHD. The result is a cluster of small LHDs and is called a *cascading Latin hypercube design*. Such designs allow one to explore both the local (in small subregions) and the global (over the entire experimental region) behavior of the response.

Suppose one uses a LHD consisting of n points in $\mathbb{R}^d$. After fitting a predictor to the data, suppose one decides the fit is inadequate and m additional runs of the computer simulator are necessary. Is it possible to select the m runs in such a way that the resulting set of $m+n$ runs is a LHD? In general, the answer is no. Figure 5.5 displays a 2-point LHD in two dimensions with the two points randomly placed in two of the four cells. This cannot be extended to a 3-point LHD in two dimensions, because both points are in the same cell when the design space is partitioned into nine cells (outlined by the dashed lines). However, the 2-point LHD could be extended to a 4-point LHD in two dimensions because the two points would now be in two separate cells when the design space is partitioned into 16 cells.

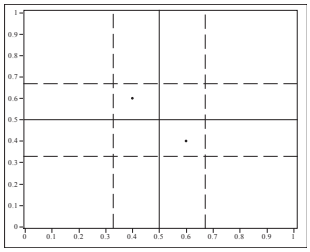


Fig. 5.5 A 2-point LHD that cannot be extended to a 3-point LHD. Points are placed at random in the four cells for a 2-point LHD. The cells are outlined by the solid lines. The dashed lines outline the nine cells for a 3-point LHD. Notice both points are in the same cell.

Notice that if in the original LHD the points were chosen at random in the n cells, and if $m = an$ for some positive integer a, it is possible to add the m points in such a way the $m + n$ points are a LHD. In the initial LHD, the domain of each input variable was subdivided into n intervals. Subdivide each of these n intervals into $a + 1$ intervals so that now the domain of each input variable is subdivided into $(a + 1)n$ intervals. The Cartesian product of these intervals constitutes a partitioning of the d-dimensional sample space into $[(a + 1)n]^d$ cells. Each of the n points in the original design are in exactly one of these cells. Choose a subset of $(a + 1)n$ cells in such a way that they include the n cells containing points from the original design and so that the projections of the centers of all $(a + 1)n$ points onto each component axis yield $(a+1)n$ distinct points on the axis. In terms of the method described before Example 5.1, this will mean that one can select only certain $(a + 1)n \times d$ matrices $\boldsymbol{\Pi}$ having permutations of $\{1, 2, \ldots, (a + 1)n\}$ as columns. Notice that if in the original LHD the points were chosen at the center of the n cells, it is still possible to add m points in such a way that the resulting design is a LHD with points at the center of cells, provided a is even.

5.3 Latin Hypercube Designs Satisfying Additional Criteria

Figure 5.3 displays an LHD that would probably not be considered space-filling because the points lie along a straight line and are perfectly correlated. By comparison, the LHD in Figure 5.2 appears more space-filling and the points appear much less correlated. ls it possible to identify special types of LHDs that have additional desirable properties?

5.3.1 Orthogonal Array-Based Latin Hypercube Designs

A possible strategy for avoiding LHDs that do not appear to be space-filling is to select those for which the points are uncorrelated. Owen (1992a) and Tang (1993) discuss methods for constructing an LHD with an underlying orthogonal array structure, thus assuring that in some dimension the points in the LHD appear uncorrelated. Tang (1993) uses an orthogonal array to specify a restricted permutation of $\{1, 2, \ldots, n\}$ in selecting the matrix $\boldsymbol{\Pi}$ from which the LHD is formed. Let $OA(n, d, 1, s, t)$ be an $n \times d$ orthogonal array on s symbols of strength t with $\lambda = 1$. For the kth column of $OA(n, d, 1, s, t)$ let $r_{k,l+1}$ be the number of runs with entry l. Note that $r_{k,l_1+1} = r_{k,l_2+1} = r$ for all l_1 and l_2 so that $rs = n$. We can form a $\boldsymbol{\Pi}$ based on $OA(n, d, 1, s, t)$ by selecting each column of $OA(n, d, 1, s, t)$ and replacing the r entries of $OA(n, d, 1, s, t)$ with level l by a permutation of the integers $rl + 0, rl + 1, \ldots, rl + (r - 1)$ for all $l = 1, 2, \ldots, s$. The LHD formed from this $\boldsymbol{\Pi}$ will have all univariate projections uniformly distributed and all t-variate projections uniformly distributed. Tang (1993) refers to an LHD constructed in this way as an OA-based LHD.

Another way to think about an OA-based LHD is that the structure of the orthogonal array is used to restrict the placement of the points within the unit hypercube (assume for this discussion that we are interested in LHDs on the d-dimensional unit hypercube). In the context of the previous discussion, for the kth column of $OA(n, d, 1, s, t)$ we consider the non-overlapping division of $[0, 1]$ into s equal length intervals of the form $[0, \frac{1}{s}) \cup [\frac{1}{s}, \frac{2}{s}) \cup \cdots \cup [\frac{s-1}{s}, 1]$. Because $OA(n, d, 1, s, t)$ is an orthogonal array, each of the s symbols appears equally often in each column and we let r denote the number of times each symbol appears in a given column. For a given level $l_j = 0, 1, \ldots, s - 1$ we define the non-overlapping division of the interval $[\frac{l_j}{s}, \frac{l_j+1}{s})$ into r subintervals of the form

$$[\frac{l_j}{s} + \frac{i}{sr}, \frac{l_j}{s} + \frac{i+1}{sr}), i = 0, 1, \ldots, r - 1.$$

For column k let $p_{k_1}, p_{k_2}, \ldots, p_{k_r}$ be a random permutation of the integers $0, 1, \ldots, r-1$. Then the r points corresponding to level l_j are randomly (or systematically) placed one each in the Cartesian product intervals

$$[\frac{l_j, s}{+} \frac{p_{k_i}, sr}{,} \frac{l_j, s}{+} \frac{p_{k_i} + 1, sr}{)}, k = 1, 2, \ldots, d.$$

Notice for each column of $OA(n, d, 1, s, t)$, $n = rs$ and the Latin hypercube intervals $[\frac{i,n}{.} \frac{i+1,n}{)}$ are identical to the substratification described so that the resulting array, with placement of points imposed by the strength t orthogonal array is indeed an LHD with t-dimensional projection properties consistent with $OA(n, d, 1, s, t)$.

Example 5.4. Suppose we start with the $OA(4, 2, 1, 2, 2)$

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{pmatrix}$$

To obtain $\boldsymbol{\Pi}$, in each column we replace the symbol 1 with a random permutation of the integers 1, 2, and the symbol 2 with a random permutation of the integers 3,4 One possibility is

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 4 & 3 \\ 3 & 4 \end{pmatrix}$$

■

Example 5.5. Suppose we start with the $OA(9, 3, 1, 3, 2$

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 3 \\ 3 & 2 & 1 \\ 3 & 3 & 2 \end{pmatrix}$$

To obtain $\boldsymbol{\Pi}$, in each column we replace the symbol 1 with a random permutation of the integers 1, 2, 3, the symbol 2 with a random permutation of the integers 4, 5, 6, and the symbol 3 by a random permutation of the integers 7, 8, 9. One possibility is

$$\begin{pmatrix} 1 & 3 & 2 \\ 3 & 4 & 5 \\ 2 & 8 & 7 \\ 6 & 2 & 4 \\ 5 & 5 & 9 \\ 4 & 7 & 1 \\ 9 & 1 & 8 \\ 7 & 6 & 3 \\ 8 & 9 & 6 \end{pmatrix}$$

■

Note that in the step for constructing $\boldsymbol{\Pi}$ from the initial orthogonal array, many choices for the permutations are possible, hence from a given initial orthogonal array, many $\boldsymbol{\Pi}$ can be constructed. One can impose an additional criterion to select one of these $\boldsymbol{\Pi}$, thus insuring that the final LHD has an additional desirable property.

The orthogonal array structure imposed on the design is appealing in that it leads to uniformity in all t-variate projections when the strength of the orthogonal array is t. This helps achieve an additional degree of space-fillingness not easily achieved by a random LHD or one that is numerically optimized according to some criterion.

An important drawback is that OA-based LHDs are limited in the run sizes that are possible. For example, even starting with an orthogonal array on 2 symbols of strength 2 the run size must be a multiple of 4. In addition, the method of construction is not always readily adaptable to algorithmic generation. For these reasons Loeppky et al (2012) introduce a more flexible class of designs, called projection array based designs, that have space-filling properties analogous to OA-based LHDs, but exist for all run sizes.

5.3.2 Orthogonal Latin Hypercube Designs

Another attempt to find LHDs that have additional good properties is due to Ye (1998). He discusses a method for constructing LHDs for which all columns are orthogonal to each other. In general, the columns of an $n \times d$ LHD are formed from an $n \times d$ matrix $\boldsymbol{\Pi}$ whose columns are permutations of the integers $\{1, 2, \ldots, n\}$, as discussed in Subsection 5.2.2. By restricting to only certain permutations of $\{1, 2, \ldots, n\}$, Ye (1998) is able to generate LHDs with orthogonal columns, which he calls orthogonal latin hypercubes (OLHs). The method of construction, in particular the set of permutations needed to generate orthogonal columns is as follows.

Let $\mathbf{e}$ be the $2^{m-1} \times 1$ column vector with entries $\{1, 2, \ldots, 2^{m-1}\}$. Define the permutations

$$A_k = \prod_{j=1}^{2^{m-k-1}} \left\{ \prod_{i=1}^{2^{k-1}} ((j-1)2^k + i \quad j2^k + 1 - i) \right\}, k = 1, \ldots, m-1 \tag{5.3.1}$$

where $\prod$ represents the composition of permutations and $(r \quad s)$ represents the transposition of rows r and s. An quivalent dfinition of thse permutations expressed as permutation matrices is

$$\mathbf{A_k} = \{\bigotimes_{\mathbf{j=1}}^{\mathbf{m-1-k}} \mathbf{I}\} \bigotimes \{\bigotimes_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{R}\} \tag{5.3.2}$$

where $\mathbf{I}$ is the 2×2 identity matrix,

$$\mathbf{R} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and $\bigotimes$ is the Kronecker product.

Next, let $\mathbf{M}$ be the $2^{m-1} \times (2m-2)$ matrix with columns

$$\{\mathbf{e}, \mathbf{A}_1\mathbf{e}, \ldots, \mathbf{A}_{m-1}\mathbf{e}, \mathbf{A}_{m-1}\mathbf{A}_1\mathbf{e}, \ldots, \mathbf{A}_{m-1}\mathbf{A}_{m-2}\mathbf{e}\}.$$

For $k = 1, \ldots m-1$ define

$$\mathbf{a}_k = \{\bigotimes_{j=1}^{m-1} \mathbf{B}_j\}$$

where

$$\mathbf{B}_{m-k} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \mathbf{B}_i = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

for $i \neq m-k$. Also denote the $2^{m-1} \times 1$ vector of 1s by $\mathbf{1}$.

Now define $\mathbf{S}$ to be the $2^{m-1} \times (2m-2)$ matrix with columns

$$\{\mathbf{1}, \mathbf{a}_1, \ldots, \mathbf{a}_{m-1}, \mathbf{a}_1 \times \mathbf{a}_2, \ldots, \mathbf{a}_1 \times \mathbf{a}_{m-1}\}.$$

where here $\times$ is the element wise product. Let $\mathbf{T} = \mathbf{M} \times \mathbf{S}$. Finally, consider the $2^m + 1 \times (2m-2)$ matrix $\mathbf{O}$ whose first 2^{m-1} rows are $\mathbf{T}$, whose next row consists of all 0s, and whose last 2^{m-1} rows are the "mirror image" of $\mathbf{T}$, namely the rows of $-\mathbf{T}$ in reverse order. From $\mathbf{O}$ remove the row consisting of all 0s and rescale levels be equidistant. Let $\mathbf{O}^*$ denote the resulting $2^m \times (2m-2)$ matrix.

Ye (1998) shows that the columns of $\mathbf{O}$ are orthogonal to each other, the elementwise square of each column of $\mathbf{O}$ is orthogonal to all the columns of $\mathbf{O}$, and that the elementwise product of every two columns of $\mathbf{O}$ is orthogonal to all columns in $\mathbf{O}$. In other words, if $\mathbf{O}$ is used as the design matrix for a second-order response surface, all estimates of linear, bilinear, and quadratic effects are uncorrelated with the estimates of linear effects. The same holds true for $\mathbf{O}^*$.

Example 5.6. Consider the case where $m = 3$. Then

$$A_1 = \prod_{j=1}^{2}((j-1)2+1 \quad j2)$$
$$= (1 \quad 2)(3 \quad 4)$$

$$A_2 = \prod_{i=1}^{2}(i \quad 4+1-i)$$
$$= (1 \quad 4)(2 \quad 3)$$

and so

$$\mathbf{e} = (1,2,3,4)^\top$$

$$\mathbf{A}_1\mathbf{e} = (2,1,4,3)^\top$$

$$\mathbf{A}_2\mathbf{e} = (4,3,2,1)^\top$$

$$\mathbf{A}_2\mathbf{A}_1\mathbf{e} = (3,4,1,2)^\top$$

and

$$\mathbf{M} = \begin{pmatrix} 1 & 2 & 4 & 3 \\ 2 & 1 & 3 & 4 \\ 3 & 4 & 2 & 1 \\ 4 & 3 & 1 & 2 \end{pmatrix}.$$

Next, notice

$$\mathbf{a}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \bigotimes \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$
$$= \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$

$$\mathbf{a}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \bigotimes \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$= \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

so that

$$\mathbf{S} = \{\mathbf{1}, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_1 \times \mathbf{a}_2\} = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Then

$$\mathbf{T} = \mathbf{M} \times \mathbf{S} = \begin{pmatrix} 1 & -2 & -4 & 3 \\ 2 & 1 & -3 & -4 \\ 3 & -4 & 2 & -1 \\ 4 & 3 & 1 & 2 \end{pmatrix}.$$

Hence,

$$\mathbf{O} = \begin{pmatrix} 1 & -2 & -4 & 3 \\ 2 & 1 & -3 & -4 \\ 3 & -4 & 2 & -1 \\ 4 & 3 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ -4 & -3 & -1 & -2 \\ -3 & 4 & -2 & 1 \\ -2 & -1 & 3 & 4 \\ -1 & 2 & 4 & -3 \end{pmatrix}$$

and

$$\mathbf{O}^* = \begin{pmatrix} 0.5 & -1.5 & -3.5 & 2.5 \\ 1.5 & 0.5 & -2.5 & -3.5 \\ 2.5 & -3.5 & 1.5 & -0.5 \\ 3.5 & 2.5 & 0.5 & 1.5 \\ -3.5 & -2.5 & -0.5 & -1.5 \\ -2.5 & -3.5 & -1.5 & 0.5 \\ -1.5 & -0.5 & 2.5 & 3.5 \\ -0.5 & 1.5 & 3.5 & -2.5 \end{pmatrix}$$

■

Ye (1998) also shows that the construction described above can be modified to yield a class of OLHs. First, one can replace **e** by any of its permutations. Second, one can reverse any of the signs of any subset of columns of **O** or $\mathbf{O}^*$. The resulting arrays are all OLHs in the sense of having all the properties mentioned prior to Example 5.6.

5.3.3 Symmetric Latin Hypercube Designs

Unfortunately, OLHs exist only for very limited values of n, namely $n = 2^m$ or $n = 2^m + 1, m \geq 2$. Ye et al (2000) introduce a more general class of LHDs, called symmetric LHDs, to overcome this limitation. An LHD is called a symmetric LHD (SLHD) if it has the following property: in an $n \times d$ LHD with levels $1, 2, \ldots, n$, if $(a_1, a_2, \ldots, a_d)$ is one of the rows, then $(n + 1 - a_1, n + 1 - a_2, \ldots, n + 1 - a_d)$ must

be another row. Ye et al (2000) do not discuss the construction of SLHDs, but when n is even one obtains an SHLD as follows. The first row can be any $1 \times d$ vector $(a_{11}, a_{12}, \ldots, a_{1d})$ where the $(a_{1j}$ are elements of $\{1, 2, \ldots, n\}$. The second row is $(n + 1 - a_{11}, n + 1 - a_{12}, \ldots, n + 1 - a_{1d})$. The third row can be any $1 \times d$ vector $(a_{31}, a_{32}, \ldots, a_{3d})$ where a_{3j} can be any of the integers $1, 2, \ldots, n$ that is not equal to either a_{1j} or $n + 1 - a_{1j}$. The fourth row is $(n + 1 - a_{31}, n + 1 - a_{32}, \ldots, n + 1 - a_{3d})$. Continue on in this manner, adding the odd rows so that the entries in column j have not year appeared in the previous rows of the column. The even rows have entries $n + 1$ minus the entry in the previous row. When n is odd let the first row be $(\frac{n+1}{2}, \frac{n+1}{2}, \ldots, \frac{n+1}{2})$. The second row can be any $1 \times d$ vector $(a_{21}, a_{22}, \ldots, a_{2d})$ where the a_{2j} are elements of $\{1, 2, \ldots, n\}$ except $\frac{n+1}{2}$. The third row is $(n + 1 - a_{21}, n + 1 - a_{22}, \ldots, n + 1 - a_{2d})$. The fourth row can be any $1 \times d$ vector $(a_{41}, a_{42}, \ldots, a_{4d})$ where a_{4j} can be any of the integers $1, 2, \ldots, n$ that is not equal to $\frac{n+1}{2}$, a_{2j} or $n + 1 - a_{3j}$. Continue on in this manner, adding the even rows so that the entries in column j have not year appeared in the previous rows of the column. The odd rows have entries $n + 1$ minus the entry in the previous row.

Note that the non space-filling LHD in Figure 5.3 is an SLHD, so SLHDs need not be "good" LHDs.

Example 5.7. To construct an SLHD with $n = 10$ and $d = 3$, suppose we begin with row $(1, 6, 6)$. Following the algorithm described previously, we might obtain the following SLHD.

$$\begin{pmatrix} 1 & 6 & 6 \\ 10 & 5 & 5 \\ 2 & 2 & 3 \\ 9 & 9 & 8 \\ 3 & 1 & 9 \\ 8 & 10 & 2 \\ 4 & 3 & 4 \\ 7 & 8 & 7 \\ 5 & 7 & 1 \\ 6 & 4 & 10 \end{pmatrix}$$

To construct an SLHD with $n = 9$ and $d = 3$, suppose we begin with rows $(5, 5, 5)$ and $(1, 6, 6)$. Following the algorithm described previously, we might obtain the following SLHD.

$$\begin{pmatrix} 5 & 5 & 5 \\ 1 & 6 & 6 \\ 9 & 4 & 4 \\ 2 & 2 & 3 \\ 8 & 8 & 7 \\ 3 & 1 & 9 \\ 7 & 9 & 1 \\ 4 & 3 & 8 \\ 6 & 7 & 2 \end{pmatrix}$$

■

Ye et al (2000) point out that SLHDs have some orthogonality properties. In a polynomial response surface, least squares estimation of the linear effect of each variable is uncorrelated with all quadratic effects and bi-linear interactions (but not necessarily with the linear effects of other variables). This follows from results in Ye (1998) because OLHs have the same symmetry properties as SLHDs but also possess additional orthogonality that guarantees that linear effects are uncorrelated.

SLHDs form a subclass of all LHDs. As we discuss later, one can apply additional criteria to select a particular design from the class of all $n \times d$ LHDs, from the class of all $n \times d$ OLHs, or from the class of all $n \times d$ SLHDs. For the latter, Ye et al (2000) propose a column-wise exchange algorithm that replaces an SLHD with another SLHD, allowing one to search the class of $n \times d$ SLHDs for a design that optimizes some additional property of the design.

The orthogonality properties of OLHs and SLHDs are useful if one plans to fit second order or higher response surface models to the data using standard least squares. However, if one intends to fit a predictor, such as the EBLUP discussed in Chapter 3, in which the generalized least squares estimate of the regression parameters is used, the benefits of orthogonality are less clear.

5.4 Designs Based on Measures of Distance

In this subsection, we consider criteria for selecting a design that are based on a measure or metric that quantifies the spread of a set of points. For all distance-based criteria discussed below, the domain of each input is normalized to the interval [0,1] otherwise inputs with larger ranges can dominate the computation of a maximin design, say. If the input space in the original problem is rectangular

$$\prod_{\ell=1}^{d} [a_\ell, b_\ell]$$

then

$$x_\ell = \frac{x_\ell - a_\ell}{b_\ell - a_\ell}, \quad \ell = 1, \ldots, d$$

is used to scale and shift the input space to $[0, 1]^d$; the inverse transform is used to place the computed design on the scale of the original design problem.

The first method considered in this section to measure the spread of n points in a design is by the distance of the closest *no two* points in the design. To simultaneously define distances for both rectangular and non-rectangular input regions $\mathcal{X}$ (Section 5.8); let ρ denote an arbitrary metric on $\mathcal{X}$. Let $\mathcal{D}$ be an n-point design consisting of *distinct* input sites $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ with $\boldsymbol{x}_\ell \in \mathcal{X}$, $\ell = 1, \ldots, n$. For example, one important distance measure is p^{th} order distance between $\boldsymbol{w}, \boldsymbol{x} \in \mathcal{X}$ for $p \geq 1$, which is defined by

$$\rho_p(\boldsymbol{w}, \boldsymbol{x}) = \left[\sum_{j=1}^{d} |w_j - x_j|^p \right]^{1/p}. \tag{5.4.1}$$

Rectangular ("Manhattan") and Euclidean distances are the cases $p = 1$ and $p = 2$, respectively. Then one way to measure of the closeness of the n points in $\mathcal{D}$ is the smallest distance between any two points in $\mathcal{D}$, i.e.,

$$\min_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{D}} \rho_p(\boldsymbol{x}_1, \boldsymbol{x}_2). \tag{5.4.2}$$

A design that maximizes (5.4.2) is said to be a *maximin distance design* and is denoted by $\mathcal{D}_{Mm}$; thus

$$\min_{x_1, x_2 \in \mathcal{D}_{Mm}} \rho_p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \max_{\mathcal{D} \subset \mathcal{X}} \min_{x_1, x_2 \in \mathcal{D}} \rho_p(\boldsymbol{x}_1, \boldsymbol{x}_2). \tag{5.4.3}$$

In an intuitive sense, therefore, $\mathcal{D}_{Mm}$ designs guarantee that no two points in the design are too close, and hence the design points are spread over $\mathcal{X}$.

One criticism of the maximin principle is that it judges the goodness of a design by the minimum among all $\binom{n}{2}$ input vectors rather than all possible differences. Figure 5.8 illustrates such a pair of designs both of which have minimum interpoint distance of 0.30; the point $(0.2, 0.2)$ in the left panel design has been moved to $(0.025, 0.025)$ the design in the right panel is, intuitively, more space-filling than the design in the left panel. More careful inspection of these designs shows that the second smallest interpoint distance is greater for for right panel design than the left panel design. By using a more careful definition of minimaxity in which the number of pairs of the inputs with smallest, second smallest etc distances are accounted for, Morris and Mitchell (1995) were able to rank cases of equal minimum interpoint distance and eliminate such anomalies. Another criterion that accounts for the distances among all pairs of design vectors is the average of all ($\binom{n}{2}$) interpoint distances will be introduced below.

In sum, despite this initial criticism, Mm designs are often visually attractive and can be justified theoretically under certain circumstances (Johnson et al (1990)).

A second way in which points in a design $\mathcal{D}$ might be regarded as spread out over a design space $\mathcal{X}$ is for no point in $\mathcal{X}$ to be "too far" from a point in the design $\mathcal{D}$. To make this precise, again let $\rho_p(\cdot, \cdot)$ be a metric on $\mathcal{X}$. Denote the distance between an arbitrary input site $\boldsymbol{x} \in \mathcal{X}$ and a design $\mathcal{D} \subset \mathcal{X}$ by $\rho_p(\boldsymbol{x}, \mathcal{D})$, where

$$\rho_p(\boldsymbol{x}, \mathcal{D}) = \min_{x_i \in \mathcal{D}} \rho_p(\boldsymbol{x}, \boldsymbol{x}_i)\,.$$

An n-point design $\mathcal{D}_{mM}$ is defined to be *minimax distance design* if the maximum distance between arbitrary points $\boldsymbol{x} \in \mathcal{X}$ and the candidate design $\mathcal{D}_{mM}$ is a minimum over all designs $\mathcal{D}$ whose input vectors $\boldsymbol{x}_\ell \in \mathcal{X}$, $\ell = 1, \ldots, n$ namely

$$\min_{\mathcal{D}} \max_{x \in \mathcal{X}} \rho(\boldsymbol{x}, \mathcal{D}) = \max_{x \in \mathcal{X}} \rho(\boldsymbol{x}, \mathcal{D}_{mM}). \tag{5.4.4}$$

Another approach to spreading out points in the design space is to consider the distribution of distances between *all pairs of input vectors* and not merely the distance between the closest pair of input vectors. One example of such an approach minimizes the "average" of the reciprocals of the distances between pairs of design points. To describe the details of this proposal, it convenient to again let $\mathcal{D}$ be an arbitrary n-point design consisting of *distinct* input sites $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ from a rectangular or non-rectangular input region $\mathcal{X}$. Define the average reciprocal distance (ARD) among inputs in $\mathcal{D}$ to be

$$m_{(p,\lambda)}(\mathcal{D}) = \left(\frac{1}{\binom{n}{2}} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}} \left[\frac{1}{\rho_p(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right]^{\lambda} \right)^{1/\lambda}, \quad \lambda \geq 1 . \tag{5.4.5}$$

The combinatorial coefficient $\binom{n}{2}$ is the number of different pairs of points that can be drawn from a total of n distinct objects. For example, when $\lambda = 1$, the criterion function $m_{(p,1)}(\mathcal{D})$ is inversely proportional to the harmonic mean of the distances between design points.

For fixed (p, λ), an $n \times d$ design $\mathcal{D}_{av}$ is a *minimal ARD* (mARD) design if

$$m_{(p,\lambda)}(\mathcal{D}_{av}) = \min_{\mathcal{D} \subset \mathcal{X}} m_{(p,\lambda)}(\mathcal{D}) . \tag{5.4.6}$$

The optimality condition (5.4.6) favors designs that possess nonredundancy in the location of input sites; specifically the criterion does not allow design points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ that are (simultaneously) the same in *all* coordinates, i.e., with $\boldsymbol{x}_i = \boldsymbol{x}_j$. When $\lambda = 1$, the optimality condition (5.4.6) selects designs which maximize this harmonic mean, of course, preventing any "clumping" of design points. The nonredundancy requirement can be seen even more clearly for large values of λ. Taking $\lambda \to \infty$, the criterion function (5.4.5) becomes

$$m_{(p,\infty)}(\mathcal{D}) = \max_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}} \frac{1}{\rho_p(\boldsymbol{t}_i, \boldsymbol{t}_j)} . \tag{5.4.7}$$

Minimizing the right hand side of (5.4.7) is equivalent to maximizing (5.4.2). Thus, an n-point design$\mathcal{D}_{Mm}$ satisfying condition (5.4.6) for $\lambda = \infty$, namely,

$$m_{(p,\infty)}(\mathcal{D}_{Mm}) = \min_{\mathcal{D} \subset \mathcal{X}} m_{(p,\infty)}(\mathcal{D}) ,$$

is, in fact, a maximin distance design as defined previously because this criterion is equivalent to maximizing the minimum distance between all pairs of design points,

$$\max_{\mathcal{D} \subset \mathcal{X}} \min_{\boldsymbol{t}_i, \boldsymbol{t}_j \in \mathcal{D}} \rho_p(\boldsymbol{t}_i, \boldsymbol{t}_j) \propto \frac{1}{m_{(p,\infty)}(\mathcal{D}_{Mm})} .$$

Before considering an example, we note several computational strategies that have been used to find optimal space-filling designs. Mm designs can be computed by solving the mathematical programming problem

$$\begin{aligned} &\max z \\ &\text{subject to} \\ &z \le \rho_p(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad 1 \le i < j \le n \\ &\mathbf{0}_d \le \boldsymbol{x}_\ell \le \mathbf{1}_d, \quad 1 \le \ell \le n \end{aligned} \tag{5.4.8}$$

in which an addition decision variable z has been added to the unknown $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$; z is a lower bound for all distances in (5.4.8). While this problem can be solved by standard non-linear programming algorithms for "small" n, the computational difficulty with this approach is the number of constraints on z grows on the order of n^2 (see Stinstra et al (2003)).

Do we want to show the geometry of the Mm and mM designs? other methods for finding Mm, mM, mARD designs?

Example 5.8. Figure 5.6 displays Minimum ARD designs with Euclidean distance ($p = 2$) for $\lambda = 1$ and $\lambda = \infty$ when $n = 6$ and $d = 2$; by (5.4.7) the latter design is Mm design for this Euclidean distance case. Both designs concentrate points on or near the boundary of $\mathcal{X}$ so that the projections of the design points onto either axis produces multiple observations in 1-d. If the output depends primarily on one of the inputs, say x_1, this means that such a design will not fully explore x_1 space. We can remedy this feature of the design by restricting the class of available designs to only include, say, LHDs. This provides a computationally-convenient method of generating space-filling designs for computer experiments. Figure 5.2 is an example of a mARD within the class of LHDs for $p = 1$ and $\lambda = 1$. The use of multiple criteria to select designs is discussed further below. ■

As noted above, neither the Mm nor the mARD optimal designs need not have projections that are nonredundant. To reiterate, consider a computer experiment involving $d = 5$ input variables, only three of which (say) are active. In this event, a desirable property of an optimal $n \times 5$ design is nonredundancy of input sites projected onto the three-dimensional subspace of the active inputs. Such designs can be generated by computing the criterion values (5.4.5) for each relevant projection of the full design $\mathcal{D}$ and averaging these to form a new criterion function which is then minimized by choice of design $\mathcal{D}$. The approach is implemented by the Algorithms for the Construction of Experimental Designs (ACED) software of Welch (1985), among other packages. The Welch (1985) software was used to compute the optimal designs of this section.

Formally, the projection approach sketched in the previous paragraph can be described as follows. Let J denote the index set of subspace dimensions in which nonredundancy of input sites is desired. For each $j \in J$, let $\{S_{kj}\}$ denote the k^{th} design in an enumeration of all j-dimensional projections of $\mathcal{D}$ for $k = 1, \ldots, \binom{n}{j}$, where $\binom{n}{j} = n!/(j!(n-j)!)$ is the number of subsets of size j that can be drawn from n distinct objects. Because the maximum distance apart that points can lie depends

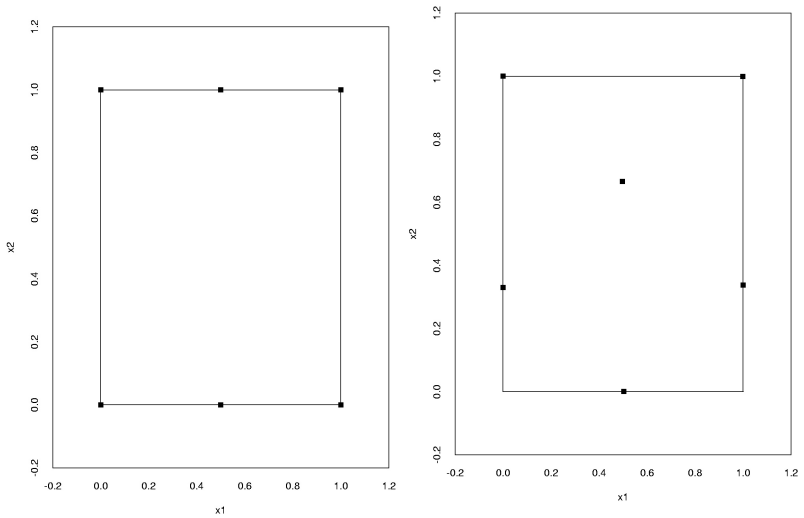


Fig. 5.6 Minimum ARD designs with respect to Euclidean distance ($p = 2$) for $\lambda = 1.0$ (left panel) and for $\lambda = \infty$ (right panel)

on the dimension of the space, it is essential that the $\rho_p(\cdot, \cdot)$ of points in j-d be normalized by this maximum distance of $j^{1/p}$ in order for distances to be comparable across different dimensional space.

