Model Robust Regression: Combining Parametric, Nonparametric, and Semiparametric Methods

by

James Edward Mays

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

DOCTOR OF PHILOSOPHY
IN
STATISTICS

APPROVED:

Jeffrey B. Birch, Chairman

Raymond H. Myers

Eric P. Smith

Marion R. Reynolds, Jr.

Clint W. Coakley

September 21, 1995

Blacksburg, VA

Keywords: Model misspecification, Local linear regression, Bandwidth, Mixing

MODEL ROBUST REGRESSION:
COMBINING PARAMETRIC, NONPARAMETRIC, AND SEMIPARAMETRIC METHODS

by

James E. Mays

Jeffrey B. Birch, Chairman

Statistics

Abstract

In obtaining a regression fit to a set of data, ordinary least squares regression depends directly on the parametric model formulated by the researcher. If this model is incorrect, a least squares analysis may be misleading. Alternatively, nonparametric regression (kernel or local polynomial regression, for example) has no dependence on an underlying parametric model, but instead depends entirely on the distances between regressor coordinates and the prediction point of interest. This procedure avoids the necessity of a reliable model, but in using no information from the researcher, may fit to irregular patterns in the data. The proper combination of these two regression procedures can overcome their respective problems. Considered is the situation where the researcher has an idea of which model should explain the behavior of the data, but this model is not adequate throughout the entire range of the data. An extension of partial linear regression and two methods of model robust regression are developed and compared in this context. These methods involve parametric fits to the data and nonparametric fits to either the data or residuals. The two fits are then combined in the most efficient proportions via a mixing parameter. Performance is based on bias and variance considerations.

Acknowledgements

I would like to extend my deepest appreciation to my advisor, Dr. Jeffrey Birch. His statistical knowledge and professional guidance have contributed greatly to the successful completion of this dissertation. My thanks also goes to all of the faculty and staff of the Department of Statistics for providing a strong learning environment, with special thanks to my committee members Raymond H. Myers, Eric P. Smith, Marion R. Reynolds, Jr., and Clint W. Coakley for their added interest in my research. Not to be overlooked is the much appreciated computer advice and assistance provided by Michele Marini and Scott Harper.

I am also grateful for the many close friends that I have made throughout my studies at Virginia Tech, who have made this part of my life a truly enjoyable experience. And a special thanks to those who lended a helping hand when it was needed most with the final preparation of this dissertation.

Finally, I would especially like to thank my family for their unending love and support, which has been a constant source of strength throughout my life.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction and Motivation

Historically, the regression problem of describing the behavior of some response variable y via a combination of explanatory, or regressor variables $X_1, X_2, \ldots, X_k$ has received a tremendous amount of attention. All of the regression problem scenarios, and the vast number of solutions presented for these problems are too numerous to mention. The research presented in this paper returns to the basic foundations of many of these regression procedures--the simple idea of fitting a curve to a scatter of points. The goal is a procedure for completing this task which displays better performance and is more versatile than the popular solutions that currently exist.

1.A Statement of the Problem

The basic regression problem involves a variable y whose response in a particular process is explained by one or more regressor variables $X_1, X_2, \ldots, X_k$ according to a model of the form

$$y = f(X_1, X_2, \ldots, X_k) + \varepsilon.$$

The term ε is a random error from the process, often assumed to have mean 0 and variance σ^2. The classical parametric regression viewpoint is that the function f is assumed to have a known parametric form, where the parameters are estimated from the data. Important inferences made from the resulting regression analysis depend heavily on the validity of the chosen function for f. Clearly, if f is misspecified, even over only portions of the data, these inferences may be misleading. Thus, knowledge of the appropriate model is crucial when applying the classical parametric regression procedures (such as ordinary least squares).

At the opposite extreme from these parametric procedures are the aptly named nonparametric procedures. Here the function f is considered to be unknown and the user has no parametric specification as to its form. Hence, nonparametric regression must rely totally on the data itself to determine a fit to the scatter of points. Many of these procedures exist, but kernel regression (and local polynomial regression) receive the emphasis in this current work. To see how nonparametric procedures fit the data, consider a point $\mathbf{x}_0 = (X_{10}, \ldots, X_{k0})'$ where the prediction of $E(y) = f(\mathbf{x}_0)$ is desired. The basic idea is that if f is at least somewhat smooth, then the best information on $f(\mathbf{x}_0)$ should come from the y-values at regressor locations $\mathbf{x}_i$ that are closest to $\mathbf{x}_0$. This is exactly what is done in kernel regression--the prediction of $f(\mathbf{x}_0)$ at $\mathbf{x}_0$ is obtained as a weighted sum of y observations, with the weights dependent on the distances of the respective regressor locations from the point of prediction. The greater the distance from $\mathbf{x}_0$, the smaller the weight assigned to the observation at that location. Once this procedure is applied to get predictions at all of the regressor locations, a nonparametric fit to the data is obtained. No closed form expression for f is achieved, but the fit obtained may suggest to the user such a form to study.

Just as in the parametric case, the nonparametric procedures have their disadvantages. Since no information is included from the user, nonparametric fits may fit to superfluous or irregular patterns in the data. This may result in misleading inferences about the process. Also, nonparametric fits tend to be more variable than parametric fits because they rely so much on the scatter of data itself and do not have the underlying stability of a specified functional form. A third drawback, which is intimately connected with the first two, is the problem of how best to determine the weighting scheme to be used. This is discussed later in detail as the problem of bandwidth (or smoothing parameter) selection.

In practice, rarely does the researcher know the exact form of the true function f in the regression equation. However, often he may have an idea, or at least be suspicious, of how the data may behave. For instance, if studying the growth rate of adolescents ages

10-20 (regressing growth rate vs. age), the researcher may strongly suspect some quadratic behavior in the data. In this case, a quadratic model would be specified and a parametric procedure used. However, the researcher may also suspect some deviation in the data from the quadratic model due to the "growth spurt" that occurs over two or three years in a teenager (around age 14 for girls and age 16 for boys). This phenomenon would create an abnormality (a "bump" or peak area) in the quadratic structure of the data. So the researcher has a dilemma. Using a (parametric) quadratic model would explain most of the data, but would be inadequate in capturing the growth spurt phenomenon, whereas using a nonparametric fit would ignore the information that the researcher has about the underlying structure. The usual solution to this problem would be to just settle for the nonparametric fit. A second possible solution would be to fit the quadratic model, perform a test for lack of fit, and use this model if no lack of fit is detected (if lack of fit is detected, use the nonparametric fit). It is very possible that even with the presence of the growth spurt deviation from the quadratic, a lack of fit test would conclude that the specified model is adequate. Thus, the parametric fit would be used and the resulting inferences would likely be misleading due to the inability of the procedure to detect the growth spurt.

1. B Direction of Research

The research presented in this paper provides a solution to the problem described above. That is, how does the researcher obtain a fit that both incorporates his knowledge about a parametric model and is able to detect specific deviations in the data from this model? The solution to this problem should also be versatile enough to handle the following two cases: (1) the researcher believes in a parametric model that is a gross misspecification of the true model (i.e., *robustness* to a misspecified model), and (2) the specified model is adequate throughout all of the data and no specific deviations need to be detected (i.e., a *simple* procedure that would just perform ordinary parametric regression when that is all that is needed). Actually, three possible solutions are studied

and compared to ordinary least squares (OLS) and kernel (or local polynomial) regression. All three procedures involve the *combination* of a parametric fit and a nonparametric fit. The parametric fitting technique used throughout this work is OLS. For the nonparametric fitting technique, kernel regression is used when introducing and explaining the three proposed procedures. However, the final form of the procedures (for implementation) involves the better performing local linear regression as the nonparametric fitting technique.

The first of these procedures was developed by Einsporn (1987) and Einsporn and Birch (1993) and was entitled HATLINK. Here an OLS fit to the data and a kernel fit to the data are combined in a convex combination via a mixing parameter λ. This λ ranges from 0 to 1 based on the amount of misspecification of the specified model. That is, $\lambda = 0$ gives the usual OLS fit (when the model is appropriate), and $\lambda = 1$ gives the kernel fit (when the model is badly misspecified). Due to its ability to handle the varying degrees of model misspecification, this procedure is called Model Robust Regression 1 (MRR1). The second procedure is an adaptation of the semiparametric procedure of partial linear regression (PLR) developed by Speckman (1988). Here the underlying model for the y_i ($i = 1, \ldots, n$) is thought of as being composed of a linear parametric part ($\mathbf{x}_i'\beta$) and a nonparametric part ($m(\mathbf{x}_i')$), where $\mathbf{x}_i' = (X_{1i}, X_{2i}, \ldots, X_{ki})$, β is a vector of unknown parameters, and m is an unknown function. The idea, related to partial correlation analysis of a subset of independent variables in OLS, is to estimate β based on the matrix of regressors $\mathbf{X}$ and the vector of responses $\mathbf{y}$ after "adjusting" them both for partial information from the ~~non~~parametric portion, and to estimate m based on a nonparametric fit to the residuals from the resulting parametric fit. These two fits are added to give the final fit. PLR uses simultaneous fitting techniques and involves the use of residuals to fine tune the fit.

The final procedure to be studied is a proposed method which combines the techniques of these first two procedures. The simplicity of the MRR1 fit is maintained, but the method itself is improved by introducing the use of residuals to fine tune the fit, as

in PLR. This method, denoted MRR2 (Model Robust Regression 2), begins by obtaining a parametric (OLS) fit to the data, based on the user's specified model. A nonparametric (kernel) fit is then obtained on the residuals from the initial OLS fit. A portion of this residual fit is then added back to the OLS fit to give the final MRR2 fit. The residual fit provides the extra structure in the data that the OLS fit cannot capture. What "portion" of the fit to add back is determined by a parameter $\lambda \in [0,1]$, as in the MRR1 procedure.

Inherent in the development and comparison of these three procedures are several other topics that need to be addressed. These include the method of bandwidth choice in kernel regression, the method of choosing the mixing parameter λ, the development of a criterion for comparing the performance of the different methods, the methods of obtaining predictions at points other than data locations, and the development of diagnostics such as error variance estimates and confidence intervals. These issues are addressed in the context of a one-regressor model, keeping in mind the desire to later extend the results to the multiple regression case.

The next chapter gives a brief discussion of parametric regression (OLS in particular), while Chapter 3 gives a detailed review of nonparametric regression (with most emphasis on kernel regression). Also discussed in Chapter 3 is local polynomial regression and its benefits over kernel regression. Chapter 4 provides some discussion of semiparametric procedures, including the introduction of the partial linear model. Then in Chapter 5 the three model robust regression methods--PLR, MRR1, and MRR2--are developed. Chapter 6 presents several comparisons among the three procedures based on a mean squared error criterion. Comparisons are made on several sets of generated and actual data. Chapter 7 contains a preliminary study of data-driven methods that could be used to obtain the fits for the various techniques. Simulation results are given in Chapter 8 in order to substantiate the findings in Chapters 6 and 7. Included in these simulations is a check on the validity of the mean squared error criterion used in making the key comparisons in this work. Finally, Chapter 9 outlines some additional developments of the procedures and some areas for further research.

Chapter 2: Ordinary Least Squares

Consider again the problem of predicting the value of a response variable y which is explained by one or more regressor variables according to the model

$$y_i = f(X_{1i}, X_{2i}, \ldots, X_{ki}) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where the errors ε_i are assumed to be iid with mean 0 and variance σ^2. The most common parametric approach to this problem is to express the model above as a (parametric) linear model, written in matrix notation as

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

where $\mathbf{y}$ is an n dimensional vector of responses, $\mathbf{X}$ is an $n\times(k+1)$ matrix of k regressor variables augmented with a column of ones, $\boldsymbol{\beta}$ is a $k+1$ dimensional vector of *unknown parameters*, and $\boldsymbol{\varepsilon}$ is an n dimensional vector of unknown errors. Note here that a regressor variable may be a function of other regressors, such as a polynomial term. The current work emphasizes the "single regressor model", having one explanatory variable X in the model, with all other possible terms defined to be polynomial expressions of this regressor ($X^2, X^3, \ldots$). The goal of this parametric approach is to obtain the estimate $\hat{\boldsymbol{\beta}}$ of the unknown $\boldsymbol{\beta}$ in order to achieve estimates of mean response $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$. A common technique for obtaining this estimate is that of ordinary least squares (OLS). OLS minimizes the sum of squared residuals ($\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$), which results in the estimate $\hat{\boldsymbol{\beta}}_{ols} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. Assuming that $\boldsymbol{\varepsilon}$ is $N(\mathbf{0}, \sigma^2\mathbf{I})$, where $\mathbf{I}$ is the $n\times n$ identity matrix, $\hat{\boldsymbol{\beta}}_{ols}$

is the optimal (uniform minimum variance unbiased (UMVU)) estimator of β. Thus, the fits at the data locations are obtained as

$$\hat{\mathbf{y}}_{ols} = \mathbf{X}\hat{\beta}_{ols} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}^{(ols)}\mathbf{y}, \tag{2.1}$$

where $\mathbf{H}^{(ols)}$ is the OLS "hat" matrix.

This hat matrix plays a crucial role in many inferences that are based on the OLS regression fit, and several of the properties of the OLS hat matrix are extended to the other procedures in this current work. Some of these properties of $\mathbf{H}^{(ols)} = (h_{ij}^{(ols)})$ are as follows:

(i) $-1 \le h_{ij}^{(ols)} \le 1$, (2.2)

(ii) $tr(\mathbf{H}^{(ols)}) = \sum_{i=1}^{n} h_{ii}^{(ols)} = p$, where tr = trace and $p = k+1$, (2.3)

(iii) $\sum_{j=1}^{n} h_{ij}^{(ols)} = 1$ for each i (row sums equal 1), (2.4)

(iv) $Var(\hat{\mathbf{y}}_{ols}) = Var(\mathbf{H}^{(ols)}\mathbf{y}) = \sigma^2\mathbf{H}^{(ols)}\mathbf{H}'^{(ols)} = \sigma^2\mathbf{H}^{(ols)}$ (2.5)
(since $\mathbf{H}^{(ols)}$ is symmetric and idempotent),

(v) residuals $\mathbf{e}^{(ols)} = \mathbf{y} - \hat{\mathbf{y}}_{ols} = (\mathbf{I} - \mathbf{H}^{(ols)})\mathbf{y}$, (2.6)

(vi) $\hat{\sigma}^2_{ols} = \dfrac{\sum e_i^{2\,(ols)}}{n-p} = \dfrac{\sum e_i^{2\,(ols)}}{tr[(\mathbf{I} - \mathbf{H}^{(ols)})(\mathbf{I} - \mathbf{H}^{(ols)})']}$. (2.7)

For the development and further discussion of these properties, see Myers (1990) and Hoaglin and Welsch (1978). Also, consider again equation (2.1), and notice that the fit for y_i at x_i can be expressed as

$$\hat{y}_i^{(\text{ols})} = \sum_{j=1}^{n} h_{ij}^{(\text{ols})} y_j \,. \tag{2.8}$$

Thus, the fitted value $\hat{y}_i$ is obtained as a weighted sum of the observations y_j, $j = 1, \ldots, n$, and an observation with a large h_{ij} has a significant influence on the fit. The value of each h_{ij} is directly related to the choice of model by the user. For instance, in simple linear regression,

$$h_{ij}^{(ols)} = \frac{1}{n} + \frac{(x_i - \bar{x})(x_j - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \,, \tag{2.9}$$

so $h_{ij}^{(\text{ols})}$ reflects the distance that x_j is from the mean $\bar{x}$. For fitting at x_i, data points at locations x_j far from the mean have heavy influence, while some data points relatively close to x_i may have almost no influence at all (especially if x_i is near $\bar{x}$).

With the weighting scheme being a direct consequence of the prescribed model, one should be very confident in the model before applying ordinary least squares to make inferences. If the model is chosen correctly, then OLS gives optimal results. However, if the model is incorrect, then OLS could give poor predictions at some data locations, and subsequent inferences could be very misleading. Consider simple linear regression again as an example. Suppose the true model is quadratic, but a linear model is specified. Predictions at locations where the quadratic structure is prevalent would then be poor, because most weight is given to observations at X locations far away from the point of prediction, with little information coming from the quadratic structure itself. A much better approach (when in doubt about the true model) is to use a weighting scheme that places more weight on observations close to the point of prediction rather than on observations far away. This is the idea behind nonparametric regression, to be discussed in Chapter 3. For a detailed discussion of OLS and other parametric regression techniques, such as maximum likelihood estimates, and the many extensions and applications of these techniques, see Myers (1990).

Chapter 3: Nonparametric Regression

3.A Introduction

This chapter contains a discussion of the procedures that attempt to solve the problem of obtaining fits (or predictions) of a response variable when no, or incomplete, information is available on the underlying model. With no such (parametric) information, these nonparametric procedures use only the data itself to provide these fits. Recalling that results considered in the current work are for the "single regressor" model, the problem of interest now is obtaining $\hat{y}$ for the model $y = f(X) + \varepsilon$, where f is some unknown function. As in the parametric case, the fitted value $\hat{y}$ is obtained via a weighted sum of the y observations. However, now there is a different rationale behind the choice of weights. The idea is as follows: if interested in predicting $f(x_0)$ at x_0 and if f is at least somewhat smooth, then the observations with the most information about $f(x_0)$ should be those located at points x_i closest to x_0. Thus, the weighting scheme used to assign weights to the y_i's is based on a decreasing function of the distances of their locations x_i from x_0. Points close to x_0 receive large weights, while points far from x_0 receive little or no weight. More details on this weighting scheme and the most popular techniques for achieving these weights are presented in the following sections.

3.B Kernel Regression

A widely used and thoroughly investigated nonparametric regression technique is that of kernel regression. Due to its computational simplicity and its straightforward extension to the multivariate case, this procedure is used extensively in the current research.

3.B.1 Procedure

The end result of kernel regression is to obtain the appropriate weights $h_{ij}^{(\mathrm{ker})}$ to give fitted values according to the expression

$$\hat{y}_i^{(\mathrm{ker})} = \sum_{j=1}^{n} h_{ij}^{(\mathrm{ker})} y_j , \tag{3.B.1}$$

which can also be thought of as $\hat{f}(x_i)$, for $i = 1, \ldots, n$. In matrix notation, equation (3.B.1) can be expressed as

$$\hat{\mathbf{y}}_{\mathrm{ker}} = \mathbf{H}^{(\mathrm{ker})}\mathbf{y}, \tag{3.B.2}$$

where $\mathbf{H}^{(\mathrm{ker})} = (h_{ij}^{(\mathrm{ker})})$ is denoted as the kernel "hat" matrix. A common method of obtaining the weights $h_{ij}^{(\mathrm{ker})}$ is that proposed by Nadaraya (1964) and Watson (1964), who defined

$$h_{ij}^{(\mathrm{ker})} = \frac{K\left(\frac{x_i - x_j}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h}\right)}, \tag{3.B.3}$$

where the function $K(\mathrm{u})$ is a *decreasing* function of $|\mathrm{u}|$, and $h > 0$ is the bandwidth (smoothing parameter). Further discussion of K and h follows in this and subsequent sections. The numerator of (3.B.3) satisfies the notion of giving more weight to observations at locations close to x_i (the location of the fit), and less weight to observations far away. The denominator is present to make the rows of $\mathbf{H}^{(\mathrm{ker})}$ sum to one, as with $\mathbf{H}^{(\mathrm{ols})}$. For obtaining a prediction $\hat{y}_0$ at a non-data point x_0, one simply replaces x_i with x_0 in (3.B.3) and calculates

$$\hat{y}_0^{(\text{ker})} = \sum_{j=1}^{n} h_{0j}^{(\text{ker})} y_j \,. \tag{3.B.4}$$

Obtaining such predictions over the entire range of the data would result in an estimated regression curve $\hat{f}$ which may provide some insight into the true form of the underlying function f. Unfortunately, no explicit closed form expression for f can be obtained from the kernel fit, or from other nonparametric techniques.

3.B.2 Kernel Functions

Kernel regression gets its name from the function $K(\mathrm{u})$ in (3.B.3), which is called the kernel function. A decreasing function of $|\mathrm{u}|$, $K(\mathrm{u})$ may be taken to be a probability density function (such as standard normal, where $K(\mathrm{u}) \propto \frac{1}{\sqrt{2\pi}} e^{-u^2}$), a function defined to be zero outside a certain range of u, or one of many other functional forms. Two typical forms of kernels used in the literature are

$$K(\mathrm{u}) = c(1-\mathrm{u}^2)^d,\ c,d > 0, \tag{3.B.5}$$

$$K(\mathrm{u}) = \frac{1}{1+c\mathrm{u}^2},\ c > 0. \tag{3.B.6}$$

It has been shown by several authors that for practical purposes the choice of the kernel function is not critical to the performance of kernel regression. Härdle (1990) illustrates this point for the general case of twice differentiable kernels, where his performance criterion is the mean integrated squared error (MISE) of the predicted function $\hat{f}(x)$. Minimizing the portion of MISE that is a function only of the kernel K, Gasser, Müller, and Mammitzsch (1985) found the "optimal" kernel to be the Epanechnikov kernel (Epanechnikov (1969)):

$$K(\mathrm{u}) = .75(1-\mathrm{u}^2)I(|\mathrm{u}| \le 1), \tag{3.B.7}$$

where I is the indicator function. From this result, Härdle then calculates the efficiencies of several other commonly used kernels. Efficiencies are calculated as the optimal MISE from the Epanechnikov kernel divided by the MISE for the particular kernel of interest. Results are in Table 3.B.1.

Table 3.B.1 Efficiencies of twice-differentiable kernels.

Kernel	$K(u)$	Efficiency
Epanechnikov	$(\frac{3}{4})(1-u^2)I(\lvert u\rvert \le 1)$	1
Quartic	$(\frac{15}{16})(1-u^2)^2 I(\lvert u\rvert \le 1)$	.995
Triangular	$(1-\lvert u\rvert)\, I(\lvert u\rvert \le 1)$	.989
Gauss	$(2\pi)^{-1/2}\exp(\frac{-u^2}{2})$	.961
Uniform	$(\frac{1}{2})I(\lvert u\rvert \le 1)$	.943

Based on these results, Härdle concludes that the choice of kernel function should be based on other considerations (besides MISE), such as computational efficiency. Due to this consideration and its similarity to a spline smoother matrix (discussed later in section 3.C.2), the kernel function employed in this current work is the simplified Normal (or Gauss) kernel given by

$$K(u) = e^{-u^2}. \qquad (3.B.8)$$

Even though the form of kernel chosen for kernel regression is not critical, the choice of bandwidth (h in equation 3.B.3) is crucial in obtaining a "good" kernel fit.

3.B.3 Bandwidth Choice

Recall that in the kernel regression weighting scheme, observations at locations x_i close to the point of prediction x_0 receive the most weight, with weights decreasing for observations as their distance from x_0 increases. How fast these weights decrease as the distance from x_0 increases is determined by the bandwidth h. This is turn controls the smoothness of the resulting estimate of f. For example, if h is very small (close to zero), then almost all of the weight is placed on the point of prediction itself, with the rest of the weight on only the closest (local) observations to this point. This would result in a fit that essentially "connects the dots", and is said to be undersmoothed, or overfit (with high variance). At the opposite extreme, if h is very large (close to the range of the x-values), then the weight is spread almost evenly throughout all of the observations. This would result in a fit that essentially takes the value $\bar{y}$ at each data point, i.e., fits the mean. This fit would be considered oversmoothed, or underfit (with high bias). The problem of choosing an appropriate bandwidth (smoothing parameter) is thus the crucial element in obtaining the proper kernel fit. By a proper or "good" fit, one usually means that it strikes the proper balance between the variance and the bias (or squared bias). This goal leads to minimization of a mean squared error criterion (or other global error criterion) as a logical starting point for determining what bandwidth to select for a given data set. Much research has been dedicated to this problem of bandwidth selection and numerous procedures have been developed. The next subsection gives an overview of some of the most popular of these techniques.

Summary of Techniques for Bandwidth Choice

The most popular and practical way to determine if a selected bandwidth is appropriate is to evaluate its performance based on some global error measure for the regression curve. As mentioned previously, this measure is often a form of mean (or average) squared error, which incorporates both bias and variance considerations. To begin a discussion of bandwidth choice, Härdle (1990) gives three widely accepted

quadratic error measures: average squared error (ASE), integrated squared error (ISE), and a conditional average squared error (CASE). These measures are as follows:

$$\text{ASE} = d_A(h) = n^{-1}\sum_{j=1}^{n}\left[\hat{f}_h(x_j) - f(x_j)\right]^2 w(x_j), \tag{3.B.9}$$

$$\text{ISE} = d_I(h) = \int\left[\hat{f}_h(x) - f(x)\right]^2 w(x)g(x)dx, \tag{3.B.10}$$

$$\text{CASE} = d_C(h) = E[d_A(h) \mid x_1, \ldots, x_n], \tag{3.B.11}$$

where $\hat{f}_h$ ($\hat{m}_h$ in Härdle (1990)) is the (bandwidth dependent) kernel estimate of f (m in Härdle (1990)), $g(x)$ is the density of the X's (equals 1 if X's are fixed), and $w(x)$ is a nonnegative weight function. This $w(x)$ is present to reduce boundary effects on rates of convergence, and was found by Härdle to not significantly influence the actual choice of h. In the current work, $w(x)$ is usually taken to be the constant value one for simplicity. Härdle gives a theorem showing the asymptotic equivalence (in convergence rates) of these distance measures. ASE is actually a discrete approximation to ISE, and in practice is much easier to compute. Based on these considerations, ASE (or "MSE") is emphasized as a performance criterion in Härdle (1990) and in the current research.

Härdle states that "the basic idea behind all smoothing parameter selection algorithms is to estimate the ASE or equivalent measures (up to some constant)", and hopefully the smoothing parameter that minimizes this estimate is also a good estimate for the smoothing parameter that minimizes the ASE itself. In taking this approach and expanding ASE, Härdle illustrates several important findings based on the portion of ASE that must be estimated to give $\hat{\text{ASE}}$ apart from a constant. This portion of ASE to be estimated is what serves as the criterion for choosing the bandwidth. A naive estimate was found to be the usual estimate of prediction error (denoted $p(h)$), involving the sum of squared errors:

$$p(h) = n^{-1}\text{SSE} = n^{-1}\sum_{j=1}^{n}\left[y_j - \hat{f}_h(x_j)\right]^2 w(x_j). \qquad (3.B.12)$$

However, n^{-1}SSE is a biased estimate of ASE (as shown by Härdle (1990)) and tends to overfit the data by choosing the smallest possible bandwidth. Härdle then presents three methods of finding an unbiased estimate of ASE (denoted MSE from this point on) in order to get a better selector of bandwidth.

Cross-Validation (PRESS)

The first of these methods is the "leave-one-out" method of *cross-validation* (Stone (1974)), which results in expectation of zero for the bias that parallels the bias from equation (3.B.12). Here, the fits $\hat{y}_i$ (= $\hat{f}_h(x_i)$) are obtained through the usual form of a weighted sum of the y_j's, but with observation y_i left out. Notationally, the "minus i" fit at x_i is given by $\hat{y}_{i,-i} = \sum_{j\neq i} h_{ij,-i}y_j$, where the $h_{ij,-i}$ are the weights formed ignoring observation y_i. One can express $h_{ij,-i}$ as $h_{ij}/(1-h_{ii})$, where h_{ij} comes from the hat matrix based on all observations (Myers (1990)). The expression to be minimized is given by the cross-validation function

$$\text{CV}(h) = n^{-1}\sum_{i=1}^{n}\left(y_i - \hat{y}_{i,-i}\right)^2 w(x_i) \qquad (3.B.13)$$

(Härdle (1990)). Suppressing the dependence on n and w, CV(h) is just the PRESS statistic used in a wide variety of regression procedures. This PRESS statistic (Allen (1974)) is as follows:

$$\text{PRESS} = \sum_{i=1}^{n}(y_i - \hat{y}_{i,-i})^2. \qquad (3.B.14)$$

Wong (1982) and Rice (1984a) give consistency results for the method of choosing bandwidth by cross-validation in the equispaced data situation. The cross-validation (PRESS) procedure is an attempt to resolve the overfitting problem of using the prediction error estimate *p(h)* in (3.B.12). While it does result in fits that are less dependent on individual observations (and thus have larger bandwidths), this PRESS procedure has been observed in a wide variety of applications to still tend to overfit the data by not choosing a bandwidth large enough. So a modified version of PRESS seems necessary. One such version is explained in the next subsection.

Penalizing Functions

The second method described by Härdle of choosing h based on MSE uses penalizing functions to adjust *p(h)* so that small values of h are less likely to be chosen. This penalizing of small h's is accomplished via a penalizing function $\Xi(u)$, which is increasing in u. The general idea is to adjust the (biased) prediction error *p(h)* of equation (3.B.12) by $\Xi[n^{-1}h^{-1}K(0)/\hat{g}_h(x_j)]$, where $\hat{g}_h(x_j)$ is the Rosenblatt-Parzen kernel density estimator of the (marginal) density of X at the value x_j, as described in Härdle (1990). For more details on the development of this Ξ, see Appendix A. In the current context, one can think of $\hat{g}_h(x_j)$ as the denominator of $h_{ij}^{(ker)}$ (equation 3.B.3), divided by h. This adjustment to *p(h)* results in the following *general penalized function* to minimize:

$$G(h) = n^{-1}\sum_{j=1}^{n}\left(y_j - \hat{f}_h(x_j)\right)^2 \Xi\left[n^{-1}h^{-1}K(0) / \hat{g}_h(x_j)\right] w(x_j). \qquad (3.B.15)$$

Härdle shows that the bias from *p(h)* alone is eliminated and that asymptotically $G(h)$ is roughly equal to ASE (up to a shift). With $\Xi(u)$ chosen to be increasing in u, it is clear that $G(h)$ penalizes values of h too low.

Many forms of Ξ are possible, with a typical choice being the simple $\Xi(u) = 1+2u$ of Shibata (1981). Working in the fixed design model (fixed x's, which eliminates the term $\hat{g}_h(x_j)$ in (3.B.15)), the general penalized function can be written as

$$G(h) = p(h)\ \Xi(n^{-1}h^{-1}).$$

Rice studied the behavior of five forms of $G(h)$ based on five prevalent forms of $\Xi(n^{-1}h^{-1})$: (i) Generalized Cross-validation Ξ_{GCV}, (ii) Akaike's Information Criterion Ξ_{AIC}, (iii) Finite Prediction Error Ξ_{FPE}, (iv) Shibata's Ξ_S, and (v) Rice's own bandwidth selector $\Xi_T(n^{-1}h^{-1})$ $= [1-2n^{-1}h^{-1}K(0)]^{-1}$ (for formulas and details, see Rice (1984a)). In his paper, Rice developed an asymptotically optimal bandwidth selector that is an unbiased estimate of the risk function (expected squared error loss), and then showed that all five of these penalized selectors are asymptotically equivalent to this optimal selector. The important result from Rice for the current work, however, is that despite asymptotic equivalence, the selectors behave differently for finite data simulations. In comparing the five penalized functions above, along with the cross-validation selector of (3.B.13) and the selector based on unbiased risk estimation, Rice found that his T selector and cross-validation performed best, followed by GCV and the unbiased risk estimator. The AIC, FPE, and S penalized selectors performed poorly. (The performance measure used was efficiency relative to the optimal risk function). The conclusion was that selectors that penalize small bandwidths (i.e., penalize undersmoothing (overfitting)) perform better in general. Härdle (1990) further studied the performances of the five penalized functions with other simulated data. He found that Rice's T selector did indeed work well in cases where protection against undersmoothing was desirable (i.e., needed reduction of variance), but did not perform well when protection against oversmoothing was desirable (i.e., needed reduction of bias). Härdle found the Generalized Cross-validation (GCV) selector to give the best overall performance. So Rice's general conclusion is probably too broad a statement, and more work needs to be done to find a selection function that performs well for protection against both bias and variance.

Rice does argue, however, that when considering mean squared error, there is much less chance of encountering oversmoothing problems when choosing the bandwidth. His argument is supported by Chiu (1990), who uses Fourier analysis and the sample variation in bandwidth estimates to show why often in simulation studies most bandwidth selectors are biased toward undersmoothing. When deciding on a selection procedure, one should be sure to take this point into consideration. All of these results for finite sample cases have by no means been rigorously proven and established in the general statistical setting, but these findings provide valuable information for what types of bandwidth selectors to use in specific situations. The ideas here are used directly in forming the bandwidth selectors used in the current research (as described in the upcoming subsections entitled *PRESS** and *PRESS***).

Plug-in Method and Asymptotic Results

The third of Härdle's methods for obtaining an unbiased estimate of MSE in selecting h is what he calls the "plug-in" procedure. This procedure is based on the asymptotic expansion of MSE, and the optimal estimate of h involves unknowns, including σ^2 and the second derivative of the underlying function f, and is proportional to $n^{-1/5}$. Estimates from some preliminary smoothing process are "plugged in" for the unknowns to give the estimate of h. Based on its strictly asymptotic nature and the additional complication of estimating extra unknowns, this plug-in method is not appropriate for the current research, which deals with smaller samples and simpler procedures. Also, the plug-in method restricts the user to a certain smoothness class for the regression function f (to twice differentiable functions in the case above). Under certain assumptions on f, the range of h, and the marginal density of X, Härdle and Marron (1985) have shown that the Cross-validation function CV(h) (equation (3.B.13)) and the General penalized function G(h) (equation (3.B.15)) themselves choose bandwidths that are *asymptotically optimal* (in the sense that these functions approximate (up to a constant) the measure MSE ($d_A(h)$ of (3.B.9))) uniformly over h. In addition, these results of optimal bandwidths hold

uniformly over smoothness classes. This independence from the "smoothness" (degree of differentiability) of the true f is very advantageous in the practical setting, and is not shared by the plug-in method. And finally, Härdle, Hall, and Marron (1988) prove two convergence rate theorems that show that even if one knew the "unknowns" in the plug-in method, this method would still have a rate of convergence no better than that of cross-validation or penalized functions.

Even though these asymptotic results are not used explicitly in this current work, there are a few other important asymptotic results mentioned here that are useful in illustrating some of the points made in subsequent chapters. The usual type of asymptotics in kernel regression is to study the behavior of estimators as $n \to \infty$, with $h \to 0$ and $nh \to \infty$. These conditions ensure that as the sample size increases, a smaller range of values around the point of prediction results, but with an increasing number of observations in this range of values. The relevant results given here are bias and variance formulas, explained in Chu and Marron (1991), for the Nadaraya-Watson estimate of (3.B.3). For the technical assumptions underlying these expressions, see Appendix B. In the fixed design case (fixed x's), and under the first three technical assumptions,

$$\text{Bias}(\hat{f}(x)) = \int h^{-1} K\left(\frac{x-t}{h}\right)\left(f(t) - f(x)\right)dt + O(n^{-1}), \qquad (3.B.16)$$

$$\text{Var}(\hat{f}(x)) = n^{-1}h^{-1}\sigma^2 \int K^2(u)du + O(n^{-2}h^{-2}), \qquad (3.B.17)$$

with proofs in Chu (1989). The O notation for the higher order terms is a measure of the *order of magnitude* of these terms. Simply stated, $a_n = O(b_n)$ means that the sequence $\{a_n\}$ is of roughly the same size or order of magnitude as the sequence $\{b_n\}$. More formally, $a_n = O(b_n)$ if the ratio $|a_n/b_n|$ is bounded for large n (see Bishop, Fienberg, and Holland (1975) for more details). Intuitively, one may think of the higher order terms of (3.B.16) being $O(n^{-1})$ as saying that all of these terms have denominators with terms at least as large as n.

