Semiparametric Techniques for Response Surface Methodology

Stephanie M. Pickle

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Jeffrey B. Birch, Chair
Timothy J. Robinson, Co-Chair
Samantha Prins
Dan Spitzner
G. Geoffrey Vining

June 28, 2006
Blacksburg, Virginia

Keywords: Response Surface Methodology, Semiparametric Regression, Robust Parameter Design, Genetic Algorithm

Semiparametric Techniques for Response Surface Methodology

Stephanie M. Pickle

(ABSTRACT)

Many industrial statisticians employ the techniques of Response Surface Methodology (RSM) to study and optimize products and processes. A second-order Taylor series approximation is commonly utilized to model the data; however, parametric models are not always adequate. In these situations, any degree of model misspecification may result in serious bias of the estimated response. Nonparametric methods have been suggested as an alternative as they can capture structure in the data that a misspecified parametric model cannot. Yet nonparametric fits may be highly variable especially in small sample settings which are common in RSM. Therefore, semiparametric regression techniques are proposed for use in the RSM setting. These methods will be applied to an elementary RSM problem as well as the robust parameter design problem.

Dedication

To Wayne and my family for their love, support, encouragement, and patience.

Acknowledgments

I would like to express my deepest appreciation to my advisor, Dr. Jeffrey Birch, for his support and guidance throughout this research. I would also like to thank my co-advisor, Dr. Timothy Robinson, for his valuable advice and assistance. Many thanks also goes to my committee, Dr. Samantha Prins, Dr. Dan Spitzner, and Dr. G. Geoffrey Vining, for their helpful comments and suggestions. I would also like to thank the staff and graduate students of the Virginia Polytechnic Institute & State University Department of Statistics for their moral support.

I would also like to express my gratitude to Dr. Deborah Olsen for her advice and support. Special thanks are also due to the Institutional Research staff who were always good for a laugh when I needed it.

Contents

List of Figures

List of Tables

List of Acronyms

AED	Average Euclidean Distance
AFE	Average Number of Function Evaluations
ASE	Average Squared Error
ASEL	Average Squared Error Loss
ASEM	Average Squared Error for Mean
ASEV	Average Squared Error for Variance
ASDT	Average Squared Distance from Target
AVEMSE	Average Mean Squared Error
CCD	Central Composite Design
ED	Euclidean Distance
EWLLR	Estimated Weighted Local Linear Regression
EWLS	Estimated Weighted Least Squares
FE	Number of Function Evaluations
GA	Genetic Algorithm
DMRR	Dual Model Robust Regression
KER	Kernel Regression
LLR	Local Linear Regression
LPR	Local Polynomial Regression
MC	Monte Carlo
MCA	Monte Carlo Average
MCMSE	Monte Carlo Mean Squared Error

MMRR	Means Model Robust Regression
MRR1	Model Robust Regression 1
MRR2	Model Robust Regression 2
MSE	Mean Squared Error
OLS	Ordinary Least Squares
RPD	Robust Parameter Design
RSM	Response Surface Methodology
SDT	Squared Distance from Target
SEL	Squared Error Loss
SIMSE	Simulated Integrated Mean Squared Error
SIMSEM	Simulated Integrated Mean Squared Error for Mean
SIMSEV	Simulated Integrated Mean Squared Error for Variance
VMRR	Variance Model Robust Regression

Chapter 1

Introduction

Originally proposed by Box and Wilson (1951), Response Surface Methodology (RSM) is a sequential experimental strategy used by many industrial statisticians to study and optimize products or processes. RSM consists of three main phases the first of which involves experimental design strategies. In this phase, the researcher must choose a proper design that will allow for adequate estimation of the relationship between the explanatory variables and one or more responses. During the second phase, regression modeling techniques are employed to complete such an estimation. Finally, optimization methods are utilized to identify the settings of the explanatory variables which optimize the response(s). The optimization results found in the final phase heavily depend on the estimated models built in the second phase. Therefore, our research will focus on modeling techniques for RSM with some mention of optimization methods. Throughout our work, we will assume that a satisfactory experimental design has been constructed and the data have already been collected.

Historically, RSM involves running a series of small experiments and modeling the data parametrically to find the optimal settings. However, in many industrial settings, parametric models may not adequately represent the true relationships between the variables. For this

reason, classic RSM may not be appropriate for all applications, but it does in fact provide the foundation for several. We propose new methods that extend the ideas of classic RSM to include new advances in regression and optimization.

When modeling the data parametrically, researchers must make certain assumptions about the relationship between the regressors and the response(s). Researchers tend to assume the relationship is not extremely complex and that a first- or second-order polynomial provides an adequate approximation of the true underlying function. Therefore, the data are usually fit with at most a second-order model. In practice, however, these relationships are not always so well behaved. Process means, and especially process variances, may be of such complexity that they cannot be adequately modeled parametrically, especially via first- or second-order models. Furthermore, any degree of model misspecification can lead to highly biased estimates and miscalculated optimal regressor settings.

Recently, nonparametric regression techniques have been proposed to overcome the issues associated with parametric models [see Vining and Bohn (1998) and Anderson-Cook and Prewitt (2005)]. In general, nonparametric smoothing techniques use curves to describe the relationship between the response(s) and regressors without any parameters. Nonparametric methods can provide superior fits by capturing structure in the data that a misspecified parametric model cannot. Unfortunately, in sparse data situations, which are typically the case with response surface experiments, these techniques often result in highly variable estimates. Moreover, nonparametric methods ignore any knowledge the researcher may have about the relationship between the variables.

Mays, Birch, and Starnes (2001) investigate methods which are essentially hybrids of the parametric and nonparametric methods. These semiparametric approaches produce estimated functions which are characterized by lower bias than the parametric approaches and lower variance than the nonparametric approaches. These methods were originally developed for situations where the response information was obtained at values of the regressors

uniformly placed over the design space whereas a main underpinning of RSM is the use of cost-efficient experimental designs with strategically placed design points. The goal of our research is to adapt these semiparametric techniques to the RSM setting.

Even though some real-world applications involve more than one response, the most basic RSM problems involve only a single response. We will first apply the semiparametric techniques to the elementary situation with one response of interest and the assumption of constant variance. We will then investigate the case in which the variance is no longer assumed to be constant. In this case, we are interested in not only the process mean but also the process variance. In fact, in many industrial applications, the process variance is often the main culprit of poor product and process quality. This problem is commonly referred to as the dual model problem. Although not studied in this research, our methods could also easily be extended to the multiple response problem.

Once the response surface has been modeled, the goal becomes finding the operating conditions for the explanatory variables such that the response(s) is optimized. The RSM literature is rich with analytic optimization techniques for parametric models. Yet most of the methods are based on gradient techniques which require continuous functions with derivatives for the estimated response(s). Since the response estimates from nonparametric and semiparametric methods do not result in closed form expressions, these optimization routines are no longer applicable. For their nonparametric models, Vining and Bohn (1998) utilize a simplex search which does not require the calculation of derivatives; however, simplex methods tend to be time consuming and often only find local optima (Haupt and Haupt, 2004). Therefore, we propose the use of genetic algorithms for optimization.

The genetic algorithm (GA), originally proposed by Holland (1975), has become a popular optimization technique, especially for functions that do not have known parametric forms. Instead of using derivatives to find the optimal solutions, the GA uses the principles of genetics and evolutionary concepts such as selection, crossover, and mutation. Furthermore,

the GA uses an intelligent, sequential search strategy which enables users to find global solutions more efficiently (Goldberg, 1989). Thus, we will use the GA for process optimization in this research.

This dissertation is organized as follows: Chapter 2 reviews the current modeling techniques used in RSM, parametric regression and nonparametric regression. In Chapter 3, the nonparametric approach is studied in more detail as we investigate the degrees of freedom for the method. Chapter 4 addresses the basic RSM scenario in which we have a single response of interest and the homogeneity of the variances assumption is valid. Our semiparametric technique will be introduced and used to model the response surface. In Chapter 5, our semiparametric technique is applied to the dual model problem which involves the estimation of both the process mean and process variance. Examples from the RSM literature and simulation studies will be used to compare the performance of the modeling techniques in both Chapters 4 and 5. In Chapter 6, we apply the genetic algorithm to semiparametric optimization and compare its performance to the Nelder-Mead simplex optimization method. Finally, Chapter 7 contains a summary of completed work and potential areas of future research.

Chapter 2

Current Modeling Techniques for Response Surface Methodology

2.1 Introduction

Many industrial statisticians, engineers, and other researchers employ the techniques of Response Surface Methodology (RSM), a sequential experimental strategy originally proposed by Box and Wilson (1951). RSM combines ideas from experimental design, regression model building, and optimization to study and optimize products or processes.

The most elementary RSM problem is one in which the researcher studies a single response of interest. The relationship between the k explanatory variables, $x_1, x_2, \ldots, x_k$, and the response, y, is:

$$y = f(x_1, x_2, \ldots, x_k) + \varepsilon,$$

where f is the true response function and ε is the error term that represents sources of variability not accounted for in f. The form of the true response function is unknown and

may be very complicated. The error term is assumed to have a normal distribution with mean zero and constant variance σ^2. Thus, the response function of y is also the mean function of y.

Since the true relationship between the explanatory variables and the response is unknown, we must approximate the form of f. The function must be well estimated as misspecification of the model form can have serious implications in process optimization. Classic RSM utilizes parametric regression models for the function approximation and more recently nonparametric modeling techniques have been proposed. We will now describe these current methods, parametric and nonparametric, in detail.

2.2 Parametric Approach

The traditional approach to modeling the relationship between the explanatory variables and the response is to assume that the underlying functional form can be expressed parametrically. A second-order Taylor series approximation, given by:

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i<j} \beta_{ij} x_i x_j + \varepsilon,$$

where the β's are the regression coefficients to be estimated, is commonly employed in RSM. When n observations are observed, the second-order linear model may be expressed in matrix notation as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\mathbf{y}$ is a $n \times 1$ vector of responses, $\mathbf{X}$ is a $n \times \left(1 + 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}\right)$ model matrix, $\boldsymbol{\beta}$ is a $\left(1 + 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}\right) \times 1$ vector of unknown parameters, and $\boldsymbol{\varepsilon}$ is the $n \times 1$ vector of random errors.

The method of ordinary least squares (OLS) provides uniform minimum variance unbiased estimates (UMVUE) of the parameters in $\boldsymbol{\beta}$ such that the sum of squared errors, given as:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

where $\hat{y}_i = \mathbf{x}_i' \hat{\boldsymbol{\beta}}$ and $\mathbf{x}_i'$ is the i^{th} row of the $\mathbf{X}$ matrix, is minimized. The estimated responses can be written as:

$$\hat{\mathbf{y}}^{(OLS)} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}^{(OLS)}\mathbf{y},$$

where the $n \times n$ matrix $\mathbf{H}^{(OLS)}$ is commonly referred to as the ordinary least squares "HAT" matrix since it transforms the observed y values into the $\hat{y}$ values. For more details on ordinary least squares and the HAT matrix, see Myers (1990).

2.3 Nonparametric Approach

Situations may arise in which the relationship between the explanatory variables and the response is not adequately modeled parametrically via second-order polynomials. In these situations, any degree of model misspecification may result in serious bias of the estimated response. Furthermore, the optimal regressor settings may be miscalculated. For such situations, nonparametric methods have recently been suggested as they can capture structure in the data that a misspecified parametric model cannot.

Nonparametric smoothing techniques use curves to describe the relationship between the explanatory variables and the response without any parameters. Thus, the response is modeled as:

$$y = h(x_1, x_2, \ldots, x_k) + \varepsilon,$$

where h is assumed to have an unknown but reasonably smooth form. Similar to parametric regression, the estimator is a linear combination of the response values; however, the weighting schemes in some nonparametric regression methods assign more weight to observations

closest to the point of prediction. The nonparametric fit is more flexible than the parametric fit as it is not confined to the user's specified form.

Myers (1999) suggests the use of nonparametric Response Surface Methodology (NPRSM) in the following three scenarios:

(i) The researcher is interested in optimizing a response.

(ii) The researcher is less interested in an interpretive function (i.e., interpreting the estimated regression coefficients) and more interested in studying the shape of the response surface.

(iii) The functional form of the relationship between the explanatory variables and the response is not well behaved.

Several fitting techniques have been proposed in the nonparametric regression literature such as kernel regression [see for example Nadaraya (1964), Watson (1964), Priestley and Chao (1972), and Gasser and Müller (1984)], local polynomial models [see for example Fan and Gijbels (1996) and Fan and Gijbels (2000)], spline-based smoothers, and series-based smoothers [see for example Ruppert, Wand, and Carrol (2003)]. Details of two popular methods, kernel regression and local polynomial regression, are presented in the next sections.

2.3.1 Kernel Regression

As previously mentioned, nonparametric methods estimate the regression function using a weighted average of the data. For example, the kernel regression estimate of the response at the point of interest x_0 is a weighted average of the responses:

$$\hat{y}_0^{(KER)} = \frac{\sum_{i=1}^{n} h_{0i}^{(KER)} y_i}{\sum_{i=1}^{n} h_{0i}^{(KER)}},$$

where $h_{0i}^{(KER)}$ represents the weights. A common weighting scheme proposed by Nadaraya (1964) and Watson (1964) in which the weight associated with the i^{th} response at prediction point x_0 is given by:

$$h_{0i}^{(KER)} = \frac{K\left(\frac{x_0 - x_i}{b}\right)}{\sum_{i=1}^{n} K\left(\frac{x_0 - x_i}{b}\right)},$$

where K is an univariate kernel function and b is the bandwidth.

The kernel function is taken to be some appropriately chosen decreasing function in $|x_0 - x_i|$ such that observations close to x_0 receive more weight than observations far from x_0. Kernel functions are often chosen to be nonnegative and symmetric about zero. Some common kernel functions include the Gaussian, uniform, and Epanechnikov kernels. For more details on these and other kernel functions, see Härdle (1990). Studies have shown that the choice of kernel function is not crucial to the performance of the estimator (Simonoff, 1996). Thus, for convenience, we will use the simplified Gaussian kernel,

$$K\left(\frac{x_0 - x_i}{b}\right) = e^{-\left(\frac{x_0 - x_i}{b}\right)^2},$$

in our research.

For the multiple regressor case, two forms of the kernel function have been proposed. At the prediction point $\tilde{\mathbf{x}}_0 = (x_{01}, x_{02}, \ldots, x_{0k})$, the general multiplicative kernel function is defined as:

$$\mathbf{K}(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_i) \propto K\left(\left\|\frac{1}{b}(\tilde{\mathbf{x}}_0 - \tilde{\mathbf{x}}_i)\right\|\right), \tag{2.1}$$

where K is a univariate kernel function, $\tilde{\mathbf{x}}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ and $\|\|$ represents the standard L_2 (Euclidean) norm. While this form has nice theoretical properties, it is not suggested for use in practice. Instead the product, or multiplicative, kernel function has been suggested (Scott, 1992). At the prediction point $\tilde{\mathbf{x}}_0 = (x_{01}, x_{02}, \ldots, x_{0k})$, the product kernel is defined as:

$$\mathbf{K}(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_i) = \frac{1}{b^k} \prod_{j=1}^{k} K\left(\frac{\tilde{x}_{0j} - \tilde{x}_{ij}}{b}\right), \tag{2.2}$$

where again K is a univariate kernel function. In general, the multivariate forms in (2.1) and (2.2) are different; however, it can be shown that the forms are equivalent when the Gaussian kernel function is used. Since the coding of variables in response surface designs typically involves centering and scaling, the units are comparable in all directions, thus it is reasonable to use the same bandwidth, b, in all dimensions as expressed in (2.2). Therefore, the expression for the multivariate kernel fit is given by:

$$\hat{\mathbf{y}}^{(KER)} = \mathbf{H}^{(KER)}\mathbf{y},$$

where $\mathbf{H}^{(KER)}$ is the kernel HAT or smoother matrix defined as:

$$\mathbf{H}^{(KER)} = \begin{bmatrix} \mathbf{h}_1^{(KER)\prime} \\ \mathbf{h}_2^{(KER)\prime} \\ \vdots \\ \mathbf{h}_n^{(KER)\prime} \end{bmatrix}$$

and $\mathbf{h}_i^{(KER)\prime} = \left(h_{i1}^{(KER)} h_{i2}^{(KER)} \dots h_{in}^{(KER)}\right)$ and $h_{ij}^{(KER)} = \frac{\mathbf{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)}{\sum_{j=1}^{n} \mathbf{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)}$. For more details on multivariate kernel regression, see for example Scott (1992) and Simonoff (1996).

The smoothness of the estimated function is controlled by the bandwidth, b. A larger bandwidth value results in a smoother function, whereas a smaller value results in a less smooth function. However, if the bandwidth is chosen too large, the estimated function may be too smooth, resulting in estimates with low variance but high bias. On the other hand, a bandwidth that is too small may result in a rougher fit with low bias but high variance. Thus, a bandwidth should be chosen that offers a balance between bias and variance.

The choice of bandwidth is critical, and the literature is rich with bandwidth selection methods [see for example Härdle (1990) and Härdle et al. (2004)]. The trade-off between bias and variance naturally leads to the minimization of an optimality criteria such as mean squared error. Mays, Birch, and Starnes (2001) introduce a penalized cross-validation technique, PRESS**, for choosing an appropriate bandwidth. The approach chooses the bandwidth as

the value b that minimizes PRESS**, defined as:

$$\text{PRESS}^{**} = \frac{\text{PRESS}}{n - \text{trace}\left(\mathbf{H}^{(KER)}\right) + (n-1)\,\frac{SSE_{max} - SSE_b}{SSE_{max}}},$$

where SSE_{max} is the largest error sum of squares over all possible bandwidth values, SSE_b is the error sum of squares associated with a particular bandwidth value b, and the prediction error sum of squares, PRESS, is given by:

$$\text{PRESS} = \sum_{i=1}^{n} \left(y_i - \hat{y}_{i,-i}\right)^2,$$

where $\hat{y}_{i,-i}$ denotes the estimated response obtained by leaving out the i^{th} observation when estimating at location $\tilde{\mathbf{x}}_i$. Mays, Birch, and Starnes (2001) show that PRESS** performs well by guarding against very small and very large bandwidths.

2.3.2 Local Linear Regression

Kernel regression is an intuitive approach to estimation; however, it inherently has a boundary bias problem when symmetric kernel functions, such as the Gaussian, are used. For example, if the observations follow a concave down trend, the kernel estimates at the first and n^{th} order statistics of the regressor are weighted averages of values larger than the observed responses at $x_{(1)}$ and $x_{(n)}$, respectively. Thus, the estimates at $x_{(1)}$ and $x_{(n)}$ will most likely be biased. Local polynomial regression (LPR) is a smoothing technique that is robust to biased estimates at the boundary of the design space. Originally proposed by Cleveland (1979), LPR is essentially a weighted least squares problem where the weights are given by a kernel function. The polynomial form of the LPR fit can be of order one or greater and we focus on degree $p = 1$, local linear regression (LLR), in this research.

The local linear regression fit at prediction point $\tilde{\mathbf{x}}_0 = (x_{01}, x_{02}, \ldots, x_{0k})$ is given by:

$$\hat{y}_0^{(LLR)} = \tilde{\mathbf{x}}_0' \left(\mathbf{X}'\mathbf{W}_0\mathbf{X}\right)^{-1} \mathbf{X}'\mathbf{W}_0\mathbf{y} = \mathbf{h}_0^{(LLR)\prime}\mathbf{y},$$

where $\mathbf{W}_0$ is a $n \times n$ diagonal matrix containing the kernel weights associated with $\tilde{\mathbf{x}}_0$, $\mathbf{W}_0 = \left\langle h_{0i}^{(KER)} \right\rangle$. In matrix notation, the LLR estimates can be expressed as:

$$\hat{\mathbf{y}}^{(LLR)} = \mathbf{H}^{(LLR)}\mathbf{y},$$

where $\mathbf{H}^{(LLR)}$ is the local linear HAT or smoother matrix defined as:

$$\mathbf{H}^{(LLR)} = \begin{bmatrix} \mathbf{h}_1^{(LLR)\prime} \\ \mathbf{h}_2^{(LLR)\prime} \\ \vdots \\ \mathbf{h}_n^{(LLR)\prime} \end{bmatrix}, \tag{2.3}$$

where $\mathbf{h}_i^{(LLR)\prime} = \tilde{\mathbf{x}}_i' \left(\mathbf{X}'\mathbf{W}_i\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}_i$.

Since the LLR estimates are dependent on the kernel weights, bandwidth selection remains important. We can extend the PRESS** method for choosing an appropriate bandwidth to the LLR setting by using the following expression:

$$\text{PRESS}^{**} = \frac{\text{PRESS}}{n - \text{trace}\left(\mathbf{H}^{(LLR)}\right) + \left(n - (k+1)\right)\frac{SSE_{max} - SSE_b}{SSE_{max}}}.$$

For more details on local polynomial regression, see for example Fan and Gijbels (1996) and Fan and Gijbels (2000).

2.4 Summary

Parametric and nonparametric approaches to modeling each possess positive and negative attributes. The parametric approach has been extensively researched and utilized in RSM, while the nonparametric approach is a relatively new idea in RSM. The parametric approach is superior if the true, underlying function can be adequately expressed parametrically and if the user correctly specifies the parametric form. However, if the model is misspecified, the estimates may be highly biased and optimal control factor settings may be miscalculated.

On the other hand, if the user has no idea about the true form of the underlying function, nonparametric methods offer a nice alternative. Nonparametric methods can provide superior fits by capturing structure in the data that a misspecified parametric model cannot. However, nonparametric methods were originally developed for situations with large sample sizes. In small sample settings, which are customary in RSM, nonparametric techniques may fit irregularities in the data too closely thereby creating estimated functions that are highly variable. Consequently, optimization may be based on non-reproducible idiosyncrasies in the data.

In this research, we will focus on the parametric approach of least squares and the nonparametric approach of local linear regression. Note, however, that the techniques discussed can be generalized to other parametric and nonparametric methods. In Chapter 3, we will study the recently proposed nonparametric approach (LLR) in more depth by investigating the idea of degrees of freedom. In Chapters 4 and 5, we will develop another approach to modeling for RSM, a semiparametric approach that is essentially a hybrid of the parametric and nonparametric methods. Then we will compare the performance of the three approaches in a variety of possible scenarios.

Chapter 3

Degrees of Freedom for Local Linear Regression

3.1 Introduction

"What are degrees of freedom?" Good (1973) asked this question decades after many statisticians thought the definition was made perfectly clear. Unfortunately, the concept of degrees of freedom is still difficult for many to understand. Cramér (1946) defines degrees of freedom as the rank of a quadratic form, yet textbooks rarely mention this explanation. In fact, most regression texts refer to error degrees of freedom as the number of data points minus the trace of the "HAT" or smoother matrix. This interpretation works well for parametric regression; however, the definition may no longer hold when extended to local polynomial regression, specifically local linear regression. We will investigate the definition of degrees of freedom for the parametric and local linear regression settings.

Degrees are freedom are frequently used in the local linear regression (LLR) setting. The number appears in the denominator of PRESS**, a statistic that is commonly used to select

the appropriate bandwidth value for the LLR fit to data. Furthermore, when performing inference, LLR test statistics typically are approximately F or t distributed and require degrees of freedom. In this research, we focus on bandwidth selection. Thus, we will also investigate the impact the definition of degrees of freedom has on bandwidth selection in the LLR setting.

3.2 Parametric Regression

When n responses are observed, the parametric linear model may be expressed in matrix notation as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\mathbf{y}$ is a $n \times 1$ vector of responses, $\mathbf{X}$ is a $n \times (k+1)$ matrix of k explanatory variables augmented by a column of ones, $\boldsymbol{\beta}$ is a $(k+1) \times 1$ vector of unknown parameters, and $\boldsymbol{\varepsilon}$ is the $n \times 1$ vector of random errors. The error terms are assumed to be uncorrelated with a normal distribution, zero mean, and constant variance σ^2.

The method of ordinary least squares (OLS) provides uniform minimum variance unbiased estimates (UMVUE) of the parameters in $\boldsymbol{\beta}$ such that the sum of squared errors, given as:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

where $\hat{y}_i = \mathbf{x}_i'\hat{\boldsymbol{\beta}}$ and $\mathbf{x}_i'$ is the i^{th} row of the $\mathbf{X}$ matrix, is minimized. The estimated responses can be written as:

$$\hat{\mathbf{y}}^{(OLS)} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}^{(OLS)}\mathbf{y},$$

where the $n \times n$ matrix $\mathbf{H}^{(OLS)}$ is commonly referred to as the ordinary least squares "HAT" matrix since it transforms the observed y values into the $\hat{y}$ values. Two important properties of the OLS HAT matrix are:

(i) $\mathbf{H}^{(OLS)}$ is symmetric and idempotent.

(ii) The rank of $\mathbf{H}^{(OLS)}$ is equivalent to the trace of $\mathbf{H}^{(OLS)}$; that is, $\text{rank}\left(\mathbf{H}^{(OLS)}\right) = \text{trace}\left(\mathbf{H}^{(OLS)}\right) = k+1$, where k is the number of regressors in the model.

The HAT matrix can be used to write the OLS residuals, $\mathbf{e}^{(OLS)}$, as:

$$\mathbf{e}^{(OLS)} = \mathbf{y} - \hat{\mathbf{y}}^{(OLS)} = \mathbf{y} - \mathbf{H}^{(OLS)}\mathbf{y} = \left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)\mathbf{y},$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. Thus, the sum of squared errors can be rewritten as:

$$SSE^{(OLS)} = \mathbf{e}^{(OLS)\prime}\mathbf{e}^{(OLS)} = \mathbf{y}'\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)'\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)\mathbf{y}. \tag{3.1}$$

For more details on ordinary least squares and the HAT matrix, see Myers (1990).

If we let $A = \left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)'\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)$ in equation (3.1), it is evident that the sum of squared errors is a quadratic form. Therefore, by Cramér's (1946) definition

$$dfE = \text{rank}(A) = \text{rank}\left[\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)'\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)\right],$$

where dfE represents the error degrees of freedom. Since $\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right)$ is also symmetric and idempotent, we can rewrite the error degrees of freedom as:

$$\begin{aligned} dfE &= \text{rank}\left[\mathbf{I} - \left(2\mathbf{H}^{(OLS)} - \mathbf{H}^{(OLS)\prime}\mathbf{H}^{(OLS)}\right)\right] \\ &= \text{rank}\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right) \\ &= \text{trace}\left(\mathbf{I} - \mathbf{H}^{(OLS)}\right). \end{aligned}$$

Then using properties of the trace, we can again rewrite the error degrees of freedom as:

$$\begin{aligned} dfE &= \text{trace}\left(\mathbf{I}\right) - \text{trace}\left(\mathbf{H}^{(OLS)}\right) \\ &= n - p, \end{aligned}$$

where $p = k+1$ and k equals the number of regressors. Thus, the degrees of freedom for error for parametric regression are commonly defined as n minus the trace of the HAT or smoother matrix, $\mathbf{H}^{(OLS)}$. While this definition is true for OLS, it is not clear that this definition applies to local linear regression.

3.3 Local Linear Regression

Several parametric definitions of error degrees of freedom have been utilized throughout the nonparametric regression literature [see for example Hastie and Tibshirani (1990), Fox (2000), Ruppert, Wand, and Carrol (2003), Zhang (2003), and Takezawa (2006)]. Some use the definition:

$$dfE_1 = \text{trace}\left(\mathbf{I} - \mathbf{H}\right), \tag{3.2}$$

where $\mathbf{H}$ is the appropriate smoother matrix, because it is computationally simple, while others argue that

$$dfE_2 = \text{trace}[(\mathbf{I} - \mathbf{H})'(\mathbf{I} - \mathbf{H})] \tag{3.3}$$

is more suitable theoretically [see for example Zhang (2003) and Takezawa (2006)]. Furthermore, Ruppert, Wand, and Carrol (2003) suggest that dfE_2 be used in the estimation of σ^2 and other methods that are based on the sum of squared errors, such as F-tests.

Nevertheless, both equations are theoretically incorrect according to Cramér's (1946) definition as they assume that the trace and rank operations are interchangeable in the nonparametric regression setting when in fact they are not. The trace and rank of a matrix are only equivalent if the matrix is a $n \times n$ symmetric and idempotent matrix. While this is true for the parametric smoother matrix, it is not true for the nonparametric smoother matrix. In particular, it is not true for local linear regression (LLR).

Using LLR, we can express the nonparametric estimates as:

$$\hat{\mathbf{y}}^{(LLR)} = \mathbf{H}^{(LLR)}\mathbf{y},$$

where $\mathbf{H}^{(LLR)}$ is the local linear HAT or smoother matrix defined as:

$$\mathbf{H}^{(LLR)} = \begin{bmatrix} \mathbf{h}_1^{(LLR)\prime} \\ \mathbf{h}_2^{(LLR)\prime} \\ \vdots \\ \mathbf{h}_n^{(LLR)\prime} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{x}}_1' \left(\mathbf{X}'\mathbf{W}_1\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}_1 \\ \tilde{\mathbf{x}}_2' \left(\mathbf{X}'\mathbf{W}_2\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}_2 \\ \vdots \\ \tilde{\mathbf{x}}_n' \left(\mathbf{X}'\mathbf{W}_n\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}_n \end{bmatrix}, \tag{3.4}$$

$\tilde{\mathbf{x}}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$, and $\mathbf{W}_i$ is a $n \times n$ diagonal matrix containing the kernel weights associated with $\tilde{\mathbf{x}}_i$. For more details on local polynomial regression, see Fan and Gijbels (1996) and Fan and Gijbels (2000).

Unlike the parametric smoother matrix, $\mathbf{H}^{(LLR)}$ is neither symmetric nor idempotent. Thus, the sum of squared errors for LLR is given by:

$$SSE^{(LLR)} = \mathbf{e}^{(LLR)\prime}\mathbf{e}^{(LLR)} = \mathbf{y}' \left(\mathbf{I} - \mathbf{H}^{(LLR)}\right)' \left(\mathbf{I} - \mathbf{H}^{(LLR)}\right) \mathbf{y}.$$

The corresponding theoretical error degrees of freedom for LLR, using Cramér's (1946) definition, is given by:

$$dfE^{(LLR)} = \text{rank}\left[\left(\mathbf{I} - \mathbf{H}^{(LLR)}\right)' \left(\mathbf{I} - \mathbf{H}^{(LLR)}\right)\right], \tag{3.5}$$

an expression that cannot be further simplified.

While the definition in equation (3.5) is theoretically correct according to Cramér (1946), it does not work well in practice as the degree of smoothing is ignored. Smaller bandwidths will result in a rougher fit than larger bandwidths which produce a smoother fit to the data. Therefore, fits with smaller bandwidths should use more degrees of freedom for the model and less for error than do fits based on larger bandwidths. However, it should be noted that the error degrees of freedom based on equation (3.5) remains constant regardless of the bandwidth value.

This fact can be seen by looking more closely at equation (3.4) and noting that each row of the smoother matrix involves the matrix multiplication of the diagonal kernel weight matrix, $\mathbf{W}_i$. Since we are using the simplified Gaussian kernel in our research, we know that $\mathbf{W}_i$ is full rank for all i regardless of the value of the bandwidth. In fact, if we assume that $\mathbf{W}_i = \mathbf{W}$

for all i, we can clearly see that the rank of $\mathbf{H}^{(LLR)}$ does not depend on the bandwidth value:

$$\begin{aligned}
\text{rank}(\mathbf{H}^{(LLR)}) &= \text{rank}\left(\begin{bmatrix} \tilde{\mathbf{x}}_1'\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W} \\ \vdots \\ \tilde{\mathbf{x}}_n'\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W} \end{bmatrix}\right) \\
&= \text{rank}\left(\begin{bmatrix} \tilde{\mathbf{x}}_1' \\ \vdots \\ \tilde{\mathbf{x}}_n' \end{bmatrix}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\right) \\
&= \text{rank}\left(\begin{bmatrix} \tilde{\mathbf{x}}_1' \\ \vdots \\ \tilde{\mathbf{x}}_n' \end{bmatrix}\mathbf{X}'\right).
\end{aligned}$$

Thus, the rank of the nonparametric smoother matrix only depends on the rank of the $\mathbf{X}$ matrix and remains constant over all bandwidth values.

Since the degree of smoothing should be incorporated into the degrees of freedom, we will not consider the theoretical definition in equation (3.5) as a feasible possibility in practice. Therefore, in our study of the impact of degrees of freedom on bandwidth selection, we will only investigate dfE_1 and dfE_2 defined in equations (3.2) and (3.3), respectively.

3.4 Comparison of Degrees of Freedom Definitions

In this section, Monte Carlo simulations are used to compare the two definitions seen in the nonparametric literature, dfE_1 and dfE_2 defined in equations (3.2) and (3.3), respectively. For the linear model, it is well known that $\hat{\sigma}^2$, defined as:

$$\hat{\sigma}^2 = \frac{SSE}{dfE},$$

is an unbiased estimator for σ^2. Therefore, we will compare the definitions of error degrees of freedom by inspecting the quality of the estimator for each. That is, we will compare the

performance of

$$\hat{\sigma}_1^2 = \frac{SSE}{dfE_1}$$

and

$$\hat{\sigma}_2^2 = \frac{SSE}{dfE_2}.$$

In the simulation study, both estimators, $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ will have the same SSE value in their numerators. Thus, the only difference between the two is their denominators, dfE_1 and dfE_2.

Each Monte Carlo simulation will use 500 simulated data sets, each of which is based on the following underlying model:

$$\begin{aligned} y &= 20 - 10x_1 - 25x_2 - 15x_1x_2 + 20x_1^2 + 50x_2^2 \\ &\quad +\gamma\left[2\sin\left(4\pi x_1\right) + 2\cos\left(4\pi x_2\right) + 2\sin\left(4\pi x_1 x_2\right)\right] + \varepsilon, \end{aligned}$$

where $\varepsilon \sim \mathrm{N}(0,1)$ and γ represents the model misspecification parameter. Thus, the true value of σ^2 is 1. The data will be generated from a central composite design with five center runs for a total of 13 experimental runs. When coded to be between 0 and 1, factors x_1 and x_2 have five levels with values taken to be at 0.000, 0.146, 0.500, 0.854, and 1.000 for each factor. As the value of γ increases, the amount of misspecification, when compared to a quadratic model in two regressors, increases in the model. Five values of γ will be studied varying from $\gamma = 0.00$ (correct model specification) to $\gamma = 0.25$, 0.50, 0.75, and 1.00, representing increasing degrees of model misspecification. The R language is used for computations and simulations.