For $k = 1, \ldots, \binom{n}{2}$ and $j \in J$ define the minimum distance for the projected design $\mathcal{D}_{kj}$ to be

$$\min_{x_h^\star, x_\ell^\star \in \mathcal{D}_{kj}} \frac{\rho_p(\boldsymbol{x}_h^\star, \boldsymbol{x}_\ell^\star)}{j^{1/p}} \tag{5.4.9}$$

and the average reciprocal distance for $\mathcal{D}_{kj}$ to be the (modified) (5.4.5),

$$m_{J,(p,\lambda)}(\mathcal{D}_{kj}) = \left(\frac{1}{\binom{n}{2}} \sum_{\boldsymbol{x}_h^\star, \boldsymbol{x}_\ell^\star \in \mathcal{D}_{kj}} \left[\frac{j^{1/p}}{\rho_p(\boldsymbol{x}_h^\star, \boldsymbol{x}_\ell^\star)} \right]^\lambda \right)^{1/\lambda} . \tag{5.4.10}$$

Here, $\boldsymbol{x}_i^\star$ denotes the projection of $\boldsymbol{x}_i$ into the appropriate subspace determined by the values of j and k. Define the J-minimum of inputs in the design $\mathcal{D}$ to be

$$\rho_J(x, \mathcal{D}) = \min_{j \in J} \min_{k \in \{1, \ldots, \binom{n}{j}\}} \min_{\boldsymbol{x}_h^\star, \boldsymbol{x}_\ell^\star \in \mathcal{D}_{kj}} \frac{\rho_p(\boldsymbol{x}_h^\star, \boldsymbol{x}_\ell^\star)}{j^{1/p}} \tag{5.4.11}$$

and the J-average reciprocal projection design criterion function to be,

$$av_{J,(p,\lambda)}(\mathcal{D}) = \left(\frac{1}{\binom{n}{2} \times \sum_{j\in J} \binom{n}{j}} \sum_{j\in J} \sum_{k=1}^{\binom{n}{j}} \sum_{\boldsymbol{x}_h^*, \boldsymbol{x}_\ell^* \in \mathcal{D}_{kj}} \left[\frac{j^{1/p}}{\rho_p(\boldsymbol{x}_h^\star, \boldsymbol{x}_\ell^\star)} \right]^\lambda \right)^{1/\lambda}$$
$$= \left(\frac{1}{\sum_{j\in J} \binom{n}{j}} \sum_{j\in J} \sum_{k=1}^{\binom{n}{j}} [m_{J,(p,\lambda)}(\mathcal{D}_{kj})]^\lambda \right)^{1/\lambda} . \tag{5.4.12}$$

An n-point design $\mathcal{D}_{MmP}$ is maximum with respect the projection criterion (**??**) provided

$$\rho_J(x, \mathcal{D}_{MmP}) = \max_{\mathcal{D}} \rho_J(x, \mathcal{D}) \tag{5.4.13}$$

and is $\mathcal{D}_{avp}$ is minimal ARD with respect to the projection criterion (5.4.12) if

$$\mathrm{av}_{J,(p,\lambda)}(\mathcal{D}_{avp}) = \min_{\mathcal{D}\subset\mathcal{X}} \mathrm{av}_{J,(p,\lambda)}(\mathcal{D}) . \tag{5.4.14}$$

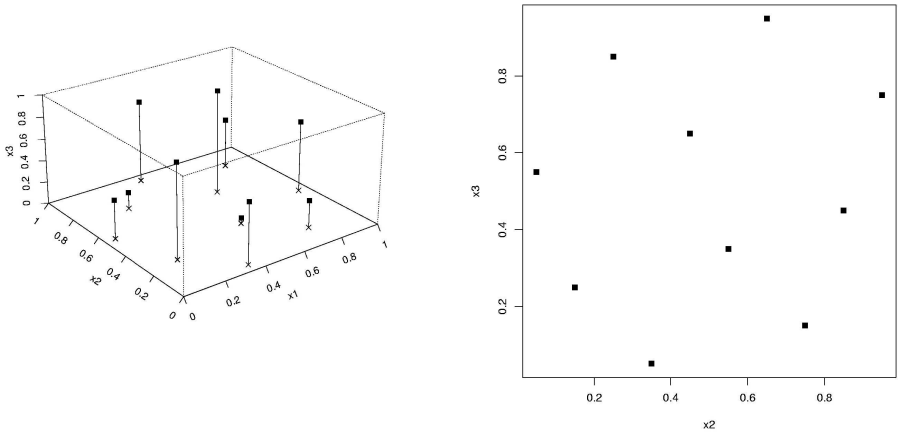


Fig. 5.7 Left panel: a 3-d plot of a $n = 10$ point optimal mARD design within the class of LHDs when $p = \lambda = 1$ and $J = \{2, 3\}$. Right panel: projection of left panel design onto x_1-x_2 plane.

Example 5.9. The optimal average projection designs (5.4.14) will also be space-filling if the class of available designs is restricted to LHDs. As an example, let $n = 10$ and $d = 3$. An optimal mARD design in the class of LHDs was generated with the specifications $p = \lambda = 1$ and $J = \{2, 3\}$. Figure 5.7 presents the design 3-d and the projection of the design onto the (x_2, x_3) subspace. Note that $1 \notin J$, as LHDs are nonredundant in each one-dimensional subspace by definition. ■

Mm and mARD designs with specified projection dimensions J are alternatives to randomly LHDs and randomized orthogonal arrays for producing designs that are space-filling and are (reasonably) uniformly spread out when projected onto certain lower dimensional subspaces. Unlike randomized orthogonal arrays that only exist for certain values of n, these designs can be generated for any sample size.

??

5.5 Distance-based Designs for Non-rectangular Regions

Section 5.2-5.4 described several criteria for constructing space-filling designs when the input region is a hyper-rectangular. This section describes how *maximin distance* (Mm) and the *minimum ARD* (mARD) criteria from Section 5.4 have been applied to non-rectangular input regions.

As the following example illustrates, non-rectangular input regions occur naturally in many applications where the range of one or more inputs is related to that of others. Hayeck (2009) studied the effects of four variables, one a biomechanical engineering design input (x_1) and three environmental inputs ($x_2 - x_4$), on the functioning of a total elbow prosthesis. The biomechanical input was the tip displacement (in *mm*), and the environmental inputs were the rotation of the implant axis about the lateral axis at the tip, the rotation of the implant axis about the anterior axis at the tip, and the rotation about the implant axis (all in degrees, denoted °). The following constraints were imposed on the inputs based on anatomical considerations

$$\begin{array}{rcrcl} 0 & \leq & x_1 & \leq & 10 \\ -10 & \leq & 5x_2 + 2x_3 & \leq & 10 \\ -10 & \leq & -5x_2 + 2x_3 & \leq & 10 \\ -15 & \leq & x_4 & \leq & 15. \end{array} \tag{5.5.1}$$

These constraints state, among things, that the maximum tip displacement is 10 *mm* relative to some coordinate system; the rotation of the implant axis is 10° about lateral axis at the tip and 4° about anterior axis at the tip; and the rotation about the implant axis is ±15°. The outputs of the computational simulation where various stresses and strains in the elbow.

The bulk of this section restricts attention to input regions that are bounded polytopes, i.e., have the form

$$\{\boldsymbol{x} \in \mathbb{R}^d \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\} \tag{5.5.2}$$

for given $\boldsymbol{A}$ and $\boldsymbol{b}$. The Hayeck (2009) input region (5.5.1) satisfies (5.5.2) for

$$\boldsymbol{A} = \begin{pmatrix} +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & +5 & +2 & 0 \\ 0 & -5 & +2 & 0 \\ 0 & +5 & -2 & 0 \\ 0 & 0 & 0 & +1 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

and

$$\boldsymbol{b} = (10, 0, 10, 10, 10, 10, 15, 15)^\top .$$

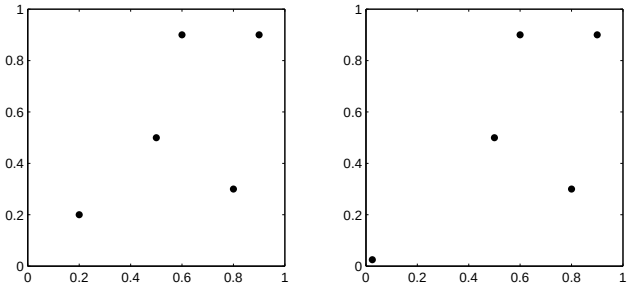


Fig. 5.8 Two designs on $[0, 1]^2$ with the same minimum interpoint distance of 0.30.

Recall that a design $\mathcal{D}_{Mm}$ is Mm provided it satisfies (5.4.3) while a design $\mathcal{D}_{av}$ is mARD provided it satisfies (5.4.6). The class of designs used in the maximization in (5.4.6) are those $\mathcal{D}$ having rows $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^\top$, $i = 1, \ldots n$, belonging to the desired input region. For example, when the input region is the bounded polytope (5.5.2), then

$$\mathcal{D} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{pmatrix}$$

where $\boldsymbol{A}\boldsymbol{x}_i \leq \boldsymbol{b}$, for $i = 1, \ldots, n$.

As noted in Section 5.4, there are several practical and philosophical difficulties associated with the computation and use of maximin designs. First, because inputs with different scales can cause the computation of a maximin designed to be dominated by those inputs having larger ranges, the determination of a maximin design for a non-rectangular input region is performed for the problem in which each input has been scaled and shifted to the interval [0,1]. For example, for the bounded input region (5.5.2), the maximum of input x_j can be obtained by solving the linear program

$$\max \quad x_j \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} .$$

Second, maximin designs need not have "space-filling" projections onto subsets of the input variables although using the J maximin criterion and selecting designs from the class of LHDs can eliminate this problem.

First consider the construction of maximin designs for the case of inputs that satisfy (5.5.2). The mathematical program (5.4.8) for the Mm design can be modified to that of solving

$$\begin{aligned} &\max z \\ &\text{subject to} \\ &z \le \rho_2(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad 1 \le i < j \le n \\ &\boldsymbol{A}\boldsymbol{x}_\ell \le \boldsymbol{b}, \qquad 1 \le \ell \le n \end{aligned} \tag{5.5.3}$$

in which the [0,1] bounds for each input are replaced by the bounded polytope constraints. Other constraints on the $\boldsymbol{x}_\ell$ can be handled similarly.

Trosset (1999)) described an approximate solution to the problem of finding a Mm design. He replaced $\rho_p(\cdot,\cdot)$ in (**??**) by a decreasing function of $\rho_p(\cdot,\cdot)$, e.g., $\phi(w) = 1/w$ which changes the minimum in (**??**) to

$$\max_{i<j} \phi(\rho_p(\boldsymbol{x}_i, \boldsymbol{x}_j)) \tag{5.5.4}$$

and then replaces the maximization in (**??**) with that of minimizing

$$\left\{ \sum_{i<j} \phi(\rho_2(\boldsymbol{x}_i, \boldsymbol{x}_j))^\lambda \right\}^{1/\lambda}. \tag{5.5.5}$$

For large λ, a design that minimizes (5.5.5) subject to $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$ is an approximate Mm design because (5.5.5) converges to (5.5.4) as $\lambda \to \infty$.

Stinstra et al (2003) introduced an algorithm that allows larger problems (5.5.3) to be solved. Their algorithm solves a set of n subproblems to update a current feasible point $(\boldsymbol{x}_1^c, \ldots, \boldsymbol{x}_n^c)$ satisfying the constraints (5.5.3) to an improved solution. The update step to find $\boldsymbol{x}_i^{c+1}$ from $\boldsymbol{x}_i^c$ when components with $\ell < i$ have been updated and those with $\ell > i$ have not been updated is

$$\begin{aligned} &\max w \\ &\text{subect to} \\ &w \le \rho_2(\boldsymbol{x}_i, \boldsymbol{x}_\ell^{c+1}), \qquad \ell < i \\ &w \le \rho_2(\boldsymbol{x}_i, \boldsymbol{x}_\ell^{c}), \qquad \ell > i \\ &\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \end{aligned} \tag{5.5.6}$$

for $(w^\star, \boldsymbol{x}_i^\star)$. Set $\boldsymbol{x}_i^{c+1} = \boldsymbol{x}_i^\star$. This cycle of n steps is repeated until a given minimum improvement in (**??**) occurs or a computational budget is exhausted.

As for rectangular input regions, Draguljić et al (2012) added criteria to that of maximinity which a design is required to satisfy. First, the consider "non-

collapsingness" which can be thought of as an attempt to provide a space-filling design in each input and is thus similar in spirit to that of the LHD criterion. Beyond non-collapsingness, they considered adding either maximin or a maximum average distance criterion for the design.

5.6 Designs Obtained from Quasi-Random Sequences

Quasi-random sequences are intended to produce finite sequences of points that fill the d-dimensional unit hypercube uniformly with the property that a design with sample size n is obtained from the design of sample size $n-1$ by adding a point to the design. Although introduced for numerically evaluating multi-dimensional integrals, they also allow one to generate space-filling designs.

Several such sequences have been proposed, including Halton sequences (Halton (1960)) Sobol´ sequences, (Sobol´ (1967) and Sobol´ (1976)) and Niederreiter sequences (Niederreite (1988)).

To construct a Halton sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ of n points on the d-dimensional unit hypercube, begin by choosing d prime numbers, or bases, $b_1, b_2, \ldots, b_d$. These could be, for example, the first d prime numbers. The base b_j will be used to construct the $j-th$ coordinates of the $\{\boldsymbol{x}_i$.

Next, select an integer m. Next, for suitably large t_{mj} (the highest power of b_j used in the representation of m in base b_j), represent the integer m in base b_j as

$$m = \sum_{k=0}^{t_{mj}} a_{jk}(m) b_j^k, \; j = 1, \ldots, d. \tag{5.6.1}$$

Next, form

$$x_{1j} = \sum_{k=0}^{t_{mj}} a_{j,k-t_{mj}-1}(m) b_j^{k-t_{mj}-1}, \; j = 1, \ldots, d. \tag{5.6.2}$$

Note that in forming x_{1j} we have simply reversed the digits in the representation of m in base b_j and placed these reversed digits after a decimal point.

Set $m = m + i - 1$ and repeat the above to form the x_{ij}.

Example 5.10. We compute the first five points in a 2-dimensional Halton sequence. We use bases $b_1 = 2$ and $b_2 = 3$. We begin with $m = 4$. In base 2, 4 is 100 and in base 3 11. Reversing the digits and adding a decimal point, $\boldsymbol{x}_1 = (.001_2, .11_3)$, where the subscript indicates the base. Converting to base 10, $.001_2 = 0\times2^{-1}+0\times2^{-2}+1\times2^{-3} = 1/8 = 0.125$ and $.11_3 = 1\times3^{-1} + 1\times3^{-2} = 1/3 + 1/9 = 0.444$. Thus, the first point in our Halton sequence is $\boldsymbol{x}_1 = (.125, .444)$.

Next, we increase$m$ by 1 to 5. In base 2, 5 is 101 and in base 3 12. Reversing the digits and adding a decimal point, $btx_2 = (.101_2, .21_3)$. Converting to base 10, $.101_2 = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} = 1/2 + 1/8 = 0.625$ and $.21_3 = 2\times3^{-1} +$

$1 \times 3^{-2} = 2/3 + 1/9 = 0.7784$. Thus, the second point in our Halton sequence is $btx_2 = (.625, .778)$.

The next 3 points correspond to $m = 6, 7, 8$. In base 2, these are 110, 111, and 1000. In base 3, these are 20, 21, and 22. Reversing digits and adding a decimal point, $btx_3 = (.011_2, .02_3)$, $btx_4 = (.111_2, .12_3)$, and $btx_5 = (.0001_2, .22_3)$. Converting to base 10, one finds $\boldsymbol{x}_3 = (.375, .222)$, $btx_4 = (.875, .556)$, and $\boldsymbol{x}_5 = (.0625, .8889)$. Figure 5.9 shows the resulting 5-point design.

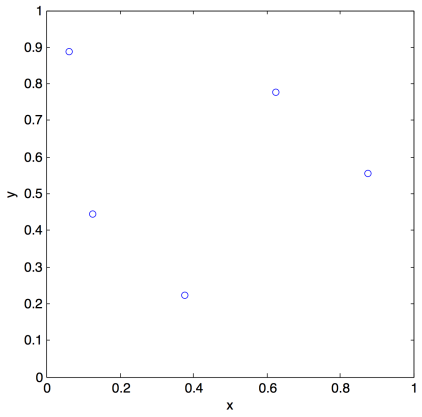


Fig. 5.9 5 point, $d = 2$ variable Halton sequence.

Halton sequences are relatively easy to calculate and have been found to be acceptably uniform for lower dimensions (d up to about 10). For higher dimensions the quality degrades rapidly because two-dimensional planes occur in cycles with decreasing periods.

Methods for creating sequences that behave better (appear uniform even in higher dimensions) have been developed by Sobol´ and Niederreiter.

To introduce the construction of the Sobol′ sequence consider working in one-dimension. To generate a sequence of values $x_1, x_2 ...$ with $0 < x_i < 1$, first we need to construct a set of *direction numbers* $v_1, v_2, ...$ Each v_i is a binary fraction that can be written $v_i = \frac{m_i}{2^i}$, where m_i is an odd integer such that $0 < m_i < 2^i$.

To obtain m_i the construction starts by choosing a primitive polynomial in the field $\mathbb{Z}_2$, i.e. one may choose $P = x^u + a_1 x^{u-1} + ... + a_{u-1} x + 1$ where each a_i is 0 or 1 and P is an arbitrary chosen primitive polynomial of degree u in $\mathbb{Z}_2$. Then, the m_i's can be calculated recurrently as

$$m_i = 2a_1 m_{i-1} \oplus 2^2 a_2 m_{i-2} \oplus ... \oplus 2^{u-1} a_{u-1} m_{i-u+1} \oplus 2^u m_{i-u} \oplus m_{i-u}$$

where each term is expressed in base 2 and $\oplus$ denotes a bit-by-bit exclusive-or operation, i.e

$$0 \oplus 0 = 0,\ 0 \oplus 1 = 1 \oplus 0 = 1,\ 1 \oplus 1 = 0.$$

When using a primitive polynomial of degree d, the initial values $m_1, ..., m_u$ can be arbitrarily chosen provided that each m_i is odd and $m_i < 2^i$, $i = 1, ..., u$.

Example 5.11. If we choose the primitive polynomial $x^3 + x + 1$ and the initial values $m_1 = 1$, $m_2 = 3$, $m_3 = 7$, m_i's are calculated as follows:

$$m_i = 4m_{i-2} \oplus 8m_{i-3} \oplus m_{i-3}.$$

Then
$m_4 = 12 \oplus 8 \oplus 1 = 1100 \oplus 1000 \oplus 0001 = 0101 = 0 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 2^0 = 5$
$m_5 = 28 \oplus 24 \oplus 3 = 11100 \oplus 11000 \oplus 00011 = 00111 = 7$
$m_6 = 20 \oplus 56 \oplus 7 = 010100 \oplus 111000 \oplus 000111 = 43$
and
$v_1 = \frac{m_1}{2^1} = \frac{1}{2^1} = 0.1$ in binary
$v_2 = \frac{m_2}{2^2} = \frac{3}{2^2} = 0.11$ in binary
$v_3 = \frac{m_3}{2^3} = \frac{7}{2^3} = 0.111$ in binary
$v_4 = \frac{m_4}{2^4} = \frac{5}{2^4} = 0.0101$ in binary, and so on.

In order to generate the sequence $x_1, x_2, ...$, Sobol′ proposed using

$$x_i = b_1 v_1 \oplus b_2 v_2 \oplus \cdots$$

and

$$x_{i+1} = x_i \oplus v_c$$

where $\cdots b_3 b_2 b_1$ is the binary representation of i and b_c is the rightmost zero-bit in the binary representation of i.

The first few values of x are thus generated as follows. To start the recurrence, take $x_0 = 0$.

$$
\begin{aligned}
Initialization: \; x_0 &= 0 \\
i &= 0 \text{ in binary so} \\
c &= 1 \\
Step\ 1: \; x_1 &= x_0 \oplus v_1 \\
&= 0.0 \oplus 0.1 \text{ in binary} \\
&= 0.1 \text{ in binary} \\
&= \frac{1}{2} \\
i &= 01 \text{ in binary so} \\
c &= 2 \\
Step\ 2: \; x_2 &= x_1 \oplus v_2 \\
&= 0.10 \oplus 0.11 \text{ in binary} \\
&= 0.01 \text{ in binary} \\
&= \frac{1}{4} \\
i &= 10 \text{ in binary so} \\
c &= 1
\end{aligned}
$$

$$
\begin{aligned}
Step\ 3: \; x_3 &= x_2 \oplus v_1 \\
&= 0.01 \oplus 0.10 \text{ in binary} \\
&= 0.11 \text{ in binary} \\
&= \frac{3}{4} \\
i &= 011 \text{ in binary so} \\
c &= 3
\end{aligned}
$$

and so on.

To generalize this procedure to s dimensions, Sobol´ (1976) shows that in order to obtain $O(\log^s n)$ discrepancy, where n represents the number of points, it suffices to choose s distinct primitive polynomials, calculate s sets of *direction numbers* and then generate each component x_{ij} of the quasi-random vector separately.

The uniformity of a Sobol´ sequence can be very sensitive to the starting values, especially in higher dimensions. Various criteria exist for starting values, $m_1, m_2, \ldots$ to improve uniformity.

Example 5.12. In Section 3.3, Sobol´ sequences were used to select the values of the environmental variables and compared to other methods. The left-hand panel of Figure 5.10 displays one of the six, two-dimensional projections of the 40 point Sobol´ sequence in $d = 4$ variables that were used as inputs to generate the fire containment data displayed in Figure 1.2. Example 3.8 also uses this 40 point data

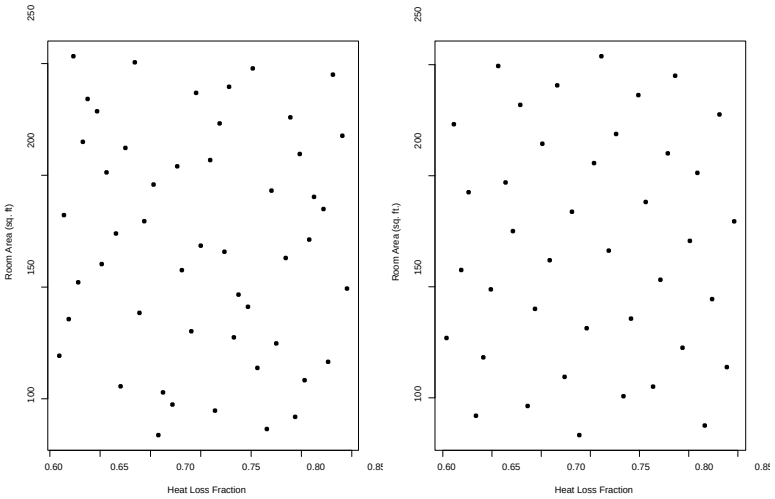


Fig. 5.10 Left Panel—projection of the 40 point, $d = 4$ variable Sobol´ sequence described in Section 1.3 onto the room area × heat loss fraction plane; Right Panel—projection of the 40 point maximin LHD for the same four variables into the room area × heat loss fraction plane.

set. The corresponding two-dimensional projection for the 40 point maximin LHD is shown in the right-hand panel of the same figure. It is clear from Figure 5.10 that the LHD is more evenly spread out than the design based the Sobol´ sequence. Thus, if it is important that the design be evenly spread out, the LHD appears to be preferable. On the other hand, the design based on the Sobol´ sequence appears to exhibit a greater variety of inter-point distances (distances between pairs of points in the design) than the LHD. If a greater variety of inter-point distances provides more information about the correlation parameters (and hence allows one to better estimate these parameters), then designs based on a Sobol´ sequence (or other types of sequences that have been used in numerical integration) may be preferable to the LHD. ■

Niederreite (1988) proposed a new method of generating quasi-Monte Carlo sequences intended to improve on Sobol´ sequences. Let $\triangle(N)$ denote $n \times D_n^*$, where D_n^* is the star discrepancy. It is believed that the best possible bound for the discrepancy of the first n terms of a sequence of points in $[0, 1)^s$ is of the form

$$\triangle(n) \leq C_s(\log n)^s + O((\log n)^{s-1})$$

for all $n \geq 2$. The methods proposed by Niederreiter yield sequences with the lowest C_s currently known.

We will not discuss the construction of Niederreiter sequences, but details can be found in, for example, Lemieux (2009).

Code exists for generating these sequences and a Google search online will identify several sources. R code exists for generating Sobol´ sequences. To obtain and use this code, do the following.

- Open R
- Use Package Installer and Package Manager (under the Packages&Data menu) to make sure that fBasics, fCalendar, fExtremes, fMultivar, fOptions, fPortfolio, and fSeries are installed and loaded.
- Use the runif.sobol command. To see how to use this, click on the fOptions package in the R Package Manager window and then on the runif.sobol link.

The basic format is
> runif.sobol([number of runs], [number of variables or dimensions])
but additional options are available.

Halton sequences in d dimensions can be generated in Matlab using the p = haltonset(d) command.

Sobol´ sequences in d dimensions can be generated in Matlab using the p = sobolset(d) command.

Both commands yield a very large sequence of points.

Both commands are available in the Statistics toolbox. See the Help menu in Matlab for more information.

One can also find online Matlab code for generating Niederreiter sequences. For example, see
people.sc.fsu.edu/~ burkardt/m_src/niederreiter2/niederreiter2.html.

As mentioned previously, Halton, Sobol´, and Niederreiter sequences have the useful property that a longer sequence can be constructed from a shorter one by adding points to the shorter sequence. This is in contrast to LHDs, where except for special cases, the entire design must be recomputed if a LHD containing more points is desired. More research is needed to determine what features of a design are important for estimating the correlation parameters in our models. This question is very difficult to answer analytically, and extensive empirical studies would be useful for better understanding what sorts of designs perform well and for which models.

??

5.7 Uniform Designs

In Section 5.2 we considered criteria for selecting a space-filling design based on sampling methods and, in Section 5.4, criteria based on distances between points. In this section, we consider a third intuitive design principle based on comparing the distribution of the points in a design to the uniform distribution.

As in Subsection 5.2.3, suppose that the vector of inputs is d-dimensional and denoted by $\boldsymbol{x} = (x_1, \ldots, x_d)$. Also again assume that $\boldsymbol{x}$ must fall in the d-dimensional hyper-rectangle $\mathcal{X} = \bigtimes_{i=1}^{d} [a_i, b_i]$. Let $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ denote the set of n points at which we will observe the response $y(\boldsymbol{x})$. If we wish to emphasize that $\boldsymbol{x}$ is a

random variable, we will use the notation $\boldsymbol{X}$. This would be the case, for example, if we are interested in $\mathrm{E}\{y(\boldsymbol{X})\}$. Below we take $\boldsymbol{X} \sim F(\cdot)$ where

$$F(\boldsymbol{x}) = \prod_{i=1}^{d} \left(\frac{x_i - a_i}{b_i - a_i} \right) \tag{5.7.1}$$

is the uniform distribution on $\mathcal{X}$ (other choices of distribution function are possible).

Fang et al (2000) and Fang et al (2005) discuss the notion of the *discrepancy* of a design $\mathcal{D}$, which measures the extent to which $\mathcal{D}$ differs from a completely uniform distribution of points. To be specific, let F_n be the empirical distribution function of the points in $\mathcal{D}$, namely

$$F_n(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} I\{\boldsymbol{X}_i \le \boldsymbol{x}\}, \tag{5.7.2}$$

where $I\{E\}$ is the indicator function of the event E and the inequality is with respect to the componentwise ordering of vectors in $\mathbb{R}^d$. The L_∞ discrepancy, sometimes called *star discrepancy* or simply discrepancy, is denoted $D_\infty(\mathcal{D})$ and is defined as

$$D_\infty(\mathcal{D}) = \sup_{\boldsymbol{x} \in \mathcal{X}} | F_n(\boldsymbol{x}) - F(\boldsymbol{x}) | . \tag{5.7.3}$$

This is perhaps the most popular measure of discrepancy and is the Kolmogorov-Smirnov statistic for testing fit to the uniform distribution.

Example 5.13. Suppose $d = 1$ and $\mathcal{X} = [0, 1]$ is the unit interval. It is not too difficult to show that the n point set

$$\mathcal{D} = \left\{ \frac{1}{2n}, \frac{3}{2n}, \ldots, \frac{2n-1}{2n} \right\}$$

has discrepancy $D_\infty(\mathcal{D}) = 1/2n$ because $F(x) = x$ in this case. ■

Another important measure of discrepancy is the L_p discrepancy of $\mathcal{D}$ which is denoted by $D_p(\mathcal{D})$ and defined by

$$D_p(\mathcal{D}) = \left[\int_{\mathcal{X}} |F_n(\boldsymbol{x}) - F(\boldsymbol{x})|^p \, d\boldsymbol{x} \right]^{1/p}. \tag{5.7.4}$$

The L_∞ discrepancy of $\mathcal{D}$ is a limiting case of L_p discrepancy obtained by letting p go to infinity.

Niederreiter (1992) discusses the use of discrepancy for generating uniformly distributed sequences of points in the context of quasi-Monte Carlo methods. Designs taking observations at sets of points with small discrepancies would be considered more uniform or more spread out than designs corresponding to sets with larger discrepancies. *Uniform designs* take observations at a set of points that minimizes D_p.

Other than the fact that it seems intuitively reasonable to use designs that are spread uniformly over $\mathcal{X}$, why might one consider using a uniform design? One rea-

son that has been proposed is the following. Suppose we are interested in estimating the mean of $g(y(\boldsymbol{X}))$,

$$\mu = \mathrm{E}\{g(y(\boldsymbol{X}))\} = \int_{\mathcal{X}} g(y(\boldsymbol{x})) \frac{1}{\prod_{i=1}^{d}(b_i - a_i)} d\boldsymbol{x},$$

where $g(\cdot)$ is some known function. We consider the properties of the naíve moment estimator

$$T = T(y(\boldsymbol{X}_1), \ldots, y(\boldsymbol{X}_n)) = \frac{1}{n} \sum_{j=1}^{n} g(y(\boldsymbol{X}_j)).$$

The Koksma-Hlawka inequality (Niederreiter (1992)) gives an upper bound on the absolute error of this estimator, namely

$$\mid T(y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_n)) - \mu \mid \leq D_\infty(\mathcal{D}) V(g),$$

where $V(g)$ is a measure of the variation of g that does not depend on $\mathcal{D}$ (see page 19 of Niederreiter (1992) for the definition of $V(g)$). For fixed $g(\cdot)$, this bound is a minimum when $\mathcal{D}$ has minimum discrepancy. This suggests that a uniform design may control the maximum absolute error of T as an estimator of μ. Also, because this holds for any $g(\cdot)$, it suggests that uniform designs may be robust to the choice of $g(\cdot)$ because they have this property regardless of the value of $g(\cdot)$.

However, just because an upper bound on the absolute error is minimized, it does not necessarily follow that a uniform design minimizes the maximum absolute error over $\mathcal{X}$ or has other desirable properties. Furthermore, in the context of computer experiments, we are usually not interested in estimating μ. Thus, the above is not a completely compelling reason to use a uniform design in computer experiments as discussed here.

Wiens (1991) provides another reason for considering uniform designs. Suppose we believe the response $y(\boldsymbol{x})$ follows the regression model

$$y(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i f_i(\boldsymbol{x}) + \varphi(\boldsymbol{x}) + \epsilon,$$

where the $\{f_i\}$ are known functions, the β_i unknown regression parameters, φ is an unknown function representing model bias, and ϵ normal random error. Wiens (1991) shows that under certain conditions on φ, the uniform design is best in the sense of maximizing the power of the overall F test of the regression.

Fang et al (2000) provide yet another reason why one may wish to use uniform designs. They note that in orthogonal designs, the points are typically uniformly spread out over the design space. Thus, there is the possibility that uniform designs may often be orthogonal. To explore this further, they use computer algorithms to find designs that minimize a variety of measures of discrepancy and in doing so generate a number of orthogonal designs. Efficient algorithms for generating designs that minimize certain measures of discrepancy, therefore, may be useful in searching for orthogonal designs.

Fang et al (2000) discuss a method for constructing (nearly) uniform designs. For simplicity, assume $\mathcal{X}$ is $[0,1]^d$. In general, finding a uniform design is not easy. One way to simplify the problem is to reduce the domain of $\mathcal{X}$. Obviously, a uniform design over this reduced domain may not be close to uniform over $\mathcal{X}$, but suitable selection of a reduced domain may yield designs which are nearly uniform. Based on the uniform design for $d = 1$, we might proceed as follows. Let $\boldsymbol{\Pi} = (\Pi_{ij})$ be an $n \times d$ matrix such that each column of $\boldsymbol{\Pi}$ is a permutation of the integers $\{1, 2, \ldots, n\}$. Let $\boldsymbol{X}(\boldsymbol{\Pi}) = (x_{ij})$ be the $n \times d$ matrix defined by

$$x_{ij} = (\Pi_{ij} - 0.5)/n$$

for all i, j. The n rows of $\boldsymbol{X}$ define n points in $\mathcal{X} = [0,1]^d$. Hence, each matrix $\boldsymbol{\Pi}$ determines an n point design. For example, when $d = 1$, if $\boldsymbol{\Pi} = (1, 2, \ldots, n)^\top$, then

$$\boldsymbol{X}(\boldsymbol{\Pi}) = \left(\frac{1}{2n}, \frac{3}{2n}, \ldots, \frac{2n-1}{2n}\right)^\top,$$

which is the uniform design in $d = 1$ dimension. Note that the n rows of $\boldsymbol{X}(\boldsymbol{\Pi})$ correspond to the sample points of an LHD with points at the centers of each sampled cell. One might search over the set $\mathcal{P}$ of all possible permutations $\boldsymbol{\Pi}$, selecting the $\boldsymbol{\Pi}$ that produces the n point design with minimum discrepancy. One would hope that this choice of design is nearly uniform over $\mathcal{X}$. Fang et al (2000) describe two algorithms for conducting such a search (see Section 5 of their paper). Bratley et al (1994) is an additional source for an algorithm that can be used to generate low-discrepancy sequences of points and hence (near) uniform designs.

The discrepancies D_∞ for two designs that appear to be equally uniform may not be the same. The following example illustrates such a case.