Shifting to the random design case, with the two additional assumptions given in Appendix B, the bias and variance expressions are given by

$$\text{Bias}(\hat{f}(x)) = \frac{\int K\left(\frac{x-t}{h}\right)\left(f(t)-f(x)\right)g(t)dt}{\int K\left(\frac{x-t}{h}\right)g(t)dt} + O(n^{-1/2}h^{-1/2}), \qquad (3.B.18)$$

$$\text{Var}(\hat{f}(x)) = n^{-1}h^{-1}g(x)\,\sigma^2\int K^2(u)du + o(n^{-1}h^{-1}), \qquad (3.B.19)$$

where $g(x)$ is the marginal density of X (proofs in Chu (1989)). The o notation is another measure of order of magnitude, and $a_n = o(b_n)$ means that the sequence $\{a_n\}$ is of a smaller order of magnitude than is $\{b_n\}$, or that the ratio $|a_n/b_n|$ converges to zero (Bishop *et al* (1975)). Chu and Marron (1991) also show that the bias expression of (3.B.18) can be expanded by use of Taylor's Theorem to give

$$\text{Bias}(\hat{f}(x)) = h^2\left(f''(x)g(x)+2f'(x)g'(x)\right)\left(\int u^2K(u)du\right)/\left(2g(x)\right) + O(n^{-1/2}h^{-1/2}) + o(h^2). \qquad (3.B.20)$$

Equations (3.B.19) and (3.B.20) are the general (asymptotic equations) for bias and variance, and are referenced in later discussions. For instance, one can clearly see that increasing the bandwidth h (oversmoothing) increases the bias in (3.B.20), while decreasing h (undersmoothing) increases the variance in (3.B.19). By adding the squared bias from (3.B.20) and the variance from (3.B.19), one can obtain the (asymptotic) MSE as:

$$\text{MSE}(\hat{f}(x)) \sim \nu n^{-1}h^{-1} + b^2h^4, \qquad (3.B.21)$$

where $\sim$ means the ratio approaches one in the limit, and where ν and b are constants based on the bias and variance expressions. Minimization of this MSE gives the

asymptotically optimal bandwidth $h_{AOPT} = (\nu/4b^2n)^{1/5}$, and establishes the widely referenced "optimal" bandwidth expression $h_{opt} \propto n^{-1/5}$ (often just taken as $h_{opt} = n^{-1/5}$).

Bootstrapping

One other technique that has been investigated for choosing a bandwidth is that of bootstrapping, a procedure developed for constructing sampling distributions empirically from the data at hand. To obtain a "bootstrap sample" from an original sample of size n, one draws many (B) samples, each of size n, with replacement, from the original sample. Now, suppose θ is the parameter of interest in a certain problem, and that one can obtain an estimate $\hat{\theta}$ of θ from the original sample. Then one can also obtain an estimate θ^* of θ from each of the B bootstrap samples. The idea behind bootstrapping (the "bootstrap principle") is that the observed distribution of the θ^*'s approximates the true distribution of $\hat{\theta}$. Thus, this distribution of the θ^*'s can be used to gain insight about the true behavior of the estimate $\hat{\theta}$. (See Stine (1989) for a brief discussion, including the choice of resampling from residuals or from the actual data).

Faraway (1990) uses bootstrapping as a method of choosing the bandwidth in kernel regression. Here a bootstrap sample of fits $\hat{f}_j^*(x)$, $j = 1, \ldots, B$, (based on resampling residuals from an initial fit), is used to estimate the mean squared error, and the h which minimizes this estimated MSE is used for the final kernel fit. In this procedure, Faraway describes how to form the estimate of MSE so that it is consistent with the true MSE. Due to its highly computer intensive nature, this approach to bandwidth selection does not fit well into the general framework of the current research, and hence are not studied further at this time. More details can be found in Faraway (1990). Bootstrapping has proven most applicable in obtaining standard errors and confidence intervals for estimates, and in the exploration of estimator performance with real data, where the parameters are unknown (as opposed to Monte Carlo simulations, which have artificial data with known parameters). These topics are discussed briefly in later sections.

*PRESS**

In the current research, two of the above philosophies of bandwidth choice are combined to give a selection criterion. The first component of this selector is the cross-validation quantity PRESS. As discussed in the previous sections, PRESS attempts to overcome the overfitting problem of choosing h too small by giving a fit less dependent on individual observations. However, PRESS has often been observed to still select a bandwidth smaller than that desired. The scope of this problem lends itself naturally to the application of penalized functions. These were discussed earlier in the context of penalizing the prediction error $p(h) = n^{-1}$SSE against small bandwidths. Recall that studies by Rice, Härdle, and Chiu discovered no uniformly "best" form of penalizing function. Härdle did find, however, that (penalized) Generalized Cross-validation appeared to perform well over a wide range of smoothness problems, and Rice found the usual cross-validation criterion (along with his T criterion) to perform extremely well in cases where protection against overfitting was desired. Based on these findings and the arguments by Rice and Chiu that the overfitting problem occurs much more frequently than the underfitting problem, a bandwidth selector candidate for the current research is taken to be a "penalized PRESS" diagnostic, denoted PRESS*.

Developed by Einsporn (1987), PRESS* penalizes PRESS instead of the usual prediction error. This merging of procedures maintains the versatility of cross-validation, while also introducing extra protection against overfitting. The "penalty" in PRESS* for small bandwidths comes from dividing PRESS by $[n - \text{tr}(\mathbf{H}^{(\text{ker})})]$. To see how this penalty works, consider fitting at x_i. As the bandwidth gets smaller, the individual weights on x_i (the $h_{ii}^{(\text{ker})}$'s) get larger, and thus $\text{tr}(\mathbf{H}^{(\text{ker})})$ gets larger. The denominator $[n - \text{tr}(\mathbf{H}^{(\text{ker})})]$ gets smaller, and thus penalizes (increases PRESS*) for small bandwidths. Also, the term $\text{tr}(\mathbf{H}^{(\text{ker})})$ by itself may be thought of as a measure of model adequacy. Related to OLS, where $\text{tr}(\mathbf{H}^{(\text{ols})}) = p$ (the number of parameters that need to be estimated), $\text{tr}(\mathbf{H}^{(\text{ker})})$ can be thought of as the "equivalent" model degrees of freedom for kernel regression (Cleveland (1979)). This quantity can be interpreted loosely as the number of parameters that would

be needed to obtain a comparable parametric fit. Thus, it is desired to have $\text{tr}(\mathbf{H}^{(\text{ker})})$ as small as possible to reflect a fit that is not overly complex (or variable). The form of penalty function in PRESS* is slightly different from those mentioned earlier, and is introduced in the hope of giving more consistent results for different levels of smoothing problems. Einsporn reported that in preliminary simulation studies, PRESS* led to improved kernel predictions compared to the cross-validation method. A more detailed study of PRESS* has been carried out in the current work, and results are given later in Chapters 7 and 8. It is shown that PRESS*, while it does correctly provide protection against small bandwidths when needed, often selects bandwidths that are much too large (especially for the proposed model-robust procedures). A possible solution to this problem is the introduction into PRESS* of a penalty for large bandwidths. This would provide a penalty for bias to join the existing penalty for variance. This modification of PRESS* leads to a second candidate, PRESS**, for determining the bandwidth.

*PRESS***

The basic idea behind PRESS** is to introduce a penalty for large bandwidths that is comparable to the penalty $[n-\text{tr}(\mathbf{H}^{(\text{ker})})]$ for small bandwidths, which is already present in the denominator of PRESS*. (The new penalty term will also appear in the denominator). Noting that $[n - \text{tr}(\mathbf{H}^{(\text{ker})})] \to 0$ as $h \to 0$, and $[n - \text{tr}(\mathbf{H}^{(\text{ker})})] \to n - 1$ as $h \to \overset{\infty}{1}$, it is desired to have the new penalty term approach 0 as $h \to \overset{\infty}{1}$ and approach $n - 1$ as $h \to 0$. This new penalty term will be comprised of sums of squares error (SSE) terms. It is known that SSE increases as h increases, and SSE is maximized when $h = 1$ (when "fitting the mean"). Also, SSE $\to 0$ as $h \to 0$. Letting SSE_{mean} = SSE with $h = 1$, and SSE_h = SSE at any candidate h, it is then clear that the expression $\frac{\text{SSE}_{\text{mean}} - \text{SSE}_h}{\text{SSE}_{\text{mean}}}$ is between 0 and 1. This expression approaches 0 for $h \to 1$ and approaches 1 for $h \to 0$. Multiplying this expression by $(n - 1)$ then gives a penalty term that approaches 0 for $h \to 1$ and

approaches $n - 1$ for $h \to 0$. This was the penalty structure desired, and PRESS** for choosing h for kernel regression is expressed as

$$\text{PRESS}^{**} = \frac{\text{PRESS}}{n - \text{tr}(\mathbf{H}^{(\text{ker})}) + (n-1)\dfrac{\text{SSE}_{\text{mean}} - \text{SSE}_h}{\text{SSE}_{\text{mean}}}} \; .$$

In general, for selecting a parameter θ for any procedure with hat matrix $\mathbf{H}$, and defining SSE_{max} to be the maximum sum of squares error across all θ values, PRESS** may be defined as

$$\text{PRESS}^{**} = \frac{\text{PRESS}}{n - \text{tr}(\mathbf{H}) + (n-1)\dfrac{\text{SSE}_{\text{max}} - \text{SSE}_\theta}{\text{SSE}_{\text{max}}}} \; . \qquad (3.B.22)$$

The performance of PRESS** is analyzed (and compared with PRESS*) in Chapters 7 and 8.

3.B.4 Variations of Kernel Regression

The bulk of the current work involves applications of the kernel techniques described in the previous subsection. These techniques were chosen based on considerations such as simplicity, popularity, versatility, and observed performance, but are by no means the only versions of kernel regression available. Some of these other variations are included here for completeness and to possibly suggest improvements for future research.

Priestley-Chao, Gasser-Müller Estimates

Considered first are two techniques of obtaining the kernel weights $h_{ij}^{(\text{ker})}$ in equation (3.B.1), instead of using the Nadaraya-Watson weights of (3.B.3). For the fixed

design case with nearly equispaced x_i's on [0,1], Priestley and Chao (1972) proposed using the following weights for predicting at the point x_0:

$$h_{0j}^{(\text{ker,PC})} = n(x_j - x_{j-1})\left(\frac{1}{h}\right)K\left(\frac{x_0 - x_j}{h}\right), \; j = 1, \ldots, n. \qquad \text{(3.B.23)}$$

These Priestley-Chao weights can be interpreted in terms of the Nadaraya-Watson weights by replacing the denominator in (3.B.3) with $[n(x_j - x_{j-1})]^{-1}$ for $x_0 \in (x_{j-1}, x_j)$. The Priestley-Chao estimate is based on restrictive assumptions and performs poorly in many situations. Gasser and Müller (1979) proposed an improvement to this estimate by defining weights

$$h_{0j}^{(\text{ker,GM})} = n\int_{s_{i-1}}^{s_i} \frac{1}{h} K\left(\frac{x_0 - u}{h}\right) du, \qquad \text{(3.B.24)}$$

where $x_{i-1} \le s_{i-1} \le x_i$ is chosen between the ordered x-data. The Gasser-Müller estimate is studied by Chu and Marron (1991) as a "convolution" of a kernel function with some function representing the raw data. The idea here is to construct a histogram with the i^{th} bin ($I = 1, \ldots, n$) centered at x_i and having height y_i, and then to obtain the prediction at x_0 by the convolution (continuous moving average) of this histogram (a step function) with the kernel function. Centered bins are obtained through defining $s_j = \frac{1}{2}(x_j + x_{j+1})$. This convolution estimator performs well in the case of nonuniform x's, but is much more difficult to compute than previous estimators, especially if considering the extension to the multivariate case.

Variable Bandwidth Selectors

The discussion about selecting the bandwidth h so far has dealt exclusively with "global" procedures. In other words, a single bandwidth is chosen and used throughout the entire data set. However, certain data sets may behave in such a way that varying the

choice of bandwidth at different X locations may prove beneficial. A bandwidth that is optimal in one region may be too small or too large to perform adequately in another region. This need for a locally adaptive bandwidth selector is most noticeable in extremely nonuniform (or "misbehaved") data, where there may be gaps between x's, regions of clumps of observations or sparse observations, or extreme change in local curvature when moving from one location to another. Several approaches in the literature for choosing local bandwidths are mentioned here. One attempt at adapting to varying local densities of the x's is to use *k-nearest neighbor* (*k-NN*) estimates. For a discussion of *k-NN* regression (with references), see Härdle (1990) or Altman (1992). The idea here is to apply a weight function (which could be a form of a kernel) that only assigns weight to the k observations that are closest in location to the point of prediction. So, if one thinks of the spread of the weight function as being indexed by a bandwidth h, then areas of high density of x's would result in small h's, whereas areas of low density of x's would result in large h's. Cleveland (1979) uses the idea of nearest neighbor regression in his robust regression procedure, to be discussed in the next section. In actuality, *k-NN* regression and kernel regression are two separate procedures, but conceptually they are performing the same task. For instance, choosing h for kernel regression is directly related to choosing k for *k-NN* regression. The two procedures do not behave differently enough to warrant further discussion or comparison here (more details can be found in Härdle (1990)).

Müller and Stadtmüller (1987) suggest an approach that adapts the choice of h to local curvature of the data. This procedure involves estimating derivatives of the underlying function f at each X location, and the size of bandwidth chosen decreases as the estimated curvature increases. A third procedure, which is gaining prominence as a locally adaptive smoother, is the use of bootstrapping at individual x locations. Härdle and Bowman (1988) develop results that use the bootstrap to acquire an approximation to a distribution of a kernel estimator, and then use this bootstrap distribution to obtain an estimate of local mean squared error at each data point. Local bandwidths can be chosen

to minimize these local MSE estimates. Faraway (1990) also briefly mentions local smoothing based on the bootstrap, and makes the additional point that there may be some irregularity in the local bandwidths, especially if the number of bootstrap samples B is not sufficiently large. He suggests smoothing the bandwidths (with an initial global bandwidth determined by all of the data) to give local bandwidths that are somewhat smooth as a function of X. Local bandwidth selection has been shown, mainly through squared errors of the estimates, to provide some improvement over global bandwidths. However, this improvement comes jointly with a significant increase in computations. In the spirit of simplicity, the current research employs the more straightforward global bandwidth procedure. At this point, the initial comparisons of several procedures are being carried out, and more computationally advanced "improvements" to these procedures may be studied in the future. Additionally, these improvements should benefit each of the procedures similarly, and the basic results of the comparisons to come (in later chapters) should not significantly change.

Robust Kernel Regression

One other addition to the basic kernel procedure could be protection against outliers among the y's. This robustness problem is addressed by Cleveland (1979) in his article on locally weighted regression. Here he uses an iterative reweighting procedure in which the usual weights h_{ij} are downweighted for points that have large residuals e_i, $i = 1, \ldots, n$, from the previous iteration. Cleveland uses the robust weighting function of the bisquare to perform the downweighting:

$$\begin{aligned} B(u) &= (1 - u^2)^2 \text{ , for } |u| < 1 \\ &= 0 \quad \text{ , for } |u| \geq 1 \text{ ,} \end{aligned}$$

with u replaced by $e_j/6s$ for the j^{th} downweighting value, where s is the median of the $|e_i|$. (Coping with outliers is not addressed in the current research).

Boundary Adjustments

One of the biggest problems inherent in kernel regression is predicting at the boundaries of the data. As x approaches the boundary points, the kernel weights become asymmetric, and bias and variance can be affected. As an illustration, consider trying to obtain the prediction $\hat{y}_0$ at the point x_0 at the right boundary of the data. The only points available (other than x_0 itself) to receive kernel weights are those points to the left of x_0. Now, if the data (and the true function f) are increasing toward the right boundary, then all y-values in the weighted sum used to obtain $\hat{y}_0$ are less than or equal to the value y_0 at x_0. Thus, the prediction $\hat{y}_0$ will be too low, due to being biased at the boundary.

Several techniques have been developed that attempt to handle this bias problem. Rice (1984b) presents a rather straightforward approach that involves adjustments to ensure that the bias and variance near the boundaries are of the same order of magnitude as in the interior. Rice's modified estimate at the boundaries can be expressed as follows:

$$\hat{f}^{(R)}(x) = \hat{f}_h(x) + \beta\left[\hat{f}_h(x) - \hat{f}_{\alpha h}(x)\right], \qquad (3.B.25)$$

where $\hat{f}_h$ and $\hat{f}_{\alpha h}$ are kernel estimators with bandwidths h and αh, respectively, and α and β are constants. Rice gives expressions for α and β that achieve the desired bias and variance properties. The expression in (3.B.24) can be written as a linear combination of the kernel estimates in the form

$$\hat{f}^{(R)}(x) = (1 - R)\hat{f}_h(x) + R\hat{f}_{\alpha h}(x), \qquad (3.B.26)$$

where $R = -\beta$. The estimate in (3.B.25) is a (generalized) *jackknife estimator*, and is discussed in this context by Härdle (1990). Other techniques that have been proposed to solve boundary problems include use of modified boundary kernels (Gasser and Müller

(1979)) and reflection methods (Hall and Wehrly (1991)). Details of these methods are left out here, because in the current research the method applied first for handling boundary problems is *local polynomial regression*. This is another nonparametric regression technique that is gaining prominence in the recent literature, and is discussed in more detail in the next section.

3.C Other Nonparametric Methods

The kernel method is but one of several widely applicable nonparametric procedures of fitting a curve to a set of data. Due to simplicity and a straightforward extension to the multivariate case, kernel regression has received the bulk of the attention in the background information so far. However, due to boundary bias problems, kernel regression is not used as the primary nonparametric fitting technique in the final form of the model-robust procedures to be developed in this paper. Instead, local linear regression is implemented in order to overcome boundary bias and a few other drawbacks of kernel regression. The general procedure of local polynomial regression is described in detail in this section. For completeness, one of the biggest "competitors" of kernel regression--spline regression--is also described, but is not used.

3.C.1 Local Polynomial Regression

All of the nonparametric techniques discussed thus far for fitting y_i at a point x_i obtain the fits based on a weighted sum of the n observations:

$$\hat{y}_i = \hat{f}(x_i) = \sum_{j=1}^{n} h_{ij} y_j \ ,$$

where observations y_j at locations close to x_i are given the largest weights. Unfortunately, this simple weighting scheme has several drawbacks. The first of these is boundary problems, as discussed in the previous section. The second flaw is that bias and variance in the interior may also be inflated if the x's are nonuniform or if there is substantial curvature present in the underlying regression function. Additionally, these problems

become worse in the multidimensional case. One rather successful approach to solving these problems is *local polynomial regression*, introduced by Cleveland (1979). This technique obtains the fitted value $\hat{y}_i$ as the fitted value of a d^{th} degree polynomial fit to the data using a weighted least squares regression, where the weights w_{ij} are assigned to each observation based on an initial kernel fit to the data.

Before describing local polynomial regression in more detail, a general, and simplified, overview of weighted least squares is given. Suppose one is fitting a d^{th} order polynomial model $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \varepsilon_i$, or $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ in matrix notation. Recall that the OLS estimate would be

$$\hat{\mathbf{y}}_{\text{ols}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{ols}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}^{(\text{ols})}\mathbf{y}, \text{ or } \hat{y}_i^{(\text{ols})} = \sum_{j=1}^{n} h_{ij}^{(\text{ols})} y_j .$$

Here the prediction weights h_{ij} are functions strictly of the regressor values. Now suppose we also want the prediction weights to reflect some phenomenon present in the y's, such as heterogeneous variances. With unequal variances, for instance, one would want to place more weight on y-values with small variances and less weight on those with large variances. To represent this additional weighting, one introduces weights w_{ij} into the weighted sum to obtain weighted least squares (WLS) fits

$$\hat{y}_i^{(\text{wls})} = \sum_{j=1}^{n} w_{ij} h_{ij}^{(\text{ols})} y_j , \tag{3.C.1}$$

$$\text{or } \hat{y}_i^{(\text{wls})} = \mathbf{x_i}'(\mathbf{X}'\mathbf{W}(x_i)\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}(x_i)\mathbf{y}, \; i = 1, \ldots, n , \tag{3.C.2}$$

where $\mathbf{x_i}'$ is the i_{th} row of the $\mathbf{X}$ matrix, and $\mathbf{W}(x_i)$ is an $n \times n$ diagonal matrix of the weights w_{ij}. For more discussion on weighted least squares, see Myers (1990).

In local polynomial regression (LPR), the weights w_{ij} described above come from kernel weights from an initial fit to the data. To see exactly how LPR works, consider fitting y_i at the point x_i. First, a kernel fit may be obtained for the entire data set, resulting in the kernel hat matrix $\mathbf{H}^{(\text{ker})}$, which can be written as

$$\mathbf{H}^{(\text{ker})} = \begin{bmatrix} \mathbf{h}_1'^{(\text{ker})} \\ \mathbf{h}_2'^{(\text{ker})} \\ \vdots \\ \mathbf{h}_n'^{(\text{ker})} \end{bmatrix}, \qquad (3.C.3)$$

where $\mathbf{h}_i'^{(\text{ker})}$ is the i_{th} row of $\mathbf{H}^{(\text{ker})}$. Recall that to obtain the fitted value for y_i at location x_i using just this kernel hat matrix, one would find

$$\hat{y}_i^{(\text{ker})} = \sum_{j=1}^{n} h_{ij}^{(\text{ker})} y_j = \mathbf{h}_i'^{(\text{ker})} \mathbf{y} ,$$

where the $h_{ij}^{(\text{ker})}$, $j = 1, \ldots, n$, are the n elements of the i^{th} row of $\mathbf{H}^{(\text{ker})}$. Also recall that $h_{ij}^{(\text{ker})}$ gives weight to y_j based on its distance from x_i. These $h_{ij}^{(\text{ker})}$, for fixed i, serve as the weights w_{ij} in weighted least squares. Notice also that the $h_{ij}^{(\text{ker})}$'s (w_{ij}'s) differ for different i's. As stated earlier, the idea behind local polynomial regression is to obtain the fit at x_i as the fitted value of a d^{th} order polynomial fit to the observations close to x_i using weighted least squares regression. Defining the weights for WLS as the elements of the i^{th} row of $\mathbf{H}^{(\text{ker})}$, one obtains $w_{ij} = h_{ij}^{(\text{ker})}$. Thus, the WLS *diagonal* weight matrix for local polynomial regression (LPR), for fitting at x_i, is given by

$$\mathbf{W}^{\text{LPR}}(x_i) = diag(\mathbf{h}_i^{(\text{ker})}) = \begin{bmatrix} h_{i1}^{(\text{ker})} & & & 0 \\ & h_{i2}^{(\text{ker})} & & \\ & & \ddots & \\ 0 & & & h_{in}^{(\text{ker})} \end{bmatrix} = (w_{ij}) . \qquad (3.C.4)$$

The estimated coefficients for the local polynomial regression fit at x_i are then given by

$$\hat{\boldsymbol{\beta}}_i^{(\text{LPR})} = (\mathbf{X}'\mathbf{W}^{\text{LPR}}(x_i)\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{\text{LPR}}(x_i)\mathbf{y}, \qquad (3.C.5)$$

and thus the fit at x_i is obtained as

$$\hat{y}_i^{(\text{LPR})} = \mathbf{x}_i'\hat{\boldsymbol{\beta}}_i^{(\text{LPR})} = \mathbf{x}_i'(\mathbf{X}'\mathbf{W}^{\text{LPR}}(x_i)\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{\text{LPR}}(x_i)\mathbf{y} = \mathbf{h}_i'^{(\text{LPR})}\mathbf{y} . \qquad (3.C.6)$$

In matrix notation, the n fitted values can be expressed as $\hat{\mathbf{y}}_{LPR} = \mathbf{H}^{(LPR)}\mathbf{y}$, where

$$\mathbf{H}^{(LPR)} = \begin{bmatrix} \mathbf{h}_1'^{(LPR)} \\ \mathbf{h}_2'^{(LPR)} \\ \vdots \\ \mathbf{h}_n'^{(LPR)} \end{bmatrix}. \qquad (3.C.7)$$

Cleveland (1979) and Hastie and Loader (1993) present this development with further discussion, and Stone (1980, 1982) shows optimal convergence rates for LPR in a certain minimax sense. Discussed in these papers is the proper choice of the order d of the local polynomial. For the majority of cases, a first order fit (*local linear regression* (*LLR*)) is an adequate choice. Cleveland notes that LLR strikes a good balance between computational ease and the flexibility to reproduce patterns in the data (i.e., reduce bias). Fan (1992) presents asymptotic optimality properties and advantageous small sample properties via simulations for LLR. However, in those cases where sharp curvature is present in the data, LLR may fail to capture peaks and valleys in the data structure, and *local quadratic regression* (*LQR*) may be needed to provide an adequate fit. Unfortunately, increasing the order d of the local polynomial increases the variance of the estimate. All authors agree that in practical applications, there is usually no need for polynomials of order $d > 2$. In a given situation, the choice of $d = 1$ or $d = 2$ should be made by the user to strike the proper balance between bias and variance.

Although presented here as a separate procedure, kernel regression is actually just a special case of local polynomial regression, namely that of taking the local polynomial model to be a single parameter "location model". In this case, $\mathbf{X}$ in equation (3.C.6) is just $(1, \ldots, 1)'$, and it can easily be shown that (3.C.6) simplifies to (3.B.1) with weights $h_{ij}^{(ker)}$ given by the Nadaraya-Watson weights of (3.B.3). Thus, local polynomial regression can be thought of as taking kernel regression (which locally fits a location model at each point) and extending it to using local fits of higher dimension at each point. The general

consensus is that, in routine cases, local polynomial regression (in particular, LLR) tends to perform as well or better than the basic kernel procedure. LPR simultaneously addresses the problems of boundary bias and nonuniform x's, and is easily extended to the multivariate setting. The only noticeable drawback of LPR is increased variance of the fits, especially at the boundaries. So once again, the choice of estimator depends on both bias and variance considerations. Preliminary results in this research have shown that local linear regression generally outperforms kernel regression for the procedures to be developed here.

3.C.2 Spline Regression

Spline regression is another widely used nonparametric fitting procedure. For a thorough review of this procedure, see Eubank (1988) or Silverman (1985). The spline regression estimate is defined to be the function $\hat{g}$ that minimizes

$$S(g) = \sum_{i=1}^{n} \left(y_i - g(x_i)\right)^2 + \delta \int \left(g''(x)\right)^2 dx \,, \qquad (3.C.8)$$

where δ denotes a smoothing parameter. This δ controls the trade-off between the goal to produce a good fit to the data (first term in (3.C.8)) and the desire to produce a curve without too much rapid local variation (second term in (3.C.8)). The second term in (3.C.8) can be thought of as a "roughness penalty" since the integral would be large for a function g that fluctuates rapidly. The solution of minimizing (3.C.8) over all twice differentiable functions yields the solution $\hat{g}(x)$, which is a *cubic spline*. This cubic spline can be shown (Reinsch (1967)) to possess the following properties:

(i) $\hat{g}(x)$ is a cubic polynomial between two successive x-values;

(ii) at the data points x_i, the curve $\hat{g}(x)$ and its first two derivatives are continuous, but there may be a discontinuity in the third derivative;

(iii) $\hat{g}(x)$ is linear ($\hat{g}''(x) = 0$) outside of the range of the data.

Although computational schemes are available for finding $\hat{g}(x)$, these schemes are much more computationally intensive than kernel estimates, mainly due to the definition of $\hat{g}(x)$ as the minimizer of a functional form.

Related to this increased complexity, Silverman (1984) has shown a close relation between spline regression and local bandwidth kernel regression. Silverman shows that the spline estimate for predicting at x_0 is nearly equivalent to using local bandwidth kernel regression with weights

$$h_{oj}^{(S)} = n^{-1}h(x_o)^{-1}\frac{1}{f_X(x_o)}K_S\left(\frac{x_o - x_j}{h(x_o)}\right), \qquad (3.C.9)$$

where $f_X(x_0)$ is the marginal density of X at x_0, $h(x_0)$ is the (local) bandwidth at x_0 given by

$$h(x_o) = \delta^{1/4}n^{-1/4}f_X(x_o)^{-1/4}, \qquad (3.C.10)$$

and K_S is the "effective" kernel function given by

$$K_S(u) = \tfrac{1}{2}\exp\left(\frac{-|u|}{\sqrt{2}}\right)\sin\left(\frac{|u|}{\sqrt{2}} + \frac{\pi}{4}\right). \qquad (3.C.11)$$

The relationship between the spline estimate and its "effective" (or "equivalent") kernel estimate has been studied in some detail (Messer (1991)), and often authors develop new procedures using kernel regression for simplicity, noting that there is a straightforward extension for those who would rather use splines. This is the approach being taken in the current research, with the additional note that kernel methods are more easily extended to the multidimensional case than are splines.

3.D Nonparametric or Parametric?

With such an abundance of parametric and nonparametric fitting procedures available, it may be a task in itself for one to choose which procedure to use. In two situations this choice is easy. If the user knows the parametric form of the underlying function f, then a parametric procedure should be used. OLS, described in Chapter 2, gives optimal (UMVU) results in this case. At the other extreme, if the user has no idea about the true form of f, then a nonparametric procedure should be used. Several of these techniques have been described in this chapter, with recommendations for kernel or local linear regression. The trouble with method selection arises when the user has some idea about the parametric form of f, but this form is not adequate throughout the entire range of the data. Using a parametric procedure in this situation would not be appropriate because the resulting fit would be misleading (biased) at points where the data deviates from the specified model. This leads one to consider using a nonparametric procedure. While this approach would be able to capture the different deviations in the data, it would ignore any information that the user has about the underlying structure of the data, resulting in a more variable fit than is probably necessary.

The proposed research presents some possible solutions to this dilemma. Methods are developed which *combine* the parametric and nonparametric procedures (OLS and kernel regression, for example) in order to both incorporate any information the user has about a parametric model and to detect deviations in the data from this model. The proposed methods are very flexible in terms of handling different amounts of model misspecification, and by combining the "best" (bias and variance properties) of both procedures, provide noticeable improvements over the two procedures when used individually. The next chapter contains a brief overview of *semiparametric* procedures (and the partial linear model), which take the approach of combining parametric and nonparametric expressions in the same model. These procedures have been developed for a slightly different problem than that considered here, but the general idea is extended to

develop one of the three proposed methods. This and the other two methods are presented in Chapter 5.

Chapter 4: Semiparametric Procedures

4.A Introduction

Consider now the concept of combining a parametric fit and a nonparametric fit in the same model. The goal of the current research is to develop methods which combine a parametric polynomial fit (by OLS) with a nonparametric fit (kernel or local polynomial), where both fits are based on the (single) regressor X. This particular problem has received very little attention in the literature, but some techniques have been developed for problems closely related to this. One of these techniques (discussed below) is that of *semiparametric modeling*, which combines a parametric fit based on certain regressors with a nonparametric fit based on other regressors. This technique is extended in the next chapter to the case where both fits are based on the same regressors. For a brief introduction to semiparametric models and a discussion of several forms that they can take, see Härdle (1990).

4.B Partial Linear Model

The form of semiparametric model that has received the most attention is the *partial linear model*. In this model, the response y depends on two sets of regressors (X,T), where the mean response is linearly related to $X \in \Re^p$ (parametric component), but cannot be easily parameterized in terms of $T \in \Re^d$ (nonparametric component). This model can be expressed as

$$y_i = \mathbf{x}_i'\beta + f(\mathbf{t}_i) + \varepsilon_i \quad (1 \le i \le n), \tag{4.B.1}$$

where the $\mathbf{x}_i'$ are fixed known $p \times 1$ vectors, β is an unknown vector of parameters, the $\mathbf{t}_i$ are fixed known $d \times 1$ vectors, and f: $\Re^d \to \Re$ is an unknown (smooth) regression function

(Speckman (1988)). The term "partial linear model" is derived from the linear structure of the parametric component $\mathbf{x}_i'\beta$ of the model. To obtain the estimates of the responses (the $\hat{y}$'s), one must obtain the estimates $\hat{\beta}$ and $\hat{f}$ of the unknown β and f in (4.B.1).

The approach taken by Speckman (1988) for obtaining these estimates begins by supposing that f in (4.B.1) can be parameterized as $\mathbf{f} = (f(t_1), \ldots, f(t_n))' = \mathbf{T}\gamma$, where $\mathbf{T}$ is an $n \times q$ matrix of full rank and γ is an additional parameter vector. In order for the $n \times (p+q)$ matrix $(\mathbf{X},\mathbf{T})$ to have full rank, Speckman assumes for simplicity that the unit vector $(1, \ldots, 1)'$ is in the span of $\mathbf{T}$, but not of $\mathbf{X}$. (This is also going to be an important consideration when this procedure is modified to the case of only one set of regressors in the next chapter). Model (4.B.1) can now be written in matrix notation as

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{T}\gamma + \varepsilon \,. \tag{4.B.2}$$

By taking the derivative of $(\mathbf{y} - \mathbf{X}\beta - \mathbf{T}\gamma)'(\mathbf{y} - \mathbf{X}\beta - \mathbf{T}\gamma)$, first with respect to β (with γ fixed), and then with respect to γ (with β fixed), and setting these equations equal to zero, one can obtain the following normal equations for (4.B.2):

$$\begin{aligned} \mathbf{X}'\mathbf{X}\beta &= \mathbf{X}'(\mathbf{y} - \mathbf{T}\gamma), \\ \mathbf{T}\gamma &= \mathbf{P}_{\mathbf{T}}(\mathbf{y} - \mathbf{X}\beta), \end{aligned} \tag{4.B.3}$$

where $\mathbf{P}_{\mathbf{T}} = \mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'$ denotes projection onto the column space of $\mathbf{T}$. Speckman presents two approaches for obtaining the estimates of β and $\mathbf{f}$ (or $\mathbf{T}\gamma$ here) from these normal equations. The first is due to Green, Jennison, and Seheult (1985). Their method begins with substituting for $\mathbf{T}\gamma$ in the first equation of (4.B.3) with the second equation of (4.B.3) and solving for β to obtain

$$\hat{\beta} = (\mathbf{X}'(\mathbf{I}-\mathbf{P}_T)\mathbf{X})^{-1}\mathbf{X}'(\mathbf{I}-\mathbf{P}_T)\mathbf{y},$$
$$\mathbf{T}\hat{\gamma} = \mathbf{P}_T(\mathbf{y}-\mathbf{X}\hat{\beta}) . \qquad (4.B.4)$$

Then Green *et al* proposed replacing the projection operator $\mathbf{P}_T$ by a "smoother" M to obtain the final estimates. Taking this smoother to be the kernel hat matrix $\mathbf{H}^{(ker)}$ from kernel smoothing on T defines the Green-Jennison-Seheult (GJS) estimates as

$$\hat{\beta}_{GJS} = (\mathbf{X}'(\mathbf{I}-\mathbf{H}^{(ker)})\mathbf{X})^{-1}\mathbf{X}'(\mathbf{I}-\mathbf{H}^{(ker)})\mathbf{y},$$
$$\hat{\mathbf{f}}_{GJS} (= \mathbf{T}\hat{\gamma}) = \mathbf{H}^{(ker)}(\mathbf{y}-\mathbf{X}\hat{\beta}_{GJS}) . \qquad (4.B.5)$$

The second approach for obtaining $\hat{\beta}$ and $\hat{\mathbf{f}}$ has a little more intuitive appeal, and is the approach used in the current research. As in the GJS method above, the same steps are taken to arrive at equations (4.B.4). However, at this point, since $\mathbf{P}_T$ is idempotent, one can write (4.B.4) equivalently as

$$\hat{\beta} = (\mathbf{X}'(\mathbf{I}-\mathbf{P}_T)'(\mathbf{I}-\mathbf{P}_T)\mathbf{X})^{-1}\mathbf{X}'(\mathbf{I}-\mathbf{P}_T)'(\mathbf{I}-\mathbf{P}_T)\mathbf{y},$$
$$\mathbf{T}\hat{\gamma} = \mathbf{P}_T(\mathbf{y}-\mathbf{X}\hat{\beta}) . \qquad (4.B.6)$$

By inspection of these estimates, one can think of the estimate of β as coming from first adjusting $\mathbf{X}$ and $\mathbf{y}$ for the nonparametric component, and then regressing the *partial* residual $(\mathbf{I}-\mathbf{P}_T)\mathbf{y}$ on the *partial* residual $(\mathbf{I}-\mathbf{P}_T)\mathbf{X}$. Replacing $\mathbf{P}_T$ with $\mathbf{H}^{(ker)}$ and defining the partial residuals (after "adjustment" for dependence on T) as

$$\tilde{\mathbf{X}} = (\mathbf{I}-\mathbf{H}^{(ker)})\mathbf{X} ,$$
$$\tilde{\mathbf{y}} = (\mathbf{I}-\mathbf{H}^{(ker)})\mathbf{y} , \qquad (4.B.7)$$

Speckman obtains the following estimates for β and $\mathbf{f}$:

$$\hat{\beta}_p = (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{y}},$$
$$\hat{\mathbf{f}}_p = \mathbf{H}^{(\ker)}(\mathbf{y} - \mathbf{X}\hat{\beta}_p). \qquad (4.B.8)$$

Intuitively, these estimates may be interpreted as normal equations for a (parametric) regression model with partially adjusted residuals. Also, note that if the smoother M that replaces $\mathbf{P}_T$ is chosen to be symmetric and idempotent, then estimates (4.B.5) and (4.B.8) are identical. However, in using $\mathbf{H}^{(\ker)}$ as the smoother, one obtains two distinct sets of estimates, since $\mathbf{H}^{(\ker)}$ is neither symmetric nor idempotent.