To ensure that both estimators have the same SSE value in their numerators, the bandwidth for the LLR fit will be fixed at the optimal value, b_0, which is found by minimizing the *average* $MSE(\hat{y})$, $AVEMSE$, over all possible bandwidth values, where

$$AVEMSE = \frac{\sum MSE(\hat{y})}{n}.$$

Using theoretical expressions for the bias and variance of fits, $MSE(\hat{y})$ is calculated as:

$$MSE(\hat{y}) = [Bias(\hat{y})]^2 + Var(\hat{y})$$

The theoretical expressions for bias and variance of parametric, nonparametric, and semi-parametric fits can be found in Appendices A and B. $AVEMSE$ is used as the "optimal" bandwidth selection procedure because it gives a fair trade-off between the bias and variance of the fitted values.

To compare the quality of the estimators, we will look at their Monte Carlo average values, $MCA(\hat{\sigma}_i^2)$ for $i = 1$ and 2, defined as:

$$MCA(\hat{\sigma}_i^2) = \frac{\sum_{j=1}^{500} \hat{\sigma}_{ij}^2}{500}.$$

However, we also need to investigate both the accuracy and precision of the estimators. Thus, we want to look at the mean squared error of the estimators, $MSE(\hat{\sigma}_i^2)$ for $i = 1$ and 2, as it is a measure that incorporates both the bias and the variance of each estimator. In fact, for comparison purposes in our simulation study, we will use the Monte Carlo mean squared error of the estimators, $MCMSE(\hat{\sigma}_i^2)$ for $i = 1$ and 2, given by:

$$\begin{aligned} MCMSE(\hat{\sigma}_i^2) &= MC[Bias(\hat{\sigma}_i^2)]^2 + MCVar(\hat{\sigma}_i^2) \\ &= \left(\frac{\sum_{j=1}^{500} (\hat{\sigma}_{ij}^2 - \sigma^2)^2}{500} \right) + MCVar(\hat{\sigma}_i^2), \end{aligned}$$

where $\sigma^2 = 1$.

Table 3.1 provides a comparison of the two estimators based on the MCA and $MCMSE$ values for the varying degrees of model misspecification. Note that $\hat{\sigma}_1^2$ underestimates the true $\sigma^2 = 1$, on the average, while $\hat{\sigma}_1^2$ overestimates the value. Also notice that as the amount of model misspecification increases, the quality of both estimators slightly worsen as their $MCMSE$ values increase. However, $\hat{\sigma}_1^2$ consistently outperforms $\hat{\sigma}_2^2$ as it results in a smaller $MCMSE$ value for all γ values. It is interesting to note that these results do not support Ruppert, Wand, and Carrol (2003) who suggest the use of dfE_2 in the estimation of σ^2.

Table 3.1: Monte Carlo average (MCA) values and Monte Carlo mean squared error ($MCMSE$) values for 500 Monte Carlo runs. Best values in bold.

		MCA		$MCMSE$	
γ	b_0	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$
0.00	0.221	**0.9742**	1.0432	**0.8696**	0.9984
0.25	0.220	**0.9723**	1.0394	**0.8737**	0.9990
0.50	0.220	**0.9748**	1.0420	**0.8738**	0.9995
0.75	0.218	**0.9745**	1.0378	**0.8816**	1.0007
1.00	0.216	**0.9754**	1.0351	**0.8891**	1.0018

3.5 Impact of Degrees of Freedom Definition on Choice of Bandwidth in Local Linear Regression

We now investigate the impact the definition for degrees of freedom has on bandwidth selection, our primary reason for studying the topic. When using PRESS** to select a bandwidth for LLR, the original formula involves dfE_1 defined in equation (3.2); that is, the formula involves the trace of the smoother matrix:

$$\text{PRESS}_1^{**} = \frac{\text{PRESS}}{n - \text{trace}\left(\mathbf{H}^{(LLR)}\right) + (n - (k+1))\,\frac{SSE_{max} - SSE_b}{SSE_{max}}},$$

where SSE_{max} is the largest error sum of squares over all possible bandwidth values, SSE_b is the error sum of squares associated with a particular bandwidth value b, and the prediction error sum of squares, PRESS, is given by:

$$\text{PRESS} = \sum_{i=1}^{n} \left(y_i - \hat{y}_{i,-i}\right)^2,$$

where $\hat{y}_{i,-i}$ denotes the estimated response obtained by leaving out the i^{th} observation when estimating at location $\tilde{\mathbf{x}}_i$. The term $[n - \text{trace}(\mathbf{H}^{(LLR)})]$ penalizes for small bandwidths

thereby providing protection against overfitting. As the bandwidth gets smaller, the trace of $\mathbf{H}^{(LLR)}$ gets larger. Thus $[n - \text{trace}(\mathbf{H}^{(LLR)})]$ gets smaller and increases PRESS_1^{**}. On the other hand, the term $\left[(n-(k+1))\,\frac{SSE_{max}-SSE_b}{SSE_{max}}\right]$ protects against underfitting by penalizing large bandwidths. If dfE_2 defined in equation (3.3) is used instead, the formula changes as follows:

$$\text{PRESS}_2^{**} = \frac{\text{PRESS}}{\text{trace}\left[(\mathbf{I}-\mathbf{H}^{(LLR)})'(\mathbf{I}-\mathbf{H}^{(LLR)})\right] + (n-(k+1))\,\frac{SSE_{max}-SSE_b}{SSE_{max}}},$$

and the term $\left[\text{trace}[(\mathbf{I}-\mathbf{H}^{(LLR)})'(\mathbf{I}-\mathbf{H}^{(LLR)})]\right]$ is the penalty term for small bandwidths. Using the theoretically optimal bandwidth value, b_0, as a reference, we will compare the bandwidth values given by the different formulas for PRESS^{**}.

The same simulation set up that was used in the previous section will be used here. Comparisons will be based on the distance the selected bandwidth value is from the optimal value as well as the corresponding Monte Carlo simulated integrated mean squared error ($SIMSE$) values given by:

$$SIMSE = \frac{\sum ASE}{500},$$

where

$$ASE = \frac{\sum\left(E[y_i] - \hat{y}_i^{(LLR)}\right)^2}{1600},$$

where ASE denotes the average squared error for the estimates for each of the 500 simulated data sets and $E[y_i]$ is the true underlying model. Note that the average squared error values, ASE, are calculated across 1600 $\mathbf{x}_0$ locations (based on a 40×40 uniform grid of points in the regressor space) for each of the 500 simulated data sets. This will provide an indication of the performance of the methods over the entire response surface.

Table 3.2 provides a comparison of the bandwidth values chosen by each method for the varying degrees of model misspecification. Note that both methods yield estimated bandwidths that are too large on the average. This result is to be expected as Mays (1995) found that

even though PRESS** performed better than other bandwidth selectors, including PRESS, PRESS*, and generalized cross-validation (GCV), PRESS** still chose bandwidths that were consistently a little large. Also notice that although both definitions result in very close bandwidth values, the original PRESS$_1^{**}$ consistently outperforms PRESS$_2^{**}$ as it yields a bandwidth closer to the theoretically optimal value, b_0, for all γ values.

Table 3.2: Average bandwidth values for 500 Monte Carlo runs. Best values in bold.

γ	b_0	PRESS$_1^{**}$	PRESS$_2^{**}$
0.00	0.221	**0.4929**	0.4949
0.25	0.220	**0.4922**	0.4942
0.50	0.220	**0.4924**	0.4946
0.75	0.218	**0.4935**	0.4959
1.00	0.216	**0.4957**	0.4983

Table 3.3 provides a comparison of the methods based on the $SIMSE$ values for the varying degrees of model misspecification. Table 3.3 also includes the theoretically optimal $SIMSE$ values based on the theoretically optimal bandwidth values, b_0. Note that even though the bandwidths selected by PRESS$_1^{**}$ and PRESS$_2^{**}$ are larger than the optimal bandwidths, their resulting fits are not too far from optimal. Thus, PRESS** performs well as a bandwidth selection criteria as it yields bandwidths that provide adequate fits. Looking closer at the $SIMSE$ values for PRESS$_1^{**}$ and PRESS$_2^{**}$, we see that the two result in very close $SIMSE$ values, yet the original PRESS$_1^{**}$ consistently outperforms PRESS$_2^{**}$ as it yields a smaller $SIMSE$ value for all γ values.

Table 3.3: Simulated integrated mean squared error ($SIMSE$) values for 500 Monte Carlo runs. Best values in bold.

γ	b_0	PRESS_1^{**}	PRESS_2^{**}
0.00	5.5374	**5.9604**	5.9658
0.25	7.3823	**7.5380**	7.5414
0.50	9.6327	**9.8439**	9.8451
0.75	12.1368	**12.8733**	12.8749
1.00	14.8501	**16.6219**	16.6271

3.6 Conclusions

Even though they are not theoretically correct by Cramér's (1946) definition, nonparametric methods have continued to use the parametric definitions for error degrees of freedom dfE_1 and dfE_2 defined in equations (3.2) and (3.3), respectively. The degree of smoothing is critical in nonparametric methods, and these two definitions incorporate this factor whereas the theoretically correct definition in equation (3.5) does not. These two definitions were studied to determine which is more appropriate in the LLR setting, dfE_1 which is simple computationally or dfE_2 which is closer to the theoretical definition.

Two simulation studies were conducted to compare the two definitions. In both studies, the two definitions have relatively close results; however, dfE_1 yields superior results in each. In the first simulation study, dfE_1 gives a better estimate of σ^2 as it yields a smaller $MCMSE$ value over all degrees of potential model misspecification. In the second simulation study, dfE_1 produces better bandwidth and $SIMSE$ values. PRESS_1^{**}, which is based on dfE_1, yields bandwidth values closer to the theoretically optimal bandwidth value for all degrees of potential model misspecification. Thereby suggesting that dfE_2 penalizes PRESS too much for small bandwidths. Furthermore, the bandwidth chosen by the original PRESS_1^{**} results

in a better fit to the entire response surface as it has smaller $SIMSE$ values for all degrees of potential model misspecification. Therefore, the computationally simple definition dfE_1 defined in equation (3.2) appears to be the best choice when performing bandwidth selection in the local linear regression setting.

Chapter 4

Nonparametric and Semiparametric Approaches to Response Surface Methodology

4.1 Introduction

Response Surface Methodology (RSM) is a technique commonly employed by industrial statisticians, engineers, and other researchers to study and optimize products or processes. A sequential experimental strategy originally proposed by Box and Wilson (1951), RSM combines ideas from experimental design and regression model building to gain knowledge about the relationship between several explanatory variables and one or more responses. Furthermore, RSM utilizes optimization methods to find the operating conditions for the explanatory variables that will optimize the response(s).

The most elementary RSM problem is one in which the researcher searches for the oper-

ating conditions that will result in an optimal value for a single response of interest. The relationship between the k explanatory variables, $x_1, x_2, \ldots, x_k$, and the response, y, is:

$$y = f(x_1, x_2, \ldots, x_k) + \varepsilon,$$

where f is the true response function and ε is the error term that represents sources of variability not accounted for in f. The form of the true response function is unknown and may be very complicated. The error term is assumed to have a normal distribution with mean zero and constant variance σ^2. Thus, the response function of y is also the mean function of y.

Since the true relationship between the explanatory variables and the response is unknown, we must approximate the form of f. The function must be well estimated as misspecification of the model form can have serious implications in process optimization. Classic RSM utilizes parametric regression models for the function approximation; however, in many situations, parametric models may not adequately approximate the true relationship. Therefore, we propose the use of nonparametric and semiparametric modeling techniques. We will compare the three methods, parametric, nonparametric, and semiparametric, with applications and simulations. We also suggest the use of a more flexible optimization routine, the genetic algorithm, for determining the settings of the explanatory variables which optimize the response.

4.2 Parametric Approach

The traditional approach to modeling the relationship between the explanatory variables and the response is to assume that the underlying functional form can be expressed parametrically. Researchers tend to assume the relationship is not extremely complex and that a first- or second-order polynomial provides an adequate approximation of the true underlying function. Therefore, the data are usually fit with at most a second-order model. When n ob-

servations are observed, the second-order linear model may be expressed in matrix notation as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\mathbf{y}$ is a $n \times 1$ vector of responses, $\mathbf{X}$ is a $n \times \left(1 + 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}\right)$ model matrix, $\boldsymbol{\beta}$ is a $\left(1 + 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}\right) \times 1$ vector of unknown parameters to be estimated, and $\boldsymbol{\varepsilon}$ is the $n \times 1$ vector of random errors. Utilizing the method of ordinary least squares (OLS), the estimated responses can be written as:

$$\hat{\mathbf{y}}^{(OLS)} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}^{(OLS)}\mathbf{y},$$

where the $n \times n$ matrix $\mathbf{H}^{(OLS)}$ is commonly referred to as the ordinary least squares "HAT" matrix since it transforms the observed y values into the $\hat{y}$ values. For more details on ordinary least squares and the HAT matrix, see Myers (1990).

Once the response has been estimated, the goal becomes finding the operating conditions for the explanatory variables such that the response is optimized. The researcher may have one of three possible goals: minimize the response, maximize the response, or achieve a particular target value for the response.

Contour plots, which graphically show the relationship between the response and two explanatory variables, may be used to achieve any of these goals. The researcher can visually identify general locations in these two-dimensional graphs where their goal is satisfied. In fact, some software packages offer interactive contour plots which allow the researcher to obtain accurate settings for the explanatory variables.

A more formal approach to optimization is calculus based and involves the calculation of derivatives. When minimizing or maximizing a response, the researcher first finds a stationary point by taking the first derivative of the estimated response function, setting it equal to zero, and solving for the corresponding regressor values. Next, the regressor values

are substituted into the second derivative of the response function. If the result is positive, then the stationary point is a minimum. On the other hand, if the result is negative, the stationary point is a maximum.

When searching for a minimum or maximum response, the objective function of interest that is minimized or maximized is the estimated response function. However, when the goal of optimization is to achieve a target value for the response, the objective function takes on a different form such as the estimated squared distance from target ($\widehat{SDT}$), defined as:

$$\widehat{SDT} = (\hat{y} - T)^2, \tag{4.1}$$

where T denotes the target value for the response. The goal now becomes minimizing the $\widehat{SDT}$ function and can be completed using calculus as before.

Note that in all three scenarios the determined set of optimal conditions is highly dependent on quality estimation of the response. Misspecification of the model form can have serious implications in process optimization. For more details on process optimization, see Myers and Montgomery (2002).

4.3 Nonparametric Approach

Situations may arise in which the relationship between the explanatory variables and the response is not adequately modeled parametrically via second-order polynomials. In these situations, any degree of model misspecification may result in serious bias of the estimated response. Furthermore, the optimal regressor settings may be miscalculated. For such situations, nonparametric methods may be suggested as they can capture structure in the data that a misspecified parametric model cannot.

Nonparametric smoothing techniques use curves to describe the relationship between the explanatory variables and the response without any parameters. Thus, the response is

modeled as:

$$y = h(x_1, x_2, \ldots, x_k) + \varepsilon,$$

where h is assumed to have an unknown but reasonably smooth form. Similar to parametric regression, the estimator is a linear combination of the response values; however, the weighting schemes in some nonparametric regression methods assign more weight to observations closest to the point of prediction. The nonparametric fit is more flexible than the parametric fit as it is not confined to the user's specified form.

Although Myers (1999) emphasizes the need for research in the area of nonparametric RSM, little effort has been devoted to the topic. Vining and Bohn (1998) and Anderson-Cook and Prewitt (2005) have suggested the use of nonparametric regression for the dual response problem, yet little work has been done for the basic RSM problem of a single response with the homogeneous variance assumption.

Several fitting techniques have been proposed in the nonparametric regression literature such as kernel regression [see for example Nadaraya (1964), Watson (1964), Priestley and Chao (1972), and Gasser and Müller (1984)], local polynomial models [see for example Fan and Gijbels (1996) and Fan and Gijbels (2000)], spline-based smoothers, and series-based smoothers [see for example Ruppert, Wand, and Carrol (2003)]. Local polynomial regression (LPR) is a popular class of nonparametric smoothing methods and is particularly appealing in response surface applications due to its robustness to biased estimates at the boundary of the design space. LPR is essentially a weighted least squares problem where the weights are given by a kernel function. The polynomial form of the LPR fit can be of order one or greater, and we focus on degree $p = 1$, local linear regression (LLR), in this research.

For the multiple regressor case, at point $\tilde{\mathbf{x}}_0 = (x_{01}, x_{02}, \ldots, x_{0k})$ where prediction is desired, we define the kernel function as:

$$\mathbf{K}(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_i) = \frac{1}{b^k} \prod_{j=1}^{k} K\left(\frac{\tilde{x}_{0j} - \tilde{x}_{ij}}{b}\right), \tag{4.2}$$

where $\tilde{\mathbf{x}}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$, $K\left(\frac{\tilde{x}_{0j} - \tilde{x}_{ij}}{b}\right)$ is a univariate kernel function, and b is the bandwidth. The choice of kernel function is not crucial to the performance of the estimator (Simonoff, 1996). Thus, for convenience, we will use the simplified Gaussian kernel, $K(u) = e^{-u^2}$.

The smoothness of the estimated function is controlled by the bandwidth, b. Since the coding of variables in response surface designs typically involves centering and scaling, the units are comparable in all directions. Thus, it is reasonable to use the same bandwidth, b, in all dimensions as expressed in (4.2). The choice of bandwidth is critical, and the literature is rich with bandwidth selection methods [see for example Härdle (1990), Härdle et al. (2004)]. Typically the bandwidth is chosen to minimize some optimality criteria such as MSE. Mays, Birch, and Starnes (2001) introduce a penalized cross-validation technique, PRESS**, for choosing an appropriate bandwidth. The approach chooses the bandwidth as the value b that minimizes PRESS**, defined as:

$$\text{PRESS}^{**} = \frac{\text{PRESS}}{n - \text{trace}\left(\mathbf{H}^{(LLR)}\right) + (n - (k+1)) \frac{SSE_{max} - SSE_b}{SSE_{max}}},$$

where SSE_{max} is the largest error sum of squares over all possible bandwidth values, SSE_b is the error sum of squares associated with a particular bandwidth value b, k is the number of regressors, and the prediction error sum of squares, PRESS, is given by:

$$\text{PRESS} = \sum_{i=1}^{n} \left(y_i - \hat{y}_{i,-i}\right)^2,$$

where $\hat{y}_{i,-i}$ denotes the estimated response obtained by leaving out the i^{th} observation when estimating at location $\tilde{\mathbf{x}}_i$. The LLR smoother matrix, $\mathbf{H}^{(LLR)}$, is defined below. Mays, Birch, and Starnes (2001) show that PRESS** performs well by guarding against very small and very large bandwidths.

The local linear regression fit at prediction point $\tilde{\mathbf{x}}_0 = (x_{01}, x_{02}, \ldots, x_{0k})$ is given by:

$$\hat{y}_0^{(LLR)} = \tilde{\mathbf{x}}_0' \left(\mathbf{X}'\mathbf{W}_0\mathbf{X}\right)^{-1} \mathbf{X}'\mathbf{W}_0\mathbf{y} = \mathbf{h}_0^{(LLR)\prime}\mathbf{y},$$

where $\mathbf{W}_0$ is a $n \times n$ diagonal matrix containing the kernel weights associated with $\tilde{\mathbf{x}}_0$, $\mathbf{W}_0 = \left\langle h_{0i}^{(KER)} \right\rangle$ where $h_{0i}^{(KER)} = \frac{\mathbf{K}(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_i)}{\sum_{i=1}^{n} \mathbf{K}(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_i)}$. In matrix notation, the LLR estimates can be expressed as:

$$\hat{\mathbf{y}}^{(LLR)} = \mathbf{H}^{(LLR)}\mathbf{y},$$

where $\mathbf{H}^{(LLR)}$ is the local linear HAT or smoother matrix defined as:

$$\mathbf{H}^{(LLR)} = \begin{bmatrix} \mathbf{h}_1^{(LLR)\prime} \\ \mathbf{h}_2^{(LLR)\prime} \\ \vdots \\ \mathbf{h}_n^{(LLR)\prime} \end{bmatrix}, \tag{4.3}$$

where $\mathbf{h}_i^{(LLR)\prime} = \tilde{\mathbf{x}}_i' \left(\mathbf{X}'\mathbf{W}_i\mathbf{X}\right)^{-1} \mathbf{X}'\mathbf{W}_i$. For more details on local polynomial regression, see for example Fan and Gijbels (1996) and Fan and Gijbels (2000).

Similar to the parametric approach, once an estimate of the response function has been calculated, process optimization will be carried out. Unfortunately, the optimization method suggested for the parametric approach is based on techniques which require continuous functions with derivatives for the estimated function. Since the estimates from local polynomial regression do not result in closed form expressions, this optimization routine and other common techniques are no longer applicable. Therefore, we advocate the use of the genetic algorithm for optimization.

The genetic algorithm (GA), originally developed by Holland (1975), has become a popular optimization technique, especially for functions that do not have known parametric forms. Instead of using derivatives to find optimal solutions, the GA uses evolutionary concepts such as selection, crossover, and mutation. Furthermore, the GA uses an intelligent, sequential search strategy which enables the user to find global solutions more efficiently (Goldberg, 1989). Thus, we will use the GA for process optimization. For more detail of our GA approach, please see Chapter 6.

4.4 Semiparametric Approach

When used individually, both parametric and nonparametric regression methods have shortcomings. If the user misspecifies the parametric model, the estimates may be highly biased. Whereas the nonparametric fits may be highly variable, especially in small sample settings which are common in RSM. Semiparametric methods have been proposed in the nonparametric regression literature to overcome these drawbacks. Semiparametric techniques are especially useful in situations where the user has partial knowledge of the underlying model or the data contains important "bumps" that parametric models cannot capture. Early papers on semiparametric regression include Speckman (1988) and Einsporn and Birch (1993). More recently, Mays, Birch, and Starnes (2001) introduced model robust regression 2 (MRR2) as an improvement to previous approaches.

MRR2 combines a parametric fit to the raw data with a nonparametric fit to the residuals from the parametric fit via a mixing parameter, λ. In this research, the parametric fit is by least squares and the nonparametric fit is by local linear regression; however, other parametric and nonparametric methods could be utilized. The vector of residuals, $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}^{(OLS)}$, represents the structure in the data which is not captured by the user's specified parametric model. The vector of residuals is fit nonparametrically via LLR resulting in the vector of smoothed residuals,

$$\hat{\mathbf{r}} = \mathbf{H}_r^{(LLR)}\mathbf{r},$$

where $\mathbf{H}_r^{(LLR)}$ is computed similarly to the LLR smoother matrix in (4.3) but with the "response" variable being the residuals from the OLS fit to the raw data. The MRR2 estimates are then obtained by adding a portion of the LLR smoothed residuals back to the original parametric fit, yielding:

$$\hat{\mathbf{y}}^{(MRR2)} = \hat{\mathbf{y}}^{(OLS)} + \lambda\hat{\mathbf{r}},$$

where $\lambda \in [0, 1]$.

Similar to the choice of bandwidth in LLR, the choice of the mixing parameter, λ, involves a bias-variance trade-off. Mays, Birch, and Starnes (2001) derive the following data driven expression for the asymptotically optimal value of the mixing parameter for MRR2:

$$\hat{\lambda}_{opt} = \frac{\left\langle \hat{\mathbf{r}}, \mathbf{y} - \hat{\mathbf{y}}^{(OLS)} \right\rangle}{\left\| \hat{\mathbf{r}} \right\|^2}. \tag{4.4}$$

The notation $\langle\rangle$ represents the inner product and $\|\|$ represents the standard L_2 (Euclidean) norm.

As shown in Mays, Birch, and Starnes (2001), MRR2 performs as well as OLS when there is no model misspecification and MRR2 performs as well or better than LLR when there is extreme misspecification. For low to moderate levels of misspecification, MRR2 performs better than both OLS and LLR. These results are proven in an asymptotic sense as well as demonstrated in small sample settings where the response information was obtained at values of the regressors uniformly placed over the design space. A main underpinning of RSM is the use of cost-efficient experimental designs with strategically placed design points. Therefore, in this research, we will investigate the use of MRR2 in such situations.

Similar to the parametric and nonparametric approaches, once the response function has been estimated, process optimization will be carried out. Furthermore, as in the nonparametric approach, the genetic algorithm will be used for optimization since the estimated function does not take on a closed form expression.

4.5 Examples

To illustrate the semiparametric method (MRR2) and how it can provide an improved fit over traditional parametric RSM methods, two examples are presented in this section. The first involves a process in which the goal of the study is to maximize the response of interest. The second involves a process in which the goal is to achieve a target value for the response

of interest.

4.5.1 The Chemical Process Example

The purpose of the chemical process study from Myers and Montgomery (2002) was to relate yield (y) to temperature (x_1) and time (x_2). The experiment used a rotatable central composite design (CCD) with three center runs for a total of 11 experimental runs. The goal of the study was to maximize the yield. Table 4.1 provides the results of the experiment.

Table 4.1: The chemical process data.

i	x_1	x_2	y
1	-1	-1	88.55
2	1	-1	85.80
3	-1	1	86.29
4	1	1	80.44
5	-1.414	0	85.50
6	1.414	0	85.39
7	0	-1.414	86.22
8	0	1.414	85.70
9	0	0	90.21
10	0	0	90.85
11	0	0	91.31

From the discussion in Myers and Montgomery (2002), we can assume that the user has specified a second-order model. This model appears satisfactory with a R^2 value of 83.88%; however, it yields a somewhat low adjusted R^2 value of only 67.77%. Therefore, nonparametric and semiparametric methods offer alternative approaches which may provide a better fit to the data.

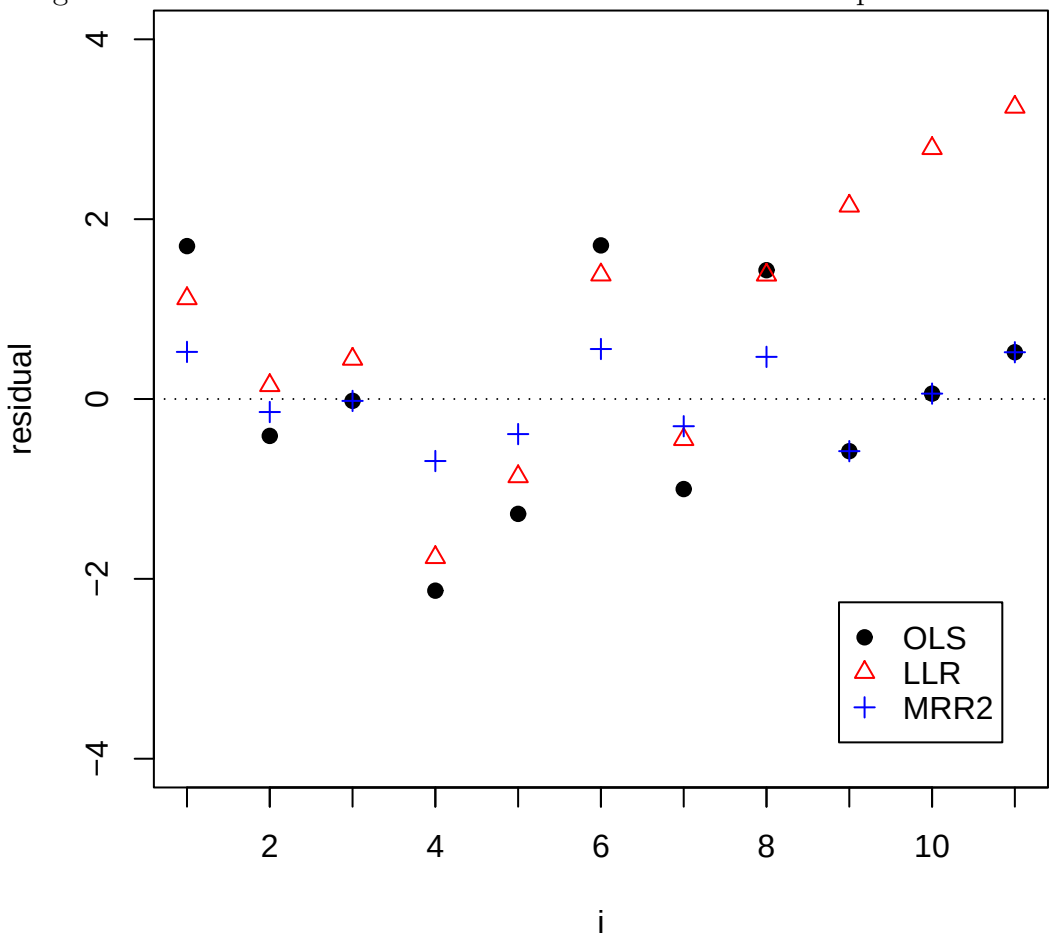
The nonparametric, using LLR, and semiparametric, using LLR to smooth the parametric

residuals, approaches involve the choice of an appropriate global bandwidth, b, for the kernel function. Using PRESS**, we obtain a bandwidth of 0.52 for the response model in the nonparametric approach. A bandwidth of 0.31 was chosen by PRESS** for the nonparametric smooth of the OLS residuals in the semiparametric fit. Furthermore, the semiparametric approach involves the choice of an appropriate mixing parameter, λ. Using equation (4.4), the estimated asymptotically optimal data driven mixing parameter is found to be 1.0. Therefore, the addition of the entire nonparametric residual fit provides necessary correction to the parametric fit.

Figure 4.1 depicts the residuals from the OLS, LLR, and MRR2 models. For the most part, the OLS residuals appear to be randomly scattered around zero. However, a slight concave upward trend can be seen in the first six residuals. Although this pattern does not continue throughout the entire set of residuals, it may indicate some degree of model inadequacy. This trend is still evident in the LLR residuals. Furthermore, the last three LLR residuals, which are the replications at the center location, are much larger than both the OLS and MRR2 residuals at those locations. The poor LLR fit at the three center runs is most likely due to the fact that the center runs receive more than twice the weight the remainder of the design locations do (0.1619 compared to 0.0642). Notice that the concave upward trend is barely discernible in the MRR2 residuals. In fact, the MRR2 residuals are close to "ideal" as their deviations from zero are less pronounced. Thereby suggesting the semiparametric approach outperforms the parametric and nonparametric approaches.

Table 4.2 provides the MSE, R^2, and adjusted R^2 values for the OLS, LLR, and MRR2 models. As previously mentioned, the parametric model performs satisfactorily with a MSE value of 3.16, a R^2 value of 83.88%, and an adjusted R^2 value of 67.77%. However, the addition of the smoothed residual fit provides significant improvement as the semiparametric model has a MSE value of 0.71, a R^2 value of 97.81%, and an adjusted R^2 value of 92.73%. Notice that the nonparametric model performs worse than the parametric and semiparametric models. As previously mentioned, LLR does not perform well in response surface

Figure 4.1: Plot of model residuals in order for the chemical process data.



problems with sparse data and yet, in spite of this, MRR2, which consists partly of LLR fits to the residuals, offers much improvement over OLS and LLR.

Table 4.2: Comparison of model performance statistics for the chemical process data. Best values in bold.

Approach	MSE	R^2	R^2_{adj}
Parametric	3.16	83.88%	67.77%
Nonparametric	5.70	67.17%	41.90%
Semiparametric	**0.71**	**97.81%**	**92.73%**

Using the genetic algorithm to maximize the estimated response, we obtain the optimal re-

gressor settings displayed in Table 4.3. Even though the regressor settings for the parametric, nonparametric, and semiparametric approaches are different, all three sets yield estimates extremely close to one another. However, one must remember that estimation at the optimal regressor settings is based on the correctness of the model. Given that the semiparametric approach has superior model performance statistics, we believe that the semiparametric approach yields optimization results that are a more accurate reflection of the location of the true optimum and the value of the optimum than its parametric and nonparametric counterparts. Ultimately, the only way to determine which approach gives the best optimization results is to perform a confirmatory experiment. Unfortunately, we cannot do so for this example, but we will use simulations to compare the three approaches in general.

Table 4.3: Comparison of the recommended regressor settings for the chemical process data. Best value in bold.

Approach	x_1	x_2	$\hat{y}$
Parametric	-0.17	-0.18	**90.978**%
Nonparametric	-0.37	-0.48	88.296%
Semiparametric	-0.29	-0.35	89.319%

4.5.2 The Motor Oil Example

The motor oil study from Myers and Montgomery (2002) investigates the process used to make a polymer additive for motor oil. The purpose of the study was to examine the effect of reaction time (x_1) and catalyst addition rate (x_2) on the average molecular weight (y). The experiment used a rotatable CCD with five center runs for a total of 13 experimental runs. The goal of the study was to develop a process capable of producing a product with an average molecular weight of 2900. Table 4.4 provides the results of the experiment.

From the discussion in Myers and Montgomery (2002), we can assume that the user has

Table 4.4: The motor oil data.

i	x_1	x_2	y
1	-1	-1	2320
2	1	-1	2925
3	-1	1	2340
4	1	1	2000
5	-1.414	0	3180
6	1.414	0	2925
7	0	-1.414	1930
8	0	1.414	1860
9	0	0	2980
10	0	0	3075
11	0	0	2790
12	0	0	2850
13	0	0	2910

specified a second-order model. Performance statistics show the model is satisfactory with a R^2 value of 92.65% and an adjusted R^2 value of 87.41%. However, stepwise variable selection suggests that the addition of cubic terms could offer some improvement. Unfortunately, the CCD does not contain enough data points to fit a full cubic model. Again, nonparametric and semiparametric methods offer alternative approaches which may provide a better fit to the data.

The nonparametric, using LLR, and semiparametric, using LLR to smooth the parametric residuals, approaches involve the choice of an appropriate global bandwidth, b, for the kernel function. Using PRESS**, we obtain a bandwidth of 0.45 for the response model in the nonparametric approach. A bandwidth of 0.31 was chosen by PRESS** for the nonparametric smooth of the OLS residuals in the semiparametric fit. Furthermore, the semiparametric approach involves the choice of an appropriate mixing parameter, λ. Using equation (4.4), the estimated asymptotically optimal data driven mixing parameter is found to be 1.0.