Example 5.14. Suppose $d = 2$, $\mathcal{X} = [0,1]^2$, and consider the class of all designs generated by the set of permutations $\mathcal{P}$ introduced in the previous paragraph. One member of this class of designs is

$$\mathcal{D}_{diag} = \left\{\left(\frac{1}{2n}, \frac{1}{2n}\right), \left(\frac{3}{2n}, \frac{3}{2n}\right), \ldots, \left(\frac{2n-1}{2n}, \frac{2n-1}{2n}\right)\right\}.$$

This n point design takes observations along the diagonal extending from the origin to the point $(1, 1)$. Intuitively, we would expect $\mathcal{D}_{diag}$ to be a poor design, because it takes observations only along the diagonal and does not spread observations over $[0,1]^2$. To compute the discrepancy of $\mathcal{D}_{diag}$, we first compute the empirical distribution function F_n for $\mathcal{D}_{diag}$ at an arbitrary point $\boldsymbol{x} = (x_1, x_2)$ in $[0,1]^2$. Notice that points in $\mathcal{D}_{diag}$ have both coordinates equal and it is not too hard to show from Equation (**??**) that

$$F_n(x_1, x_2) = \frac{\text{number of pts. in } \mathcal{D}_{diag} \text{ with first coordinate} \leq \min\{x_1, x_2\}}{n}.$$

Notice that $F_n(\cdot,\cdot)$ is constant almost everywhere except for jumps of size $1/n$ at points for which one of the coordinates takes one of the values $\frac{1}{2n}, \frac{3}{2n}, \ldots, \frac{2n-1}{2n}$. In particular, $F_n(x_1, x_2)$ has value $\frac{m}{n}$ $(1 \le m \le n)$ on the set $\mathcal{X}_m$:

$$\left\{(x_1, x_2) \in [0,1]^2 : \frac{2m-1}{2n} \le \min\{x_1, x_2\} < \frac{2m+1}{2n}\right\}.$$

Recall from (**??**) that $F(\cdot)$ is the uniform distribution

$$F(\boldsymbol{x}) = x_1 x_2$$

on $\mathcal{X} = [0,1]^2$. On $\mathcal{X}_m$, the minimum value of $F(\boldsymbol{x})$ is $\left(\frac{2m-1}{2n}\right)^2$ and the supremum of $F(\boldsymbol{x})$ is $\frac{2m+1}{2n}$. This supremum is obtained in the limit as $\epsilon \to 0$ along the sequence of points $\left(\frac{2m+1}{2n} - \epsilon, 1\right)$. Thus, over $\mathcal{X}_m$, the supremum of $|F_n(\boldsymbol{x}) - F(\boldsymbol{x})|$ is either $\left|\frac{m}{n} - \left(\frac{2m-1}{2n}\right)^2\right|$ or $\left|\frac{m}{n} - \frac{2m+1}{2n}\right| = \frac{1}{2n}$. For $1 \le m \le n$, it is not difficult to show that

$$\left|\frac{m}{n} - \left(\frac{2m-1}{2n}\right)^2\right| > \frac{1}{2n}.$$

Hence, over the set of all points $\boldsymbol{x}$ for which $F_n(\boldsymbol{x})$ has value $\frac{m}{n}$, the supremum of $|F_n(\boldsymbol{x}) - F(\boldsymbol{x})|$ is

$$\frac{m}{n} - \left(\frac{2m-1}{2n}\right)^2 = \frac{nm - m^2 + m}{n^2} - \frac{1}{4n^2},$$

and this occurs at the point $(\frac{2m-1}{2n}, \frac{2m-1}{2n}) \in \mathcal{D}_{diag}$. Using calculus, one can show that the value of m that maximizes $\frac{nm-m^2+m}{n^2} - \frac{1}{4n^2}$ is $\frac{n+1}{2}$ if n is odd, and $\frac{n}{2}$ if n is even. If n is odd, one gets

$$D_\infty(\mathcal{D}_{diag}) = \sup_{\{\boldsymbol{x} \in \mathcal{X}\}} |F_n(\boldsymbol{x}) - F(\boldsymbol{x})| = \frac{1}{4} + \frac{1}{2n}$$

and if n is even,

$$D_\infty(\mathcal{D}_{diag}) = \frac{1}{4} + \frac{1}{2n} - \frac{1}{4n^2}.$$

However, notice that when n is odd, *any* design corresponding to a permutation in $\mathcal{P}$ and taking $\frac{n+1}{2}$ of its observations at points which are less than or equal to $(1/2, 1/2)$ (under componentwise ordering of vectors) will have support on a set with a discrepancy that is greater than or equal to that of $\mathcal{D}_{diag}$. To see this, simply notice this discrepancy must be at least equal to the value of $|F_n(\boldsymbol{x}) - F(\boldsymbol{x})|$ at $\boldsymbol{x} = (1/2, 1/2)$, which is equal to $D_\infty(\mathcal{D}_{diag})$. Likewise, if n is even, *any* design taking half of its observations at points less than or equal to $\left(\frac{n-1}{2n}, \frac{n-1}{2n}\right)$ will have support on a set with a discrepancy that is greater than or equal to that of $\mathcal{D}_{diag}$. Thus, $\mathcal{D}_{diag}$ is more uniform than any such design, even if such a design spreads points more evenly over $[0,1]^2$ than simply placing them along the diagonal.

Now consider the n point design,

$$\mathcal{D}_{antidiag} = \left\{\left(\frac{1}{2n}, \frac{2n-1}{2n}\right), \left(\frac{3}{2n}, \frac{2n-3}{2n}\right), \ldots, \left(\frac{2n-1}{2n}, \frac{1}{2n}\right)\right\}.$$

This design takes observations along the antidiagonal that runs from the point $(0, 1)$ to the point $(1, 0)$. For this design, we notice that when n is odd, $F_n(\boldsymbol{x}) = 0$ at $\boldsymbol{x} = \left(\frac{1}{2} - \epsilon, \frac{n+2}{2n} - \epsilon\right)$ and so, at this $\boldsymbol{x}$,

$$|F_n(\boldsymbol{x}) - F(\boldsymbol{x})| = \left(\frac{1}{2} - \epsilon\right)\left(\frac{n+2}{2n} - \epsilon\right).$$

In the limit as $\epsilon \to 0$,

$$|F_n(\boldsymbol{x}) - F(\boldsymbol{x})| \to \frac{1}{4} + \frac{1}{2n}.$$

One can show that this is, in fact, the supremum value of $|F_n(\boldsymbol{x}) - F(\boldsymbol{x})|$ for $\mathcal{D}_{antidiag}$, hence its discrepancy is $D_\infty(\mathcal{D}_{antidiag}) = \frac{1}{4} + \frac{1}{2n}$. Notice that $\frac{1}{4} + \frac{1}{2n}$ is also the value of $D_\infty(\mathcal{D}_{diag})$, so D_∞ considers $\mathcal{D}_{diag}$ and $\mathcal{D}_{antidiag}$ equally uniform when n is odd.

When n is even, by considering the point $\boldsymbol{x} = \left(\frac{n+1}{2n} - \epsilon, \frac{n+1}{2n} - \epsilon\right)$, one can show that in the limit as $\epsilon \to 0$,

$$|F_n(\boldsymbol{x}) - F(\boldsymbol{x})| \to \frac{1}{4} + \frac{1}{2n} + \frac{1}{4n^2}.$$

In this case, $D_\infty(\mathcal{D}_{antidiag})$ is at least as large as $\frac{1}{4} + \frac{1}{2n} + \frac{1}{4n^2}$. Notice that this quantity is larger than the discrepancy of $\mathcal{D}_{diag}$ when n is even, so in this case $\mathcal{D}_{diag}$ is a more uniform design than $\mathcal{D}_{antidiag}$. Most readers would consider both designs to be equally uniform. ■

This example shows that discrepancy, at least as measured by D_∞, may not adequately reflect our intuitive notion of what it means for points to be evenly spread over $\mathcal{X}$. Other measures of discrepancy may perform better. In view of Wiens (1991), uniform designs may be promising, but additional study of their properties in the context of computer experiments is needed. It should be noted that in Fang et al (2000), the design $\mathcal{D}_{diag}$ is eliminated from consideration because only matrices $\boldsymbol{\Pi}$ of rank d are considered, and the matrix $\boldsymbol{\Pi}$ corresponding to $\mathcal{D}_{diag}$ is of rank 1.

Constructing uniform designs is nontrivial. The Mathematics Department of Hong Kong Baptist University maintains a web site with information about uniform designs, including lists of publications about uniform designs and tables of uniform designs. The web site is located at www.math.hkbu.edu.hk/UniformDesign/.

JMP version 7 and later also generates uniform designs. JMP uses the centered L2 discrepancy measure of Hickernell (1998). To generate uniform designs, one must run the Space Filling Design command under the DOE menu. See the Help menu in JMP for details.

Figure 5.11 displays a 40 point uniform design and a 40 point maximin LHD for comparison purposes. Both were generated using the JMP software package.

5.8 Chapter Notes

5.8.1 Proof That T_L is Unbiased and of Theorem 5.1

We use the same notation as in Section 5.2.3. To compute $\mathrm{E}\{T_L\}$, we need to describe how the LH sample is constructed. For each i, divide the range $[a_i, b_i]$ of the i^{th} coordinate of $\boldsymbol{X}$ into n intervals of equal marginal probability $\frac{1}{n}$ under F. Sample once from each of these intervals and let these sample values be denoted $X_{i1}, X_{i2}, \ldots, X_{in}$. Form the $d \times n$ array

$$\begin{pmatrix} X_{11} & X_{12} & \ldots & X_{1n} \\ X_{21} & X_{22} & \ldots & X_{2n} \\ & & \vdots & \\ X_{d1} & X_{d2} & \ldots & X_{dn} \end{pmatrix}$$

and then randomly permute the elements in each row using independent permutations. The n columns of the resulting array are the LH sample. This is essentially the procedure for selecting a LH sample that was discussed in Section 5.2.1. Another way to select a LH sample is as follows. The Cartesian product of the d subintervals $[a_i, b_i]$ partitions $\mathcal{X}$ into n^d cells, each of probability $1/n^d$. Each of these n^d cells can be labeled by a set of d coordinates

$$\boldsymbol{m}_i = (m_{i1}, m_{i2}, \ldots, m_{id}),$$

where $1 \le i \le n^d$ and m_{ij} is a number between 1 and n corresponding to which of the n intervals of $[a_j, b_j]$ is represented in cell i. For example, suppose $n = 3$, $d = 2$, $[a_1, b_1] = [a_2, b_2] = [0, 1]$, and $F(\cdot)$ is uniform. We divide $[a_1, b_1]$ into the three intervals $[0, \frac{1}{3})$, $[\frac{1}{3}, \frac{2}{3})$, and $[\frac{2}{3}, 1]$. Similarly for $[a_2, b_2]$. In this case the cell $[\frac{1}{3}, \frac{2}{3}) \times [\frac{1}{3}, \frac{2}{3})$ would have cell coordinates (2, 2).

To obtain a LH sample, select a random sample of n of the n^d cells, say $\boldsymbol{m}_{i_1}, \boldsymbol{m}_{i_2}, \ldots, \boldsymbol{m}_{i_n}$, subject to the condition that for each j, the set $\{m_{i_\ell j}\}_{\ell=1}^n$ is a permutation of the integers $1, 2, \ldots, n$. We then randomly select a single point from each of these n cells. For a LH sample obtained in this manner, the density of $\boldsymbol{X}$, given $\boldsymbol{X} \in$ cell i, is

$$f(\boldsymbol{x} \mid \boldsymbol{X} \in \text{cell } i) = \begin{cases} \frac{1}{n^d} f(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \text{cell } i \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the distribution of the output $y(\boldsymbol{X})$ under LH sampling is

$$\begin{aligned} P(y(\boldsymbol{X}) \le y) &= \sum_{i=1}^{n^d} P(y(\boldsymbol{X}) \le y \mid \boldsymbol{X} \in \text{cell } i) P(\boldsymbol{X} \in \text{cell } i) \\ &= \sum_{i=1}^{n^d} \int_{\text{cell } i \text{ and } y(\boldsymbol{x}) \le y} n^d f(\boldsymbol{x}) \left(\frac{1}{n^d}\right) d\boldsymbol{x} \\ &= \int_{y(\boldsymbol{x}) \le y} f(\boldsymbol{x}) d\boldsymbol{x}, \end{aligned}$$

which is the same as for random sampling. Hence we have $\mathrm{E}\{T_L\} = \mu$.

To compute $\mathrm{Var}\{T_L\}$, we view our sampling as follows. First we select the $\boldsymbol{X}_i$ independently and randomly according to the distribution of F from each of the n^d cells. We next independently select our sample of n cells as described above, letting

$$W_i = \begin{cases} 1 \text{ if cell } i \text{ is in our sample} \\ 0 \text{ otherwise} \end{cases}$$

and

$$G_i = g(y(\boldsymbol{X}_i)).$$

Then

$$\begin{aligned} \mathrm{Var}\{T_L\} &= \mathrm{Var}\left\{ \frac{1}{n} \sum_{j=1}^{n} G_j \right\} \\ &= \frac{1}{n^2} \left[\sum_{i=1}^{n^d} \mathrm{Var}\{W_i G_i\} \right. \\ &\quad \left. + \sum_{i=1}^{n^d} \sum_{j=1, j \ne i}^{n^d} \mathrm{Cov}\left((W_i \times G_i), (W_j \times G_j) \right) \right]. \end{aligned}$$

To compute the variances and covariance on the right-hand side of this expression, we need to know some additional properties of the W_i. Using the fundamental rule that the probability of an event is the proportion of samples in which the event occurs, we find the following. First, $P(W_i = 1) = n/n^d = 1/n^{d-1}$ so W_i is Bernoulli with probability of success $1/n^{d-1}$. Second, if W_i and W_j correspond to cells having at least one common cell coordinate, then these two cells cannot both be selected, hence $\mathrm{E}\{(W_i W_j)\} = 0$. Third, if W_i and W_j correspond to cells having no cell coordinates in common, then

$$\mathrm{E}\{W_i \times W_j\} = P\{W_i = 1, W_j = 1\} = \frac{1}{n^{d-1}(n-1)^{d-1}}.$$

This follows from the fact that, taking order into account, there are $n^d(n-1)^d$ pairs of cells with no coordinates in common and in our sample of size n, there are $n(n-1)$ such pairs.

Using the fact that for two random variables Z and V, $\mathrm{Var}\{Z\} = \mathrm{E}\{\mathrm{Var}\{Z \mid V\}\} + \mathrm{Var}\{\mathrm{E}\{Z \mid V\}\}$, we have

$$\begin{aligned}\mathrm{Var}\{W_i \times G_i\} &= \mathrm{E}\{\mathrm{Var}\{W_i\, G_i \mid W_i\}\} + \mathrm{Var}\{\mathrm{E}\{W_i\, G_i \mid W_i\}\} \\ &= \mathrm{E}\{W_i^2\, \mathrm{Var}\{G_i \mid W_i\}\} + \mathrm{Var}\{W_i \mathrm{E}\{G_i \mid W_i\}\} \\ &= \mathrm{E}\{W_i^2\, \mathrm{Var}\{G_i\} + \mathrm{Var}\{W_i\, \mathrm{E}\{G_i\}\} \qquad (5.8.1) \\ &= \mathrm{E}\{W_i^2\}\mathrm{Var}\{G_i\} + \mathrm{E}^2\{G_i\}\mathrm{Var}\{W_i\},\end{aligned}$$

where in (5.8.1) above we use the fact that $\boldsymbol{X}_i$ (and hence G_i) and W_i are independent. Letting

$$\mu_i = \mathrm{E}\{g(y(\boldsymbol{X}_i))\} = \mathrm{E}\{g(y(\boldsymbol{X})) \mid \boldsymbol{X} \in \text{cell } i\}$$

and recalling that W_i is Bernoulli, we have

$$\begin{aligned}\sum_{i=1}^{n^d} \mathrm{Var}\{W_i\, G_i\} &= \sum_{i=1}^{n^d} \left[\mathrm{E}\{W_i^2\}\mathrm{Var}\{G_i\} + \mathrm{E}^2\{G_i\}\mathrm{Var}\{W_i\}\right] \\ &= \frac{1}{n^{d-1}} \sum_{i=1}^{n^d} \Big[E\{G_i - \mu_i\}^2 \\ &\quad + \frac{1}{n^{d-1}}(1 - \frac{1}{n^{d-1}})\mu_i^2\Big] \\ &= \frac{1}{n^{d-1}} \sum_{i=1}^{n^d} \Big[\int_{\text{cell } i} (g(y(\boldsymbol{x}))) - \mu + \mu - \mu_i)^2\ n^d\, f(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + \frac{1}{n^{d-1}}(1 - \frac{1}{n^{d-1}})\mu_i^2\Big] \\ &= n\, \mathrm{Var}\{y(\boldsymbol{X})\} - \frac{1}{n^{d-1}} \sum_{i=1}^{n^d} \left[(\mu - \mu_i)^2 + \frac{1}{n^{d-1}}(1 - \frac{1}{n^{d-1}})\mu_i^2\right].\end{aligned}$$

Because W_ℓ and $G_\ell = g(y((\boldsymbol{X}_\ell)))$ are independent, then for $i \neq j$,

$$\begin{aligned}\mathrm{Cov}\left((W_i \times G_i), (W_j \times G_j)\right) &= \mathrm{E}\{W_i\, G_i\, W_j\, G_j\} \\ &\quad - \mathrm{E}\{W_i\, G_i\}\, \mathrm{E}\left\{W_j\, G_j\right\} \\ &= \mathrm{E}\left\{W_i\, W_j\right\}\, \mathrm{E}\left\{G_i\, G_j\right\} \\ &\quad - \mathrm{E}\{W_i\}\, \mathrm{E}\{G_i\}\, \mathrm{E}\left\{W_j\right\}\, \mathrm{E}\left\{G_j\right\} \\ &= \mathrm{E}\left\{W_i\, W_j\right\}\, \mathrm{E}\{G_i\}\, \mathrm{E}\left\{G_j\right\} \\ &\quad - \frac{1}{n^{d-1}} \mathrm{E}\{G_i\}\, \frac{1}{n^{d-1}}\, \mathrm{E}\left\{G_j\right\} \\ &= \mathrm{E}\left\{W_i\, W_j\right\} \mu_i\, \mu_j - \frac{1}{n^{2d-2}}\, \mu_i\, \mu_j.\end{aligned}$$

Hence

$$\sum_{i=1}^{n^d}\sum_{j=1,j\neq i}^{n^d}\mathrm{Cov}\left(W_i G_i, W_j G_j\right) = \sum_{i=1}^{n^d}\sum_{j=1,j\neq i}^{n^d}\left[\mathrm{E}\left\{W_i W_j\right\}\mu_i\mu_j - \frac{1}{n^{2d-2}}\mu_i\mu_j\right].$$

Recall that $\mathrm{E}\{W_i W_j\} = 0$ if cells i and j have at least one common cell coordinate. Let R denote the $n^d(n-1)^d$ pairs of cells (with regards to order) having no cell coordinates in common. On this set we saw that

$$\mathrm{E}\left\{W_i W_j\right\} = \frac{1}{n^{d-1}(n-1)^{d-1}}$$

so we have

$$\begin{aligned}\mathrm{Var}\left\{\frac{1}{n}\sum_{j=1}^{n} G_j\right\} = \frac{1}{n^2}\Bigg[&n\,\mathrm{Var}\left\{g(y(\boldsymbol{X})\right\} - \frac{1}{n^{d-1}}\sum_{i=1}^{n^d}(\mu-\mu_i)^2\\ &+\frac{1}{n^{d-1}}\left(1-\frac{1}{n^{d-1}}\right)\sum_{i=1}^{n^d}\mu_i^2\\ &+\frac{1}{n^{d-1}(n-1)^{d-1}}\sum_{R}\mu_i\mu_j - \frac{1}{n^{2d-2}}\sum_{i=1}^{n^d}\sum_{j=1,j\neq i}^{n^d}\mu_i\mu_j\Bigg].\end{aligned}$$

Notice that

$$\begin{aligned}\sum_{i=1}^{n^d}\mu_i &= \sum_{i=1}^{n^d}\mathrm{E}\left\{g(y(\boldsymbol{X}))\mid \boldsymbol{X}\in\text{cell } i\right\}\\ &= \sum_{i=1}^{n^d}\int_{\text{cell } i} g(y(\boldsymbol{x}))n^d f(\boldsymbol{x})d\boldsymbol{x}\\ &= n^d\int_{\mathcal{X}} g(y(\boldsymbol{x}))f(\boldsymbol{x})d\boldsymbol{x} = n^d\mu.\end{aligned}$$

So

$$\begin{aligned}
\mathrm{Var}\left\{\frac{1}{n}\sum_{j=1}^{n} G_j\right\} &= \frac{1}{n}\mathrm{Var}\left\{g(y(\boldsymbol{X}))\right\} - \frac{1}{n^{d+1}}\sum_{i=1}^{n^d}\left(\mu^2 - 2\mu_i\mu + \mu_i^2\right) \\
&\quad + \left(\frac{1}{n^{d+1}} - \frac{1}{n^{2d}}\right)\sum_{i=1}^{n^d}\mu_i^2 \\
&\quad + \frac{1}{n^{d+1}(n-1)^{d-1}}\sum_{R}\mu_i\mu_j \\
&\quad - \frac{1}{n^{2d}}\sum_{i=1}^{n^d}\sum_{j=1,j\neq i}^{n^d}\mu_i\mu_j \\
&= \mathrm{Var}\left\{T_R\right\} + \frac{1}{n}\mu^2 - \frac{1}{n^{2d}}\left(\sum_{i=1}^{n^d}\mu_i\right)^2 \\
&\quad + \frac{1}{n^{d+1}(n-1)^{d-1}}\sum_{R}\mu_i\mu_j \\
&= \mathrm{Var}\left\{T_R\right\} - \frac{n-1}{n}\mu^2 \\
&\quad + \left(\frac{n-1}{n}\right)\left(\frac{1}{n^d(n-1)^d}\right)\left(\sum_{R}\mu_i\mu_j\right) \\
&= \mathrm{Var}\left\{T_R\right\} \\
&\quad - \left(\frac{n-1}{n}\right)\left(\frac{1}{n^d(n-1)^d}\right)\left(\sum_{R}\mu^2\right) \\
&\quad + \left(\frac{n-1}{n}\right)\left(\frac{1}{n^d(n-1)^d}\right)\left(\sum_{R}\mu_i\mu_j\right) \\
&= \mathrm{Var}\left\{T_R\right\} \\
&\quad + \left(\frac{n-1}{n}\right)\left(\frac{1}{n^d(n-1)^d}\right) \\
&\quad \times \sum_{R}(\mu_i - \mu)(\mu_j - \mu) \qquad (5.8.2) \\
&\leq \mathrm{Var}\left\{T_R\right\},
\end{aligned}$$

provided the last term in (5.8.2) is less than or equal to 0. Thus, whether LH sampling is superior to simple random sampling depends on the sign of this term, which in turn depends on the nature of g and f. Note also that LH sampling is superior to stratified random sampling with proportional sampling if

$$\left(\frac{n-1}{n}\right)\left(\frac{1}{n^d(n-1)^d}\right)\sum_{R}(\mu_i - \mu)(\mu_j - \mu) \;< -\frac{1}{n}\sum_{i=1}^{I} p_i(\mu - \mu_i)^2$$

McKay et al (1979) prove that under the assumptions of Theorem **??**, if $(y(x_1, \ldots, x_d)$ is monotonic in each of its arguments and $g(w)$ is a monotonic function of w), then $\sum_R (\mu_i - \mu)(\mu_j - \mu) \leq 0$. This completes the proof of Theorem **??**. □

5.8.2 The Use of LHDs in a Regression Setting

Owen (1992b) presents a multivariate extension of Theorem 5.3 and its application to computer experiments when fitting a regression to output data (rather the constant mean described in Section 5.2.3). The basis for the application is the following multivariate version of Theorem 5.3. The setting is as follows. Suppose that $\boldsymbol{X}$ has independent components with distribution functin $F(\cdot)$, $\boldsymbol{y}(\boldsymbol{X}) = (y_1(\boldsymbol{X}), \ldots, y_k(\boldsymbol{X}))^\top$, $\overline{\boldsymbol{Y}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}(\boldsymbol{X}_i)$ and $\boldsymbol{\mu} = \int_{\mathcal{X}} \boldsymbol{y}(\boldsymbol{x})\, d\boldsymbol{x}$.

Corollary 5.1. Let $r_\ell(\boldsymbol{x})$ be the residual from additivity for $y_\ell(\boldsymbol{x})$ and define

$$\sigma_{ij} = \int_{\mathcal{X}} r_i(\boldsymbol{x})\, r_j(\boldsymbol{x})\, dF(\boldsymbol{x}).$$

Let $\boldsymbol{\Sigma}$ be the $d \times d$ matrix whose (i, j) entry is σ_{ij}. Then $\sqrt{n}(\overline{\boldsymbol{Y}} - \boldsymbol{\mu})$ tends in distribution to $N_k(\boldsymbol{0}, \boldsymbol{\Sigma})$ as $n \to \infty$.

Let $\boldsymbol{Z}(\boldsymbol{x})$ be a vector valued function for which a linear model $\boldsymbol{Z}^\top(\boldsymbol{x})\boldsymbol{\beta}$ is an appropriate approximation to $\boldsymbol{Y}(\boldsymbol{x})$. The "population" least squares value of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta}_{\text{POP}} \equiv \left[\int_{\mathcal{X}} \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{Z}^T(\boldsymbol{x})\, dF(\boldsymbol{x})\right]^{-1} \int_{\mathcal{X}} \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{Y}(\boldsymbol{x})\, dF(\boldsymbol{x}).$$

Assuming $\int_{\mathcal{X}} \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{Z}^\top(\boldsymbol{x})\, dF(\boldsymbol{x})$ is known or easily computable (this would be the case for polynomial regression, for example), we can estimate $\boldsymbol{\beta}_{\text{POP}}$ by

$$\widehat{\boldsymbol{\beta}}_{\text{POP}} = \left[\int_{\mathcal{X}} \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{Z}^\top(\boldsymbol{x})\, dF(\boldsymbol{x})\right]^{-1} \frac{1}{n}\sum_{i=1}^n \boldsymbol{Z}(\boldsymbol{X}_i)Y(\boldsymbol{X}_i).$$

The variance of $\widehat{\boldsymbol{\beta}}_{\text{POP}}$ is of the "sandwich" form

$$\left[\int_{\mathcal{X}} \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{Z}^\top(\boldsymbol{x})\, dF(\boldsymbol{x})\right]^{-1} \boldsymbol{\Sigma} \left[\int_{\mathcal{X}} \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{Z}^\top(\boldsymbol{x})\, dF(\boldsymbol{x})\right]^{-1},$$

where $\boldsymbol{\Sigma}$ is defined in Corollary 5.1 above using the j^{th} component of $\boldsymbol{Z}(\boldsymbol{x})$ times $\boldsymbol{Y}(\boldsymbol{x})$ in place of $Y_j(\boldsymbol{x})$ in the definition of $r_j(\boldsymbol{x})$. Appealing to Theorem **??**, one might argue that to the extent that $\boldsymbol{Z}(\boldsymbol{x})\,\boldsymbol{Y}(\boldsymbol{x})$ is additive, the regression may be more accurately estimated from a LHD than from a design based on a simple random sample.

Owen (1992b) discusses some other estimators of $\boldsymbol{\beta}_{\text{POP}}$. The point is that when a linear model is likely to provide a good approximation to $y(\boldsymbol{x})$, using a LHD fol-

lowed by regression modeling is not an unreasonable way to conduct computer experiments.

5.8.3 Other Space-Filling Designs

The methods discussed in this chapter are not the only ones that generate space-filling designs. The literature on numerical integration contains numerous suggestions for constructing evenly-spaced designs. Niederreiter (1992) contains a wealth of information about such designs, including their mathematical properties.

One possibility is to choose points on a regularly spaced grid superimposed on the experimental region. For example, if the experimental region is $\mathcal{X} = [0, 1]^d$, the d-fold Cartesian product of the n point set

$$S = \left\{ \frac{1}{2n}, \frac{3}{2n}, \ldots, \frac{2n-1}{2n} \right\}$$

would be a grid consisting of n^d points. Grid designs consist of an array of evenly spaced points, but projections onto subspaces have many replicated points.

An improvement over grids is obtained by the method of good lattice points. Such designs are appealing in that they appear evenly spaced and in some cases have attractive properties in numerical integration. Niederreiter (1992) discusses these designs in more detail. Bates et al (1996) consider lattice designs in the context of computer experiments.

Nets form another class of designs that appear space-filling and which are popular in numerical integration. See (Niederreiter (1992)) and (Owen (1995)) for more details.

Because these designs are intended for use in numerical integration, they are generally used in situations where a large sample size is employed. Their properties tend to be for large numbers of observation and their small-sample behavior is not clear (and thus their usefulness in computer experiments in which the total number of observations is constrained to be small).

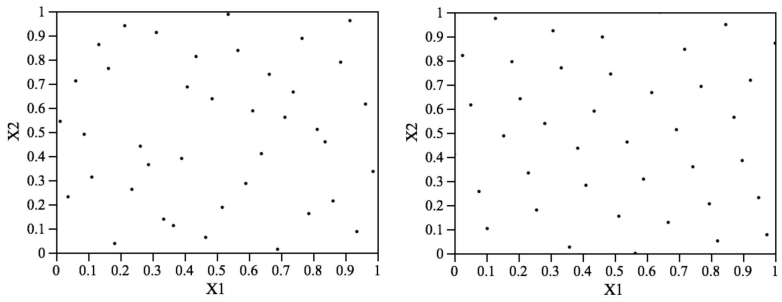


Fig. 5.11 Left Panel—a 40 point uniform design generated using the JMP software package; Right Panel—a 40 point maximin LHD generated using the JMP software package.

Chapter 7
Sensitivity Analysis and Variable Screening

7.1 Introduction

This chapter discusses sensitivity analysis and the related topic of variable screening. The set-up is as follows. A vector of inputs $\boldsymbol{x} = (x_1, \ldots, x_d)$ is given which potentially affect a "response" function $y(\boldsymbol{x}) = y(x_1, \ldots, x_d)$. Sensitivity analysis (SA) seeks to quantify how variation in $y(\boldsymbol{x})$ can be apportioned to the inputs x_1, $\ldots$, x_d *and to the interactions among these inputs*. Variable selection is more decision oriented in that it seeks to simply determine, for each input, whether that input is "active" or not. However, the two notions are related and variables screening procedures use some form of SA to assess the activity of each candidate input. Hence SA will be described first and then, using SA tools, two approaches to variable selection will be presented.

To fix ideas concerning SA, consider the function

$$y(x_1, x_2) = x_1 + x_2. \tag{7.1.1}$$

with domain $(x_1, x_2) \in (0, 1) \times (0, 2)$. One form of SA is based on examining the *local change* in $y(\boldsymbol{x})$ as x_1 or x_2 increases by a small amount starting from (x_1^0, x_2^0). This change can be determined from the partial derivatives of $y(\cdot)$ with respect to x_1 and x_2; in this example,

$$\frac{\partial y(x_1 x_2)}{\partial x_1} = 1 = \frac{\partial y(x_1 x_2)}{\partial x_2}, \tag{7.1.2}$$

so that we can assert that small changes in the inputs parallel to the x_1 or the x_2 axes *starting from any input* have the same effect on $y(\cdot)$.

A more *global assessment* of the sensitivity of $y(\boldsymbol{x})$ with respect to any component x_i, $i = 1, \ldots, d$, examines the change in $y(\boldsymbol{x})$ as x_i ranges over its domain for *fixed* values of the remaining inputs. In the case of (7.1.1), for fixed x_1^0 it is easy to see that the range of $y(x_1^0, x_2)$ as x_2 varies over $(0, 2)$, is $2 = \max_{x_2} y(x_1^0, x_2) - \min_{x_2} y(x_1^0, x_2) = y(x_1^0, 2) - y(x_1^0, 0)$ which is *twice* as large as

$1 = \max_{x_1} y(\cdot, x_2^0) - \min_{x_1} y(x_1^0, x_2)$, the range of $y(x_1, x_2^0)$ over x_1 for any fixed x_2^0. Thus this second method of assessing sensitivity concludes that $y(x_1, x_2)$ is twice as sensitive to x_2 as x_1.

This example illustrates two approaches that have been used to assess the influence of inputs on a given output. *Local sensitivity analysis* measures the change in the slope of the tangent to $y(\boldsymbol{x})$ at $\boldsymbol{x}$ in the direction of a given input axis j, fixing the remaining inputs. *Global sensitivity analysis* measures the change in $y(\boldsymbol{x})$ as one (or more inputs) vary over their entire *domain* when the remaining inputs are fixed. As the example above shows, the different criteria can lead to different conclusions about the sensitivity of $y(\boldsymbol{x})$ to its inputs. When it is determined that certain inputs have relatively *little* effect on the output, we can set these inputs to nominal values, and reduce the dimensionality of the problem allowing us to perform a more exhaustive investigation of a predictive model with a fixed budget for runs.

Sensitivity analysis is also useful for identifying interactions between variables. When interactions do not exist, the effect of any given input is the same regardless of the values of the other inputs. In this case, the relationship between the output and inputs is said to be additive and is readily understandable. When interactions exist, the effects of some inputs on the output will depend on the values of other inputs.

The remainder of this chapter will emphasize methods of quantifying the *global sensitivity analysis* of a code with respect to each of its inputs and then to estimating these sensitivity indices. It will also describe a companion method of visualizing the sensitivity of a code to each input based on *elementary effects*. An efficient class of designs called one-at-a-time designs will be introduced for estimating elementary effects.

For a comprehensive discussion of local and global sensitivity measures and their estimation based on training data, one should refer to the book length descriptions in Saltelli et al (2000) and Saltelli et al (2004).

7.2 Classical Approaches to Sensitivity Analysis

7.2.1 Sensitivity Analysis Based on Scatterplots and Correlations

Possibly the simplest approach to sensitivity analysis uses familiar graphical and numerical tools. A scatterplot of each input versus the output of the code provides a visual assessment of the marginal effect of each input on the output. The product moment correlations between each input and the output indicate the extent to which there is *linear association* between the outputs and the input. Scatterplots are generally more informative than correlations because nonlinear relationships can be seen in plots, whereas correlations only indicate the presence of straight-line relationships.

As an example, Figure 1.5 on page 11 plots the failure depth of pockets punched into sheet metal (the output) versus clearance and versus fillet radius, two character-

istics of the machine tool used to form the pockets. The scatterplot of failure depth versus clearance shows an increasing trend, suggesting that failure depth is sensitive to clearance. However, in the scatterplot of failure depth versus fillet radius, no trend appears to be present, suggesting that failure depth may not be sensitive to fillet radius.

One limitation of marginal scatterplots is that they do not allow assessment of possible interaction effects. Three graphical methods that can be used to explore two-factor interaction effects are: three-dimensional plots of the output versus pairs of inputs; two-dimensional plots that use different plotting symbols to represent the (possibly grouped) values of a second input; and a series of two-dimensional plots each of whose panels use only the data corresponding to a (possibly grouped) value of the second input. The latter are called "trellis plots." Graphical displays that allow one to investigate 3-way and higher interactions are possible but typically require some form of dynamic ability to morph the figure and experience in interpretation.

7.2.2 Sensitivity Analysis Based on Regression Modeling

Regression analysis provides another sensitivity analysis methodology that builds on familiar tools. The method below is most effective when the design is orthogonal or nearly orthogonal and a first-order linear model in the inputs $x_1, \ldots, x_d$ (nearly) explains the majority of the variability in the output.