Speckman proves several theorems regarding the rates of convergence of the biases and variances of the estimates in (4.B.5) and (4.B.8). Under several assumptions, he proves that asymptotically the variance of $\hat{\beta}_p$ converges at the "parametric rate" $n^{-1/2}$, and the bias of $\hat{\beta}_p$ converges at the "nonparametric rate" $o(h^{2p})$. He also proves that the bias of $\hat{\beta}_{GJS}$ converges at a slower rate ($O(h^p)$), providing more incentive for using the estimates $\hat{\beta}_p$ and $\hat{\mathbf{f}}_p$ of (4.B.8). For $\hat{\mathbf{f}}$, Speckman proves that the biases and variances for both $\hat{\mathbf{f}}_{GJS}$ and $\hat{\mathbf{f}}_p$ all converge at the same rate as when the parametric term β is not present in the model. All of these results suggest that $\hat{\beta}_p$ and $\hat{\mathbf{f}}_p$ should perform extremely well as estimates in the partial linear model. However, as Speckman points out, the results above are asymptotic in nature, and more work is needed to determine small sample properties.

Recall again that the current research focuses on obtaining regression estimates based on only one set of regressors X (or more explicitly, the single regressor model). The semiparametric methods explained in this chapter have been developed with two sets of regressors (X and T) in mind. A natural setting for these methods is analysis of covariance, where the covariate t_i enters the model in a nonparametric fashion, and the treatments of interest enter parametrically through $x_i\beta$, where, for the case of two treatments, $x_i = 1$ for the first treatment group and $x_i = 0$ for the second treatment group (which could be a control group). The treatment effect is then given by $\hat{\beta}$, which also

provides an F-test to test for significant differences between the treatments. Speckman (1988) gives several examples in which he applies his semiparametric procedure to this setting of analysis of covariance. In the next chapter, Speckman's procedure is modified for models with a single regressor X to give the first of the three proposed model-robust regression procedures. The key idea for this procedure is the use of residuals from the parametric fit to determine the nonparametric portion of the fit. Also, this semiparametric procedure involves *simultaneously* fitting a parametric and a nonparametric model, whereas the other procedures combine two *separate* parametric and nonparametric fits.

Chapter 5: Model Robust Regression

Recall that the problem of interest in the current research is how to obtain a regression fit that both incorporates some (parametric) knowledge about the underlying model, and is able to detect specific deviations in the data from this model. The solution to this problem should be able to handle cases ranging from the specified parametric model being the true underlying model, to the specified model being a gross misspecification of the true model. The approach taken here is to combine a parametric regression fit, which is based on the researcher's knowledge of the underlying model, with a nonparametric regression fit, which is designed to capture any structure in the data that the parametric fit fails to explain. This chapter contains the development of the three proposed methods of combining these two fits in order to achieve a final fit that is robust to the varying degrees of model misspecification (hence the name *model-robust regression*). The next chapter contains comparisons, based on an MSE criterion, among these methods and the individual parametric and nonparametric methods.

5.A Partial Linear Regression (PLR)

The first model-robust procedure is a modification of Speckman's semiparametric procedure described in chapter 4. Since it is based on the partial linear model concept, this procedure is called *partial linear regression* (*PLR*). Recall that the current research addresses the issue of obtaining a regression fit for data based on a single regressor X. All other possible terms in the parametric model specified by the user are defined to be polynomial expressions of this regressor (X^2, X^3, . . .). The semiparametric methods described in chapter 4 were developed for the situation of two sets of regressors (X and T) in the model, and thus need to be modified for models with only one set of regressors ("X" = (X, X^2, X^3, . . .) as described above). The straightforward approach taken here is to simply "replace" T with X in any step involving T in the semiparametric procedure. In

other words, all steps that would normally involve operations on T are still carried out, but now they are performed on X. The following discussion, analogous to the discussion of Speckman's semiparametric procedure in chapter 4, gives the steps of the proposed PLR procedure.

For PLR (with single regressor X), the partial linear model is expressed as

$$y_i = \mathbf{x}_i'\beta + f(x_i) + \varepsilon_i \quad (1 \le i \le n), \tag{5.A.1}$$

where the $\mathbf{x}_i'$ are fixed known $p\times 1$ vectors (comprised of the polynomial expressions of X), β is an unknown vector of parameters, x_i is the i^{th} value of the regressor X, and f: $\Re \rightarrow \Re$ is an unknown (smooth) regression function (see (4.B.1) for the analogy). Here y_i is explained by the sum of a linear (parametric) function of X and a nonparametric function of X. The estimates of β and f are obtained following the same procedure as that of Speckman, explained in chapter 4. First, suppose that f can be parameterized as $\mathbf{f} = (f(x_1), \ldots, f(x_n))' = \mathbf{T}\gamma$, where $\mathbf{T}$ is an $n\times q$ matrix of full rank and γ is an additional parameter vector. Note here (and in the previous chapter) that $\mathbf{T}$ is just a label for a "dummy" matrix, which is introduced solely for the purpose of providing a somewhat intuitive approach for obtaining estimates of β and f. At no point does one need to actually perform an operation on $\mathbf{T}$, or to even know the values of the elements of $\mathbf{T}$. Speckman used the label $\mathbf{T}$ to "represent" his second set of regressors (T), but here $\mathbf{T}$ still "represents" X, the only regressors available. The label $\mathbf{T}$ is maintained here for simplicity--to keep the notation exactly the same as that of chapter 4. Now, model (5.A.1) can be expressed in matrix notation as

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

just as in (4.B.2). As was the case previously, the unit vector $(1, \ldots, 1)'$ is taken to be in the span of $\mathbf{T}$, but not of $\mathbf{X}$. As is shown shortly, this must be the case for PLR. Following the same steps as Speckman, the estimates for β and γ may be expressed as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'(\mathbf{I}-\mathbf{P}_T)'(\mathbf{I}-\mathbf{P}_T)\mathbf{X})^{-1}\mathbf{X}'(\mathbf{I}-\mathbf{P}_T)'(\mathbf{I}-\mathbf{P}_T)\mathbf{y},$$
$$\mathbf{T}\hat{\boldsymbol{\gamma}} = \mathbf{P}_T(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}),$$

as in (4.B.6), where $\mathbf{P}_T = \mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'$. The main difference between PLR and Speckman's semiparametric procedure arises in the selection of the smoother M that replaces the projection operator $\mathbf{P}_T$. As before, a kernel hat matrix $\mathbf{H}^{(ker)}$ may be used, but here instead of obtaining $\mathbf{H}^{(ker)}$ from smoothing on T (which no longer exists), $\mathbf{H}^{(ker)}$ is obtained from kernel smoothing on X. This kernel hat matrix is labeled as $\mathbf{H}_P^{(ker)}$, since the kernel fit needed for PLR is generally different from that needed for fitting the data with a kernel fit alone (by $\mathbf{H}^{(ker)}$). Now, define the partial residuals (after adjustment for the nonparametric component of X) as

$$\tilde{\mathbf{X}} = (\mathbf{I}-\mathbf{H}_P^{(ker)})\mathbf{X},$$
$$\tilde{\mathbf{y}} = (\mathbf{I}-\mathbf{H}_P^{(ker)})\mathbf{y},$$

as in (4.B.7). Replacing $\mathbf{P}_T$ with $\mathbf{H}_P^{(ker)}$ and then substituting $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{y}}$ into the estimates $\hat{\boldsymbol{\beta}}$ and $\mathbf{T}\hat{\boldsymbol{\gamma}}$ above (or, equivalently, regressing $\tilde{\mathbf{y}}$ on $\tilde{\mathbf{X}}$) gives the estimates

$$\hat{\boldsymbol{\beta}} = (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{y}},$$
$$\hat{\mathbf{f}} = \mathbf{H}_P^{(ker)}(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}),$$

analogous with (4.B.8). From the expression for $\hat{\boldsymbol{\beta}}$, the definition of $\tilde{\mathbf{X}}$, and the fact that each row of $\mathbf{H}_P^{(ker)}$ sums to one, it can be shown that the unit vector $(1, \ldots, 1)'$ cannot be included as one of the columns of $\mathbf{X}$. The reason for this is that $\tilde{\mathbf{X}}$ would then contain the zero vector as a column, and $(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}$ would not exist (for details, see Appendix C). Thus, the "$\mathbf{X}$" matrix for PLR is the $n \times k$ matrix of k regressors *not* augmented with a column of ones, and the PLR $\boldsymbol{\beta}$ vector is $k \times 1$, not $(k+1) \times 1$ (since there is no intercept term). To distinguish this matrix from the "usual" $\mathbf{X}$ matrix with a column of ones, the PLR $\mathbf{X}$ matrix is denoted $\mathbf{X}_P$. Thus, the final PLR estimates of $\boldsymbol{\beta}$ and $\mathbf{f}$ $(= \mathbf{T}\boldsymbol{\gamma})$ are as follows:

$$\hat{\beta}_{PLR} = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{y},$$
$$\hat{f}_{PLR} = H_P^{(ker)}(y - X_P\hat{\beta}_{PLR}), \qquad (5.A.2)$$

where $\tilde{X} = (I - H_P^{(ker)})X_P$, and $(y - X_P\hat{\beta}_{PLR})$ denotes the residuals from a parametric fit based on X_P and $\hat{\beta}_{PLR}$. The fitted values for the observations are then given by

$$\hat{y}_{PLR} = X_P\hat{\beta}_{PLR} + \hat{f}_{PLR}, \qquad (5.A.3)$$

$$\text{or} \quad \hat{y}_{PLR} = H^{(PLR)}y, \qquad (5.A.4)$$

where

$$H^{(PLR)} = (h_{ij}^{(PLR)}) = H_P^{(ker)} + \tilde{X}(\tilde{X}'\tilde{X})^{-1}\tilde{X}'(I - H_P^{(ker)}) \qquad (5.A.5)$$

is the PLR "hat matrix" (when using kernel regression).

An important point needs to be made here. The nonparametric portion of the fit ($\hat{f}_{PLR}$, the fit to the residuals) may be improved upon by using $H_P^{(LPR)}$ from a local polynomial (linear or quadratic) fit. This, in fact, is done in the final implementation of this procedure (LLR is used). However, in obtaining $\tilde{X}$ and $\tilde{y}$, $H_P^{(ker)}$ is always used, so that $\tilde{X}$ and $\tilde{y}$ are always defined the same way. To support this idea, several preliminary examples were studied where $H_P^{(LPR)}$ was used in constructing $\tilde{X}$ and $\tilde{y}$. The resulting PLR fits were somewhat erratic and problems often arose in choosing the bandwidth for this procedure. Using $H_P^{(ker)}$ never resulted in such problems. So, it is important to remember that when PLR is presented as using LPR as the nonparametric fitting technique, this means that LPR is used for the residual fit, but *not* for obtaining $\tilde{X}$ and $\tilde{y}$. Also, this would result in the PLR hat matrix

$$H^{(PLR)} = H_P^{(LPR)} + (I - H_P^{(LPR)})X_P(\tilde{X}'\tilde{X})^{-1}\tilde{X}'(I - H_P^{(ker)}). \qquad (5.A.6)$$

As shown in (5.A.3) above, PLR obtains fitted values as the sum of a parametric fit based on the regressors in $\mathbf{X}_P$, and a nonparametric fit based on the residuals from this parametric fit. The key notion here is that these two fits are obtained "*simultaneously*", with one having a direct impact on the other. In particular, the (nonparametric) kernel hat matrix (obtained as the initial step of PLR) is used to form $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{y}}$, which are used to obtain $\hat{\beta}_{PLR}$ and hence the parametric portion of the fit, with the residuals from this parametric fit being smoothed with the original kernel hat matrix to give the nonparametric portion of the fit. It is easily seen from this abbreviated description how intricate the parametric and nonparametric components really are for PLR. This complexity in the basic structure of PLR is considered here as a slight drawback of this procedure, and the other two proposed methods attempt to resolve this complexity problem by providing simpler, more intuitive methods for combining the parametric and nonparametric fits. Also, notice that the parametric fit $\mathbf{X}_P\hat{\beta}_{PLR}$ always crosses the y-axis at zero, since no intercept term is contained in the model. Thus, in general, the parametric fit is inadequate by itself, and the nonparametric portion, in addition to capturing special structure in the data, must also correct for this inadequacy. (Figures 6.C.5 (a) and (b) in chapter 6 illustrate these fitting characteristics for an example with generated data). Also, the nonparametric fit is always included, in its raw form, to obtain the final fit, even if the underlying curve is quite smooth and a "usual" parametric fit would be sufficient. This characteristic of PLR also may be a drawback in many situations, since the use of an unnecessary nonparametric fit may add excess variance to the final fit. A possible improvement to PLR in these cases would be to somehow provide for a better parametric fit, and to allow varying "amounts" of the nonparametric fit to be used (more emphasis on nonparametric portion when parametric portion is inadequate, and less emphasis on nonparametric portion when parametric portion is adequate). These needs are addressed by the final two model-robust procedures.

5.B Model Robust Regression 1 (MRR1)

A model-robust procedure which addresses the shortcomings of PLR mentioned in the previous section was developed by Einsporn (1987) and Einsporn and Birch (1993) and was entitled HATLINK. Called *Model Robust Regression 1* (*MRR1*) here, this is a very simple, but effective method, which combines the fit of a parametric model with the fit of a nonparametric model, both to the raw data, in a convex combination via a mixing parameter λ. In the current work, the parametric model is of the form $\mathbf{y} = \mathbf{X}\beta + \varepsilon$ and the method of OLS is used to estimate the parameters, β. The nonparametric model is fit using some nonparametric method such as kernel regression or local polynomial regression. Kernel regression is used in the development of MRR1 in the next section. (For results from using LPR instead of kernel, one can just replace $\mathbf{H}^{(ker)}$ with $\mathbf{H}^{(LPR)}$ in all of the following expressions).

5.B.1 Development

In notational form, letting $\hat{\mathbf{y}}_{ols} = \mathbf{X}\hat{\beta}_{ols} = \mathbf{H}^{(ols)}\mathbf{y}$ be the OLS fitted values and $\hat{\mathbf{y}}_{ker} = \mathbf{H}^{(ker)}\mathbf{y}$ be the kernel fitted values, the MRR1 fitted values are obtained simply as

$$\hat{\mathbf{y}}_{MRR1} = \lambda\hat{\mathbf{y}}_{ker} + (1-\lambda)\hat{\mathbf{y}}_{ols}\,, \tag{5.B.1}$$

where $\lambda \in [0,1]$. In terms of hat matrices, (5.B.1) can be written as

$$\begin{aligned}\hat{\mathbf{y}}_{MRR1} &= \lambda\mathbf{H}^{(ker)}\mathbf{y} + (1-\lambda)\mathbf{H}^{(ols)}\mathbf{y} \\ &= \left[\lambda\mathbf{H}^{(ker)} + (1-\lambda)\mathbf{H}^{(ols)}\right]\mathbf{y} \\ &= \mathbf{H}^{(MRR1)}\mathbf{y}\,,\end{aligned} \tag{5.B.2}$$

and the MRR1 hat matrix is seen to be $\mathbf{H}^{(\text{MRR1})} = (h_{ij}^{(\text{MRR1})}) = \lambda\mathbf{H}^{(\text{ker})} + (1-\lambda)\mathbf{H}^{(\text{ols})}$. (This simple "link" between the two hat matrices was the origin of the term "HATLINK"). Also, the fit for an individual observation at x_i can be obtained as

$$\hat{y}_i^{(\text{MRR1})} = \sum_{j=1}^{n} h_{ij}^{(\text{MRR1})} y_j = \sum_{j=1}^{n} \left[\lambda h_{ij}^{(\text{ker})} + (1-\lambda) h_{ij}^{(\text{ols})}\right] y_{ij} . \qquad (5.B.3)$$

The key idea of MRR1 is the introduction of the mixing parameter λ. The purpose of λ is to combine the parametric and nonparametric fits in the most efficient proportions to achieve an adequate fit to the data. This λ ranges from 0 to 1 based on the amount of misspecification of the user's parametric model. If the parametric model gives an adequate fit, then λ should be close to 0, which gives a fit based mainly on the parametric fit. On the other hand, if the parametric model has been greatly misspecified, then λ should be close to 1, which gives a fit based mainly on the nonparametric fit. In cases where the specified model is somewhat adequate, but cannot capture all of the structure in the data, a λ near the middle of [0,1] may be appropriate to allow for a proportion of the nonparametric fit to enter the final fit in an attempt to capture this extra structure. Olkin and Spiegelman (1987) introduced this same technique as a semiparametric method of density estimation. In their article, they use likelihood and pseudolikelihood functions to prove, under certain regularity conditions, that when the specified model is correct, the rate of convergence of their estimator is the same as that of the traditional maximum likelihood estimator, and when the specified model is incorrect, the rate of convergence is the same as when using a kernel estimator.

Notice that MRR1 combines two *separate* fits to the data, one based on OLS (for the parametric model) and one on kernel regression (for the nonparametric model). So, at each x_i, there are two fitted values available for estimating y_i. Based on the value of λ, MRR1 selects a value between these two fitted values for the final estimate of y_i. For example, if λ is large (say, around .80), then the MRR1 fitted value would be closer to the

kernel fitted value, and vice-versa for small λ. This characteristic of MRR1, although nice in its simplicity, can also be a drawback of the procedure. In particular, if locations exist in the data where OLS and kernel regression either both give fitted values too high or both give fitted values too low, then MRR1 has no way of correcting for these insufficient fits. The third proposed model-robust procedure resolves this problem. Before presenting this final procedure, one more issue needs to be addressed about MRR1: choice of λ.

5.B.2 Choosing λ

The problem of selecting λ is similar to that of choosing the bandwidth in kernel regression, as discussed in section 3.B.3. Many of the bandwidth choice considerations are also present here for choosing λ, such as which error criterion to use (SSE, PRESS, penalizing functions, plug-in estimates, etc.), global vs. local procedures, and guarding against methods that overfit or underfit the data (here, this would be choosing λ too large or too small, respectively). An example of an overfitting procedure would be to use SSE to choose λ, as this would nearly always result in using all of the kernel fit, i.e., a λ of 1. In the current research, the approaches taken to solve these problems of choosing λ closely parallel the conclusions reached about choosing the bandwidth h. For the results given here, the criterion for choosing λ was taken to be PRESS* (the "penalized PRESS") or PRESS**. As with choosing h, it was hoped that PRESS* would incorporate the desirable properties of both PRESS and penalizing functions, and would provide extra protection against choosing λ too large (overfitting). Also, this *global* procedure was used to maintain the simplicity of the procedure. PRESS** was used in order to protect against cases where PRESS* might choose a λ too small (underfitting). Einsporn (1987), in addition to the PRESS* criterion, also developed some C_p-based criteria for selecting λ. Mallow's C_p statistic is essentially an estimate for the sums of the individual variances and squared biases of $\hat{y}_i$, $i = 1, \ldots, n$, standardized by σ^2 (for more details on C_p, see Myers (1990)). Einsporn discusses how C_p strikes the proper balance between an increasing

variance and a decreasing bias as λ ranges from 0 to 1 (i.e., as more kernel is added to the fit). He develops four versions of this C_p criterion, based on different estimates of σ^2 and different expressions for the error degrees of freedom. The comparative behaviors of these criteria depend on the amount of model misspecification present, and no one version has been found to be uniformly best. However, in considering overall performance, with special emphasis placed on the situation of small to moderate model misspecification (the interest of the current research), Einsporn's C_p3 criterion has been determined to perform best. (Einsporn (1987) presents numerous simulation studies in this regard). This C_p3 criterion is given by

$$C_p3(\lambda) = \mathrm{tr}(\mathbf{H}^{(\mathrm{MRR1})}(\lambda)) + \frac{\left[s^2(\lambda) - s^2_{\mathrm{ols}}\right]\left[n - \mathrm{tr}(\mathbf{H}^{(\mathrm{MRR1})}(\lambda))\right]}{s^2_{\mathrm{ols}}}, \qquad (5.B.4)$$

where s^2_{ols} is the OLS estimate of variance for the user's model (as in equation 2.7), $\mathbf{H}^{(\mathrm{MRR1})}(\lambda)$ is the MRR1 hat matrix for a certain λ, and $s^2(\lambda)$ is the estimate of variance for the MRR1 fit for a certain λ, given by

$$s^2(\lambda) = \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{(\mathrm{MRR1})}(\lambda)\right)^2}{n - \mathrm{tr}(\mathbf{H}^{(\mathrm{MRR1})}(\lambda))}. \qquad (5.B.5)$$

In various simulations, Einsporn found C_p3 and PRESS* to behave very similarly, and Einsporn and Birch (1993) selected PRESS* as their selection criterion when applying their HATLINK procedure. The same is done here, but with a more thorough study of comparisons among different selection procedures for h and λ, as presented in Chapter 7.

5.C Model Robust Regression 2 (MRR2)

The final model-robust procedure is developed in this section, and is motivated by the desire to improve upon the shortcomings of PLR and MRR1. Basically, this procedure has as its origin the MRR1 procedure, with adjustments made based on the PLR procedure. Due to its close relation to MRR1, this proposed procedure is entitled *Model Robust Regression 2* (*MRR2*).

5.C.1 Development

MRR2 maintains the simplicity of MRR1 by once again using separate parametric and nonparametric fits to construct the final fit, and MRR2 also makes use of a mixing parameter λ. The difference arises in how the two individual fits are obtained, and in how they are combined together. The parametric portion of MRR2 is obtained as a parametric fit to the *raw data* (as is the case with MRR1). Using OLS to fit $\mathbf{y} = \mathbf{X}\beta + \varepsilon$, this fit may be expressed as $\hat{\mathbf{y}}_{ols} = \mathbf{X}\hat{\beta}_{ols}$. However, the nonparametric portion of MRR2, instead of coming from a nonparametric fit to the raw data, comes from a nonparametric fit to the *residuals* from the parametric fit. Denoting these residuals as $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\beta}_{ols}$, the nonparametric fit (using kernel) may be expressed as $\hat{\mathbf{r}} = \mathbf{H}_2^{(ker)}\mathbf{r}$. The kernel hat matrix $\mathbf{H}_2^{(ker)}$ is still formulated as described in chapter 3, but note that the bandwidth that determines $\mathbf{H}_2^{(ker)}$ is now based on fitting residuals, not the raw data. In other words, due to a different bandwidth, $\mathbf{H}_2^{(ker)}$ for MRR2 is not the same as $\mathbf{H}^{(ker)}$ for MRR1. Now that the parametric and nonparametric fits, $\hat{\mathbf{y}}_{ols}$ and $\hat{\mathbf{r}}$, respectively, have been obtained, the final question is how to combine them for the final fit.

The solution to this problem is very simple and intuitive. The procedure is to first obtain the parametric fit to the data, and then add to this a portion of the nonparametric fit to the residuals. The idea here is for the parametric fit, if not overly misspecified, to explain most of the structure in the data, and then for the nonparametric fit to capture any "left over" structure not captured by the initial parametric fit. This "left over" structure is

naturally contained in the residuals. Also, instead of always adding back the entire nonparametric fit, a portion of this fit is added, determined by the parameter $\lambda \in [0,1]$. This λ is chosen in the same fashion as the λ in MRR1 (by PRESS* or PRESS** for the current research), and increases from 0 to 1 as the amount of model misspecification in the parametric portion increases. Formally, the MRR2 fitted values are obtained as

$$\hat{\mathbf{y}}_{\text{MRR2}} = \hat{\mathbf{y}}_{\text{ols}} + \lambda \hat{\mathbf{r}} \ . \tag{5.C.1}$$

In terms of hat matrices, (5.C.1) can be expressed as

$$\begin{aligned} \hat{\mathbf{y}}_{\text{MRR2}} &= \mathbf{H}^{(\text{ols})}\mathbf{y} + \lambda \mathbf{H}_2^{(\text{ker})}\mathbf{r} \\ &= \mathbf{H}^{(\text{ols})}\mathbf{y} + \lambda \mathbf{H}_2^{(\text{ker})}(\mathbf{y} - \mathbf{H}^{(\text{ols})}\mathbf{y}) \\ &= \left[\mathbf{H}^{(\text{ols})} + \lambda \mathbf{H}_2^{(\text{ker})}(\mathbf{I} - \mathbf{H}^{(\text{ols})})\right]\mathbf{y} \\ &= \mathbf{H}^{(\text{MRR2})}\mathbf{y} \ , \end{aligned} \tag{5.C.2}$$

and the MRR2 hat matrix is seen to be $\mathbf{H}^{(\text{MRR2})} = (h_{ij}^{(\text{MRR2})}) = \mathbf{H}^{(\text{ols})} + \lambda \mathbf{H}_2^{(\text{ker})}(\mathbf{I} - \mathbf{H}^{(\text{ols})})$. Individual fitted observations are obtained as

$$\hat{y}_i^{\text{MRR2}} = \sum_{j=1}^{n} h_{ij}^{(\text{MRR2})} y_j \ . \tag{5.C.3}$$

(Note once again that $\mathbf{H}^{(\text{LPR})}$ can be substituted in for $\mathbf{H}^{(\text{ker})}$ in all of the expressions above to give the results of using LPR as the nonparametric fitting technique for MRR2).

5.C.2 Advantages

The goal of this section is to summarize the improvements made in MRR2 over PLR and MRR1, and to support the contention that MRR2 should be the best overall procedure for fitting data in situations of small to moderate model misspecification. First,

MRR2 is simpler and more intuitive than PLR in the sense that it involves two separate fits as opposed to two simultaneous fits. Second, recall that PLR always uses the entire nonparametric fit, which may lead to a higher variance than is necessary in the final fit. The presence of λ in MRR2 resolves this problem, because if the parametric fit is adequate, a small λ (close to zero) prevents the use of an unnecessary nonparametric fit. Also recall that the PLR parametric fit is usually very inadequate due to the intercept term being absent, and the nonparametric fit must make up for this inadequacy. MRR2 does not have this problem, since there are no "restrictions" on the parametric fit. For example, in using a "regular" OLS fit (*with* intercept term), a much more adequate parametric fit may be obtained. These are the main advantages of MRR2 over PLR.

MRR2 also has a few advantages over MRR1. First, MRR2 overcomes the MRR1 problem of cases where both of the component fits are inaccurate in the same direction (above or below the true y-value), with no way to correct this in the final fit. In these cases, there is a bias problem present in certain locations for MRR1. MRR2 resolves this problem by obtaining the basic (parametric) fit, and then adding to this a (nonparametric) *residual* fit. This residual fit provides flexibility to correct for any inaccuracies in the parametric fit. This introduction of residuals is an attempt to combine the most advantageous part of PLR with the simplicity of MRR1. It is also conjectured that applying the nonparametric fit to the residuals instead of the raw data would provide fits that are somewhat less variable. Ideally, in MRR2, the main structure of the data is removed by the parametric fit, leaving residuals to explain the remaining structure. Thus, the structure left in the residuals should be much less complex than that of the raw data (for some intuition, just think of the scale of the data (larger) versus the scale of the residuals (smaller)). So, the nonparametric fitting procedure (kernel or local polynomial) should not have to "work" as hard to fit the residuals of MRR2 as it does for the data of MRR1, and the variance properties of MRR2 may be somewhat better.

As seen above, MRR2 has been developed to combine the best bias and variance properties of PLR and MRR1. Of course, with additional considerations present for such

problems as bandwidth choice, choice of λ, and the countless number of possible data sets that could be encountered, it would not be appropriate to make general conclusions at this point about the different procedures. The next chapter contains preliminary comparisons based on an MSE criterion and the desire for a nice smooth function to fit the data. Several generated data sets are used for initial performance comparisons, and then the methods are applied to an actual data set. It will become apparent that in *most* cases PLR and MRR2 give very similar fits, both displaying an improved performance over MRR1. This behavior suggests using MRR2 as the model-robust procedure since it is much simpler than PLR, yet performs as well, or better.

Chapter 6: Initial Comparisons

The previous chapters have described five different regression techniques for fitting a set of data: an individual parametric fit, an individual nonparametric fit, and the three model-robust methods (PLR, MRR1, and MRR2) which combine the parametric and nonparametric fits. The purpose of this chapter is to provide comparisons of performances among these techniques in the situation where some knowledge is present about the form of the true underlying model, but this model is not adequate throughout the entire range of the data. For these comparisons, the parametric fitting technique is taken to be OLS, and the nonparametric fitting technique is local linear regression.

The first section of this chapter establishes the general set-up of the underlying model from which comparisons among all five procedures are made. Based on this general framework, an MSE criterion for each procedure is then developed in the following section. Several examples are then presented which supply the results of interest.

6.A Underlying Model (General Expression)

Note that the ultimate goal now is to develop an MSE criterion which can be calculated based on the same underlying model for each of the five fitting techniques to be compared. The first task, then, is to develop an expression for the underlying model from which each of the five MSE's can be derived. This expression should be such that any generated data set can be represented in this way. Since the model-robust methods being studied involve fits to both parametric and nonparametric models, and since the cases of interest are where partial, but not complete information is available about the parametric model, it seems natural to express any underlying model as a combination of parametric and nonparametric functions. Also, since PLR is the most "complex" fitting technique, in that it obtains simultaneous fits to parametric and nonparametric functions in the same

model, this serves as the starting point for developing the general expression for the underlying model.

For the development here, the *most* general expression for the underlying model can be written

$$\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon}\,, \tag{6.A.1}$$

where $\mathbf{g}(x) = [g(x_1), \ldots, g(x_n)]'$ and g is some *general* regression function. However, this model does not satisfy the characteristics described above (especially for PLR) and is made more specific as follows. In the spirit of the partial linear model on which PLR is based, $\mathbf{g}$ is divided into parametric and nonparametric portions and (6.A.1) may be expressed as

$$\mathbf{y} = \mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f} + \boldsymbol{\varepsilon}\,, \tag{6.A.2}$$

where $\mathbf{X}_P$ is the $n\times p = n\times k$ PLR $\mathbf{X}$ matrix (without a column of ones), $\boldsymbol{\beta}_P$ is a $k\times 1$ vector of parameters, and $\mathbf{f} = [f(x_1), \ldots, f(x_n)]'$, where f is an unknown (smooth) regression function. Since f is allowed to essentially take on any functional form, any specified model (involving any $\mathbf{g}$) for generating data can be expressed as (6.A.2) for PLR. For instance, a portion of the specified model may be defined as $\mathbf{X}_P\boldsymbol{\beta}_P$ (i.e., extracting the parametric part), and then the remaining portion of the specified model, whatever form it may be, is defined as $\mathbf{f}$. In other words, $\mathbf{f}$ can be thought of as "picking up" any part of the specified model that is left over after defining a part of it to be $\mathbf{X}_P\boldsymbol{\beta}_P$ (i.e., $\mathbf{f} = \mathbf{g} - \mathbf{X}_P\boldsymbol{\beta}_P$).

The results above are now applied to MRR1 and MRR2, with the resulting expressions also being appropriate for the individual parametric and nonparametric methods. The underlying function $\mathbf{g}$ continues to be split up into a parametric portion and a nonparametric portion, similar to (6.A.2). Since MRR1 and MRR2 use the linear model

$\mathbf{X}\beta$ ($\mathbf{X}$ augmented with a column of ones) as their parametric portion, $\mathbf{g}$ can now be expressed as $\mathbf{g} = \mathbf{X}\beta + \mathbf{f}$, and the underlying model (6.A.1) may be expressed as

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{f} + \boldsymbol{\varepsilon}, \qquad (6.A.3)$$

where $\mathbf{X}$ is the $n\times(k+1)$ matrix of regressors (containing a column of ones), β is a $(k+1)\times 1$ vector of parameters, and $\mathbf{f} = [f(x_1), \ldots, f(x_n)]'$, where f is an unknown (smooth) regression function. The components of (6.A.3) serve the same purpose as the corresponding components of (6.A.2), and any specified model (6.A.1) can be expressed as (6.A.3) for MRR1 and MRR2 (and for the individual parametric and nonparametric procedures). Note that even though any generated data is being expressed as coming from a single model, the actual procedures for MRR1 and MRR2 are not based on a *single* underlying model (unlike PLR, which is based on the partial linear model). Instead, MRR1 and MRR2 obtain fits based on two separate models, one parametric and one nonparametric. It is important to keep this distinction between the model-robust procedures in mind (MRR1 and MRR2 vs. PLR). Generated data for MRR1 and/or MRR2 is expressed as a single model in order to keep all of the procedures in the same general framework (for a given $\mathbf{g}$), which allows for the development of the MSE criterion in the next section. Also note that $\mathbf{g}$ has now been expressed in two different parametric forms: $\mathbf{X}_P\beta_P$ for PLR and $\mathbf{X}\beta$ for the other methods, with $\mathbf{X}\beta$ being a little more flexible by providing for an intercept term. This difference also results in the respective $\mathbf{f}$'s being unequal (namely, $(\mathbf{f}_1 = \mathbf{g} - \mathbf{X}_P\beta_P) \neq (\mathbf{f}_2 = \mathbf{g} - \mathbf{X}\beta)$). To make all expressions equal, one may take the intercept term of β (the classical β_0) to be zero, so that $\mathbf{X}\beta = \mathbf{X}_P\beta_P$. This is the approach taken in the current work. Getting all components equal is not a necessity, but does simplify the calculations of the MSE's developed in the next section (only one "$\mathbf{X}\beta$" and one $\mathbf{f}$ need to be kept track of, instead of two of each).