Therefore, the addition of the entire nonparametric residual fit provides necessary correction to the parametric fit.

Figure 4.2 depicts the residuals from the OLS, LLR, and MRR2 models. For the most part, the OLS residuals appear to be randomly scattered around zero. However, some of the residual values are extremely large. Also notice that many of the LLR residual values are even larger than the OLS values. On the other hand, the MRR2 residuals are much closer to "ideal" as their deviations from zero are far less pronounced. Thereby suggesting the semiparametric approach outperforms the parametric and nonparametric approaches.

Figure 4.2: Plot of model residuals in order for the motor oil data.

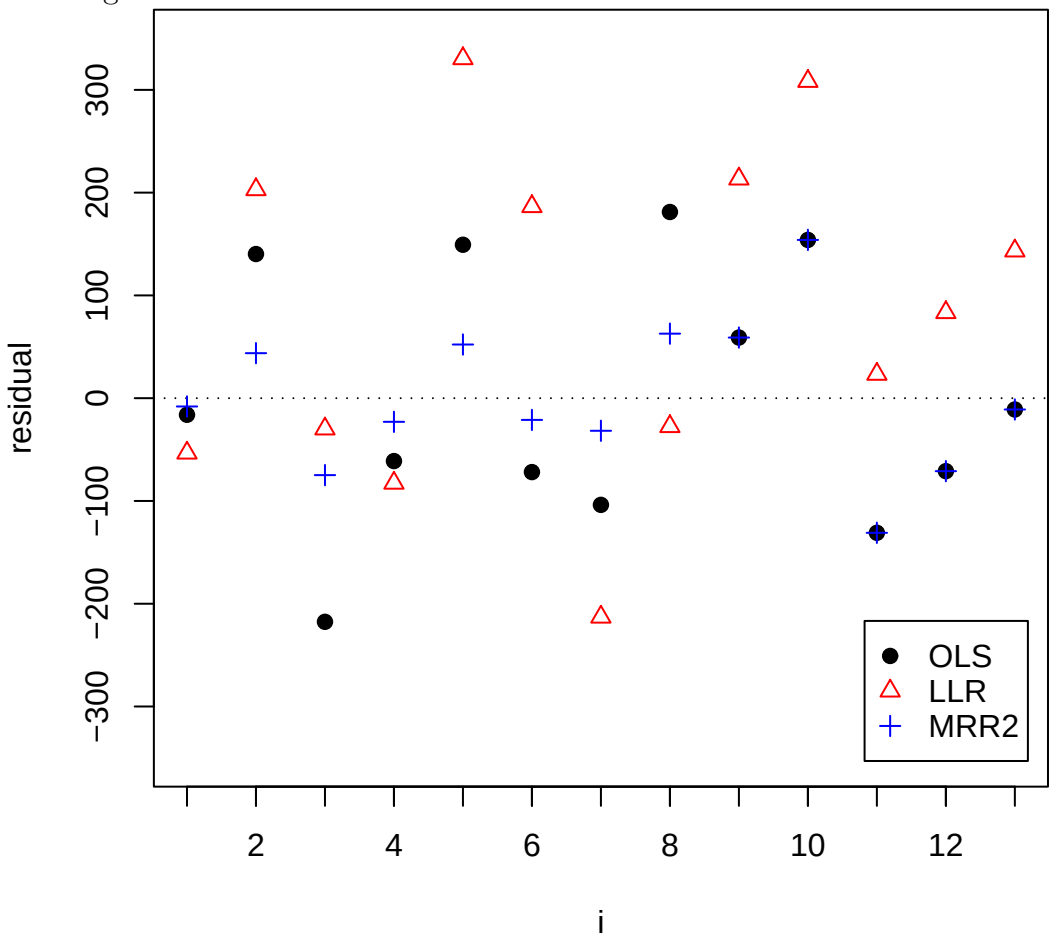


Table 4.5 provides the MSE, R^2, and adjusted R^2 values for the OLS, LLR, and MRR2 models. As previously mentioned, the parametric model performs satisfactorily with a MSE

value of 27,372.02, a R^2 value of 92.65%, and an adjusted R^2 value of 87.41%. However, the addition of the smoothed residual fit provides significant improvement as the semiparametric model has a MSE value of 12,982.53, a R^2 value of 97.48%, and an adjusted R^2 value of 94.03%. Notice that the nonparametric model performs worse than the parametric and semiparametric models. It is interesting, however, that while LLR performs poorly on the raw data, the LLR fit to the OLS residuals provides a significant improvement in the MRR2 fit.

Table 4.5: Comparison of model performance statistics for the motor oil data. Best values in bold.

Approach	MSE	R^2	R^2_{adj}
Parametric	27,372.02	92.65%	87.41%
Nonparametric	58,938.01	84.26%	72.89%
Semiparametric	**12,982.53**	**97.48%**	**94.03%**

Using the genetic algorithm with the estimated squared distance from target ($\widehat{SDT}$) objective function given in (4.1), we obtain the optimal regressor settings displayed in Table 4.6. Even though the regressor settings for the parametric and semiparametric approaches are very different, both sets yield estimates extremely close to the target response of 2900. However, the semiparametric is slightly better. In addition, one must remember that estimation at the optimal regressor settings is based on the correctness of the model. Given that the semiparametric approach has superior model performance statistics, we believe that this new method provides a better fit to the response surface and more accurate optimization results than its parametric and nonparametric counterparts. Ultimately, the only way to determine which approach gives the best optimization results is to perform a confirmatory experiment. Unfortunately, we cannot do so for this example, but we will use simulations to compare the three approaches in general. These simulation results are presented in the next section.

Table 4.6: Comparison of the recommended regressor settings for the motor oil data. Best values in bold.

Approach	x_1	x_2	$\hat{y}$	$\widehat{SDT}$
Parametric	0.00	0.38	2900.063	0.004
Nonparametric	1.00	0.26	2874.621	644.071
Semiparametric	1.00	0.40	**2899.954**	**0.002**

4.6 Simulations

In the chemical process and motor oil examples, the semiparametric fit was observed to be superior to its parametric and nonparametric counterparts. In this section, we compare the three methods more generally via a simulation study. Each Monte Carlo simulation will use 500 simulated data sets, each of which is based on the following underlying model:

$$\begin{aligned} y &= 20 - 10x_1 - 25x_2 - 15x_1x_2 + 20x_1^2 + 50x_2^2 \\ &\quad +\gamma\left[2\sin\left(4\pi x_1\right) + 2\cos\left(4\pi x_2\right) + 2\sin\left(4\pi x_1 x_2\right)\right] + \varepsilon, \end{aligned} \tag{4.5}$$

where $\varepsilon \sim \mathrm{N}(0,1)$ and γ represents the model misspecification parameter. As in the examples, we assume a full second-order model is specified by the user in the parametric method. We also assume this model is specified by the user in the semiparametric method. To mimic the motor oil example, the data will be generated as if a rotatable CCD with five center runs was run for a total of 13 experimental runs. Therefore, when coded to be between 0 and 1, factors x_1 and x_2 have five levels with values taken to be at 0.000, 0.146, 0.500, 0.854, and 1.000 for each factor. The design appears in Table 4.7. As the value of γ increases, the amount of misspecification increases in the model. Five degrees of model misspecification will be studied ($\gamma = 0.00$, 0.25, 0.50, 0.75, and 1.00). Of course, at $\gamma = 0.00$, the user's model is correctly specified. The R language is used for computations and simulations. (Note that this is the same simulation setup as Chapter 3.)

Table 4.7: The central composite design for the Monte Carlo simulations.

i	x_1	x_2
1	0.146	0.146
2	0.854	0.146
3	0.146	0.854
4	0.854	0.854
5	0.000	0.500
6	1.000	0.500
7	0.500	0.000
8	0.500	1.000
9	0.500	0.500
10	0.500	0.500
11	0.500	0.500
12	0.500	0.500
13	0.500	0.500

Figure 4.3 shows the response surface for the true underlying model when $\gamma = 0.00$, and the response surfaces of the function for the varying degrees of model misspecification ($\gamma =$ 0.25, 0.50, 0.75, and 1.00) appear in Figures 4.4 through 4.7, respectively. Notice that as γ increases, the curvature of the surface becomes much more pronounced. The power (P) of the lack of fit test at $\alpha = 0.05$ varies from P = 0.05 at $\gamma = 0.00$ up to P = 0.71 at $\gamma =$ 1.00. Table 4.8 shows the five levels of model misspecification along with the corresponding powers of the lack of fit test.

Figure 4.3: Response surface for the true underlying model when $\gamma = 0.00$.

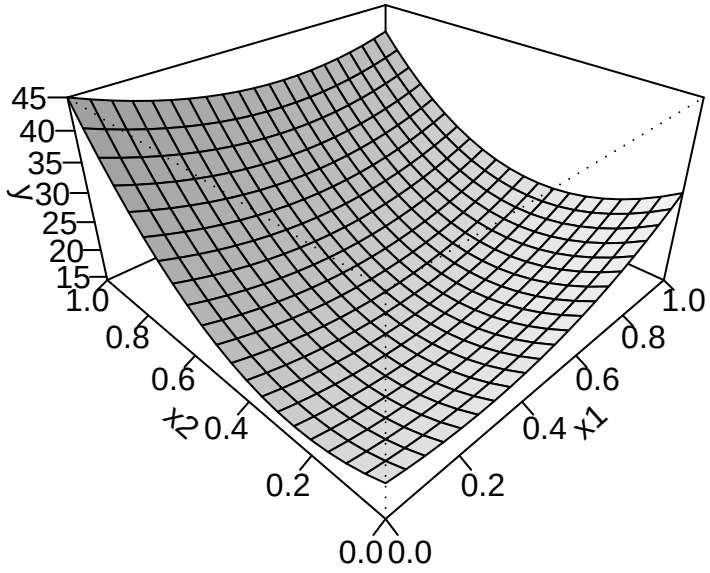


Figure 4.4: Response surface for the model when $\gamma = 0.25$.

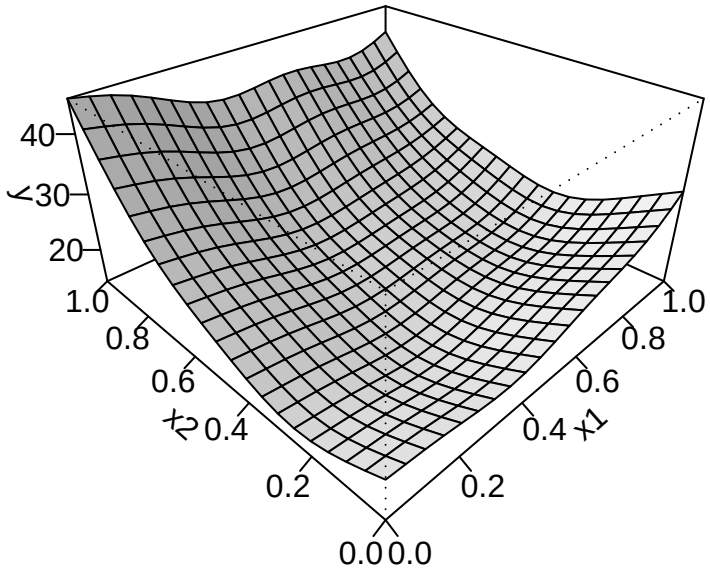


Figure 4.5: Response surface for the model when $\gamma = 0.50$.

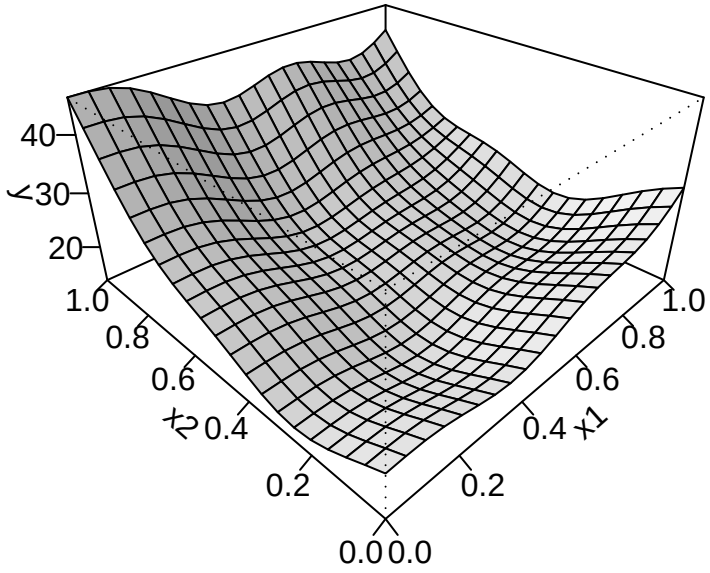


Figure 4.6: Response surface for the model when $\gamma = 0.75$.

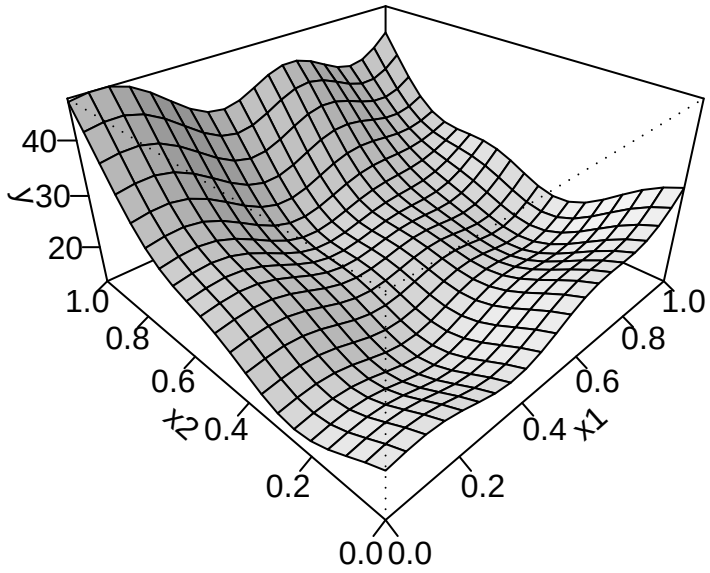


Figure 4.7: Response surface for the model when $\gamma = 1.00$.

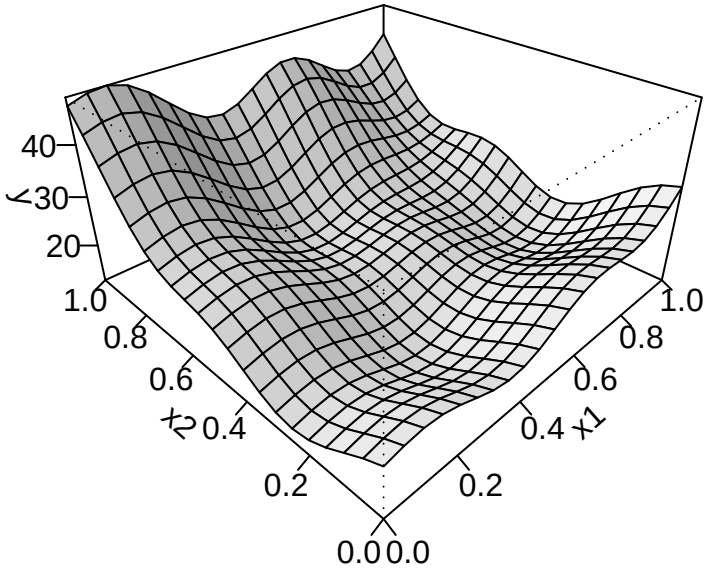


Table 4.8: Power (P) of the lack of fit test at $\alpha = 0.05$ for the five levels of model misspecification.

γ	P
0.00	0.05
0.25	0.09
0.50	0.23
0.75	0.47
1.00	0.71

The three methods, parametric, nonparametric, and semiparametric, will be used to fit the simulated data sets. PRESS** will be used for bandwidth selection for the LLR fit to the raw data in the nonparametric approach and for the LLR fit to the OLS residuals in the semiparametric approach. The semiparametric approach will also use equation (4.4) to determine the estimated asymptotically optimal data driven mixing parameter. After the response surfaces have been fit, optimization will be carried out. Assuming that the goal is to achieve a target value of 15 for the response, the genetic algorithm (GA) with the estimated squared distance from target ($\widehat{SDT}$) objective function given in equation (4.1) will be used

to obtain the optimal regressor settings. Details concerning the GA can be found in Chapter 6.

Comparisons of the three methods will be based on the Monte Carlo simulated integrated mean squared error ($SIMSE$) values given by:

$$SIMSE = \frac{\sum ASE}{500},$$

where

$$ASE = \frac{\sum \left(E[y_i] - \hat{y}_i\right)^2}{1600},$$

where ASE denotes the average squared error for the estimates for each of the 500 simulated data sets and $E[y_i]$ is the true underlying model. Note that the average squared error values, ASE, are calculated across 1600 $\mathbf{x}_0$ locations (based on a 40×40 uniform grid of points in the regressor space) for each of the 500 simulated data sets. This will provide an indication of the fit performance of the methods over the entire response surface.

The three methods will also be compared by the accuracy of their optimization results. Comparisons will be based on the average Euclidean distance from the optimum defined as:

$$AED = \frac{\sum ED}{500},$$

where

$$ED = \sqrt{(x_1^* - x_{opt1}^*)^2 + (x_2^* - x_{opt2}^*)^2},$$

where ED denotes the Euclidean distance between the GA's chosen optimal location, $\mathbf{x}^* = [x_1^*, x_2^*]$, and the true optimum, $\mathbf{x}_{opt}^* = [x_{opt1}^*, x_{opt2}^*]$. The accuracy of the optimization results will also be compared via the average true squared distance from target ($ASDT$),

$$ASDT = \frac{\sum SDT}{500},$$

where

$$SDT = \left(\mu(\mathbf{x}^*) - 15\right)^2,$$

where $\mu(\mathbf{x}^*)$ denotes the true mean response when the optimal location chosen by the algorithm, $\mathbf{x}^*$, is substituted into the true underlying function given by equation (4.5). Even though the true optimal location may not be found by a method, a near-optimal location may be found. Thus, $ASDT$ measures the performance of the locations chosen by each method to see if they are in fact near-optimal.

Table 4.9 provides a comparison of the three approaches based on the $SIMSE$ values for the varying degrees of model misspecification. For the scenario in which the researcher correctly specifies the form of the underlying model (i.e., $\gamma = 0.00$), we would expect the parametric approach to be superior. The first row of Table 4.9 shows that the parametric approach performs best as it yields a $SIMSE$ value of 0.5161. The semiparametric approach is a close second with a value of 0.6022, whereas the nonparametric fit is much worse with a $SIMSE$ value of 5.9604.

The remaining rows of Table 4.9 provide the $SIMSE$ values for the scenario in which the researcher misspecifies the model (i.e., $\gamma > 0.00$). The semiparametric approach outperforms its parametric and nonparametric counterparts for moderate levels of model misspecification (i.e., $\gamma \geq 0.50$). Also note that the nonparametric method is inferior to the other two methods for all levels of model misspecification. The poor performance of the nonparametric method is most likely due to the sparsity of the data as well as the fact that the highest level of model misspecification in the simulation ($\gamma = 1.00$) only corresponds to a power of 0.71. Thus, we conclude that the semiparametric fit is at least highly competitive or superior to the other methods over the entire range of γ values.

Table 4.10 provides a comparison of the three approaches based on the AED values for the varying degrees of model misspecification. When the model is correctly specified ($\gamma = 0.00$), there are nine true optimal locations that achieve a target value of 15 for the response. Note, however, the GA is only run once for each simulated data set. Thus, the GA only yields one solution; whereas, derivative-based methods can yield multiple solutions for the parametric

Table 4.9: Simulated integrated mean squared error ($SIMSE$) values for 500 Monte Carlo runs. Best values in bold.

γ	Parametric	Nonparametric	Semiparametric
0.00	**0.5161**	5.9604	0.6022
0.25	**0.9219**	7.5380	1.0423
0.50	2.2780	9.8439	**2.0362**
0.75	4.3681	12.8733	**3.9306**
1.00	7.2909	16.6219	**6.5812**

approach. In practice, if the user fits a fully parametric model and suspects multiple optima, it is suggested to run the GA several times or use traditional optimization techniques. The parametric approach results in the smallest AED values for four of these nine. When the model is misspecified (i.e., $\gamma > 0.00$), there is only one true optimal location. Note that for all values of γ, the parametric and semiparametric approaches have relatively close AED values while the nonparametric approach yields much larger values.

Table 4.11 provides a comparison of the three approaches based on the $ASDT$ values for the varying degrees of model misspecification. For all degrees of potential model misspecification, the semiparametric approach yields the smallest $ASDT$ values. Also note that for all values of γ the parametric and semiparametric approaches have close $ASDT$ values while the nonparametric approach is clearly inferior. Thus, the results in Tables 4.10 and 4.11 show that optimization based on the semiparametric fit is at least highly competitive or superior to the other methods for all values of γ.

Table 4.10: Average Euclidean distance from the optimum (AED) values for 500 Monte Carlo runs. Best values in bold.

γ	True Optimum	Parametric	Nonparametric	Semiparametric
	(0.500, 0.250)	**0.1824**	0.4856	0.1967
	(0.300, 0.190)	0.2184	0.3119	**0.1749**
	(0.250, 0.200)	0.2335	0.2878	**0.1879**
	(0.190, 0.245)	0.2479	0.2879	**0.2031**
0.00	(0.190, 0.312)	0.2334	0.3341	**0.1973**
	(0.544, 0.312)	**0.1864**	0.5495	0.2201
	(0.545, 0.316)	**0.1863**	0.5521	0.2211
	(0.250, 0.375)	0.2040	0.4030	**0.1866**
	(0.400, 0.420)	**0.1764**	0.5108	0.2003
0.25	(0.500, 0.250)	**0.1744**	0.4963	0.1777
0.50	(0.500, 0.250)	**0.1433**	0.5060	0.2045
0.75	(0.500, 0.250)	**0.1472**	0.4994	0.2069
1.00	(0.500, 0.250)	**0.1431**	0.5019	0.1930

Table 4.11: Average true squared distance from target ($ASDT$) values for 500 Monte Carlo runs. Best values in bold.

γ	Parametric	Nonparametric	Semiparametric
0.00	0.1935	19.7488	**0.0868**
0.25	0.5825	24.2391	**0.2547**
0.50	1.1738	30.0467	**0.5965**
0.75	2.0552	38.4203	**1.3559**
1.00	2.9467	47.9611	**1.6798**

4.7 Conclusions

RSM has utilized parametric regression techniques to study products and processes since its inception. One drawback, however, is that optimization depends too heavily on the

assumption of well estimated models for the response of interest, and it is often the case that the user specified parametric models are not flexible enough to adequately model the process. Nonparametric smoothing has been considered when the user is unable to specify the explicit form for the function. However, in small sample settings, which are customary for response surface experiments, the nonparametric approach often produces estimates that are highly variable. Therefore, we suggest a semiparametric approach which combines the user specified parametric model with a nonparametric fit to provide better estimates.

Using the chemical process and motor oil studies from Myers and Montgomery (2002) to compare the parametric, nonparametric, and semiparametric approaches, we find that the semiparametric approach performs best in terms of several model performance statistics. Moreover, in the motor oil study, the optimization based on the semiparametric approach recommends regressor settings which result in the estimated response being extremely close to the target. In fact, the semiparametric approach yields the smallest squared distance from target value. As previously mentioned, confirmatory experiments are necessary to prove which approach performs best for these examples; nonetheless, the semiparametric produces superior results.

A simulation study was conducted to compare the three approaches more generally. If the user correctly specifies the model, the parametric approach yields the best fit and optimization results with the semiparametric a close second. For moderate model misspecification, the semiparametric method is always superior in terms of fit and yields competitive, if not superior, optimization results. Furthermore, the poor performance of the nonparametric approach is clearly seen in the large $SIMSE$, AED, and $ASDT$ values it produces in the simulations. Since, in practice, one never knows if the form of the underlying model has been correctly specified, we advocate the semiparametric method as it is the only one which consistently performs well over all degrees of potential misspecification.

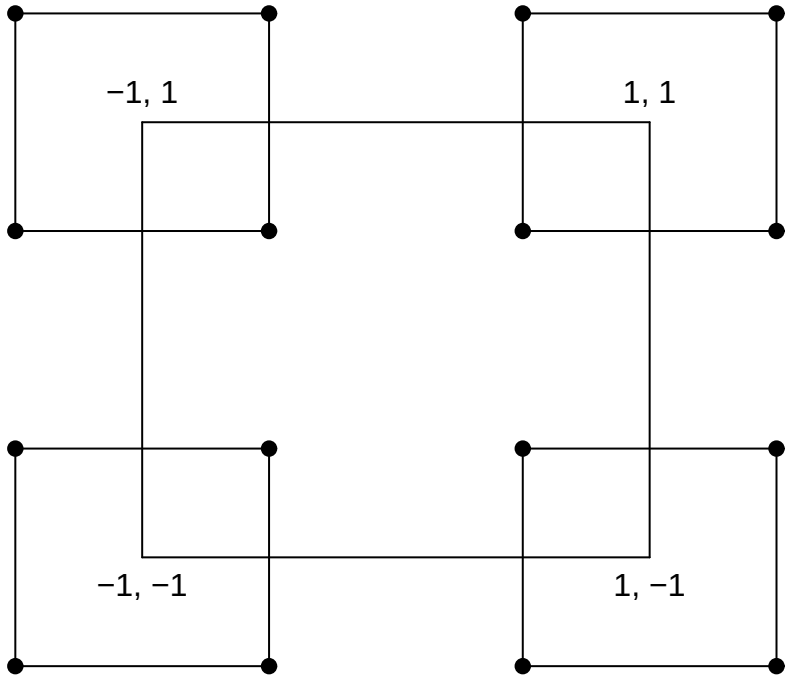
Chapter 5

A Semiparametric Approach to Robust Parameter Design

5.1 Introduction

In the mid 1980's, Japanese quality consultant Genichi Taguchi popularized a cost-efficient approach to quality improvement known as robust parameter design (RPD). Taguchi postulated that there are two types of factors which operate on a process: control factors and noise factors. Control factors are variables whose levels remain unchanged in the process once they are set, whereas the levels of the noise factors change randomly within the process and cause unwanted variation in the response, y. The goal of robust parameter design is to determine levels of the control factors which cause the response to be robust to changes in the levels of the noise variables. A popular design for studying both the impact of control factors and noise factors on a process is the crossed array. A $2^2 \times 2^2$ crossed array is shown in Figure 5.1. Variation in the process which results from uncontrollable fluctuations in the noise factors can be summarized by taking the sample variance of the points in the noise

Figure 5.1: The $2^2 \times 2^2$ crossed array.



factor space at each of the control factor settings (4 points in the $2^2 \times 2^2$ example). The process can be made robust to the variation associated with the noise factors by choosing the factor combination of the control factors corresponding to the smallest sample variance. It is often the case that the levels of the noise factors are unobservable not only in the process but also in a controlled experimental setting. In these situations, replications at the control factor settings provide the researcher with an idea of process variability and the approach to robust design is the same; namely, to choose the factor combination in the control factor space which corresponds to the smallest sample variance. It is these types of situations which will be the focus of this chapter.

Instead of using only the sample variances for describing the underlying process variance, Vining and Myers (1990) introduced a dual model response surface approach to RPD in which it is assumed that both the mean and variance can be described by separate parametric regression models. Optimal control factor settings are then found using constrained optimization (constrained estimated mean and minimized process variance with respect to

control factor settings).

If one or both models are misspecified by the researcher, the estimates may be highly biased and, consequently, the optimal control factor settings may be misspecified. Vining and Bohn (1998) point out traditional parametric models are often inadequate, particularly when modeling the variance, and suggest the use of nonparametric techniques for modeling the variance. Unfortunately, in sparse data situations, which are typically the case with response surface experiments, nonparametric techniques often result in highly variable estimates. To overcome the pitfalls associated with each method, we propose the use of semiparametric modeling within the robust design setting.

By using a semiparametric technique for modeling, we can combine parametric and nonparametric functions to improve the quality of both the mean and variance models. The resulting semiparametric estimates will have smaller bias and variance. Furthermore, these hybrid estimates will result in a better understanding of the process at hand. In turn, the optimal factor settings are less likely to be misspecified. We will illustrate the use and benefit of our proposed method with an application to the Box and Draper (1987) printing ink study and simulations. We also suggest the use of a more flexible optimization routine, the genetic algorithm, for determining optimal control factor settings.

5.2 An Overview of Parametric and Nonparametric Approaches

5.2.1 Parametric Approach

Given the data from a crossed array, there are a number of potential approaches to directly modeling the mean and variance as a function of the control factors. A general approach is to

assume the underlying functional forms for the mean and variance models can be expressed parametrically. Assuming a d point design with n_i replicates at each location ($i = 1, 2, \ldots, d$), the point estimators of the process mean and variance, $\bar{y}_i$ and s_i^2, respectively, form the data for the dual response system. Since the purpose of this research is to demonstrate the utility of a hybrid approach (combining a parametric and nonparametric approach to modeling) for robust design, we will consider an "off the shelf" model for the mean. An "off the shelf" model for the process mean is linear in the model parameters and can be written as:

$$\text{Means Model:} \qquad \bar{y}_i = \mathbf{x}_i'\boldsymbol{\beta} + g^{1/2}\left(\mathbf{x}_i^{*\prime}; \boldsymbol{\gamma}\right)\varepsilon_i, \tag{5.1}$$

where $\mathbf{x}_i'$ and $\mathbf{x}_i^{*\prime}$ are $1\times(k+1)$ and $1\times l$ vectors of means model and variance model regressors, respectively, expanded to model form, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are $(k+1)\times 1$ and $m \times 1$ vectors of mean and variance model parameters, respectively, g is the underlying variance function, and ε_i denotes the random error for the mean function. The ε_i are assumed to be uncorrelated with mean zero and variance of one. Note that the model terms for the i^{th} observation in the means model are denoted by $\mathbf{x}_i'$ while the model terms for the variance model are denoted by $\mathbf{x}_i^{*\prime}$. This allows for the fact that the process mean and variance may not depend on the same set of regressors.

Similar to the modeling of the mean, various modeling strategies have been utilized for estimating the underlying variance function. Bartlett and Kendall (1946) demonstrated that if the errors are normal about the mean model and if the design points are replicated, the variance can be modeled via a log-linear model with the d sample variances utilized for the responses. A great deal of work has also been done using generalized linear models for estimating the variance function. Although not an exhaustive list, the reader is referred to Box and Meyer (1986), Aitkin (1987), Grego (1993), Lee and Nelder (2003), and Myers, Brenneman, and Myers (2005). As mentioned previously, the purpose of this research is to demonstrate the utility of a hybrid approach to modeling; thus, we choose an "off the shelf" approach to variance modeling. The log-linear model proposed by Bartlett and Kendall (1946) is a popular one [see Vining and Myers (1990) and Myers and Montgomery (2002)]

and is written explicitly as:

$$\text{Variance Model:} \qquad \ln(s_i^2) = \ln\left(g(\mathbf{x}_i^*;\boldsymbol{\gamma})\right) = g^*\left(\mathbf{x}_i^*\right) + \eta_i = \mathbf{x}_i^{*\prime}\boldsymbol{\gamma} + \eta_i, \tag{5.2}$$

where η_i denotes the model error term whose expectation is assumed to be zero and whose variance, $\delta^2 = \frac{2}{n_i - 1}$, is assumed constant across the d design points. Note that the variance is only constant if $n_i = n$ for all i.

Assuming the model forms for the mean and variance given in (5.1) and (5.2), the model parameters are estimated using the following estimated weighted least squares (EWLS) algorithm:

Step 1: Fit the variance model, $\ln(s_i^2) = \mathbf{x}_i^{*\prime}\boldsymbol{\gamma} + \eta_i$, via ordinary least squares (OLS), obtaining $\hat{\boldsymbol{\gamma}}^{(OLS)} = \left(\mathbf{X}^{*\prime}\mathbf{X}^*\right)^{-1}\mathbf{X}^{*\prime}\mathbf{y}^*$ where $\mathbf{y}^*$ is the $d \times 1$ vector of log transformed sample variances.

Step 2: Use $\hat{\sigma}_i^2 = \exp\left(\mathbf{x}_i^{*\prime}\hat{\boldsymbol{\gamma}}^{(OLS)}\right)$ as the estimated variances to compute the $d \times d$ estimated variance-covariance matrix for the means model, $\hat{\mathbf{V}} = diag\left(\hat{\sigma}_1^2, \hat{\sigma}_2^2, \ldots, \hat{\sigma}_d^2\right)$.

Step 3: Use $\hat{\mathbf{V}}^{-1}$ as the estimated weight matrix to fit the means model, yielding $\hat{\boldsymbol{\beta}}^{(EWLS)} = \left(\mathbf{X}'\hat{\mathbf{V}}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\hat{\mathbf{V}}^{-1}\bar{\mathbf{y}}$ where $\bar{\mathbf{y}}$ denotes the $d \times 1$ vector of sample averages.

The algorithm above yields the following estimates of the process mean and variance functions:

$$\text{Estimated Process Mean:} \qquad \widehat{E[y_i]}^{(EWLS)} = \mathbf{x}_i'\hat{\boldsymbol{\beta}}^{(EWLS)} \tag{5.3}$$

$$\text{Estimated Process Variance:} \qquad \widehat{Var[y_i]}^{(OLS)} = \exp\left(\mathbf{x}_i^{*\prime}\hat{\boldsymbol{\gamma}}^{(OLS)}\right). \tag{5.4}$$

Once estimates of the mean and variance have been calculated, the goal becomes finding the operating conditions for the control factors such that the mean is as close as possible to the target while maintaining minimum process variance.