The regression approach to sensitivity analysis first standardizes the output $y(\boldsymbol{x})$ and all the inputs $x_1, \ldots, x_d$. If n runs of the simulator code have been made, each variable is standardized by subtracting that variable's mean and dividing the difference by the sample standard deviation. For example, fix an input j, $1 \le j \le d$, and let $x_{1,j}, \ldots, x_{n,j}$ denote the values of this variable for the n runs. Let $\overline{x}_j$ denote the mean of $x_{1,j}, \ldots, x_{n,j}$ and s_j their standard deviation. The standardized value of $x_{i,j}$ is defined to be

$$x^{\star}_{i,j} = \frac{x_{i,j} - \overline{x}_j}{s_j}, \quad 1 \le i \le n.$$

In a similar fashion standardize the output values yielding $y^{\star}_i$, $1 \le i \le n$. Now fit the *first-order regression model*

$$y^{\star} = \beta^{\star}_0 + \beta^{\star}_1 x^{\star}_1 \cdots + \beta^{\star}_d x^{\star}_d \tag{7.2.1}$$

to the standardized variables. The regression coefficients in (7.2.1) are called the *standardized regression coefficients* (SRCs); $\beta^{\star}_j$ measures the change in $y^{\star}$ due to a unit standard deviation change in input j. Because all variables have been placed on a common scale, the magnitudes of the estimated SRCs indicate the relative sensitivity of the output to each input. The output is judged most sensitive to those inputs whose SRCs are largest in absolute value.

The validity of the method depends on the overall fit of the regression model, either as indicated by standard goodness-of-fit measures such as the coefficient of

determination R^2. If the overall fit is poor, the SRCs do not reflect the effect of the inputs on the output. In addition, regression-based methods are most effective when the input design is orthogonal or at least space-filling so that changes in the output due to one input can not be masked by changes in another.

Example 7.1. Recall that Subsection 1.7 described the failure depth for a computational model of the operation of punching symmetric rectangular pockets in automobile steel sheets. Table 7.1 lists the regression coefficients for model (7.2.1). This analysis is likely to be reasonable because the R^2 associated with the fitted model is 0.9273. The estimated regression coefficients suggest that the output is most sensitive to *Clearance* and then, equally so, to the two inputs *Fillet Radius* and *Punch Plan View Radius*. The other inputs are of lesser importance. ■

Input	Estimated $\beta_i^\star$ in (7.2.1)
Clearance	0.8705
Fillet Radius	0.2490
Punch Plan View Radius	0.2302
Width	0.0937
Length	0.0681
Lock Bead Distance	0.0171

Table 7.1 Estimated SRCs for the fitted standardized model (7.2.1).

Example 7.2. Subsection 1.3 described a computer simulator of the temporal evolution of a fire in an enclosed room. This example selected, as output, the time until the smoke plume of the fire reached five feet above the fire source. The inputs affecting this time were the room area, room height, heat loss fraction, and height of the fire source above the room. Figure 3.10 on page 85 shows the marginal relationship between the output and each input based on a 40 point Sobol´ design. From these plots, it appears that the output is most sensitive to room area, and not very sensitive to the remaining inputs. We perform a sensitivity analysis of the output

Input	Est. $\beta_i^\star$ in (7.2.1)
Heat Loss Frac.	0.1283
Fire Height	0.5347
Room Height	0.3426
Room Area	0.9066

Table 7.2 Estimated SRCs for the fitted standardized model (7.2.1).

function based on the 40 point training data for this example. Fitted model (7.2.1) has $R^2 = 0.98$ suggesting that the output is highly linear in the four inputs and the regression approach to sensitivity analysis is likely to be accurate. Table 7.2 lists the regression coefficients for this model. These values suggest that the single most

important input is *Room Area*, followed by *Fire Height*, *Room Height*, and lastly by *Heat Loss Fraction*. ■

There are a number of variants on regression-based models. Partial correlation coefficients (PCCs) between the output and the inputs can be used to assess sensitivity. PCCs measure the strength of the linear relationship between the output and a given input, after adjusting for any linear effects of the other inputs. The relative sizes of PCCs are used to assess the sensitivity of the output to the inputs.

As for SRCs, the same two circumstances will compromise the validity of PCCs. If the overall fit of the model is poor or there is a high degree of collinearity among the predictors PCCs need not provide accurate information about the sensitivity of the output to the inputs.

A third variant of the regression approach finds rank transforms of both the inputs and the outputs. The rank transformation is carried out as follows. Suppose that a variable has N values; assign rank 1 to the lowest values, rank 2 to the next lowest, and rank N to the largest. Use the average rank for ties. Then fit a *first-order regression model* to the transformed data. The estimated standardized regression coefficients or partial correlations are used to assess the sensitivity of the output to the inputs. Once again, if the overall fit of the first-order regression model is poor, the standardized regression coefficients or partial correlations do not adequately describe the effect of the inputs on the output and this analysis does not provide good information about sensitivities.

In practice, it has been observed that the regression model for the ranked transformed data often has higher R^2 values than that for the regression model based on the standardized data. This may be because the rank transformation removes certain nonlinearities present in the original data. Thus, when monotone (but nonlinear) trends are present, there are some advantages to conducting a sensitivity analysis using the rank transformed data. However, when one uses the rank transformed data, one must keep in mind that the resulting measures of sensitivity give us information on the sensitivity of the ranked transformed output to the rank transformed inputs, rather than on the original variables.

A method that takes explicit account of the statistical significance of the estimated regression coefficients is a Stepwise Regression Algorithm applied to the standardized inputs. For example, if a forward stepwise regression is used, the first variable entered would be considered the most influential input, the second variable entered would be considered the second most influential input, etc. As is usual in stepwise regression, one continues until the amount of variation explained by adding further variables is not considered meaningful according to some criterion selected by the user. Statistics such as the mean squared error, the F-statistic for testing whether the addition of another variable significantly improves the model, the coefficient of determination R^2, or the adjusted R^2 can be used to determine when to stop the stepwise regression. For more on stepwise regression, see any standard text on regression, for example Draper and Smith (1981).

Whether one uses the standardized or the rank transformed data, we get no information about possible interactions or on non-monotone effects of variables when we fit first-order models. If one has reason to believe that interactions are present,

or that the relation between the output and some of the inputs is nonlinear and non-monotone, these regression methods will not give reliable information about sensitivities. One may wish to consider fitting higher-order models such as a second-order response surface to the output. Such a model allows one to explore second-order (quadratic) effects of inputs and two-factor interaction (cross-product) effects. For more on response surface methods, see Box and Draper (1987).

7.3 Sensitivity Analysis Based on Elementary Effects

The Elementary Effects (EEs) of a function $y(\boldsymbol{x}) = y(x_1, \ldots, x_d)$ having d inputs measures the sensitivity of $y(\boldsymbol{x})$ to x_j by directly measuring the change in $y(\boldsymbol{x})$ when x_j alone is altered. From a geometric viewpoint, EEs are the slopes of secant lines parallel to each of the input axes. In symbols, given $j \in \{1, \ldots, d\}$, the j^{th} EE of $y(\boldsymbol{x})$ at distance Δ is

$$d_j(\boldsymbol{x}) = \frac{y(x_1, \ldots, x_{j-1}, x_j + \Delta, x_{j+1}, \ldots, x_d) - y(\boldsymbol{x})}{\Delta} . \tag{7.3.1}$$

So specifically, $d_j(\boldsymbol{x})$ is the slope of the secant line connecting $y(\boldsymbol{x})$ and $y(\boldsymbol{x} + \Delta \boldsymbol{e}_j)$ where $\boldsymbol{e}_j = (0, 0, \ldots, 1, 0, \ldots, 0)$ is the j^{th} unit vector. For "small" Δ, $d_j(\boldsymbol{x})$ is a numerical approximation to the j^{th} partial derivative of $y(\boldsymbol{x})$ with respect to x_j evaluated at $\boldsymbol{x}$ and hence is a local sensitivity measures. However, in most of the literature, EEs are evaluated for "large" Δ at a widely sampled set of inputs $\boldsymbol{x}$ and hence are global sensitivity measures measuring the (normalized) overall change in the output as each input moves parallel to its axis.

Example 7.3. To gain intuition about the interpretation of EEs, consider the following simple analytic "output" function

$$y(\boldsymbol{x}) = 1.0 + 1.5x_2 + 1.5x_3 + 0.6x_4 + 1.7x_4^2 + 0.7x_5 + 0.8x_6 + 0.5x_5 \times x_6, \tag{7.3.2}$$

of $d = 6$ inputs where $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$, and $x_j \in [0, 1]$ for $j = 1, \ldots, 6$. Notice that $y(\boldsymbol{x})$ is functionally independent of x_1, is linear in x_2 and x_3, is non-linear in x_4, and contains an interaction in x_5 and x_6. Note that *if the range of* x_3 where modified to be $[0, 2]$, then larger range of x_3 compared with x_2 would mean that any reasonable assessment of global sensitivity of $y(\boldsymbol{x})$ to the inputs should conclude that x_3 is more active x_2.

Straightforward algebra gives the value $y(\boldsymbol{x} + \Delta \boldsymbol{e}_j) - y(\boldsymbol{x})$, $j = 1, \ldots, 6$, and hence the EEs of $y(\boldsymbol{x})$ can be calculated exactly as

1. $d_1(\boldsymbol{x}) = 0$,
2. $d_2(\boldsymbol{x}) = 1.5 = d_3(\boldsymbol{x})$,
3. $d_4(\boldsymbol{x}) = +0.6 + 1.7\Delta + 3.4x_4$,
4. $d_5(\boldsymbol{x}) = +0.7 + 0.5x_6$, and $d_6(\boldsymbol{x}) = +0.8 + 0.5x_5$.

The EEs for this example are interpreted as follows. The EE of the totally inactive variable x_1 is zero because $y(\boldsymbol{x})$ is functionally independent of x_1. The EEs of the additive linear terms x_2 and x_3 are the *same* non-zero constant, 1.5, and hence (7.3.2) is judged to be equally sensitive to x_2 and x_3. In general, the EEs of additive linear terms are *local sensitivity measures*; from the global viewpoint which assesses the sensitivity of $y(\boldsymbol{x})$ to each input by the change in the output as the input moves over its range, (7.3.2) is more sensitive x_3 than x_2 because x_3 has a larger range than x_2. The EE of the quadratic term x_4 depends on *both* the starting x_4 and Δ; hence for *fixed* Δ $d_4(\boldsymbol{x})$ will vary with x_4 alone. Lastly, the EEs of the interacting x_5 and x_6 depend on both of these inputs. ■

EEs can be used as an exploratory data analysis tool, as follows. Suppose that each $d_j(\boldsymbol{x})$, $j = 1, \ldots, d$, has been computed for r vectors, say $\boldsymbol{x}_1^j, \ldots, \boldsymbol{x}_r^j$. Let $\overline{d_j}$ and S_j denote the sample mean and sample standard deviation, respectively, of $d_j(\boldsymbol{x}_1^j), \ldots, d_j(\boldsymbol{x}_r^j)$. Then as Morris (1991) states, an input x_j having

- a *small* $\overline{d_j}$ and *small* S_j is non-influential;
- a *large* $\overline{d_j}$ and *small* S_j has a strong linear effect on $y(\boldsymbol{x})$;
- a *large* S_j (and either a *large* or *small* $\overline{d_j}$) either has a non-linear effect in x_j or x_j has strong interactions with other inputs.

Example 7.3 [*Continued*] To illustrate, consider (7.3.2) in Example 7.3. Suppose that $y(\boldsymbol{x})$ is evaluated for each row of Table 7.3. Table 7.3 contains five *blocks* of six rows; each block begins with a row in bold-face font. The difference of the $y(\boldsymbol{x})$ values computed from consecutive pairs of rows provide one $d_j(\boldsymbol{x})$ value. Note that every successive pair of rows differs by $\pm\,|\Delta| = 0.3$ in one input location. In all, these output function evaluations provide $r = 5$ $d_j(\boldsymbol{x})$ evaluations for five different $\boldsymbol{x}$. The construction of the input table, "the design" of these runs, will be discussed in the subsequent paragraphs. Here, only the interpretation of the plot is considered.

Figure 7.1 plots the $\overline{d_j}$ and S_j values from the individual EEs computed in the previous paragraph. Because $S_1 = 0 = S_2 = S_3$, the plot shows that $d_1(\boldsymbol{x})$, $d_2(\boldsymbol{x})$, and $d_3(\boldsymbol{x})$ are each constant and have values 0.0, 1.5, and 1.5, respectively (as the theoretical calculations showed). Hence these $(\overline{d_j}, S_j)$ points are interpreted as saying that $y(\boldsymbol{x})$ is functionally independent of x_1, and contains an additive linear term in each of x_2 and x_3. Because $S_j > 0$ for $j = 4, 5$, and 6, the corresponding $d_j(\boldsymbol{x})$ values vary with $\boldsymbol{x}$. Hence one can conclude that either $y(\boldsymbol{x})$ is not linear in x_j or that x_j interacts with the other inputs. ■

How should one select runs in order to efficiently estimate EEs of $y(\boldsymbol{x})$? By definition, each $d_j(\boldsymbol{x})$ requires two function evaluations. Hence, at first impulse, a total of $2 \times r$ function evaluations would be required to estimate r EEs. Morris (1991) proposed a more efficient *one-at-time (OAT) sampling design* for estimating the EEs when the input region is rectangular. This method is particularly useful for providing a sensitivity analysis of an *expensive* black-box computer simulator. To introduce the method, consider the sequence of six runs of a function $y(\boldsymbol{x}) = y(x_1, x_2, x_3, x_4, x_5)$

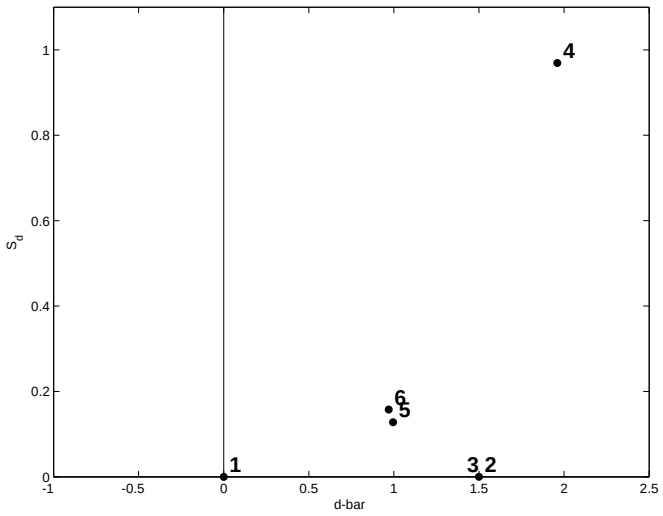


Fig. 7.1 Plot of $\overline{d_j}$ and S_j values for the EEs computed for function $y(\boldsymbol{x})$ in (7.3.2) using the design in Table 7.3

Run	x_1	x_2	x_3	x_4	x_5
1	0.8	0.7	1.0	0.7	0.7
2	0.5	0.7	1.0	0.7	0.7
3	0.5	1.0	1.0	0.7	0.7
4	0.5	1.0	0.7	0.7	0.7
5	0.5	1.0	0.7	1.0	0.7
6	0.5	1.0	0.7	1.0	0.4

having $d = 5$ inputs with input domain $[0, 1]^5$. The function $y(\boldsymbol{x})$ evaluated at Runs 1 and 2 can used to compute $d_1(0.8, 0.7, 1.0, 0.7, 0.7)$ for $\Delta = -0.3$. Similarly the differences of each consecutive pairs of rows provide, in order, estimates of $d_2(\boldsymbol{x})$, $d_3(\boldsymbol{x})$, $d_4(\boldsymbol{x})$, and $d_5(\boldsymbol{x})$ for different $\boldsymbol{x}$ but each using $|\Delta| = 0.3$. Such a set of rows is termed a *tour* of input space. For a given distance $|\Delta|$, OAT designs provide a set of $(d + 1) \times d$ tours with each tour starting at a randomly selected point of the input space and having successive rows that differ in a single input by $\pm\Delta$.

Each tour of the Morris (1991) OAT design is $(d + 1) \times d$ matrix whose rows are valid input vectors for $y(\cdot)$. The tour is determined a starting vector $\boldsymbol{x}$, a permutation of $(, 12, \ldots, d)$ that specifies the input to be modified in successive pairs of rows, by the choice of $\Delta > 0$, and by $d \times 1$ vector $(\pm 1, \ldots, \pm 1)$ of directional movements for Δ in the successive pairs of rows. For example the first tour in Table 7.3 uses starting vector $\boldsymbol{x} = (0.8, 0.7, 1.0, 0.7, 0.7)$, alters the inputs in succeeding rows in the order $(1, 2, 3, 4, 5)$ with $\Delta = 0.3$ and signs $(-1, +1, -1, +1, -1)$. Each row of the tour is an element of $[0, 1]^5$, which is the valid input region in this case. In examples where the design consisting of multiple tours, Morris (1991) selects the magnitude of Δ to be 30% of the common range of each variable, takes a random permutation

of $(1, \ldots, d)$, makes a random selection of the directional sign to be associated with each input, and selects the starting $\boldsymbol{x}$ *randomly* from a gridding of the input space that is made in such a way all rows of selected tour belong to the domain of $y(\boldsymbol{x})$.

A number of enhancements have been proposed to the basic Morris (1991) OAT design. Campolongo et al (2007) suggest ways of making OAT designs more space-filling. They propose selecting the desired number, r, of tours to maximize a heuristic distance criterion between all pairs of tours. Their R package `sensitivity` implements this criterion; it was used to construct the design in Table7.3. Pujol (2009) proposed a method of constructing OAT designs whose projections onto subsets of the input space are not collapsing. This is important if $y(\boldsymbol{x})$ depends only a subset of "active" inputs. For example, suppose that $y(\boldsymbol{x})$ depends (primarily) on x_j where $j \in \{1, 3, 5\}$ and other x_ℓ are "inactive." If multiple $\boldsymbol{x}$ inputs from the selected EE design have common x_j for $j \in \{1, 3, 5\}$, then $y(\boldsymbol{x})$ evaluations at these $\boldsymbol{x}$ would produce (essentially) the same output and hence little information about the input-output relationship. Campolongo et al (2011) introduced an OAT design that spreads starting points using Sobel´ sequence and differential Δ for each pair of input vectors. Finally Sun et al (2014) introduced an OAT design that can be be used for non-rectangular input regions and an alternative method to Campolongo et al (2007) for spreading the secant lines of the design over the input space.

7.4 Global Sensitivity Analysis Based on a Functional ANOVA Decomposition

Often, one of the first tasks in the analysis of simulator output $y(\boldsymbol{x})$ is the rough assessment of the sensitivity of $y(\boldsymbol{x})$ to each input x_j. In a combined physical system/simulator setting, the analogous question is the determination of the sensitivity of the *mean response* of the physical system to each input.

Sobol´ (1990), Welch et al (1992), and Sobol´ (1993) introduced plotting methods to make such an assessment. They introduce the use of main effect plots and joint effect plots. They also define various numerical "sensitivity" indices (SIs) to make such assessments. This section will define these effect plots and a functional ANOVA decomposition of the output $y(\boldsymbol{x})$ that is used to define "global SIs."

More formal variable screening methods have been developed for computer simulators by Linkletter et al (2006) and Moon et al (2012). The methodology in both papers assumes training data is available to construct a Gaussian Process emulator of the output at arbitrary input sites (with Gaussian correlation function). Linkletter et al (2006) use the (posterior distribution of the) estimated process correlation for each input to assess its impact on $y(\boldsymbol{x})$ while Moon et al (2012) calculate each inputs' "total effect" index for the same purpose.

To ease the notional burden, from this point on assume that $y(\boldsymbol{x})$ has a *hyperrectangular* input domain which is taken to be $[0, 1]^d$. If the input domain of $y(\boldsymbol{x})$ is $\prod_{j=1}^{d} [a_j, b_j]$, one should apply the methods below to the function

x_1	x_2	x_3	x_4	x_5	x_6
0.50	**0.60**	**0.50**	**0.40**	**0.50**	**0.35**
0.80	0.60	0.50	0.40	0.50	0.35
0.80	0.90	0.50	0.40	0.50	0.35
0.80	0.90	0.80	0.40	0.50	0.35
0.80	0.90	0.80	0.10	0.50	0.35
0.80	0.90	0.80	0.10	0.20	0.35
0.80	0.90	0.80	0.10	0.20	0.05
0.85	**0.3**	**0.9**	**0.65**	**0.3**	**0.40**
0.55	0.30	0.90	0.65	0.3	0.40
0.55	0.00	0.90	0.65	0.30	0.40
0.55	0.00	0.60	0.65	0.30	0.40
0.55	0.00	0.60	0.95	0.30	0.40
0.55	0.00	0.60	0.95	0.60	0.40
0.55	0.00	0.60	0.95	0.60	0.10
0.65	**0.00**	**0.35**	**0.75**	**0.45**	**0.60**
0.35	0.00	0.35	0.75	0.45	0.60
0.35	0.3	0.35	0.75	0.45	0.60
0.35	0.3	0.05	0.75	0.45	0.60
0.35	0.3	0.05	0.45	0.45	0.60
0.35	0.3	0.05	0.45	0.75	0.60
0.35	0.3	0.05	0.45	0.75	0.90
0.9	**0.05**	**0.35**	**0.05**	**0.4**	**1.0**
0.60	0.05	0.35	0.05	0.40	1.00
0.60	0.35	0.35	0.05	0.40	1.00
0.60	0.35	0.05	0.05	0.40	1.00
0.60	0.35	0.05	0.35	0.40	1.00
0.60	0.35	0.05	0.35	0.10	1.00
0.60	0.35	0.05	0.35	0.10	0.70
0.40	**0.35**	**0.60**	**0.00**	**0.35**	**0.60**
0.10	0.35	0.60	0.00	0.35	0.60
0.10	0.05	0.60	0.00	0.35	0.60
0.10	0.05	0.90	0.00	0.35	0.60
0.10	0.05	0.90	0.30	0.35	0.60
0.10	0.05	0.90	0.30	0.05	0.60
0.10	0.05	0.90	0.30	0.05	0.30

Table 7.3 An OAT design with $r = 5$ complete tours for a $d = 6$ input function with domain $[0, 1]^6$ and $|\Delta| = 0.30$.

$$y^\star(x_1, \ldots, x_d) = y(a_1 + x_1(b_1 - a_1), \ldots, a_d + x_d(b_d - a_d)\,.$$

When the input domain is *not* a hyperrectangle, there are several papers that consider analogs of the effect function definitions and sensitivity indices defined below; however this topic is an area of active research as of the writing of this book (Loeppky et al (2011)).

Section 7.4.1 introduces (uncentered) *main effect* and *joint effect* functions for a given $y(\boldsymbol{x})$, which are weighted $y(\boldsymbol{x})$ averages. Then Section 7.4.2 describes an ANOVA-like expansion of $y(\boldsymbol{x})$ in terms of centered and orthogonalized versions of the main and joint effect functions. Section 7.4.3 defines sensitivity indices for

individual inputs and groups of inputs in terms of the variability of these functional ANOVA components. Section 7.5 provides methods for estimating these plots and indices based on a set of $y(\boldsymbol{x})$ runs. The emphasis in Section 7.5 will be on methods that for cases of simulators having "expensive" code runs so that limited amounts of training data are available.

7.4.1 Main Effect and Joint Effect Functions

Denote the *overall mean* of $y(\cdot)$ by

$$y_0 \equiv \int_0^1 \cdots \int_0^1 y(x_1, \ldots, x_d) \prod_{j=1}^{d} dx_j. \tag{7.4.1}$$

More generally, the average (7.4.1) can be weighted, as for example, in scientific settings where is there is uncertainty in the input values that can be specified by independent input distributions. From this viewpoint, (7.4.1) and the development below assumes that the weight function corresponds to assuming that $\boldsymbol{X} = (X_1, \ldots, X_d)$ has independent and identically distributed (i.i.d.) $U(0, 1)$ component distributions. However, the definitions and analysis below can be easily generalized to allow weight functions that correspond to independent x_j components with arbitrary densities, say $g(\boldsymbol{x}) = \prod_{j=1}^{d} g_j(x_j)$, where $g_j(\cdot)$ is a density on $[0, 1]$. Thus the overall mean y_0 can be interpreted as the expectation $E[y(\boldsymbol{X})]$ where the weight function gives the joint distribution of $\boldsymbol{X}$.

Similarly, for any fixed $i \in \{1, \ldots, d\}$, the i^{th} main effect function associated with $y(\boldsymbol{x})$ is defined to be

$$u_i(x_i) = \int_0^1 \cdot \int_0^1 y(x_1, \ldots, x_d) \prod_{\ell \neq i} dx_\ell = E[y(\boldsymbol{X})|X_i = x_i]; \tag{7.4.2}$$

which is the average $y(\boldsymbol{x})$ value when x_i is fixed. The expectation notation uses the fact that the components of $\boldsymbol{X}$ are independent.

The idea of averaging $y(\boldsymbol{x})$ when a single input is fixed can be extended to fixing multiple inputs. Select a nonempty subset Q of $\{1, \ldots d\}$ for which $\overline{Q} = Q \setminus \{1, \ldots d\}$ is also non-empty (so that the integral below averages over at least one variable). Let $\boldsymbol{x}_Q$ denote the vector of components x_i with $i \in Q$ in some linear order. Define the average value of $y(x_1, \ldots, x_d)$ when $\boldsymbol{x}_Q$ is *fixed* to be

$$u_Q(\boldsymbol{x}_Q) = \int_0^1 \cdots \int_0^1 y(x_1, \ldots, x_d) \prod_{i \notin Q} dx_i = E[y(\boldsymbol{X})|\boldsymbol{X}_Q = \boldsymbol{x}_Q] \ .$$

For completeness, set

$$u_{12\ldots d}(x_1, \ldots, x_d) \equiv y(x_1, \ldots, x_d)$$

(when $Q = \{1, \ldots d\}$).

The function $u_Q(\boldsymbol{x}_Q)$ is called the *joint effect function* of $y(\boldsymbol{x})$ with respect to x_Q. Joint effect functions are also called *uncentered* effect functions to make explicit the fact that their average over any single x_i, $i \in Q$ (or collection of x_i for any proper subset of i in Q) need not be zero. However, the mean of any $u_Q(\boldsymbol{X}_Q)$, $Q \subset \{1, \ldots, d\}$, with respect to *all* $\boldsymbol{X}_Q$ is y_0, i.e.,

$$E\left[u_Q(\boldsymbol{X}_Q)\right] = \int_0^1 \cdots \int_0^1 u_Q(\boldsymbol{x}_Q)\, d\boldsymbol{x}_Q = y_0\,. \tag{7.4.3}$$

ANOVA-type centered versions of the $u_Q(\boldsymbol{x}_Q)$ will be considered in the next subsection.

Example 7.4. Suppose $y(x_1, x_2) = 2x_1 + x_2$ is defined on $[0, 1]^2$. Then the overall mean and u_Q effect functions are

$$\begin{aligned} y_0 &= \int_0^1 \int_0^1 (2x_1 + x_2) dx_2 dx_1 = 1.5, \\ u_1(x_1) &= \int_0^1 (2x_1 + x_2) dx_2 = 0.5 + 2x_1, \\ u_2(x_2) &= \int_0^1 (2x_1 + x_2) dx_1 = 1.0 + x_2, \quad \text{and} \\ u_{12}(x_1, x_2) &= y(x_1, x_2) = 2x_1 + x_2. \end{aligned}$$

Illustrating the fact (7.4.3) that y_0 is the mean of every $u_Q(\boldsymbol{X}_Q)$ with respect to $\boldsymbol{X}_Q$, it is simple to calculate that

$$\int_0^1 u_1(x_1)\, dx_1 = \int_0^1 u_2(x_1)\, dx_2 = \int_0^1 u_{11}(x_{11})\, dx_1\, dx_2 = 1.5.$$

Plots of the main effect functions, shown in Figure 7.2, provide accurate information of how this simple function behaves in x_1 and x_2. ■

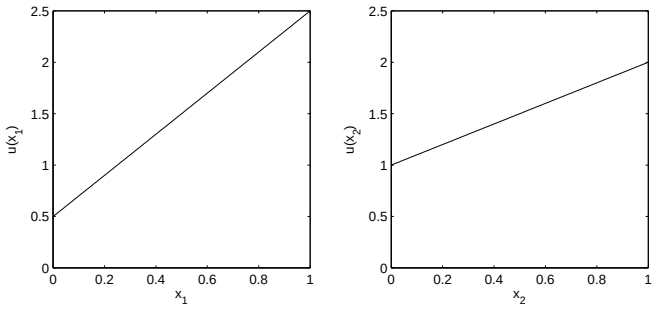


Fig. 7.2 Plots of the main effect functions $u_1(x_1)$ and $u_2(x_2)$ for $y(x_1, x_2) = 2x_1 + x_2$.

Example 7.5. The so-called g-function with d inputs is defined to be

$$y(x_1, \ldots, x_d) = \prod_{i=1}^{d} \frac{|4x_i - 2| + c_i}{1 + c_i} \tag{7.4.4}$$

where $\boldsymbol{x} \in [0,1]^d$ and $c = (c_1, \ldots, c_d)$ has non-negative components (Saltelli and Sobol´ (1995)).

Note that $y(\boldsymbol{x})$ is a product of functions of each input and does not involve stand-alone "linear" terms in any inputs. As the definition of the effect function states, and this example is meant to emphasize, $u_i(x_i)$ contains the contributions of every x_i component no matter whether they appear as a stand-alone term or as part of an interaction.

The value of

$$q(x) = \frac{|4x - 2| + c}{1 + c}, \tag{7.4.5}$$

over $x \in [0,1]$ forms a pair of line segments that are symmetric about $x = 1/2$, with minimum value of $c/(1+c) = q(1/2)$ and maximum value of $(2+c)/(1+c) = q(0) = q(1)$. Thus it is straightforward to calculate that

$$\int_0^1 \frac{|4x - 2| + c}{1 + c} dx = 1.0 \tag{7.4.6}$$

for every $c \geq 0$ because this integral is the sum of the areas of two identical trapezoids (triangles when $c = 0$). The parameter c determines how "active" x is in the associated $q(x)$. Figure 7.3 illustrates the level of activity of $q(x)$ on c; three $q(x)$ with $c \in \{0, 5, 25\}$, are plotted. Clearly, the smaller the value of c, the more "active" is x.

Returning to the g-function with arbitrary numbers of inputs, (7.4.4), and arbitrary vector of parameters $\boldsymbol{c} = (c_1, \ldots, c_d) \geq 0$, the main and joint effect functions of $y(\boldsymbol{x})$ are simple to calculate using (7.4.6). The overall mean is

$$y_0 = \int_0^1 \cdots \int_0^1 y(x_1, \ldots, x_d) \prod_{\ell=1}^{d} dx_\ell = \prod_{\ell=1}^{d} \int_0^1 \frac{|4x_i - 2| + c}{1 + c} dx_\ell = 1.0 \, .$$

The i^{th} main effect function is

$$u_i(x_i) = \frac{|4x_i - 2| + c_i}{1 + c_i} \times 1 = \frac{|4x_i - 2| + c_i}{1 + c_i} \quad i = 1, \ldots, d \, .$$

Thus the main effect plots of individual inputs have essentially the symmetric form shown in Figure 7.3.

In a similar way, given a nonempty subset Q of $\{1, \ldots d\}$,

$$u_Q(x_Q) = \prod_{i \in Q} \frac{|4x_i - 2| + c_i}{1 + c_i} \, . \tag{7.4.7}$$

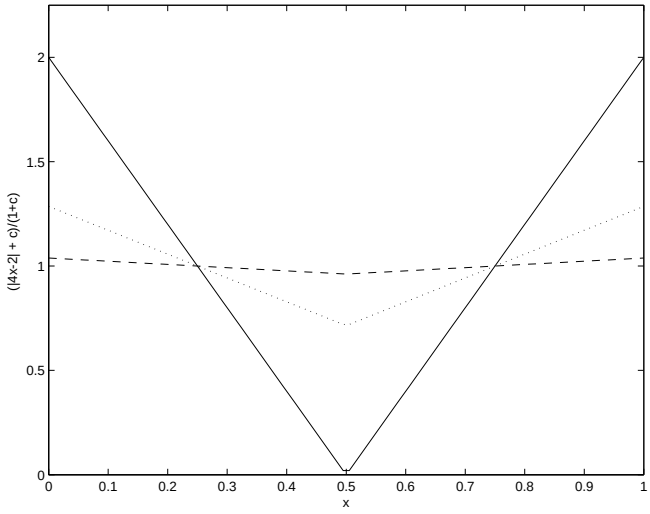


Fig. 7.3 $q(x) \equiv \frac{|4x-2|+c}{1+c}$ for $c = 0$ (solid line), $c = 5$ (dashed line), and $c = 25$ (dotted line).

For example,

$$u_{12}(x_1, x_2) = \frac{|4x_1 - 2| + c_1}{1 + c_1} \times \frac{|4x_2 - 2| + c_2}{1 + c_2} \,. \tag{7.4.8}$$

which is plotted in Figure 7.4 for $c_1 = 0.25$ and $c_2 = 10.0$. Clearly this function shows that x_1, the input associated with the smaller c_i, is more active than x_2. Visualizations of higher-order effect functions, while more difficult to display effec-

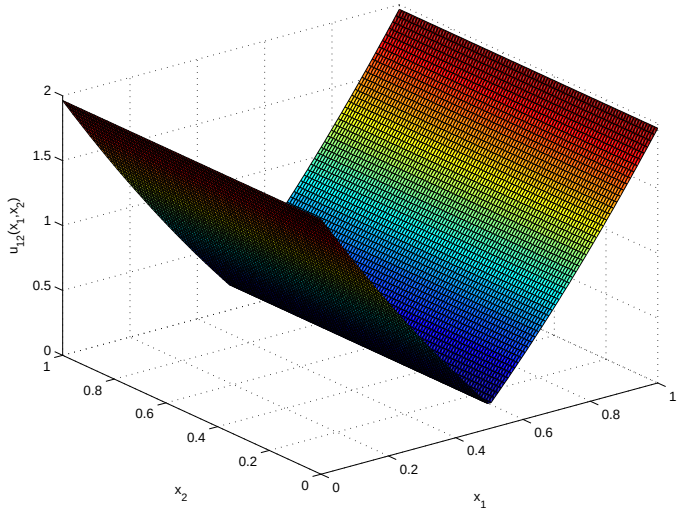


Fig. 7.4 Joint effect function (7.4.8) for $c_1 = 0.25$ and $c_2 = 10.0$.

tively, can also be made. ■

In general, plots of the main effect functions $(x_i, u_i(x_i))$ and, for pairs of inputs (x_i, x_j), the joint effect functions $\big((x_i, x_j), u_i(x_i, x_j)\big)$ can be used to provide a rough visual understanding of the change in the averaged $y(\boldsymbol{x})$ with respect to each single input or pairs of inputs. Section 7.5 will describe methods of estimating the $u_\varrho(\boldsymbol{x}_\varrho)$ based a set of training data obtained from the output function.