6.B MSE Criterion

Discussed earlier in section 3.B.3 as a bandwidth choice criterion for nonparametric regression, mean squared error (MSE) is also used here as the diagnostic for making comparisons among the various fitting techniques. Based on fitting a particular data set, formulas are developed in this section for the MSE of the fits at any point in the range of the data. If concerned only with the data points themselves, these formulas can be used to obtain the n MSE's desired. These n MSE's can then be converted to a single number by calculating the *average MSE* (*AVEMSE*) across all of the fitted values at the specific data points. However, in comparing entire fits for the various techniques, it may be more appropriate to compute an *integrated MSE* ($\int MSE$) across all of the locations in the entire range of the data. The formulas derived here also allow for this calculation (or at least an excellent approximation of it). This approximate integrated MSE (called *INTMSE*) is formed by taking the average of the MSE calculations at 1000 locations from 0 to 1 (the range of the transformed x's). This INTMSE is used as the final criterion for comparing the different fitting techniques, whereas AVEMSE is used to find the "optimal" h and λ for the various procedures, as described below..

Before deriving the MSE formulas, one other crucial point must be addressed: how to treat the selection of the bandwidth h and the mixing parameter λ (when needed). The approach taken in this research follows that used by Speckman (1988). Namely, h and λ are both considered as *fixed* quantities when calculating the MSE's. Specifically, h and λ are taken to be the "optimal" bandwidth and mixing parameter for each particular procedure, where "optimal" refers to minimizing the AVEMSE over all possible values of h and λ. Using the notation of Härdle (1990), these optimal values may be labeled h_o and λ_o. The optimal h and λ (when both are needed) are found separately, not jointly as a pair. For example, in MRR1, h_o is the h which minimizes AVEMSE when performing a nonparametric fit to the data, while λ_o is then the λ which minimizes AVEMSE for the final fit that combines this already determined nonparametric fit (based on h_o) with a parametric fit. The MSE formulas for determining h_o and λ_o are developed shortly as part

of the development of the other MSE formulas for the five competing procedures. Obtaining the fixed, optimal h_o and λ_o serves two important purposes. First, it makes the derivations of the MSE formulas much simpler than if h and λ were chosen by data-driven methods. Secondly, data-driven methods (such as PRESS* or PRESS**) for selecting h and λ can be evaluated by comparing the chosen h and λ to the optimal h_o and λ_o. Some such comparisons are provided in Chapter 7. Now, with h and λ fixed as h_o and λ_o, and the underlying model for generated data defined as in the previous section, the MSE formulas can be derived.

The following strategy is taken in developing and presenting these formulas. First, the detailed derivations presented below and in the Appendix, and the initial formulas which result from these derivations, are for the MSE of the *vector* of fitted values ($\hat{\mathbf{y}}$) at the actual data locations. This is done in order to provide for the very simple calculation of AVEMSE, which is the important selector of the optimal h and λ. The formulas for the MSE at any individual location within the range of the data can then be determined via a straightforward extension of the steps used to obtain MSE($\hat{\mathbf{y}}$), simply be considering an individual point $\mathbf{x}_o$ instead of the entire "data matrix" $\mathbf{X}$ or $\mathbf{X}_P$. The second point in this presentation strategy is to carry out all derivations with OLS used as the parametric technique and kernel regression used as the nonparametric technique. To obtain the results when using local polynomial regression (which is used in all of the examples to come), one can simply replace "ker" with "LPR" in all of the derivations, *except* for PLR. This different derivation for PLR is given in Appendix D.

The five MSE formulas (for OLS, kernel regression, MRR1, MRR2, and PLR) are each derived from the underlying model $\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon}$ developed in the previous section, where $\mathbf{g}$ is expressed as $\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f}$ (for PLR), or $\mathbf{X}\boldsymbol{\beta} + \mathbf{f}$ with $\beta_0 = 0$ (for OLS, kernel, MRR1, and MRR2). Actually, the formulas derived here are the bias and variance formulas for each procedure. Of course, the MSE can then be obtained by squaring the bias and adding this to the variance.

OLS

Consider first the simplest case, OLS. Here $\hat{\mathbf{y}}_{ols} = \mathbf{X}\hat{\boldsymbol{\beta}}_{ols} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}^{ols}\mathbf{y}$, and the bias of $\hat{\mathbf{y}}_{ols}$ is

$$\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}_{ols}) &= \mathrm{E}(\hat{\mathbf{y}}_{ols}) - \mathrm{E}(\mathbf{y}) \qquad (6.B.1)\\
&= \mathrm{E}(\mathbf{H}^{(ols)}\mathbf{y}) - \mathrm{E}(\mathbf{y})\\
&= \mathbf{H}^{(ols)}\mathrm{E}(\mathbf{y}) - \mathbf{g}\\
&= \mathbf{H}^{(ols)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) - \mathbf{X}\boldsymbol{\beta} - \mathbf{f} \qquad (\text{since } \mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{f} + \boldsymbol{\varepsilon},\ \mathrm{E}(\boldsymbol{\varepsilon}) = \mathbf{0})\\
&= \mathbf{H}^{(ols)}\mathbf{X}\boldsymbol{\beta} + \mathbf{H}^{(ols)}\mathbf{f} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f}\\
&= \mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(ols)})\mathbf{f} \qquad (\text{since } \mathbf{H}^{(ols)}\mathbf{X} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = \mathbf{X})\\
&= -(\mathbf{I} - \mathbf{H}^{(ols)})\mathbf{f}\,.
\end{aligned}$$

The variance of $\hat{\mathbf{y}}_{ols}$ is given by

$$\begin{aligned}
\text{Var}(\hat{\mathbf{y}}_{ols}) &= \text{Var}(\mathbf{H}^{(ols)}\mathbf{y})\\
&= \mathbf{H}^{(ols)}\text{Var}(\mathbf{y})\mathbf{H}'^{(ols)}\\
&= \mathbf{H}^{(ols)}(\sigma^2\mathbf{I})\mathbf{H}'^{(ols)} \qquad (\text{since } \text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}) \qquad (6.B.2)\\
&= \sigma^2\mathbf{H}^{(ols)}\mathbf{H}'^{(ols)}\\
&= \sigma^2\mathbf{H}^{(ols)}\,,
\end{aligned}$$

as in (2.5). Note that as the true model deviates farther from the linear model $\mathbf{X}\boldsymbol{\beta}$ (i.e., has a more prominent $\mathbf{f}$ component), the bias increases (while the variance is unaffected). If the true model is just $\mathbf{X}\boldsymbol{\beta}$, then the bias is zero as described in Chapter 2. The bias and variance of the fitted value $\hat{y}_o$ at any individual location $\mathbf{x}_o' = (\ 1\ x_o\ x_o^2\ \cdots\ )$ can be obtained through similar arguments, starting with the underlying model written in the form $y_o = \mathbf{x}_o'\boldsymbol{\beta} + f(x_o) + \varepsilon$. The resulting formulas are given by

$$\text{Bias}(\hat{y}_{o,\,ols}) = \mathbf{x}_o'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{f} - f(x_o)\ , \qquad (6.B.3)$$

$$\text{Var}(\hat{y}_{o,\,ols}) = \sigma^2 \mathbf{x}_o'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_o \ . \qquad (6.B.4)$$

The bias and variance expressions for the remaining procedures are derived by the same techniques used above for OLS. Details are provided in Appendix D.

Kernel (or LPR)

For kernel regression, $\hat{\mathbf{y}}_{ker} = \mathbf{H}^{(ker)}\mathbf{y}$, where the $h_{ij}^{(ker)}$ are defined as in (3.B.3). The bias of $\hat{\mathbf{y}}_{ker}$ is given by $E(\hat{\mathbf{y}}_{ker}) - E(\mathbf{y})$, and after some calculations (see Appendix D.1), one can obtain

$$\text{Bias}(\hat{\mathbf{y}}_{ker}) = -(\mathbf{I} - \mathbf{H}^{(ker)})(\mathbf{X}\boldsymbol{\beta} + \mathbf{f}). \qquad (6.B.5)$$

With calculations analogous to those in (6.B.2) for OLS, the variance of $\hat{\mathbf{y}}_{ker}$ can be obtained as

$$\text{Var}(\hat{\mathbf{y}}_{ker}) = \sigma^2\mathbf{H}^{(ker)}\mathbf{H}'^{(ker)}, \qquad (6.B.6)$$

recalling that $\mathbf{H}^{(ker)}$ is not generally symmetric or idempotent. Also recall that the bandwidth is taken to be fixed (at h_o). Here h_o is the h that minimizes the AVEMSE calculated from the bias and variance formulas in (6.B.5) and (6.B.6). Note that if the bandwidth is chosen close to zero so that $\mathbf{H}^{(ker)}$ is the identity matrix, then the kernel bias is zero, but the variance is maximized as $\sigma^2\mathbf{I}$. This is what occurs when one "connects the dots" to obtain a kernel fit, and illustrates the concept of a trade-off between bias and variance.

Similar to the steps in OLS, the bias and variance of $\hat{y}_o$ (for any $\mathbf{x}_o'$) can be obtained as

$$\text{Bias}(\hat{y}_{o,\,\text{ker}}) = [\mathbf{h}_o'^{(\text{ker})}\mathbf{X} - \mathbf{x}_o']\beta + \mathbf{h}_o'^{(\text{ker})}\mathbf{f} - f(x_o) \ , \qquad (6.B.7)$$

$$\text{Var}(\hat{y}_{o,\,\text{ker}}) = \sigma^2\, \mathbf{h}_o'^{(\text{ker})}\mathbf{h}_o^{(\text{ker})} \ , \qquad (6.B.8)$$

where $\mathbf{h}_o'^{(\text{ker})}$ is the row of a kernel hat matrix determined by the distances from the data points (the x's) to x_o. One can think of obtaining $\mathbf{h}_o'^{(\text{ker})}$ as the row of $\mathbf{H}^{(\text{ker})}$ corresponding to $\mathbf{x}_o'$ when $\mathbf{x}_o'$ is inserted as a row in the $\mathbf{X}$ matrix in the usual kernel procedure. (The elements of $\mathbf{h}_o'^{(\text{ker})}$ are obtained as described in the discussion preceeding equation (3.B.4) in section 3.B.1).

MRR1

The bias and variance equations become a bit more complicated for the three model-robust methods, since they involve two fitting procedures instead of one. The first of these to be dealt with is MRR1, where $\hat{\mathbf{y}}_{\text{MRR1}} = \mathbf{H}^{(\text{MRR1})}\mathbf{y} = [\lambda\mathbf{H}^{(\text{ker})} + (1-\lambda)\mathbf{H}^{(\text{ols})}]\mathbf{y}$. Derived in Appendix D.2, the bias and variance for the MRR1 fitted values (with fixed h and λ) are given by

$$\text{Bias}(\hat{\mathbf{y}}_{\text{MRR1}}) = -\lambda(\mathbf{I} - \mathbf{H}^{(\text{ker})})\mathbf{X}\beta - (\mathbf{I} - \mathbf{H}^{(\text{MRR1})})\mathbf{f}, \qquad (6.B.9)$$

$$\begin{aligned}\text{Var}(\hat{\mathbf{y}}_{\text{MRR1}}) &= \sigma^2\mathbf{H}^{(\text{MRR1})}\mathbf{H}'^{(\text{MRR1})} \qquad (6.B.10)\\ &= \sigma^2\left\{(1-\lambda)\left[\mathbf{I} - \lambda(\mathbf{I} - \mathbf{H}^{(\text{ker})})\right]\mathbf{H}^{(\text{ols})} + \lambda\mathbf{H}^{(\text{MRR1})}\mathbf{H}'^{(\text{ker})}\right\}.\end{aligned}$$

Here h is fixed as h_o, which is the bandwidth that minimizes the AVEMSE based on the bias and variance equations given in (6.B.5) and (6.B.6) (and thus is the same as h_o for the individual kernel procedure). Also, λ is fixed as λ_o, which is the λ that minimizes the AVEMSE calculated from equations (6.B.9) and (6.B.10) above, with $\mathbf{H}^{(\text{ker})}$ having

already been determined by h_o. Note that for $\lambda = 0$ (using no kernel), these equations simplify to equations (6.B.1) and (6.B.2) for OLS. Likewise, for $\lambda = 1$ (using all kernel), they reduce to equations (6.B.5) and (6.B.6) for kernel regression. Also, if the underlying model is chosen such that $\mathbf{f} = \mathbf{0}$, then it is desired to have $\lambda = 0$ to eliminate the bias and to achieve the minimum variance $\sigma^2\mathbf{H}^{(ols)}$. If $\mathbf{X}\beta = \mathbf{0}$, then $\lambda = 1$ is desired to give bias $-(\mathbf{f} - \mathbf{H}^{(ker)}\mathbf{f}) \equiv -(\text{“}\mathbf{y}\text{”} - \mathbf{H}^{(ker)}\text{“}\mathbf{y}\text{”})$ and variance $\sigma^2\mathbf{H}^{(ker)}\mathbf{H}'^{(ker)}$ as in kernel regression. Here “**y**” is thought of as observations directly generated from a nonlinear function, which should be fit better with kernel regression than with OLS.

Again, starting with the underlying model $y_o = \mathbf{x}_o'\beta + f(x_o) + \varepsilon$, the bias and variance of $\hat{y}_o$ can be obtained as

$$\text{Bias}(\hat{y}_{o,\,MRR1}) = -\lambda[\mathbf{x}_o' - \mathbf{h}_o'^{(ker)}\mathbf{X}]\beta + \mathbf{h}_o'^{(MRR1)}\mathbf{f} - f(x_o) \quad , \qquad (6.B.11)$$

$$\text{Var}(\hat{y}_{o,\,MRR1}) = \sigma^2\,\mathbf{h}_o'^{(MRR1)}\mathbf{h}_o^{(MRR1)} \quad , \qquad (6.B.12)$$

where $\mathbf{h}_o'^{(MRR1)} = [(1-\lambda)\mathbf{x}_o'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' + \lambda\mathbf{h}_o'^{(ker)}]$ is the row of a MRR1 hat matrix that would be determined by $\mathbf{x}_o'$, and $\mathbf{h}_o'^{(ker)}$ is the row of a kernel hat matrix determined by $\mathbf{x}_o'$ (same $\mathbf{h}_o'^{(ker)}$ as described previously).

MRR2

Recall that for MRR2, $\hat{\mathbf{y}}_{MRR2} = \mathbf{H}^{(MRR2)}\mathbf{y} = [\mathbf{H}^{(ols)} + \lambda\mathbf{H}_2^{(ker)}(\mathbf{I} - \mathbf{H}^{(ols)})]\mathbf{y}$, from $\hat{\mathbf{y}}_{MRR2} = \mathbf{H}^{(ols)}\mathbf{y} + \lambda\mathbf{H}_2^{(ker)}\mathbf{r}$ (as in (5.C.2)), where $\mathbf{H}_2^{(ker)}$ is the kernel hat matrix from a kernel fit to the residuals **r** from the OLS fit. The bias and variance for $\hat{\mathbf{y}}_{MRR2}$ (with h and λ fixed) are derived in Appendix D.3 and may be expressed as follows:

$$\text{Bias}(\hat{\mathbf{y}}_{MRR2}) = -(\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{f} \qquad (6.B.13)$$

$$\mathrm{Var}(\hat{\mathbf{y}}_{\mathrm{MRR2}}) = \sigma^2 \mathbf{H}^{(\mathrm{MRR2})} \mathbf{H}'^{(\mathrm{MRR2})} \qquad (6.B.14)$$
$$= \sigma^2 [\mathbf{H}^{(\mathrm{ols})} + \lambda^2 \mathbf{H}_2^{(\mathrm{ker})} (\mathbf{I} - \mathbf{H}^{(\mathrm{ols})}) \mathbf{H}_2'^{(\mathrm{ker})}] \, .$$

Here, the bias and variance expressions are a little more complicated, mainly due to the combining of a fit to the data and a fit to residuals in order to construct $\mathbf{H}^{(\mathrm{MRR2})}$. This makes it more difficult to get an intuitive feel for the behavior of these equations. One does notice, though, that the bias is independent of the linear term $\mathbf{X}\boldsymbol{\beta}$ and is affected only by the form of $\mathbf{f}$. This is expected since MRR2 always uses a parametric fit to $\mathbf{X}\boldsymbol{\beta}$, and OLS gives an unbiased estimate, $\mathbf{X}\hat{\boldsymbol{\beta}}_{\mathrm{ols}}$. One must be careful not to be misled by the complexity of these bias and variance expressions. The MRR2 fitting procedure itself is very simple; the complexity arises as an artifact of the steps necessary to develop equations comparable with those of the competing procedures. For the MRR2 results above, h is fixed at the optimal h_o, which is different from the h_o for kernel and MRR1. Now h_o is chosen as the bandwidth that minimizes the AVEMSE for the kernel fit to the residuals $\mathbf{r}$ from the OLS fit. This kernel fit may be expressed as $\hat{\mathbf{r}} = \mathbf{H}_2^{(\mathrm{ker})}\mathbf{r}$, and the AVEMSE is calculated from the following equations (derived in Appendix D.3):

$$\mathrm{Bias}(\hat{\mathbf{r}}) = -(\mathbf{I} - \mathbf{H}_2^{(\mathrm{ker})})(\mathbf{I} - \mathbf{H}^{(\mathrm{ols})})\mathbf{f} \qquad (6.B.15)$$

$$\mathrm{Var}(\hat{\mathbf{r}}) = \sigma^2 \mathbf{H}_2^{(\mathrm{ker})} (\mathbf{I} - \mathbf{H}^{(\mathrm{ols})}) \mathbf{H}_2'^{(\mathrm{ker})} \, . \qquad (6.B.16)$$

The mixing parameter λ is fixed as λ_o, which minimizes the AVEMSE calculated from (6.B.13) and (6.B.14), with $\mathbf{H}_2^{(\mathrm{ker})}$ already determined by h_o.

The bias and variance of $\hat{y}_o$ can be obtained in a similar fashion to OLS, kernel, and MRR1 as

$$\mathrm{Bias}(\hat{y}_{o,\,\mathrm{MRR2}}) = \mathbf{h}_o'^{(\mathrm{MRR2})}\mathbf{f} - f(x_o) \, , \qquad (6.B.17)$$

$$\mathrm{Var}(\hat{y}_{o,\,\mathrm{MRR2}}) = \sigma^2\, \mathbf{h}_o'^{(\mathrm{MRR2})}\, \mathbf{h}_o^{(\mathrm{MRR2})}\ , \qquad (6.B.18)$$

where $\mathbf{h}_o'^{(\mathrm{MRR2})} = [\mathbf{x}_o'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' + \lambda \mathbf{h}_{o,2}'^{(\mathrm{ker})}(\mathbf{I} - \mathbf{H}^{(\mathrm{ols})})]$ is the row of a MRR2 hat matrix that would be determined by $\mathbf{x}_o'$, and $\mathbf{h}_{o,2}'^{(\mathrm{ker})}$ is the row of a kernel hat matrix determined by $\mathbf{x}_o'$ when fitting the *residuals* from the OLS fit. Note that $\mathbf{h}_{o,2}'^{(\mathrm{ker})}$ is different from $\mathbf{h}_o'^{(\mathrm{ker})}$ due to the different bandwidths resulting from fitting the data (for kernel) vs. fitting the residuals (for MRR2).

PLR

The final procedure is PLR, where fitted values are expressed as $\hat{\mathbf{y}}_{\mathrm{PLR}} = \mathbf{H}^{(\mathrm{PLR})}\mathbf{y} = \mathbf{X}_{\mathrm{P}}\hat{\boldsymbol{\beta}}_{\mathrm{PLR}} + \hat{\mathbf{f}}_{\mathrm{PLR}}$ (as in (5.A.3)), where $\mathbf{X}_{\mathrm{P}}$ is the $\mathbf{X}$ matrix without a column of ones. As mentioned in section 5.A (on PLR), the hat matrix $\mathbf{H}^{(\mathrm{PLR})}$ takes on two different forms depending on which nonparametric fitting technique is used for the residual fit (kernel or LPR). These two forms for $\mathbf{H}^{(\mathrm{PLR})}$ are given in equations (5.A.5) and (5.A.6). The formulas for the bias and variance of $\hat{\mathbf{y}}_{\mathrm{PLR}}$ (with h fixed) for each of these cases are derived in Appendix D.4 and are presented below. (Note that the bias expressions are the same for both cases, but $\mathbf{H}^{(\mathrm{PLR})}$ differs for each technique; also, recall that $\widetilde{\mathbf{X}}$ is *always* defined as $\widetilde{\mathbf{X}} = (\mathbf{I} - \mathbf{H}_{\mathrm{P}}^{(\mathrm{ker})})\mathbf{X}_{\mathrm{P}}$ (i.e., $\widetilde{\mathbf{X}}$ always uses the kernel hat matrix)).

When using kernel regression for the residual fit, one obtains

$$\mathrm{Bias}(\hat{\mathbf{y}}_{\mathrm{PLR}}) = -(\mathbf{I} - \mathbf{H}^{(\mathrm{PLR})})\mathbf{f}, \qquad (6.B.19)$$

$$\begin{aligned}\mathrm{Var}(\hat{\mathbf{y}}_{\mathrm{PLR}}) &= \sigma^2 \mathbf{H}^{(\mathrm{PLR})}\mathbf{H}'^{(\mathrm{PLR})} \qquad (6.B.20)\\ &= \sigma^2[\mathbf{H}_{\mathrm{P}}^{(\mathrm{ker})}\,\mathbf{H}_{\mathrm{P}}'^{(\mathrm{ker})} + \mathbf{H}_{\mathrm{P}}^{(\mathrm{ker})}(\mathbf{I} - \mathbf{H}_{\mathrm{P}}^{(\mathrm{ker})})'\mathbf{P}_{\widetilde{\mathbf{X}}} + \mathbf{P}_{\widetilde{\mathbf{X}}}(\mathbf{I} - \mathbf{H}_{\mathrm{P}}^{(\mathrm{ker})})\mathbf{H}_{\mathrm{P}}'^{(\mathrm{ker})} +\\ &\qquad \mathbf{P}_{\widetilde{\mathbf{X}}}(\mathbf{I} - \mathbf{H}_{\mathrm{P}}^{(\mathrm{ker})})(\mathbf{I} - \mathbf{H}_{\mathrm{P}}^{(\mathrm{ker})})'\mathbf{P}_{\widetilde{\mathbf{X}}}]\ ,\end{aligned}$$

where $\mathbf{P}_{\tilde{\mathbf{X}}} = \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'$ and $\mathbf{H}^{(PLR)}$ is as in (5.A.5). Using local polynomial regression for the residual fit yields

$$\text{Bias}(\hat{\mathbf{y}}_{PLR}) = -(\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f}, \tag{6.B.21}$$

$$\begin{aligned}\text{Var}(\hat{\mathbf{y}}_{PLR}) &= \sigma^2\mathbf{H}^{(PLR)}\mathbf{H}'^{(PLR)} \\ &= \text{(expression in Appendix D.4 (a))},\end{aligned} \tag{6.B.22}$$

where $\mathbf{H}^{(PLR)}$ is as in (5.A.6). These equations are similar to those for MRR2 in that they are rather complex (especially the variance), and the bias takes the same form of being dependent only on the form of $\mathbf{f}$. Here h is fixed as h_o, the bandwidth minimizing the AVEMSE based on equations (6.B.19) and (6.B.20), or (6.B.21) and (6.B.22) (depending on the fitting technique used for the residuals).

Now the bias and variance formulas of the fitted value $\hat{y}_o$ at any individual location $\mathbf{x}_{o,p}' = (x_o\ x_o^2 \cdots)$ (for PLR) can be obtained through similar arguments to those used to obtain the bias and variance expressions above. Starting with the underlying model written in the form $y_o = \mathbf{x}_{o,p}'\beta_p + f(x_o) + \varepsilon$, the resulting formulas are as follows. When using kernel regression for the residual fit,

$$\text{Bias}(\hat{y}_{o,\,PLR}) = \mathbf{h}_o'^{(PLR)}\mathbf{f} - f(x_o)\ , \tag{6.B.23}$$

$$\text{Var}(\hat{y}_{o,\,PLR}) = \sigma^2\,\mathbf{h}_o'^{(PLR)}\mathbf{h}_o^{(PLR)}\ , \tag{6.B.24}$$

where $\mathbf{h}_o'^{(PLR)} = [\mathbf{h}_{o,p}'^{(ker)} + (\mathbf{x}_{o,p}' - \mathbf{h}_{o,p}'^{(ker)}\mathbf{X}_P)(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{H}_P^{(ker)})]$ is the row of a PLR hat matrix that would be determined by $\mathbf{x}_{o,p}'$, and $\mathbf{h}_{o,p}'^{(ker)}$ is the row of a kernel hat matrix determined strictly by $\mathbf{x}_{o,p}'$. Finally, when using LPR for the residual fit,

$$\text{Bias}(\hat{y}_{o,\,\text{PLR}}) = \mathbf{h}_o'^{(\text{PLR})}\mathbf{f} - f(x_o) \quad , \tag{6.B.25}$$

$$\text{Var}(\hat{y}_{o,\,\text{PLR}}) = \sigma^2\, \mathbf{h}_o'^{(\text{PLR})}\, \mathbf{h}_o^{(\text{PLR})} \quad , \tag{6.B.26}$$

where now $\mathbf{h}_o'^{(\text{PLR})} = [\mathbf{h}_{o,p}'^{(\text{LPR})} + (\mathbf{x}_{o,p}' - \mathbf{h}_{o,p}'^{(\text{LPR})}\mathbf{X}_P)(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}\,(\mathbf{I} - \mathbf{H}_P^{(\text{ker})})]$ is the row of a PLR hat matrix that would be determined by $\mathbf{x}_{o,p}'$, and $\mathbf{h}_{o,p}'^{(\text{LPR})}$ is the row of a LPR hat matrix determined by $\mathbf{x}_{o,p}'$ when fitting to the residuals in the PLR procedure.

With the bias and variance equations (6.B.1)-(6.B.26), the MSE's for each of the five procedures can easily be obtained. By averaging the MSE's for a given procedure across the fitted values at the data points, the average MSE (AVEMSE) can now be obtained. This is the criterion used for determining the "optimal" h_o and λ_o for the different procedures. Also, by averaging the MSE's of many (1000) locations across the range of the data, a good estimate for the integrated MSE (INTMSE) can be obtained. This is the key diagnostic for comparing the performances of the five procedures for several generated data sets in the next section. The reason AVEMSE is used instead of INTMSE to choose h_o and λ_o is to allow for fairer comparisons with data-driven methods for choosing h and λ. In other words, it seems more appropriate to determine the optimal fit (which serves as the basis for comparisons) based on only the data points rather than on a global criterion, because this is what *data*-driven methods are restricted to.

6.C Examples

6.C.1 Introduction

The five fitting techniques (OLS, LLR, MRR1, MRR2, and PLR) are now compared for three different generated data sets and one actual data set. In all situations, the X-data is scaled to be between 0 and 1 in order to have a good reference for the behavior of the different techniques across different data sets. For example, bandwidth

values may be interpreted the same for all data sets, whereas it would be difficult to compare bandwidth magnitudes across different data sets if they were all on different scales. Graphical comparisons are provided by plots of the regression curves for each procedure, and numerical comparisons are provided by several performance diagnostics. The diagnostics include df_{model}= trace(Hat matrix), SSE, PRESS, PRESS*, and INTMSE. For "good" performance, it is desired to have all of these as small as possible. Recall that $df_{model} = tr(\mathbf{H})$ can be interpreted as "the number of parameters that would be needed to obtain a comparable parametric fit", and in this sense measures the "complexity" of the particular fit of interest. Of course, $df_{model} = p$ for OLS since this is the parametric fitting technique used here. Also, h and λ are chosen as the optimal h_o and λ_o based on minimizing AVEMSE (the average MSE of the fits at the actual data points), so each fitting technique is doing its best to keep AVEMSE as low as possible. However, since one is usually interested in the fit of the regression curve across the entire range of the data, a "global" measure is more appropriate than AVEMSE in actually making the final comparisons of the different procedures. Thus, the key diagnostic for the comparisons is taken to be INTMSE (which provides the best measure of the trade-off between bias and variance for the entire curve). Also, note that INTMSE is based on theoretical formulas and does not depend on the particular data generated for each of the examples. The other diagnostics are data-dependent and would change for different generated data sets (from the same underlying model). Thus, the values of df_{model}, SSE, PRESS, and PRESS* are used as supplemental diagnostics to INTMSE for these examples (since each example is but one of many possible generated sets of data for the particular underlying model). More faith may be placed in these data-dependent values if these values were obtained as average values over many simulated data sets for each underlying model. Some such simulations are provided in Chapter 8 (for df_{model} and PRESS). A final note is that LLR (local linear regression) is used as the nonparametric fitting technique in all of the procedures in order to remove boundary bias problems inherent in kernel regression. For

the first example, however, an individual kernel fit is also shown to illustrate how LLR can improve the fit at the boundaries.

6.C.2 Example 1

For this example, data is generated from the underlying model

$$y = 2(X - 5.5)^2 + 5X + 3.5\sin\left(\frac{\pi(X-1)}{2.25}\right) + \varepsilon \qquad (6.C.1)$$

at ten evenly spaced X-values from 1 to 10, where $\varepsilon \sim N(0,16)$. The term involving the sine function introduces a deviation from a quadratic model, and the argument of the sine function results in the sine completing two full periods over the interval [1,10]. This model was introduced by Einsporn (1987), who studied fitting techniques when changing the amplitude of the sine function (the 3.5 here). Einsporn found that the usual lack of fit test has power of only .226 at $\alpha = .05$ when the user has specified a quadratic model, namely $y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$, instead of (6.C.1). In other words, the user would believe the quadratic model is adequate and base inferences on that model, resulting in possible misleading conclusions. This is a good example of a case where the specified model (the quadratic) is adequate throughout most, but not all of the data. Figure 6.C.1 shows the *true* underlying model without the error term. Notice the "dip" between $X = 3$ and $X = 6$, which an OLS fit to the specified model is likely to be unable to capture. A kernel or local linear fit may be used to help capture this dip, but recall that this would ignore the known quadratic structure and result in a fit higher than necessary in variance.

The above observations are illustrated in Figures 6.C.2 (a), (b), and (c), which show the true curve and the raw data generated from (6.C.1) (with the error term), along with the quadratic OLS fit (figure (a)), the kernel fit (figure (b)), and the LLR fit (figure (c)). Note that the OLS fit is smooth (low variance), but fails to capture the dip and does not fit well at several other points (high biases). The kernel fit (based on $h_o = .086$) fits

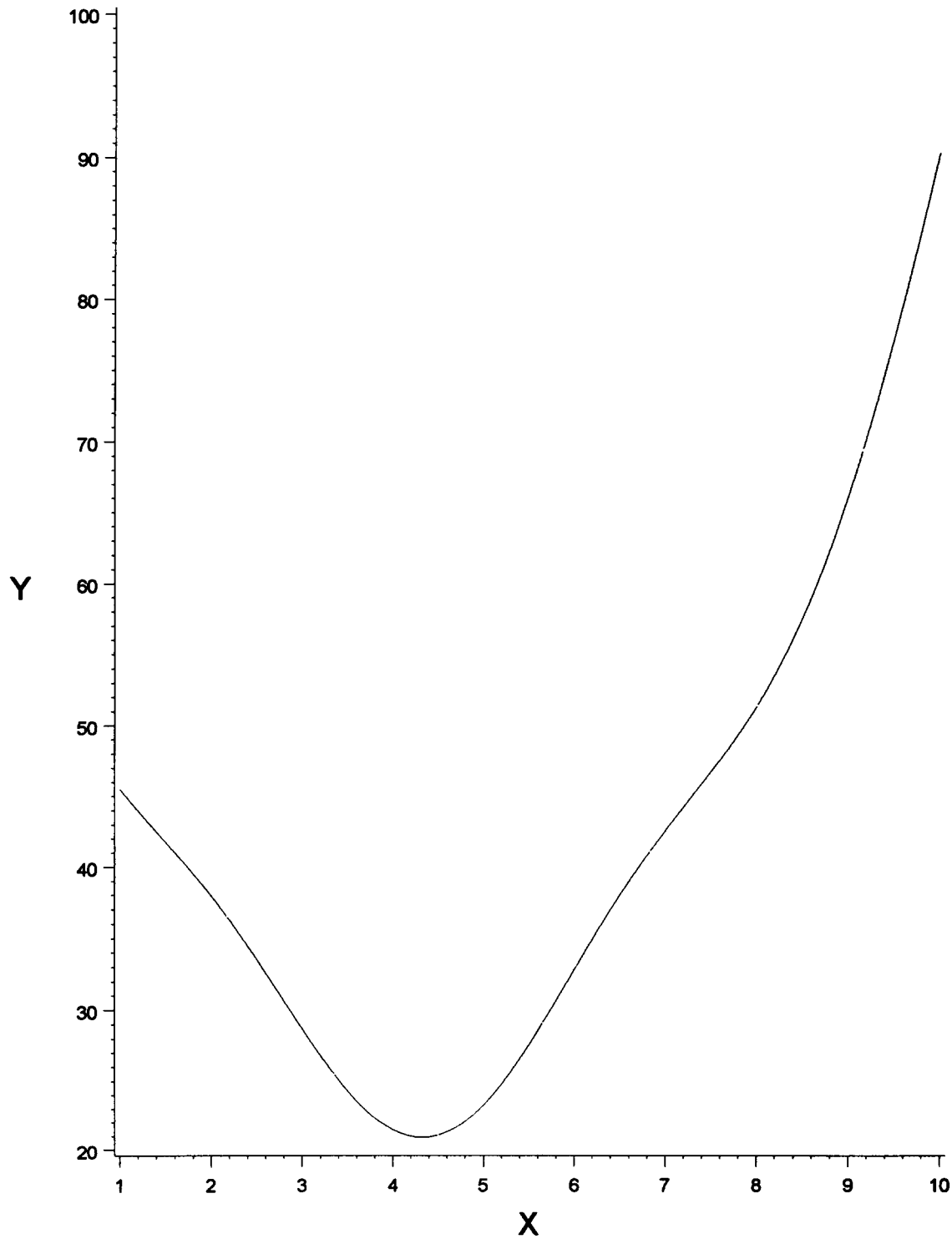


Figure 6.C.1. True underlying curve from Equation (6.C.1) for Example 1.