Any control factor which influences the expression in (5.4) is known as a *dispersion* factor. Any control factor that does not influence the expression in (5.4) but does influence the expression in (5.3) is known as an *adjustment* factor. When both dispersion and adjustment factors are present, the robust design problem can be approached in a two-step fashion. Specifically, levels of the dispersion factors are chosen so as to minimize the estimated process variance in (5.4), and then the levels of the adjustment factors are chosen so as to bring the estimated process mean in (5.3) to a desired level. If only dispersion factors are present and these factors also influence the process mean, the researcher is left with finding the levels of the control factors that yield a desirable trade-off between low variance and a deviation from the targeted mean. This is often accomplished via minimization of an objective function such as the estimated squared error loss ($\widehat{SEL}$):

$$\widehat{SEL} = E\,\widehat{[y(\mathbf{x}) - T]}^2 = \{\widehat{E\,[y(\mathbf{x})]} - T\}^2 + \widehat{Var\,[y(\mathbf{x})]}, \tag{5.5}$$

where T denotes the target value for the process mean, $\widehat{E\,[y(\mathbf{x})]}$ denotes the estimated process mean, and $\widehat{Var\,[y(\mathbf{x})]}$ denotes the estimated process variance. Minimization can be accomplished via non-linear programming using a method such as the generalized reduce gradient or the Nelder-Mead simplex algorithm. The squared error loss approach is also useful when adjustment factors are present but are not strong enough to bring the mean to the targeted value. Note that the determined set of optimal operating conditions is highly dependent on quality estimation of both the mean and variance functions. Misspecification of the forms of either the mean or variance models can have serious implications in process optimization.

5.2.2 Nonparametric Approach

Situations may arise in which the user cannot explicitly state parametric forms for the dual model. In these situations, parametric specifications may result in serious bias of the estimated mean and/or variance. To prevent the bias induced by parametric model

misspecification, Vining and Bohn (1998) and Anderson-Cook and Prewitt (2005) suggest the use of nonparametric regression for estimating the process mean and variance. Expressing the dual model where the mean and variance functions (h and g^*, respectively) are assumed to have unknown but smooth forms we have:

$$\text{Means Model:} \qquad \bar{y}_i = h\left(\tilde{\mathbf{x}}_i\right) + g^{1/2}\left(\tilde{\mathbf{x}}_i^*\right)\varepsilon_i$$

$$\text{Variance Model:} \qquad \ln(s_i^2) = g^*\left(\tilde{\mathbf{x}}_i^*\right) + \eta_i,$$

where $\tilde{\mathbf{x}}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ and $\tilde{\mathbf{x}}_i^* = (x_{i1}^*, x_{i2}^*, \ldots, x_{il}^*)$.

Similar to parametric regression, estimators are linear combinations of the response values $\bar{y}_i$ and $\ln(s_i^2)$; however, the weighting schemes in some nonparametric regression methods assign more weight to observations closest to the point of prediction. The nonparametric fits are more flexible than the parametric fits as they are not confined to the user's specified form. This enables the nonparametric approach to more adequately fit processes whose underlying models have more complicated forms than those expressed by the linear models in (5.1) and (5.2).

Several fitting techniques have been proposed in the nonparametric regression literature such as kernel regression [see for example Nadaraya (1964), Watson (1964), Priestley and Chao (1972), and Gasser and Müller (1984)], local polynomial models [see for example Fan and Gijbels (1996) and Fan and Gijbels (2000)], spline-based smoothers, and series-based smoothers [see for example Ruppert, Wand, and Carrol (2003)]. Vining and Bohn (1998) first applied nonparametric smoothing in the RPD setting by using the Gasser-Müller estimator for the dual response problem. Anderson-Cook and Prewitt (2005) continued with this idea by using the Nadaraya-Watson estimator and local polynomial regression, the method used in this research. Local polynomial regression (LPR) is a popular class of nonparametric smoothing methods and is particularly appealing in response surface applications due to its robustness to biased estimates at the boundary of the design space. LPR is essentially a

weighted least squares (WLS) problem where the weights are given by a kernel function. The polynomial form of the local polynomial fit can be of order one or greater, and we focus on degree $p = 1$, local linear regression (LLR), in this research.

For the multiple regressor case, at point $\tilde{\mathbf{x}}_0 = (x_{01}, x_{02}, \ldots, x_{0k})$ where prediction is desired, we define the kernel function as:

$$\mathbf{K}(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_i) = \frac{1}{b^k} \prod_{j=1}^{k} K\left(\frac{\tilde{x}_{0j} - \tilde{x}_{ij}}{b}\right), \tag{5.6}$$

where $\tilde{\mathbf{x}}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$, $K\left(\frac{\tilde{x}_{0j} - \tilde{x}_{ij}}{b}\right)$ is a univariate kernel function, and b is the bandwidth. Note that when estimating both the mean and variance nonparametrically, a different kernel function may be used for the mean than for the variance since the regressors effecting the mean do not necessarily effect the variance. The choice of kernel function is not crucial to the performance of the estimator (Simonoff, 1996). Thus, for convenience, we will use the simplified Gaussian kernel, $K(u) = e^{-u^2}$.

The smoothness of the estimated function is controlled by the bandwidth, b. Since the coding of variables in response surface designs typically involves centering and scaling, the units are comparable in all directions. Thus, it is reasonable to use the same bandwidth, b, in all dimensions as expressed in (5.6). The choice of bandwidth is critical, and the literature is rich with bandwidth selection methods [see for example Härdle (1990), Härdle et al. (2004)]. Typically the bandwidth is chosen to minimize some optimality criteria such as MSE. Mays, Birch, and Starnes (2001) introduce a penalized cross-validation technique, PRESS**, for choosing an appropriate bandwidth. The approach chooses the bandwidth as the value b that minimizes PRESS**, defined as:

$$\text{PRESS}^{**} = \frac{\text{PRESS}}{d - \text{trace}\left(\mathbf{H}^{(LLR)}\right) + \left(d - (k+1)\right)\frac{SSE_{max} - SSE_b}{SSE_{max}}},$$

where SSE_{max} is the largest error sum of squares over all possible bandwidth values, SSE_b is the error sum of squares associated with a particular bandwidth value b, k is the number

of regressors, and the prediction error sum of squares, PRESS, is given by:

$$\text{PRESS} = \sum_{i=1}^{d} \left(y_i - \hat{y}_{i,-i}\right)^2,$$

where $\hat{y}_{i,-i}$ denotes the estimated response obtained by leaving out the i^{th} observation when estimating at location $\tilde{\mathbf{x}}_i$. The LLR smoother matrix, $\mathbf{H}^{(LLR)}$, is defined as:

$$\mathbf{H}^{(LLR)} = \begin{bmatrix} \mathbf{h}_1^{(LLR)\prime} \\ \mathbf{h}_2^{(LLR)\prime} \\ \vdots \\ \mathbf{h}_d^{(LLR)\prime} \end{bmatrix}$$

where $\mathbf{h}_i^{(LLR)\prime}$ is defined below. Mays, Birch, and Starnes (2001) show that PRESS** performs well by guarding against very small and very large bandwidths.

The nonparametric estimate of the dual model is found using the following estimated weighted local linear regression (EWLLR) algorithm:

Step 1: Fit the variance model, $\ln(s_i^2) = g^*(\mathbf{x}_i^{*\prime}) + \eta_i$, via local linear regression (LLR), obtaining $\hat{\boldsymbol{\gamma}}^{(LLR)} = \left(\mathbf{X}^{*\prime}\mathbf{W}_i^*\mathbf{X}^*\right)^{-1}\mathbf{X}^{*\prime}\mathbf{W}_i^*\mathbf{y}^*$, where $\mathbf{y}^*$ is the $d \times 1$ vector of log transformed sample variances and $\mathbf{W}_i^*$ is the diagonal matrix containing the kernel weights associated with $\tilde{\mathbf{x}}_i^*$, $\mathbf{W}_i^* = diag\left(h_{i1}^{(KER)}, h_{i2}^{(KER)}, \ldots, h_{id}^{(KER)}\right)$ with $h_{ij}^{(KER)} = \frac{\mathbf{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)}{\sum_{j=1}^{d}\mathbf{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)}$.

Step 2: Use $\hat{\sigma}_i^2 = \exp(\tilde{\mathbf{x}}_i^{*\prime}\hat{\boldsymbol{\gamma}}^{(LLR)})$ as the estimated variances to compute the $d \times d$ variance-covariance matrix for the means model, $\hat{\mathbf{V}} = diag\left(\hat{\sigma}_1^2, \hat{\sigma}_2^2, \ldots, \hat{\sigma}_d^2\right)$.

Step 3: Use $\hat{\mathbf{V}}^{-1}$ as the estimated weight matrix to fit the means model via estimated weighted local linear regression (EWLLR), yielding $\hat{\boldsymbol{\beta}}^{(EWLLR)} = \left(\mathbf{X}'\mathbf{W}_i\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}_i\bar{\mathbf{y}}$, where $\bar{\mathbf{y}}$ denotes the $d \times 1$ vector of sample averages and $\mathbf{W}_i = \left\langle\sqrt{h_{ij}^{(KER)}}\right\rangle \hat{\mathbf{V}}^{-1} \left\langle\sqrt{h_{ij}^{(KER)}}\right\rangle$ where $\left\langle\sqrt{h_{ij}^{(KER)}}\right\rangle$ is the diagonal matrix containing the square roots of the kernel weights associated with $\tilde{\mathbf{x}}_i$, $\left\langle\sqrt{h_{ij}^{(KER)}}\right\rangle = diag\left(\sqrt{h_{i1}^{(KER)}}, \sqrt{h_{i2}^{(KER)}}, \ldots, \sqrt{h_{id}^{(KER)}}\right)$ with $h_{ij}^{(KER)} = \frac{\mathbf{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)}{\sum_{j=1}^{d}\mathbf{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)}$.

The algorithm yields the following estimates of the process mean and variance functions:

$$\begin{aligned}
\text{Estimated Process Mean:} \quad \widehat{E[y_0]}^{(EWLLR)} &= \tilde{\mathbf{x}}_0' \hat{\boldsymbol{\beta}}^{(EWLLR)} \\
&= \tilde{\mathbf{x}}_0' \left(\mathbf{X}'\mathbf{W}_0\mathbf{X}\right)^{-1} \mathbf{X}'\mathbf{W}_0\bar{\mathbf{y}} \\
&= \mathbf{h}_0^{(EWLLR)\prime}\bar{\mathbf{y}} \qquad (5.7) \\
\text{Estimated Process Variance:} \quad \widehat{Var[y_0]}^{(LLR)} &= \exp\left(\tilde{\mathbf{x}}_0^{*\prime}\hat{\boldsymbol{\gamma}}^{(LLR)}\right) \\
&= \exp\left[\tilde{\mathbf{x}}_0^{*\prime}\left(\mathbf{X}^{*\prime}\mathbf{W}_0^*\mathbf{X}^*\right)^{-1}\mathbf{X}^{*\prime}\mathbf{W}_0^*\mathbf{y}^*\right] \\
&= \exp\left(\mathbf{h}_0^{(LLR)\prime}\mathbf{y}^*\right). \qquad (5.8)
\end{aligned}$$

For more information regarding weighted LLR, the reader is referred to Lin and Carroll (2000). Under the assumption of normality of $\bar{\mathbf{y}}$ and $\mathbf{y}^*$, the estimates of $E[y_0]$ and $Var[y_0]$ given by (5.7) and (5.8) are the local maximum likelihood estimates of Fan, Heckman, and Wand (1995).

Similar to the parametric approach to robust design, once estimates of the mean and variance functions have been calculated, a squared error loss approach will be used for process optimization. Unfortunately, most of the analytic optimization methods suggested for the parametric approach are based on gradient techniques which require continuous functions with derivatives for the estimated mean and variance functions. Since the mean and variance estimates from nonparametric methods do not result in closed form expressions, these optimization routines are no longer applicable. Vining and Bohn (1998) utilize a simplex search based on the AMOEBA algorithm (Vetterling et al., 1992) which does not require the calculation of derivatives; however, simplex methods tend to be time consuming and often find local, as opposed to global optima [for details, see Haupt and Haupt (2004)]. Therefore we advocate the use of genetic algorithms for optimization.

The genetic algorithm (GA), originally developed by Holland (1975), has become a popular optimization technique. It is especially useful for optimizing functions that do not have known parametric forms as it does not require derivatives to find the optimal solutions. Instead, the GA is based on the principles of genetics and uses evolutionary concepts such as

selection, crossover, and mutation to find the optimal solutions. Furthermore, GA uses an intelligent, sequential search strategy which enables the user to find global, not local, solutions more efficiently (Goldberg, 1989). Thus, we will use the GA for process optimization. Details of our GA algorithm are presented in Chapter 6.

5.2.3 Parametric vs. Nonparametric

Parametric and nonparametric approaches to modeling each possess positive and negative attributes. The parametric method is superior if the true, underlying functions can be adequately expressed parametrically and if the user correctly specifies the parametric forms. However, if either of the models is misspecified, the estimates may be highly biased and optimal control factor settings may be miscalculated. On the other hand, if the user has no idea about the true form of the underlying functions, nonparametric methods offer a nice alternative. Nonparametric methods can provide superior fits by capturing structure in the data that a misspecified parametric model cannot. However, nonparametric methods were originally developed for situations with large sample sizes whereas a main underpinning of RSM is the use of cost-efficient experimental designs (i.e., small sample sizes). In small sample settings, nonparametric fitting techniques may fit irregularities in the data too closely thereby creating estimated mean and variance functions that are highly variable. Consequently, optimization may be based on non-reproducible idiosyncrasies in the data. Mays, Birch, and Starnes (2001) introduce methods which are essentially hybrids of the parametric and nonparametric methods. These semiparametric approaches produce estimated functions which are characterized by lower bias than parametric approaches and lower variance than nonparametric approaches. The details of this hybrid approach appear in the next section.

5.3 A Semiparametric Approach

When used individually, both parametric and nonparametric regression methods have shortcomings. In this section, we present the use of semiparametric techniques for overcoming some of these drawbacks especially in situations where the user has partial knowledge of the underlying model or the data contains important "bumps" that parametric models cannot capture. The semiparametric estimates proposed combine individual parametric and nonparametric fits via appropriately chosen mixing parameters. We detail the methodologies below.

5.3.1 Model Robust Regression 1 (MRR1)

Einsporn and Birch (1993) proposed a semiparametric method for modeling the mean response for assumed constant error variance. Their technique, model robust regression 1 (MRR1), combines parametric and nonparametric fits to the raw data in a convex combination via a mixing parameter, λ. For instance, if $\hat{\mathbf{y}}^{(OLS)}$ denotes the vector of ordinary least squares estimates of the mean and $\hat{\mathbf{y}}^{(LLR)}$ denotes the vector of local linear regression estimates of the mean, then the MRR1 estimated mean responses are obtained as:

$$\hat{\mathbf{y}}^{(MRR1)} = \lambda\hat{\mathbf{y}}^{(LLR)} + (1-\lambda)\hat{\mathbf{y}}^{(OLS)},$$

where $\lambda \in [0, 1]$. For cases where the user's specified parametric model is correct, the optimal value of the mixing parameter, λ, is 0 and as the amount of model misspecification increases, λ increases from 0 to 1.

Similar to the choice of bandwidth in LLR, the choice of mixing parameter, λ, involves a bias-variance trade-off. Mays, Birch, and Starnes (2001) derive the following data driven

expression for the asymptotically optimal value of the mixing parameter, λ, for MRR1:

$$\hat{\lambda}_{opt}^{(MRR1)} = \frac{\left\langle \hat{\mathbf{y}}_{-i}^{(LLR)} - \hat{\mathbf{y}}_{-i}^{(OLS)}, \mathbf{y} - \hat{\mathbf{y}}^{(OLS)} \right\rangle}{\left\| \hat{\mathbf{y}}^{(LLR)} - \hat{\mathbf{y}}^{(OLS)} \right\|^2}, \tag{5.9}$$

where the i^{th} observations of $\hat{\mathbf{y}}_{-i}^{(LLR)}$ and $\hat{\mathbf{y}}_{-i}^{(OLS)}$ are $\hat{y}_{i,-i}^{(LLR)}$ and $\hat{y}_{i,-i}^{(OLS)}$, $i = 1, \ldots, n$, respectively. The values $\hat{y}_{i,-i}^{(LLR)}$ and $\hat{y}_{i,-i}^{(OLS)}$ denote the LLR and OLS estimates, respectively, obtained by leaving out the i^{th} observation when estimating at $\tilde{\mathbf{x}}_i$. The notation $\langle\ \rangle$ represents the inner product and $\|\ \|$ represents the standard L_2 (Euclidean) norm.

MRR1 produces a smooth estimate that captures important anomalies in the data, which parametric methods are incapable of modeling. By containing a parametric portion for the overall fit, MRR1 brings stability to the overall fit and eliminates over-fitting the data, a problem associated with nonparametric regression. Thus, MRR1 estimates often have smaller bias and variance than their individual parametric and nonparametric counterparts, especially for small sample sizes. It should be noted, however, if there are locations in the data where both the parametric and nonparametric estimates are too high or too low, then the MRR1 estimates will also be too high or too low as the method has no means to correct for the error.

5.3.2 Model Robust Regression 2 (MRR2)

Model robust regression 2 (MRR2) was introduced by Mays, Birch, and Einsporn (2000) as an improvement to the MRR1 approach for estimating the mean with constant variance. Similar to MRR1, MRR2 combines a parametric fit and a nonparametric fit via a mixing parameter; however, the parametric fit is to the raw data (as in MRR1) while the nonparametric fit is to the residuals from the parametric fit. The vector of residuals ($\mathbf{r}$) represents the structure in the data which is not captured by the user specified parametric model. The vector of residuals is fit nonparametrically via LLR resulting in the following vector of smoothed

residuals:

$$\hat{\mathbf{r}} = \mathbf{H}_r^{(LLR)}\mathbf{r},$$

where $\mathbf{H}_r^{(LLR)}$ is computed similarly to the LLR smoother matrix in (5.7) but with the "response" variable being the residuals from the OLS fit to the raw data. The MRR2 estimates are then obtained by adding a portion of the LLR smoothed residuals back to the original parametric fit, yielding:

$$\hat{\mathbf{y}}^{(MRR2)} = \hat{\mathbf{y}}^{(OLS)} + \lambda\hat{\mathbf{r}},$$

where $\lambda \in [0, 1]$. The size of the mixing parameter for MRR2 does not necessarily represent the amount of model misspecification as it does in MRR1. Instead, the value of λ indicates the amount of correction needed from the residual fit. Notice that $\lambda = 1$ actually represents only a 50% contribution from the residual fit as the coefficient for the parametric fit is always one. Similar to MRR1, Mays, Birch, and Starnes (2001) derive a data driven expression for the asymptotically optimal mixing parameter. The expression is given as:

$$\hat{\lambda}_{opt}^{(MRR2)} = \frac{\left\langle \hat{\mathbf{r}}, \mathbf{y} - \hat{\mathbf{y}}^{(OLS)} \right\rangle}{\|\hat{\mathbf{r}}\|^2}. \tag{5.10}$$

Mays, Birch, and Starnes (2001) show that both MRR methods perform as well or better than the individual parametric and nonparametric methods for varying degrees of model misspecification both in an asymptotic sense as well as small sample settings.

5.3.3 Dual Model Robust Regression (DMRR)

Robinson and Birch (2002) extend the MRR techniques to models with non-constant error variance. Robinson and Birch (2002) consider the unreplicated design case and therefore use a residual-based variance estimate. Similar to the results of Mays, Birch, and Starnes (2001), Robinson and Birch (2002) show that dual model robust regression (DMRR) is asymptotically superior to its parametric and nonparametric counterparts. Robinson and Birch (2002)

also show that DMRR performs better in moderate sample settings. Consequently, we propose DMRR as a natural tool for the small sample settings of RSM and the RPD problem. Although originally developed for the regression case with little or no replication, we demonstrate here its extension to replicated, cost-efficient designs in which the variance may be modeled directly.

Assuming that the process mean and variance functions can be expressed as functions with two components, a user supplied parametric component and a "lack of fit" component, the dual model can be written as:

$$\text{Means Model:} \qquad \bar{y}_i = h\left(\tilde{\mathbf{x}}_i; \boldsymbol{\beta}\right) + m\left(\tilde{\mathbf{x}}_i\right) + g^{1/2}\left(\tilde{\mathbf{x}}_i^*; \boldsymbol{\gamma}\right)\varepsilon_i,$$

$$\text{Variance Model:} \qquad \ln(s_i^2) = g^*\left(\tilde{\mathbf{x}}_i^*; \boldsymbol{\gamma}\right) + l\left(\tilde{\mathbf{x}}_i^*\right) + \eta_i.$$

Regarding notation, $h\left(\tilde{\mathbf{x}}_i; \boldsymbol{\beta}\right)$ and $g^*\left(\tilde{\mathbf{x}}_i^*; \boldsymbol{\gamma}\right)$ denote the user specified parametric forms for the mean and variance functions [for purposes of discussion in this chapter we assume $h\left(\tilde{\mathbf{x}}_i; \boldsymbol{\beta}\right) = \mathbf{x}_i'\boldsymbol{\beta}$ and $g^*\left(\tilde{\mathbf{x}}_i^*; \boldsymbol{\gamma}\right) = \mathbf{x}_i^{*\prime}\boldsymbol{\gamma}$]. The "lack of fit" components for the mean and variance functions, $m\left(\tilde{\mathbf{x}}_i\right)$ and $l\left(\tilde{\mathbf{x}}_i^*\right)$, respectively, represent the portions of the mean and variance functions which cannot be captured parametrically. The only assumptions placed on m and l is that they are reasonably smooth functions.

As mentioned earlier, Mays, Birch, and Starnes (2001) demonstrate that MRR2 performs better than MRR1 overall. However, for variance modeling, there is nothing to guarantee positive fits in the MRR2 approach. Hence for the dual modeling problem, we suggest the use of MRR1 for estimation of the variance and MRR2 for estimation of the means model. Thus, the following algorithm can be used to find the DMRR estimates:

Step 1: Fit the variance model, $\ln(s_i^2) = \mathbf{x}_i^{*\prime}\boldsymbol{\gamma} + l\left(\tilde{\mathbf{x}}_i^*\right) + \eta_i$, via model robust regression 1

(MRR1). MRR1 yields the variance model robust regression (VMRR) estimates:

$$\begin{aligned}\widehat{Var[\mathbf{y}]}^{(VMRR)} &= \exp\left[\lambda_\sigma \hat{\mathbf{y}}^{*(LLR)} + (1-\lambda_\sigma)\,\hat{\mathbf{y}}^{*(OLS)}\right] \\ &= \exp\left[\lambda_\sigma \mathbf{H}_\sigma^{(LLR)}\mathbf{y}^* + (1-\lambda_\sigma)\,\mathbf{H}_\sigma^{(OLS)}\mathbf{y}^*\right] \\ &= \exp\left[\mathbf{H}_\sigma^{(VMRR)}\mathbf{y}^*\right],\end{aligned}$$

where $\lambda_\sigma \in [0,1]$ is the variance model mixing parameter and $\mathbf{H}_\sigma^{(VMRR)}$ is the smoother matrix for the VMRR fit to the log transformed sample variances.

Step 2: Use $\hat{\sigma}_i^2 = \exp\left(\mathbf{h}_{i,\sigma}^{(VMRR)\prime}\mathbf{y}^*\right)$ as the estimated variances to compute the estimated variance-covariance matrix for the means model, $\hat{\mathbf{V}} = diag\left(\hat{\sigma}_1^2, \hat{\sigma}_2^2, \ldots, \hat{\sigma}_d^2\right)$, where $\mathbf{h}_{i,\sigma}^{(VMRR)\prime}$ is the i^{th} row of $\mathbf{H}_\sigma^{(VMRR)}$.

Step 3: Use $\hat{\mathbf{V}}^{-1}$ as the estimated weight matrix to obtain the parametric estimate of the means model via estimated weighted least squares (EWLS). EWLS yields $\hat{\boldsymbol{\beta}}^{(EWLS)} = \left(\mathbf{X}'\hat{\mathbf{V}}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\hat{\mathbf{V}}^{-1}\bar{\mathbf{y}}$ and $\widehat{E[y_i]}^{(EWLS)} = \mathbf{x}_i'\hat{\boldsymbol{\beta}}^{(EWLS)} = \mathbf{H}_\mu^{(EWLS)}\bar{\mathbf{y}}$.

Step 4: Form the residuals from the EWLS fits found in Step 3, $\mathbf{r} = \bar{\mathbf{y}} - \widehat{E[\mathbf{y}]}^{(EWLS)}$, and perform local linear regression on the residuals. LLR yields $\hat{\mathbf{r}} = \mathbf{H}_r^{(LLR)}\mathbf{r}$, where $\mathbf{H}_r^{(LLR)}$ is the smoother matrix for the LLR fit to the residuals from the EWLS fit to the means model.

Step 5: Obtain the means model robust regression (MMRR) estimates via MRR2 as:

$$\begin{aligned}\widehat{E[\mathbf{y}]}^{(MMRR)} &= \widehat{E[\mathbf{y}]}^{(EWLS)} + \lambda_\mu \hat{\mathbf{r}} \\ &= \left[\mathbf{H}_\mu^{(EWLS)} + \lambda_\mu \mathbf{H}_r^{(LLR)}\left(\mathbf{I} - \mathbf{H}_\mu^{(EWLS)}\right)\right]\bar{\mathbf{y}}, \\ &= \mathbf{H}_\mu^{(MMRR)}\bar{\mathbf{y}}\end{aligned}$$

where $\lambda_\mu \in [0,1]$ is the means model mixing parameter.

For the nonparametric estimates, the bandwidths, b_μ and b_σ, will be chosen as the values that minimize PRESS**. The mixing parameters, λ_σ and λ_μ, for the variance and mean

fits, respectively, will be chosen via the asymptotically optimal expressions for MRR1 and MRR2 as given in (5.9) and (5.10), respectively, with $\hat{\mathbf{y}}^{(OLS)}$ replaced with $\hat{\mathbf{y}}^{(EWLS)}$. Similar to the parametric and nonparametric approaches, once estimates of the mean and variance functions have been calculated, a squared error loss approach will be used for process optimization. Furthermore, as in the nonparametric approach, the genetic algorithm will be used for optimization since the estimates of the mean and variance functions do not take on closed form expressions.

5.4 The Printing Ink Example

The Box and Draper (1987) printing ink study has been analyzed throughout the RPD literature [see for example Vining and Myers (1990) and Vining and Bohn (1998)]. The purpose of the study was to examine the effect of three factors, speed (x_1), pressure (x_2), and distance (x_3), on a printing machine's ability to apply ink to package labels. The experiment used a 3^2 complete factorial with three replicates at each design point. The goal of the study was to find an optimal location where the process variance is minimized and the process mean achieves a target value of 500. Table 5.1 provides the results of the experiment. Note that two locations ($i = 10$ and 14) have a sample standard deviation of zero. Thus, we will replace the observed sample variances, s_i^2, with $s_i^2 + 1$ to accommodate the log transformation for the variance model. For our purposes, we will assume the user has specified a first-order model for the log transformed variance model and a second-order model for the mean.

Table 5.1: The printing ink data.

i	x_{1i}	x_{2i}	x_{3i}	y_{1i}	y_{2i}	y_{3i}	$\bar{y}_i$	s_i
1	-1	-1	-1	34	10	28	24.00	12.49
2	0	-1	-1	115	116	130	120.33	8.39
3	1	-1	-1	192	186	263	213.67	42.83
4	-1	0	-1	82	88	88	86.00	3.46
5	0	0	-1	44	178	188	136.67	80.41
6	1	0	-1	322	350	350	340.67	16.17
7	-1	1	-1	141	110	86	112.33	27.57
8	0	1	-1	259	251	259	256.33	4.62
9	1	1	-1	290	280	245	271.67	23.63
10	-1	-1	0	81	81	81	81.00	0.00
11	0	-1	0	90	122	93	101.67	17.67
12	1	-1	0	319	376	376	357.00	32.91
13	-1	0	0	180	180	154	171.33	15.01
14	0	0	0	372	372	372	372.00	0.00
15	1	0	0	541	568	396	501.67	92.50
16	-1	1	0	288	192	312	264.00	63.50
17	0	1	0	432	336	513	427.00	88.61
18	1	1	0	713	725	754	730.67	21.08
19	-1	-1	1	364	99	199	220.67	133.82
20	0	-1	1	232	221	266	239.67	23.46
21	1	-1	1	408	415	443	422.00	18.52
22	-1	0	1	182	233	182	199.00	29.44
23	0	0	1	507	515	434	485.33	44.64
24	1	0	1	846	535	640	673.67	158.21
25	-1	1	1	236	126	168	176.67	55.51
26	0	1	1	660	440	403	501.00	138.94
27	1	1	1	878	991	1161	1010.00	142.45

The nonparametric and semiparametric approaches involve the choice of an appropriate global bandwidth for the kernel function. Using PRESS**, we obtain a bandwidth of 0.63 for the variance model and 0.52 for the means model in the nonparametric approach. These bandwidths meet the recommendations of Anderson-Cook and Prewitt (2005). A bandwidth of 0.51 was chosen for the nonparametric smooth of the EWLS residuals in the semiparametric fit to the mean.

Furthermore, the semiparametric approach involves the choice of appropriate mixing parameters, λ_σ and λ_μ. For the variance model, the asymptotically optimal data driven mixing parameter is found to be 0.6812. This value suggests there is a moderate amount of variance model misspecification; that is, there are some trends in the data that the parametric model cannot adequately capture. The asymptotically optimal data driven mixing parameter for the means model is found to be 1.0. Therefore, the addition of the entire nonparametric residual fit provides necessary correction to the parametric means model.

Table 5.2 provides the MSE and adjusted R^2 values for the parametric, nonparametric and semiparametric approaches. The semiparametric method is clearly superior for the variance model as it results in the lowest MSE value, 5.74, and the highest adjusted R^2 value, 23.40%. The nonparametric method appears to be best for the means model as it has the lowest MSE and highest adjusted R^2 values; however, the semiparametric is highly competitive. This data set is an example of the situation in which one method does not clearly dominate the others, a common problem when dealing with multiple responses. For this data, the semiparametric method is preferred as it yields the best fit to the variance model, and when the variance estimates are incorporated into the means model it yields a highly competitive fit.

Using the genetic algorithm with the estimated squared error loss ($\widehat{SEL}$) objective function as given in (5.5), we obtain the optimal factor settings displayed in Table 5.3. Notice that all three approaches suggest an x_1 value of 1.000. Also the nonparametric and semiparametric

Table 5.2: Comparison of model performance statistics for the printing ink data. Best values in bold.

	$MeansModel$		$VarianceModel$	
Approach	MSE	R^2_{adj}	MSE	R^2_{adj}
Parametric	7.07	96.72%	5.89	21.33%
Nonparametric	**4.09**	**97.90%**	5.90	21.26%
Semiparametric	4.49	97.79%	**5.74**	**23.40%**

methods both recommend an x_2 value of 1.000. The most dramatic difference in the solutions is seen in the x_3 coordinate. While all three methods suggest negative values for the factor, the actual settings for x_3 vary greatly. Coincidentally, x_3 is the most significant factor in the parametric variance model.

In Table 5.3, we also see the optimal factor settings for the semiparametric approach ($x_1 = 1.000, x_2 = 1.000, x_3 = -0.522$) yield an estimated process mean of 497.629, which is closest to the target value of 500, and an estimated process variance of 1019.523, the lowest among the three methods. As a result, the semiparametric approach also results in the lowest estimated SEL value. Thus, we conclude this new method performs better than its parametric and nonparametric counterparts. Ultimately, the only way to determine which approach gives the best optimization results is to perform a confirmatory experiment. Unfortunately, we cannot do so for this example, but we can use simulations to compare the three approaches in general.

Table 5.3: Comparison of the recommended optimal factor settings for the printing ink data. Best values in bold.

Approach	x_1	x_2	x_3	$\widehat{E[y_i]}$	$\widehat{Var[y_i]}$	$\widehat{SEL}$
Parametric	1.000	0.358	-0.112	497.619	1723.693	1729.363
Nonparametric	1.000	1.000	-0.352	496.866	1088.455	1098.276
Semiparametric	1.000	1.000	-0.522	**497.629**	**1019.523**	**1025.150**

5.5 Simulations

In the printing ink example, the semiparametric fit was observed to be superior to its parametric and nonparametric counterparts in terms of $\widehat{SEL}$. In this section, we compare the three methods more generally in terms of fit via a simulation study. The performance of the semiparametric approach will be compared to the parametric and nonparametric approaches in four scenarios: the researcher correctly specifies the forms of both the underlying mean and variance functions, the researcher correctly specifies the form of the underlying variance function but misspecifies the means model, the researcher correctly specifies the form of the means model but misspecifies the variance model, and the researcher incorrectly specifies the forms of both the underlying mean and variance functions. For each scenario, Monte Carlo simulations will be used to generate 500 data sets, each of which are based on the following

underlying dual model:

$$\begin{aligned} y_i &= h\left(\tilde{\mathbf{x}}_i;\boldsymbol{\beta}\right) + m\left(\tilde{\mathbf{x}}_i\right) + g^{1/2}\left(\tilde{\mathbf{x}}_i^*;\boldsymbol{\gamma}\right)\varepsilon_i \\ &= 20 - 10x_{1i} - 25x_{2i} - 15x_{1i}x_{2i} + 20x_{1i}^2 + 50x_{2i}^2 \\ &\quad +\gamma_\mu\left[10\sin\left(4\pi x_{1i}\right) + 10\cos\left(4\pi x_{2i}\right) + 10\sin\left(4\pi x_{1i}x_{2i}\right)\right] \\ &\quad +g^{1/2}\left(\tilde{\mathbf{x}}_i^*\right)\varepsilon_i \end{aligned} \tag{5.11}$$

$$\begin{aligned} \ln\left(\sigma_i^2\right) &= g^*\left(\tilde{\mathbf{x}}_i^*;\boldsymbol{\gamma}\right) + l\left(\tilde{\mathbf{x}}_i^*\right) \\ &= 1.5 - x_{1i} + 1.5x_{2i} + \gamma_\sigma\left[-4x_{1i}x_{2i} + 2x_{1i}^2 + x_{2i}^2\right], \end{aligned} \tag{5.12}$$

where $\varepsilon_i \sim \mathrm{N}(0,1)$, γ_μ represents the means model misspecification parameter, and γ_σ represents the variance model misspecification parameter. As in the printing ink example, we assume a full second-order model is specified by the user for the mean and a first-order model specification for the log transformed sample variances. In all scenarios, factors x_1 and x_2 have four levels with values taken to be 0, 1/3, 2/3, and 1 for each factor. The data will be generated as if a 4^2 complete factorial experiment was run with three replicates at each design point for a total of 16 design points and 48 experimental runs. The design point locations appear in Table 5.4. As the values of γ_μ and γ_σ increase, the amount of misspecification increases in the means and variance model, respectively. Five degrees of means model misspecification will be studied (γ_μ = 0.00, 0.25, 0.50, 0.75, and 1.00), and five degrees of variance model misspecification will be studied (γ_σ = 0.00, 0.25, 0.50, 0.75, and 1.00). The R language is used for computations and simulations.