7.4.2 Functional ANOVA Decomposition

The uncentered $u_\varrho(\boldsymbol{x}_\varrho)$ describe average changes in $y(\boldsymbol{x})$; $u_\varrho(\boldsymbol{x}_\varrho)$ values on the same scale and in the same range as $y(\boldsymbol{x})$. The sensitivity indices that we shall define shortly will be based on the *variance* of these $u_\varrho(\boldsymbol{x}_\varrho)$ where the variance is with respect to a uniform distribution of the x_i inputs, $i = 1, \ldots d$. Viewed with this goal in mind, the (uncentered) main and joint effect functions have an important defect that limits their useful for constructing sensitivity indices. When viewed as functions of random $\boldsymbol{X}_i$ inputs, different effect functions are, in general, correlated. For example, if X_1 and X_2 are independent $U(0, 1)$ random variables, $cov(u_1(X_1), u_2(X_2))$ need not equal 0.

Thus Sobol´ (1990), and Sobol´ (1993) advocated the use of a functional ANVOA-like decomposition of $y(\boldsymbol{x})$ that modifies the joint effect functions to produce uncorrelated (and zero mean) versions of the $u_\varrho(\boldsymbol{x}_\varrho)$ (see also Hoeffding (1948)). The modified functions will be used to define "sensitivity" indices.

Specifically, Sobol´ (1993) proposed use of the (unique) decomposition of $y(\boldsymbol{x})$,

$$y(\boldsymbol{x}) = y_0 + \sum_{i=1}^{d} y_i(x_i) + \sum_{1 \le i < j \le d} y_{ij}(x_i, x_j) + \ldots + y_{1,2,\ldots,d}(x_1, \ldots, x_d) \tag{7.4.9}$$

that results in terms having zero means and such that any pair of these functions are orthogonal. The mean and orthogonality properties will be defined once formulas are given that define the components of (7.4.9). These terms are called *corrected (mean or joint) effect functions* because of the fact that they (turn out to) have mean zero.

To define the component functions in (7.4.9), first fix i, $1 \le i \le d$, and set

$$y_i(x_i) = u_i(x_i) - \ y_0 = \int_0^1 \cdots \int_0^1 y(\boldsymbol{x}) \prod\nolimits_{\ell \ne i} d\boldsymbol{x}_\ell \ - \ y_0 \tag{7.4.10}$$

to be the centered main effect function of input x_i. Similarly, for any fixed (i, j), $1 \le i < j \le d$, define

$$y_{ij}(x_i, x_j) = u_{ij}(x_i, x_j) - y_i(x_i) - y_j(x_j) - y_0$$
$$= \int_0^1 \cdots \int_0^1 y(\boldsymbol{x}) \prod_{\ell \neq i,j} d\boldsymbol{x}_\ell - y_i(x_i) - y_j(x_j) - y_0 \qquad (7.4.11)$$

to be the centered interaction effect function of inputs x_i and x_j. Higher-order interaction terms are defined in a recursive manner; if Q is a non-empty subset of $\{1, \ldots d\}$,

$$y_Q(\boldsymbol{x}_Q) = u_Q(\boldsymbol{x}_Q) - \sum_E y_E(\boldsymbol{x}_E) - y_0 \qquad (7.4.12)$$

where the sum over all non-empty proper subsets E of Q; $E \subset Q$ is proper provided $E \neq Q$. For example, if $y(\boldsymbol{x})$ has three or more arguments,

$$y_{123}(x_1, x_2, x_3) = u_{123}(x_1, x_2, x_3) - y_{12}(x_1, x_2) - y_{13}(x_1, x_3) - y_{23}(x_2, x_3)$$
$$-y_1(x_1) - y_2(x_2) - y_3(x_3) - y_0$$

In particular,

$$y_{1,2,\ldots,d}(x_1, x_2, \ldots, x_d) = u_{1,2,\ldots,d}(x_1, x_2, \ldots, x_d) - \sum_E y_E(x_E) - y_0$$
$$= y(x_1, x_2, \ldots, x_d) - \sum_E y_E(x_E) - y_0$$

so that (7.4.9) holds.

The centered effect functions in (7.4.9) have two properties that make them extremely useful for defining sensitivity indices. First, each has zero mean in the sense that for any $(i_1, \ldots, i_s)$ and any $i_k \in \{i_1, \ldots, i_s\}$,

$$E\left[y_{i_1,\ldots,i_s}(x_{i_1}, \ldots, X_{i_k}, \ldots, x_{is})\right] = \int_0^1 y_{i_1,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s})\, dx_{i_k} = 0. \qquad (7.4.13)$$

Second, the centered effect functions are *orthogonal* meaning that for any $(i_1, \ldots, i_s) \neq (j_1, \ldots, j_t)$,

$$Cov\left(y_{i_1,\ldots,i_s}(X_{i_1}, \ldots, X_{i_s}), y_{j_1,\ldots,j_t}(X_{j_1}, \ldots, X_{j_t})\right) =$$
$$\int_0^1 \cdots \int_0^1 y_{i_1,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s}) \times y_{j_1,\ldots,j_t}(x_{j_1}, \ldots, x_{j_t}) \prod_\ell dx_\ell = 0 \qquad (7.4.14)$$

where the product in (7.4.14) is over all $\ell \in \{i_1, \ldots, i_s\} \cup \{j_1, \ldots, j_t\}$ (see Section 7.7).

Example 7.4 [*Continued*] Using y_0, and the $u_Q(\cdot)$ effect functions calculated previously, we have

$$y_1(x_1) = u_1(x_1) - y_0 = 0.5 + 2x_1 - 1.5 = -1 + 2x_1$$
$$y_2(x_2) = u_2(x_2) - y_0 = 1.0 + x_2 - 1.5 = -0.5 + x_2$$
$$y_{12}(x_1, x_2) = u_{12}(x_1, x_2) - y_1(x_1) - y_2(x_2) - y_0 = 0.$$

The function $y_{12}(x_1, x_2) = 0$ suggests the lack of interaction between x_1 and x_2. It is straightforward to verify the properties (7.4.13) and (7.4.14) for this example because

$$\begin{aligned} E\left[y_1(X_1)\right] &= \int_0^1 (-1 + 2x_1)dx_1 = 0 \\ E\left[y_2(X_2)\right] &= \int_0^1 (-0.5 + x_2)dx_2 = 0 \\ E\left[y_{12}(X_1, x_2)\right] &= 0 = E\{y_{12}(x_1, X_2)\} \end{aligned} \tag{7.4.15}$$

and, for example,

$$Cov(y_1(X_1), y_{12}(X_1, X_2)) = 0. \qquad \blacksquare$$

Example 7.5 [*Continued*] Recalling formula (7.4.7), the first two centered effect functions are

$$\begin{aligned} y_i(x_i) = u_i(x_i) - y_0 &= \frac{|4x_i - 2| + c_i}{1 + c_i} - 1 \\ &= \frac{|4x_i - 2| - 1.0}{1 + c_i}, \end{aligned}$$

for $1 \le i \le d$, and

$$\begin{aligned} y_{ij}(x_i, x_j) &= u_{ij}(x_i, x_j) - y_i(x_i) - y_j(x_j) - y_0 \\ &= \frac{|4x_i - 2| + c_i}{1 + c_i} \times \frac{|4x_j - 2| + c_j}{1 + c_j} - \frac{|4x_i - 2| - 1.0}{1 + c_i} - \frac{|4x_j - 2| - 1.0}{1 + c_j} - 1, \end{aligned}$$

for $1 \le i < j \le d$. $\blacksquare$

The corrected effects will be used to partition the variance of $y(\boldsymbol{X})$ into components of variance that are used to *define* global sensitivity indices for an arbitrary set of input variables. Recall that $X_1, \ldots, X_d$ are independent and identically distributed $U(0, 1)$ random variables and

$$y_0 = E\left[y(\boldsymbol{X})\right] .$$

Define the *total variance* V of $y(\boldsymbol{x})$ to be

$$v = E\left[(y(\boldsymbol{X}) - y_0)^2\right] = E\left[y^2(\boldsymbol{X})\right] - y_0^2.$$

Recalling that for any subset $Q \subset \{1, \ldots, d\}$, $y_Q(\boldsymbol{X}_Q)$ has mean zero let

$$v_Q = Var(y_Q(\boldsymbol{X}_Q)) = E\left[y_Q^2(\boldsymbol{X}_Q)\right]$$

denote the variance of the term $y_Q(\boldsymbol{X}_Q)$ in (7.4.9). Using (7.4.9), we calculate

$$
\begin{aligned}
v &= E\left[(y(\boldsymbol{X}) - y_0)^2\right] \\
&= E\left[\left(\sum_{i=1}^{d} y_i(X_i) + \sum_{i<j} y_{ij}(X_i, X_j) + \ldots + y_{1,2,\ldots,d}(X_1, \ldots, X_d)\right)^2\right] \\
&= \sum_{i=1}^{d} E\left[y_i^2(X_i)\right] + \sum_{i<j} E\left[y_{ij}^2(X_i, X_j)\right] + \cdots + E\left[y_{1,2,\ldots,d}^2(X_1, \ldots, X_d)\right] \\
&\quad + \sum E\left[y_E(\boldsymbol{X}_E) y_{E^\star}(\boldsymbol{X}_{E^\star})\right] \qquad (7.4.16)
\end{aligned}
$$

where the sum in (7.4.16) is over all nonempty subsets E and $E^\star$ of $\{1, \ldots, d\}$ for which $E \neq E^\star$, and thus

$$
\begin{aligned}
v &= \sum_{i=1}^{d} E\left[y_i^2(X_i)\right] + \sum_{i<j} E\left[y_{ij}^2(X_i, X_j)\right] + \cdots \\
&\quad + E\left[y_{1,2,\ldots,d}^2(X_1, \ldots, X_d)\right] \qquad (7.4.17) \\
&= \sum_{i=1}^{d} v_i + \sum_{i<j} v_{ij} + \cdots + v_{1,2,\ldots,d} \qquad (7.4.18)
\end{aligned}
$$

with (7.4.17) holding because the components of (7.4.9) are orthogonal, and (7.4.18) because each term of the Soboĺ decomposition has zero mean.

The functional decomposition (7.4.9) can, in a more formal way, be modified to result in the classical ANOVA decomposition of a model with d quantitative factors. Suppose that, instead of $\boldsymbol{X}$ taking a uniform distribution over $[0, 1]^d$, the input factor space is regarded as the *discrete* set of points $\{0, \frac{1}{n-1}, \ldots, \frac{n-2}{n-1}, 1\}^d$ with a *discrete uniform distribution* over these n values. This would arise, for example, if the inputs formed an n^d factorial with n levels for each factor that are coded 0, $\frac{1}{n-1}$, $\ldots$, $\frac{n-2}{n-1}$, 1. Replacing each *integral* over $[0, 1]$ that defines a term in (7.4.9) by an *average* over the n discrete values, y_0 becomes $\overline{y}$, the overall mean of all the $y(\cdot)$, the $\{y_i\}_i$ become the usual ANOVA estimates of main effects in a complete factorial, the $\{y_{ij}\}_{ij}$ become the usual ANOVA estimates of two-factor interactions in a complete factorial, and so on. Finally, it is clear that v in (7.4.18), is the mean corrected sum of squares of all the $y(\cdot)$, v_i is the sum of squares for the i^{th} factor and so forth. Thus, the decomposition (7.4.18) is the usual ANOVA decomposition into sums of squares for main effects, and higher-way interactions.

7.4.3 Global Sensitivity Indices

For any subset $Q \subset \{1, \ldots, d\}$, define the sensitivity index (SI) of $y(\boldsymbol{x})$ with respect the set of inputs x_i, $i \in Q$, to be

$$S_Q = \frac{v_Q}{v}.$$

In particular S_i, corresponding to $Q = \{i\}$, is called the *first-order or main effect sensitivity index* of input x_i, $i = 1, \ldots, d$; S_i measures the proportion of the variation V that is due to input x_i. For $i < j$, S_{ij} is called the *second-order sensitivity index*; S_{ij} measures the proportion of V that is due to the joint effects of the inputs x_i and x_j. Higher-order sensitivity indices are defined analogously. By construction, the sensitivity indices satisfy

$$\sum_{i=1}^{d} S_i + \sum_{1 \le i < j \le d} S_{ij} + \cdots + S_{1,2,\ldots,d} = 1.$$

We illustrate these definitions and the interpretations of SIs with examples.

Example 7.4 [*Continued*] Recalling $y_1(x_1)$, $y_2(x_2)$, and $y_{12}(x_1, x_2)$ calculated previously, we calculate that

$$\begin{aligned} v &= Var(y(X_1, X_2)) = Var(2X_1 + X_2) = 4/12 + 1/12 = 5/12 \\ v_1 &= Var(y_1(X_1)) = Var(-1 + 2X_1) = 4/12 \\ v_2 &= Var(y_2(X_2)) = Var(-0.5 + X_2) = 1/12 \\ v_{12} &= Var(y_{12}(X_1, X_2)) = Var(0) = 0 \end{aligned}$$

so that $v = v_1 + v_2 + v_{12}$ and

$$S_1 = \frac{4/12}{1/12} = 4.0,\ S_2 = \frac{1/12}{1/12} = 1.0,\ \text{and } S_{12} = 0.0.$$

The interpretation of these values coincides with our intuition about $y(x_1, x_2)$: x_1 is more important than x_2 while there is no interaction between x_1 and x_2. The only deviation from our intuition is that, based on the functional relationship, the reader might have assessed that x_1 was *twice* as important x_2. ■

Before considering other examples, we use the S_Q sensitivity indices to define the so-called *total sensitivity index* (TSI) of $y(\boldsymbol{x})$ with respect to a given input x_i; T_i is meant to include interactions of x_i with all other inputs. The *total sensitivity of input i* is defined to be *sum* of all the sensitivity indices involving the i^{th} input; in symbols,

$$T_i = S_i + \sum_{j>i} S_{ij} + \sum_{j<i} S_{ji} + \cdots + S_{1,2,\ldots,d}\ . \tag{7.4.19}$$

For example, when $d = 3$,

$$T_1 = S_1 + S_{12} + S_{13} + S_{123}. \tag{7.4.20}$$

By construction, $T_i \ge S_i$, $i = 1, \ldots, d$ and the difference $T_i - S_i$ measures the influence of x_i due to its interactions with other variables. In principle, the calculation of the set of T_i requires that one determine a total of $\sum_{i=1}^{d} \binom{d}{i}$ variances, v_Q. But there is at least one method of making this computation more efficient which we describe next.

For arbitrary $Q \subset \{1, \ldots, d\}$, let

$$v_Q^u = Var\left(u_Q(\boldsymbol{X}_Q)\right) = Var\left(E\left\{y(\boldsymbol{X})|\boldsymbol{X}_Q\right\}\right) \tag{7.4.21}$$

to be the variance of the uncorrected effect. The quantity v_Q^u can be interpreted as the average reduction in uncertainty in $y(\boldsymbol{x})$ when $\boldsymbol{x}_Q$ is fixed because

$$v_Q^u = Var\left(y(\boldsymbol{X})\right) - E\left\{Var\left(y(\boldsymbol{X})|\boldsymbol{X}_Q\right)\right\} .$$

Consider two special cases of the uncorrected effect function variances. From (7.4.10), the variance of the uncorrected effect function $u_i(x_i)$ of the input x_i is

$$v_i^u = Var\left(y_i(X_i) + y_0\right) = v_i. \tag{7.4.22}$$

Thus the main effect sensitivity index of input x_i, S_i, can be computed from

$$S_i = \frac{v_i^u}{v}. \tag{7.4.23}$$

Using (7.4.11), and the orthogonality property (7.4.14), the variance of the uncorrected effect $u(x_i, x_j)$ is

$$v_{ij}^u = Var\Big(y_i(X_i) + y_j(X_j) + y_{ij}(X_i, X_j) + y_0\Big) = v_i + v_j + v_{ij}. \tag{7.4.24}$$

Equation (7.4.24) contains both the variance of the main effects and the variance of the interaction effect of inputs x_i and x_j. Thus $v_Q^u \neq v_Q$ when Q contains more than one input.

Equation (7.4.24) can be extended to arbitrary Q. We illustrate the usefulness of this expression by developing a formula for T_i where i is a fixed integer, $i = 1, \ldots, d$, using $Q = \{1, \ldots, d\} - \{i\}$. Let $\boldsymbol{X}_{-i}$ denote the vector of all components of $\boldsymbol{X}$ *except* X_i. Then

$$\begin{aligned}
v_{-i}^u &= Var\left(u_{-i}(\boldsymbol{X}_{-i})\right) \\
&= Var\big(y_{1,2,\ldots,i-1,i+1,\ldots,d}\left(\boldsymbol{X}_{-i}\right) + \cdots + y_1\left(\boldsymbol{X}_1\right) + y_2\left(\boldsymbol{X}_2\right) + \cdots \\
&\quad + y_{i-1}\left(\boldsymbol{X}_{i-1}\right) + y_{i+1}\left(\boldsymbol{X}_{i+1}\right) + \cdots + y_d\left(\boldsymbol{X}_d\right) + y_0 \big) \\
&= Var\left(\sum_{Q:\, i \notin Q} y_Q(\boldsymbol{X}_Q)\right) \\
&= v_{1,2,\ldots,i-1,i+1,\ldots,d} + \cdots + v_d + \cdots + v_{i+1} + v_{i-1} + \cdots + v_1 \\
&= \sum_{Q:\, i \notin Q} v_Q \, .
\end{aligned} \tag{7.4.25}$$

Equation (7.4.25) is the sum of all v_Q components *not involving* the subscript i in the variance decomposition (7.4.18). Thus $v - v_{-i}^u$ is the sum of all v_Q components that *do involve* the input x_i. Hence T_i can be expressed as

$$T_i = \frac{v - v^u_{-i}}{v}. \tag{7.4.26}$$

Thus if one is interested in estimating only the $2d$ *main effect* and *total effect* SIs, $\{S_i\}_i$ and $\{T_i\}_i$, for $i = 1, \ldots, d$, only $2d$ uncorrected effect variances (7.4.21) need be determined, rather than $\sum_{i=1}^{d} \binom{d}{i}$ variance terms used in their definition.

Example 7.5 [*Continued*] Recall the g-function (7.4.4) with d arguments,

$$y(\boldsymbol{x}) = \prod_{i=1}^{d} \frac{|4x_i - 2| + c_i}{1 + c_i}\ . \tag{7.4.27}$$

We calculate its associated S_i and T_i values. We will use the fact that if $X \sim U(0, 1)$, then

$$\begin{aligned} Var\,(|4X - 2|) &= 16 Var\,(|X - 1/2|) \\ &= 16\left\{E[(X - 1/2)^2] - (E[|X - 1/2|])^2\right\} = 1/3\ . \end{aligned}$$

Hence the total variance of $y(\boldsymbol{x})$ is

$$\begin{aligned} v &= Var\,(y(\boldsymbol{X})) \\ &= Var\left(\prod_{\ell=1}^{d} \frac{|4X_\ell - 2| + c_\ell}{1 + c_\ell}\right) \\ &= E\left\{\prod_{\ell=1}^{d}\left(\frac{|4X_\ell - 2| + c_\ell}{1 + c_\ell}\right)^2\right\} - 1.0 \\ &= \prod_{\ell=1}^{d} E\left\{\left(\frac{|4X_\ell - 2| + c_\ell}{1 + c_\ell}\right)^2\right\} - 1.0 \\ &= \prod_{\ell=1}^{d}\left[Var\left(\frac{|4X_\ell - 2| + c_\ell}{1 + c_\ell}\right) + 1\right] - 1 \\ &= \prod_{\ell=1}^{d}\left[\frac{1}{(1 + c_\ell)^2} Var(|4X_\ell - 2|) + 1\right] - 1 \\ &= \prod_{\ell=1}^{d}\left(\frac{1}{3(1 + c_\ell)^2} + 1\right) - 1\ . \end{aligned} \tag{7.4.28}$$

For $i = 1, \ldots, d$, the numerator of S_i is the variance of the first-order effect function $y_i(X_i)$

$$v_i = Var\,(y_i(X_i)) = Var\left(\frac{|4X_i - 2| + c_i}{1 + c_i}\right) = \frac{1}{3(1 + c_i)^2} \tag{7.4.29}$$

and hence

$$S_i = v_i/v = (7.4.29)/(7.4.28)\,. \tag{7.4.30}$$

In a similar fashion, for fixed $i = 1, \ldots, d$, the uncorrected effect function

$$u_{-i}(\boldsymbol{x}_{-i}) = \prod_{\ell \neq i} \frac{|4X_\ell - 2| + c_\ell}{1 + c_\ell}$$

has variance

$$\begin{aligned} v_{-i}^u &= Var\left(u_{-i}(\boldsymbol{X}_{-i})\right) \\ &= Var\left(\prod_{\ell \neq i} \frac{|4X_\ell - 2| + c_\ell}{1 + c_\ell}\right) \\ &= \prod_{\ell \neq i} \left(\frac{1}{3(1 + c_\ell)^2} + 1\right) - 1 \end{aligned}$$

following algebra similar to that used to derive v. Hence, after some simplification,

$$T_i = \frac{v - v_i^u}{v} = \frac{\left(\frac{1}{1+3(1+c_i)^2}\right) \prod_{\ell=1}^d \left(\frac{1}{3(1+c_\ell)^2} + 1\right)}{\prod_{\ell=1}^d \left(\frac{1}{3(1+c_\ell)^2} + 1\right) - 1} . \tag{7.4.31}$$

As a specific example, consider the $d = 2$ case illustrated in Figure 7.4 where $c_1 = 0.25$ and $c_2 = 10.0$. Calculation of (7.4.30) and (7.4.31) give the values in Table 7.4.

i	S_i	T_i
1	0.9846	0.9873
2	0.0127	0.0154

Table 7.4 Main effect and total effect sensitivity indices for the function (7.4.27) when $d = 2$.

The S_i and T_i are interpreted as saying that (1) x_1 is far more active input than x_2 and (2) there is virtually no interaction between x_1 and x_2 because $S_{12} = T_1 - S_1 = T_2 - S_2 = 0.0027$. Figure 7.4, the joint main effect function of x_1 and x_2, essentially shows the function $y(x_1, x_2)$ for this $d = 2$ example. This plot verifies the interpretations (1) and (2) are correct. For each $x_2^0 \in [0, 1]$, $\{y(x_1, x_2^0) : 0 \leq x_1 \leq 1\}$ has a v-shaped profile that is independent of x_2^0; for each $x_1^0 \in [0, 1]$, $\{y(x_1^0, x_2) : 0 \leq x_2 \leq 1\}$ is a horizontal line with height depending on x_1^0. ■

Example 7.6. This section concludes with an example that shows both the strength and weakness of trying to summarize the behavior of a potentially complicated function by several real numbers. Consider

$$y(x_1, x_2, x_3) = (x_1 + 1)cos(\pi x_2) + 0x_3 = cos(\pi x_2) + x_1 cos(\pi x_2) + 0x_3$$

defined on $[0, 1]^3$. The formula shows that $y(\boldsymbol{x})$ has a term depending only on x_2, an x_1 by x_2 "interaction" term, and does not depend on x_3. A plot of (x_1, x_2) versus

$y(x_1, x_2, 0.5)$, which is the same for any x_3, is shown in Figure 7.5. Any reasonable measure of the sensitivity of $y(\cdot)$ to the inputs should show that x_3 has *zero* influence on the function while both x_1 and x_2 are influential.

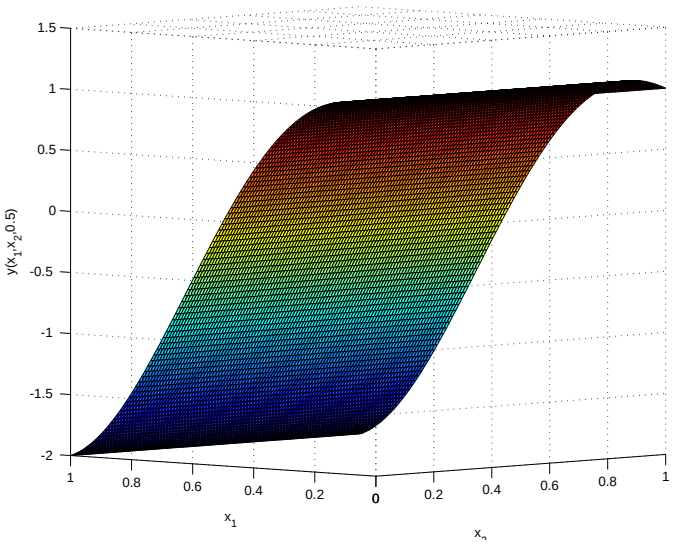


Fig. 7.5 The function $y(x_1, x_2, 0.5) = (x_1 + 1)cos(\pi x_2) + 0x_3$ versus x_1 and x_2.

Using the facts that

$$\int_0^1 \cos(\pi x)\, dx = 0 \quad \text{and} \quad \int_0^1 \cos^2(\pi x)\, dx = \frac{1}{2},$$

it is straightforward to compute that the overall mean is

$$y_0 = \int_0^1 \int_0^1 \int_0^1 (x_1 + 1)\cos(\pi x_2) dx_1\, dx_2\, dx_3 = 0$$

and the uncentered effect functions are

$$\begin{aligned}
u_1(x_1) &= \int_0^1 \int_0^1 (x_1 + 1)\cos(\pi x_2) dx_2\, dx_3 = 0 \\
u_2(x_2) &= \int_0^1 \int_0^1 (x_1 + 1)\cos(\pi x_2) dx_1\, dx_3 = \frac{3}{2}\cos(\pi x_2) \\
u_3(x_3) &= \int_0^1 \int_0^1 (x_1 + 1)\cos(\pi x_2) dx_1\, dx_2 = 0
\end{aligned}$$

which are also the centered effects, $y_1(x_1)$, $y_2(x_2)$, $y_3(x_3)$, respectively, because $y_0 = 0$. That $y_3(x_3) = 0$ is expected while $y_1(x_1) = 0$ may be unexpected. However, in

this artificial example, for each fixed $(x_1, x_3) = (x_1^0, x_3^0)$, the function $y(x_1^0, x_2, x_3^0)$ is symmetric about $x_2 = 1/2$ and is constant with integral zero with respect to x_2. Indeed, any function with constant average $y(\boldsymbol{x}_s, \boldsymbol{x}_{-s})$ over the inputs $\boldsymbol{x}_{-s}$ would also have constant mean $u(\boldsymbol{x}_s)$ with respect to the inputs $\boldsymbol{x}_s$.

Returning to the specifics of this example, the variance of $y(X_1, X_2, X_3)$ is

$$\begin{aligned} v &= Var\left((X_1 + 1)\cos(\pi X_2)\right) \\ &= E\left[(X_1 + 1)^2 \cos^2(\pi X_2)\right] = E\left[(X_1 + 1)^2\right] E\left[(\cos^2(\pi X_2)\right] = \frac{7}{6} \end{aligned}$$

which gives the main effect sensitivities

$$S_1 = \frac{Var(u_1(X_1))}{v} = 0 = S_3$$

while

$$S_2 = \frac{Var(\frac{3}{2}\cos(\pi X_2))}{v} = \frac{27}{28} \approx 0.964\,.$$

The zero main effect for x_1 is, perhaps, unexpected. It is due to the fact that the integral of $y(x_1, x_2)$ over the x_2-term is zero; any other function with a centered interaction term would also have $S_1 = 0$, e.g., $y = x_1(x_2 - 0.5)$. The large value of 0.964 for the main effect of x_2 may also not be consistent with the readers' intuition; this large value illustrates again that S_i depends on *every* x_i term that is part of the $y(\boldsymbol{x})$ formula, *not merely additive terms* $\beta \times x_i$.

To continue the example, we compute the total effects for each input using the formula (7.4.31). First, note that

$$u_{-1}(\boldsymbol{x}_{-1}) = u_{23}(x_2, x_3) = \int_0^1 y(x_1, x_2, x_3)dx_1 = 1.5\ \cos(\pi x_2)$$

and similarly

$$u_{-2}(\boldsymbol{x}_{-2}) = u_{13}(x_1, x_3) = 0 \ \text{ and } \ u_{-3}(\boldsymbol{x}_{-3}) = u_{12}(x_1, x_2) = (x_1 + 1)\cos(\pi x_2)$$

so

$$v_{-1}^u = Var(1.5\ \cos(\pi x_2)) = 9/8, \quad v_{-2}^u = 0, \ \text{ and } \ v_{-3}^u = v = 7/6$$

yielding

$$T_1 = \frac{v - v_{-1}^u}{v} = \frac{6}{7}\left(\frac{7}{6} - \frac{9}{8}\right) = \frac{1}{28} \approx 0.036, \ \ T_2 = 1, \ \text{ and } \ T_3 = 0.$$

The result that $T_3 = 0$ implies that $S_{13} = 0 = S_{23} = S_{123}$ as one expects from the functional form of $y(\boldsymbol{x})$. Indeed the remaining interaction must be $S_{12} = T_1 - S_1 == 1/28$ from (7.4.20). This small value for the S_{12} interaction may, again, not

be consistent with the reader's intuition but shows that once the main effect functions are subtracted from $u_{12}(x_1, x_2)$, there is very little variability in $y_{12}(x_1, x_2)$.

Indeed, it is interesting to note that for this example the variances of the centered and uncentered functions $y_{12}(x_1, x_2)$ and $u_{12}(x_1, x_2)$, respectively, can be quite different. In this case calculation gives

$$y_{12}(x_1, x_2) = (x_1 - 0.5)\cos(\pi x_2)$$

so that $v_{12} = Var(y_{12}(X_1, X_2)) = 1/24 \ll 7/6 = Var(u_{12}(X_1, X_2)) = v^u_{12}$. In general, the 2-*d* sensitivity index for inputs x_i and x_j, S_{ij}, subtracts the associated main effect functions which can greatly reduce the variance of the averaged function values. ■

7.5 Estimating Effect Plots and Global Sensitivity Indices

This section will describe how quadrature, empirical (plug-in) Bayesian, and fully Bayesian methods can be used to estimate effect plots and *main effect* and *total effect* sensitivity indices based on training data, $(\boldsymbol{x}_i, y(\boldsymbol{x}_i))$, $i = 1, \ldots, n$. With one exception, these methods assume that $y(\boldsymbol{x})$ can be modeled as a realization of a (non-stationary) Gaussian process

$$Y(\boldsymbol{x}) = \sum_{k_1=0}^{m_{k_1}} \cdots \sum_{k_d=0}^{m_{k_d}} \beta_{k_1 \ldots k_d} \prod_{j=1}^{d} x_j^{k_j} \; + \; Z(\boldsymbol{x}) \equiv \sum_{\ell=1}^{p} \beta_\ell f_\ell(\boldsymbol{x}) \; + \; Z(\boldsymbol{x}) \tag{7.5.1}$$

which of the regression plus stationary process form, where the $\{\beta_{k_1 \ldots k_d}\}_{k_1 \ldots k_d}$ are unknown regression coefficients, the powers $k_1, \ldots k_d$ are specified, and $Z(\boldsymbol{x})$ is a stationary Gaussian process with separable parametric correlation function $R(\cdot)$, i.e.,

$$Cov\left(Z(\boldsymbol{x}_r), Z(\boldsymbol{x}_s)\right) = \prod_{j=1}^{d} R(x_{rj} - x_{sj} | \, \boldsymbol{\psi}_j) \tag{7.5.2}$$

where $\boldsymbol{\psi}_j$ is the unknown parameter associated with the j^{th} input, possibly a vector. For example, the methods described below can be applied to both the Gaussian correlation function,

$$R_G(h \,|\, \psi) = \exp\left[-\psi \, h^2\right] \qquad \psi > 0 \,, \tag{7.5.3}$$

and the cubic correlation function

$$R_C(h|\,\psi) = \begin{cases} 1 - 6\left(\frac{h}{\psi}\right)^2 + 6\left(\frac{|h|}{\psi}\right)^3, & |h| \le \frac{\psi}{2} \;; \\ 2\left(1 - \frac{|h|}{\psi_j}\right)^3, & \frac{\psi}{2} \le |h| \le \psi \;; \\ 0, & \psi < |h| \,, \end{cases} \tag{7.5.4}$$

where $\psi > 0$. Separable Bohman correlation functions can also be implemented in this approach and, in priniple the Matern and power exponential functions.

7.5.1 Estimated Effect Plots

Given output function $y(\boldsymbol{x}) = y(x_1, \ldots, x_d)$, recall that the main effect plot for input x_q, $q \in \{1, \ldots, d\}$, is the plot of $(x_q, u_q(x_q))$ where

$$\begin{aligned} u_q(x_q) &= \int_0^1 \cdots \int_0^1 y(x_1, \ldots, x_d) \prod_{\ell \neq q} dx_\ell \\ &= \int_0^1 \cdots \int_0^1 y(x_1, \ldots, x_d)\, d\boldsymbol{x}_{-j} = E_x \left[y(\boldsymbol{X}) | X_i = x_q \right] \end{aligned} \tag{7.5.5}$$

is the average value of $y(\boldsymbol{x})$ when the q^{th} input is fixed and the averaging is over all possible values for inputs x_j, $j \neq q$. The alternate notation $d\boldsymbol{x}_{-j}$ indicates integration over all inputs *except* x_j and the expectation notation emphasizes that $\boldsymbol{X}$ can also be thought of as random vector. More generally, one can examine changes in the average value of $y(\boldsymbol{x})$ when two or more inputs are fixed, e.g., joint effect plots are 3d plots of $(x_q, x_v, u_{qv}(x_q, x_v))$ where

$$u_{qv}(x_q, x_v) = \int_0^1 \cdots \int_0^1 y(x_1, \ldots, x_d) \prod_{\ell \neq q,v} dx_\ell$$

Two methods of estimating $u_q(x_q)$ based on training data will be described. The first is a *quadrature-based* estimation and the second is a *Bayesian* method. Both methods can be extended to estimate the joint- or higher-effect function $u_{qv}(x_q, x_v)$.