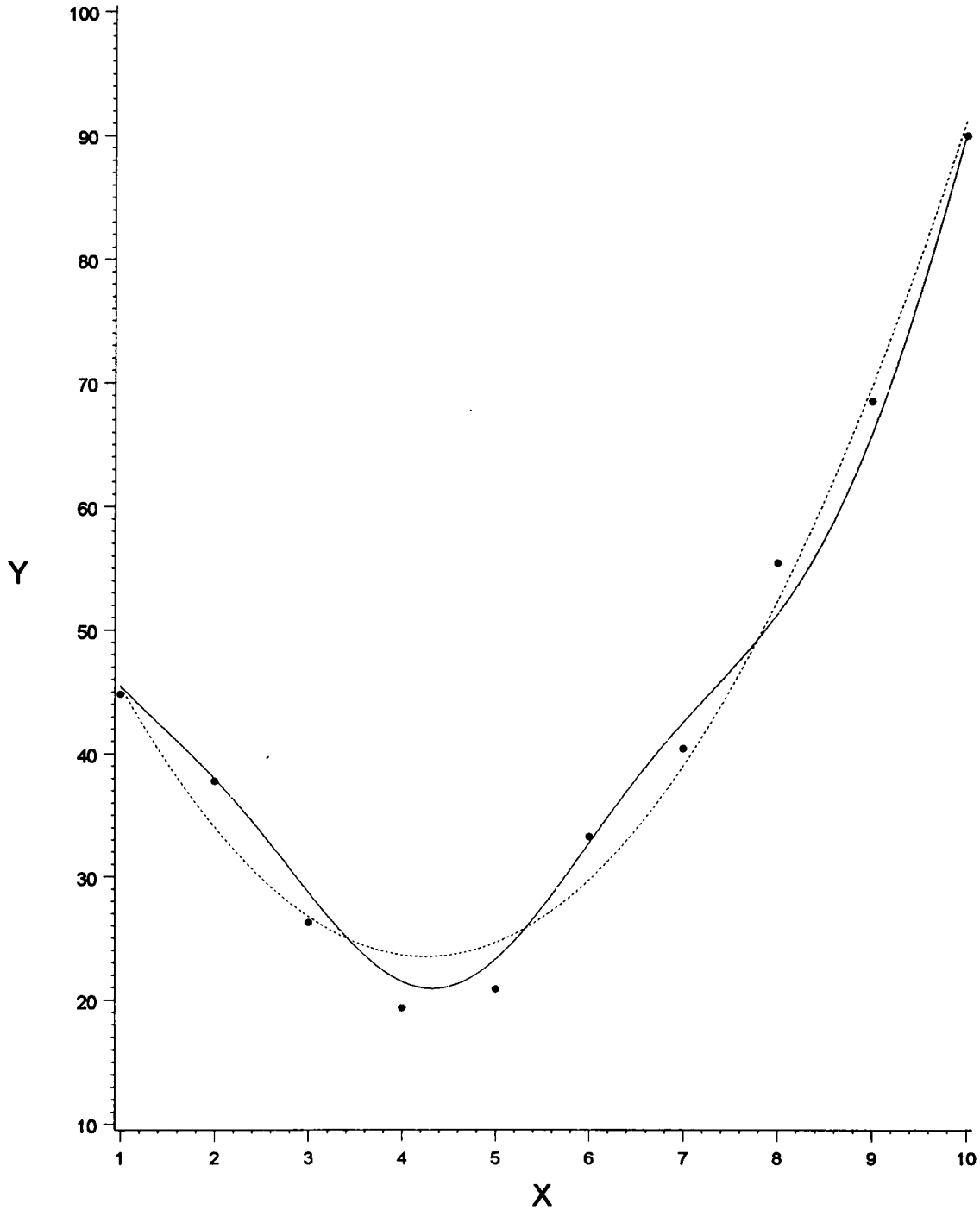


Figure 6.C.2 (a). Plot of generated data for Example 1, with quadratic OLS fit.

[••• Raw data —— True curve ····· OLS]

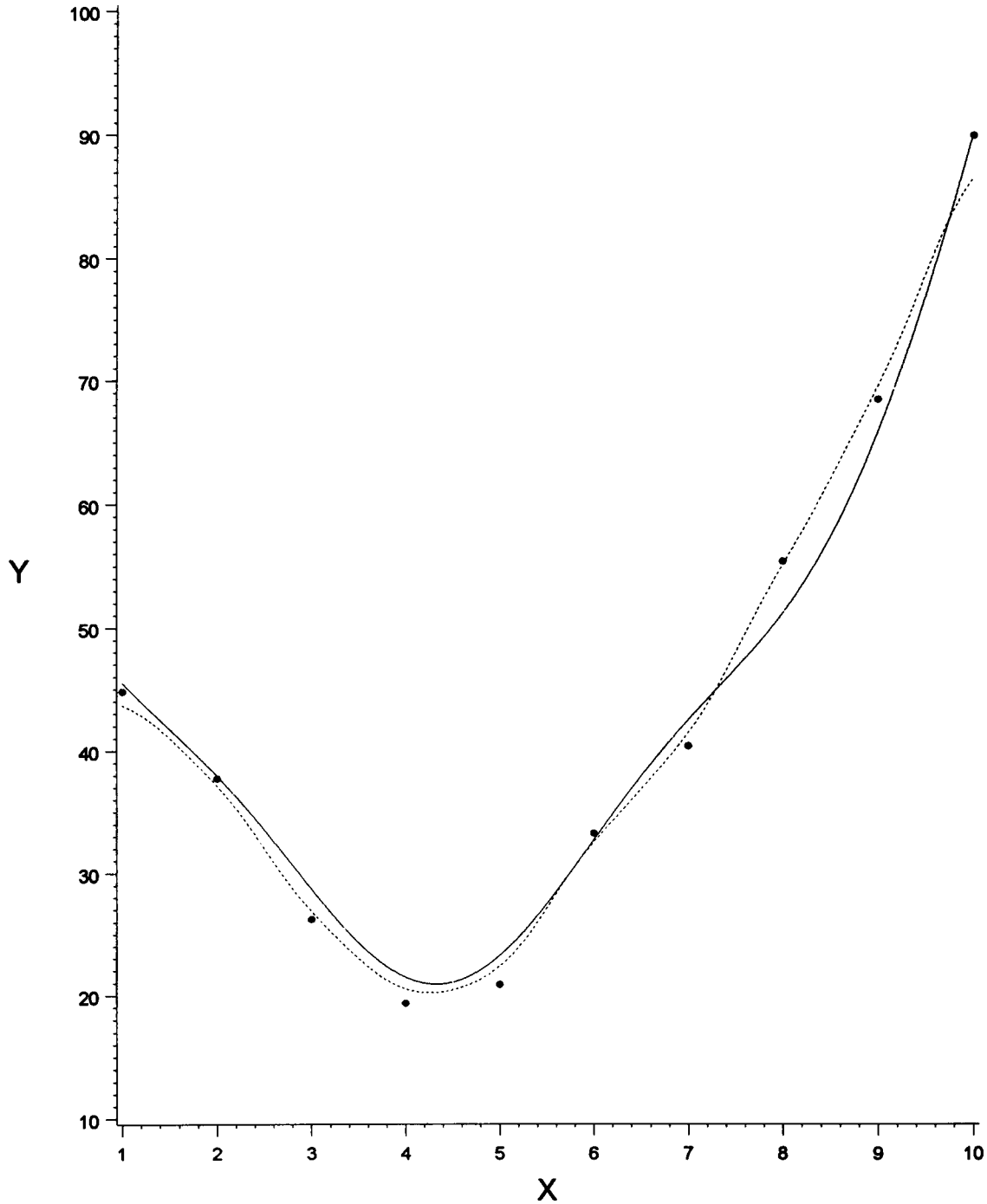


Figure 6.C.2 (b). Plot of generated data for Example 1, with Kernel fit.

[••• Raw data —— True curve ····· Kernel]

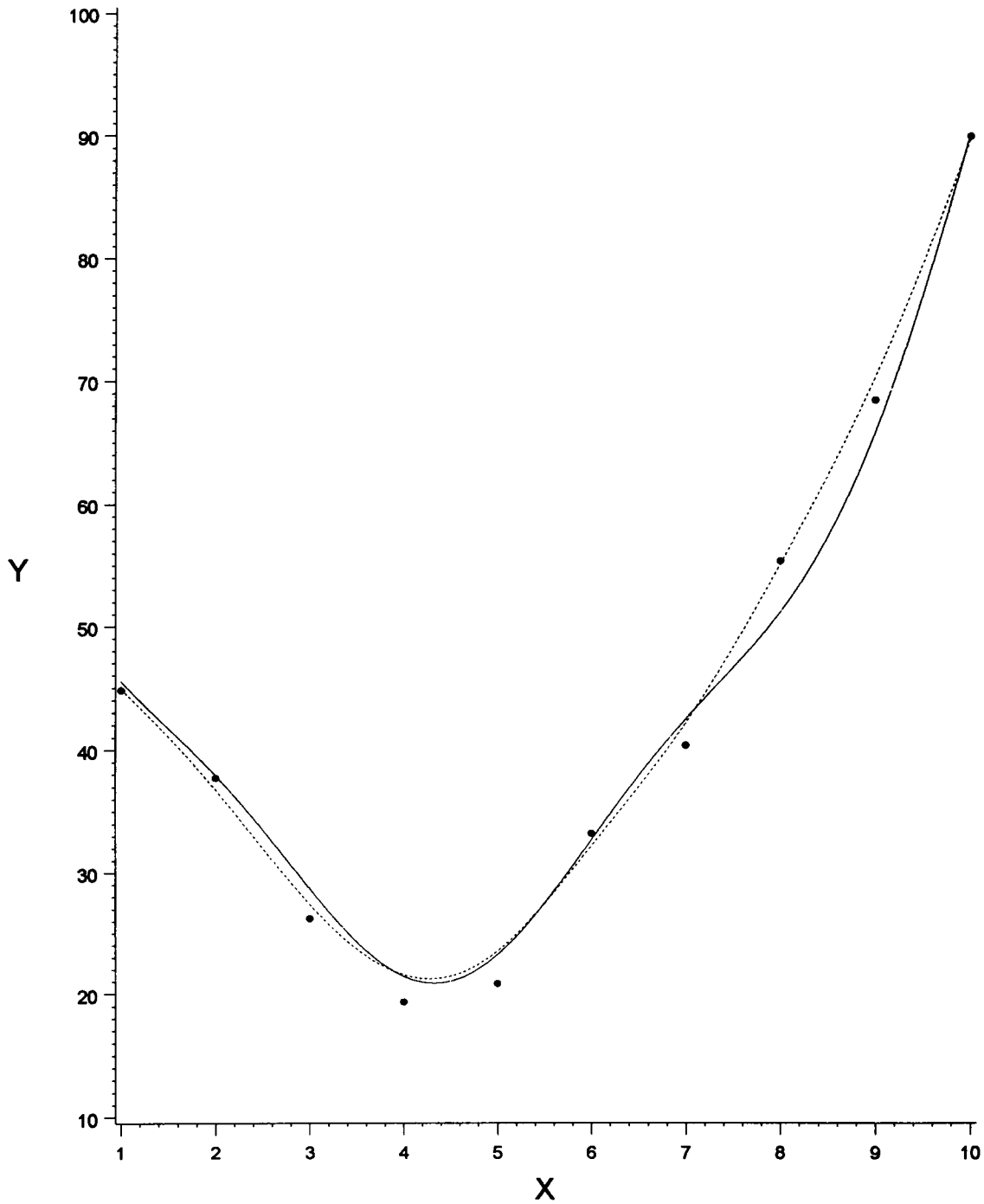


Figure 6.C.2 (c). Plot of generated data for Example 1, with Local Linear fit.

[••• Raw data —— True curve ····· LLR]

close to the data points, but is not very smooth (low bias, high variance). Notice how the kernel fits too low at the boundaries, illustrating the boundary bias problem. The LLR fit (based on $h_o = .115$) solves this boundary problem, and supports the use of LLR instead of kernel as the nonparametric fitting technique for all of the results to follow. Figure 6.C.3 displays how the MRR1 fit, based on a λ_o of .503 (almost even weight on OLS and LLR), combines the OLS and LLR fits, and Figure 6.C.4 gives just the MRR1 fit along with the true curve. The MRR1 fit maintains the smoothness (low variance) of OLS, while using LLR information to pull the fit closer to the data where needed (lowering bias). Also, notice that the MRR1 fit is always between the OLS and LLR fits (actually at $\lambda \approx 50\%$ of the distance from OLS to LLR). As discussed in section 5.B.1, if either OLS or LLR fits poorly at a certain data point, then MRR1 may be unable to correct for this inadequacy. This is illustrated to a certain degree at points $X = 2$, $X = 6$, and at the dip, where OLS does not fit well. MRR2 and PLR should do as well or better at fitting these locations.

Figures 6.C.5 and 6.C.6 give the individual regression fits for MRR2 and PLR, respectively, while Figure 6.C.7 gives these two fits along with that for MRR1. The MRR2 fit is based on $h_o = .152$ and $\lambda_o = .713$, while the PLR fit has $h_o = .153$. MRR2 and PLR give fits very similar to each other, and these fits are on the whole slightly better than the fit from MRR1. The most noticeable difference is at the dip in the data, where MRR2 and PLR give slightly improved fits. Notice that all three model-robust methods give much better fits than the individual OLS fit, and it is shown shortly that these model-robust procedures have much lower variances than LLR (even though LLR does look somewhat smooth). These are precisely the improvements hoped for from the proposed methods. These improvements are supported numerically shortly, but first some brief illustrations are given as to exactly how the MRR2 and PLR fits are constructed.

For MRR2, recall that the first step is to obtain a parametric fit to the data; this is the quadratic OLS fit in Figure 6.C.2 (a). Then the residuals from this parametric fit are fit using a nonparametric technique; this is the LLR fit in Figure 6.C.8, where the residuals from the OLS fit are plotted on a wider scale than the data in order to show the structure

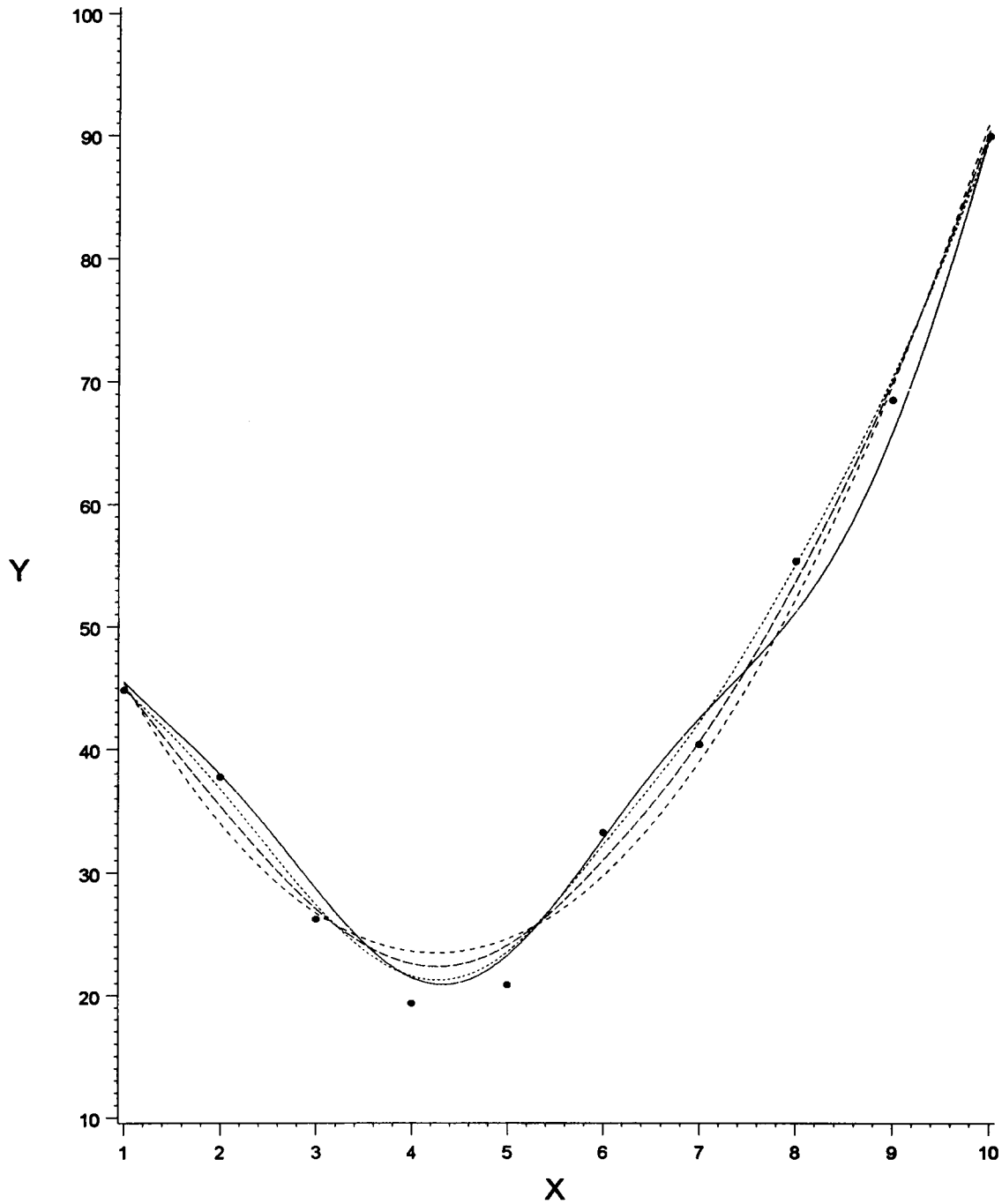


Figure 6.C.3. Plot of generated data for Example 1, with quadratic OLS, LLR, and MRR1 regression curves.

[••• Raw data —— True curve - - - OLS ····· LLR ---- MRR1]

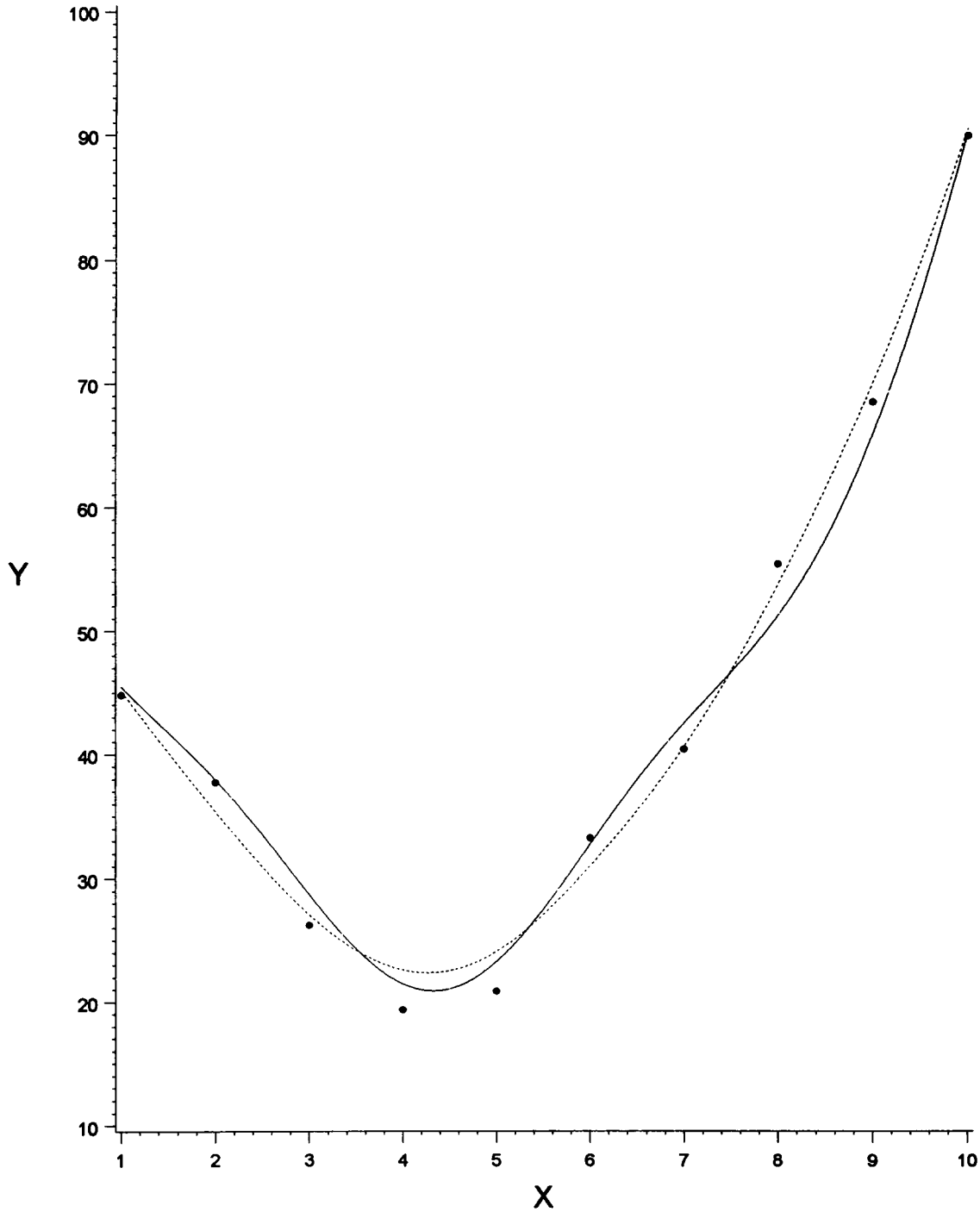


Figure 6.C.4. Plot of generated data for Example 1, with MRR1 fit.

[••• Raw data —— True curve ····· MRR1]

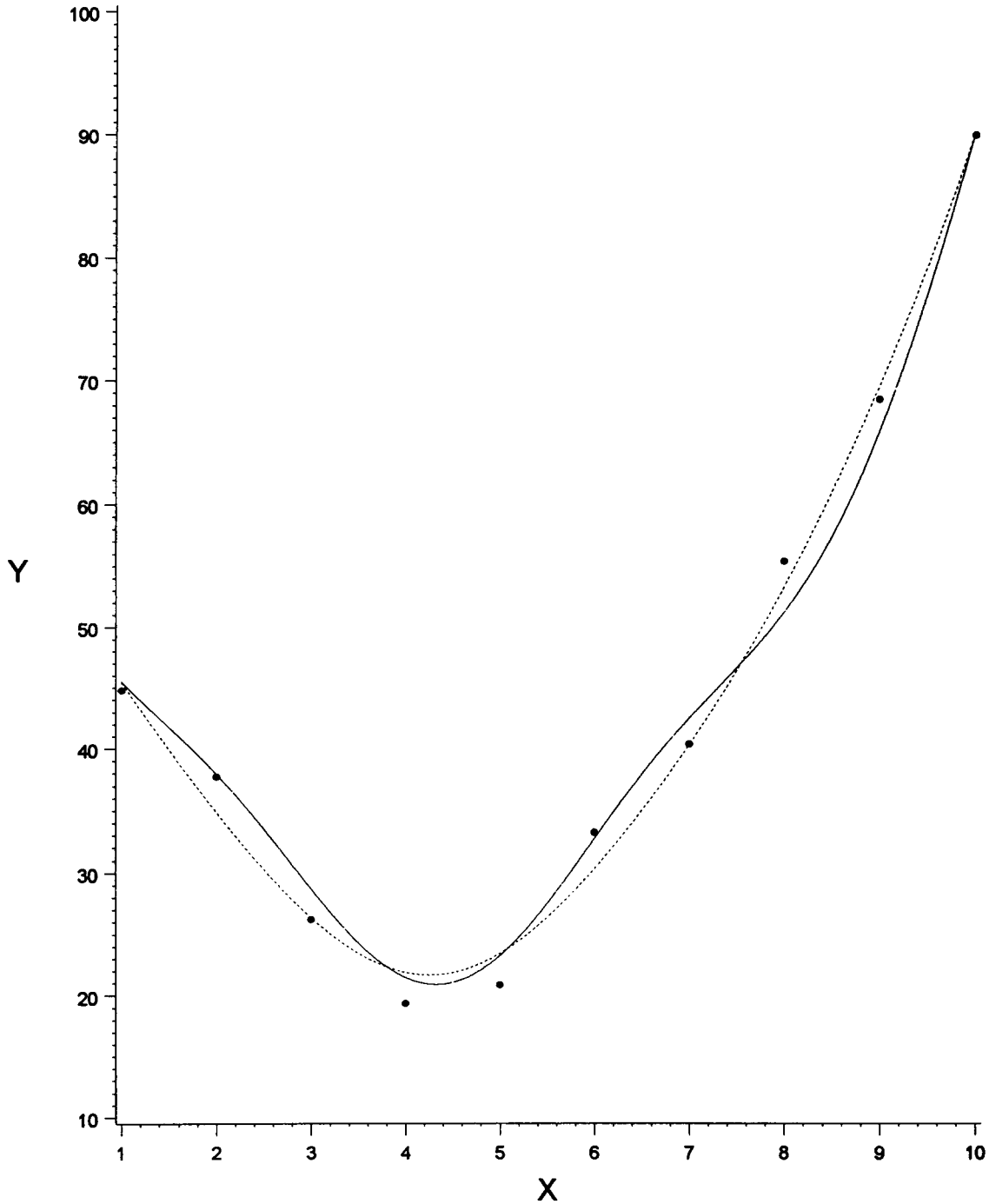


Figure 6.C.5. Plot of generated data for Example 1, with MRR2 fit.

[••• Raw data —— True curve ····· MRR2]

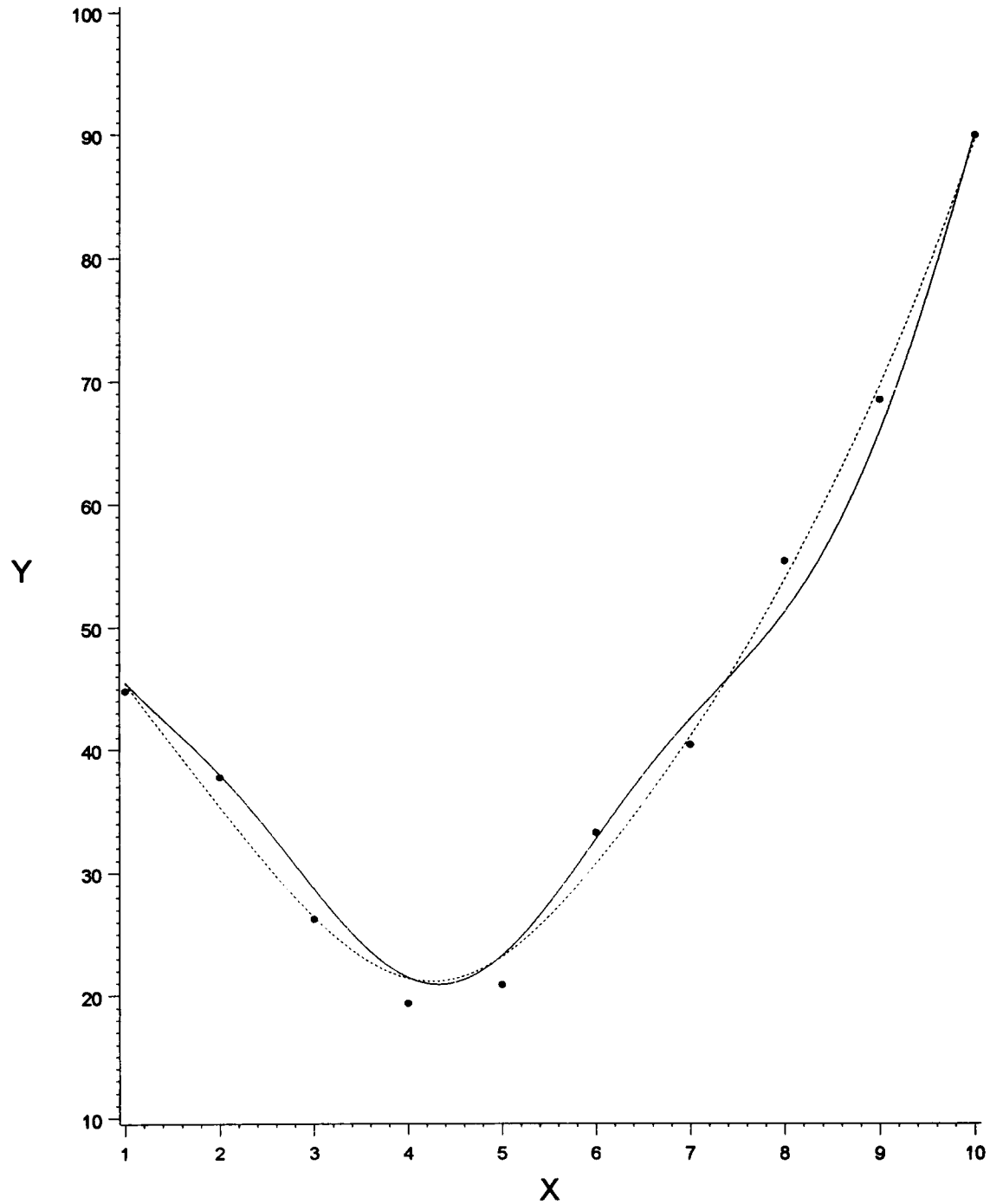


Figure 6.C.6. Plot of generated data for Example 1, with PLR fit.

[••• Raw data —— True curve ····· PLR]

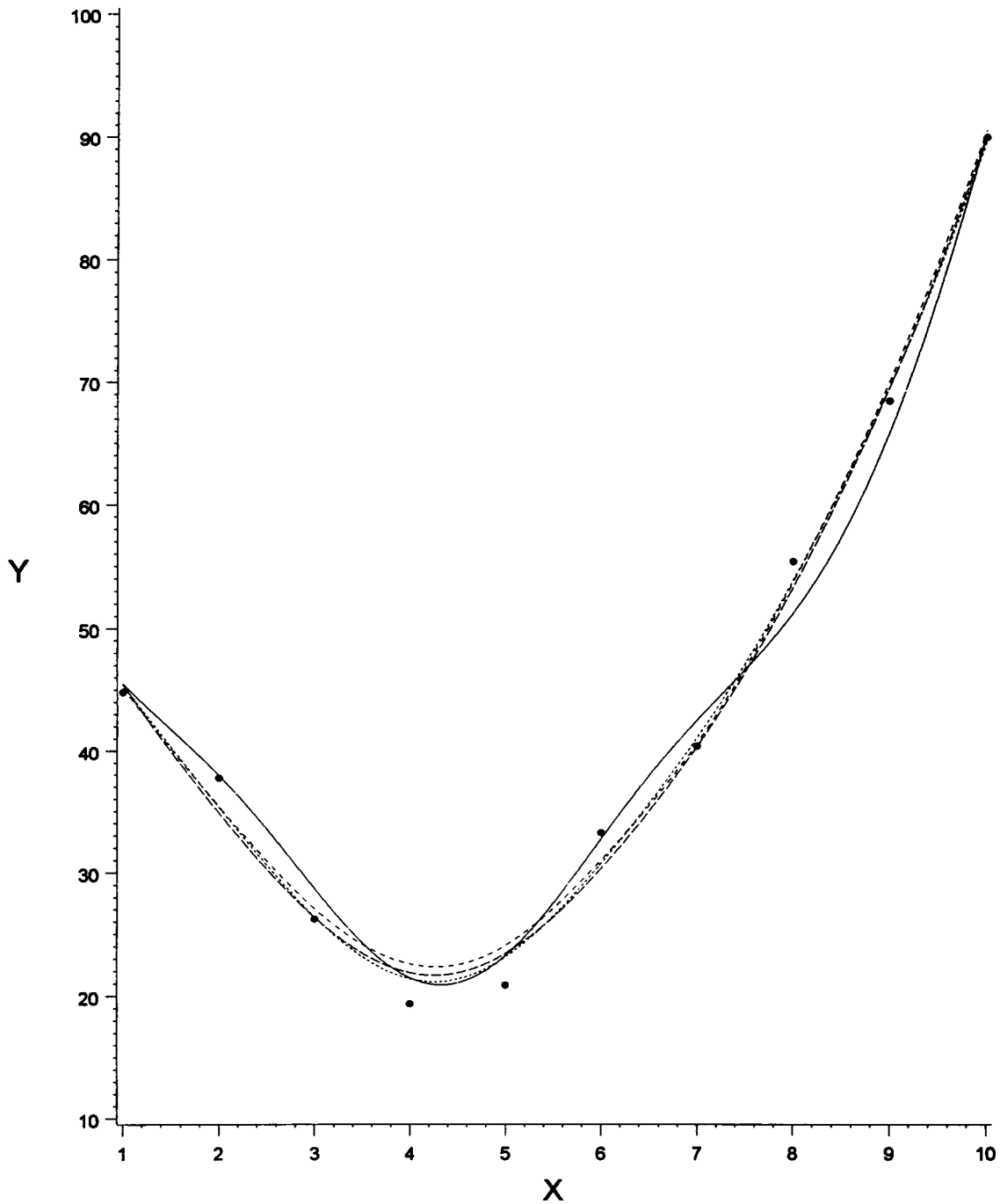


Figure 6.C.7. Plot of generated data for Example 1, with MRR1, MRR2, and PLR regression curves (based on quadratic parametric models).

[••• Raw data —— True curve - - - MRR1 ----- MRR2 ····· PLR]

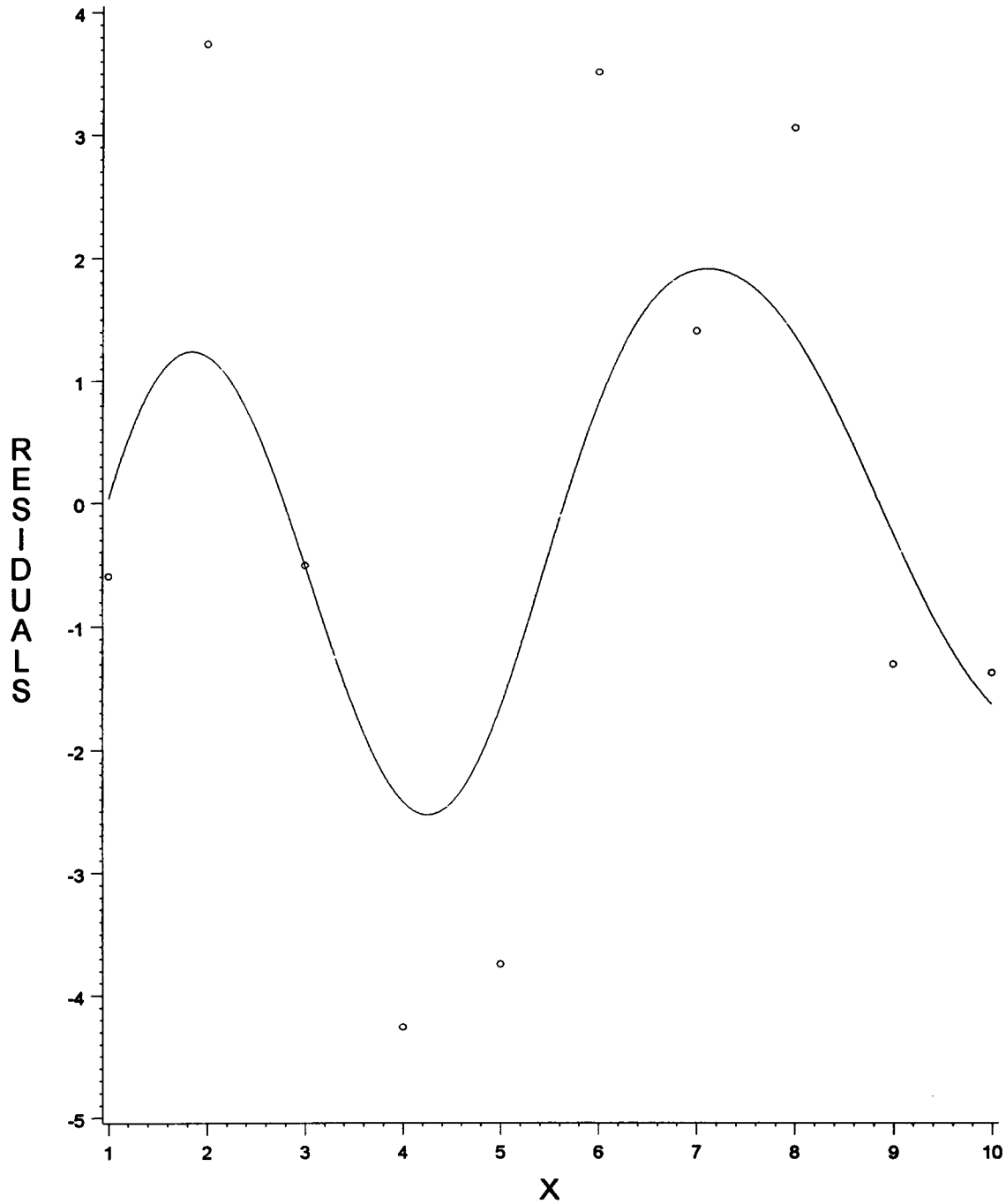


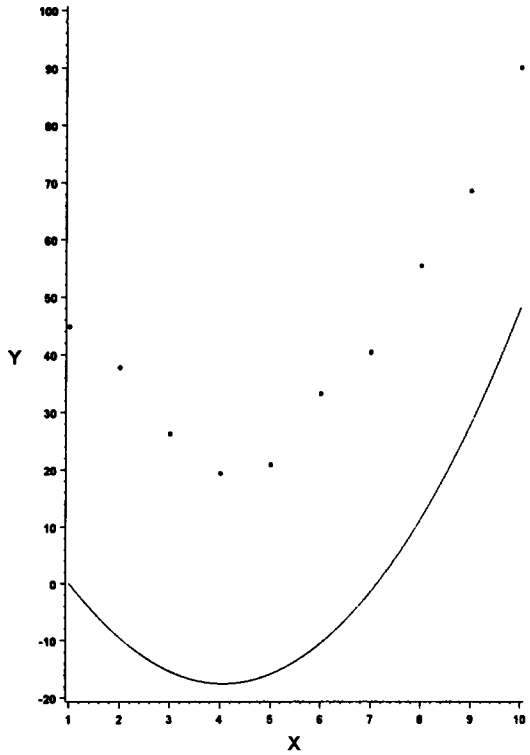
Figure 6.C.8. MRR2 LLR fit to residuals from a quadratic OLS fit, for Example 1.