Figure 5.2 shows the response surface for the true underlying means model when $\gamma_\mu = 0.00$, and the response surfaces of the mean function for the varying degrees of model misspecification (γ_μ = 0.25, 0.50, 0.75, and 1.00) appear in Figures 5.3 through 5.6, respectively. Notice that as γ_μ increases, the curvature of the mean surface becomes much more pronounced. Figure 5.7 shows the response surface for the true underlying variance model when γ_σ = 0.00, and the response surfaces of the variance function for the varying degrees of model

Table 5.4: The design point locations for the Monte Carlo simulations.

i	x_1	x_2
1	0	0
2	0	1/3
3	0	2/3
4	0	1
5	1/3	0
6	1/3	1/3
7	1/3	2/3
8	1/3	1
9	2/3	0
10	2/3	1/3
11	2/3	2/3
12	2/3	1
13	1	0
14	1	1/3
15	1	2/3
16	1	1

misspecification ($\gamma_\sigma = 0.25$, 0.50, 0.75, and 1.00) appear in Figures 5.8 through 5.11, respectively. Again notice the curvature of the variance surface becomes much more pronounced as γ_σ increases. When the variance model is correctly specified, the power (P) of the means model lack of fit test at $\alpha = 0.05$ varies from P = 0.12 at $\gamma_\mu = 0.00$ up to P = 0.99 at $\gamma_\mu = 1.00$. Table 5.5 shows the levels of potential model misspecification along with the corresponding powers of the lack of fit test.

Figure 5.2: Response surface for the true underlying means model when $\gamma_\mu = 0.00$.

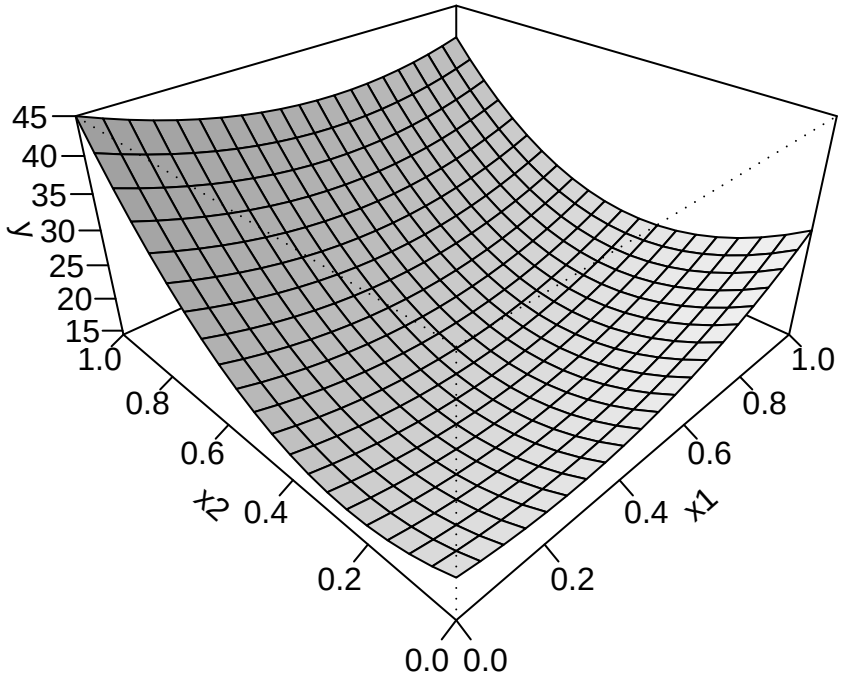


Figure 5.3: Response surface for the means model when $\gamma_\mu = 0.25$.

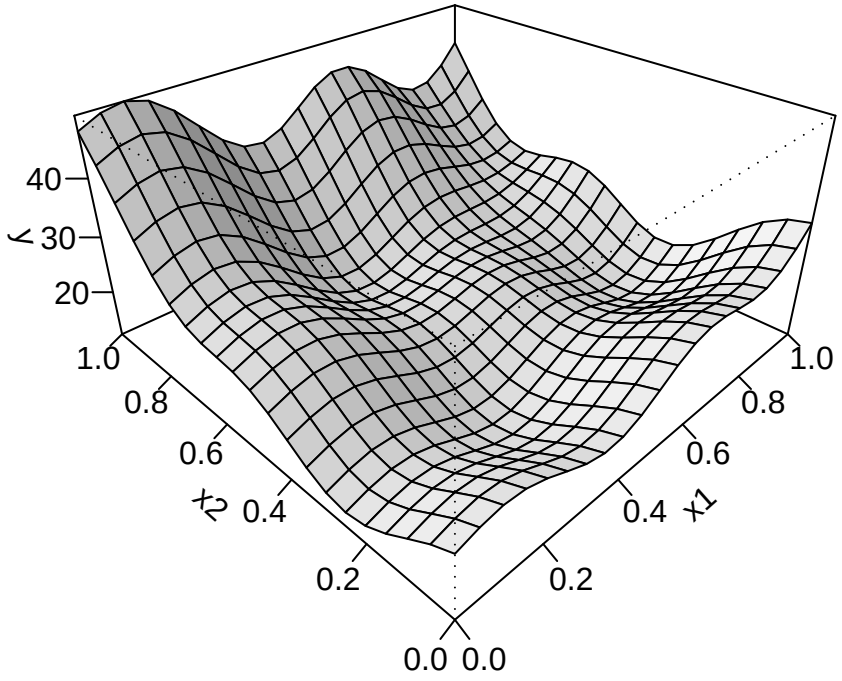


Figure 5.4: Response surface for the means model when $\gamma_\mu = 0.50$.

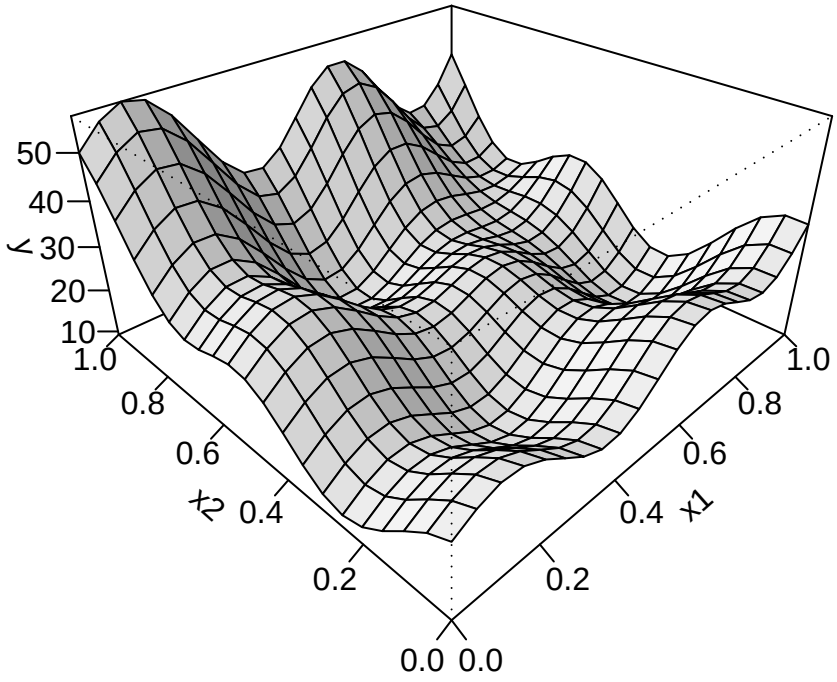


Figure 5.5: Response surface for the means model when $\gamma_\mu = 0.75$.

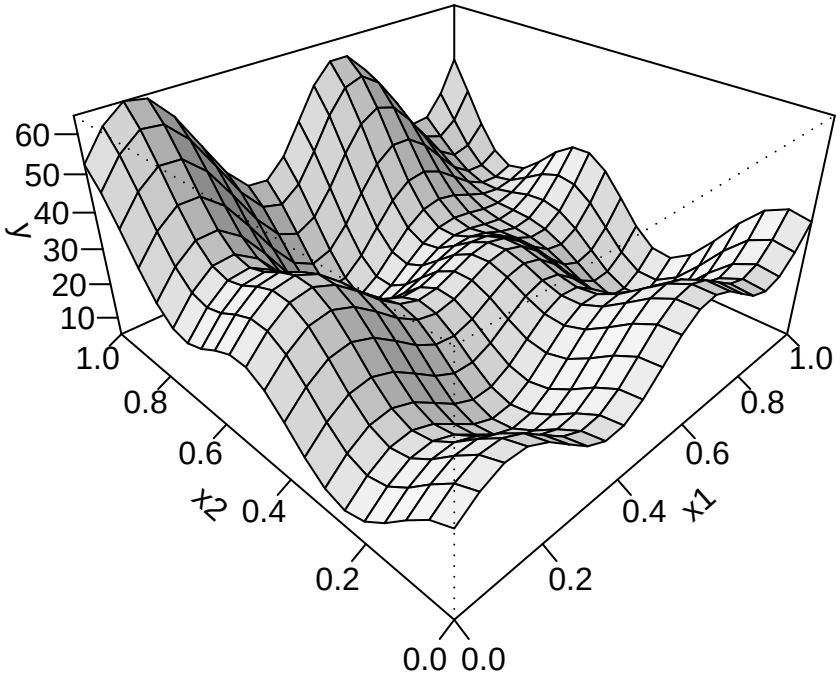


Figure 5.6: Response surface for the means model when $\gamma_\mu = 1.00$.

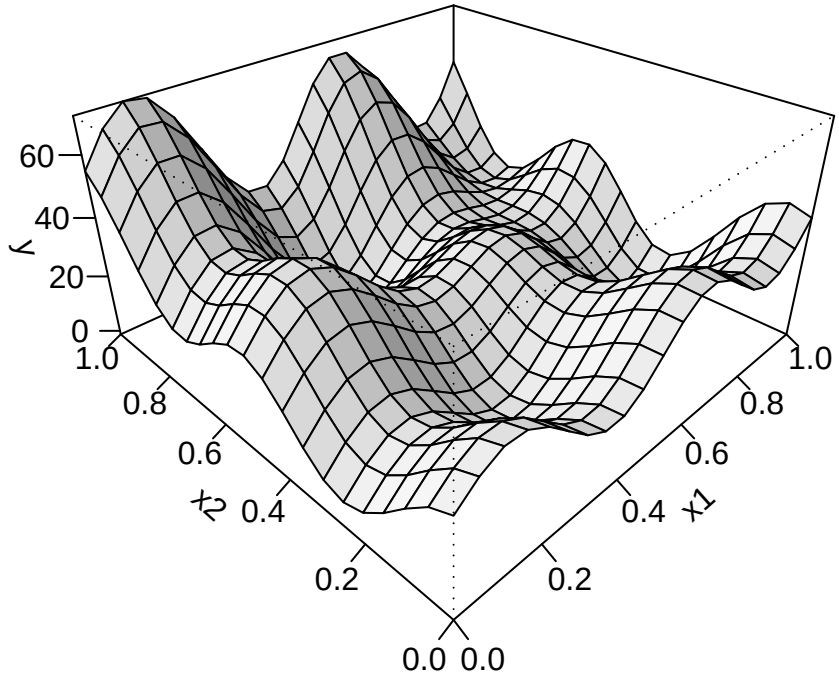


Figure 5.7: Response surface for the true underlying variance model when $\gamma_\sigma = 0.00$.

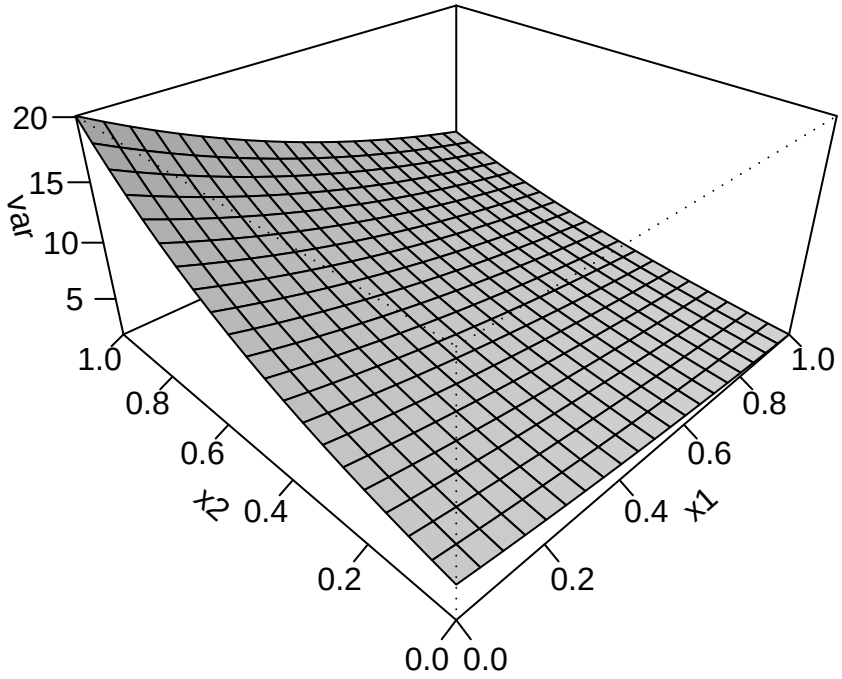


Figure 5.8: Response surface for the variance model when $\gamma_\sigma = 0.25$.

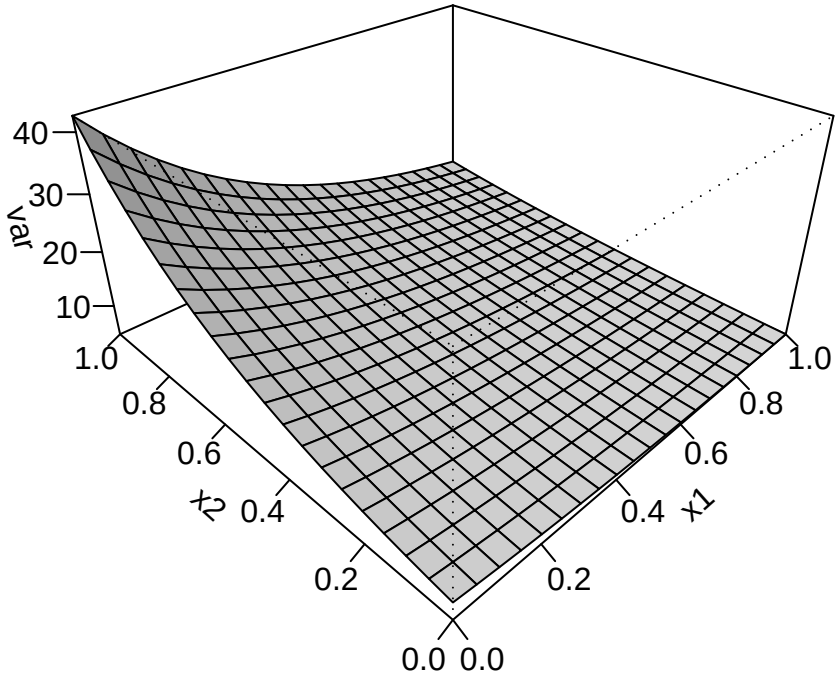


Figure 5.9: Response surface for the variance model when $\gamma_\sigma = 0.50$.

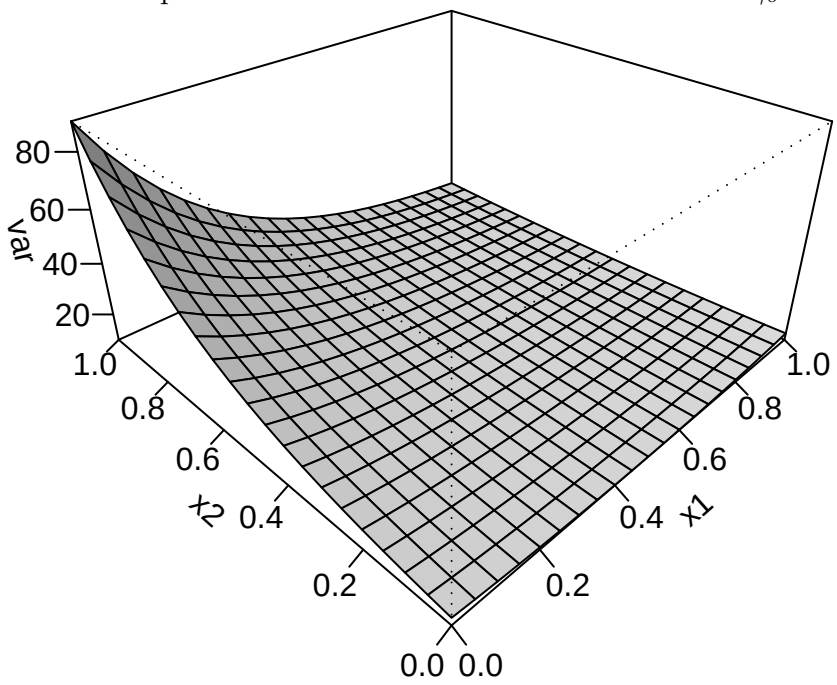


Figure 5.10: Response surface for the variance model when $\gamma_\sigma = 0.75$.

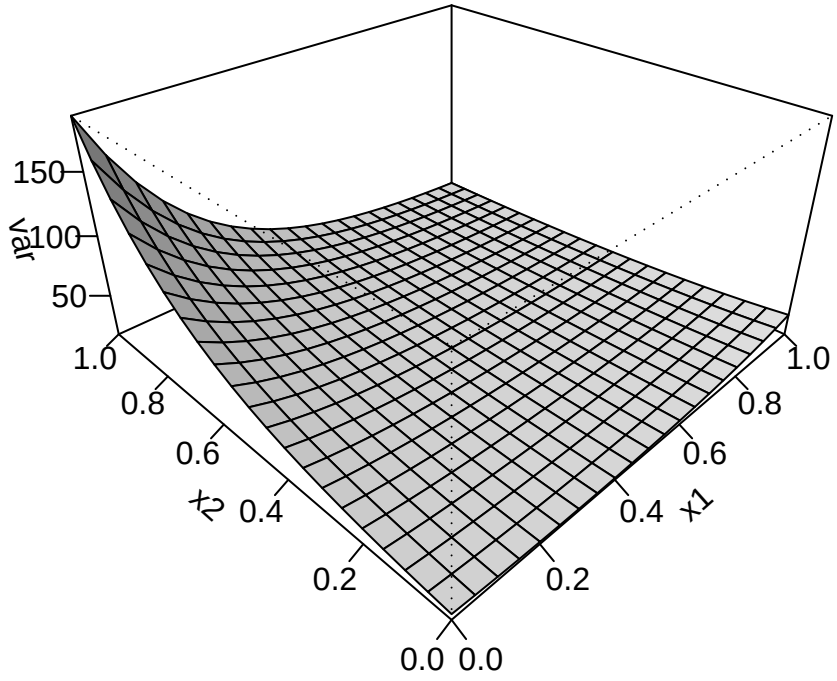


Figure 5.11: Response surface for the variance model when $\gamma_\sigma = 1.00$.

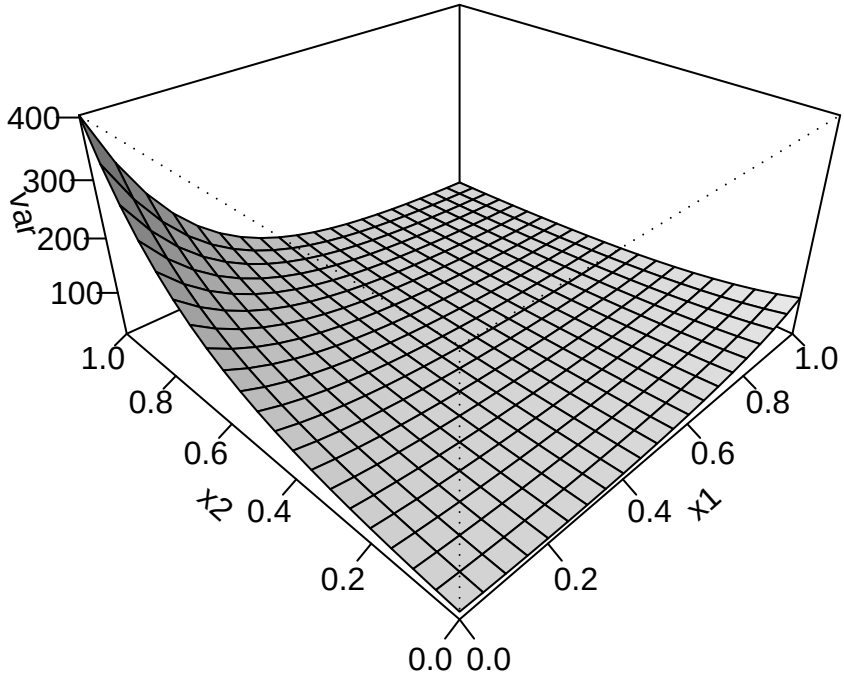


Table 5.5: Power (P) of the means model lack of fit test at $\alpha = 0.05$ for the levels of potential model misspecification.

	Power (P)				
γ_μ	$\gamma_\sigma = 0.00$	$\gamma_\sigma = 0.25$	$\gamma_\sigma = 0.50$	$\gamma_\sigma = 0.75$	$\gamma_\sigma = 1.00$
0.00	0.05000	0.05000	0.05000	0.05000	0.05000
0.25	0.11716	0.12652	0.14018	0.15911	0.18472
0.50	0.42813	0.47895	0.54771	0.63174	0.72506
0.75	0.84959	0.89368	0.93702	0.97015	0.98941
1.00	0.99022	0.99572	0.99874	0.99977	0.99998

The three methods, parametric, nonparametric, and semiparametric, will be used to fit the simulated data sets. After the response surfaces have been fit, optimization will be carried out. Assuming that the goal is to a achieve a target value of 15 for the mean response while minimizing the process variance, the genetic algorithm (GA) with the estimated squared error loss ($\widehat{SEL}$) objective function given in equation (5.5) will be used to obtain the optimal regressor settings. Details concerning the GA can be found in Chapter 6.

The Monte Carlo simulated integrated mean squared error for the mean and variance estimates given by:

$$SIMSEM = \frac{\sum ASEM}{500}, \qquad ASEM = \frac{\sum \left(E[y_i] - \hat{y}_i\right)^2}{1600},$$

$$SIMSEV = \frac{\sum ASEV}{500}, \qquad ASEV = \frac{\sum \left(\sigma_i^2 - \hat{\sigma}_i^2\right)^2}{1600}$$

will be used for comparisons. Regarding notation, $ASEM$ and $ASEV$ denote the average squared error for the mean and variance estimates, respectively, for each of the 500 simulated data sets and $E[y_i]$ and σ_i^2 are the true underlying mean and variance models, respectively. Note that for the simulated integrated mean squared error values, the average squared error values, $ASEM$ and $ASEV$, are calculated across 1600 $\mathbf{x}_0$ locations (based on a 40×40 uniform grid of points in the regressor space) for each of the 500 simulated data sets. This

will provide an indication of the performance of the methods over the entire response surface. These results are found in Tables 5.6 and 5.7.

The three methods will also be compared by the accuracy of their optimization results. Comparisons will be based on the average Euclidean distance from the optimum defined as:

$$AED = \frac{\sum ED}{500},$$

where

$$ED = \sqrt{(x_1^* - x_{opt1}^*)^2 + (x_2^* - x_{opt2}^*)^2},$$

where ED denotes the Euclidean distance between the GA's chosen optimal location, $\mathbf{x}^* = [x_1^*, x_2^*]$, and the true optimum, $\mathbf{x}_{opt}^* = [x_{opt1}^*, x_{opt2}^*]$. The accuracy of the optimization results will also be compared via the average true squared error loss ($ASEL$),

$$ASEL = \frac{\sum SEL}{500},$$

where

$$SEL = \{E\left[y(\mathbf{x}^*)\right] - 15\}^2 + Var\left[y(\mathbf{x}^*)\right],$$

where $E\left[y(\mathbf{x}^*)\right]$ denotes the true process mean and $Var\left[y(\mathbf{x}^*)\right]$ denotes the true process variance when the optimal location chosen by the GA, $\mathbf{x}^*$, is substituted into the true underlying functions given by equations (5.11) and (5.12). Even though the true optimal location may not be found by a method, a near-optimal location may be found. Thus, $ASEL$ measures the performance of the locations chosen by each method to see if they are in fact near-optimal. These results are found in Tables 5.8 and 5.9.

5.5.1 Means and Variance Models Correctly Specified

For the scenario in which the researcher correctly specifies the forms of both the underlying mean and variance functions, we would expect the parametric approach to be superior for

both models. In this scenario, the means model is given by (5.11) with $\gamma_\mu = 0.00$ and the variance model is given by (5.12) with $\gamma_\sigma = 0.00$. The first row of Table 5.6 and the first row of Table 5.7 provide a comparison of the fits produced by the three approaches based on the SIMSE values for this scenario. Regarding the estimated mean, Table 5.6 shows that the parametric approach performs best as it yields a SIMSEM value of 0.4335. The semiparametric approach is a close second with a value of 0.6151, whereas the nonparametric fit is much worse with a SIMSEM value of 7.9748. As for the estimated variance, Table 5.7 shows that the parametric approach performs best with a SIMSEV value of 17.9190.

The first row of Table 5.8 and the first row of Table 5.9 provide a comparison of the optimization of the three approaches based on the AED and $ASEL$ values, respectively. The semiparametric approach yields the most accurate optimization results with an AED value of 0.0724 and an $ASEL$ value of 4.0493. Notice that the parametric method is second with an AED value of 0.1011 and an $ASEL$ value of 4.2950. Note when $\gamma_\mu = \gamma_\sigma = 0.00$, one would expect the parametric approach to do best. It is interesting, however, that in this scenario the semiparametric method is a close competitor to the parametric approach in terms of $SIMSEM$ yet is better in terms of AED and $ASEL$.

5.5.2 Variance Model Correctly Specified, Means Model Misspecified

The rows of Table 5.6 corresponding to $\gamma_\sigma = 0.00$ and $\gamma_\mu > 0.00$ (i.e., rows 6, 11, 16, and 21) provide the SIMSEM values for the scenario in which the researcher correctly specifies the variance model but misspecifies the mean model. The semiparametric approach outperforms its parametric and nonparametric counterparts for moderate levels of mean misspecification (i.e., $\gamma_\mu \leq 0.50$) and competes closely with the nonparametric approach for more extreme levels of misspecification (i.e., $\gamma_\mu \geq 0.75$). Thus, the semiparametric fit to the mean is at least highly competitive or superior to the other methods for the entire range of γ_μ.

Unlike the results in Chapter 4, the nonparametric approach actually performs best in some cases. This may be due to the fact that we are working with a larger sample size. However, it is most likely due to the fact that more extreme levels of model misspecification are present in this chapter's study. Here $\gamma_\mu = 0.75$ and $\gamma_\mu = 1.00$ correspond to power levels of 0.84959 and 0.99022, whereas the highest level of misspecification in Chapter 4 only corresponds to a power of 0.71.

The rows of Tables 5.8 and 5.9 corresponding to $\gamma_\sigma = 0.00$ and $\gamma_\mu > 0.00$ (i.e., rows 6, 11, 16, and 21) provide a comparison of the optimization of the three approaches based on the AED and $ASEL$ values, respectively. The semiparametric approach outperforms its parametric and nonparametric counterparts in terms of AED and $ASEL$ for all levels of mean misspecification (i.e., $\gamma_\mu \geq 0.25$). It is interesting to note that the parametric approach becomes less competitive in terms of AED and $ASEL$ as the amount of mean misspecification increases.

5.5.3 Means Model Correctly Specified, Variance Model Misspecified

The rows of Table 5.6 corresponding to $\gamma_\mu = 0.00$ and $\gamma_\sigma > 0.00$ (i.e., rows 2 through 5) show the SIMSEM values for the scenario in which the researcher correctly specifies the means model but misspecifies the form of the variance. Even though the estimated variances are incorporated in the weighted estimation of the mean, variance misspecification has little impact on the estimated mean. Table 5.6 shows that the SIMSEM values consistently increase for all three approaches as the variance misspecification increases. Nonetheless, the parametric and semiparametric approaches still perform best when the means model is correctly specified, whereas the nonparametric fit is consistently subpar. As for the estimated variance, rows 2 through 5 of Table 5.7 provide the SIMSEV values for each of the methods. The semiparametric fit is best when there is moderate to large variance misspecification (i.e.,

$\gamma_\sigma > 0.50$) and is competitive for lower degrees of misspecification.

The rows of Tables 5.8 and 5.9 corresponding to $\gamma_\mu = 0.00$ and $\gamma_\sigma > 0.00$ (i.e., rows 2 through 5) provide a comparison of the optimization of the three approaches based on the AED and $ASEL$ values, respectively. Similar to the fit results from Table 5.6, the parametric and semiparametric approaches yield the best optimization results in terms of AED and $ASEL$ when the means model is correctly specified, whereas the optimization results based on the nonparametric fits are consistently subpar.

5.5.4 Means and Variance Models Misspecified

The remainder of the results in Table 5.6 provide a comparison for the scenario in which the researcher misspecifies the forms of both the underlying mean and variance functions (i.e., $\gamma_\mu > 0.00$ and $\gamma_\sigma > 0.00$). Again, it appears that variance misspecification has little impact on mean estimation. The biggest impact on mean estimation is the user's specification of the underlying means model. Overall, the semiparametric approach yields the best fit if there is small to moderate mean misspecification and/or moderate to large variance misspecification. It is important to note that even when the semiparametric approach is not best in terms of SIMSEM or SIMSEV, its fit is highly competitive with the superior approach. While the parametric and nonparametric methods are best in some situations (parametric SIMSEM and SIMSEV values best when $\gamma_\mu = \gamma_\sigma = 0.00$ and nonparametric SIMSEM values best when $\gamma_\mu \geq 0.75$), these methods are noticeably inferior when there are small to moderate levels of misspecification.

The remainder of the results in Tables 5.8 and 5.9 provide a comparison of the optimization of the three approaches based on the AED and $ASEL$ values, respectively, for this scenario. The semiparametric approach yields the best results in terms of $ASEL$ if there is small to large mean misspecification. In terms of AED, the semiparametric approach is best for

nearly all instances.

Table 5.6: Simulated integrated mean squared error values for the means model (SIMSEM) for 500 Monte Carlo runs. Best values in bold.

γ_μ	γ_σ	Parametric	Nonparametric	Semiparametric
	0.00	**0.4335**	7.9748	0.6151
	0.25	**0.4503**	8.2321	0.6520
0.00	0.50	**0.4806**	8.4893	0.7148
	0.75	**0.5285**	8.6231	0.8200
	1.00	**0.6001**	8.6842	0.9801
	0.00	11.7193	16.2605	**11.1885**
	0.25	11.6521	16.3836	**10.9981**
0.25	0.50	11.6099	16.6595	**10.8863**
	0.75	11.5983	17.1650	**10.8326**
	1.00	11.9108	18.0681	**11.1520**
	0.00	44.5181	42.9345	**40.9599**
	0.25	44.4003	42.9855	**40.4008**
0.50	0.50	44.8819	43.4888	**40.1894**
	0.75	44.8342	43.8464	**39.9063**
	1.00	44.8238	45.4837	**39.8345**
	0.00	98.8521	**87.5581**	90.1062
	0.25	98.6950	**87.5757**	89.9734
0.75	0.50	98.5082	**87.7151**	87.9934
	0.75	98.3134	88.2083	**87.1736**
	1.00	98.1409	90.1727	**86.6895**
	0.00	174.7138	**150.1532**	158.6082
	0.25	174.5361	**150.1094**	156.7779
1.00	0.50	174.2771	**150.2388**	155.0822
	0.75	175.6504	**151.2052**	154.1700
	1.00	174.4280	153.0754	**152.1491**

Table 5.7: Simulated integrated mean squared error values for the variance model (SIMSEV) for 500 Monte Carlo runs. Best values in bold.

γ_σ	Parametric	Nonparametric	Semiparametric
0.00	**17.9190**	19.8901	19.5246
0.25	**17.7432**	19.8656	19.3408
0.50	20.0876	21.0396	**19.9436**
0.75	25.6025	23.6397	**21.6610**
1.00	35.5426	28.2999	**25.6706**

Table 5.8: Average Euclidean distance from the optimum (AED) values for 500 Monte Carlo runs. Best values in bold.

γ_μ	γ_σ	True Optimum	Parametric	Nonparametric	Semiparametric
	0.00	(0.505, 0.223)	0.1011	0.2810	**0.0724**
	0.25	(0.498, 0.224)	0.1086	0.2638	**0.0636**
0.00	0.50	(0.490, 0.225)	0.1040	0.2754	**0.0720**
	0.75	(0.479, 0.225)	0.1141	0.2684	**0.0740**
	1.00	(0.553, 0.344)	**0.0978**	0.3060	0.1347
	0.00	(0.884, 0.287)	0.3433	0.5482	**0.2964**
	0.25	(0.878, 0.285)	**0.3318**	0.5951	0.3806
0.25	0.50	(0.821, 0.284)	**0.2583**	0.5512	0.2720
	0.75	(0.397, 0.128)	0.2677	0.3212	**0.2272**
	1.00	(0.365, 0.121)	0.3169	0.2682	**0.2520**
	0.00	(0.938, 0.237)	0.3656	0.4817	**0.1852**
	0.25	(0.909, 0.221)	0.3517	0.4699	**0.1978**
0.50	0.50	(0.792, 0.233)	0.2450	0.4566	**0.1697**
	0.75	(0.388, 0.108)	0.3374	0.3457	**0.2365**
	1.00	(0.821, 0.482)	0.2483	0.5185	**0.2298**
	0.00	(0.956, 0.221)	0.3317	0.2579	**0.1142**
	0.25	(0.914, 0.198)	0.3270	0.2962	**0.1372**
0.75	0.50	(0.400, 0.105)	0.3958	0.5410	**0.2098**
	0.75	(0.383, 0.098)	0.4254	0.5577	**0.3314**
	1.00	(0.824, 0.526)	0.2205	0.3520	**0.2201**
	0.00	(0.964, 0.212)	0.2497	0.1414	**0.0863**
	0.25	(0.917, 0.186)	0.2799	0.1897	**0.0905**
1.00	0.50	(0.395, 0.097)	0.5643	0.6287	**0.2417**
	0.75	(0.979, 0.811)	0.5391	0.4765	**0.5223**
	1.00	(0.959, 0.834)	0.5599	0.5162	**0.4947**

Table 5.9: Average true squared error loss ($ASEL$) values for 500 Monte Carlo runs. Best values in bold.