A naive quadrature estimator of $u_q(x_q)$ uses the definition of the integral

$$\widehat{u}_q(x_q) = \sum_{\Delta^\star} \widehat{y}(x_1^{star}, x_{q-1}^{star}, x_q, x_{q+1}^{star}, x_d^{star})\ Vol(\Delta^\star)$$

where $\widehat{y}(\boldsymbol{x})$ is a predictor of $y(\boldsymbol{x})$ and the sum is over a set of disjoint hyperrectanges $\Delta^\star$ that partition the $(x_1, x_{q-1}, x_{q+1}, x_d)$ domain and $(x_1^{star}, x_{q-1}^{star}, x_{q+1}^{star}, x_d^{star}) \in \Delta^\star$. Here $\widehat{y}(\boldsymbol{x})$ can be *any predictor* of $y(\boldsymbol{x})$, be it based on regression, neural net or those described Chapter 3.

A more sophisticated quadrature method is possible when the outputs $y(\boldsymbol{x})$ can be modeled as a realization of the GP (7.5.1) with *separable* correlation function (7.5.2). In this case, recall that an EBLUP of $y(\boldsymbol{x})$ based on estimated correlation parameter $(\widehat{\psi}_1, \ldots, \widehat{\psi}_d)$ has the form

$$\widehat{y}(\boldsymbol{x}) = d_0(\boldsymbol{x}) + \sum_{i=1}^{n} d_i \prod_{\ell=1}^{d} R(x_\ell - x_{i\ell} \mid \widehat{\psi}_\ell) \tag{7.5.6}$$

where

$$d_0(\boldsymbol{x}) = \sum_{k_1=0}^{m_{k_1}} \cdots \sum_{k_d=0}^{m_{k_d}} \widehat{\beta}_{k_1 \ldots k_d} \prod_{j=1}^{d} x_j^{k_j}$$

when viewed as a function of the input $\boldsymbol{x}$, where the elements of $\widehat{\boldsymbol{\beta}}$ are obtained from the weighted least squares estimator of $\boldsymbol{\beta}$ that is based on the GP model (7.5.1). In this case the

$$\begin{aligned}
\widehat{u}_q(x_q) &= \int_0^1 \cdots \int_0^1 \left(\sum_{k_1=0}^{m_{k_1}} \cdots \sum_{k_d=0}^{m_{k_d}} \widehat{\beta}_{k_1 \ldots k_d} \prod_{j=1}^{d} x_j^{k_j} + \sum_{i=1}^{n} d_i \prod_{j=1}^{d} R(x_j - x_{ij} \mid \widehat{\psi}_\ell) \right) \prod_{j \neq q} dx_j \\
&= \sum_{k_1=0}^{m_{k_1}} \cdots \sum_{k_d=0}^{m_{k_d}} \beta_{k_1 \ldots k_d} \, x_q^{k_j} \prod_{j \neq q} \left(k_j + 1\right)^{-1} \\
&\quad + \sum_{i=1}^{n} d_i \, R(x_q - x_{iq} | \, \widehat{\psi}_q) \prod_{j \neq q} \int_0^1 R(x_j - x_{ij} | \, \widehat{\psi}_j) \, dx_\ell. \qquad (7.5.7)
\end{aligned}$$

In some cases the one-dimensional integrals in (7.5.7) will have closed form expressions. For example, for the Gaussian correlation function (2.4.6),

$$\begin{aligned}
\int_0^1 R\left(x_j - x_{ij} | \, \widehat{\psi}_j\right) \, dx_j &= \int_0^1 exp\{-\widehat{\psi}_j (x_j - x_{ij})^2\} \, dx_j \\
&= \frac{\sqrt{2\pi}}{\sqrt{\psi}_j} \left\{ \Phi\left(\psi_j (1 - x_{ij})\right) - \Phi\left(\psi_j (0 - x_{ij})\right) \right\}
\end{aligned}$$

when expressed in terms of the standard normal cumulative distribution function.

A second method of estimating the main effect function $u_q(x_q)$ is Bayesian (or empirical/plug-in Bayesian if one uses plug-in estimates of model parameters instead of assessing them by draws from their posterior distribution given the data) (see Oakley (2009), Moon (2010), Svenson (2011)). The idea of the method is to *replace* $y(\cdot)$ by the GP process $Y(\cdot)$ in the defining integral (7.5.5) of $u_q(x_q)$. Under mild conditions,

$$U_q(x_q) = \int_0^1 \cdots \int_0^1 Y(x_1, \ldots, x_d) \prod_{j \neq q} dx_j = E_x \left[Y(\boldsymbol{X}) \mid X_q = x_q \right]$$

is also a Gaussian process. Intuitively the integral is also GP because $U_q(\boldsymbol{x}_q)$ is approximately a linear combination of $Y(\boldsymbol{x})$ values and, for multivariate normal random variables, this linear combinations of multivariate normal random variables have a multivariate normal distribution (see Yaglom (1962) or Adler (1990)). Formally the mean, variance, and covariance of $U_q(x_q)$ can be obtained by interchanging appropriate integrals as follows. The mean of $U_q(x_q)$ is

$$
\begin{aligned}
E_P\left[U_q(x_q)\right] &= E_P\left[E_x[Y(\boldsymbol{X}) \mid X_q = x_q]\right] \\
&= E_x\left[E_P[Y(\boldsymbol{X})\,] \mid X_q = x_q\right] \\
&= E_x\left[\beta_0\,] \mid X_q = x_q\right] = \beta_0, \qquad (7.5.8)
\end{aligned}
$$

the covariance function is

$$
\begin{aligned}
Cov_p[U_q(\boldsymbol{x}_q), U_q(\boldsymbol{x}_q^\star)] &= Cov_p\left[\int\cdots\int Y(\boldsymbol{x})\, d\boldsymbol{x}_{-q}, \int\cdots\int Y(\boldsymbol{x}^\star)\, d\boldsymbol{x}_{-q}^\star\right], \\
&= \sigma_Z^2 \int\cdots\int R(\boldsymbol{x}, \boldsymbol{x}^\star \mid \boldsymbol{\psi})\, d\boldsymbol{x}_{-q}\, d\boldsymbol{x}_{-q}^\star \\
&= \sigma_Z^2 \prod_{j\neq q}\left[\int_0^1 \int_0^1 R(x_j, x_j^\star \mid \psi_j)\, dx_j\, dx_j^\star\right] R(x_q, x_q^\star \mid \psi_q) \qquad (7.5.9)
\end{aligned}
$$

with special case, the U_q process variance

$$
\sigma_U^2 = Cov_p[U_q(\boldsymbol{x}_q), U_q(\boldsymbol{x}_q)] = \sigma_Z^2 \prod_{j\neq q}\left[\int_0^1 \int_0^1 R(x_j, x_j^\star \mid \psi_j)\; dx_j dx_j^\star\right] \times 1 \quad (7.5.10)
$$

The double integrals in (7.5.9) and (7.5.10) can be evaluated in close-form for the Gaussian and several other correlation families.

Returning to the statement of the Bayesian estimator of $u_q(x_q)$, suppose that output has been collected at the (training data) inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and $\boldsymbol{Y}^n = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n)$ is the model for the observed outputs. A Bayes estimator of $u_q(x_q)$ is the posterior mean of the $U_q(\boldsymbol{x}_q)$ given the training data, i.e.,

$$
\widehat{u}_q(x_q) = E_p\left[U_q(\boldsymbol{x}_q) \mid \boldsymbol{Y}^n\right] \qquad (7.5.11)
$$

where the subscript p in the expectation means that it with respect to the $Y(\boldsymbol{x})$ process. When the model parameters are unknown and not assessed by a prior distribution, $\widehat{u}_q(x_q)$ will depend on these parameter values. Empirical Bayes estimators of $u_q(x_q)$ estimate the parameters from the data, as in Chapter 3, and plug these values into the formula for $\widehat{u}_q(x_q)$.

To give a specific example suppose that

$$
Y(\boldsymbol{x}) = \beta_0 + Z(\boldsymbol{x})
$$

is the assumed model for the output $y(\boldsymbol{x})$ where β_0 is unknown, and $Z(\boldsymbol{x})$ is a mean zero stationary GP with unknown variance σ_Z^2, and separable covariance function

$$
Cov_p[Y(\boldsymbol{x}^1), Y(\boldsymbol{x}^2)] = \sigma_Z^2 R(\boldsymbol{x}^1 - \boldsymbol{x}^2; \boldsymbol{\theta}) = \sigma_Z^2 \prod_{j=1}^{d} R(x_j^1 - x_j^2 \mid \psi_j). \qquad (7.5.12)
$$

which is known up to a vector of unknown parameters $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_d)$. Again, the subscript p on the covariance operator indicates an expectation with respect to the process. Now suppose that output has been collected at the inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and it desired to predict $U_q(\cdot)$ at x_0. It can be shown that

$$\left(U_q(\boldsymbol{x}_0), Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n\right) = \left(U_q(\boldsymbol{x}_0), \boldsymbol{Y}^n\right)$$

has the joint multivariate normal distribution

$$\begin{pmatrix} U_q(\boldsymbol{x}_0) \\ \boldsymbol{Y}^n \end{pmatrix} \sim N_{1+n}\left[\begin{pmatrix} \beta_0 \\ \mathbf{1}_n\beta_0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \boldsymbol{\Sigma}_{nu} \\ \boldsymbol{\Sigma}_{nu} & \boldsymbol{\Sigma}_{nn} \end{pmatrix}\right],$$

say, where all components of the covariance can depend on $(\beta_0, \sigma_Z^2, \boldsymbol{\psi})$. In particular, σ_U^2 is given in (7.5.10), $\boldsymbol{\Sigma}_{un} = \left(Cov_p(U_q(\boldsymbol{x}_0), Y(\boldsymbol{x}_i))\right)$ is $1 \times n$ vector that can be calculated by an interchange of integrals (see Chen et al (2005, 2006); Svenson et al (2013) give formulas the cases of Gaussian, cubic, and Bohman correlation functions), $\boldsymbol{\Sigma}_{nu} = \boldsymbol{\Sigma}_{nu}^\top$, and $\boldsymbol{\Sigma}_{nn}$ is the $n \times n$ matrix of variances and covariances of the $\boldsymbol{Y}^n$ values given in (7.5.23). Thus, given the model parameters $(\beta_0, \sigma_Z^2, \boldsymbol{\psi})$ and using the formula (B.1.3) for the conditional multivariate normal distribution

$$E_p\left[U_q(\boldsymbol{x}_q) \mid \boldsymbol{Y}^n, (\boldsymbol{\beta}, \sigma_Z, \boldsymbol{\psi})\right] = \beta_0 + \boldsymbol{\Sigma}_{un}\boldsymbol{\Sigma}_{nn}^{-1}\left(\boldsymbol{Y}^n - \mathbf{1}_n\beta_0\right). \tag{7.5.13}$$

Either plug-in estimates of the unknown parameters can be inserted into (7.5.13) or, if a prior has been specified for $(\beta_0, \sigma_Z^2, \boldsymbol{\psi})$, then a fully Bayesian estimate of $u_q(x_q)$ is

$$\widehat{u_q}(x_q) = E_{\left[(\beta_0, \sigma_Z^2, \boldsymbol{\psi}) \mid \boldsymbol{Y}^n\right]}\left[\beta_0 + \boldsymbol{\Sigma}_{un}\boldsymbol{\Sigma}_{nn}^{-1}\left(\boldsymbol{Y}^n - \mathbf{1}_n\beta_0\right).\right] \tag{7.5.14}$$

where the expectation is with respect to the posterior of the parameters given the calculated output data.

tjs note-I have essentially stopped in 7.5 right here

Example 7.5 [*Continued*] **TBD follow up Example 7.4 with $d = 5$ Effect plots using both quadrature and process-quadature methods EBLUP methods**

7.5.2 Estimation of Sensitivity Indices

The idea of this method use to replace $y(\boldsymbol{x})$ in the variance expressions v, v_i^u, and v_{-i}^u, $i = 1, \ldots, n$, by a predictor $\widehat{y}(\boldsymbol{x})$, and integrate the appropriate expectation expressions to estimate the variances. A naive form of this method estimates the uncentered effect function $u_Q(\boldsymbol{x}_Q)$ by

$$\widehat{u}_Q(\boldsymbol{x}_Q) = \int_{[0,1]^{|\overline{Q}|}} \widehat{y}(x_1, \ldots, x_d) \prod_{i \notin Q} dx_i = \frac{1}{n} \sum_{\ell=1}^{n} \widehat{y}(\boldsymbol{x}_Q, \boldsymbol{x}_{-Q,\ell}) w_\ell$$

where $\widehat{y}(\boldsymbol{x}_Q, \boldsymbol{x}_{-Q,\ell})$ is a REML or other EBLUP of $y(\boldsymbol{x}_Q, \boldsymbol{x}_{-Q,\ell})$; the weights $\{w_\ell\}$ and points $\{\boldsymbol{x}_{-Q,\ell}\}$ depend on the selected quadrature method.

When the correlation function $R(\cdot|\,\boldsymbol{\psi})$ of the process is separable, a more accurate method uses the fact that $\widehat{y}(\boldsymbol{x})$ can be reduced to a product of one-dimensional integrals and these can integrated explicitly for certain correlation functions. The statistical software `JMP` uses this method to estimate sensitivity indices for the separable Gaussian and cubic correlation functions. The steps are sketched below.

Consider prediction based on the stationary Gaussian process model

$$Y(\boldsymbol{x}) = \beta_0 + Z(\boldsymbol{x}) \tag{7.5.15}$$

where β_0 is unknown and $Z(\boldsymbol{x})$ is a stationary Gaussian process on $[0,1]^d$ having zero mean, variance σ_z^2, and has separable correlation function

$$\prod_{\ell=1}^{d} R(h_\ell|\,\psi_\ell)$$

where ψ_ℓ is the (vector of) parameter(s) associated with the ℓ^{th} input. Two specific examples of this structure are based on the one-dimensional Gaussian (2.4.6) or the cubic correlation functions (2.4.9).

Recall that, when viewed as a function of the (new) input vector x, EBLUPs of $y(\boldsymbol{x})$ based on (7.5.15) have the form

$$\widehat{y}(\boldsymbol{x}) = d_0 + \sum_{i=1}^{n} d_i \prod_{\ell=1}^{d} R(x_{0\ell} - x_{i\ell}|\,\widehat{\psi}_\ell) \tag{7.5.16}$$

where $d_0 = \widehat{\beta}_0$ is the weighted least squares estimator of β_0 that is stated below equation below (3.3.4) for an arbitrary set of linear regressors.

Consider estimation of the total variance

$$V = Var\,(y(\boldsymbol{X})) = E\left\{y^2(\boldsymbol{X}))\right\} - (E\{y(\boldsymbol{X}))\})^2 = E\left\{y^2(\boldsymbol{X}))\right\} - (y_0)^2 \tag{7.5.17}$$

which requires estimating two expectation terms.

Starting with estimation of $(y_0)^2$, substituting (7.5.16) into the definition of y_0 yields

$$\begin{aligned}\widehat{y_0} &= \int_0^1 \cdots \int_0^1 \widehat{y}(\boldsymbol{x})\, d\boldsymbol{x} \\ &= d_0 + \sum_{i=1}^{n} d_i \int_0^1 \cdots \int_0^1 \prod_{\ell=1}^{d} Rx_\ell - x_{i\ell}|\,\widehat{\psi}_\ell) \prod_{\ell=1}^{d} dx_\ell \\ &= d_0 + \sum_{i=1}^{n} d_i \prod_{\ell=1}^{d} \int_0^1 R(x_\ell - x_{i\ell}|\,\widehat{\psi}_\ell)\, dx_\ell \\ &= d_0 + \sum_{i=1}^{n} d_i\, S(\boldsymbol{x}_i, \widehat{\boldsymbol{\psi}}),\end{aligned}$$

where

$$S(\boldsymbol{x}_i, \widehat{\boldsymbol{\psi}}) = \prod_{\ell=1}^{d} \int_0^1 R(x_\ell - x_{i\ell}|\,\widehat{\psi}_\ell)\, dx_\ell \tag{7.5.18}$$

Often the one-dimensional integrals in (7.5.18) will have closed form expressions. For example, for the Gaussian correlation function (2.4.6), each term of the product is

$$\begin{aligned}\int_0^1 R\left(x - x_{i\ell}|\,\widehat{\psi}\right) dx &= \int_0^1 exp\{-\psi(x - x_{i\ell})^2\}\, dx \\ &= \frac{\sqrt{2\pi}}{\sqrt{\psi}} \{\Phi(\psi(1 - x_{i\ell})) - \Phi(\psi(0 - x_{i\ell}))\}\end{aligned}$$

when expressed in terms of the standard normal cumulative distribution function.

Returning to the first term in (7.5.17), the squared expectation is estimated by

$$\widehat{E}\left\{y^2(\boldsymbol{X})\right\} = \int_0^1 \cdots \int_0^1 \left(d_0 + \sum_{i=1}^{n} d_i \prod_{\ell=1}^{d} R(x_\ell - x_{i\ell}|\,\widehat{\psi}_\ell)\right)^2 d\boldsymbol{x} \tag{7.5.19}$$

The squared integrand in (7.5.19) is

$$\begin{aligned} & d_0^2 + \sum_{i=1}^{n} d_i^2 \prod_{\ell=1}^{d} R^2(x_\ell - x_{i\ell}|\,\widehat{\psi}_\ell) \\ & + 2\, d_0 \sum_{i=1}^{n} d_i \prod_{\ell=1}^{d} R(x_\ell - x_{i\ell}|\,\widehat{\psi}_\ell) \\ & + 2 \sum_{1 \le q < i \le n} d_q d_i \prod_{\ell=1}^{d} R(x_\ell - x_{q\ell}|\,\widehat{\psi}_\ell) R(x_\ell - x_{i\ell}|\,\widehat{\psi}_\ell)\,. \end{aligned} \tag{7.5.20}$$

The integrals of the terms in (7.5.20) can be expressed as products of one dimensional integrals. For example, the integral of the 3^{rd} term of (7.5.20) is

$$2\, d_0 \sum_{i=1}^{n} d_i\, S(\boldsymbol{x}_i, \widehat{\boldsymbol{\psi}})$$

Similarly, the integral of the final term in (7.5.20) is

$$2 \sum_{1 \le q < i \le n} d_q d_i\, D(\boldsymbol{x}_q, \boldsymbol{x}_i, \boldsymbol{\psi})$$

where

$$D(\boldsymbol{x}_q, \boldsymbol{x}_i, \boldsymbol{\psi}) = \prod_{\ell=1}^{d} \int_0^1 R(x_\ell - x_{q\ell}|\, \widehat{\psi}_\ell) R(x_\ell - x_{i\ell}|\, \widehat{\psi}_\ell) dx_{0\ell}\ .$$

Differencing these component estimators we obtain

$$\begin{aligned} \widehat{v} = d_0^2 &+ \sum_{i=1}^{n} d_i^2\, D(x_i, x_i, \widehat{\boldsymbol{\psi}}) + 2\, d_0 \sum_{i=1}^{n} d_i\, S(\boldsymbol{x}_i, \widehat{\boldsymbol{\psi}}) \\ &+ 2 \sum_{1 \le q < i \le n} d_q d_i\, D(x_q, x_i, \widehat{\boldsymbol{\psi}}) \\ &- \left(d_0 + \sum_{i=1}^{n} d_i\, S(\boldsymbol{x}_i, \widehat{\boldsymbol{\psi}}) \right)^2, \end{aligned}$$

which can be further simplified. The terms v_i^u and v_{-i}^u can be calculated using similar strategies.

7.5.3 Process-based Estimators of Sensitivity Indices

The next method of estimating the S_i and T_i sensitivity indices is Bayesian or empirical/plug-in Bayesian depending on whether unknown parameters are modeled hierarchically or estimated (see Oakley (2009), Moon (2010), Svenson (2011)). The idea of the method is to replace $y(\cdot)$ by the process $Y(\cdot)$ in the relevant integrals that involve $u_Q(\boldsymbol{x}_Q)$ functions and use the fact that for Gaussian process models, $Y(\boldsymbol{x})$, the integral

$$U_Q(\boldsymbol{x}_Q) \equiv \int_0^1 \cdots \int_0^1 Y(x_1, \ldots, x_d) \prod_{i \notin Q} dx_i = E\left[Y(\boldsymbol{X}) | \boldsymbol{X}_Q = \boldsymbol{x}_Q \right],$$

is (under mild conditions) a process for which the joint distribution of $U_Q(\boldsymbol{x}_Q)$ and the vector of $Y(\cdot)$ values at the training data sites, is multivariate normal. Heuristically, $U_Q(\boldsymbol{x}_Q)$ is approximately a linear combination of $Y(\boldsymbol{x})$ values and, for multivariate normal random variables, this linear combination and the responses at the training data sites will have a multivariate normal distribution (in the limit-see Yaglom (1962) or Adler (1990) **Yaglom?? Adler??**).

Specifically suppose that output has been collected at the inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and that $Y(\boldsymbol{x})$ is a stationary Gaussian process with unknown mean β_0, unknown variance σ_Y^2 and separable covariance function

$$Cov_p[Y(\boldsymbol{x}^1), Y(\boldsymbol{x}^2)] = \sigma_Y^2 R(\boldsymbol{x}_1 - \boldsymbol{x}_2; \boldsymbol{\theta}) = \sigma_Y^2 \prod_{\ell=1}^{d} R(x_\ell^1, x_\ell^2; \psi_\ell) \tag{7.5.21}$$

where $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_d)$ is the vector of unknown parameters of the correlation function. Then it can be shown that

$$(U_Q(\boldsymbol{x}_Q), Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n) = (U_Q(\boldsymbol{x}_Q), Y^n)$$

has the joint multivariate normal distribution

$$N_{1+n}\left[\begin{pmatrix} \beta_0 \\ \mathbf{1}_n \beta_0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \boldsymbol{\Sigma}_{nu} \\ \boldsymbol{\Sigma}_{nu} & \boldsymbol{\Sigma}_{nn} \end{pmatrix}\right],$$

say, where $\boldsymbol{\Sigma}_{un}$ is $1 \times n$, $\boldsymbol{\Sigma}_{nu} = \boldsymbol{\Sigma}_{nu}^\top$, and $\boldsymbol{\Sigma}_{nn}$ is $n \times n$ can be calculated in terms of the unknown model parameters $(\beta_0, \sigma_Y^2, \boldsymbol{\psi})$. Using these moment expressions, the posterior mean of the variance of $U_Q(\boldsymbol{x}_Q)$ given the training data,

$$\widehat{V}_Q^u = E_P\left\{Var\left[U_Q(\boldsymbol{x}_Q)\right] \mid \boldsymbol{Y}^n = \boldsymbol{y}^n\right\}, \tag{7.5.22}$$

is an estimator of v_Q^u that can be calculated explicitly and hence the main effect and total effect sensitivity indices can be estimated by plug-in of the unknown parameters or by averaging $\widehat{V}_Q^u$ over draws from the posterior distribution of the parameters given $\boldsymbol{Y}^n$.

New from here

This section presents Bayesian and plug-in Bayesian approaches for estimating sensitivity indices in the case where the *observed* output at input site $\boldsymbol{x} \in \mathcal{X}$ can be modeled as a draw, $y(\boldsymbol{x})$, from a (smooth) Gaussian stochastic process, $Y(\boldsymbol{x})$, possibly corrupted by additive noise, say numerical. The function $y(\boldsymbol{x})$ is regarded as the *true* output. In this section the process $Y(\boldsymbol{x})$ need not be stationary but is assumed to be separable with covariance

$$Cov_p[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_k)] = \sigma^2 R(\boldsymbol{x}_i, \boldsymbol{x}_k \mid \boldsymbol{\psi}) = \sigma^2 \prod_{j=1}^{d} R(x_{ij}, x_{kj} \mid \psi_j) \tag{7.5.23}$$

where σ^2 is the process variance and $R(\cdot, \cdot \mid \psi_j)$ is known up to an unknown (vector of) parameter(s) ψ_j. Here, and below, $Cov_p[\cdot, \cdot]$ and $E_p[\cdot]$ denote covariance and expectation with respect to the process $Y(\boldsymbol{x})$ to distinguish them from expectations $E_g[\cdot]$ with respect $\boldsymbol{X}$. As in Section **??**, this approach allows estimation of sensitivity indices from runs based on an arbitrary design.

To simplify the expressions derived below, this section makes the following additional assumptions. First, not only has the input space been scaled to be $[0, 1]^d$,

but the weight function $g(\cdot)$ is uniform on $[0, 1]$. Second, we take the process $Y(\boldsymbol{x})$ to have mean

$$\boldsymbol{f}^{\top}(\boldsymbol{x})\boldsymbol{\beta} = E_p\left[Y(\boldsymbol{x})\right] = \sum_{k_1=0}^{m_{k_1}} \cdots \sum_{k_d=0}^{m_{k_d}} \beta_{k_1 \ldots k_d} \prod_{j=1}^{d} x_j^{k_j}. \tag{7.5.24}$$

As Kaufman et al (2011) demonstrate, a non-constant mean is essential when using compactly supported correlations to emulate computer simulator codes efficiently for large designs; the polynomial mean (7.5.24) is general enough to account for a wide variety of "large scale" trends. Third, while nothing in the calculations below requires that the process be stationary, the specific correlation functions used in the sample calculations all satisfy the stationary condition

$$R(x_{ij}, x_{kj} | \, \psi_j) = R(x_{ij} - x_{kj} | \, \psi_j).$$

Finally, to allow a greater breath of applications, it is assumed that the *observed output*, $z_{sim}(\boldsymbol{x})$, from the simulator runs is the true simulator output $y(\boldsymbol{x})$ plus noise, possibly numerical. The model for $z_{sim}(\boldsymbol{x})$ is

$$Z_{sim}(\boldsymbol{x}) = Y(\boldsymbol{x}) + \epsilon_{sim}(\boldsymbol{x}), \tag{7.5.25}$$

where $\epsilon_{sim}(\boldsymbol{x})$ is a white noise process with mean zero and variance σ_ϵ that is independent of $Y(\boldsymbol{x})$. The term $\epsilon_{sim}(\boldsymbol{x})$ can be thought of as a means of explicitly modeling non-deterministic behaviour of the computer output or of enhancing numerical stability in the estimation of the correlation parameters. For deterministic outputs, $\epsilon_{sim}(\boldsymbol{x})$ can be set to zero in the formulae below.

Assuming that evaluations are made at inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, the $n \times 1$ vector of observed outputs is viewed as a realization of the stochastic process

$$\boldsymbol{Z}_{sim} = (Z_{sim}(\boldsymbol{x}_1), \ldots, Z_{sim}(\boldsymbol{x}_n))^{\top}$$

which has mean vector $\boldsymbol{F}\boldsymbol{\beta}$ with $\boldsymbol{F} = \left[\boldsymbol{f}(\boldsymbol{x}_1), \ldots, \boldsymbol{f}(\boldsymbol{x}_n)\right]^{\top}$ and covariance matrix

$$\boldsymbol{\Sigma}^{Z}_{sim} = \sigma^2 \boldsymbol{R} + \sigma^2_\epsilon \boldsymbol{I}_n = \sigma^2 \left(\boldsymbol{R} + a\boldsymbol{I}_n\right)$$

with $a = \sigma^2_\epsilon / \sigma^2$, where the $(i, k)^{th}$ element of the $n \times n$ matrix $\boldsymbol{R}$ is $R(\boldsymbol{x}_i, \boldsymbol{x}_k; \boldsymbol{\psi})$ and $\boldsymbol{I}$ is the $n \times n$ identity matrix.

7.5.4 Process-based estimators of sensitivity indices

The sensitivity indices (7.4.19) and (7.4.23) are defined in terms of the true simulator output $y(\boldsymbol{x})$. Bayesian estimation of these quantities replaces $y(\cdot)$ by the process $Y(\cdot)$ in the relevant definitions resulting, for example, in the random uncorrected effect function

$$V_Q^u = Var_g\left[E_g[Y(\boldsymbol{X})|\boldsymbol{X}_Q]\right]$$

for $Q \subseteq \{1, \ldots d\}$. A Bayesian estimator of v_Q^u is the posterior mean of V_Q^u given the observed code runs z_{sim}; that is,

$$\widehat{v_Q^u} = E_P\left[V_Q^u |\, \boldsymbol{Z}_{sim} = z_{sim}\right] . \tag{7.5.26}$$

where $E_P[\,\cdot \,|\, \boldsymbol{Z}_{sim}]$ denotes the conditional expectation with respect to the process $Y(\cdot)$ given $\boldsymbol{Z}_{sim}$. For Gaussian process models, the joint distribution of the integrated process, V_Q^u, and $\boldsymbol{Z}_{sim}$ is multivariate normal; this allows the posterior expected value (7.5.26) of the integrated process given the training data to be calculated explicitly.

A formula for (7.5.26) is presented in the following theorem. The proof of this result is rather long and technical and the details can be found in the Supplementary Material. The expression for (7.5.26) assumes that *all* Gaussian Process parameters are known. In the fully Bayesian approach to estimation, priors are placed on the unknown parameters and (7.5.26) is averaged over draws from the posterior distribution of parameters given $\boldsymbol{Z}_{sim}$. In the empirical Bayesian approach, estimates of the unknown parameters are plugged into the (7.5.26) formula. The notation in Theorem 7.1 is given in a style that facilitates function calls in a computer program. In particular, it uses the following notation:

$$\begin{aligned}
\mathbf{S1}_k(x;\psi) &= \int_0^1 w^k R\,(w, x|\,\psi)\; dw, \quad k = 0, 1, 2, \ldots, \\
\mathbf{S2}(x_1, x_2;\psi) &= \int_0^1 R(w, x_1|\,\psi) R(w, x_2|\,\psi)\; dw, \\
\mathbf{D}(\psi) &= \int_0^1 \int_0^1 R(w, x|\,\psi)\; dx\; dw, \\
\mathbf{m1}(\boldsymbol{\beta}) &= \sum_{k_1=0}^{m_{k_1}} \cdots \sum_{k_d=0}^{m_{k_d}} \beta_{k_1 \ldots k_d} \prod_{j=1}^{d} \left(k_j + 1\right)^{-1}, \\
\mathbf{m2}(\boldsymbol{\beta}) &= \sum_{k_1,\ldots k_d} \sum_{k_1',\ldots k_d'} \beta_{k_1,\ldots k_d} \beta_{k_1',\ldots k_d'} \\
&\qquad \times \left[\prod_{j \notin Q} \left(k_j + 1\right)\left(k_j' + 1\right)\right]^{-1} \left[\prod_{\ell \in Q} \left(k_\ell + k_\ell' + 1\right)\right]^{-1} .
\end{aligned}$$

Theorem 7.1. *Assume that the true simulator output, $y(\boldsymbol{x})$ can be modeled by a stationary Gaussian process $Y(\cdot)$ with mean and covariance function of the form (7.5.24) and (7.5.23), respectively. Also assume that the observed output $\boldsymbol{z}_{sim}$ at the training data sites, is modeled by a process $Z_{sim}(\boldsymbol{x})$ satisfying (7.5.25). For a fixed $Q \subseteq \{1, \ldots d\}$,*

$$\begin{aligned}\widehat{v}_Q^u &= E_P\left\{V_Q^u |\, \boldsymbol{Z}_{sim} = \boldsymbol{z}_{sim}\right\} \\ &= \left\{\sigma_Y^2 \prod_{j\notin Q} \mathbf{D}(\psi_j) - trace\left[\left(\boldsymbol{\Sigma}_{sim}^Z\right)^{-1} \boldsymbol{C}\right]\right\} \\ &\quad + \left\{\mathbf{m2}(\boldsymbol{\beta}) - \mathbf{m1}^2(\boldsymbol{\beta}) + 2\left(\boldsymbol{v}^\top - \mathbf{m1}(\boldsymbol{\beta})\boldsymbol{q}^\top\right)\left(\boldsymbol{\Sigma}_{sim}^Z\right)^{-1}\left(\boldsymbol{z}_{sim} - \boldsymbol{F}^\top\boldsymbol{\beta}\right)\right. \\ &\quad \left. + \left(\boldsymbol{z}_{sim} - \boldsymbol{F}^\top\boldsymbol{\beta}\right)^\top\left(\boldsymbol{\Sigma}_{sim}^Z\right)^{-1}\left(\boldsymbol{C} - \boldsymbol{q}\boldsymbol{q}^\top\right)\left(\boldsymbol{\Sigma}_{sim}^Z\right)^{-1}\left(\boldsymbol{z}_{sim} - \boldsymbol{F}^\top\boldsymbol{\beta}\right)\right\} \\ &\quad - \left\{\sigma_Y^2 \prod_{j=1}^{d} \mathbf{D}(\psi_j) - trace\left[\left(\boldsymbol{\Sigma}_{sim}^Z\right)^{-1}\boldsymbol{q}\boldsymbol{q}^\top\right]\right\}, \end{aligned} \tag{7.5.27}$$

where $\boldsymbol{q}$ *is the* $n \times 1$ *vector with* i^{th} *element*

$$q_i = q(\boldsymbol{x}_i, \boldsymbol{\psi}) = \sigma^2 \prod_{j=1}^{d} \mathbf{S1}_0(x_{ij}; \psi_j), \quad 1 \le i \le n,$$

$\boldsymbol{C}$ *is the* $n \times n$ *matrix with* $(i,k)^{th}$ *element*

$$C_{ik} = \sigma^4 \prod_{j\notin Q} \mathbf{S1}_0(x_{ij};\psi_j)\, \mathbf{S1}_0(x_{kj};\psi_j) \prod_{j\in Q} \mathbf{S2}(x_{ij}, x_{kj};\psi_j), \quad 1 \le i,k \le n\,,$$

$\boldsymbol{v}$ *is the* $n \times 1$ *vector with* i^{th} *element*

$$\begin{aligned}\boldsymbol{v}(\boldsymbol{x}_i, \boldsymbol{\beta}, \boldsymbol{\psi}, \boldsymbol{\beta}) &= \left[\sigma^2 \prod_{j\notin Q} \mathbf{S1}_0\left(x_{ij};\psi_j\right)\right] \\ &\quad \times \sum_{k_1=0}^{m_{k_1}} \cdots \sum_{k_d=0}^{m_{k_d}} \left\{\beta_{k_1\ldots k_d} \prod_{j\notin Q}\left(k_j+1\right)^{-1} \prod_{\ell\in Q} \mathbf{S1}_{k_\ell}(x_{h\ell};\psi_\ell)\right\}, \quad 1 \le i \le n,\end{aligned}$$

Proof. The proof of Theorem 1 involves three steps: (i) the derivation of the distribution of the process $U_Q(\boldsymbol{x}_Q) \equiv E_g[Y(\boldsymbol{X})|\boldsymbol{X}_Q = \boldsymbol{x}_Q]$; (ii) the determination of the conditional distribution of $[U_Q(\boldsymbol{x}_Q)|\, \boldsymbol{Z}_{sim}]$; and (iii) obtaining an expression for $E_P\left[Var_g\left(U_Q(\boldsymbol{x}_Q)\right) |\, \boldsymbol{Z}_{sim}\right]$. The details are given in the Supplementary Material.