[ooo Residuals —— LLR]

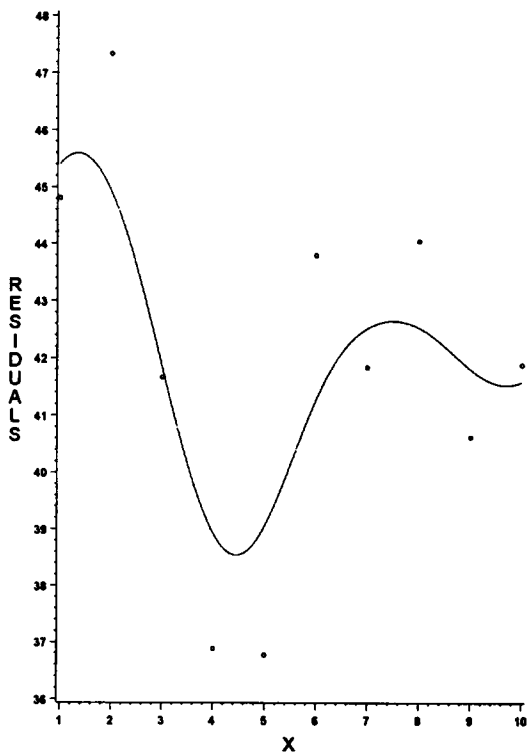
of the LLR fit. A certain proportion (λ = .713 here) of the fit to the residuals is then added to the OLS fit to the data to give the final MRR2 fit in Figure 6.C.5.

Recall that the PLR fit is obtained through adding together simultaneous parametric and nonparametric fits. The parametric fit is obtained by first adjusting $\mathbf{X}_P$ and $\mathbf{y}$ for the nonparametric portion of $\mathbf{X}_P$, and then regressing the partial residual $(\mathbf{I} - \mathbf{H}_P^{(ker)})\mathbf{y}$ on the partial residual $(\mathbf{I} - \mathbf{H}_P^{(ker)})\mathbf{X}_P$. This fit always intersects the y-axis at zero, since no intercept term is contained in $\mathbf{X}_P$. This parametric fit is given in Figure 6.C.9 (a), with $\mathbf{H}_P^{(ker)}$ based on h_o = .153. Note that the regression curve is not even close to the data, but does display the general parametric form of the underlying model. The "jump" from this curve to the data, along with any special structure in the data, is captured by the nonparametric fit to the residuals from this parametric fit, as illustrated in Figure 6.C.9 (b). This fit is determined by $\mathbf{H}_P^{(LLR)}$, which is based on the same h_o = .153. Notice that the residuals are scattered around approximately 42 in magnitude, so when this entire residual fit is added to the fit to the data (to give the PLR fit in Figure 6.C.6), it corrects for the insufficiency caused by the lack of intercept term.

Table 6.C.1 gives the numerical results of interest for this example. The key diagnostic INTMSE is smallest for MRR2, and is quite a bit lower for the model-robust methods than for the individual OLS and LLR procedures. In comparing the model-robust methods, note that MRR2 is uniformly better than MRR1 and almost uniformly better than PLR (except for SSE). Also, df_{model} is lowest for MRR2 for the three model-robust methods, indicating a relatively simpler regression fit, and is much lower than for the individual LLR fit. Figures 6.C.10 (a)-(c) display the squared bias, the variance, and the MSE curves for the model-robust methods. Figure 6.C.10 (a) shows a bias problem in MRR1 (due to the large bias in OLS, especially at the dip (X = 4)). Figure 6.C.10 (b) shows the larger variance for PLR (due to its inclusion of an entire LLR fit), and the presence of larger variances at the boundaries for each of the procedures. This variance increase at the boundaries should not be considered a major problem with LLR, as this phonomenon is also present when kernel regression is used. Figure 6.C.10 (c) shows the



(a) PLR parametric



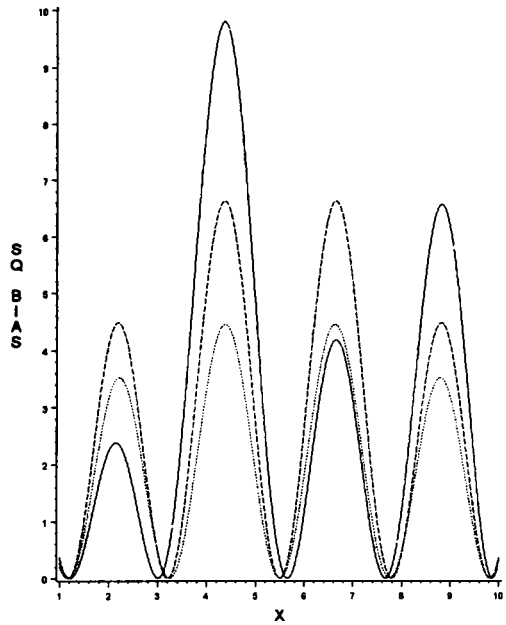
(b) PLR nonparametric

Figure 6.C.9 (a), (b). PLR parametric fit based on quadratic model, and PLR nonparametric fit (using LLR) to the residuals from the parametric fit of (a), for Example 1.

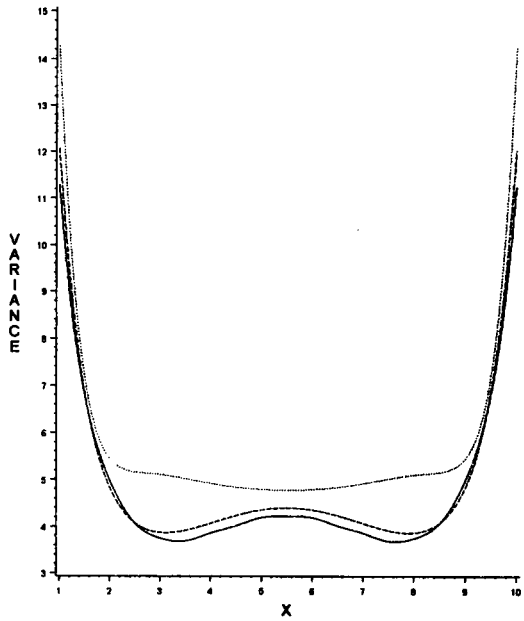
[••• Raw data ooo Residuals]

Table 6.C.1. Bandwidth, mixing parameter, and performance diagnostics for Example 1. Bandwidth and mixing parameter minimize AVEMSE. Key values for comparisons are underlined.

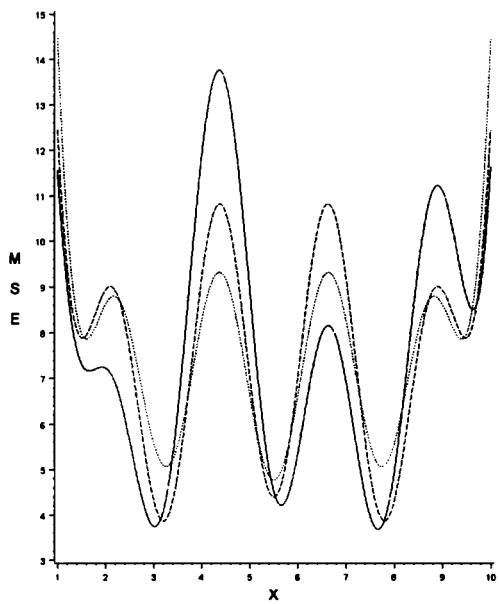
	h_o	λ_o	df_{model}	SSE	PRESS	PRESS*	INTMSE
OLS	---	---	3	74.24	135.03	19.29	9.42
LLR	.115	---	6.23	22.63	198.73	54.05	8.67
MRR1	.115	.503	4.67	37.97	111.35	20.90	**7.66**
MRR2	.152	.713	**4.61**	36.08	**106.75**	**19.81**	**7.57**
PLR	.153	---	5.28	**26.25**	179.61	38.07	**7.60**



(a) Squared Bias



(b) Variance



(c) MSE

Figure 6.C.10 (a)-(c). Squared Bias, Variance, and MSE plots for MRR1, MRR2, and PLR, for Example 1.

[—— MRR1 ---- MRR2 ····· PLR]

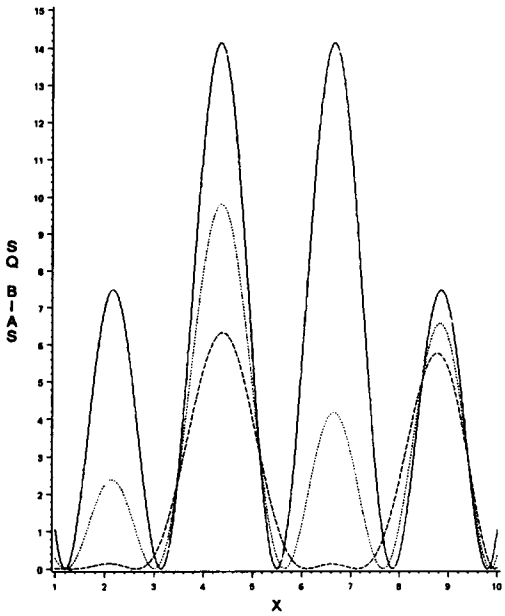
MSE structure obtained by adding together the plots in 6.C.10 (a) and 6.C.10 (b). The three procedures behave rather similarly. Of note though are the high MSE for MRR1 at the dip (X = 4, 5), and the higher MSE for PLR at the endpoints. Also, except for the endpoints, the MSE plots for MRR2 and PLR are relatively close, with PLR slightly better in the center of the data, and MRR2 better elsewhere. Figures 6.C.11 (a)-(c) display the squared bias, the variance, and the MSE curves for OLS, LLR, and MRR1. The bias problem of OLS and the variance problem of LLR show up clearly in figures (a) and (b). However, the key observation to be made here is in Figure 6.C.11 (c), which shows that the model-robust procedure MRR1 greatly reduces both of these problems at the same time. MRR1 appears to be capturing the best of both individual procedures: the small variance of OLS and the small bias of LLR. This is exactly what was desired when the model-robust procedures were developed.

6.C.3 Example 2

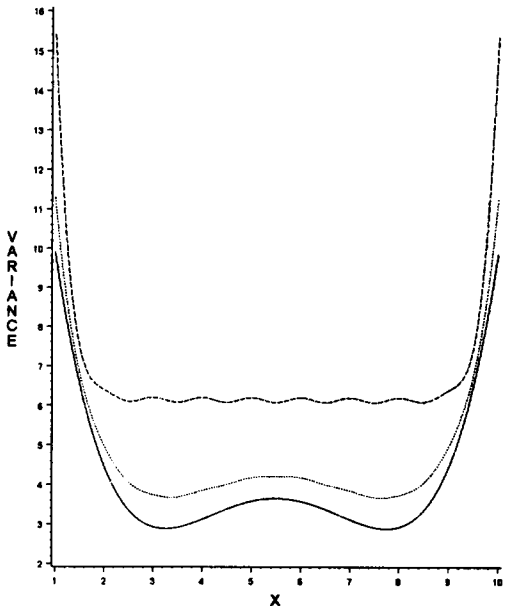
This example illustrates the use of the fitting methods when the data has a sine wave structure, but a polynomial model is specified to be used. The underlying model is taken to be

$$y = 5\sin(2\pi X) + \varepsilon \qquad (6.C.2)$$

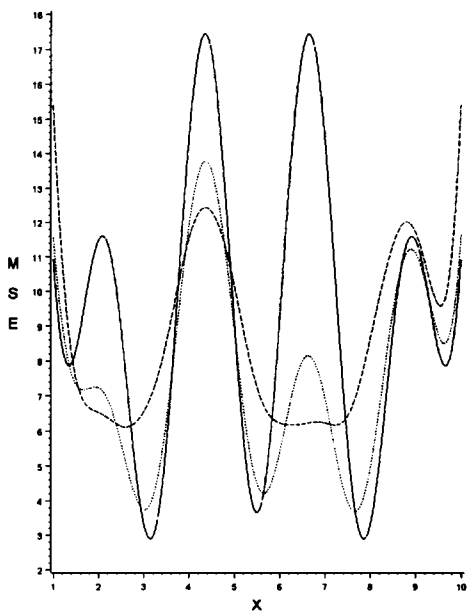
for X = 0 to 1 by .05, where $\varepsilon \sim N(0,1)$. This gives a basic one period sine wave with amplitude five. To provide a clearer interpretation of how well the final curves actually fit this sine structure, the generated data used is actually from (6.C.2) without the error term (i.e., the "true" underlying data is used). For calculations, the variance σ^2 is taken to be one. The natural polynomial model specified for this type of data (not knowing it was really from a sine function), would be the cubic model, $y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$. This being the case, Figure 6.C.12 displays the data along with the OLS, LLR, and MRR1 fits. The optimal bandwidth was found to be h_o = .086 and the optimal λ to be λ_o = .479.



(a) Squared Bias



(b) Variance



(c) MSE

Figure 6.C.11 (a)-(c). Squared Bias, Variance, and MSE plots for OLS, LLR, and MRR1, for Example 1.

[— OLS ---- LLR ····· MRR1]

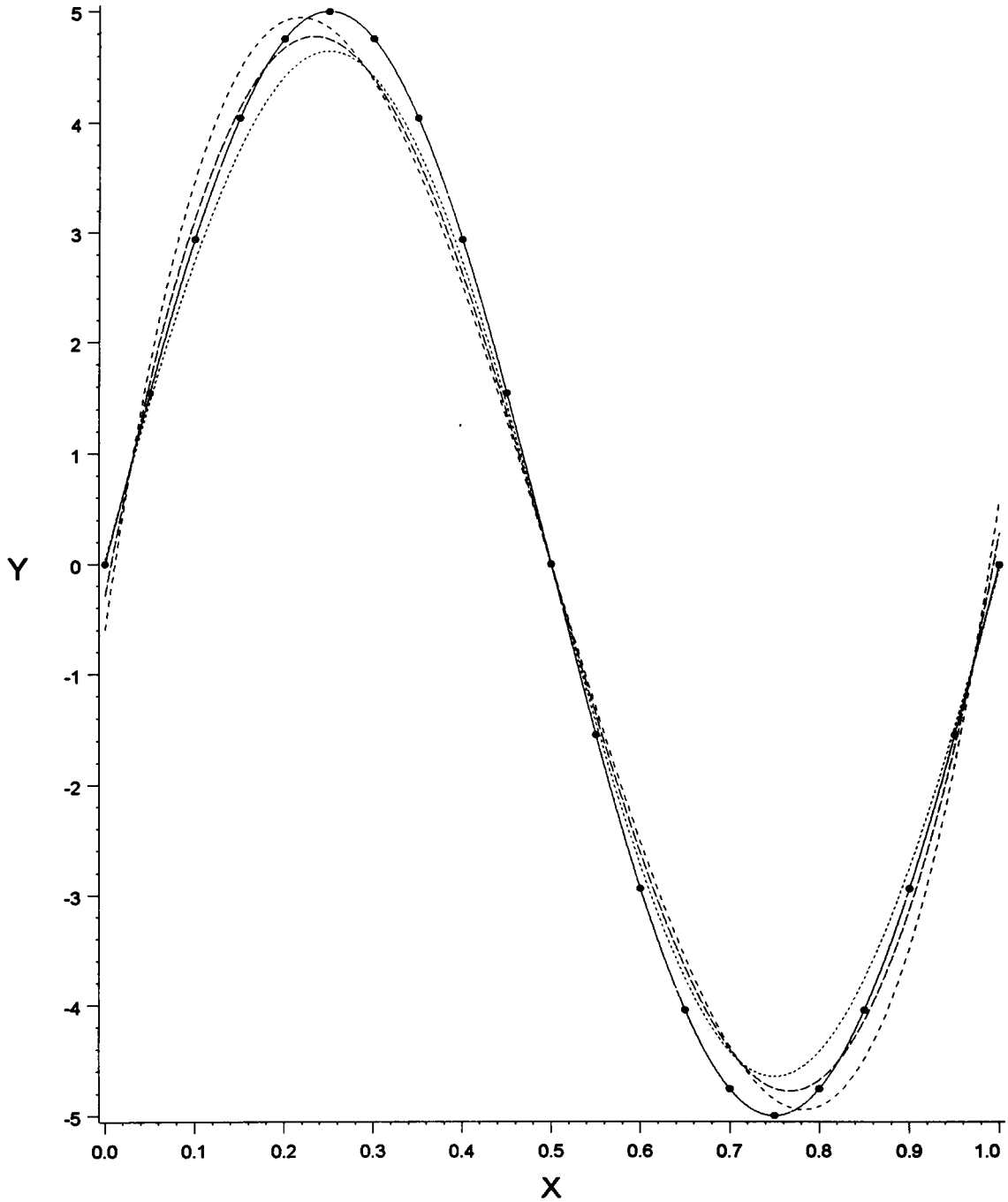


Figure 6.C.12. Plot of generated data for Example 2, with cubic OLS, LLR, and MRR1 regression curves.

[••• Raw data —— True curve - - - OLS ····· LLR ---- MRR1]

Notice that the cubic OLS curve takes the same general shape as the sine function, but cannot capture the exact form of the sine. This is the model misspecification for this example (a sine is not a cubic function). Local linear regression also has some problems fitting this data, namely where the sine curve peaks and dips. This problem of fitting in areas of sharp curvature was briefly mentioned in section 3.B.4 when discussing the need for variable bandwidth selectors. Another approach here would be to use local *quadratic* regression in an attempt to better fit the sharp curvature. This has been done, but the fits are only a little improved and there is hardly any change in the performance diagnostics. Thus, the use of LLR is maintained here for consistency with other examples. The MRR1 fit provides some improvement, but still cannot capture the "sharp" peak and dip.

Figure 6.C.13 gives the fits of the three model-robust methods. PLR does no better than MRR1, most noticeable in the high curvature areas. This stems from the parametric portion of PLR fitting poorly at the peak and dip, leaving a residual structure with high curvature at these locations. The PLR nonparametric portion cannot capture this curvature, and so neither does the final PLR fit. MRR2, on the other hand, gives much improved fits at the areas of high curvature. The initial OLS fit removes much of the structure at these points, leaving residuals that are much easier to fit than those in PLR. From inspection of Figure 6.C.12, notice that the residual $(y - \hat{y})$ from the OLS fit at the right boundary is rather large and negative, whereas the four residuals preceding this point are all positive. Kernel regression cannot adequately fit the negative residual at the boundary due to the weights given to the proceeding positive residuals. A similar phenomenon occurs at the left boundary, with the signs on the residuals reversed. Local linear regression overcomes this problem. However, a couple of the large residuals from these points preceding the endpoints are not fit extremely well by LLR and result in higher biases for MRR2 (due to curvature in the residual structure).. The differences in these biases, though, are not as significant as the differences at the curvature areas (where MRR2 is best).

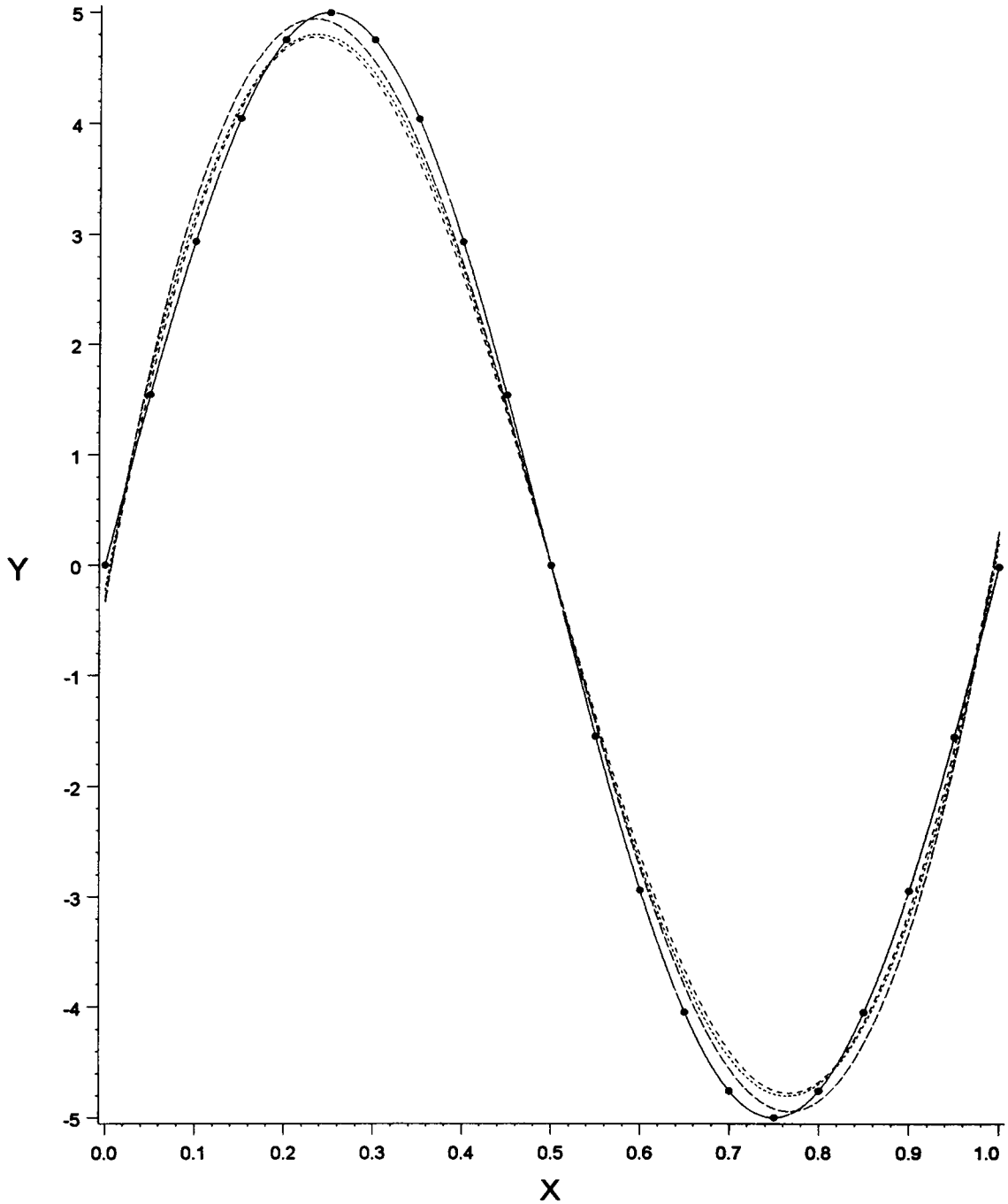


Figure 6.C.13. Plot of generated data for Example 2, with MRR1, MRR2, and PLR regression curves (based on cubic parametric models).

[• • • Raw data —— True curve - - - MRR1 ---- MRR2 ····· PLR]

Performance diagnostics are provided in Table 6.C.2. Again, the three model-robust methods outperform the individual OLS and LLR methods. Based on INTMSE, MRR1 and MRR2 perform a little better than PLR. Among the model-robust procedures, df_{model} is lowest for MRR2 (= 5.70), and is not much greater than the $p = 4$ for OLS.

6.C.4 Example 3

For this final example based on generated data, model (6.C.1) from Example 1 is used once again, but now *two* observations are taken at each of ten evenly spaced X-values from 1 to 10. Taking two observations at each point adds some distortion to the data in terms of giving a wider spread of data points about the true curve. At the same time, however, these replicated data points provide more information about the underlying model at each point than would be provided by just one observation. This extra "local" information should result in better performances for the nonparametric portions of the various fitting techniques. These improvements should in turn lead to better final fits for all of these procedures. The question is whether one procedure benefits more than the others.

Figure 6.C.14 shows the true curve and generated data, along with the quadratic OLS, the LLR, and the MRR1 fits. The extra distortion in the data results in a LLR fit that is a little more "structured" (not as smooth as in Example 1) (with $h_o = .099$), and hence results in a MRR1 fit (with $\lambda_o = .686$) that follows more closely the pattern of the true curve than did the MRR1 fit of Example 1 (given in Figure 6.C.4). This extra structure in the MRR1 curve allows it to compare a little more favorably to MRR2 and PLR, which is apparent in Figure 6.C.15 (as compared to Figure 6.C.7 of Example 1). The MRR2 and PLR curves are very similar to each other, and are still slightly better fits than the MRR1 curve, most notably in the dip area. Again, the model-robust procedures are an improvement over the individual parametric and nonparametric procedures.

Numerical results supporting this contention are given in Table 6.C.3. The model-robust procedures perform similarly, with PLR and MRR2 having the lower INTMSE's.

Table 6.C.2. Bandwidth, mixing parameter, and performance diagnostics for Example 2. Bandwidth and mixing parameter minimize AVEMSE. Key values for comparisons are underlined.

	h_o	λ_o	df_{model}	SSE	PRESS	PRESS*	INTMSE
OLS	---	---	4	3.13	6.80	.400	.298
LLR	.086	---	8.05	1.24	2.87	.222	.322
MRR1	.086	.479	5.94	1.24	3.64	.242	**.244**
MRR2	.140	.961	**5.70**	**1.12**	3.95	.258	**.245**
PLR	.141	---	5.87	**0.97**	**2.94**	**.195**	**.250**

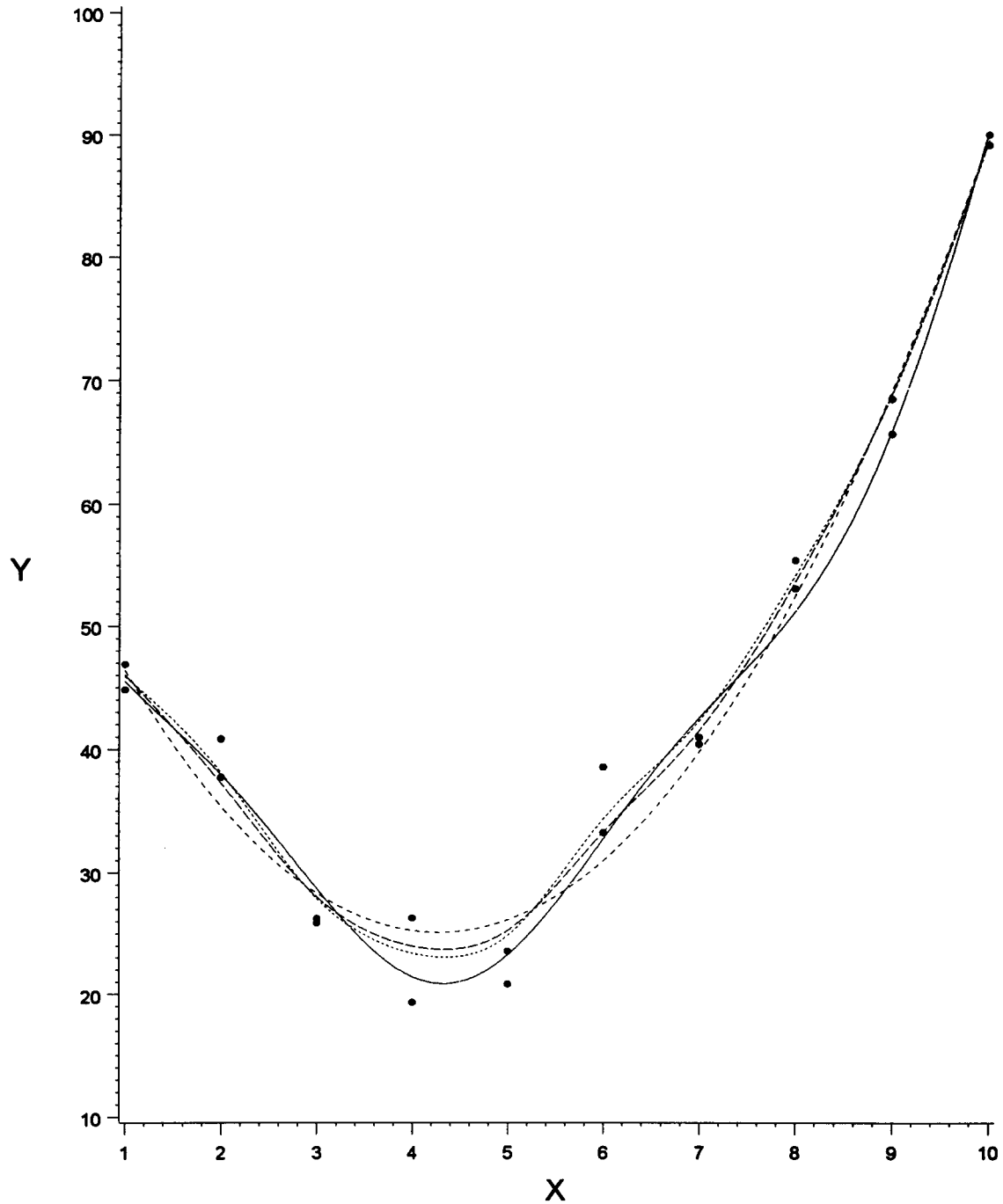


Figure 6.C.14. Plot of generated data for Example 3, with quadratic OLS, LLR, and MRR1 regression curves.

[••• Raw data —— True curve - - - OLS ····· LLR ---- MRR1]

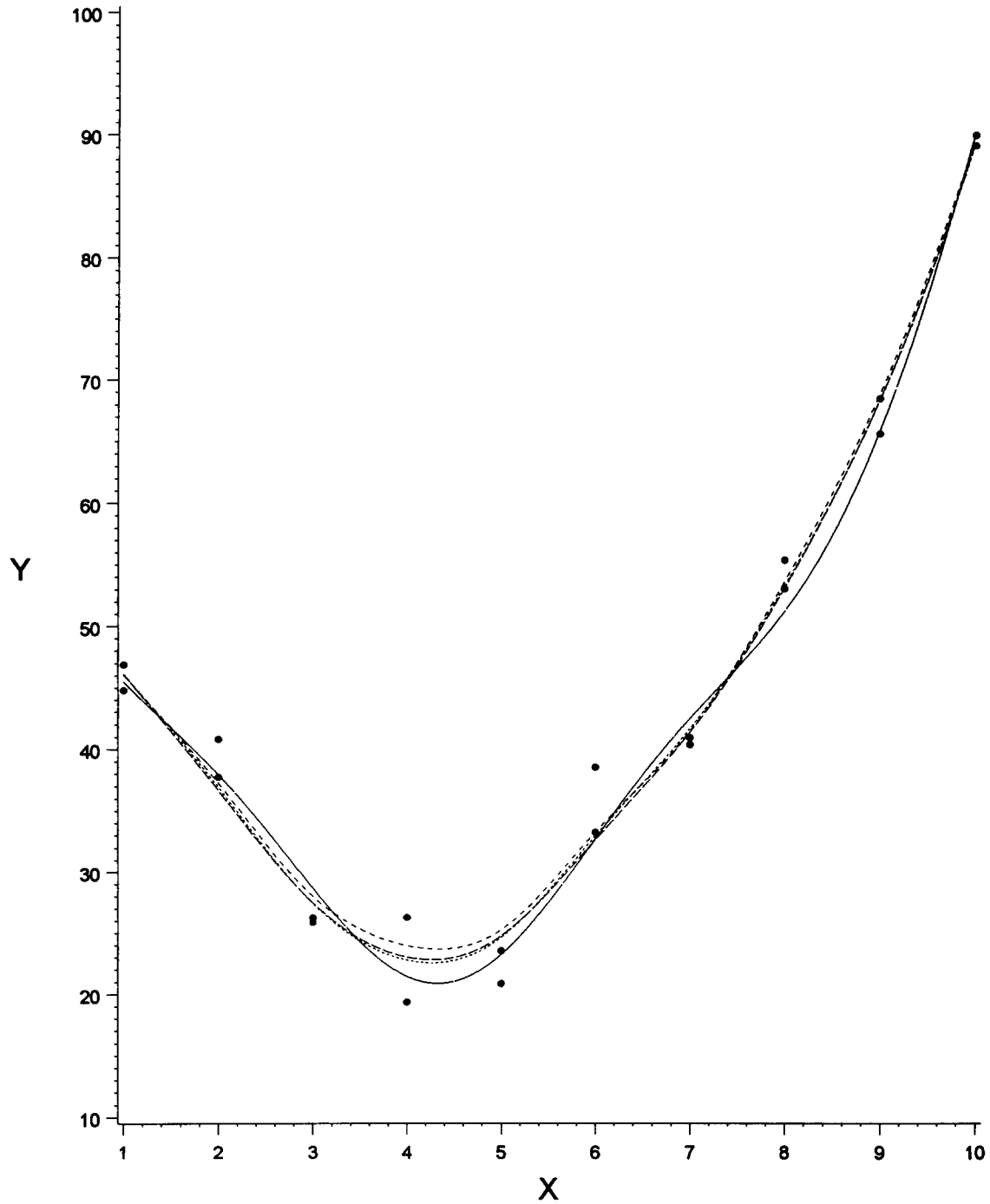


Figure 6.C.15. Plot of generated data for Example 3, with MRR1, MRR2, and PLR regression curves (based on quadratic parametric models).

[••• Raw data —— True curve - - - MRR1 ---- MRR2 ····· PLR]

Table 6.C.3. Bandwidth, mixing parameter, and performance diagnostics for Example 3. Bandwidth and mixing parameter minimize AVEMSE. Key values for comparisons are underlined.

	h_o	λ_o	df_{model}	SSE	PRESS	PRESS*	INTMSE
OLS	---	---	3	205.44	264.62	15.57	7.37
LLR	.099	---	7.08	96.41	211.99	16.41	4.96
MRR1	.099	.686	**5.80**	116.14	212.08	14.93	**4.69**
MRR2	.119	.879	**5.83**	115.83	209.83	**14.81**	**4.42**
PLR	.118	---	6.27	**107.93**	**207.10**	15.08	**4.40**

Note the significant decrease in INTMSE from Example 1 (Table 6.C.1) to this example. This is due to the increased local information from the replicates, which leads to significantly decreased biases and variances. The squared bias, variance, and MSE plots for this example are virtually identical in structure to those in Figures 6.C.10 (a)-(c), just with lower values. As a whole, it appears that all fitting procedures were affected similarly by the introduction of replicated observations.