γ_μ	γ_σ	Parametric	Nonparametric	Semiparametric
	0.00	4.2950	8.0181	**4.0493**
	0.25	**2.9750**	7.0030	4.1309
0.00	0.50	**2.0077**	7.2870	4.2284
	0.75	**1.3938**	7.6302	4.3042
	1.00	**1.1006**	5.7866	4.3498
	0.00	10.7660	15.9554	**7.9375**
	0.25	11.2540	15.4123	**8.5887**
0.25	0.50	10.4378	15.7669	**7.7797**
	0.75	10.8649	18.1747	**5.9049**
	1.00	10.3401	15.7474	**8.4816**
	0.00	34.1655	18.4602	**10.1547**
	0.25	34.0395	18.4733	**11.3221**
0.50	0.50	33.2392	16.4935	**7.9387**
	0.75	33.5957	20.2035	**8.3529**
	1.00	31.6775	20.8441	**11.1490**
	0.00	31.1030	27.1435	**20.3383**
	0.25	35.5223	29.1970	**22.6705**
0.75	0.50	52.0394	16.6875	**11.6994**
	0.75	45.8278	21.2485	**13.7411**
	1.00	51.6164	20.6265	**12.4291**
	0.00	88.1398	81.1504	**26.8886**
	0.25	74.3165	54.8601	**23.4247**
1.00	0.50	76.7317	53.8253	**17.2676**
	0.75	82.0923	47.5193	**10.2459**
	1.00	75.9720	30.5004	**15.0197**

5.6 Conclusions

The dual model response surface approach to RPD has been shown to work well when the variance of response is not constant over the experimental region and can be successfully modeled using regression methods. One drawback, however, is that optimization depends too heavily on the assumption of well estimated models for the process mean and variance, and it is often the case that user specified parametric models are not flexible enough to adequately model the process mean and variance. Vining and Bohn (1998) and Anderson-Cook and Prewitt (2005) suggest the use of nonparametric smoothing when the user is unable to specify the explicit forms for the mean and/or variance functions. However, in small sample settings, which are customary for response surface experiments, the nonparametric approach often produces estimates that are highly variable. Consequently, we suggest a semiparametric approach, which combines the user's specified parametric model with a nonparametric fit, to provide better estimates of both the process mean and variance.

Using the Box and Draper (1987) printing ink data to compare the parametric, nonparametric, and semiparametric approaches, we find that the semiparametric approach performs best in terms of fit and optimization. The optimization based on the semiparametric approach recommends control factor settings which result in the estimated mean being closest to target as well as the smallest estimate of process variance. As previously mentioned, confirmatory experiments are necessary to prove which approach performs best for this example; however, the semiparametric approach produces better results than its parametric and nonparametric counterparts previously seen in the literature.

A simulation study was conducted to compare the three approaches more generally. Variance model misspecification was observed to have little impact on the quality of the estimated mean. If the user correctly specifies the mean and variance models, the parametric approach yields the best fit and optimization results with the semiparametric method a close second. The nonparametric method, on the other hand, is vastly inferior in terms of $SIMSEM$.

The nonparametric fit, while best for large degrees of mean misspecification, is only slightly better than the proposed semiparametric approach. When the mean is misspecified, the parametric method is clearly inferior in terms of fit and $ASEL$. For small to moderate mean misspecification, the semiparametric method is always superior in terms of fit and $ASEL$. Since, in practice, one never knows if the forms of the underlying models have been correctly specified, we advocate a method that performs consistently well over all degrees of potential misspecification. The semiparametric method is the only one which consistently performs well.

Chapter 6

Semiparametric Optimization Using the Genetic Algorithm

6.1 Introduction

Optimization has been a major component of Response Surface Methodology (RSM) since its inception. Once the response of interest has been estimated, optimization techniques are employed to identify settings of the explanatory variables which optimize the process. Historically, parametric regression has provided estimates of the response. However, semiparametric regression is a comparable, if not superior, alternative, especially in situations where the user's parametric model is likely to be misspecified. Unlike its parametric counterpart, the estimates from the semiparametric regression methods do no result in closed form expressions. Therefore, common optimization methods, such as analytic techniques which require continuous functions with derivatives for the estimated function, are no longer applicable. In their work involving nonparametric RSM, Vining and Bohn (1998) utilize a simplex search which does not require the calculation of derivatives. However, simplex

methods tend to be time consuming and often find local optima (Haupt and Haupt, 2004). Therefore, we propose the use of a more flexible optimization routine, the genetic algorithm (GA), when using semiparametric regression. The proposed GA has been successfully used in the chemical process and motor oil studies of Chapter 4 and the printing ink study of Chapter 5, as well as the simulation studies in these chapters. We will now compare the accuracy and efficiency of the simplex algorithm with our GA via simulation studies.

6.2 The Nelder-Mead Simplex Algorithm

Introduced by Nelder and Mead (1965), the simplex method has become a popular direct search optimization technique as it does not require derivatives, only function evaluations. A simplex is the most elementary geometrical figure that can be formed in N dimensions with $N + 1$ sides. For example, a triangle is a simplex in two dimensions and a tetrahedron is a simplex in three dimensions. The method works by creating an initial simplex with $k+1$ sides where k is the number of explanatory regressors. The method then applies different operations to rescale the simplex based on the local behavior of the surface until it is within an acceptable error.

The Nelder-Mead simplex algorithm is easily implemented with the R language via the "optim" procedure. The flow chart in Figure 6.1 illustrates the steps of the algorithm. First, the user must choose an initial starting value for each parameter of the problem. In our situation of optimizing a response(s) or some function of a response(s), these initial starting values constitute a single location in the design space; that is, factor settings for the design variables. Using this starting point as a vertex, the algorithm then creates a simplex. For details on the creation of the initial simplex, see Nash (1990). Next, the fitness at each vertex of the simplex is evaluated; i.e., the objective function to be optimized is evaluated. In the single response problem of Chapter 4, the objective function to be minimized is either

the estimated response, $\hat{y}$, or the estimated squared distance from target ($\widehat{SDT}$), defined as:

$$\widehat{SDT} = (\hat{y} - T)^2,$$

where T denotes the target value for the response. And in the dual model problem of Chapter 5, the objective function to be minimized is the estimated squared error loss ($\widehat{SEL}$), defined as:

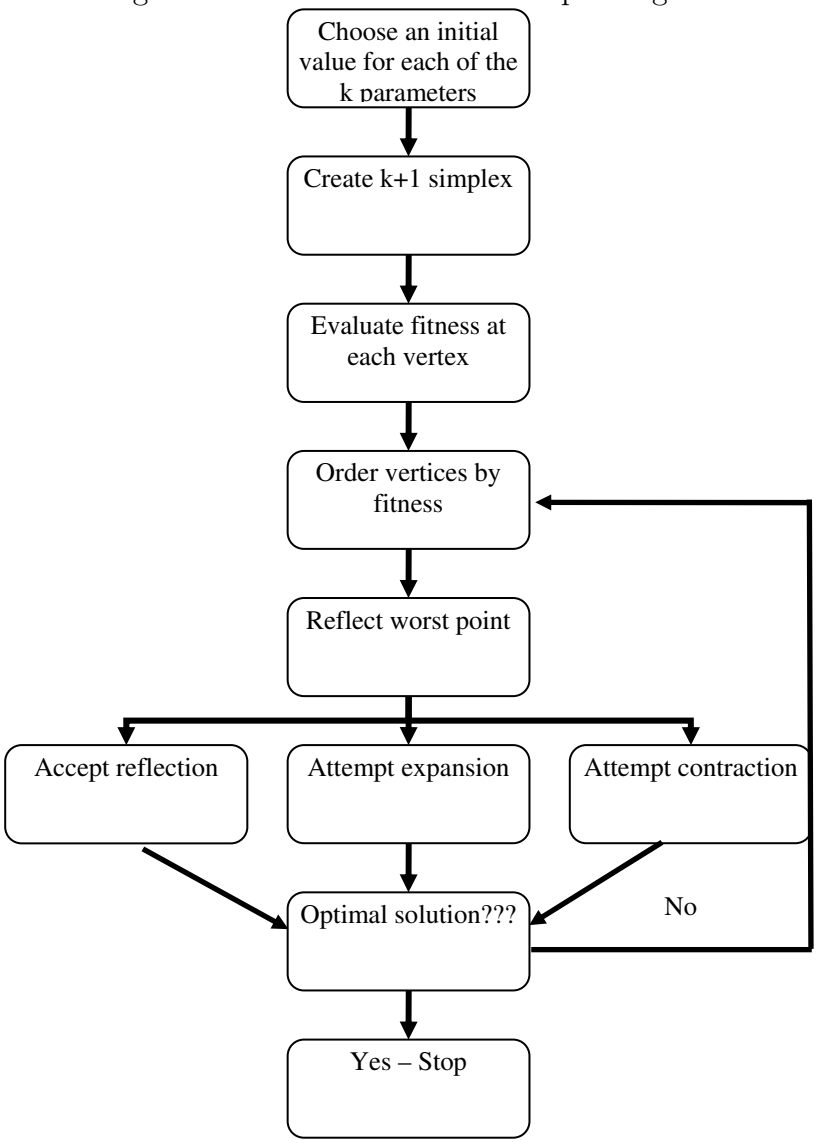
$$\widehat{SEL} = \{\widehat{E\,[y(\mathbf{x})]} - T\}^2 + \widehat{Var\,[y(\mathbf{x})]},$$

where $\widehat{E\,[y(\mathbf{x})]}$ denotes the estimated process mean, T denotes the target value for the process mean, and $\widehat{Var\,[y(\mathbf{x})]}$ denotes the estimated process variance.

After ordering the vertices by their fitness values, the worst point is reflected in the simplex through the face opposite it according to the reflection coefficient ($\alpha > 0$), whose value is commonly taken to be $\alpha = 1$. If the reflected point results in a fitness value that is better than the second worst point, then the reflected point replaces the worst point in the simplex. If the reflected point has a better fitness than the best point, then the reflection is expanded according to the expansion coefficient ($\gamma > 1$), whose value is commonly taken to be $\gamma = 2$. If the reflected point has the worst fitness, then the simplex contracts according to the contraction coefficient ($0 < \beta < 1$), whose value is commonly taken to be $\beta = 0.5$. A new simplex is then formed and the process iterates until convergence.

The Nelder-Mead simplex algorithm was originally developed for use with deterministic functions. Consequently, when used with stochastic optimization problems, the method often converges to local optimum (Barton and Ivey, 1996). Other major drawbacks of the method include sensitivity to starting values, computational inefficiency, and lack of theory (Wright, 1996). For more details on the Nelder-Mead simplex algorithm, see Vetterling et al. (1992).

Figure 6.1: The Nelder-Mead simplex algorithm.



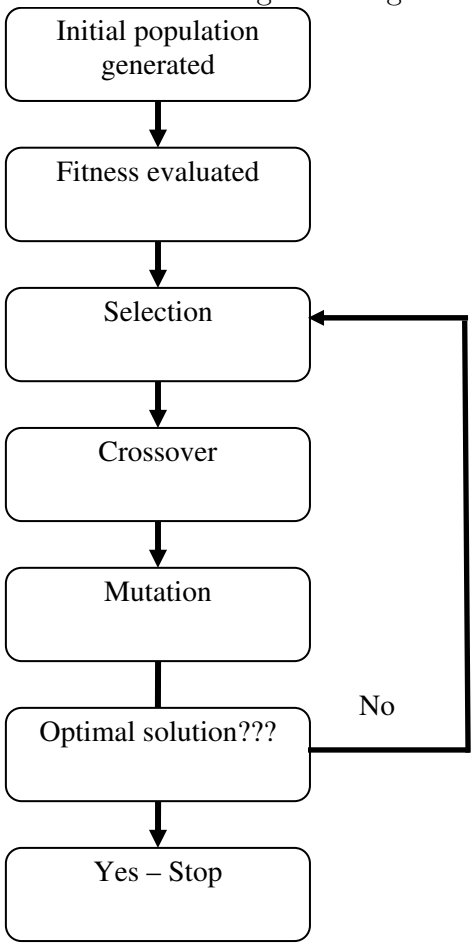
6.3 The Genetic Algorithm

To overcome the weaknesses of the Nelder-Mead simplex algorithm, we propose the use of the genetic algorithm (GA). Originally developed by Holland (1975) and popularized by his student Goldberg (1989), GA is an optimization technique based on the principles of genetics and natural selection. Genetic operators, such as selection, crossover, and mutation, are applied to regressor settings while searching for the optimum.

The GA has appeared in some of the latest RSM literature; however, it has mostly been used for design construction [see for example, Borkowski (2003), Heredia-Langner et al. (2003), and Heredia-Langner et al. (2004)]. More recently, Ortiz et al. (2004) use the GA for the multiple-response optimization problem involving a desirability function. Their work, based on parametric models, shows the GA consistently performs better than the traditional generalized reduced gradient (GRG) approach. Using the GA and RSM literature as a basis, we have constructed an algorithm to optimize semiparametric regression estimates which do not have closed form expressions. A flow chart of the basic GA appears in Figure 6.2. While our GA performs traditional operations on the regressor values, it also includes some adjustments that make it more applicable to the RSM setting.

The GA begins by generating an initial population of size m chromosomes, or potential solutions, completely at random. In the GA literature, a chromosome is a string of genes and each gene is an instance of a particular allele. In our situation of optimizing a response(s) or some function of a response(s), a chromosome is a $1 \times k$ vector of coded factors that represents a location in the design space and a gene is the coded setting of one factor. In our GA, these points are randomly generated from the uniform distribution. Next, the fitness of each chromosome is evaluated and the chromosomes are ranked by their fitness values. A selection procedure referred to as elitism is performed next. This step retains the two best chromosomes, and thereby guarantees the best solutions that have evolved thus far will remain unchanged and survive to the next generation. The random pairing

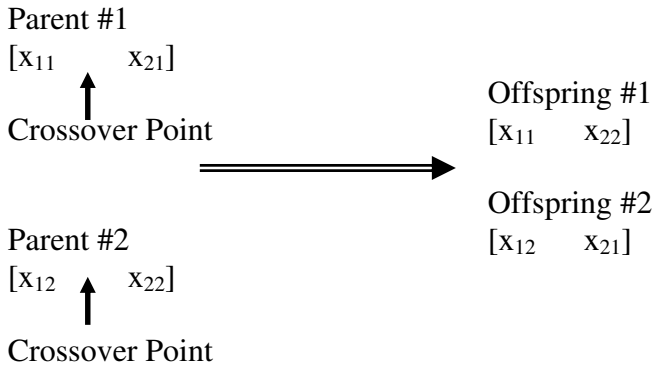
Figure 6.2: The basic genetic algorithm.



selection technique, which uses a uniform random number generator to select locations for reproduction, is then carried out on the remaining $m-2$ chromosomes, or parents.

Following selection, the crossover procedure is carried out with probability α_c to create new chromosomes, or offspring. A single crossover point is identified and parent one passes its genes to the left of the crossover point to offspring one and its genes to the right of the crossover point to offspring two. At the same time, parent two passes its genes to the left of the crossover point to offspring two and those to the right to offspring one. Figure 6.3 illustrates a single point crossover for our scenario with two regressors.

Figure 6.3: Single point crossover.



Following crossover, uniform mutation is carried out with probability α_m and the allele values of some genes randomly change according to a uniform random number generator. That is, α_m is the probability that the j^{th} gene of the i^{th} chromosome, x_{ij}, will be replaced with a random deviate from the (0,1) uniform distribution. In our GA, we have also included two additional mutation operators that change regressor values to the extremes of the design space, $x_{ij}=0$ and $x_{ij}=1$, with probability α_{zero} and α_{one}, respectively. These are common optimal locations in many RSM problems; thus, these operators ensure they are evaluated at some point in the algorithm. Finally, the next generation of potential solutions is completely formed and the algorithm iterates until convergence.

The GA offers many advantages over the simplex method and other optimization routines. To

begin, the GA does not require a differentiable function and it is relatively easy to implement. We have used the R language to code our algorithm and the code can be found in Appendix C. In addition, the GA performs well with complex functions and high dimensionality. Also, unlike the simplex method, the initial solutions are chosen completely at random by the algorithm and do not have to be "great" solutions; that is, they do not have to be close to the true optimum. In fact, even poor solutions may contain some useful information that can be extracted during crossover (Heredia-Langner et al., 2003). Furthermore, the mutation operator helps prevent convergence to local optimum by allowing the entire solution space to be explored, not just the local neighborhood of the initial population.

One drawback of the GA is that it does not guarantee the exact global optimum, only a near-global optimum. Nevertheless, this does not cause great worry as confirmatory experiments are suggested after any optimization is performed on a process. The main limitation of the GA is the number of tuning parameters that must be set (population size, m; crossover rate, α_c; mutation rates α_m, α_{zero}, and α_{one}). Very little theoretical guidance is available and quite often these parameter values are tailored to the specific problem at hand. However, some general guidelines have been discussed and parameter ranges have been suggested in the literature.

Population size, m, can affect both the accuracy and efficiency of the GA. If the population size is too small, the algorithm may not gain enough information about the entire solution space. Consequently, the GA may not converge to a near-global optimum. On the other hand, if the population size is too large, the algorithm requires more function evaluations. The GA is much more likely to converge to a near-global optimum; however, the rate of convergence is much slower. Mayer, Belward, and Burrage (2001) recommend using a population size that is larger than the dimensionality of the problem being studied and prefer a population size equal to twice the dimensionality. For our problem with two regressors, this recommendation would be m equal to 4. Other recommended population sizes range from 20 to 100 with 50 being the most commonly used size [see for example, De Jong (1975) and

Heredia-Langner et al. (2004)].

Crossover rate, α_c, and mutation rates, α_m, α_{zero}, and α_{one}, can also affect the quality of the search. A low crossover rate can place too much emphasis on the current population and not introduce new information quick enough. Recommended crossover rates range from 0.5 to 0.95 [see for example, De Jong (1975), Grefenstette (1986), and Ortiz et al. (2004)]. A high mutation rate can ignore information from each generation and reduce the GA to a purely random search. Recommended mutation rates range from 0.001 to 0.4 [see for example, De Jong (1975), Borkowski (2003), Bäck (1996), and Heredia-Langner et al. (2004)].

6.4 Simulations

In Chapters 4 and 5, simulations studies found semiparametric regression to perform consistently well over all degrees of potential model misspecification. Optimization will now be carried out on these semiparametric estimates via the Nelder-Mead simplex algorithm and our genetic algorithm. The Nelder-Mead simplex algorithm will use the traditional parameter values, $\alpha = 1$, $\gamma = 2$, and $\beta = 0.5$. Since it is argued the Nelder-Mead simplex algorithm is sensitive to starting values, we will use nine different starting values to investigate this claim. These starting values appear in Table 6.1. Two population sizes will studied in the GA, $m = 4$ which is twice the dimension of the problem and $m = 50$. Other parameter values for the GA will be set at $\alpha_c = 0.90$, $\alpha_m = \alpha_{zero} = \alpha_{one} = 0.20$. The two optimization algorithms will be compared for accuracy and efficiency. The R language is used for computations and simulations.

Table 6.1: Nine starting values for the Nelder-Mead simplex algorithm.

x_1	x_2
0.25	0.25
0.25	0.50
0.25	0.75
0.50	0.25
0.50	0.50
0.50	0.75
0.75	0.25
0.75	0.50
0.75	0.75

6.4.1 Single Response

As in Chapter 4, Monte Carlo simulations will be used to generate 500 data sets, each of which is based on the following underlying model:

$$\begin{aligned} y &= 20 - 10x_1 - 25x_2 - 15x_1x_2 + 20x_1^2 + 50x_2^2 \\ &\quad +\gamma\left[2\sin\left(4\pi x_1\right) + 2\cos\left(4\pi x_2\right) + 2\sin\left(4\pi x_1x_2\right)\right] + \varepsilon, \end{aligned} \tag{6.1}$$

where $\varepsilon \sim \mathrm{N}(0,1)$ and γ represents the model misspecification parameter. The data will be generated as if a rotatable CCD with five center runs was run for a total of 13 experimental runs. Therefore, when coded to be between 0 and 1, factors x_1 and x_2 have five levels with values taken to be at 0.000, 0.146, 0.500, 0.854, and 1.000 for each factor. As the value of γ increases, the amount of misspecification increases in the model. Five degrees of model misspecification will be studied ($\gamma = 0.00$, 0.25, 0.50, 0.75, and 1.00). The goal is to achieve a target value of 15 for the response. Therefore, the objective function to be minimized is the estimated squared distance from target ($\widehat{SDT}$), defined as:

$$\widehat{SDT} = (\hat{y} - 15)^2, \tag{6.2}$$

where $\hat{y}$ denotes the semiparametric estimated response.

The efficiency of each optimization algorithm will be measured by the average number of evaluations of function (6.2), defined as:

$$AFE = \frac{\sum FE}{500}, \tag{6.3}$$

where FE denotes the number of function evaluations required for a single data set. Accuracy comparisons will be based on the average Euclidean distance from the optimum defined as:

$$AED = \frac{\sum ED}{500}, \tag{6.4}$$

where

$$ED = \sqrt{(x_1^* - x_{opt1}^*)^2 + (x_2^* - x_{opt2}^*)^2},$$

where ED denotes the Euclidean distance between each algorithm's chosen optimal location, $\mathbf{x}^* = [x_1^*, x_2^*]$, and the true optimum, $\mathbf{x}_{opt}^* = [x_{opt1}^*, x_{opt2}^*]$. The accuracy of the optimization algorithms will also be compared via the average true squared distance from target ($ASDT$),

$$ASDT = \frac{\sum SDT}{500},$$

where

$$SDT = (\mu(\mathbf{x}^*) - 15)^2,$$

where $\mu(\mathbf{x}^*)$ denotes the true mean response when the optimal location chosen by the algorithm, $\mathbf{x}^*$, is substituted into the true underlying function given by equation (6.1). Even though the true optimal location may not be found by an algorithm, a near-optimal location may be found. Thus $ASDT$ measures the performance of the locations chosen by each algorithm to see if they are in fact near-optimal. The results appear in Tables 6.2, 6.3, and 6.4.

Tables 6.2 and 6.3 provide the simulation results for no model misspecification ($\gamma = 0.00$) and Table 6.4 provides the simulation results for model misspecification ($\gamma \geq 0.25$). When the model is correctly specified ($\gamma = 0.00$), there are nine true optimal locations that achieve

a target value of 15 for the response, and when the model is misspecified ($\gamma \geq 0.25$), there is only one true optimal location (0.500, 0.250). Note that when the Nelder-Mead simplex algorithm begins close to the true global optimum, it performs very well. In fact, it often results in the smallest AFE, AED, and $ASDT$ values. However, when the Nelder-Mead simplex algorithm begins far from the true global optimum, its AED and $ASDT$ values are extremely large, some $ASDT$ values are even in the hundreds of thousands. This confirms the fact that the simplex method is highly sensitive to starting values. Thus, if the user has no a priori knowledge of the exact location of the global optimum, the Nelder-Mead simplex is not recommended for use.

Comparing the performance of the Nelder-Mead on average to the GA with population size $m = 4$ and the GA with $m = 50$, we see that the Nelder-Mead algorithm is more efficient as it consistently results in smaller AFE values; yet the GA with $m = 4$ is highly competitive. The GA is also very accurate. In fact, the GA results in much smaller AED and $ASDT$ values than the Nelder-Mead simplex on average. Note that as γ increases, the AFE values stay fairly consistent for both optimization algorithms, while the AED values slightly increase and the $ASDT$ values increase for both the GA and the Nelder-Mead. Also note that the increase in the $ASDT$ values is significantly greater for the Nelder-Mead simplex. Looking closer at the results for the two GA's, the GA with $m = 4$ converges much quicker than the GA with $m = 50$ as expected; however, the AED and $ASDT$ values are highly competitive for the two GA's. The benefit of the large population GA does not seem to outweigh the cost of the additional function evaluations in this scenario. Thus, the GA with $m = 4$ appears to be the best optimization procedure overall as it is highly competitive with the Nelder-Mead in terms of efficiency and far superior in terms of accuracy, on average, when compared to the nine different starting values for the Nelder-Mead.

Table 6.2: AFE and $ASDT$ values for single response when the model is correctly specified ($\gamma = 0.00$) based on 500 Monte Carlo runs. Best values in bold.

γ	Optimization Algortihm	AFE	$ASDT$
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	42.7600	0.1961
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	44.0800	45.3631
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	45.1200	271.6051
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	**41.6800**	0.1766
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	45.4400	2493.1325
0.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	45.4400	3825.4262
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	42.3600	9.2982
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	44.9600	8634.5763
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	48.5600	6,589,491.4857
	GA $m = 4$	90.4444	0.8488
	GA $m = 50$	1051.3889	**0.0868**

Table 6.3: *AED* values for single response when the model is correctly specified ($\gamma = 0.00$) based on 500 Monte Carlo runs. Best values in bold.

		True Optimum	
Optimization Algortihm	(0.500, 0.250)	(0.300, 0.190)	(0.250, 0.200)
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	0.2408	0.1790	**0.1743**
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	0.3636	0.4964	0.5233
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	0.4223	0.5950	0.6442
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	**0.1466**	0.3071	0.3409
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	0.7488	0.8931	0.9154
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	0.8936	1.0636	1.1133
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	0.3214	0.4852	0.5172
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	1.1174	1.1966	1.1996
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	3.1447	3.2761	3.3174
GA $m = 4$	0.2339	0.2379	0.2455
GA $m = 50$	0.1967	**0.1749**	0.1879

		True Optimum	
Optimization Algortihm	(0.190, 0.245)	(0.190, 0.312)	(0.544, 0.312)
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	**0.1614**	**0.1193**	0.2613
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	0.5434	0.5144	0.3156
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	0.7091	0.7249	0.4128
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	0.3733	0.3541	**0.0831**
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	0.9241	0.8837	0.6833
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	1.1806	1.1989	0.8775
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	0.5437	0.5164	0.2584
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	1.1788	1.1247	1.0622
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	3.3766	3.3968	3.1343
GA $m = 4$	0.2500	0.2460	0.2426
GA $m = 50$	0.2031	0.1973	0.2201

Table 6.3 Continued:

Optimization Algortihm	True Optimum (0.545, 0.316)	(0.250, 0.375)	(0.400, 420)
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	0.2616	**0.0522**	**0.1302**
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	0.3131	0.4431	0.3155
Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	0.4142	0.6912	0.5853
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	**0.0796**	0.2877	0.1537
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	0.6799	0.8065	0.6828
Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	0.8783	1.1647	1.0500
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	0.2553	0.4409	0.2908
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	1.0588	1.0569	0.9874
Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	3.1350	3.3719	3.2774
GA $m = 4$	0.2432	0.2359	0.2309
GA $m = 50$	0.2211	0.1866	0.2003

Table 6.4: Comparison of optimization algorithms for single response when the model is misspecified ($\gamma \geq 0.25$) based on 500 Monte Carlo runs. Best values in bold.

γ	Optimization Algortihm	AFE	AED	$ASDT$
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	44.0400	0.2366	0.8589
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	44.5600	0.3972	125.1006
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	44.5200	0.4024	207.2351
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	**41.6400**	0.1803	1.6461
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	45.5200	0.9113	11,992.4241
0.25	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	45.0400	0.8133	1,947.0327
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	42.3200	0.3453	11.1358
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	45.7200	1.3646	76,413.7244
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	48.0000	2.6172	4,362,298.7390
	GA $m = 4$	94.2857	0.2510	0.7242
	GA $m = 50$	1121.4286	**0.1777**	**0.2547**
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	45.1200	0.2409	3.6701
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	44.4800	0.4042	100.0627
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	44.7200	0.3712	173.0582
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	**42.5200**	0.2237	7.9055
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	46.4400	1.1083	422,642.9830
0.50	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	45.3600	0.7548	1,469.4944
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	42.5600	0.3815	18.5599
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	46.1200	2.2500	55,226,807.3612
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	47.2000	2.0216	518,767.9418
	GA $m = 4$	92.4000	0.2507	2.4942
	GA $m = 50$	1130.0000	**0.2045**	**0.5965**

Table 6.4 Continued:

γ	Optimization Algortihm	AFE	AED	$ASDT$
0.75	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	45.1200	0.2532	8.7429
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	45.4000	0.4147	102.6201
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	44.2400	0.3495	125.7016
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	43.4000	0.2800	32.4195
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	46.6000	1.0542	162,649.1375
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	45.2800	0.6980	893.4541
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	**42.8400**	0.4267	38.0380
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	47.0400	2.1158	726,137.9969
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	46.3600	1.7072	85,977.9253
	GA $m = 4$	101.8182	0.2255	2.3802
	GA $m = 50$	840.9091	**0.2069**	**1.3559**
1.00	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	46.3200	0.3089	152.1378
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	45.7600	0.4436	324.4695
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	44.5200	0.3380	108.8807
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	**43.7600**	0.3545	164.4654
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	47.0400	1.0237	43,717.0346
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	45.4800	0.6522	645.8617
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	43.8000	0.4903	85.4979
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	47.4800	2.2704	728,407.7623
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	46.4000	1.5407	43,438.9238
	GA $m = 4$	83.0000	0.3247	2.0806
	GA $m = 50$	1112.5000	**0.1930**	**1.6798**

6.4.2 Dual Model

As in Chapter 5, Monte Carlo simulations will be used to generate 500 data sets, each of which is based on the following underlying dual model:

$$\begin{aligned} y_i &= h\left(\tilde{\mathbf{x}}_i;\boldsymbol{\beta}\right) + m\left(\tilde{\mathbf{x}}_i\right) + g^{1/2}\left(\tilde{\mathbf{x}}_i^*;\boldsymbol{\gamma}\right)\varepsilon_i \\ &= 20 - 10x_{1i} - 25x_{2i} - 15x_{1i}x_{2i} + 20x_{1i}^2 + 50x_{2i}^2 \\ &\quad +\gamma_\mu\left[10\sin\left(4\pi x_{1i}\right) + 10\cos\left(4\pi x_{2i}\right) + 10\sin\left(4\pi x_{1i}x_{2i}\right)\right] \\ &\quad +g^{1/2}\left(\tilde{\mathbf{x}}_i^*\right)\varepsilon_i \end{aligned} \tag{6.5}$$

$$\begin{aligned} \ln\left(\sigma_i^2\right) &= g^*\left(\tilde{\mathbf{x}}_i^*;\boldsymbol{\gamma}\right) + l\left(\tilde{\mathbf{x}}_i^*\right) \\ &= 1.5 - x_{1i} + 1.5x_{2i} + \gamma_\sigma\left[-4x_{1i}x_{2i} + 2x_{1i}^2 + x_{2i}^2\right], \end{aligned} \tag{6.6}$$

where $\varepsilon_i \sim \mathrm{N}(0,1)$, γ_μ represents the means model misspecification parameter, and γ_σ represents the variance model misspecification parameter. The data will be generated as if a 4^2 complete factorial experiment was run with three replicates at each design point for a total of 16 design points and 48 experimental runs. Therefore, when coded to be between 0 and 1, factors x_1 and x_2 have four levels with values taken to be 0, 1/3, 2/3, and 1 for each factor. As the values of γ_μ and γ_σ increase, the amount of misspecification increases in the means and variance model, respectively. Five degrees of means model misspecification will be studied ($\gamma_\mu = 0.00$ (means model correct), 0.25, 0.50, 0.75, and 1.00), and five degrees of variance model misspecification will be studied ($\gamma_\sigma = 0.00$ (variance model correct), 0.25, 0.50, 0.75, and 1.00). The goal is to achieve a target value of 15 for the mean response while minimizing the process variance. Therefore, the objective function to be minimized is the estimated squared error loss, ($\widehat{SEL}$), defined as:

$$\widehat{SEL} = \{\widehat{E\left[y(\mathbf{x})\right]} - 15\}^2 + \widehat{Var\left[y(\mathbf{x})\right]},$$

where $\widehat{E\left[y(\mathbf{x})\right]}$ denotes the semiparametric estimated process mean and $\widehat{Var\left[y(\mathbf{x})\right]}$ denotes the semiparametric estimated process variance.

Again, the efficiency of each optimization algorithm will be measured by the average number of function evaluations, AFE, defined in equation (6.3), and the accuracy will be based on the average Euclidean distance from the optimum, AED, defined in equation (6.4). The accuracy of the optimization algorithms will also be compared via the average true squared error loss, ($ASEL$), defined as:

$$ASEL = \frac{\sum SEL}{500},$$

where

$$SEL = \{E\left[y(\mathbf{x}^*)\right] - 15\}^2 + Var\left[y(\mathbf{x}^*)\right].$$

Regarding notation, $E\left[y(\mathbf{x}^*)\right]$ denotes the true process mean and $Var\left[y(\mathbf{x}^*)\right]$ denotes the true process variance when the optimal location chosen by the algorithm, $\mathbf{x}^*$, is substituted into the true underlying functions given by equations (6.5) and (6.6). The results appear in Tables 6.5 through 6.9.