The estimate $\hat{v}$ of the total variance v is given by (7.5.27) for $Q = \{1,\ldots,d\}$. The main effect sensitivity index S_j in (7.4.23) for the individual input x_j is estimated by

$$\widehat{S}_j = \widehat{v}_j^u / \widehat{v} \tag{7.5.28}$$

where $\widehat{v}_j^u$ is obtained from (7.5.27) with $Q = \{j\}$. The total effect sensitivity index is estimated by

$$\widehat{T}_j = (\widehat{v} - \widehat{v}_{-j}^u)/\widehat{v}, \tag{7.5.29}$$

where $\widehat{v}_{-j}^u$ is obtained from (7.5.27) with $Q = \{1,\ldots,i-1,i+1,\ldots,d\}$.

Given the model parameters, all components of $\widehat{v}_Q^u$ are specified above except the integrals $\mathbf{S1}_k$, $\mathbf{D}$, $\mathbf{S2}$, which depend on the user-selected correlation function $R(\cdot,\cdot|\,\psi)$. Formulas for these integrals are stated next for the Gaussian and Bohman

correlation functions and, in the Supplementary Material, for the cubic correlation function $R(w, x|\,\psi) = R_C(w - x|\,\psi)$ for $\psi > 0$ where

$$R_C(h|\,\psi) = \begin{cases} 1 - 6\left(\frac{h}{\psi}\right)^2 + 6\left(\frac{|h|}{\psi}\right)^3, & |h| < \frac{\psi}{2}; \\ 2\left(1 - \frac{|h|}{\psi}\right)^3, & \frac{\psi}{2} \le |h| < \psi; \\ 0, & \psi \le |h|\,. \end{cases}$$

7.5.5 Formulae for the Gaussian correlation function

For the Gaussian correlation function, $R(w, x|\,\psi) = R_G(w - x|\,\psi)$ where

$$R_G(h|\,\psi) = \exp\left[-\psi h^2\right],$$

where $\psi > 0$. A formula for $\mathbf{S1}_k$ can be derived using results of Dhrymes (2005) for the moments of a truncated normal random variable. Application of this result gives

$$\begin{aligned} \mathbf{S1}_k(x;\psi) &= \int_0^1 w^k \exp[-\psi(w-x)^2]\,dw \\ &= \sqrt{\frac{\pi}{\psi}}\left\{\Phi\left(\sqrt{2\psi}(1-x)\right)\sum_{r=0}^{k}\binom{k}{r}x^{k-r}(2\psi)^{-r/2}I_r^{h_1}\right. \\ &\quad \left. - \Phi\left(-x\sqrt{2\psi}\right)\sum_{r=0}^{k}\binom{k}{r}\eta^{k-r}(2\psi)^{-r/2}I_r^{h_0}\right\} \end{aligned} \tag{7.5.30}$$

where $h_0 = -x\sqrt{2\psi}$ and $h_1 = (1-x)\sqrt{2\psi}$, while I_r^h is defined recursively by $I_0^h = 1$, $I_1^h = -\phi(h)/\Phi(h)$, and for $r \in \{2, 3, 4, \ldots\}$ by

$$I_r^h = \frac{1}{\Phi(h)}\left[-h^{r-1}\phi(h) + (r-1)I_{r-2}^h\right], \tag{7.5.31}$$

where $\phi(\cdot)$ denotes the probability density function (pdf) of the standard normal distribution. In particular, $\mathbf{S1}_0(x;\psi)$ becomes

$$\begin{aligned} \mathbf{S1}_0(x;\psi) &= \int_0^1 \exp\left[-\psi(w-x)^2\right]\,dw \\ &= \sqrt{\frac{\pi}{\psi}}\left[\Phi\left(\sqrt{2\psi}(1.0-x)\right) - \Phi\left(-x\sqrt{2\psi}\right)\right]. \end{aligned}$$

Formulae for **S2** and **D** are

$$\mathbf{S2}(x_1, x_2; \psi) = \int_0^1 \exp[-\psi(w - x_1)^2] \exp[-\psi(w - x_2)^2]\, dw$$
$$= \exp\left[-\frac{1}{2}\psi(x_1 - x_2)^2\right] \mathbf{S1}_0\left(\frac{x_1 + x_2}{2}; 2\psi\right),$$

$$\mathbf{D}(\psi) = \int_0^1 \int_0^1 \exp\left[-\psi(w - x)^2\right]\, dx\, dw$$
$$= \frac{1}{\psi}\left[\sqrt{2\pi}\phi\left(\sqrt{2\psi}\right) - 1\right] + \sqrt{\frac{\pi}{\psi}}\left[2\Phi\left(\sqrt{2\psi}\right) - 1\right].$$

7.5.6 Formulae using the Bohman correlation function

For the Bohman correlation function, $R(w, x|\,\psi) = R_B(w - x|\,\psi)$ where

$$R_B(h|\,\psi) = \begin{cases} \left(1 - \frac{|h|}{\psi}\right)\cos\left(\frac{\pi|h|}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi|h|}{\psi}\right), & |h| < \psi; \\ 0, & |h| \geq \psi \end{cases}$$

with $\psi > 0$. The integrals $\mathbf{S1}_0$, $\mathbf{S1}_k$, $\mathbf{S2}$, and $\mathbf{D}$ are as follows. Letting $l^* = l^*(x) = \min(\pi, x\pi/\psi)$ and $u^* = u^*(x) = \min(\pi, (1.0 - x)\,\pi/\psi)$,

$$\mathbf{S1}_0(x; \psi) = \frac{1}{u - l}\left\{\frac{4\psi}{\pi^2} - \frac{2\psi}{\pi^2}\cos(l^*(x)) - \frac{2\psi}{\pi^2}\cos(u^*(x))\right.$$
$$\left. + \left\{\left(\frac{\psi}{\pi} - \frac{\psi\, l^*(x)}{\pi^2}\right)\sin(l^*(x)) + \left(\frac{\psi}{\pi} - \frac{\psi\, u^*(x)}{\pi^2}\right)\sin(u^*(x))\right\}.$$

For the integral $\mathbf{S1}_k(x; \psi)$, let $l^* = \max(0, x - \psi)$ and $u^* = \min(1, x + \psi)$, then

$$\mathbf{S1}_k(x; \psi) = \int_0^1 w^k R(w, x, ; \psi)\, dw$$
$$= \int_{l^*}^{x} w^k \left\{\left(1 - \frac{x - w}{\psi}\right)\cos\left(\frac{\pi(x - w)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(x - w)}{\psi}\right)\right\} dw$$
$$+ \int_x^{u^*} w^k \left\{\left(1 - \frac{w - x}{\psi}\right)\cos\left(\frac{\pi(w - x)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(w - x)}{\psi}\right)\right\} dw$$
$$= \mathbf{T}(x, -1, l^*, \eta) + \mathbf{T}(-x, +1, \eta, u^*),\ \text{say},$$

where

$$\begin{aligned}
\mathbf{T}(d,s,a,b) &= \int_a^b w^k \left\{ \left(1 - \frac{d+sw}{\psi}\right) \cos\left(\frac{\pi(d+sw)}{\psi}\right) + \frac{1}{\pi} \sin\left(\frac{\pi(d+sw)}{\psi}\right) \right\} dw \\
&= \left(1 - \frac{d}{\psi}\right) \cos\left(\frac{d\pi}{\psi}\right) \left(\frac{\psi}{\pi s}\right)^{k+1} \mathbf{P1}(k, a', b') \\
&- \left(1 - \frac{d}{\psi}\right) \sin\left(\frac{d\pi}{\psi}\right) \left(\frac{\psi}{\pi s}\right)^{k+1} \mathbf{P2}(k, a', b') \\
&- \frac{s}{\psi} \cos\left(\frac{d\pi}{\psi}\right) \left(\frac{\psi}{\pi s}\right)^{k+2} \mathbf{P1}(k+1, a', b') \\
&+ \frac{s}{\psi} \sin\left(\frac{d\pi}{\psi}\right) \left(\frac{\psi}{\pi s}\right)^{k+2} \mathbf{P2}(k+1, a', b') \\
&+ \frac{1}{\pi} \sin\left(\frac{d\pi}{\psi}\right) \left(\frac{\psi}{\pi s}\right)^{k+1} \mathbf{P1}(k, a', b') \\
&+ \frac{1}{\pi} \cos\left(\frac{d\pi}{\psi}\right) \left(\frac{\psi}{\pi s}\right)^{k+1} \mathbf{P2}(k, a', b'),
\end{aligned}$$

after additional algebra, where $a' = sa\pi/\psi$, $a', b' = sb\pi/\psi$ and **P1** and **P2** are defined recursively as

$$\begin{aligned}
&\mathbf{P1}(k, a', b') \\
&= \begin{cases} \sin(b') - \sin(a'), & k = 0; \\ (b')^k \sin(b') - (a')^k \sin(a') - k\,\mathbf{P2}(k-1, a', b'), & k \geq 1, \end{cases}
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{P2}(k, a', b') \\
&= \begin{cases} \cos(a') - \cos(b'), & k = 0; \\ (a')^k \cos(a') - (b')^k \cos(b') + k\,\mathbf{P1}(k-1, a', b'), & k \geq 1. \end{cases}
\end{aligned}$$

The integral for $\mathbf{D}(\psi)$ is defined piecewise by

$$\mathbf{D}(\psi) = \begin{cases} \frac{4\psi}{\pi^2} + \frac{2\psi^2}{\pi^2} - \frac{4\psi}{\pi^2}(\psi - 1.0), & 0 < \psi < 1.0 \\ \frac{4\psi}{\pi^2} + \frac{2\psi^2}{\pi^2}\left\{1 + \left(\frac{1.0-\psi}{\psi}\right)\cos\left(\frac{\pi}{\psi}\right) - \frac{3}{\pi}\sin\left(\frac{\pi}{\psi}\right)\right\}, & 1.0 \leq \psi. \end{cases}$$

To calcuate the integral $\mathbf{S2}(x_1, x_2; \psi)$, the w regions of $[0, 1]$ for which $R(w, x_1; \psi)R(w, x_2; \psi) \neq 0$ must be identified. These regions will depend on the relationship between $|x_1 - x_2|$ and ψ. The following formulae assume, without loss of generality, that $x_1 < x_2$. There are different expressions for **S2** depending on whether $2\psi \leq |x_1 - x_2|$, $\psi \leq |x_1 - x_2| < 2\psi$, or $|x_1 - x_2| < \psi$. In the Supplementary Material, these integrals are simplified and shown to be as follows.

Case 1: For (x_1, x_2) satisfying $|x_1 - x_2| \geq 2\psi$, $R(w, x_1; \psi)R(w, x_2; \psi) = 0$ for all $w \in [0, 1]$; hence

$$\mathbf{S2}(x_1, x_2; \psi) = 0\,.$$

Case 2: For (x_1, x_2) satisfying $\psi \leq |x_1 - x_2| < 2\psi$,

$$\mathbf{S2}(x_1, x_2; \psi) = \int_{x_2-\psi}^{x_1+\psi} \left\{\left(1 - \frac{w - x_1}{\psi}\right)\cos\left(\frac{\pi(w - x_1)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(w - x_1)}{\psi}\right)\right\} \times \left\{\left(1 - \frac{(x_2 - w)}{\psi}\right)\cos\left(\frac{\pi(x_2 - w)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(x_2 - w)}{\psi}\right)\right\} dw\,.$$

Case 3: For (x_1, x_2) satisfying $|x_1 - x_2| < \psi$, first let $l^* = \max(0, x_1 - \psi)$ and $u^* = \min(1, x_2 + \psi)$, then

$$\begin{aligned}\mathbf{S2}(x_1, x_2; \psi) = &\int_{l^*}^{x_1} \left\{\left(1 - \frac{x_1 - w}{\psi}\right)\cos\left(\frac{\pi(x_1 - w)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(x_1 - w)}{\psi}\right)\right\} \\ &\times \left\{\left(1 - \frac{(x_2 - w)}{\psi}\right)\cos\left(\frac{\pi(x_2 - w)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(x_2 - w)}{\psi}\right)\right\} dw \\ &+ \int_{x_1}^{x_2} \left\{\left(1 - \frac{x - x_1}{\psi}\right)\cos\left(\frac{\pi(w - x_1)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(w - x_1)}{\psi}\right)\right\} \\ &\times \left\{\left(1 - \frac{(x_2 - w)}{\psi}\right)\cos\left(\frac{\pi(x_2 - w)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(x_2 - w)}{\psi}\right)\right\} dw \\ &+ \int_{x_2}^{u^*} \left\{\left(1 - \frac{w - x_1}{\psi}\right)\cos\left(\frac{\pi(w - x_1)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(w - x_1)}{\psi}\right)\right\} \\ &\times \left\{\left(1 - \frac{w - x_2}{\psi}\right)\cos\left(\frac{\pi(w - x_2)}{\psi}\right) + \frac{1}{\pi}\sin\left(\frac{\pi(w - x_2)}{\psi}\right)\right\} dw\,.\end{aligned}$$

7.6 Variable Selection

- Linkletter et al (2006)
- Moon et al (2012)

7.7 Chapter Notes

Some possible notes

7.7.1 Elementary Effects

- Pujols (2001) introduced a ***non − collapsing*** OAT design, i.e., projections of input vectors from the OAT are not identical.
- Campolongo et al (2007) propose a criterion for ***spreadingther completetours*** and use of $\overline{|d_j|} = \frac{1}{n}\sum_{i=1}^{r} |d_j(\boldsymbol{x}_1^j)|$

- Campolongo et al (2011) introduces ***radialOATdesigns*** that spreads starting points using a Sobol´ sequence and differential Δ for each inputs

7.7.2 Orthogonality of Sobol´ Terms

Among other authors, Van Der Vaart (1998) (Section 11.4) shows that any component of (7.4.9), say $y_Q(\boldsymbol{X}_Q)$, where $Q \subset \{1, 2, \ldots, d\}$, must have zero mean when integrated with respect to any input X_i, with $i \in Q$, and any pair of terms in (7.4.9) must be *pairwise orthogonal*. We give a proof of these two facts.

Lemma 7.1 *For $Q = \{j_1, \ldots, j_s\} \subseteq \{1, \ldots, d\}$,*

$$\int_0^1 y_Q(\boldsymbol{x}_Q)\, dx_{j_k} = 0 \tag{7.7.1}$$

for any $j_k \in Q$.

Proof: The proof proceeds by induction on the number of elements in Q. When $Q = \{j\}$, say, then from (7.4.1), (7.4.2) and the definition of y_0, (7.7.1) holds for any main effect function $y_j(x_j)$. Suppose that $Q \subseteq \{1, \ldots, d\}$ contains two or more elements, and assume that (7.7.1) holds for all proper subsets of $E \subset Q$. Fix $\ell \in Q$, and let $Q\backslash\ell$ denote the set difference of Q and $\{\ell\}$, which is non-empty by definition of Q. Partition the non-empty subsets of Q into the collection $\mathcal{U}_+$ of subsets E that *contain* ℓ, and the collection $\mathcal{U}_-$ of subsets E that *do not contain* ℓ; note that $Q\backslash\ell \in \mathcal{U}_-$. Then by the definition (7.4.12) of $y_Q(\boldsymbol{x}_Q)$,

$$\begin{aligned}
\int y_Q(\boldsymbol{x}_Q)\, dx_\ell &= \int \left\{ u_Q(\boldsymbol{x}_Q) - \sum_{E \subset Q} y_E(\boldsymbol{x}_E) - y_0 \right\} dx_\ell \\
&= \int u_Q(\boldsymbol{x}_Q)\, dx_\ell - \sum_{E \in \mathcal{U}_+} \int y_E(\boldsymbol{x}_E)\, dx_\ell \\
&\quad - \sum_{E \in \mathcal{U}_-} y_E(\boldsymbol{x}_E) - y_0\,,
\end{aligned} \tag{7.7.2}$$

where the third and fourth terms use the fact that their integrands do not depend on x_ℓ (because $\ell \notin E$ for $E \in \mathcal{U}_-$).

By definition of $u_Q(\boldsymbol{x}_Q)$, the first term of (7.7.2) is

$$\int u_Q(\boldsymbol{x}_Q)\, dx_\ell = \int \int y(\boldsymbol{x}_Q, \boldsymbol{x}_{-Q})\ dx_Q\, dx_\ell = u_{Q\ell}(\boldsymbol{x}_{Q\ell}).$$

The second term of (7.7.2) is zero since (7.7.1) holds for all proper subsets of Q by assumption. This gives that (7.7.2) is

$$\int y_Q(\boldsymbol{x}_Q)\, g(x_\ell) dx_\ell = u_{Q\ell}(\boldsymbol{x}_{Q\ell}) - 0 - \left(\sum_{E \in \mathcal{U}_-; E \neq Q\ell} y_E(\boldsymbol{x}_E) + y_{Q\ell}(\boldsymbol{x}_{Q\ell}) \right) - y_0 \,,$$

which is zero by definition of $u_{Q\ell}(\boldsymbol{x}_{Q\ell})$. ■

Notice that Lemma 7.7.2 implies that the mean of each $y_Q(\boldsymbol{X}_Q)$ with respect to $\boldsymbol{X}_Q$ is zero, i.e., $E\{y_Q(\boldsymbol{X}_Q)\} = 0$ for any $Q \subseteq \{1, \ldots, d\}$. This is a stronger form of centering than that of $u_Q(\boldsymbol{X}_Q)$ by y_0 which also satisfies $E\{u_Q(\boldsymbol{x}_Q) - y_0\} = 0$ but for which $\int_0^1 (u_Q(\boldsymbol{x}_Q) - y_0)\ dx_{j_k}$ need not be zero for any $j_k \in Q$.

Lemma 7.7.2 also implies that the orthogonality in (7.4.14) holds. Suppose that $(i_1, \ldots, i_s) \neq (j_1, \ldots, j_t)$; pick any integer k that is in exactly one of $(i_1, \ldots, i_s)$ or $(j_1, \ldots, j_t)$ (there has to be at least one such integer), and integrate

$$\int_0^1 \cdots \int_0^1 y_{i_1,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s}) \times y_{j_1,\ldots,j_t}(x_{j_1}, \ldots, x_{j_t}) \prod_\ell dx_\ell = 0 \qquad (7.7.3)$$

in the order k and then over $(i_1, \ldots, i_s) \cup (j_1, \ldots, j_t) \setminus \{k\}$ (in any order). The inner integral is zero and thus (7.4.14) holds.

- Saltelli and his co-authors given numerous methods to estimate first order and total Sobol' indices)
- Helton & Storlie and co-authors describe smoothing and metamodel-based methods
- Kucherenko & Sobol recent works provide links between derivative-based measures and Sobol´ indices
- Other research that uses the emulation of the model output by a Gaussian process. It seems that these works have already been done (perhaps differently?) in Oakley & O'Hagan (2004), Chen et al. (2005) and Marrel et al. (2009)

Additional possible references

- W. Chen, R. Jin and A. Sudjianto: Analytical metamodel-based global sensitivity analysis and uncertainty propagation for robust design. J. Mech. Des. 2005,127:875-86.
- S. DA VEIGA, F. WAHL and F. GAMBOA : Local polynomial estimation for sensitivity analysis on models with correlated inputs. Technometrics, 51(4):452-463, 2009.
- J.C. HELTON, J.D. JOHNSON, C.J. SALABERRY and C.B. STORLIE : Survey of sampling-based methods for uncertainty and sensitivity analysis. Reliability Engineering and System Safety, 91:1175-1209, 2006.
- T. HOMMA and A. SALTELLI : Importance measures in global sensitivity analysis of non linear models. Reliability Engineering and System Safety, 52:1-17, 1996.
- A. MARREL, B. IOOSS, B. LAURENT and O. ROUSTANT : Calculations of the Sobol indices for the Gaussian process metamodel. Reliability Engineering and System Safety, 94:742-751, 2009.

- J.E. OAKLEY and A. O'HAGAN : Probabilistic sensitivity analysis of complex models : A Bayesian approach. Journal of the Royal Statistical Society, Series B, 66:751-769, 2004.
- A. SALTELLI : Making best use of model evaluations to compute sensitivity indices. Computer Physics Communication, 145:280-297, 2002.
- A. SALTELLI, P. ANNONI, I. AZZINI, F. CAMPOLONGO, M. RATTO and S. TARANTOLA : Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Computer Physics Communication, 181:259-270, 2010.
- I.M. SOBOL and S. KUCHERENKO : Derivative based global sensitivity measures and their links with global sensitivity indices. Mathematics and Computers in Simulation, 79:3009-3017, 2009.
- C.B. STORLIE and J.C. HELTON : Multiple predictor smoothing methods for sensitivity analysis : Description of techniques. Reliability Engineering and System Safety, 93:28-54, 2008.
- C.B. STORLIE, L.P. SWILER, J.C. HELTON and C.J. SALABERRY : Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. Reliability Engineering and System Safety, 94:1735-1763, 2009.

- Saltelli et al (2000) provide a detailed and more recent detailed summary of SI methods.
- definition of sensitivity indices for multivariate and function output (Campbell (2001); Campbell et al (2005, 2006))
-
-

7.7.3 Sensitivity Index Estimators for Regression Means

Suppose it is desired to base prediction on the regression plus stationary Gaussian process model

$$Y(\boldsymbol{x}) = \sum_{k_1=0}^{n_{k_1}} \cdots \sum_{k_d=0}^{n_{k_d}} \beta_{k_1 \ldots k_d} \prod_{i=1}^{d} x_i^{k_i} + Z(\boldsymbol{x}), \tag{7.7.4}$$

using either the quadrature or the Bayesian methods of Sections 7.5.2 or 7.5.3. Here the powers $k_1, \ldots, k_d$ are known, $\{\beta_{k_1 \ldots k_d}\}_{k_1 \ldots k_d}$ are unknown, and $Z(\cdot)$ is a stationary Gaussian process on $[0,1]^d$ having zero mean, variance σ_z^2, and separable correlation function

$$\prod_{\ell=1}^{d} R(h_\ell | \boldsymbol{\psi}_\ell)$$

where $\boldsymbol{\psi}_\ell$ is the (vector of) parameter(s) associated with the ℓ^{th} input. This model has several advantages. It is sufficiently simple that analytic expressions for quadrature-based and Bayesian sensitivity index estimators can be computed. It is general

enough to account for almost any large scale trend reasonably well (see its use in Kaufman et al (2011) as a key component to analyze large designs).

Any integrations involving $\boldsymbol{x}_0$ in formulas must now include integrations over the mean terms. For example, in (7.5.18), the integral over the input points is no longer d_0 but becomes

$$\sum_{k_1=0}^{n_{k_1}} \cdots \sum_{k_d=0}^{n_{k_d}} \beta_{k_1 \ldots k_d} \int_0^1 \cdots \int_0^1 \prod_{i=1}^{d} x_i^{k_i} \, dx_i = \sum_{k_1=0}^{n_{k_1}} \cdots \sum_{k_d=0}^{n_{k_d}} \beta_{k_1 \ldots k_d} \prod_{i=1}^{d} \frac{1}{k_i + 1} \, . \qquad (7.7.5)$$

Weighted least squares estimators of each β coefficient are substituted in (7.7.5).

Similar, but more complicated, adjustments must be made to allow a general regression mean in the formulas below.

Appendix A
List of Notation

A.1 Abbreviations

ARD — Average reciprocal distance (design) (Section TBD)
BUP — Best unbiased predictor (Section 3.2)
BLUP — Predictor having minimum mean squared prediction error in the class of predictors that are linear and unbiased (with respect to some family of distributions) (Section 3.1)
ERMSPE — Empirical root mean squared prediction error (Section 3.3)
GRF — Gaussian random function (Subsection 2.3)
GP — Gaussian process (Subsection TBD)
IMSPE — Integrated mean squared prediction error (Section 5.2)
LHD — Latin hypercube design (Subsection **??**)
LUP — Linear unbiased predictor (Section 3.2)
MLE — Maximum likelihood estimator
MMSPE — Maximum mean squared prediction error (Section 5.2)
MmLHD — Maximin Latin hypercube design (Subsection TBD)
mARD — Minimum ARD design (Section TBD)
MSPE — Mean squared prediction error (Section 3.2)
REML — Restricted (or residual) maximum likelihood (estimator) (Section 3.3)
RMSPE — Root mean squared prediction error (Equation **??**)
XVE — Cross validated estimator (Section 3.3)

A.2 Symbols

$\mathbf{0}_n$ — $n \times 1$ vector of zeroes
$\mathbf{1}_n$ — $n \times 1$ vector of ones
$(a)^+$ — $\max\{a, 0\}$ for $a \in \mathbb{R}$
$(a)^-$ — $\min\{a, 0\}$ for $a \in \mathbb{R}$
$\lceil a \rceil$ — Smallest integer greater than or equal to a for $a \in \mathbb{R}$
$\ell n(a)$ — Natural logarithm of a, $a > 0$
$|a|$ — Absolute value of a for $a \in \mathbb{R}$
$\binom{n}{j}$ — $n!/(j!(n-j)!)$, for integer j with $0 \leq j \leq n$ is the number of subsets of size j that can be drawn from n distinct objects
$Cov_p[Y(\boldsymbol{x}^1), Y(\boldsymbol{x}^2)]$ — Process model covariance of $Y(\cdot)$
$D_\infty(\mathcal{D})$ — star discrepancy (from the uniform distribution) given by (**??**)
$\det(\boldsymbol{W})$ — Determinant of the square matrix $\boldsymbol{W}$
$\xi^\alpha(\boldsymbol{x}_c)$ — Upper α quantile of the distribution of $y^s(\boldsymbol{x}_c, \boldsymbol{X}_e)$ induced by random environmental variables $\boldsymbol{X}_e$ for fixed control variable $\boldsymbol{x}_c$ (Subsection 1.4.1)
$\boldsymbol{e}_i$ — The i^{th} unit vector, $(0, \ldots, 0, 1, 0, \ldots, 0)^\top$, where 1 is in the i^{th} position.
$E_p[\cdot]$ — Expectation with respect to the process model under consideration
$f_j(\cdot)$ — The jth regression function in the stochastic model for $Y(\cdot)$ (Equation TBD)
$\boldsymbol{I}_n$ — $n \times n$ identity matrix
$I\{E\}$ — indicator function of the event E which is defined to be 1 or 0 as E is true or false (Section **??**)
$\boldsymbol{J}_n$ — $n \times n$ matrix of 1s, i.e., $\boldsymbol{J}_n \equiv \mathbf{1}_n \mathbf{1}_n^\top$
m — Number of outputs in a multiple response application (Subsection 1.4.2)
$\mu(\boldsymbol{x}_c)$ — Mean of $y^s(\boldsymbol{x}_c, \boldsymbol{X}_e)$ induced by random environmental variables $\boldsymbol{X}_e$ for fixed control variable $\boldsymbol{x}_c$ (Subsection 1.4.1)
$N(\mu, \sigma^2)$ — The univariate normal distribution with mean μ and variance σ^2
z^α — Upper α quantile of the standard normal distribution, i.e., $P\{Z \geq z^\alpha\} = \alpha$ where $Z \sim N(0, 1)$
$N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ — The p-dimensional multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$R(\cdot)$ — Correlation function (Section 2.3)
$\boldsymbol{R}$ — $n \times n$ matrix of correlations among the elements of $\boldsymbol{Y}^n$ (Section 3.3)
$\mathbb{R}$ — Real numbers
$\text{rank}(\boldsymbol{W})$ — Rank of the $r \times c$ matrix $\boldsymbol{W}$
$\sigma^2(\boldsymbol{x}_c)$ — Variance of $y(\boldsymbol{x}_c, \cdot)$ induced by the distribution of the environmental and code variables (Subsection 1.4.1)
$\cdot^\top$ — Transpose of a vector or matrix

$\mathrm{tr}(\boldsymbol{W})$ — Trace of the square matrix $\boldsymbol{W}$

T_ν — The univariate t distribution with ν degrees of freedom

t_ν^α — Upper α quantile of the t distribution with ν degrees of freedom, i.e., $P\{W \geq t_\nu^\alpha\} = \alpha$ where $W \sim T_\nu$

$T_p(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ — The p-dimensional multivariate student t distribution with ν degrees of freedom, location vector $\boldsymbol{\mu} \in \mathbb{R}^p$, and positive definite scale matrix $\boldsymbol{\Sigma}$ (Section B)

$T_1(\nu, \mu, \sigma)$ — The univariate student t distribution with ν degrees of freedom, location $\mu \in \mathbb{R}$, and scale parameter $\sigma > 0$, a special case of the $T_p(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution (Section B)

$U(a, b)$ — The uniform distribution over the interval (a, b)

$\boldsymbol{x}_c$ — Vector of control (engineering) variables (Section 2.1)

$\boldsymbol{x}_e$ — Vector of environmental (noise) variables (Section 2.1)

$\boldsymbol{x}_m$ — Vector of model (code) variables (Section 2.1)

$\boldsymbol{x}$ — Vector of *all* input variables to a given computer code including whatever control, environmental, and model variables are required by the code (Section 2.1)

$\mathcal{X}$ — Sample space for the vector of all input variables $\boldsymbol{x}$ (Section 2.1)

$y^s(\cdot)$ — The output of a computer simulator (Section TBD)

$y^p(\cdot)$ — The output of a physical experiment (Section TBD)

$y_i^s(\cdot)/y_i^p(\cdot)$ — The i^{th} of m simulator or physical experiment outputs, $1 \leq i \leq m$ (Subsection TBD)

$[\boldsymbol{X}|\boldsymbol{Y}]$ — Conditional distribution of $\boldsymbol{X}$ given $\boldsymbol{Y}$

$\|\boldsymbol{v}\|_p$ — $\left(\sum_{i=1}^n v_i^p\right)^{1/p}$, the p-norm of the vector $\boldsymbol{v} \in \mathbb{R}^n$

Appendix B
Mathematical Facts

B.1 The Multivariate Normal Distribution

There are several equivalent of ways of defining the multivariate normal distribution. Because we mention both degenerate ("singular") and nondegenerate ("nonsingular") multivariate normal distributions, we will define this distribution by the standard device of describing it indirectly as the distribution that arises by forming a certain function, affine combinations, of independent and identically distributed standard normal random variables.

Definition Suppose $\boldsymbol{Z} = (Z_1, \ldots, Z_r)$ consists of independent and identically distributed $N(0,1)$ random variables, $\boldsymbol{L}$ is an $m \times r$ real matrix, and μ is $m \times 1$ real vector. Then

$$\boldsymbol{W} = (W_1, \ldots, W_m) = \boldsymbol{LZ} + \boldsymbol{\mu}$$

is said to have the *multivariate normal distribution* (associated with $\boldsymbol{\mu}$, $\boldsymbol{L}$).

It is straightforward to compute the mean vector of $(W_1, \ldots, W_m)$ and the matrix of the variances and covariances of the $(W_1, \ldots, W_m)$ in terms of $(\boldsymbol{\mu},\boldsymbol{L})$ as

$$\boldsymbol{\mu} = E\{\boldsymbol{W}\} \text{ and } \operatorname{Cov}\{\boldsymbol{W}\} = E\{(\boldsymbol{W} - \boldsymbol{\mu})(\boldsymbol{W} - \boldsymbol{\mu})^\top\} = \boldsymbol{LL}^\top.$$

As an example, suppose Z_1 and Z_2 are independent $N(0,1)$ and

$$\boldsymbol{W} = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} = \boldsymbol{LZ} + \boldsymbol{\mu} = \begin{pmatrix} 3 & 0 \\ 5 & 0 \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

By construction, $\boldsymbol{W}$ has a multivariate normal distribution. Notice that both W_1 and W_2 can be expressed in terms of the other. In contrast,

$$\begin{pmatrix} W_1 \\ W_2 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

also has a multivariate normal distribution. In the second case we have defined the same number of linearly independent W_is as independent Z_is.

This example illustrates the fundamental dichotomy in multivariate normal distributions. A multivariate normal distribution is *nonsingular* (nondegenerate) if the rows of $\boldsymbol{L}$ are linearly independent, i.e, $\text{rank}(\boldsymbol{L}) = m$, and it is *singular* (degenerate) if the rows of $\boldsymbol{L}$ are linearly dependent, i.e, $\text{rank}(\boldsymbol{L}) < m$.