6.C.5 Application

Consider the data in Table 6.C.4, where the response y is the tensile strength (in psi) of paper, and the regressor X is the percentage of hardwood in the batch of pulp from which the paper was produced. This data is taken from Montgomery and Peck (1992), and was studied by Einsporn and Birch (1993). Montgomery and Peck argue that many users would feel it appropriate to use a quadratic model to fit this data. Their argument is based on residual plots after actually fitting a quadratic model by OLS. This tensile data is plotted in Figure 6.C.16, along with the quadratic OLS, LLR, and MRR1 fits. Here the bandwidth (h = .127) and the mixing parameter (λ = .894) are chosen by the data driven method based on PRESS* (as described in section 3.B.3). The question of how effective PRESS* is at choosing the appropriate h and λ is addressed in the next chapter. It is shown that PRESS* may often choose h and/or λ that are "far" from optimal. However, for this application, the fits for the procedures relying on h and λ chosen by PRESS* *do* appear to be adequate (they are smooth and capture most of the structure in the data), so PRESS* is used. Note that OLS does not fit well, especially at the peak in the data and near the right boundary. Local linear regression fits much better. MRR1, based on approximately 90% LLR, gives a much improved fit over OLS, but is not much different from LLR.

MRR2 and PLR provide slightly better fits in terms of capturing the peak in the data. These differences are seen in Figure 6.C.17, which displays the three model-robust regression curves. MRR2 (with h = .176 and λ = .939) and PLR (with h = .186) give nice

Table 6.C.4 Tensile Strength Data. Y = tensile strength (psi) and X = percentage of hardwood.

X	Y	X	Y
1	6.3	7	42.0
1.5	11.1	8	46.1
2	20.0	9	53.1
3	24.0	10	52.0
4	26.1	11	52.5
4.5	30.0	12	48.0
5	33.8	13	42.8
5.5	34.0	14	27.8
6	38.1	15	21.9
6.5	39.9		

Table 6.C.5. Bandwidth, mixing parameter, and performance diagnostics for Tensile Data. Bandwidth and mixing parameter minimize PRESS*. Key values for comparisons are underlined.

	h	λ	df_{model}	SSE	PRESS	PRESS*
OLS	---	---	3	<u>312.64</u>	478.88	29.93
LLR	.127	---	5.92	72.75	184.04	14.07
MRR1	.127	.894	5.61	80.70	163.26	12.19
MRR2	.176	.939	<u>4.64</u>	87.41	<u>156.14</u>	<u>10.88</u>
PLR	.186	---	<u>4.63</u>	86.14	163.60	11.38

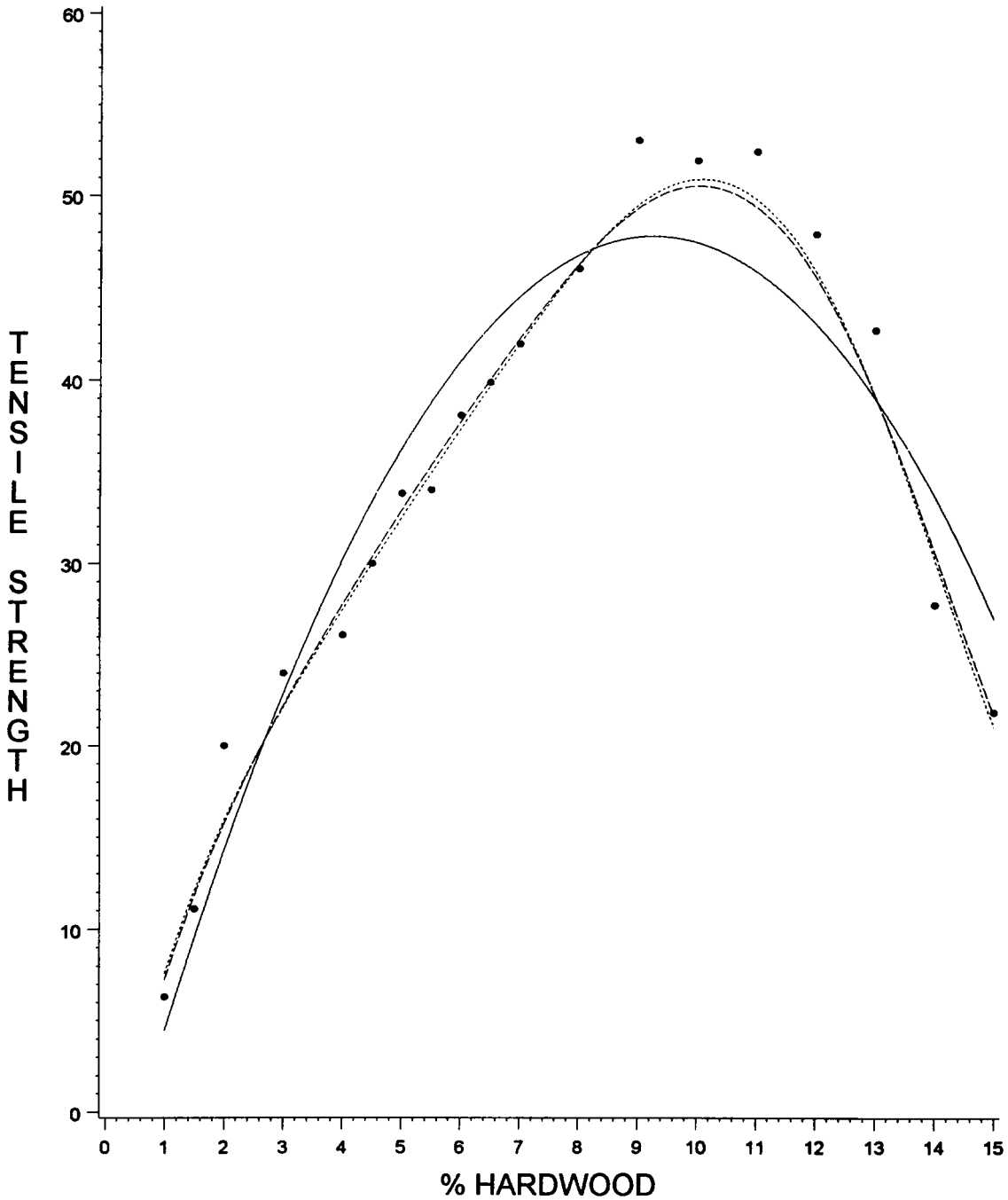


Figure 6.C.16. Plot of tensile data, with quadratic OLS, LLR, and MRR1 regression curves.

[• • • Raw data —— OLS ····· LLR ---- MRR1]

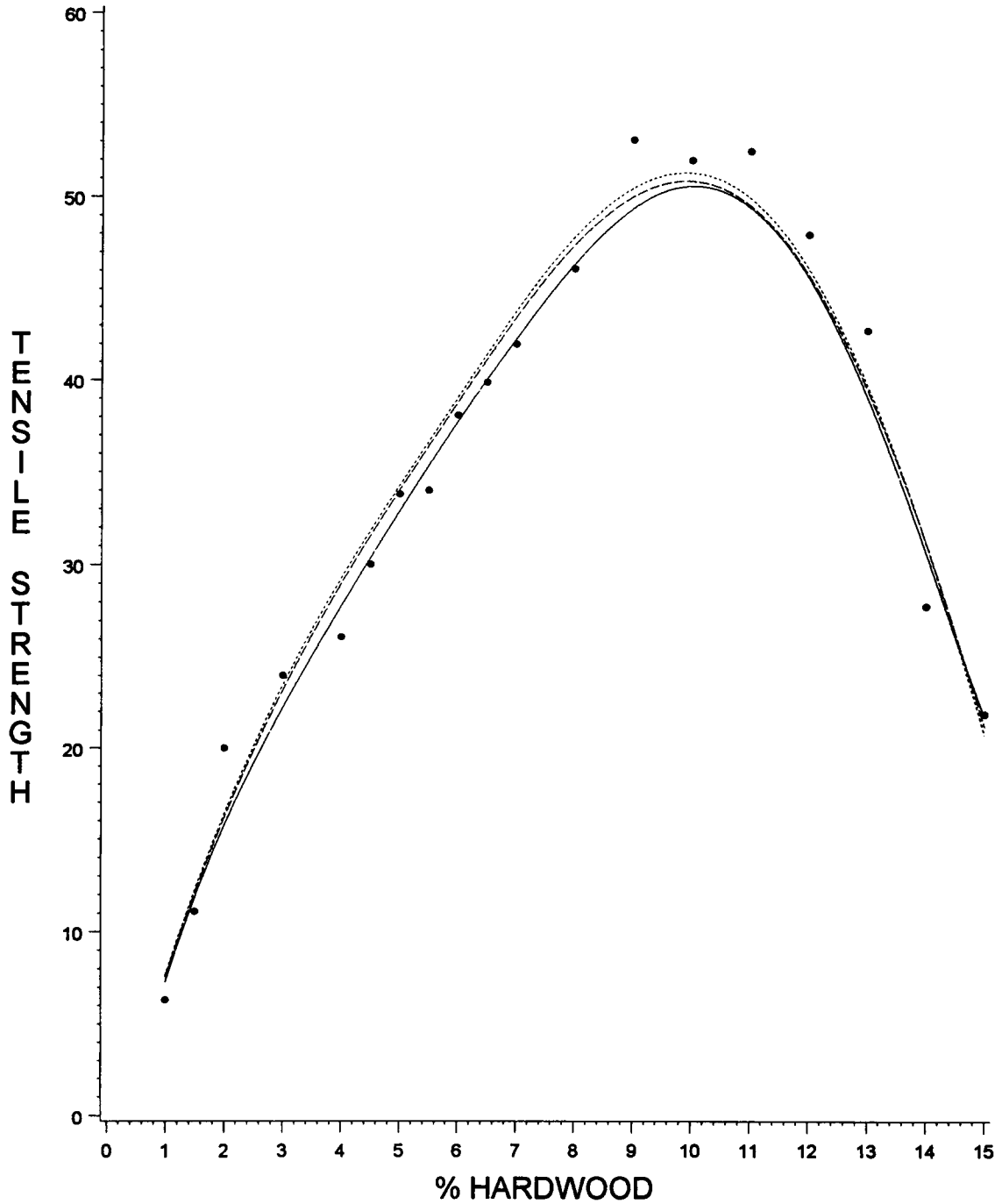


Figure 6.C.17. Plot of tensile data, with MRR1, MRR2, and PLR regression curves (based on quadratic parametric models).

[• • • Raw data —— MRR1 ---- MRR2 ····· PLR]

smooth curves that fit the data extremely well. The diagnostics of Table 6.C.5 support the observations made thus far. Note the slight improvement of MRR1 over LLR (lower df_{model}, PRESS, and PRESS*), and the tremendous improvement over OLS. Also, considering all of the diagnostics, MRR2 performs a little better than the other procedures. This is seen mainly in the PRESS and PRESS* diagnostics. A nice property of MRR2 and PLR is low df_{model} values compared to LLR, which is evident in the smoothness of their fits. Again, the model-robust procedures provide noticeable improvements over the individual parametric or nonparametric fits.

Before proceeding to the study of data-driven selectors of h and λ and the presentation of simulation results, two other important topics need to be addressed. These are (1) the development of confidence intervals for each of the procedures, and (2) a brief look at the performances of the procedures when the sample size is decreased in the previous examples.

6.D Confidence Intervals

Now that the fits for all of the competing procedures have been obtained, it is desireable to have a measure of the accuracy and precision of these fits. This is accomplished via confidence intervals on the fits. Ideally, one would like to have confidence intervals as narrow as possible and still maintain the desired coverage probabilities (90%, 95%, 99%, . . .). Inherent in the construction of these confidence intervals is the need for estimates of the error variance ($\hat{\sigma}^2$). It is desired to obtain $\hat{\sigma}^2$'s and confidence intervals (C.I.'s) for each of the fitting procedures that are as basic as possible in form. The C.I.'s developed here satisfy this notion and closely parallel the form of OLS C.I.'s. Also, the C.I.'s are developed for any location x_o in the range of the data.

The OLS $100(1-\alpha)\%$ C.I. for the true mean μ_{y_o} at the location x_o is given by

$$\hat{y}_o^{(ols)} \pm t_{n-p,\frac{\alpha}{2}} \hat{\sigma}\sqrt{\mathbf{x}_o'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_o} \ ,$$

where $\hat{y}_o^{(ols)}$ is the fitted value at the individual point $\mathbf{x}_o' = (1 \quad x_o \quad x_o^2 \cdots)$, $t_{n-p,\frac{\alpha}{2}}$ is the $\left(\frac{1-\alpha}{2}\right)^{th}$ percentile of the t-distribution with $n-p$ degrees of freedom, and $\hat{\sigma}$ is an estimate of error standard deviation (Myers (1990)). The usual estimate of σ is

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{(ols)}\right)^2}{n-p}} \quad ,$$

as given in equation (2.7). The general form of this C.I. may be expressed as

$$\hat{y}_o^{(ols)} \pm t_{n-tr(\mathbf{H}^{(ols)}),\frac{\alpha}{2}}\, \hat{\sigma}\sqrt{(\mathbf{h}_o'^{(ols)}\, \mathbf{h}_o^{(ols)})} \quad , \qquad (6.D.1)$$

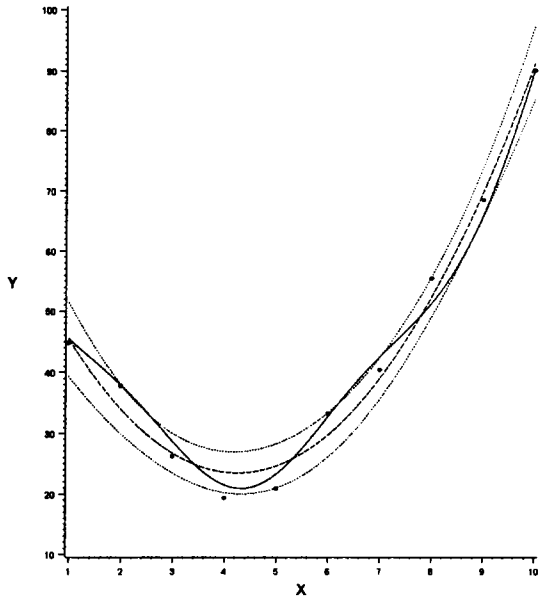
where $\mathbf{h}_o'^{(ols)} = \mathbf{x}_o'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_o$ is the row vector of OLS weights determined by $\mathbf{x}_o$. This general form is maintained for each of the confidence intervals developed in this current work. Thus, for each fitting technique, the 100(1−α)% confidence interval for μ_{y_o} is expressed as

$$\hat{y}_o^{(\bullet)} \pm t_{n-tr(\mathbf{H}^{(\bullet)}),\frac{\alpha}{2}}\, \hat{\sigma}_{(\bullet)}\sqrt{(\mathbf{h}_o'^{(\bullet)}\, \mathbf{h}_o^{(\bullet)})} \quad , \qquad (6.D.2)$$

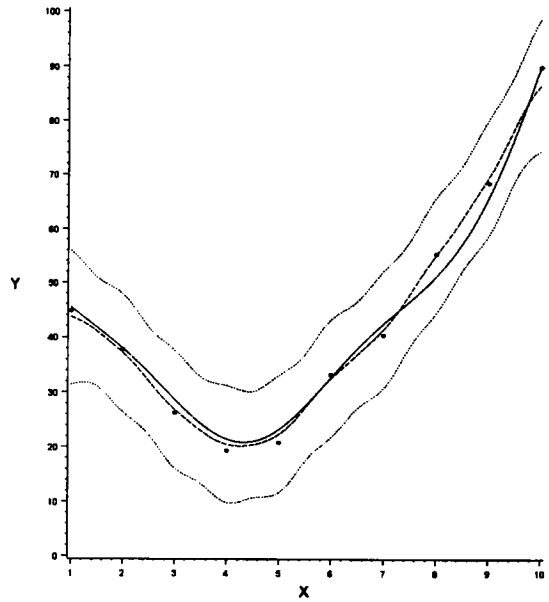
where $\hat{\sigma}_{(\bullet)} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{(\bullet)}\right)^2}{n - \mathrm{tr}(\mathbf{H}^{(\bullet)})}}$, and "•" can be replaced by any of the fitting techniques (OLS, Ker, LPR, MRR1, MRR2, or PLR). The expressions for $\mathbf{h}_o'$ for each procedure are the same as those given in section 6.B when the MSE formulas were derived. The appropriateness of the general C.I. form in (6.D.2) for the nonparametric and model-robust procedures is supported by Silverman (1985) and Hastie and Tibshirani (1987), who use confidence intervals of this form when discussing spline regression and general additive models, respectively.

Figures 6.D.1 (a)-(f) present the various fits along with their 95% confidence bands for the data in Example 1. Notice that the confidence bands for OLS in figure (a) are rather narrow, but fail to capture the true curve in several areas. The kernel confidence bands (figure (b)) are very wide and irregular due to larger variances of the fits. These bands are much improved for LLR, as seen in figure (c). The MRR1 confidence bands of figure (d) maintain the narrow width of OLS bands and the better coverage of LLR bands. The bands for the model-robust techniques MRR2 and PLR appear to be smooth, narrow, and provide adequate coverage of the true curve (see figures (e) and (f)). Table 6.D.1 provides confidence interval diagnostics for comparing the various fitting procedures. Specifically, this table contains the average confidence interval width across the actual data points for the competing procedures for Examples 1, 2, and 3. (It is shown later using simulation studies that, especially for smaller sample sizes, confidence intervals at locations between data points may become very wide (they appear to be "unstable"); thus, just the C.I.'s at the data points are averaged here). The model-robust procedures appear to be performing the best on the whole, with consistently narrow confidence intervals (always narrower than LLR, and often narrower than OLS). Actually, one does not even get the whole story with just these width values. Also of interest is the coverage probability of the various C.I.'s. Even though one C.I. is narrower than another, if it does not provide adequate coverage, then it is no good. Simulations are needed to study these coverage probabilities, and such results are provided in Chapter 8. It turns out that the model-robust procedures provide adequate coverage probabilities, while OLS coverage probabilities are often much too low. This is especially true when there is larger misspecification in the model. So once again the model-robust procedures prove beneficial over individual parametric and nonparametric procedures.

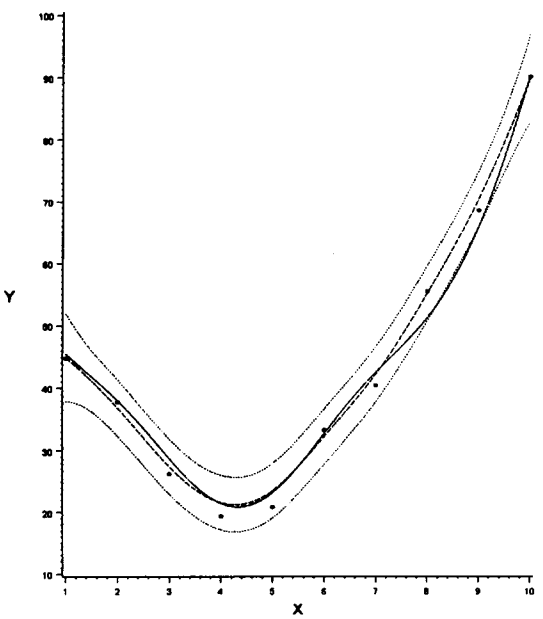
Of course, many other forms of confidence intervals have been studied in the literature. Härdle (1990) presents discussions of pointwise confidence bands derived through the establishment of the distribution of the nonparametric fits at the individual points, and variability bands for functions, derived through distribution and derivative



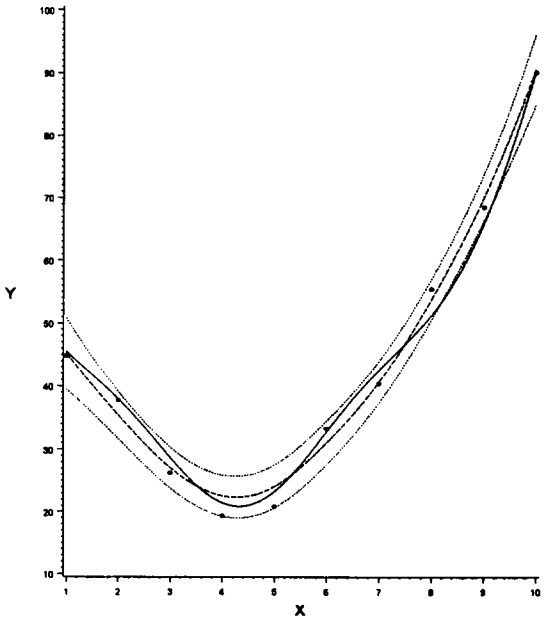
(a) OLS



(b) Ker



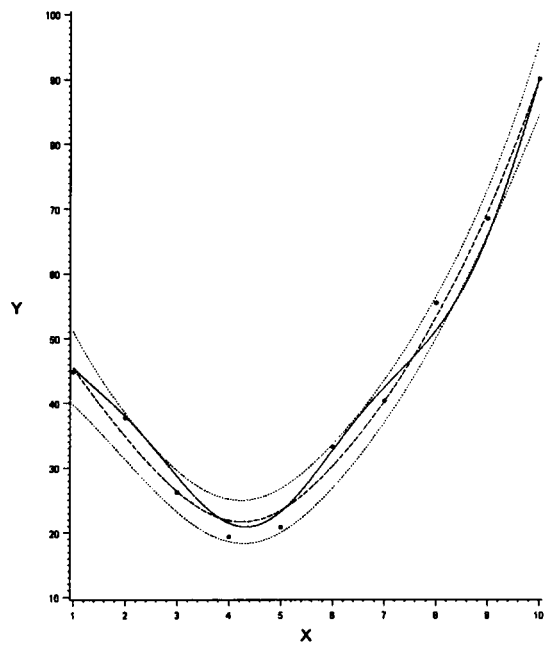
(c) LLR



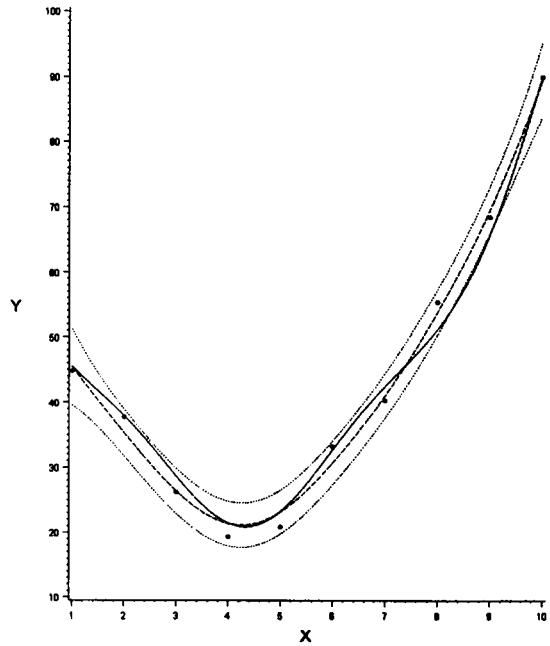
(d) MRR1

Figure 6.D.1 (a)-(d). Plot of confidence bands for OLS, Ker, LLR, and MRR1 for Example 1.

[••• Raw data —— True curve ----- Fitted curve ····· Conf. band]



(e) MRR2



(f) PLR

Figure 6.D.1 (e)-(f). Plot of confidence bands for MRR2 and PLR for Example 1.

[••• Raw data —— True curve ----- Fitted curve ····· Conf. band]

Table 6.D.1. Average Confidence Interval Widths for Examples 1, 2, and 3.
Average C.I. widths across the data points for OLS, LLR, MRR1, MRR2, and PLR.

	AVE C.I. Width		
	Example 1	Example 2	Example 3
OLS	8.19	.763	5.51
LLR	9.94	.705	6.24
MRR1	7.77	.546	5.56
MRR2	7.64	.528	5.71
PLR	7.89	.506	5.94

results for the regression fits. A popular method of constructing confidence intervals that eliminates the need for complex derivational results (which are needed for the techniques described above) is the method of bootstrapping. Härdle and Bowman (1988) and Faraway (1990) discuss how to use this resampling technique to get empirical distributions of fits on which to base the construction of C.I.'s. The current work maintains the simplicity of the general C.I. form in (6.D.2), and other techniques, such as those described above, are left for future considerations.

6.E Smaller Sample Results

The final topic of concern for these initial comparisons is the effect of very small sample sizes on the performance of the various fitting techniques. It is hoped that none of the model-robust techniques would significantly falter in this situation. Since there is less information (data) explaining the true model, the particular techniques are expected to suffer somewhat in the adequacy of their fits. However, what must be checked is whether one (or more) of the procedures is more significantly affected than the others, which would seriously damage the usefulness of that particular procedure. Three examples are used here to study the effect of smaller sample sizes. These examples are simply Examples 1, 2, and 3 discussed previously, with fewer data points generated from the underlying model, and are described below. (Here h and λ are chosen to minimize AVEMSE).

Example 1′

This example obtains data from equation (6.C.1) of Example 1, but only at six (instead of ten) evenly spaced X-values from 1 to 10. This raw data is shown in Figures 6.E.1 (a)-(b), along with the regression curves from the various fitting procedures. Notice that the fits are not as good as they were with more data points (Figures 6.C.3, 6.C.7), but the model-robust fits are still noticeably better than the individual fits. Some performance diagnostics supporting this contention are given in Table 6.E.1.

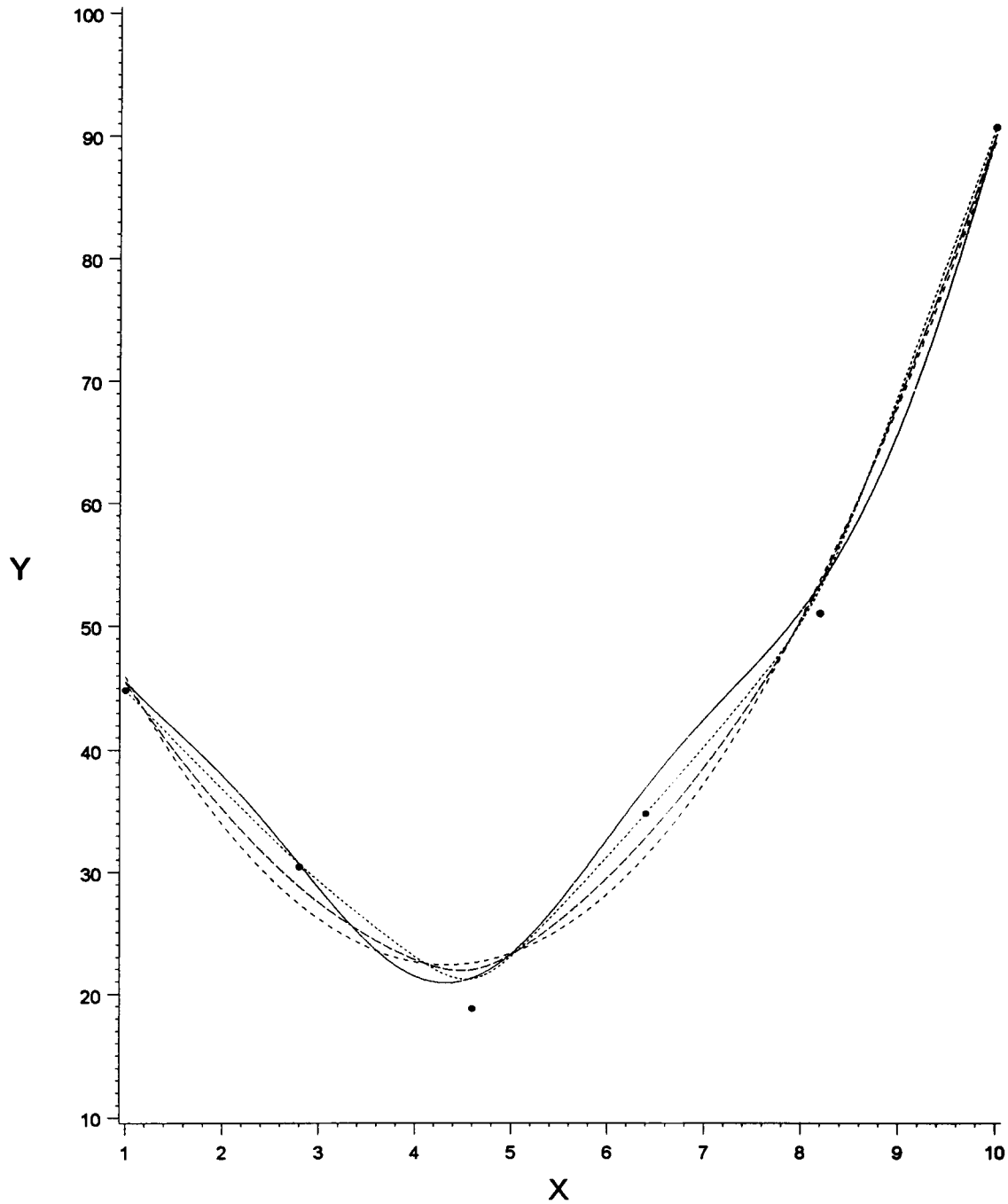


Figure 6.E.1 (a). Plot of generated data for Example 1′, with quadratic OLS, LLR, and MRR1 regression curves.

[••• Raw data —— True curve - - - OLS ····· LLR ---- MRR1]

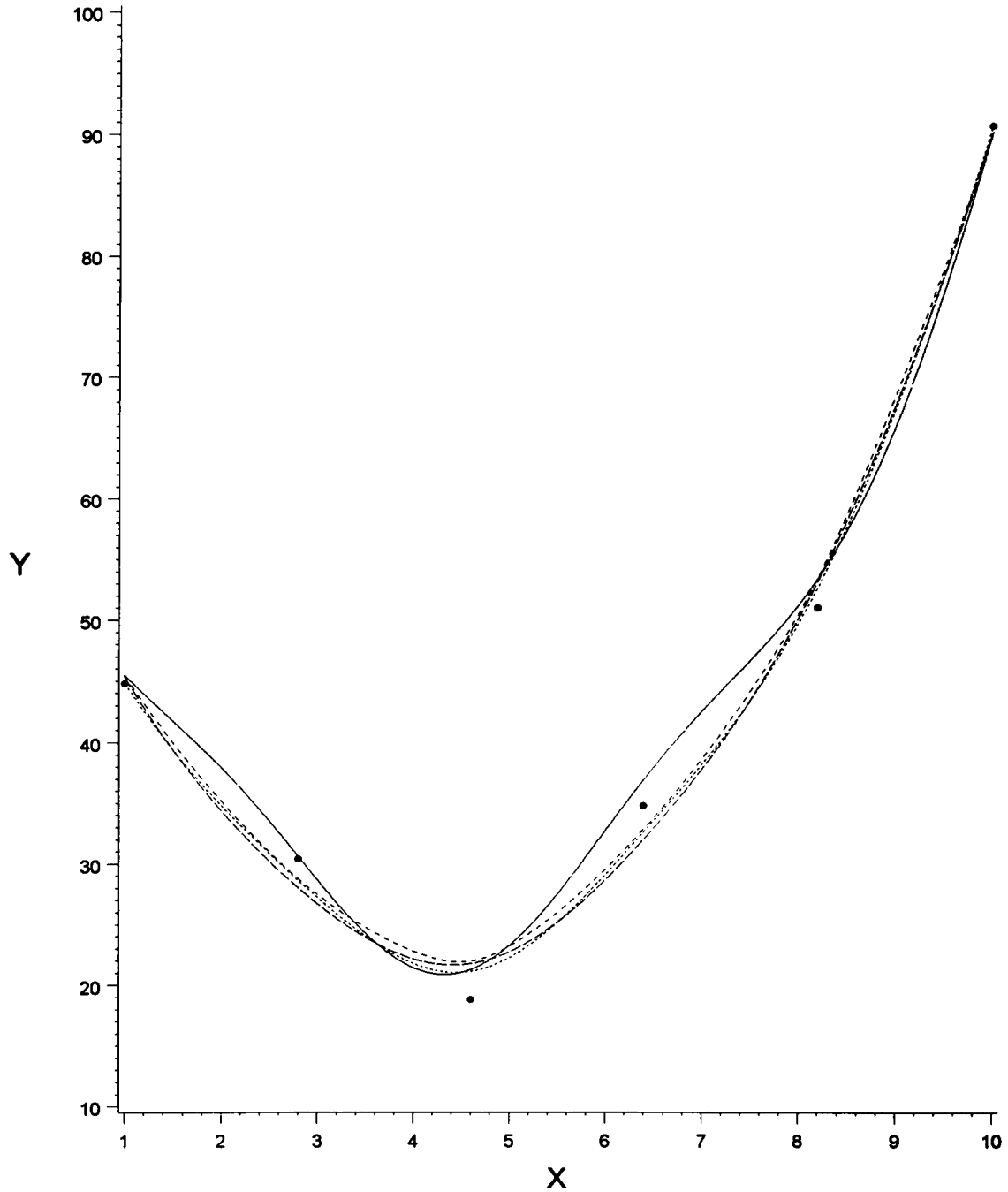


Figure 6.E.1 (b). Plot of generated data for Example 1′, with MRR1, MRR2, and PLR regression curves.

[••• Raw data —— True curve - - - MRR1 ---- MRR2 ····· PLR]

Table 6.E.1. Bandwidth, mixing parameter, and performance diagnostics for Example 1′. Bandwidth and mixing parameter minimize AVEMSE. Key values for comparisons are underlined.

	h_o	λ_o	df_{model}	SSE	PRESS	PRESS*	INTMSE
OLS	---	---	3	44.70	<u>166.18</u>	<u>55.39</u>	12.09
LLR	.133	---	5.30	<u>10.15</u>	895.32	1280.18	13.52
MRR1	.133	.433	<u>3.40</u>	23.00	<u>178.64</u>	<u>89.16</u>	<u>10.99</u>
MRR2	.169	.539	<u>3.93</u>	27.20	<u>189.61</u>	<u>91.49</u>	<u>10.97</u>
PLR	.169	---	4.75	<u>15.98</u>	540.91	431.93	<u>11.37</u>

Example 2′

Here data is obtained from equation (6.C.2) of Example 2 at six (instead of twenty-one) evenly spaced X-values from 0 to 1. As in Example 2, the "true" underlying data is used (i.e., data is from (6.C.2) without the error term). Figures 6.E.2 (a)-(b) show this data along with the various fits of interest. Notice that the MRR1 procedure uses a small amount of the LLR fit to adjust the OLS fit, whereas the MRR2 and PLR fits are very close to OLS. These considerations allow for the model-robust procedures to again perform well (although they are much closer to OLS), with MRR1 holding a slight advantage here. These conclusions are supported by the diagnostics of Table 6.E.2. It appears that as the amount of data decreases, the model-robust procedures place more emphasis on OLS (PLR does this by choosing $h \approx 1$). This characteristic is seen again in the simulation results of Chapter 8.

Example 3′

This final example obtains data from Equation (6.C.1) of Example 1, where two observations are taken at each of six (not ten, as in Example 3) equally spaced X-values from 1 to 10. Similar to Example 1′, the model-robust procedures perform noticeably better than the individual procedures. Fits are shown in Figures 6.E.3 (a)-(b), and diagnostics are given in Table 6.E.3.

The key observation from this section is that the model-robust procedures appear to hold up well when the sample size of the data is significantly decreased. More results in this regard, based on simulations, are given in Chapter 8.