The simulation results comparing the Nelder-Mead simplex algorithm, the GA with $m = 4$, and the GA with $m = 50$ with no variance model misspecification ($\gamma_\sigma = 0.00$) appear in Table 6.5. The simulation results for the varying degrees of variance model misspecification ($\gamma_\sigma = 0.25$, 0.50, 0.75, and 1.00) appear in Tables 6.6 through 6.9, respectively. Again, note that the Nelder-Mead simplex algorithm performs well when it begins close to the true global optimum. However, when the Nelder-Mead begins far from the true global optimum its AFE, AED, and $ASEL$ values are extremely large, some $ASEL$ values are even in the hundreds of thousands. Therefore, the Nelder-Mead is not recommended for use unless the use has a priori knowledge of the exact location of the true global optimum.

As the amount of means model misspecification, γ_μ, increases, the AFE values stay fairly consistent for both the Nelder-Mead simplex and the GA, while the AED and $ASEL$ values both increase. The amount of means model misspecification appears to have the largest impact on the $ASEL$ values. Note that the increase in the $ASEL$ values is much greater for the Nelder-Mead simplex.

Unlike the previous single response simulation study, the Nelder-Mead simplex algorithm is not always the most efficient procedure. In fact, the GA with $m = 4$ nearly always has the lowest AFE values. The lack of efficiency for the simplex algorithm is most likely due to the increased complexity of the dual model. Furthermore, the GA is very accurate as it results in much smaller AED and $ASEL$ values than the Nelder-Mead on average. Moreover, the performance of the Nelder-Mead becomes vastly inferior as the amount of means model misspecification (γ_μ) increases. Looking closer at the results for the two GA's, the GA with $m = 4$ converges much quicker than the GA with $m = 50$ as expected. While the AED values are highly competitive for the two GA's, the $ASEL$ values for the large population GA are consistently smaller. In some cases, the $ASEL$ values for the small population GA are nearly twice that of the large population GA. The benefit of the large population GA seems to outweigh the cost of the additional function evaluations in some scenarios. Yet, if efficiency is a major concern, the GA with $m = 4$ is the best optimization procedure overall as it is far superior than the Nelder-Mead on average in terms of efficiency and accuracy, when compared to the nine different starting values for the Nelder-Mead.

Table 6.5: Comparison of optimization algorithms for dual model for no varinace model misspecification ($\gamma_\sigma = 0.00$) based on 500 Monte Carlo runs. Best values in bold.

γ_μ	Optimization Algorithm	*AFE*	*AED*	*ASEL*
	True Optimum = (0.505, 0.223)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	**56.2000**	0.1359	4.7471
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	148.0000	0.3687	90.8464
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	425.5600	1.0144	1,721.7762
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	57.7200	0.2162	5.4384
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	71.0800	0.1625	15.2819
0.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	364.3200	0.7952	1,095.7835
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	62.5200	0.1925	5.9059
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	124.7600	0.2210	31.9560
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	512.0800	0.9263	2,227.1922
	GA $m = 4$	73.0909	0.2526	6.7447
	GA $m = 50$	1131.8200	**0.0724**	**4.0493**
	True Optimum = (0.884, 0.287)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	61.0400	0.5217	9.4052
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	101.6800	**0.2535**	20.3564
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	409.6000	0.9814	2,343.1986
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	61.9200	0.5313	12.4133
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	61.0400	0.4482	**6.3816**
0.25	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	326.1200	0.5294	698.8757
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	66.0400	0.3555	24.0836
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	86.7600	0.3876	16.7888
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	493.6800	0.8730	1,190.4185
	GA $m = 4$	**60.9091**	0.3813	8.1291
	GA $m = 50$	1188.6350	0.2964	7.9375

Table 6.5 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.938, 0.237)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	83.5200	0.8220	23,150.8627
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	84.4800	0.5591	12,645.0664
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	361.8000	2.1576	52,021.9649
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	85.9200	0.5308	47.9266
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	64.8800	0.5249	13.0611
0.50	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	331.9200	0.6043	1,316.1190
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	76.8000	0.3856	35.2200
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	68.5200	0.4544	**9.4982**
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	484.0800	2.2518	2.7111E+10
	GA $m = 4$	**63.1304**	0.4908	15.0234
	GA $m = 50$	778.2600	**0.1852**	10.1547
	True Optimum = (0.956, 0.221)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	147.4000	6.3545	3.1611E+19
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	106.5600	3.0226	2.3308E+41
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	383.1600	4.2140	8.6078E+06
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	100.1600	0.5285	126.5783
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	75.2800	0.5551	30.2019
0.75	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	326.0800	1.2592	24,256.4698
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	112.1600	0.8504	23,135.2435
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	73.6400	0.5242	21.0738
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	329.5600	3.8147	2.2879E+45
	GA $m = 4$	**71.4667**	0.4531	26.4958
	GA $m = 50$	933.3350	**0.1142**	**20.3383**

Table 6.5 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.964, 0.212)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	207.9200	10.6334	1.1503E+10
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	107.9200	4.7380	1.9217E+69
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	559.4800	11.9056	1.3501E+11
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	120.6800	0.5966	1,234.5762
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	84.4400	0.5719	49.5248
1.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	433.0400	2.3550	2.1775E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	246.3200	2.2197	71,511.3000
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	105.9600	0.7442	4,888.3317
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	422.9200	3.7869	9.6382E+16
	GA $m = 4$	**76.4762**	0.3731	28.1748
	GA $m = 50$	992.8550	**0.0863**	**26.8886**

Table 6.6: Comparison of optimization algorithms for dual model when $\gamma_\sigma = 0.25$ based on 500 Monte Carlo runs. Best values in bold.

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.490, 0.225)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	**56.3200**	0.1333	**3.5465**
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	148.0000	0.3686	86.3921
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	420.0000	1.0045	1,626.7054
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	56.7200	0.2114	4.5244
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	72.7200	0.1610	12.9778
0.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	378.2400	0.7933	1,075.4134
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	61.3600	0.2059	4.5262
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	117.3600	0.2113	27.4904
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	475.3200	0.8855	2,076.4187
	GA $m = 4$	62.0667	0.2304	6.0218
	GA $m = 50$	1267.5000	**0.0636**	4.1309
	True Optimum = (0.878, 0.285)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	61.3200	0.5167	8.5150
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	102.5600	**0.2493**	19.4113
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	430.4000	0.9606	2,259.6331
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	63.0800	0.5277	11.4572
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	**59.4800**	0.4380	**5.3716**
0.25	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	326.8800	0.5300	700.8820
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	62.0000	0.3579	22.7424
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	85.0800	0.3675	15.4697
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	508.4000	0.9569	1,350.3217
	GA $m = 4$	65.0588	0.4996	9.4170
	GA $m = 50$	1017.6450	0.3806	8.5887

Table 6.6 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.909, 0.221)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	81.8400	0.7943	27,823.9960
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	88.3600	0.5070	7,888.9960
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	383.4000	2.0409	29,052.5865
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	86.0400	0.5202	46.7485
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	63.0000	0.4947	13.3603
0.50	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	308.8000	0.6131	1,218.2369
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	69.7200	0.3843	35.7367
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	65.5600	0.4145	**10.6341**
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	525.7200	2.5057	1.4667E+20
	GA $m = 4$	**61.6667**	0.3989	17.2084
	GA $m = 50$	820.8350	**0.1978**	11.3221
	True Optimum = (0.914, 0.198)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	144.1600	6.8946	2.6196E+51
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	108.8000	4.5742	1.1572E+160
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	375.8000	3.4144	2.9920E+06
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	130.0000	5.7361	7.3466E+269
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	67.5200	0.5190	**19.0184**
0.75	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	344.2400	1.1570	12,556.0723
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	107.7200	0.8536	11,920.0963
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	73.5200	0.4767	22.1353
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	343.8000	3.6857	2.4447E+71
	GA $m = 4$	**65.9231**	0.4149	29.1455
	GA $m = 50$	926.9250	**0.1372**	22.6075

Table 6.6 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.917, 0.186)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	203.2400	9.5924	7.1667E+122
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	106.8800	9.4053	1.7505E+16
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	482.8800	4.6048	3.2643E+19
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	132.1600	1.4223	3.0571E+48
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	77.0000	0.5301	42.5564
1.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	446.7600	2.1493	1.7352E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	237.9600	2.2815	1.0060E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	108.4000	0.6915	4,566.0903
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	396.1600	4.5670	6.6243E+31
	GA $m = 4$	**55.0000**	0.3571	31.0214
	GA $m = 50$	1050.0000	**0.0905**	**23.4247**

Table 6.7: Comparison of optimization algorithms for dual model when $\gamma_\sigma = 0.50$ based on 500 Monte Carlo runs. Best values in bold.

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.490, 0.225)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	**55.8000**	0.1304	**2.7003**
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	150.1600	0.3639	80.3608
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	417.3600	0.9883	1,537.0229
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	58.2400	0.2064	3.8056
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	68.9200	0.1630	11.2863
0.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	355.6000	0.7862	963.9178
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	63.1200	0.2114	3.5521
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	111.8800	0.2018	22.9235
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	507.1600	0.9244	2,117.8514
	GA $m = 4$	76.8932	0.1936	5.9365
	GA $m = 50$	1054.1250	**0.0720**	4.2284
	True Optimum = (0.821, 0.284)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	61.9200	0.4597	7.8525
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	105.3200	**0.2187**	18.7614
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	389.2800	0.9899	2,042.1191
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	62.3200	0.4737	10.8956
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	**58.6000**	0.3774	**4.4979**
0.25	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	340.8800	0.5699	718.0905
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	67.9200	0.3216	22.1320
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	87.5200	0.3041	14.1871
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	503.44	0.9880	1,815.6442
	GA $m = 4$	78.0000	0.3588	8.2347
	GA $m = 50$	1004.1650	0.2720	7.7797

Table 6.7 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.792, 0.233)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	84.6000	0.7054	44,008.9770
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	88.8400	0.4332	7,351.0561
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	364.9200	1.9724	20,438.4541
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	83.3200	0.4284	71.5618
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	62.8800	0.3754	14.2225
0.50	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	328.2000	0.7225	1,220.0193
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	76.1600	0.3302	36.2987
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	68.9600	0.2891	8.2378
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	483.8800	2.6193	3.5376E+23
	GA $m = 4$	**54.5455**	0.3652	15.4848
	GA $m = 50$	858.1800	**0.1697**	**7.9387**
	True Optimum = (0.400, 0.105)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	156.2000	6.8111	1.5919E+88
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	98.7600	1.3805	3.5001E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	363.3600	3.4910	2.9344E+13
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	122.9600	1.1778	5.0068E+55
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	63.8800	0.2484	18.1686
0.75	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	300.1600	1.4876	7,982.9325
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	102.4800	0.9138	3,661.5886
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	70.9200	0.2117	13.9100
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	391.8000	5.9579	3.2885E+97
	GA $m = 4$	**56.6667**	0.2789	22.4549
	GA $m = 50$	900.0000	**0.2098**	**11.6994**

Table 6.7 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.395, 0.097)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	189.3600	8.7117	2.2848E+92
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	105.7600	1.9598	5.9271E+20
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	507.6400	7.0250	6.1439E+175
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	128.3200	0.9196	3.4009E+32
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	79.2400	0.2862	30.6898
1.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	403.0400	2.1983	36,320.4291
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	232.7200	3.1198	1.7819E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	100.6400	0.4393	4,527.8376
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	456.8400	4.7928	6.5540E+30
	GA $m = 4$	**74.6087**	0.3219	39.6292
	GA $m = 50$	926.0850	**0.2417**	**17.2676**

Table 6.8: Comparison of optimization algorithms for dual model when $\gamma_\sigma = 0.75$ based on 500 Monte Carlo runs. Best values in bold.

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.479, 0.225)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	**56.2400**	0.1254	**2.0849**
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	144.8000	0.3588	73.9445
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	403.3600	0.9728	1,352.1036
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	57.9200	0.1992	3.2214
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	67.2800	0.1641	9.7567
0.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	346.3200	0.7717	848.7367
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	59.8800	0.2241	3.2082
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	102.9600	0.1952	18.8210
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	486.2800	0.8946	2,008.2440
	GA $m = 4$	68.4918	0.2284	6.4587
	GA $m = 50$	888.5250	**0.0740**	4.3042
	True Optimum = (0.397, 0.128)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	62.3600	0.2791	7.5456
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	100.9600	0.3607	18.1363
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	414.2400	1.3022	1,963.7632
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	69.7200	0.3109	12.7405
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	**59.5600**	**0.1545**	**3.8149**
0.25	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	326.1600	0.8821	685.1765
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	65.3600	0.4224	21.4606
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	85.4400	0.1828	13.4515
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	533.2800	0.8971	1,910.4349
	GA $m = 4$	68.3636	0.2989	9.0154
	GA $m = 50$	890.9100	0.2272	5.9049

Table 6.8 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.388, 0.108)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	87.6000	0.5959	23,001.8226
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	94.2800	0.5082	3,272.9365
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	373.2400	2.1695	14,717.8904
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	96.3600	0.3978	45.2969
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	**60.8800**	0.2158	13.9302
0.50	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	334.5600	1.0919	1,299.1021
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	76.3600	0.5243	36.9382
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	63.6400	**0.2138**	8.4471
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	461.6400	2.0212	2.1698E+27
	GA $m = 4$	68.6957	0.3341	13.4713
	GA $m = 50$	832.6100	0.2365	**8.3529**
	True Optimum = (0.383, 0.098)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	154.9200	8.0497	3.6339E+64
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	99.6800	1.0570	1.2617E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	359.5200	3.7792	1.1394E+22
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	153.2000	2.7820	1.1216E+116
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	64.8800	0.2596	20.3277
0.75	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	300.4000	1.5122	6,916.5833
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	103.8800	0.9357	5,373.9163
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	72.6000	**0.2305**	**10.9521**
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	368.9600	3.7529	5.9079E+38
	GA $m = 4$	**72.3000**	0.3466	26.5497
	GA $m = 50$	862.5000	0.3314	13.7411

Table 6.8 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.979, 0.811)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	183.5200	7.7930	5.9201E+82
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	105.8000	2.0861	6.6235E+29
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	425.1200	3.2324	2.5467E+36
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	151.2800	2.3223	4.4947E+124
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	75.3600	0.7043	28.8679
1.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	415.0400	1.8333	60,741.8003
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	227.6800	2.7749	3.1603E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	110.6800	0.9385	9,461.7488
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	439.7200	6.0170	9.2564E+51
	GA $m = 4$	**69.6923**	0.6609	28.0668
	GA $m = 50$	973.0750	**0.5223**	**10.2459**

Table 6.9: Comparison of optimization algorithms for dual model when $\gamma_\sigma = 1.00$ based on 500 Monte Carlo runs. Best values in bold.

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.553, 0.344)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	**56.2000**	0.1615	**1.6299**
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	140.6800	0.3821	71.5572
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	417.6000	0.9538	1,384.8136
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	58.1600	0.2561	2.7576
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	65.3200	0.2350	9.0723
0.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	332.2000	0.7829	814.7211
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	60.4800	0.1464	3.3880
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	104.6000	0.2597	15.5194
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	461.6800	0.9766	1,819.4963
	GA $m = 4$	64.8750	0.2472	6.4726
	GA $m = 50$	865.6250	**0.1347**	4.3498
	True Optimum = (0.365, 0.121)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	61.3600	0.2833	7.0710
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	102.6400	0.3838	20.0446
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	429.6400	1.3243	2,029.7226
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	66.5200	0.3188	13.6788
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	58.6000	**0.1755**	**3.3883**
0.25	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	338.1200	0.9076	716.2810
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	68.0800	0.4490	20.6543
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	86.3600	0.2191	12.7467
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	481.0400	0.8594	1,389.9019
	GA $m = 4$	**58.3077**	0.3460	9.9114
	GA $m = 50$	865.3850	0.2520	8.4816

Table 6.9 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.821, 0.482)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	88.8400	0.6020	20,843.1056
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	84.4000	0.4481	3,294.7549
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	362.9600	1.7184	12,049.5568
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	93.7600	0.4433	17,336.6319
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	61.0400	0.4307	13.0143
0.50	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	327.9200	0.7858	1,215.0797
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	79.3600	**0.1778**	37.5687
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	64.2800	0.3659	**9.1009**
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	445.0400	1.8587	3.9918E+27
	GA $m = 4$	**58.8571**	0.4488	16.3132
	GA $m = 50$	764.2850	0.2298	11.1490
	True Optimum = (0.824, 0.526)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	152.5600	8.0317	1.0575E+112
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	99.3600	8.8108	1.5545E+11
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	372.9200	3.2825	4.5834E+38
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	165.2000	3.7300	1.9810E+77
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	65.0800	0.4416	25.6239
0.75	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	313.9600	1.1413	7,592.9194
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	95.2400	0.3551	6,919.9573
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	72.0000	0.4112	**9.0655**
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	367.8400	3.8625	4.8005E+75
	GA $m = 4$	**56.7273**	0.2210	17.1673
	GA $m = 50$	704.5450	**0.2201**	12.4291

Table 6.9 Continued:

γ_μ	Optimization Algorithm	AFE	AED	$ASEL$
	True Optimum = (0.959, 0.834)			
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.25]$	185.6400	10.1658	2.9412E+108
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.50]$	102.6400	2.0228	4.6382E+36
	Nelder-Mead $\mathbf{x}_{start} = [0.25, 0.75]$	518.4400	8.5573	2.6087E+277
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.25]$	159.4400	1.8336	1.1334E+79
	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.50]$	73.6000	0.7006	28.5986
1.00	Nelder-Mead $\mathbf{x}_{start} = [0.50, 0.75]$	394.6800	1.7626	42,353.9258
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.25]$	230.1600	3.3224	8.7119E+05
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.50]$	88.6000	0.8295	5,304.7217
	Nelder-Mead $\mathbf{x}_{start} = [0.75, 0.75]$	450.2000	4.5119	2.5789E+65
	GA $m = 4$	**58.0000**	0.5177	24.2765
	GA $m = 50$	775.0000	**0.4947**	**15.0197**

6.5 Conclusions

In the RSM setting, semiparametric regression methods have been shown to be comparable, if not superior, to parametric regression methods, especially in situations where the user's parametric model is likely to be misspecified. One drawback, however, is that the semiparametric estimates do not result in closed form expressions. Consequently, traditional optimization methods based on derivatives are no longer applicable. Vining and Bohn (1998) suggest the use of the Nelder-Mead simplex algorithm for optimization. However, simplex methods have been shown to have several weaknesses, including convergence to local optimum, sensitivity to starting values, and computational inefficiency. Therefore, we suggest the use of the genetic algorithm (GA), a flexible optimization routine based on the principles of genetics and natural selection.

Our GA has been successfully implemented on data sets from the RSM literature, as well as simulated data sets, for the single response problem and the dual model problem in Chapters 4 and 5, respectively. In this chapter, simulation studies were conducted to compare the two optimization procedures based on each method's level of accuracy and efficiency. Unless the user has prior knowledge of the location of the true optimum, the GA is preferred over the Nelder-Mead simplex algorithm as the simplex is far too sensitive to starting values. In addition, the GA is the superior method as it can handle more complex objective functions than the simplex. As for the recommended population size for the GA, if the user is not concerned with computational speed, the GA with a population size of $m = 50$ is recommended for use in practice as it repeatedly yields the most accurate results. On the other hand, if the user is concerned with both computational speed and accuracy, the GA with a population size of $m = 4$ is recommended as it converges much faster than the large population GA yet still remains somewhat competitive in terms of accuracy.

Chapter 7

Summary and Future Research

Response Surface Methodology (RSM) has been a popular strategy among industrial statisticians since the seminole work by Box and Wilson (1951). RSM methods have been used to study the relationship between explanatory variables and one or more responses of interest and to identify the settings of the explanatory variables which result in an optimal product or process. Traditional RSM techniques, which heavily rely on parametric models, can lead to highly biased estimates and miscalculated optimal regressor settings when the user's model is incorrectly specified. Nonparametric methods have been suggested as an alternative, yet they often result in highly variable estimates, especially in the small sample settings common in RSM. Therefore, we have proposed the use of semiparametric methods and the genetic algorithm (GA) in the RSM setting. This chapter summarizes the work from the previous chapters and then proposes areas for future research.

7.1 Summary

In Chapter 3, we investigate the recently proposed nonparametric RSM by examining the definition of degrees of freedom for this method. The parametric definitions for error degrees of freedom have been utilized throughout the nonparametric regression literature. Even though they are not theoretically correct in the local linear regression setting, these definitions are more appropriate than Cramér's (1946) theoretical definition as they incorporate the degree of smoothing. Moreover, the computationally simple definition, n minus the trace of the "HAT" or smoother matrix, outperforms the definition that is closer to the theoretical one. Utilizing this simple definition in the denominator of PRESS** yields bandwidths closer to the true optimal values and superior fits to the response surface when using local linear regression.

As we have seen in Chapters 4 and 5, the semiparametric approach yields better estimation and optimization results. In Chapter 4, we apply the methods to the simple RSM scenario in which we have a single response and the homogeneity of the variances assumption is valid. In the chemical process and motor oil studies, the semiparametric approach is superior to the parametric and nonparametric in terms of model performance statistics. Moreover, in the motor oil study, the semiparametric approach is superior in terms of optimization results. Furthermore, simulation studies show the semiparametric approach is highly competitive to the parametric when the user correctly specifies the model and superior to both the parametric and nonparametric when the model is misspecified. In Chapter 5, we find the semiparametric approach retains its superiority when extended to the dual model problem, which involves the process mean and variance. In the printing ink study, the semiparametric approach yields better optimization results than the parametric and nonparametric approaches. In addition, the semiparametric approach is the only one which consistently performs well over all degrees of potential model misspecification in the simulation studies.

Thus, building off of Myers (1999), semiparametric RSM is suggested for use in the following

scenarios:

(i) The researcher is interested in optimizing a response(s).

(ii) The researcher is less interested in an interpretive function (i.e., interpreting the estimated regression coefficients) and more interested in studying the shape of the response surface.

(iii) The researcher has partial knowledge of the true underlying model.

(iv) The data contains important "bumps" that parametric models cannot capture.

The estimates from the superior semiparametric method do not result in closed form expressions; therefore, Chapter 6 addresses the need for a more flexible optimization technique which does not require the calculation of derivatives. We propose the use of genetic algorithms over simplex methods because simplex methods have been found to be inefficient and often yield local as opposed to global optimum. The proposed GA has been successfully used in the chemical process and motor oil studies of Chapter 4 and the printing ink study of Chapter 5. The simulation studies of Chapter 6 show the GA superior to the Nelder-Mead simplex as the GA is insensitive to initial starting values and can handle complex objective functions. Moreover, the GA with a population size equal to twice the dimensionality of the problem performs best overall when efficiency and accuracy are of concern.

7.2 Future Research

The work presented in this dissertation is a start to applying semiparametric regression methods and genetic algorithms to the RSM setting. More work still needs to be done to fully understand the use of these methods in RSM. Of particular interest is the study of optimal designs for semiparametric methods. Anderson-Cook and Prewitt (2005) argue that

symmetric designs should perform better than non-symmetric designs for nonparametric smoothing. Intuitively, the same should hold for semiparametric method, yet more research needs to be completed in this area. In addition, the location and number of design points should also be studied.

In this research, global mixing parameters are used for the semiparametric methods. However, there may be areas of the response surface in which the user's parametric model performs well and little, if any, addition from the nonparametric smooth is necessary. At the same time, there may be other areas which require a great deal of smoothing. This leads to the idea of local mixing parameters. By allowing the mixing parameter to vary by location, the semiparametric method may prove to be even more superior. Future research on this topic would definitely be interesting.

Other areas for future research are extensions of semiparametric methods and the genetic algorithm. The single response and dual model RSM problems have been studied in this work. It would be interesting to extend these methods to the multiple response problem, which may involve the optimization of a desirability function, and mixture experiments. Also, the GA is very flexible and seems to offer many benefits for a variety of applications. It would be interesting to study its use in other areas of RSM beyond optimization. For example, the GA may be applicable in screening experiments to identify the factors that affect the response(s) of interest.

Bibliography

Aitkin, M., 1987. "Modeling variance heterogeneity in Normal regression using GLIM." *Applied Statistics* 36, 332-339.

Anderson-Cook, C.M., Prewitt, K., 2005. "Some guidelines for using nonparametric methods for modeling data from response surface designs." *Journal of Modern Applied Statistical Methods* 4, 106-119.

Bäck, T., 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York.

Bartlett, M.S., Kendall, D.G., 1946. "The statistical analysis of variance heterogeneity and the logarithmic transformation." *Journal of the Royal Statistical Society Series B* 8, 128-138.

Barton, R.R., Ivey, J.S., Jr., 1996. "Nelder-Mead simplex modifications for simulation optimization." *Management Science* 42, 954-973.

Borkowski, J.J., 2003. "Using a genetic algorithm to generate small exact response surface designs." *Journal of Probability and Statistical Science* 1, 65-68.

Box, G.E.P., Draper, N.R., 1987. *Empirical Model Building and Response Surface*. Wiley, New York.

Box, G.E.P., Meyer, R.D., 1986. "Dispersion effects from fractional designs." *Technometrics* 28, 19-27.

Box, G.E.P., Wilson, K.B., 1951. "On the experimental attainment of optimum conditions." *Journal of the Royal Statistical Society Series B* 13, 1-45.

Cleveland, W.S., 1979. "Robust locally weighted regression and smoothing scatterplots." *Journal of the American Statistical Association*, 74, 829-836.

Cramér, H., 1946. *Mathematical Methods of Statistics*. Princeton University Press, Princeton.

De Jong, K., 1975. "An analysis of the behavior of a class of genetic adaptive systems." PhD Dissertation. Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI.

Einsporn, R., Birch, J.B., 1993. "Model robust regression: using nonparametric regression to improve parametric regression analyses." Technical Report 93-5. Department of Statistics, Virginia Polytechnic Institute & State University, Blacksburg, VA.

Fox, J., 2000. *Nonparametric Simple Regression: Smoothing Scatterplots*. Sage, Thousand Oaks, CA.

Fan, J., Gijbels, I., 1996. *Local Polynomial Modeling and Its Applications*. Chapman & Hall, London.

Fan, J., Gijbels, I., 2000. "Local polynomial fitting." In: Schimek, M.G. (Ed.), *Smoothing and Regression: Approaches, Computation, and Application*. Wiley, New York, pp. 229-276.

Fan, J., Heckman, N.E., Wand, M.P., 1995. "Local polynomial kernel regression for generalized linear models and quasi-likelihood functions." *Journal of the American Statistical Association* 90, 141-150.

Gasser, T., Müller, H.G., 1984. "Estimating regression functions and their derivatives by the kernel method." *Scandinavian Journal of Statistics* 11, 171-185.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA.

Good, I.J., 1973. "What are degrees of freedom?" *The American Statistician* 27, 227-228.

Grefenstette, J.J., 1986. "Optimization of control parameters for genetic algorithms." *IEEE Transactions on Systems, Man, and Cybernetics*, 16, 122-128.

Grego, J.M., 1993. "Generalized linear models and process variation." *Journal of Quality Technology* 25, 288-295.

Härdle, W., 1990. *Applied Non-Parametric Regression.* Cambridge University Press, Cambridge.

Härdle, W., Müller, M., Sperlichm, S., Werwatz, A., 2004. *Nonparametric and Semiparametric Models.* Springer, Berlin.

Hastie, T.J., Tibshirani, R.J., 1990. *Generalized Additive Models.* Chapman & Hall, London.

Haupt, R.L., Haupt, S.E., 2004. *Practical Genetic Algorithms.* Wiley, New York.

Heredia-Langner, A., Carlyle, W.M., Montgomery, D.C., Borror, C.M., Runger, G.C., 2003. "Genetic algorithms for the construction of D-optimal designs." *Journal of Quality Technology* 35, 28-46.

Heredia-Langner, A., Montgomery, D.C., Carlyle, W.M., Borror, C.M., 2004. "Model-robust optimal designs: a genetic algorithm approach." *Journal of Quality Technology* 36, 263-279.

Holland, J., 1975. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor.

Lee, Y., Nelder, J.A., 2003. "Robust design via generalized linear models." *Journal of Quality Technology* 35, 2-12.

Lin, X., Carroll, R.J., 2000. "Nonparametric function estimation for clustered data when the predictor is measured without/with error." *Journal of the American Statistical Association* 95, 520-534.

Mayer, D.G., Belward, J.A., Burrage, K., 2001. "Robust parameter settings of evolutionary algorithms for the optimization of agricultural systems models." *Agricultural Systems* 69, 199-213.

Mays, J., 1995. "Model robust regression: combining parametric, nonparametric, and semiparametric methods." PhD Dissertation. Department of Statistics, Virginia Polytechnic Institute & State University, Blacksburg, VA.

Mays, J., Birch, J.B., Einsporn, R., 2000. "An overview of model-robust regression." *Journal of Statistical Computation and Simulation* 66, 79-100.

Mays, J., Birch, J.B., Starnes, B.A., 2001. "Model robust regression: combining parametric, nonparametric, and semiparametric methods." *Journal of Nonparametric Statistics* 13, 245-277.

Myers, R.H., 1990. *Classical and Modern Regression with Applications*, 2nd ed. Duxbury, Pacific Grove, CA.

Myers, R.H., 1999. "Response surface methodology - current status and future directions." *Journal of Quality Technology* 31, 30-44.

Myers, R.H., Montgomery, D.C., 2002. *Response Surface Methodology: Process and Product Optimization Using Design Experiments*, 2nd ed. Wiley, New York.

Myers, W.R., Brenneman, W.A., Myers, R.H., 2005. "A dual-response approach to robust parameter design for a generalized linear model." *Journal of Quality Technology* 37, 130-138.

Nadaraya, E., 1964. "On estimating regression." *Theory of Probability and Its Applications* 9, 141-142.

Nash, J.C., 1990. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation*, 2nd ed. Adam Hilger, Bristol, England.

Nelder, J., Mead, R., 1965. "A simplex method for function minimization." *Computer Journal* 7, 308-311.

Ortiz, F., Simpson, J.R., Pignatiello, J.J., Heredia-Langner, A., 2004. "A genetic algorithm approach to multiple-response optimization." *Journal of Quality Technology* 36, 432-450.

Priestley, M.B., Chao, M.T., 1972. "Non-parametric function fitting." *Journal of the Royal Statistical Society Series B* 34, 385-392.

Robinson, T.J., Birch, J.B., 2002. "Dual model robust regression: robust to model misspecification." Technical Report 02-2. Department of Statistics, Virginia Polytechnic Institute & State University, Blacksburg, VA.

Ruppert, D., Wand, M.P., Carroll, R.J., 2003. *Semiparametric Regression*. Cambridge University Press, Cambridge.

Scott, D.W., 1992. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York.

Simonoff, J.S., 1996. *Smoothing Methods in Statistics*. Springer-Verlag, New York.

Speckman, P., 1988. "Kernel smoothing in partial linear Models." *Journal of the Royal Statistical Society Series B* 50, 413-436.

Takezawa, K., 2006. *Introduction to Nonparametric Regression*. Wiley, Hoboken.

Vetterling, W.T., Teukolsky, S.A., Press, W.H., Flannery, B.P., 1992. *Numerical Recipes: Example Book (Fortran)*, 2nd ed. Cambridge University Press, Cambridge.

Vining, G.G., Bohn, L.L., 1998. "Response surfaces for the mean and variance using a nonparametric approach." *Journal of Quality Technology* 30, 282-291.

Vining, G.G., Myers, R.H., 1990. "Combining Taguchi and response surface philosophies: a dual response approach." *Journal of Quality Technology* 22, 38-45.

Watson, G., 1964. "Smoothing regression analysis." *Sankhya Series A* 26, 359-372.

Wright, M.H., 1996. "Direct search methods: once scorned, now respectable." *Numerical Analysis 1995*, 191-208. Addison Wesley Longman, Harlow, UK.

Zhang, C., 2003. "Calibrating the degrees of freedom for automatic data smoothing and effective curve checking." *Journal of the American Statistical Association* 98, 609-628.

Appendix A

Bias, Variance, and Mean Squared Error Derivations for Single Response Problem

A.1 Underlying Model

Since the three approaches being compared involve fits to parametric and nonparametric models, it seems logical to express the underlying model as a combination of parametric and nonparametric functions. Thus, consider the underlying model $\mathbf{y} = \mathbf{f}(\tilde{\mathbf{x}}) + \boldsymbol{\varepsilon}$ where $\mathbf{f}(\tilde{\mathbf{x}}) = \mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta}) + \mathbf{u}(\tilde{\mathbf{X}})$, $\mathrm{E}(\boldsymbol{\varepsilon})=\mathbf{0}$, and $\mathrm{Var}(\boldsymbol{\varepsilon})=\sigma^2\mathbf{I}$. In our research, the parametric portion of the model is expressed as $\mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta}) = \mathbf{X}\boldsymbol{\beta}$ and the nonparametric portion is expressed as $\mathbf{u}(\tilde{\mathbf{X}}) = [u(\tilde{\mathbf{x}}_1), \ldots, u(\tilde{\mathbf{x}}_n)]'$ where u is considered to be an unknown (smooth) function. If the researcher assumes $\mathbf{u}(\tilde{\mathbf{X}}) = \mathbf{0}$, then the analysis is conducted using the parametric approach. On the other hand, if $\mathbf{f}(\tilde{\mathbf{x}}) \neq \mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta})$ is assumed, the researcher uses the nonparametric approach. The semiparametric approach assumes that an underlying function can be decomposed into a parametric and a "remainder" portion. Therefore, $\mathbf{u}(\tilde{\mathbf{X}})$ may

be thought of as the "remainder" portion since $\mathbf{u}(\tilde{\mathbf{X}}) = \mathbf{f}(\tilde{\mathbf{x}}) - \mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta})$. For simplicity of notation, the model

$$\begin{aligned} \mathbf{y} &= \mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta}) + \mathbf{u}(\tilde{\mathbf{X}}) + \boldsymbol{\varepsilon} \\ &= \mathbf{X}\boldsymbol{\beta} + \mathbf{u}(\tilde{\mathbf{X}}) + \boldsymbol{\varepsilon} \end{aligned}$$

will be written as:

$$\begin{aligned} \mathbf{y} &= \mathbf{m} + \mathbf{u} + \boldsymbol{\varepsilon} \\ &= \mathbf{X}\boldsymbol{\beta} + \mathbf{u} + \boldsymbol{\varepsilon}. \end{aligned}$$

The next sections provide the bias and variance expressions for the response estimates for each approach, parametric, nonparametric, and semiparametric. From these expressions, one can obtain the mean squared error (MSE) by squaring the bias and adding the variance. It is important to note that all of the following results are for a fixed bandwidth (b) and a fixed mixing parameter (λ).