Suppose $\boldsymbol{W}$ has the nonsingular multivariate normal distribution defined by $\boldsymbol{\mu} \in \mathbb{R}^m$ and $m \times r$ matrix $\boldsymbol{L}$ having rank m. Let

$$\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^\top$$

denote the covariance matrix of $\boldsymbol{W}$. Notice that $\boldsymbol{\Sigma}$ must be symmetric and positive definite (the latter follows because if $\|\cdot\|_2$ denotes Euclidean norm and $\boldsymbol{z} \neq \boldsymbol{0}$, then $\boldsymbol{z}^\top\boldsymbol{\Sigma}\boldsymbol{z} = \boldsymbol{z}^\top\boldsymbol{L}\boldsymbol{L}^\top\boldsymbol{z} = \|\boldsymbol{L}^\top\boldsymbol{z}\|_2^2 > 0$ because $\text{rank}(\boldsymbol{L}) = m$). In this case it can be shown that $\boldsymbol{W} = (W_1, \ldots, W_m)$ has density

$$f(\boldsymbol{w}) = \frac{1}{(2\pi)^{m/2}(\det(\boldsymbol{\Sigma}))^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{\mu})\right\} \tag{B.1.1}$$

over $\boldsymbol{w} \in \mathbb{R}^m$. We denote the fact that $\boldsymbol{W}$ has the nonsingular multivariate normal distribution (B.1.1) by $\boldsymbol{W} \sim N_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. There are numerous algorithms for computing various quantiles associated with multivariate normal distributions. We note, in particular, Dunnett (1989) who provides a FORTRAN 77 program for computing equicoordinate percentage points of multivariate normal distributions having product correlation structure (see also Odeh et al (1988)).

Now suppose $\boldsymbol{W}$ has the singular multivariate normal distribution defined by $\boldsymbol{\mu} \in \mathbb{R}^m$ and $m \times r$ matrix $\boldsymbol{L}$ where $\text{rank}(\boldsymbol{L}) = q < m$. Then $m - q$ rows of $\boldsymbol{L}$ can be expressed as linear combinations of the remaining q rows of $\boldsymbol{L}$ and the corresponding $m - q$ components of $\boldsymbol{W} - \boldsymbol{\mu}$ can be expressed as (the same) linear combinations of the remaining q components of $\boldsymbol{W} - \boldsymbol{\mu}$. Thus, in this case, the support of $\boldsymbol{W}$ is on a hyperplane in a lower dimensional subspace of $\mathbb{R}^m$. Furthermore, the q components of $\boldsymbol{W}$ used to express the remaining variables have a nonsingular multivariate normal distribution with density on $\mathbb{R}^q$.

To illustrate the singular case, consider the toy example above. Marginally, both W_1 and W_2 have proper normal distributions with $W_1 = 3Z_1 + 2 \sim N(2, 9)$ and $W_2 = 5Z_1 \sim N(0, 25)$. This shows that choice of the q variables that have the proper normal distribution is nonunique. Given either W_1 or W_2, the other can be expressed in terms of the first. For example, given W_1, $W_2 = 5(W_1 - 2)/3$ with probability one or given W_2, $W_1 = 2 + 3W_2/5$ with probability one. In the second part of the example, W_1 and W_2 have a nonsingular bivariate normal distribution.

In the text we make use of the following integration formula, which is an application of the fact that (B.1.1) is a density function.

Lemma B.1. For any $n \times 1$ vector $\boldsymbol{v}$ and any $n \times n$ symmetric, positive definite matrix $\boldsymbol{A}$,

$$\int_{\mathbb{R}^n} \exp\left\{-\frac{1}{2}\boldsymbol{w}^\top \boldsymbol{A}^{-1}\boldsymbol{w} + \boldsymbol{v}^\top \boldsymbol{w}\right\} d\boldsymbol{w}$$
$$= (2\pi)^{n/2}(\det(\boldsymbol{A}))^{1/2} \exp\left\{\frac{1}{2}\boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v}\right\}.$$

To prove this formula consider the $N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ multivariate normal density with covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{A}$ and mean $\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{v}$. Then

$$(2\pi)^{n/2}(\det(\boldsymbol{\Sigma}))^{1/2} = \int_{\mathbb{R}^n} \exp\left\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{\mu})\right\} d\boldsymbol{w}$$
$$= \int_{\mathbb{R}^n} \exp\left\{-\frac{1}{2}\boldsymbol{w}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{w} + \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{w} - \frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right\} d\boldsymbol{w}.$$

Substituting for $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ and rearranging terms gives the result. □

Perhaps more usefully, we can interpret the proof of Lemma B.1 as stating that if $\boldsymbol{W}$ has density $f(\boldsymbol{w})$, for which

$$f(\boldsymbol{w}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{w}^\top \boldsymbol{A}^{-1}\boldsymbol{w} + \boldsymbol{v}^\top \boldsymbol{w}\right\}, \text{ then } \boldsymbol{W} \sim N_n[\boldsymbol{A}\boldsymbol{v}, \boldsymbol{A}]. \tag{B.1.2}$$

We also require the following result concerning the conditional distribution of a set of components of the multivariate normal distribution given the remaining ones.

Lemma B.2. (Conditional distribution of the multivariate normal) Suppose that

$$\begin{pmatrix} W_1 \\ W_2 \end{pmatrix} = N_{m+n}\left[\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} \end{pmatrix}\right]$$

where $\boldsymbol{\mu}_1$ is $m \times 1$, $\boldsymbol{\mu}_2$ is $n \times 1$, $\boldsymbol{\Sigma}_{1,1}$ is $m \times m$, $\boldsymbol{\Sigma}_{1,2} = \boldsymbol{\Sigma}_{2,1}^\top$ is $m \times n$, and $\boldsymbol{\Sigma}_{2,2}$ is $n \times n$. Then the conditional distribution of $W_1|W_2$ is

$$N_m\left[\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}(W_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1}\right]. \tag{B.1.3}$$

B.2 The Non-Central Student *t* Distribution

This appendix defines the univariate Student t and multivariate Student t distributions. Throughout, suppose that $\boldsymbol{\mu} \in \mathbb{R}^m$ and $\boldsymbol{\Sigma}$ is a positive definite matrix.

Definition The random vector $\boldsymbol{W} = (W_1, \ldots, W_m)$ with joint probability density

$$f(\boldsymbol{w}) = \frac{\Gamma((\nu+m)/2)}{(\det(\boldsymbol{\Sigma}))^{1/2}(\nu\pi)^{m/2}\Gamma(\nu/2)}\left(1 + \frac{1}{\nu}(\boldsymbol{w}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{\mu})\right)^{-(\nu+m)/2} \tag{B.2.1}$$

over $\boldsymbol{w} \in \mathbb{R}^m$ is said to have the *nonsingular multivariate t distribution* with ν degrees of freedom, location parameter $\boldsymbol{\mu}$, and scale matrix $\boldsymbol{\Sigma}$.

We denote the multivariate t distribution (B.2.1) by $\boldsymbol{W} \sim T_m(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The $T_m(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution has mean vector $\boldsymbol{\mu}$ provided $\nu > 1$ and has covariance matrix $\nu\boldsymbol{\Sigma}/(\nu - 2)$ provided $\nu > 2$. The "usual" univariate t and multivariate t distributions are the special cases of (B.2.1) corresponding to

- $T_1(\nu, 0, 1)$ and
- $T_m(\nu, \mathbf{0}, \boldsymbol{R})$

where $\boldsymbol{R}$ has unit diagonal elements (Tong (1980), Odeh et al (1988) and Dunnett (1989)).

In particular, if $\boldsymbol{X} \sim N_m(\mathbf{0}, \boldsymbol{R})$, where $\boldsymbol{R}$ is as in the previous paragraph, and $\boldsymbol{X}$ is independent of $V \sim \chi^2_\nu$, then

$$\boldsymbol{W} = (W_1, \ldots, W_m) \equiv \left(\frac{X_1}{\sqrt{V/\nu}}, \ldots, \frac{X_m}{\sqrt{V/\nu}} \right) \sim T_p(\nu, 0, \boldsymbol{R}).$$

Some other important relationships concerning the multivariate t distribution are

- If $\boldsymbol{W} \sim T_p(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ then $\boldsymbol{W} - \boldsymbol{\mu} \sim T_p(\nu, \mathbf{0}, \boldsymbol{\Sigma})$.
- When $p = 1$, $W \sim T_1(\nu, \mu, \sigma)$ if and only if $\frac{1}{\sigma}(W - \mu) \sim T_1(\nu, 0, 1)$.
- For arbitrary p, $\mu \in \mathbb{R}^p$, and $p \times p$ positive definite $\boldsymbol{\Sigma}$ with diagonal elements σ_i^2 for $1 \le i \le p$, $\boldsymbol{W} \sim T_p(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ if and only if $\boldsymbol{\Lambda}^{-1}(\boldsymbol{W} - \boldsymbol{\mu}) \sim T_p(\nu, 0, \boldsymbol{R})$ where

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^\top = \text{diag}(\sigma_1, \ldots, \sigma_m).$$

Dunnett (1989) provides an extension of his multivariate normal percentile program to compute equicoordinate percentage points of the multivariate t distribution for the case where the scale matrix has unit variances and a (rank one) product correlation structure.

B.3 Some Results from Matrix Algebra

The following formula for the inverse of a 2×2 partitioned matrix can be found as a special case of Theorem 8.5.11 of Harville (1997), among many other sources.

Lemma B.3 (Inversion of a partitioned matrix-I). Suppose that $\boldsymbol{B}$ is a nonsingular $n \times n$ matrix and

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{D} & \boldsymbol{A}^\top \\ \boldsymbol{A} & \boldsymbol{B} \end{pmatrix}$$

where $\boldsymbol{D}$ is $m \times m$ and $\boldsymbol{A}$ is $n \times m$. Then $\boldsymbol{T}$ is nonsingular if and only if

$$\boldsymbol{Q} = \boldsymbol{D} - \boldsymbol{A}^\top \boldsymbol{B}^{-1} \boldsymbol{A}$$

is nonsingular. In this case, $\boldsymbol{T}^{-1}$ is given by

$$\begin{pmatrix} \boldsymbol{Q}^{-1} & -\boldsymbol{Q}^{-1}\boldsymbol{A}^\top\boldsymbol{B}^{-1} \\ -\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{Q}^{-1} & \boldsymbol{B}^{-1} + \boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top\boldsymbol{B}^{-1} \end{pmatrix}. \tag{B.3.1}$$

That (B.3.1) is $\boldsymbol{T}^{-1}$ can easily be verified by multiplication. To verify the "only if" part of the lemma, see Harville (1997), for example.

The special case of Lemma B.3 corresponding to $\boldsymbol{D} = \boldsymbol{0}$ occurs frequently. If $\boldsymbol{B}$ is a nonsingular $n \times n$ matrix and $\boldsymbol{A}$ is $n \times m$, then

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{A}^\top \\ \underline{\mathrm{A}} & \underline{\mathrm{B}} \end{pmatrix}$$

has an inverse if and only if

$$\underline{\mathrm{A}}^\top \boldsymbol{B}^{-1} \underline{\mathrm{A}}$$

is nonsingular. In this case, $\boldsymbol{T}^{-1}$ is given by

$$\begin{pmatrix} -\left(\boldsymbol{A}^\top \boldsymbol{B}^{-1} \boldsymbol{A}\right)^{-1} & \left(\boldsymbol{A}^\top \boldsymbol{B}^{-1} \boldsymbol{A}\right)^{-1} \boldsymbol{A}^\top \boldsymbol{B}^{-1} \\ \boldsymbol{B}^{-1} \boldsymbol{A} \left(\boldsymbol{A}^\top \boldsymbol{B}^{-1} \boldsymbol{A}\right)^{-1} & \boldsymbol{B}^{-1} - \boldsymbol{B}^{-1} \boldsymbol{A} \left(\boldsymbol{A}^\top \boldsymbol{B}^{-1} \boldsymbol{A}\right)^{-1} \boldsymbol{A}^\top \boldsymbol{B}^{-1} \end{pmatrix}.$$

Lemma B.4. Suppose that $\boldsymbol{B}$ is a nonsingular $n \times n$ matrix, $\boldsymbol{C}$ is nonsingular $m \times m$ matrix, and $\boldsymbol{A}$ is an arbitrary $n \times m$ matrix such that $(\boldsymbol{A}^\top \boldsymbol{B}^{-1} \boldsymbol{A} + \boldsymbol{C})^{-1}$ is nonsingular. Then $\left(\boldsymbol{B} + \boldsymbol{A}\boldsymbol{C}^{-1}\boldsymbol{A}^\top\right)$ is $(n \times n)$ nonsingular with inverse given by

$$\left(\boldsymbol{B} + \boldsymbol{A}\boldsymbol{C}^{-1}\boldsymbol{A}^\top\right)^{-1} = \boldsymbol{B}^{-1} - \boldsymbol{B}^{-1}\boldsymbol{A}\left(\boldsymbol{A}^\top \boldsymbol{B}^{-1}\boldsymbol{A} + \boldsymbol{C}\right)^{-1} \boldsymbol{A}^\top \boldsymbol{B}^{-1}.$$

Proof: Multiply the right-hand expression by $\left(\boldsymbol{B} + \boldsymbol{A}\boldsymbol{C}^{-1}\boldsymbol{A}^\top\right)$ and verify that it is the identity. □

Lemma B.5. Suppose that $a \neq 0$ and $b \neq -a/n$, then

- $(a\boldsymbol{I}_n + b\boldsymbol{J}_n)^{-1} = \left(\frac{1}{a}\boldsymbol{I}_n - \frac{b}{a(a+nb)}\boldsymbol{J}_n\right)$ and
- $\det\left((a\boldsymbol{I}_n + b\boldsymbol{J}_n)\right) = a^{n-1}(a + nb)$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and $\boldsymbol{J}_n$ is the $n \times n$ matrix of ones.

References

Abrahamsen P (1997) A review of gaussian random fields and correlation functions. Tech. Rep. 917, Norwegian Computing Center, Box 114, Blindern, N0314 Oslo, Norway

Adler RJ (1981) The Geometry of Random Fields. J. Wiley, New York

Adler RJ (1990) An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes. Institute of Mathematical Statistics, Hayward, California

Allen DM (1974) The relationship between variable selection and data augmentation and a method for prediction. Technometrics 16:125–127

An J, Owen AB (2001) Quasi-regression. Journal of Complexity 17:588–607

Atkinson AC, Donev AN (1992) Optimum experimental designs. Oxford University Press

Bates RA, Buck RJ, Riccomagno E, Wynn HP (1996) Experimental design and observation for large systems. Journal of the Royal Statistical Society B 58:77–94

Berger JO, De Oliveira V, Sansó B (2001) Objective bayesian analysis of spatially correlated data. Journal of the American Statistical Association 96:1361–1374

Berk R, Bickel P, Campbell K, Fovell R, Keller-McNulty S, Kelly E, Linn R, Park B, Perelson A, Rouphail N, Sacks J, Schoenberg F (2002) Workshop on statistical approaches for the evaluation of complex computer models. Statistical Science 17(2):173–192

Bernardo MC, Buck RJ, Liu L, Nazaret WA, Sacks J, Welch WJ (1992) Integrated circuit design optimization using a sequential strategy. IEEE Transactions on Computer-Aided Design 11:361–372

Birk DM (1997) An Introduction to Mathematical Fire Modeling. Technomic Publishing, Lancaster, PA

Bochner S (1955) Harmonic Analysis and the Theory of Probability. University of California Press, Berkeley

Booker AJ, Dennis JE, Frank PD, Serafini DB, Torczon V (1997) Optimization using surrogate objectives on a helicopter test example. Tech. Rep. SSGTECH-97-027, Boeing Technical Report

Booker AJ, Dennis JE, Frank PD, Serafini DB, Torczon V, Trosset MW (1999) A rigorous framework for optimization of expensive functions by surrogates. Structural Optimization 17:1–13

Box G, Hunter W, Hunter J (1978) Statistics for Experimenters. J. Wiley, New York

Box GE, Draper NR (1987) Empirical model-building and response surfaces. John Wiley & Sons, New York

Box GE, Jones S (1992) Split-plot designs for robust product experimentation. Journal of Applied Statistics 19:3–26

Bratley P, Fox BL, Niederreiter H (1994) Algorithm 738: Programs to generate niederreiter's low-discrepancy sequences. ACM Transactions on Mathematical Software 20:494–495

Butler NA (2001) Optimal and orthogonal latin hypercube designs for computer experiments. Biometrika 88:847–857

Campbell K (2001) Functional sensitivity analysis of computer model output. In: Proceedings of the 7th Army Conference on Statistics, ?????

Campbell KS, McKay MD, Williams BJ (2005) Sensitivity analysis when model outputs are functions (tutorial). In: Hanson KM, Hemez FM (eds) Proceedings of the SAMO 2004 Conference on Sensitivity Analysis, http://library.lanl.gov/ccw/samo2004/, Los Alamos National Laboratory, Los Alamos, pp 81–89

Campbell KS, McKay MD, Williams BJ (2006) Sensitivity analysis when model outputs are functions. Reliability and System Safety 91:1468–472

Campolongo F, Cariboni J, Saltelli A (2007) An effective screening design for sensitivity analysis of large models. Environmental Modelling & Software 22:1509–1518

Campolongo F, Saltelli A, Cariboni J (2011) From screening to quantitative sensitivity analysis. a unified approach. Computer Physics Communications 43:39–52

Chang PB, Williams BJ, Notz WI, Santner TJ, Bartel DL (1999) Robust optimization of total joint replacements incorporating environmental variables. Journal of Biomechanical Engineering 121:304–310

Chang PB, Williams BJ, Bawa Bhalla KS, Belknap TW, Santner TJ, Notz WI, Bartel DL (2001) Robust design and analysis of total joint replacements: Finite element model experiments with environmental variables. Journal of Biomechanical Engineering 123:239–246

Chapman WL, Welch WJ, Bowman KP, Sacks J, Walsh JE (1994) Arctic sea ice variability: Model sensitivities and a multidecadal simulation. Journal of Geophysical Research C 99(1):919–936

Chen RB, Wang W, Wu CFJ (2011) Building surrogates with overcomplete bases in computer experiments with applications to bistable laser diodes. IEE Transactions 182:978–988

Chen W, Jin R, Sudjianto A (2005) Analytical variance-based global sensitivity analysis in simulation-based design under uncertainty. Journal of Mechanical Design 127:875–876

Chen W, Jin R, Sudjianto A (2006) Analytical global sensitivity analysis and uncertainty propogation for robust design. Journal of Quality Technology 38:333–348

Cooper LY (1980) Estimating safe available egress time from fires. Tech. Rep. 80-2172, National Bureau of Standards, Washington D.C.

Cooper LY (1997) Ventcf2: An algorithm and associated fortran 77 subroutine for calculating flow through a horizontal ceiling/floor vent in a zone-type compartmental fire model. Fire Safety Journal 28:253–287

Cooper LY, Stroup DW (1985) Aset–a computer program for calculating available safe egress time. Fire Safety Journal 9:29–45

Craig PC, Goldstein M, Rougier JC, Seheult AH (2001) Bayesian forecasting for complex systems using computer simulators. Journal of the American Statistical Association 96:717–729

Cramér H, Leadbetter MR (1967) Stationary and Related Stochastic Processes. J. Wiley, New York

Cressie NA (1993) Statistics for Spatial Data. J. Wiley, New York

Currin C, Mitchell TJ, Morris MD, Ylvisaker D (1991) Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. Journal of the American Statistical Association 86:953–963

Dean AM, Voss D (1999) Design and Analysis of Experiments. Spring-Verlag, New York, New York

Dhrymes PJ (2005) Moments of truncated (normal) distributions, unpublished note

Draguljić D, Santner TJ, Dean AM (2012) Non-collapsing spacing-filling designs for bounded polygonal regions. Technometrics 54:169–178

Draper NR, Smith H (1981) Applied Regression Analysis, 2nd Ed. J. Wiley, New York

Dunnett CW (1989) Multivariate normal probability integrals with product correlation structure. correction: **42**, 709. Applied Statistics 38:564–579

Fang KT, Lin DKJ, Winker P, Zhang Y (2000) Uniform design: theory and application. Technometrics 42:237–248

Fang KT, Li R, Sudjianto A (2005) Design and Modeling for Computer Experiments. Chapman and Hall

Fuller WA, Hasza DP (1981) Properties of predictors for autoregressive time series (corr: V76, 1023–1023). Journal of the American Statistical Association 76:155–161

Gibbs MN (1997) Bayesian gaussian processes for regression and classification. PhD thesis, Cambridge University, Cambridge, UK

Gneiting T (2002) Compactly supported correlation functions. Journal of Multivariate Analysis

Golub GH, Heath M, Wahba G (1979) Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21:215–223

Halton JH (1960) On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer Math 2:84–90

Handcock MS (1991) On cascading latin hypercube designs and additive models for experiments. Communications Statistics—Theory Methods 20:417–439

Handcock MS, Stein ML (1993) A bayesian analysis of kriging. Technometrics 35:403–410

Handcock MS, Wallis JR (1994) An approach to statistical spatial-temporal modeling of meterological fields. Journal of the American Statistical Association 89:368–390

Harville DA (1974) Bayesian inference for variance components using only error contrasts. Biometrika 61:383–385

Harville DA (1977) Maximum likelihood approaches to variance component estimation and to related problems (with discussion). Journal of the American Statistical Association 72:320–340

Harville DA (1997) Matrix algebra from a statistician's perspective. Springer-Verlag, New York, New York

Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Verlag, New York

Hayeck GT (2009) The kinematics of the upper extremity and subsequent effects on joint loading and surgical treatment. PhD thesis, Cornell University, Ithaca, NY USA

Hedayat A, Sloane N, Stufken J (1999) Orthogonal Arrays. Springer-Verlag, New York, New York

Helton JC (1993) Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. Reliability Engineering and System Safety 42:327–367

Hickernell FJ (1998) A generalized discrepancy and quadrature error bound. Math Comp 67:299–322

Hoeffding W (1948) A class of statistics with asymptotically normal distribution. The Annals of Mathematical Statistics

Jeffreys H (1961) Theory of Probability. Oxford University Press, London

John JA (1987) Cyclic Designs. Chapman & Hall Ltd, New York

John PWM (1980) Incomplete Block Designs. M. Dekker, Inc., New York

Johnson ME, Moore LM, Ylvisaker D (1990) Minimax and maximin distance designs. Journal of Statistical Planning and Inference 26:131–148

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black–box functions. Journal of Global Optimization 13:455–492

Journel AG, Huijbregts CJ (1978) Mining Geostatistics. Academic Press, London

Journel AG, Huijbregts CJ (1979) Mining Geostatistics. Academic Press, New York

Kackar RN, Harville DA (1984) Approximations for standard errors of estimators of fixed and random effects in mixed linear models. Journal of the American Statistical Association 87:853–862

Kaufman C, Bingham D, Habib S, Heitmann K, Frieman J (2011) Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. The Annals of Applied Statistics 5:24702492

Kennedy MC, O'Hagan A (2001) Bayesian calibration of computer models (with discussion). Journal of the Royal Statistical Society B 63:425–464

Kozintsev B (1999) Computations with gaussian random fields. PhD thesis, Department of Mathematics and Institute for Systems Research, University of Maryland, College Park, MD USA

Kozintsev B, Kedem B (2000) Generation of 'similar' images from a given discrete image. Journal of Computational and Graphical Statistics 9:286–302

Kreyszig E (1999) Advanced engineering mathematics. John Wiley, New York

Lehman J (2002) Sequential design of computer experiments for robust parameter design. PhD thesis, Department of Statistics, Ohio State University, Columbus, OH USA

Lemieux C (2009) Monte Carlo and Quasi-Monte Carlo sampling,. Springer, New York, NY, USA

Lempert R, Williams BJ, Hendrickson J (2002) Using global sensitivity analysis to understand policy effects and to aid in new policy contruction in integrated assessment models. Tech. rep., RAND

Linkletter C, Bingham D, Hengartner N, Higdon D, Ye KQ (2006) Variable selection for Gaussian process models in computer experiments. Technometrics 48:478–490

Loeppky JL, Sacks J, Welch WJ (2009) Choosing the sample size of a computer experiment: A practical guide. Technometrics 51(4):366–376

Loeppky JL, Williams BJ, Moore LM (2011) Gaussian process model for mixture experiments. Tech. rep., University of British Columbia

Loeppky JL, Moore LM, Williams BJ (2012) Projection array based designs for computer experiments. Journal of Statistical Planning and Inference 142:1493–1505

Lynn RR (1997) Transport model for prediction of wildfire behavior. Tech. Rep. LA13334-T, Los Alamos National Laboratory

Matérn B (1960) Spatial variation. PhD thesis, Meddelanden fran Statens Skogsforskningsinstitut, vol. 49, Num. 5

Matérn B (1986) Spatial Variation (Second Edition). Springer-Verlag, New York

Matheron G (1963) Principles of geostatistics. Economic Geology 58:1246–1266

McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21:239–245

Mitchell TJ, Morris MD, Ylvisaker D (1990) Existence of smoothed stationary processes on an interval. Stochastic Processes and their Applications 35:109–119

Mockus J, Eddy W, Mockus A, Mockus L, Reklaitis G (1997) Bayesian Heuristic Approach to Discrete and Global Optimization: Algorithms, Visualization, Software, and Applications. Kluwer Academic, New York

Montgomery GP, Truss LT (2001) Combining a statistical design of experiments with formability simulations to predict the formability of pockets in sheet metal parts. Society of Automotive Engineers 2001-01-1130

Moon H (2010) Design and analysis of computer experiments for screening input variables. PhD thesis, Department of Statistics, The Ohio State University, Columbus, Ohio USA

Moon H, Santner TJ, Dean AM (2012) Two-stage sensitivity-based group screening in computer experiments. Technometrics 54(4):376–387

Morris MD (1991) Factorial sampling plans for preliminary computational experiments. Technometrics 33:161–174

Morris MD, Mitchell TJ (1995) Exploratory designs for computational experiments. Journal of Statistical Planning and Inference 43:381–402

Neal RM (2003) Slice sampling (with discussion). Annals of Statistics 31:705—767

Nelder JA, Mead R (1965) A simplex method for function minimization. Computer Journal 7:308–313

Niederreite H (1988) Low-discrepancy and low-dispersion sequences. Journal of Number Theory 30:51–70

Niederreiter H (1992) Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia

Oakley JE (2002) Eliciting Gaussian process priors for complex computer codes. The Statistician 51:81–97

Oakley JE (2009) Decision-theoretic sensitivity analysis for complex computer models. Technometrics 5(2):121–129

Odeh R, Davenport J, Pearson N (eds) (1988) Selected Tables in Mathematical Statistics, vol 11. American Mathematical Society

O'Hagan A, Haylock RG (1997) Bayesian uncertainty analysis and radiological protection. In: Barnett V, Turkman KF (eds) Statistics for the Environment, vol 3, J. Wiley, pp 109–128

O'Hagan A, Kennedy MC, Oakley JE (1999) Uncertainty analysis and other inference tools for complex computer codes. In: Bernardo JM, Berger JO, Dawid AP, Smith AFM (eds) Bayesian Statistics, vol 6, Oxford University Press, pp 503–524

Ong K, Santner T, Bartel D (2008) Robust design for acetabular cup stability accounting for patient and surgical variability,. Journal of Biomechanical Engineering 130:031,001

Owen AB (1992a) A central limit theorem for Latin hypercube sampling. Journal of the Royal Statistical Society, Series B: Methodologic al 54:541–551

Owen AB (1992b) Orthogonal arrays for computer experiments, integration and visualization (Corr: 93V3 p261). Statistica Sinica 2:439–452

Owen AB (1995) Randomly permuted (t, m, s)-nets and (t, s) sequences. In: Niederreiter H, Shiue PJS (eds) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Springer-Verlag, New York, pp 299–317

Patterson HD, Thompson R (1971) Recovery of interblock information when block sizes are unequal. Biometrika 58:545–554

Patterson HD, Thompson R (1974) Maximum likelihood estimation of components of variance. In: Proceedings of the 8th International Biometric Conference, Biometric Society, Washington DC, pp 197–207

Prasad NGN, Rao JNK (1990) The estimation of the mean squared error of small-area estimators. Journal of the American Statistical Association 85:163–171

Pujol G (2009) Simplex-based screening designs for estimating metamodels. Reliability Engineering and System Safety 94:1156–1160

Pukelsheim F (1993) Optimal Design of Experiments. J. Wiley, New York

Raghavarao D (1971) Constructions and Combinatorial Problems in Design of Experiments. J. Wiley, New York

Reese CS, Wilson AG, Hamada M, Martz F, Ryan K (2000) Integrated analysis of computer and physical experiments. Tech. Rep. LA-UR-00-2915, Sandia Laboratories

Rinnooy Kan AHG, Timmer GT (1984) A stochastic approach to global optimization. In: Boggs PT, Byrd RH, Schnabel RB (eds) Optimization 84: Proceedings of the SIAM Conference on Numerical Optimization, SIAM, Philadelphia, pp 245–262

Ripley BD (1981) Spatial Statistics. J. Wiley & Sons, New York

Robert CP, Casella G (1999) Monte Carlo Statistical Methods. Springer-Verlag, New York

Rodríguez-Iturbe I, Mejía JM (1974) The design of rainfall networks in time and space. Water Resources Research 10:713–728

Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Statistical Science 4:409–423

Sahama AR, Diamond NT (2001) Sample size considerations and augmentation of computer experiments. Journal of Statistical Computation and Simulation 68(4):307–319

Saltelli A, Sobol´ IM (1995) About the use of rank transformation in sensitivity analysis of model output. Reliability Engineering and System Safety 50:225239

Saltelli A, Chan K, Scott E (2000) Sensitivity Analysis. John Wiley & Sons, Chichester

Saltelli A, Tarantola S, Campolongo F, Ratto M (2004) Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models. John Wiley & Sons Ltd , Chichester

Silvey SD (1980) Optimal design: An introduction to the theory for parameter. Chapman & Hall Ltd, New York

Sobol´ IM (1967) Distribution of points in a cube and approximate evaluation of integrals. USSR Comput Maths Math Phys 7:86–112

Sobol´ IM (1976) Uniformly distributed sequences with an additional uniform property. USSR Comput Maths Math Phys 16:236–242

Sobol´ IM (1990) Sensitivity estimates for non-linear mathematical models. Matematicheskoe Modelirovanie 2:112–118

Sobol´ IM (1993) Sensitivity analysis for non-linear mathematical models. Mathematical Model Comput Exp 1:407–414

Stein ML (1987) Large sample properties of simulations using latin hypercube sampling. Technometrics 29:143–151

Stein ML (1999) Interpolation of Spatial Data: some theory for kriging. Springer-Verlag, New York

Stinstra E, den Hertog D, Stehouwer P, Vestjens A (2003) Constrained maximin designs for computer experiments. Technometrics 45(4):340–346

Stone M (1974) Cross-validatory choice and assessment of statistical predictions (with discussion) (corr: 1976, vol 38, 102). Journal of the Royal Statistical Society B 36:111–147

Stone M (1977) An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. Journal of the Royal Statistical Society B 39:44–47

Street AP, Street DJ (1987) Combinatorics of experimental design. Oxford University Press, Oxford

Sun F, Dean AM, Santner TJ (2014) One-at-a-time designs for estimating elementary effects of simulator experiments with non-rectangular input regions. Submitted

Svenson J, Santner T, Dean A, Moon H (2013) Estimating sensitivity indices based on gaussian process metamodels with compactly supported correlation functions. To appear in *Journal of Statistical Planning and Inference*

Svenson JD (2011) Computer experiments: Multiobjective optimization and sensitivity analysis. PhD thesis, Department of Statistics, The Ohio State University, Columbus, Ohio USA

Tang B (1993) Orthogonal array-based latin hypercubes. Journal of the American Statistical Association 88:1392–1397

Tang B (1994) A theorem for selecting OA-based latin hypercubes using a distance criterion. Communications in Statistics - Theory and Methods 23:2047–2058

Tong YL (1980) Probability Inequalities in Multivariate Distributions. Academic Press, New York

Trosset MW (1999) Approximate maximin distance designs. In: ASA Proceedings of the Section on Physical and Engineering Sciences, American Statistical Association (Alexandria, VA), pp 223–227

Trosset MW, Padula AD (2000) Designing and analyzing computational experiments for global optimization. Tech. Rep. 00-25, Department of Computational & Applied Mathematics, Rice University

Van Der Vaart AW (1998) Asymptotic Statistics. Cambridge University Press, Cambridge, U.K.

Vecchia AV (1988) Estimation and identification for continuous spatial processes. Journal of the Royal Statistical Society B 50:297–312

Wahba G (1980) Spline bayes, regularization, and generalized cross-validation for solving approximation problems with large quantities of noisy data. In: Proceedings of the International Conference on Approximation Theory in Honor of George Lorenz, Academic Press, Austin, Texas

Walton W (1985) Aset–b: A room fire program for personal computers. Tech. Rep. 85-3144-1, National Bureau of Standards, Washington D.C.

Welch WJ (1985) Aced: Algorithms for the construction of experimental designs. The American Statistician 39:146

Welch WJ, Buck RJ, Sacks J, Wynn HP, Mitchell TJ, Morris MD (1992) Screening, predicting, and computer experiments. Technometrics 34:15–25

Wiens DP (1991) Designs for approximately linear regression: Two optimality properties of uniform designs. Statistics and Probability Letters 12:217–221

Williams BJ (2001) Perk–parametric empirical kriging with examples. Tech. Rep. 678, Department of Statistics, The Ohio State University

Williams BJ, Santner TJ, Notz WI (2000) Sequential design of computer experiments to minimize integrated response functions. Statistica Sinica 10:1133–1152

Wu CFJ, Hamada M (2000) Experiments: Planning, Analysis, and Parameter Design Optimization. J. Wiley, New York

Yaglom AM (1962) Introduction to the Theory of Stationary Random Functions. Dover, New York

Ye KQ (1998) Orthogonal column latin hypercubes and their application in computer experiments. Journal of the American Statistical Association 93:1430–1439

Ye KQ, Li W, Sudjianto A (2000) Algorithmic construction of optimal symmetric Latin hypercube designs. Journal of Statistical Planning and Inference 90(1):145–159

Zimmerman DL, Cressie NA (1992) Mean squared prediction error in the spatial linear model with estimated covariance parameters. Annals of the Institute of Statistical Mathematics 44:27–43