A.2 Parametric Approach

Consider first the parametric approach in which ordinary least squares (OLS) is utilized. Here $\hat{\mathbf{y}}^{(OLS)} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}^{(OLS)}\mathbf{y}$, and the bias of $\hat{\mathbf{y}}^{(OLS)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}^{(OLS)}) &= \text{E}(\hat{\mathbf{y}}^{(OLS)}) - \text{E}(\mathbf{y}) \\
&= \text{E}(\mathbf{H}^{(OLS)}\mathbf{y}) - E(\mathbf{y}) \\
&= \mathbf{H}^{(OLS)}E(\mathbf{y}) - \mathbf{f} && (\text{since } \text{E}[\boldsymbol{\varepsilon}] = \mathbf{0}) \\
&= \mathbf{H}^{(OLS)}\mathbf{f} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}^{(OLS)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}^{(OLS)}\mathbf{X}\boldsymbol{\beta} + \mathbf{H}^{(OLS)}\mathbf{u} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{X}\boldsymbol{\beta} + \mathbf{H}^{(OLS)}\mathbf{u} - \mathbf{X}\boldsymbol{\beta} - \mathbf{u} && (\text{since } \mathbf{H}^{(OLS)}\mathbf{X} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = \mathbf{X}) \\
&= \mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(OLS)})\mathbf{u} \\
&= -(\mathbf{I} - \mathbf{H}^{(OLS)})\mathbf{u}.
\end{aligned}$$

The variance of $\hat{\mathbf{y}}^{(OLS)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\mathbf{y}}^{(OLS)}) &= \text{Var}(\mathbf{H}^{(OLS)}\mathbf{y}) \\
&= \mathbf{H}^{(OLS)}\text{Var}(\mathbf{y})\mathbf{H}^{(OLS)\prime} \\
&= \mathbf{H}^{(OLS)}(\sigma^2\mathbf{I})\mathbf{H}^{(OLS)\prime} && (\text{since } \text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}) \\
&= \sigma^2\mathbf{H}^{(OLS)}\mathbf{H}^{(OLS)\prime} \\
&= \sigma^2\mathbf{H}^{(OLS)} && (\text{since } \mathbf{H}^{(OLS)} \text{ is symmetric and idempotent}).
\end{aligned}$$

A.3 Nonparametric Approach

Consider the nonparametric approach in which local linear regression (LLR) is utilized. Here $\hat{\mathbf{y}}^{(LLR)} = \mathbf{H}^{(LLR)}\mathbf{y}$, and the bias of $\hat{\mathbf{y}}^{(LLR)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}^{(LLR)}) &= \text{E}(\hat{\mathbf{y}}^{(LLR)}) - \text{E}(\mathbf{y}) \\
&= \text{E}(\mathbf{H}^{(LLR)}\mathbf{y}) - E(\mathbf{y}) \\
&= \mathbf{H}^{(LLR)}E(\mathbf{y}) - \mathbf{f} \\
&= \mathbf{H}^{(LLR)}\mathbf{f} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}^{(LLR)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= -(\mathbf{I} - \mathbf{H}^{(LLR)})(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}).
\end{aligned}$$

The variance of $\hat{\mathbf{y}}^{(LLR)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\mathbf{y}}^{(LLR)}) &= \text{Var}(\mathbf{H}^{(LLR)}\mathbf{y}) \\
&= \mathbf{H}^{(LLR)}\text{Var}(\mathbf{y})\mathbf{H}^{(LLR)\prime} \\
&= \mathbf{H}^{(LLR)}(\sigma^2\mathbf{I})\mathbf{H}^{(LLR)\prime} \\
&= \sigma^2\mathbf{H}^{(LLR)}\mathbf{H}^{(LLR)\prime}.
\end{aligned}$$

A.4 Semiparametric Approach

Consider the semiparametric approach in which model robust regression 2 (MRR2) is utilized. Here $\hat{\mathbf{y}}^{(MRR2)} = \mathbf{H}^{(MRR2)}\mathbf{y} = [\mathbf{H}^{(OLS)} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})]\mathbf{y}$, and the bias of $\hat{\mathbf{y}}^{(MRR2)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}^{(MRR2)}) &= \text{E}(\hat{\mathbf{y}}^{(MRR2)}) - \text{E}(\mathbf{y}) \\
&= \text{E}(\mathbf{H}^{(MRR2)}\mathbf{y}) - E(\mathbf{y}) \\
&= \mathbf{H}^{(MRR2)}E(\mathbf{y}) - \mathbf{f} \\
&= \mathbf{H}^{(MRR2)}\mathbf{f} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}^{(MRR2)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}^{(MRR2)}\mathbf{X}\boldsymbol{\beta} + \mathbf{H}^{(MRR2)}\mathbf{u} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}^{(MRR2)}\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{u} \\
&= [\mathbf{H}^{(OLS)} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})]\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{u} \\
&= \mathbf{H}^{(OLS)}\mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{u} \\
&= \mathbf{H}^{(OLS)}\mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{X}\boldsymbol{\beta} - \mathbf{H}^{(OLS)}\mathbf{X}\boldsymbol{\beta}) - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{u} \\
&= \mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta}) - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{u} \\
&= -(\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{u}.
\end{aligned}$$

The variance of $\hat{\mathbf{y}}^{(MRR2)}$ is given by:

$$\begin{aligned}
\mathrm{Var}(\hat{\mathbf{y}}^{(MRR2)}) &= \mathrm{Var}(\mathbf{H}^{(MRR2)}\mathbf{y}) \\
&= \mathbf{H}^{(MRR2)}\mathrm{Var}(\mathbf{y})\mathbf{H}^{(MRR2)\prime} \\
&= \mathbf{H}^{(MRR2)}(\sigma^2\mathbf{I})\mathbf{H}^{(MRR2)\prime} \\
&= \sigma^2\mathbf{H}^{(MRR2)}\mathbf{H}^{(MRR2)\prime} \\
&= \sigma^2[\mathbf{H}^{(OLS)} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})][\mathbf{H}^{(OLS)} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})]' \\
&= \sigma^2[\mathbf{H}^{(OLS)}\mathbf{H}^{(OLS)\prime} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})\mathbf{H}^{(OLS)\prime} \\
&\quad +\lambda\mathbf{H}^{(OLS)}(\mathbf{I} - \mathbf{H}^{(OLS)})'\mathbf{H}_r^{(LLR)\prime} + \lambda^2\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})(\mathbf{I} - \mathbf{H}^{(OLS)})'\mathbf{H}_r^{(LLR)\prime}] \\
&= \sigma^2[\mathbf{H}^{(OLS)} + \lambda\mathbf{H}_r^{(LLR)}(\mathbf{H}^{(OLS)} - \mathbf{H}^{(OLS)}) + \lambda(\mathbf{H}^{(OLS)} - \mathbf{H}^{(OLS)})\mathbf{H}_r^{(LLR)\prime} \\
&\quad +\lambda^2\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})\mathbf{H}_r^{(LLR)\prime}] \\
&= \sigma^2[\mathbf{H}^{(OLS)} + \lambda^2\mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})\mathbf{H}_r^{(LLR)\prime}].
\end{aligned}$$

Appendix B

Bias, Variance, and Mean Squared Error Derivations for Dual Response Problem

B.1 Underlying Model

Consider the underlying means model $\mathbf{y} = \mathbf{f}(\tilde{\mathbf{x}}) + \mathbf{g}(\tilde{\mathbf{x}}^*)\boldsymbol{\varepsilon}$ where $\mathbf{f}(\tilde{\mathbf{x}}) = \mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta}) + \mathbf{u}(\tilde{\mathbf{X}})$, $\mathrm{E}(\boldsymbol{\varepsilon})=\mathbf{0}$, and $\mathrm{Var}(\boldsymbol{\varepsilon})=\mathbf{I}$. The variance is given by the model $\boldsymbol{\sigma}^2 = \mathbf{g}(\tilde{\mathbf{x}}^*)$ and the transformed variance model is given by $\boldsymbol{\tau} = \mathbf{l}(\tilde{\mathbf{X}}^*; \boldsymbol{\gamma}) + \mathbf{v}(\tilde{\mathbf{X}}^*) + \boldsymbol{\eta}$ with $\mathrm{E}(\boldsymbol{\eta})=\mathbf{0}$ and $\mathrm{Var}(\boldsymbol{\eta})=\delta^2\mathbf{I}$, where

$$\boldsymbol{\tau} = \begin{bmatrix} \ln(s_1^2) \\ \ln(s_2^2) \\ \vdots \\ \ln(s_d^2) \end{bmatrix}.$$

In our research, the parametric portion of the models are expressed as $\mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta}) = \mathbf{X}\boldsymbol{\beta}$ and $\mathbf{l}(\tilde{\mathbf{X}}^*; \boldsymbol{\gamma}) = \mathbf{X}^*\boldsymbol{\gamma}$, and the nonparametric portions are expressed as $\mathbf{u}(\tilde{\mathbf{X}}) = [u(\tilde{\mathbf{x}}_1), \ldots, u(\tilde{\mathbf{x}}_k)]'$ and $\mathbf{v}(\tilde{\mathbf{X}}^*) = [v(\tilde{\mathbf{x}}_1^*), \ldots, v(\tilde{\mathbf{x}}_l^*)]'$ where u and v are considered to be

unknown (smooth) functions. For simplicity of notation, the means model

$$\begin{aligned} \mathbf{y} &= \mathbf{m}(\tilde{\mathbf{X}}; \boldsymbol{\beta}) + \mathbf{u}(\tilde{\mathbf{X}}) + \mathbf{g}(\tilde{\mathbf{x}}^*)\boldsymbol{\varepsilon} \\ &= \mathbf{X}\boldsymbol{\beta} + \mathbf{u}(\tilde{\mathbf{X}}) + \mathbf{g}(\tilde{\mathbf{x}}^*)\boldsymbol{\varepsilon} \end{aligned}$$

will be written as:

$$\begin{aligned} \mathbf{y} &= \mathbf{m} + \mathbf{u} + \mathbf{g}\boldsymbol{\varepsilon} \\ &= \mathbf{X}\boldsymbol{\beta} + \mathbf{u} + \mathbf{g}\boldsymbol{\varepsilon}, \end{aligned}$$

and the transformed variance model

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{l}(\tilde{\mathbf{X}}^*; \boldsymbol{\gamma}) + \mathbf{v}(\tilde{\mathbf{X}}^*) + \boldsymbol{\eta} \\ &= \mathbf{X}^*\boldsymbol{\gamma} + \mathbf{v}(\tilde{\mathbf{X}}^*) + \boldsymbol{\eta} \end{aligned}$$

will be written as:

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{l} + \mathbf{v} + \boldsymbol{\eta} \\ &= \mathbf{X}^*\boldsymbol{\gamma} + \mathbf{v} + \boldsymbol{\eta}. \end{aligned}$$

The next sections provide the bias and variance expressions for the response estimates for each approach, parametric, nonparametric, and semiparametric. From these expressions, one can obtain the mean squared error (MSE) by squaring the bias and adding the variance. It is important to note that all of the following results are for a fixed bandwidths (b_μ and b_σ) and fixed mixing parameters (λ_μ and λ_σ).

B.2 Parametric Approach

Consider the parametric approach in which the transformed variance model fits are obtained first by ordinary least squares (OLS) and can be written as $\hat{\boldsymbol{\tau}}^{(OLS)} = \mathbf{H}_{\sigma}^{(OLS)}\boldsymbol{\tau}$, where $\mathbf{H}_{\sigma}^{(OLS)} = \mathbf{X}^{*}(\mathbf{X}^{*\prime}\mathbf{X}^{*})^{-1}\mathbf{X}^{*\prime}$. The variance model fits can then be expressed as $\hat{\mathbf{V}}^{(OLS)} = diag(\hat{\sigma}_1^{2(OLS)}, \ldots, \hat{\sigma}_d^{2(OLS)})$, where $\hat{\sigma}_i^{2(OLS)} = \exp[\hat{\tau}_i^{(OLS)}]$ and $\hat{\tau}_i^{(OLS)} = \mathbf{x}_i^{*\prime}\hat{\boldsymbol{\gamma}}^{(OLS)}$. The means model fits are then obtained by estimated weighted least squares (EWLS) and can be written as $\hat{\mathbf{y}}^{(EWLS)} = \mathbf{H}_{\mu}^{(EWLS)}\mathbf{y}$, where $\mathbf{H}_{\mu}^{(EWLS)} = \mathbf{X}(\mathbf{X}'\hat{\mathbf{V}}^{(OLS)-1}\mathbf{X})^{-1}\mathbf{X}'\hat{\mathbf{V}}^{(OLS)-1}$. In the expectation and variance formulas that follow, note that $\hat{\mathbf{V}}^{(OLS)}$ is replaced with $\mathbf{V}$, where $\mathbf{V} = diag(\sigma_1^2, \ldots, \sigma_d^2)$ and $\sigma_i^2 = \exp[\tau_i]$ with $\tau_i = \mathbf{x}_i^{*\prime}\boldsymbol{\gamma} + v_i$. Thus, the bias and variance equations become approximate. The bias of $\hat{\mathbf{y}}^{(WLS)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}^{(WLS)}) &= \text{E}(\hat{\mathbf{y}}^{(WLS)}) - \text{E}(\mathbf{y}) \\
&= \text{E}(\mathbf{H}_{\mu}^{(WLS)}\mathbf{y}) - E(\mathbf{y}) \\
&= \mathbf{H}_{\mu}^{(WLS)}E(\mathbf{y}) - \mathbf{f} \\
&= \mathbf{H}_{\mu}^{(WLS)}\mathbf{f} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}_{\mu}^{(WLS)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}_{\mu}^{(WLS)}\mathbf{X}\boldsymbol{\beta} + \mathbf{H}_{\mu}^{(WLS)}\mathbf{u} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{X}\boldsymbol{\beta} + \mathbf{H}_{\mu}^{(WLS)}\mathbf{u} - \mathbf{X}\boldsymbol{\beta} - \mathbf{u} \\
&= \mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}_{\mu}^{(WLS)})\mathbf{u} \\
&= -(\mathbf{I} - \mathbf{H}_{\mu}^{(WLS)})\mathbf{u}.
\end{aligned}$$

The variance of $\hat{\mathbf{y}}^{(WLS)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\mathbf{y}}^{(WLS)}) &= \text{Var}(\mathbf{H}_{\mu}^{(WLS)}\mathbf{y}) \\
&= \mathbf{H}_{\mu}^{(WLS)}\text{Var}(\mathbf{y})\mathbf{H}_{\mu}^{(WLS)\prime} \\
&= \mathbf{H}_{\mu}^{(WLS)}\mathbf{V}\mathbf{H}_{\mu}^{(WLS)\prime}.
\end{aligned}$$

The bias of $\hat{\boldsymbol{\tau}}^{(OLS)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\boldsymbol{\tau}}^{(OLS)}) &= \text{E}(\hat{\boldsymbol{\tau}}^{(OLS)}) - \text{E}(\boldsymbol{\tau}) \\
&= \text{E}[\mathbf{H}_{\sigma}^{(OLS)}\boldsymbol{\tau}] - E(\boldsymbol{\tau}) \\
&= \mathbf{H}_{\sigma}^{(OLS)}\text{E}(\boldsymbol{\tau}) - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= \mathbf{H}_{\sigma}^{(OLS)}(\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= \mathbf{H}_{\sigma}^{(OLS)}\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{H}_{\sigma}^{(OLS)}\mathbf{v} - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= \mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{H}_{\sigma}^{(OLS)}\mathbf{v} - \mathbf{X}^{*}\boldsymbol{\gamma} - \mathbf{v} \\
&= \mathbf{X}^{*}\boldsymbol{\gamma} - \mathbf{X}^{*}\boldsymbol{\gamma} - (\mathbf{I} - \mathbf{H}_{\sigma}^{(OLS)})\mathbf{v} \\
&= -(\mathbf{I} - \mathbf{H}_{\sigma}^{(OLS)})\mathbf{v}.
\end{aligned}$$

The variance of $\hat{\boldsymbol{\tau}}^{(OLS)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\boldsymbol{\tau}}^{(OLS)}) &= \text{Var}(\mathbf{H}_{\sigma}^{(OLS)}\boldsymbol{\tau}) \\
&= \mathbf{H}_{\sigma}^{(OLS)}\text{Var}(\boldsymbol{\tau})\mathbf{H}_{\sigma}^{(OLS)\prime} \\
&= \mathbf{H}_{\sigma}^{(OLS)}(\delta^{2}\mathbf{I})\mathbf{H}_{\sigma}^{(OLS)\prime} \\
&= \delta^{2}\mathbf{H}_{\sigma}^{(OLS)}\mathbf{H}_{\sigma}^{(OLS)\prime} \\
&= \delta^{2}\mathbf{H}_{\sigma}^{(OLS)}.
\end{aligned}$$

B.3 Nonparametric Approach

Consider the nonparametric approach in which the transformed variance model fits are obtained first by local linear regression (LLR) and can be written as $\hat{\boldsymbol{\tau}}^{(LLR)} = \mathbf{H}_{\sigma}^{(LLR)}\boldsymbol{\tau}$. The variance model fits can then be expressed as $\hat{\mathbf{V}}^{(LLR)} = diag(\hat{\sigma}_1^{2(LLR)}, \ldots, \hat{\sigma}_d^{2(LLR)})$, where $\hat{\sigma}_i^{2(LLR)} = \exp[\hat{\tau}_i^{(LLR)}]$. The means model fits are then obtained by estimated weighted local linear regression (EWLLR) and can be written as $\hat{\mathbf{y}}^{(EWLLR)} = \mathbf{H}_{\mu}^{(EWLLR)}\mathbf{y}$. In the expectation and variance formulas that follow, note that $\hat{\mathbf{V}}^{(LLR)}$ is replaced with $\mathbf{V}$, where $\mathbf{V} = diag(\sigma_1^2, \ldots, \sigma_d^2)$ and $\sigma_i^2 = \exp[\tau_i]$ with $\tau_i = \mathbf{x}_i^{*\prime}\boldsymbol{\gamma} + v_i$. Thus, the bias and variance equations become approximate. The bias of $\hat{\mathbf{y}}^{(WLLR)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}^{(WLLR)}) &= \text{E}(\hat{\mathbf{y}}^{(WLLR)}) - \text{E}(\mathbf{y}) \\
&= \text{E}(\mathbf{H}_{\mu}^{(WLLR)}\mathbf{y}) - E(\mathbf{y}) \\
&= \mathbf{H}_{\mu}^{(WLLR)}E(\mathbf{y}) - \mathbf{f} \\
&= \mathbf{H}_{\mu}^{(WLLR)}\mathbf{f} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}_{\mu}^{(WLLR)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= -(\mathbf{I} - \mathbf{H}_{\mu}^{(WLLR)})(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}).
\end{aligned}$$

The variance of $\hat{\mathbf{y}}^{(WLLR)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\mathbf{y}}^{(WLLR)}) &= \text{Var}(\mathbf{H}_{\mu}^{(WLLR)}\mathbf{y}) \\
&= \mathbf{H}_{\mu}^{(WLLR)}\text{Var}(\mathbf{y})\mathbf{H}_{\mu}^{(WLLR)\prime} \\
&= \mathbf{H}_{\mu}^{(WLLR)}\mathbf{V}\mathbf{H}_{\mu}^{(WLLR)\prime}.
\end{aligned}$$

The bias of $\hat{\boldsymbol{\tau}}^{(LLR)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\boldsymbol{\tau}}^{(LLR)}) &= \text{E}(\hat{\boldsymbol{\tau}}^{(LLR)}) - \text{E}(\boldsymbol{\tau}) \\
&= \text{E}[\mathbf{H}_{\sigma}^{(LLR)}\boldsymbol{\tau}] - E(\boldsymbol{\tau}) \\
&= \mathbf{H}_{\sigma}^{(LLR)}\text{E}(\boldsymbol{\tau}) - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= \mathbf{H}_{\sigma}^{(LLR)}(\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= -(\mathbf{I} - \mathbf{H}_{\sigma}^{(LLR)})(\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}).
\end{aligned}$$

The variance of $\hat{\boldsymbol{\tau}}^{(LLR)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\boldsymbol{\tau}}^{(LLR)}) &= \text{Var}(\mathbf{H}_{\sigma}^{(LLR)}\boldsymbol{\tau}) \\
&= \mathbf{H}_{\sigma}^{(LLR)}\text{Var}(\boldsymbol{\tau})\mathbf{H}_{\sigma}^{(LLR)\prime} \\
&= \mathbf{H}_{\sigma}^{(LLR)}(\delta^{2}\mathbf{I})\mathbf{H}_{\sigma}^{(LLR)\prime} \\
&= \delta^{2}\mathbf{H}_{\sigma}^{(LLR)}\mathbf{H}_{\sigma}^{(LLR)\prime}.
\end{aligned}$$

B.4 Semiparametric Approach

Consider the semiparametric approach in which the transformed variance model fits are obtained first by model robust regression 1 (MRR1) and can be written as $\hat{\boldsymbol{\tau}}^{(MRR1)} = \mathbf{H}_{\sigma}^{(MRR1)}\boldsymbol{\tau} = [(1-\lambda_{\sigma})\mathbf{H}_{\sigma}^{(OLS)} + \lambda_{\sigma}\mathbf{H}_{\sigma}^{(LLR)}]\boldsymbol{\tau}$. The variance model fits can then be expressed as $\hat{\mathbf{V}}^{(MRR1)} = diag(\hat{\sigma}_1^{2(MRR1)}, \ldots, \hat{\sigma}_d^{2(MRR1)})$, where $\hat{\sigma}_i^{2(MRR1)} = \exp[\hat{\tau}_i^{(MRR1)}]$. The means model fits are then obtained by model robust regression 2 (MRR2) and can be written as $\hat{\mathbf{y}}^{(MRR2)} = \mathbf{H}_{\mu}^{(MRR2)}\mathbf{y} = [\mathbf{H}_{\mu}^{EWLS} + \lambda_{\mu}\mathbf{H}_{r}^{(LLR)}(\mathbf{I} - \mathbf{H}_{\mu}^{(EWLS)})]\mathbf{y}$, where $\mathbf{H}_{r}^{(LLR)}$ is the smoother matrix for the LLR fit to the residuals from the EWLS fit to the means model. In the expectation and variance formulas that follow, note that $\hat{\mathbf{V}}^{(MRR1)}$ is replaced with $\mathbf{V}$, where $\mathbf{V} = diag(\sigma_1^2, \ldots, \sigma_d^2)$ and $\sigma_i^2 = \exp[\tau_i]$ with $\tau_i = \mathbf{x}_i^{*\prime}\boldsymbol{\gamma} + v_i$. Thus, the bias and variance equations become approximate. The bias of $\hat{\mathbf{y}}^{(MRR2)}$ is given by:

$$
\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}^{(MRR2)}) &= \text{E}(\hat{\mathbf{y}}^{(MRR2)}) - \text{E}(\mathbf{y}) \\
&= \text{E}(\mathbf{H}_{\mu}^{(MRR2)}\mathbf{y}) - E(\mathbf{y}) \\
&= \mathbf{H}_{\mu}^{(MRR2)}E(\mathbf{y}) - \mathbf{f} \\
&= \mathbf{H}_{\mu}^{(MRR2)}\mathbf{f} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}_{\mu}^{(MRR2)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}_{\mu}^{(MRR2)}\mathbf{X}\boldsymbol{\beta} + \mathbf{H}_{\mu}^{(MRR2)}\mathbf{u} - (\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= \mathbf{H}_{\mu}^{(MRR2)}\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}_{\mu}^{(MRR2)})\mathbf{u} \\
&= [\mathbf{H}_{\mu}^{WLS} + \lambda_{\mu}\mathbf{H}_{r}^{(LLR)}(\mathbf{I} - \mathbf{H}_{\mu}^{(WLS)})]\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}_{\mu}^{(MRR2)})\mathbf{u} \\
&= \mathbf{H}_{\mu}^{WLS}\mathbf{X}\boldsymbol{\beta} + \lambda_{\mu}\mathbf{H}_{r}^{(LLR)}(\mathbf{I} - \mathbf{H}_{\mu}^{(WLS)})\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}_{\mu}^{(MRR2)})\mathbf{u} \\
&= \mathbf{H}_{\mu}^{WLS}\mathbf{X}\boldsymbol{\beta} + \lambda_{\mu}\mathbf{H}_{r}^{(LLR)}(\mathbf{X}\boldsymbol{\beta} - \mathbf{H}_{\mu}^{(WLS)}\mathbf{X}\boldsymbol{\beta}) - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}_{\mu}^{(MRR2)})\mathbf{u} \\
&= \mathbf{X}\boldsymbol{\beta} + \lambda_{\mu}\mathbf{H}_{r}^{(LLR)}(\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta}) - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}_{\mu}^{(MRR2)})\mathbf{u} \\
&= -(\mathbf{I} - \mathbf{H}_{\mu}^{(MRR2)})\mathbf{u}.
\end{aligned}
$$

The variance of $\hat{\mathbf{y}}^{(MRR2)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\mathbf{y}}^{(MRR2)}) &= \text{Var}(\mathbf{H}_{\mu}^{(MRR2)}\mathbf{y}) \\
&= \mathbf{H}_{\mu}^{(MRR2)}\text{Var}(\mathbf{y})\mathbf{H}_{\mu}^{(MRR2)\prime} \\
&= \mathbf{H}_{\mu}^{(MRR2)}\mathbf{V}\mathbf{H}_{\mu}^{(MRR2)\prime}.
\end{aligned}$$

The bias of $\hat{\boldsymbol{\tau}}^{(MRR1)}$ is given by:

$$\begin{aligned}
\text{Bias}(\hat{\boldsymbol{\tau}}^{(MRR1)}) &= \text{E}(\hat{\boldsymbol{\tau}}^{(MRR1)}) - \text{E}(\boldsymbol{\tau}) \\
&= \text{E}[\mathbf{H}_{\sigma}^{(MRR1)}\boldsymbol{\tau}] - E(\boldsymbol{\tau}) \\
&= \mathbf{H}_{\sigma}^{(MRR1)}\text{E}(\boldsymbol{\tau}) - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= \mathbf{H}_{\sigma}^{(MRR1)}(\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= \mathbf{H}_{\sigma}^{(MRR1)}\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{H}_{\sigma}^{(MRR1)}\mathbf{v} - (\mathbf{X}^{*}\boldsymbol{\gamma} + \mathbf{v}) \\
&= [(1-\lambda_{\sigma})\mathbf{H}_{\sigma}^{(OLS)} + \lambda_{\sigma}\mathbf{H}_{\sigma}^{(LLR)}]\mathbf{X}^{*}\boldsymbol{\gamma} - \mathbf{X}^{*}\boldsymbol{\gamma} - (\mathbf{I} - \mathbf{H}_{\sigma}^{(MRR1)})\mathbf{v} \\
&= (1-\lambda_{\sigma})\mathbf{H}_{\sigma}^{(OLS)}\mathbf{X}^{*}\boldsymbol{\gamma} + \lambda_{\sigma}\mathbf{H}_{\sigma}^{(LLR)}\mathbf{X}^{*}\boldsymbol{\gamma} - \mathbf{X}^{*}\boldsymbol{\gamma} - (\mathbf{I} - \mathbf{H}_{\sigma}^{(MRR1)})\mathbf{v} \\
&= (1-\lambda_{\sigma})\mathbf{X}^{*}\boldsymbol{\gamma} + \lambda_{\sigma}\mathbf{H}_{\sigma}^{(LLR)}\mathbf{X}^{*}\boldsymbol{\gamma} - \mathbf{X}^{*}\boldsymbol{\gamma} - (\mathbf{I} - \mathbf{H}_{\sigma}^{(MRR1)})\mathbf{v} \\
&= -\lambda_{\sigma}\mathbf{X}^{*}\boldsymbol{\gamma} + \lambda_{\sigma}\mathbf{H}_{\sigma}^{(LLR)}\mathbf{X}^{*}\boldsymbol{\gamma} - (\mathbf{I} - \mathbf{H}_{\sigma}^{(MRR1)})\mathbf{v} \\
&= -\lambda_{\sigma}(\mathbf{I} - \mathbf{H}_{\sigma}^{(LLR)})\mathbf{X}^{*}\boldsymbol{\gamma} - (\mathbf{I} - \mathbf{H}_{\sigma}^{(MRR1)})\mathbf{v}
\end{aligned}$$

The variance of $\hat{\boldsymbol{\tau}}^{(MRR1)}$ is given by:

$$\begin{aligned}
\text{Var}(\hat{\boldsymbol{\tau}}^{(MRR1)}) &= \text{Var}(\mathbf{H}_{\sigma}^{(MRR1)}\boldsymbol{\tau}) \\
&= \mathbf{H}_{\sigma}^{(MRR1)}\text{Var}(\boldsymbol{\tau})\mathbf{H}_{\sigma}^{(MRR1)\prime} \\
&= \mathbf{H}_{\sigma}^{(MRR1)}(\delta^{2}\mathbf{I})\mathbf{H}_{\sigma}^{(MRR1)\prime} \\
&= \delta^{2}\mathbf{H}_{\sigma}^{(MRR1)}\mathbf{H}_{\sigma}^{(MRR1)\prime}.
\end{aligned}$$

Appendix C

R Code for the Genetic Algorithm

The following is the general R code for our genetic algorithm:

```
ga.fn<-function(pars){
    ♯This function will perform the Genetic Algorithm
    ♯For some specified objective function (obj.fn)
    ♯with specified parameter values (pars)

    ♯Stopping Criteria♯
    maxiter<-500 ♯set maximum number of iterations
    satiter<-10 ♯set number of iterations during which the best results keep saturating
    epsln<-1e-8 ♯smallest gain worth recognizing

    ♯GA Parameters♯
    m<-4 ♯population size at each generation
    crossrate<-.9 ♯crossover rate
    mutrate<-.2 ♯mutation rate
    zerorate<-.2 ♯zero rate
    onerate<-.2 ♯one rate
    keep<-2 ♯keep top two from each generation to remain unchanged
```

```
num.parents<-m-keep ♯number of population that will be used for mating
mate<-ceiling((num.parents)/2) ♯number of matings

♯Create initial population and initialize 'best result' holders♯
xrange<-matrix(c(0,1,0,1),2,2)
k<-ncol(xrange)
elite<-matrix(0,keep,k) ♯matrix for elite chromosomes to remain unchanged
cross<-matrix(0,num.parents,k) ♯matrix for crossover
iter<-0 ♯generation (iteration) counter
stopcode<-0
inarow<-0 ♯number of iterations with function's value consecutively less than epsln
bestfun<-1e30 ♯essential positive infinity
bestx<-matrix(0,1,k)
f<-rep(0,m) ♯initialize function value for each of m vectors
G<-matrix(runif(m*k,0,1),m,k) ♯initial generation with population size m

♯Iterate through generations♯
while(stopcode==0){
        iter<-iter+1 ♯increments counter
        if(iter>maxiter) stopcode<-2 ♯loop will exit on stopcode=2 for exceeding
                                     ♯the maximum number of iterations
        ♯Evaluate current generation♯
        for(i in 1:m){
                f[i]<-obj.fn(pars,G[i,])
                }
        mat<-cbind(G,f) ♯create matrix of x locations and corresponding function
                        ♯values
        newmat<-mat[order(mat[,3]),] ♯sort matrix of by function values increasing
        minf<-newmat[1,3] ♯optimal function (minimum)
        bf<-min(minf, bestfun)
```

```
fgain<-bestfun-bf ♯fgain is always non-negative it measure the change
if(fgain>epsln) inarow<-0 else inarow<-(inarow+1)
if(fgain>0){
        bestfun<-bf
        bestx<-newmat[1,1:2] ♯optimal x location
        }
if(inarow>satiter) stopcode<-1 ♯loop will exit on stopcode=1 for best result
                                ♯having been achieved
♯Select elite to remain unchanged♯
elite<-newmat[1:keep,1:2]
♯Select parents for mating♯
cross<-newmat[(keep+1):m,1:2]
♯Do Crossover♯
for(i in 1:mate){
        z<-rbinom(1,1,crossrate)
        if(z==1){
                x<-ceiling(runif(1,0,(k-1)))
                temp<-cross[i,(x+1):k]
                cross[i,(x+1):k]<-cross[i+mate,(x+1):k]
                cross[i+mate,(x+1):k]<-temp
                }
        }
♯Do Mutation♯
M<-matrix(runif((num.parents*k),0,1),num.parents,k)
for(i in 1:num.parents){
        for(j in 1:k){
                zz<-rbinom(1,1,mutrate)
                if(zz==1) cross[i,j]<-M[i,j]
```

```
                    }
                }
            ♯Do Zero Gene Operator♯
            for(i in 1:num.parents){
                for(j in 1:k){
                    zzz<-rbinom(1,1,zerorate)
                    if(zzz==1) cross[i,j]<-0
                    }
                }
            ♯Do One Gene Operator♯
            for(i in 1:num.parents){
                for(j in 1:k){
                    zzzz<-rbinom(1,1,onerate)
                    if(zzzz==1) cross[i,j]<-1
                    }
                }
            G<-rbind(elite,cross)
        }
        if(iter<maxiter) status<-0
        else status<-1
        result<-c(bestx,bestfun,iter)
        return(result)
}
```

Vita

Stephanie Marie Pickle was born on January 25, 1979 in Roanoke, Virginia. In 1997, she graduated as salutatorian from Northside High School, Roanoke, Virginia, and enrolled in Roanoke College, Salem, Virginia. She graduated Magna Cum Laude in 2001 with a Bachelor of Science degree in Mathematics. In August, 2001, she entered the graduate program in the Statistics Department at Virginia Polytechnic Institute & State University. She received a Master of Science degree in Statistics in December, 2002, and completed work for a Ph.D. degree in Statistics in June, 2006. She will be joining the Engineering Research and Technology department at DuPont as a consulting statistician in July, 2006.