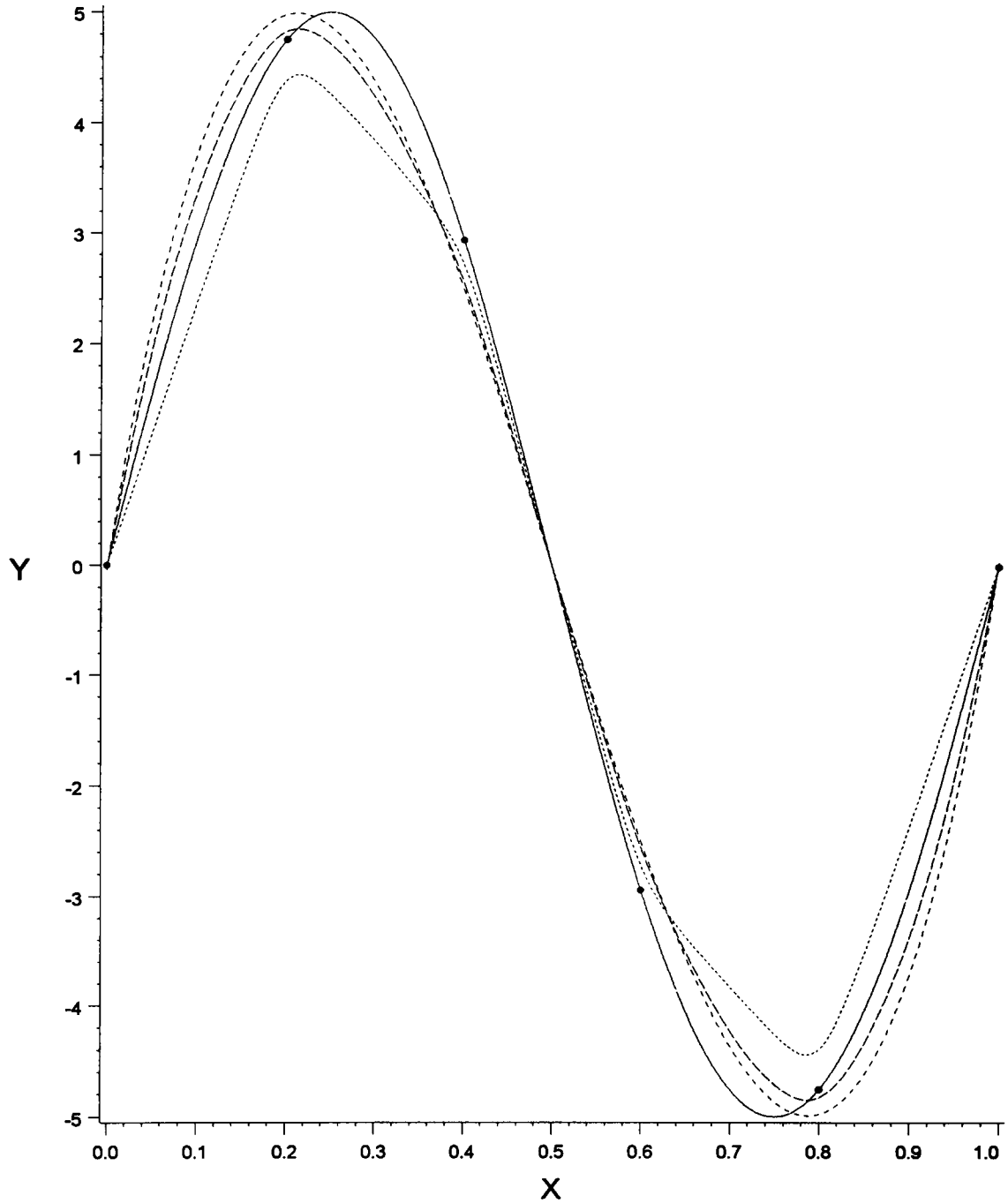


Figure 6.E.2 (a). Plot of generated data for Example 2′, with cubic OLS, LLR, and MRR1 regression curves.

[••• Raw data —— True curve - - - OLS ····· LLR ---- MRR1]

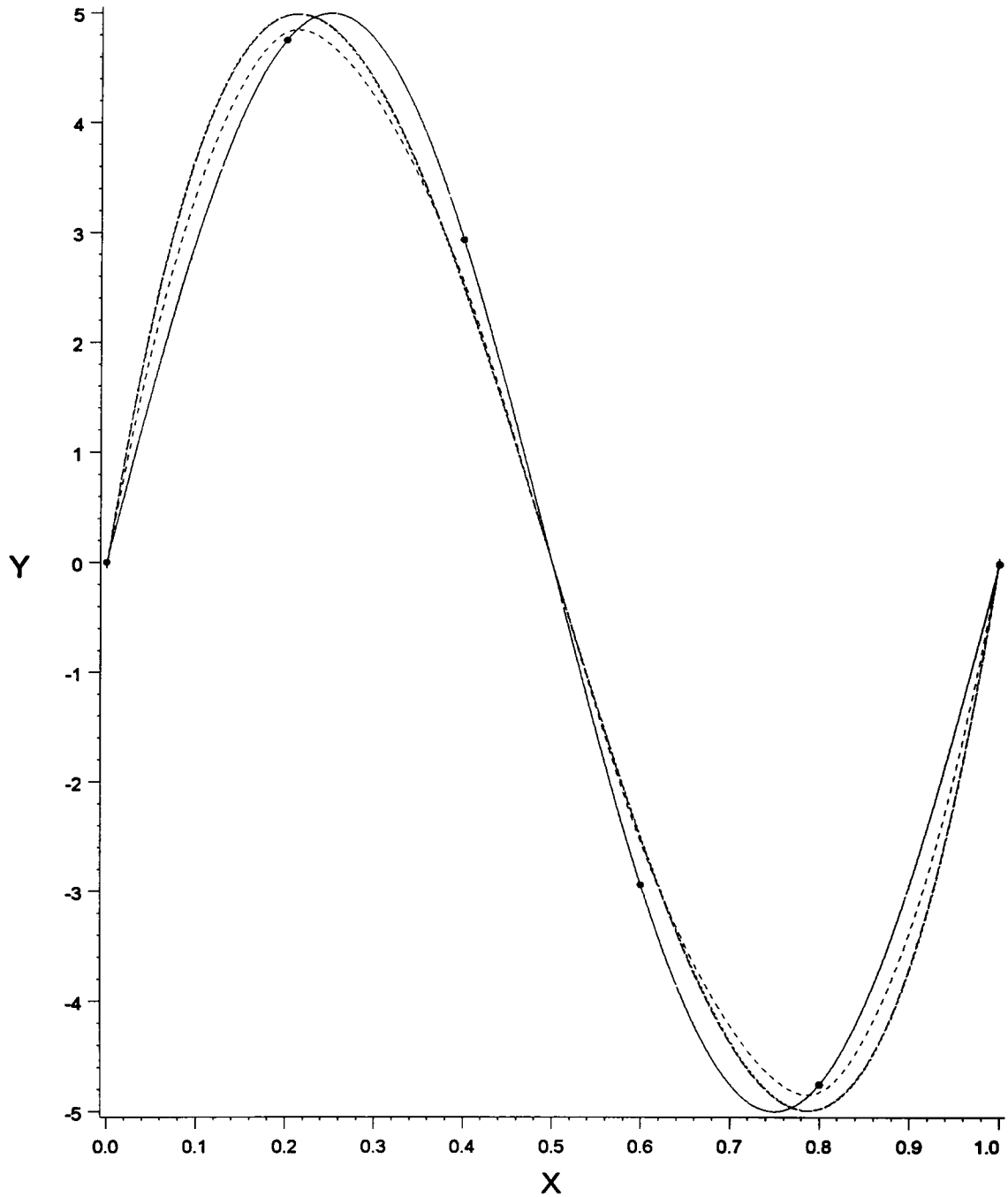


Figure 6.E.2 (b). Plot of generated data for Example 2′, with MRR1, MRR2, and PLR regression curves.

[••• Raw data —— True curve - - - MRR1 ---- MRR2 ····· PLR]

Table 6.E.2. Bandwidth, mixing parameter, and performance diagnostics for Example 2′. Bandwidth and mixing parameter minimize AVEMSE. Key values for comparisons are underlined.

	h_o	λ_o	df_{model}	SSE	PRESS	PRESS*	INTMSE
OLS	---	---	4	.500	4.44	2.22	.747
LLR	.120	---	5.55	.372	116.23	260.37	.916
MRR1	.120	.256	4.40	.316	4.24	2.65	.662
MRR2	.280	.578	4.15	.467	6.88	3.71	.738
PLR	1	---	4.00	.500	4.28	2.14	.747

Table 6.E.3. Bandwidth, mixing parameter, and performance diagnostics for Example 3′. Bandwidth and mixing parameter minimize AVEMSE. Key values for comparisons are underlined.

	h_o	λ_o	df_{model}	SSE	PRESS	PRESS*	INTMSE
OLS	---	---	3	120.20	191.53	21.28	8.90
LLR	.118	---	5.60	40.75	137.16	21.42	8.32
MRR1	.118	.617	4.60	55.93	135.27	18.28	7.23
MRR2	.141	.750	4.65	57.28	138.20	18.80	6.75
PLR	.140	---	5.23	45.67	132.89	19.62	6.79

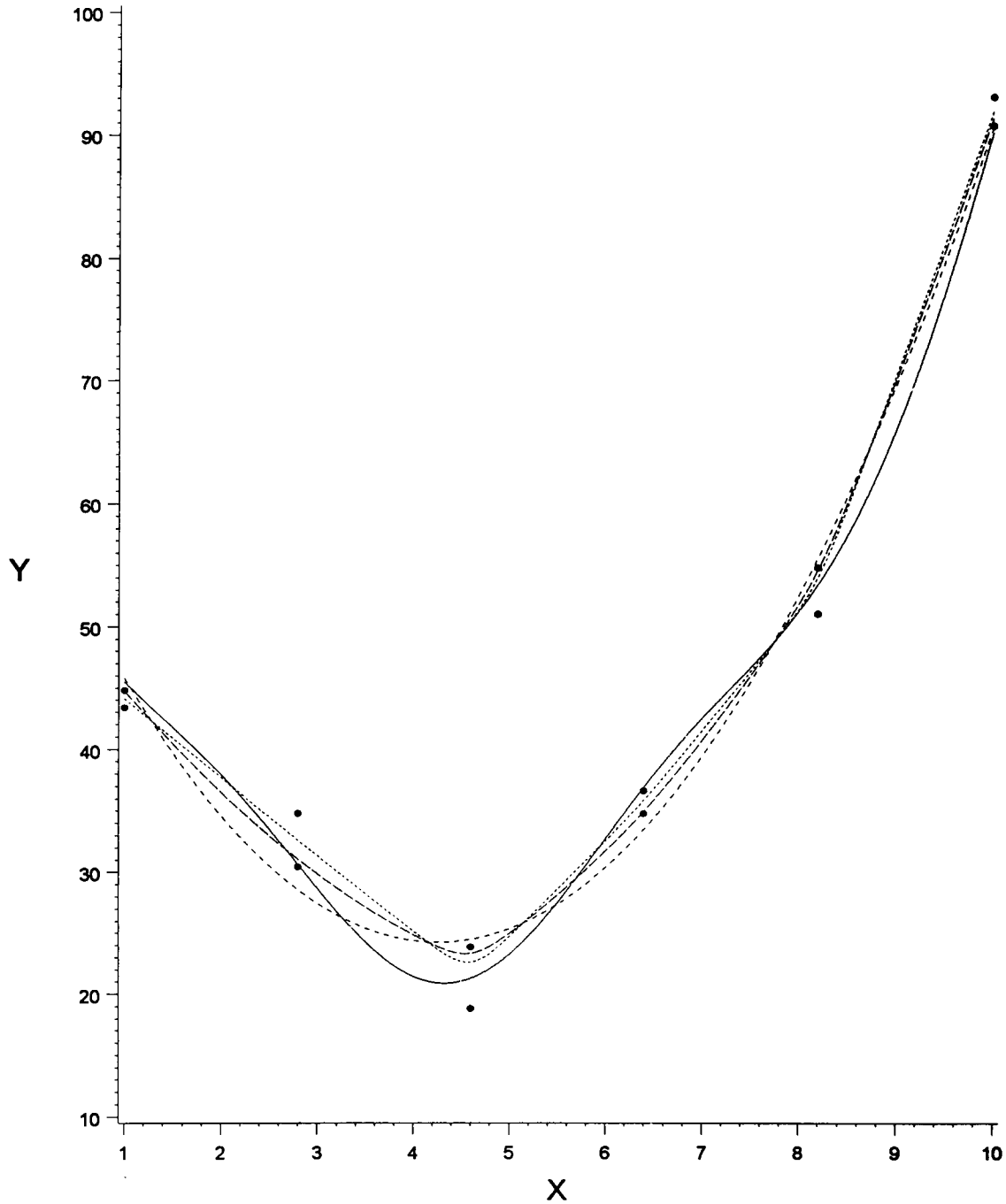


Figure 6.E.3 (a). Plot of generated data for Example 3′, with quadratic OLS, LLR, and MRR1 regression curves.

[••• Raw data —— True curve - - - OLS ····· LLR ---- MRR1]

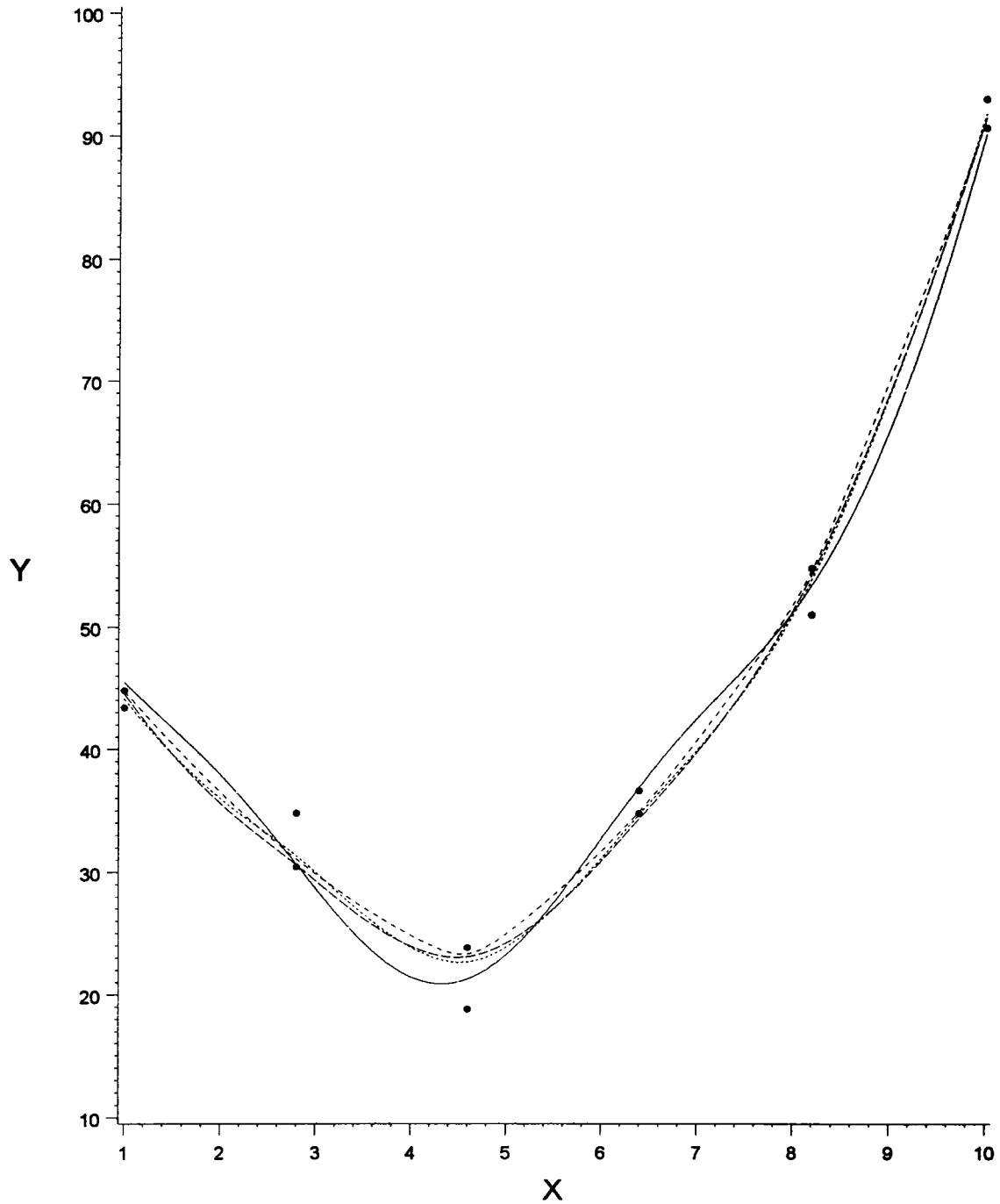


Figure 6.E.3 (b). Plot of generated data for Example 3′, with MRR1, MRR2, and PLR regression curves.

[••• Raw data —— True curve - - - MRR1 ---- MRR2 ····· PLR]

Chapter 7: Choice of Bandwidth & Mixing Parameter

7.A Optimal Criterion

Most of the results and comparisons given up to this point have been based on generated data sets for which the optimal bandwidths and mixing parameters can be determined. Since the true underlying model has been known, it has been possible to evaluate the "theoretical" MSE formulas of section 6.B in order to find h_o and λ_o (by minimizing AVEMSE). The use of h_o and λ_o gives the "best" fits possible for each of the fitting procedures developed throughout this paper. In other words, using AVEMSE to select h and/or λ for a particular procedure determines the procedure's "optimal" fit, where optimal refers to minimizing AVEMSE. Based on these optimal fits, one can then make true comparisons of the performances of the various fitting techniques. Since each technique is contributing its best fit, it is easy to make conclusions as to which techniques are outperforming the others. Except for the application (tensile data) in section 6.C.5, all of the results presented thus far have been based precisely on these considerations (with the main performance criterion being INTMSE). Thus, the conclusions that the model-robust procedures are very beneficial are based on solid arguments. (These conclusion have been based on just individual data sets, but simulation results in the next chapter will substantiate these findings).

AVEMSE is used as the "optimal" selection criterion for several reasons. First, it gives the desired measure of the tradeoff between the bias and variance of the fitted values at the data locations. Second, even though AVEMSE measures MSE values for only the actual data points, the differences in performance of the various fitting techniques are usually equally as evident in AVEMSE values as they are with the integrated MSE (INTMSE) values determined across the entire range of the data. The situation that may cause AVEMSE to give different results from INTMSE is extremely small sample sizes,

where there are wide gaps between data points (gaps where AVEMSE ignores the fitting structure). The examples considered in this chapter do not possess this problem, so AVEMSE should provide comparisons similar to those of INTMSE. This is an important point that is considered shortly when deciding on a measure of performance for comparing different data-driven selection criteria. A third reason for using AVEMSE as the optimal selection criterion (instead of a "global" measure like INTMSE) is to provide a fairer comparison with the data-driven methods. Data-driven methods have *only* the data to use in selecting h and λ, so it seems appropriate to have the "best" selection criterion to also only depend on the data. This gives a more valid basis for actually determining how well (or poorly) data-driven methods perform; i.e., they are not placed at a disadvantage to start with as they would be if using the global measure INTMSE as the optimal criterion. The next step then is to determine if there is some data-driven method that consistently chooses values of h and λ close to the optimal h_o and λ_o, thus allowing the benefits of the model-robust fitting procedures to be evident in practice. The remainder of this chapter provides a brief study of attempts at satisfying this need.

7.B Overview of Study

The following explanation gives the guidelines as to how this selection criterion study is carried out.

7.B.1 Data Sets

First of all, five different data sets are considered in evaluating methods, and these data sets are denoted Data1, Data2, . . ., Data5. Data1 is taken to be the data from Example 1 of section 6.C.2, which is generated from equation (6.C.1) and displayed in Figure 6.C.2 (a). Data2 is the data set generated again by equation (6.C.1), but without the error term. This data set is similar to the data shown in Figure 6.C.2 (a), except each point is located on the true underlying curve. Also, for calculations, σ^2 is taken to be 1 for this example. Data3 is the data from Example 3 of section 6.C.2, which is shown in

Figure 6.C.14. This data is just like that of Data1, but two observations (instead of one) are taken at each X-value. Data4 is the sine wave data of Example 2 in section 6.C.2. This data is generated from equation (6.C.2) and displayed in Figure 6.C.12. Finally, Data5 is generated from the underlying model

$$y = 2(10X - 5.5)^2 + 50X + 3.5\sin(4\pi X) + \varepsilon \qquad (7.B.1)$$

at twenty-one evenly spaced X-values from 0 to 1, where $\varepsilon \sim N(0,1)$. Actually, the data used is from (7.B.1) without the error term (i.e., the true data), and $\sigma^2=1$ is used when calculating diagnostics. This data is shown in Figure (7.B.1) along with the true curve. Note that the true curve for Data5 is very similar to that of Data1 and Data2; however, the data itself consists of twice as many observation in the same range (after transforming the data of Data1 and Data2 to be between 0 and 1). Each of the data sets Data1 through Data5 contains different characteristics and all together provide a nice range of underlying structures for fitting regression curves.

7.B.2 Performance Criterion

Of basic concern for this study is simply observing how close the h's and λ's chosen by certain data-driven methods are to the optimal h_o and λ_o. The closer the chosen values are to h_o and λ_o, the better the data-driven method is considered to be. However, since there will undoubtedly be differences between the optimal and chosen values, some type of measure is needed to determine how much the fits based on the chosen h's and λ's differ from the "optimal" fits based on h_o and λ_o. A natural diagnostic would be INTMSE, but based on considerations pointed out in the previous section, AVEMSE values are used instead for these comparisons. It was mentioned previously that for the examples used here, AVEMSE values provide comparisons across fitting techniques that are very similar to those provided by INTMSE values. Support for this statement is provided in Table 7.B.1, where the AVEMSE and INTMSE values for the optimal fits are given for each

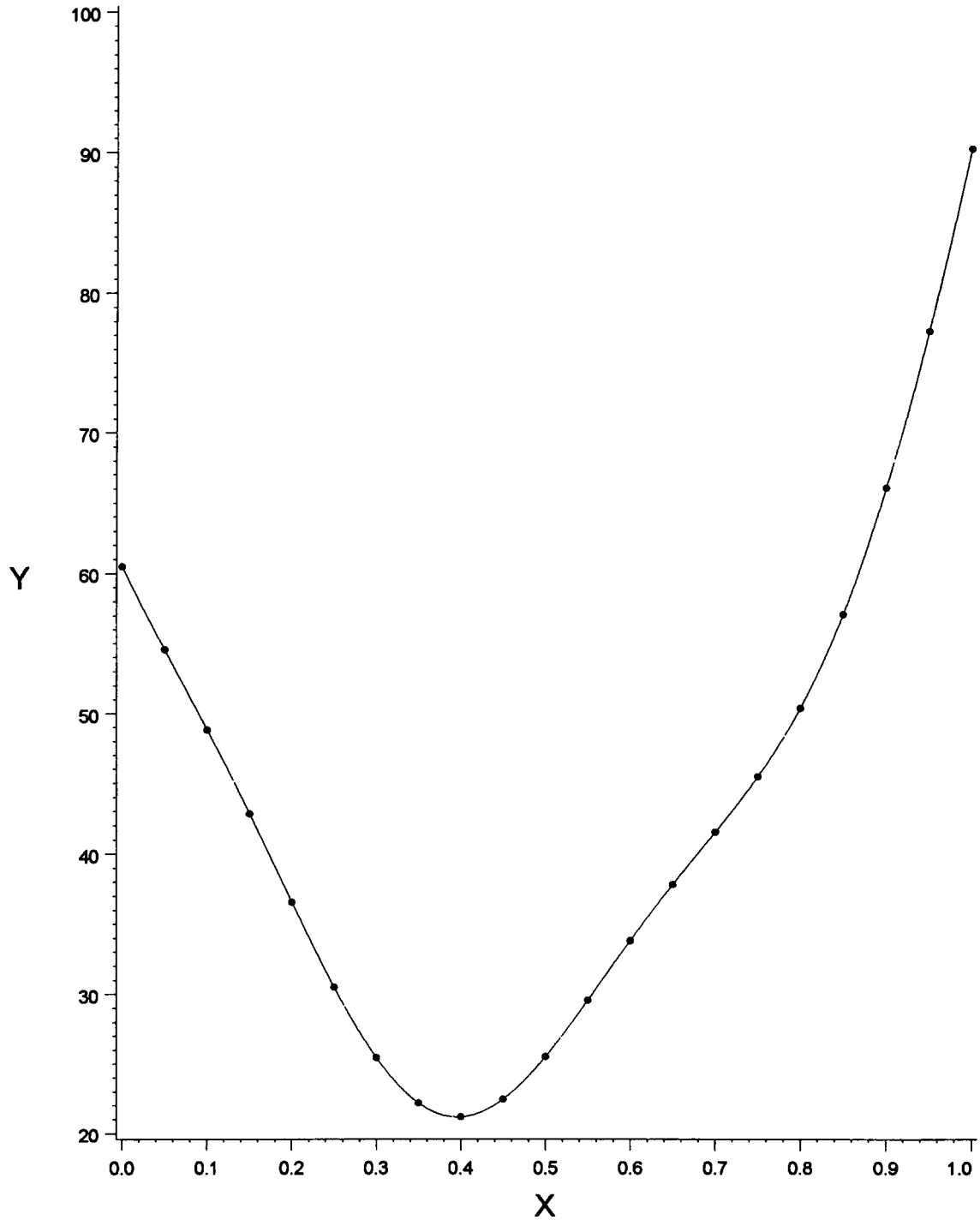


Figure 7.B.1. Plot of generated data and true curve for Data5.

[••• Raw data —— True curve]

Table 7.B.1. Comparison of optimal AVEMSE with optimal INTMSE. Values are for the fits based on optimal bandwidths and mixing parameters (h_o and λ_o which minimize AVEMSE).

		$AVEMSE_o$	$INTMSE_o$
Data1	OLS	9.885	9.417
	LLR	9.848	8.672
	MRR1	8.341	7.658
	MRR2	8.386	7.573
	PLR	8.711	7.604
Data2	OLS	5.385	5.585
	LLR	.922	.986
	MRR1	.912	1.012
	MRR2	.873	.845
	PLR	.875	.848
Data3	OLS	7.485	7.373
	LLR	5.575	4.964
	MRR1	5.066	4.693
	MRR2	4.909	4.415
	PLR	4.997	4.404
Data4	OLS	.340	.298
	LLR	.352	.322
	MRR1	.271	.244
	MRR2	.277	.245
	PLR	.281	.250
Data5	OLS	5.003	4.786
	LLR	.566	.527
	MRR1	.566	.527
	MRR2	.503	.467
	PLR	.487	.451

fitting technique, for each of the five data set examples. Another reason for using AVEMSE is because AVEMSE is minimized to obtain h_o and λ_o, and it would be beneficial to know exactly how close the AVEMSE value from a data-driven method is to this optimal minimum value. (In regard to INTMSE, it is quite possible that the INTMSE from the fit based on h_o and λ_o is not the minimum INTMSE possible). Thus, the results to come are based on reporting the chosen h's and λ's with their AVEMSE's, and comparing these values to the optimal values of h_o, λ_o, and the corresponding minimum AVEMSE.

7.C PRESS* Results

The first data-driven method analyzed is PRESS*, which was introduced in section 3.B.3. Recall that PRESS* is obtained by first calculating PRESS, and then penalizing this value for small bandwidths by dividing by $n - \text{tr}(\mathbf{H})$. Thus, PRESS* is protecting against fits that are too variable (i.e., slightly favoring larger h's and smaller λ's). Notice that no penalty is present for bias (to protect against h's too large or λ's too small). Before presenting the diagnostics comparing PRESS* to the optimal AVEMSE, a key observation needs to be discussed. This deals with the minimization of PRESS* as a function of h. Through many preliminary studies, it has been observed that PRESS* does not always follow a concave-up shape with an "ideal" minimum value. This happens on occasion when choosing h for MRR2 or PLR. In fact, there have been four patterns observed for the PRESS* curve as a function of h (from 0 to 1). These are displayed in Figures 7.C.1(a)-(d). Figure (a) shows PRESS* when selecting h for LLR for Data2, and the bandwidth is chosen as the h corresponding to the minimum of the curve. Figure (b) is PRESS* when selecting h for MRR2 for Data4, and once again the bandwidth is chosen as the h which minimizes the curve. Plots like (c) and (d) are the ones that cause problems for PRESS*. Figure (c) shows PRESS* when selecting h for MRR2 for Data3, and figure (d) shows PRESS* when selecting h for MRR2 for Data2. In both of these situations, just taking the bandwidth that minimizes PRESS* would result in h=1, which is a poor choice

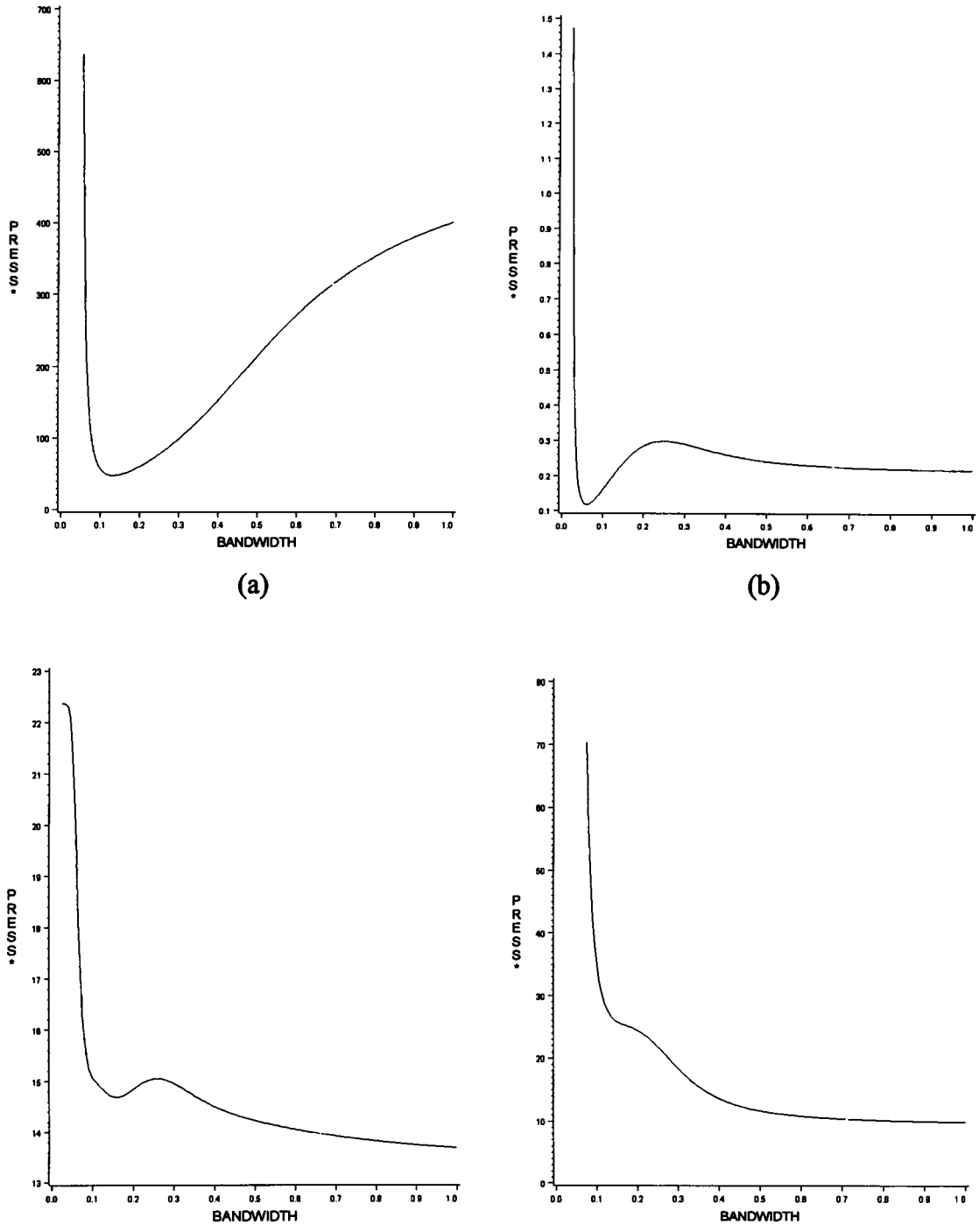


Figure 7.C.1 (a)-(d). Possible patterns for the PRESS* curve as a function of bandwidth.

of h. To resolve this possible problem of choosing h=1, it is suggested that the bandwidth be chosen as the h at the point of the *first* local minimum or at the point where the PRESS* curve first starts leveling off (i.e., where the downward slope becomes significantly less). This idea follows closely the method used in ridge regression to choose the "shrinkage parameter" k (Myers (1990)). This graphical approach to choosing h of course involves some judgment from the user, but for most preliminary examples studied it has been rather obvious how to choose the h. For the diagnostics below, all bandwidths for which PRESS* is really minimized at h=1 are denoted with a superscript of "1". The importance of noting these special occurrences becomes apparent later in this chapter.

Now for the results of PRESS*. Table 7.C.1 gives the h, λ, and AVEMSE values based on PRESS*, along with the optimal values for comparisons. (The row labeled LLR_{M2} is for choosing h for the LLR fit to the residuals in MRR2, with (AVEMSE) based on the residual fits). The final column points out where PRESS* chooses h or λ too large or too small (a "+" for too large, a "–" for too small). Double pluses or minuses indicate larger discrepancies. PRESS* performs well for Data5, but rather poorly for the other examples. Notice that most problems arise out of PRESS* choosing h too large, and on three occasions choosing h=1. For Data1 and Data2, the bandwidths are much too large. The conjecture made here is that introducing into PRESS* only a penalty for small bandwidths and no penalty for large bandwidths is the cause for the large h's seen in these examples (and especially for the h's =1). These inappropriate fits result in the model-robust procedures no longer significantly outperforming the individual OLS and LLR methods, and thus need to be improved upon. It appears that some action should be taken to try to reduce (or at least control) the size of h chosen by PRESS*. Several criteria addressing this issue are studied shortly. Do note, though, that PRESS* chooses h too small for Data4, showing that PRESS* does not always choose h too large. It will be difficult to find a criterion that overcomes the large bandwidth problem of PRESS* and is still able to fit well to Data4. A final point is that it is difficult to get a good impression of how well PRESS* selects λ without having the proper h's for each example. This may be

Table 7.C.1. Comparing h, λ, and AVEMSE values from PRESS* to optimal values. Note that MRR1 uses h from LLR, and MRR2 uses h from LLR_{M2}. Final column denotes whether the particular h or λ is too large or too small.

					PRESS*			+ if High
		h_o	λ_o	$AVEMSE_o$	h	λ	AVEMSE	– if Low
Data1	OLS			9.885			9.885	
	LLR	.115		9.848	.158		11.297	+
	MRR1		.503	8.341		.298	9.145	
	LLR_{M2}	.152		(3.828)	.360 [1]		(5.013)	+ +
	MRR2		.713	8.386		.288	9.820	
	PLR	.153		8.711	.4715		9.866	+ +
Data2	OLS			5.385			5.385	
	LLR	.065		.922	.133		3.292	+
	MRR1		.957	.912		0	5.385	– –
	LLR_{M2}	.073		(.573)	.135 [1]		(1.507)	+
	MRR2		1	.873		0	5.385	– –
	PLR	.0725		.875	.155 [1]		2.286	+
Data3	OLS			7.485			7.485	
	LLR	.099		5.575	.088		5.695	
	MRR1		.686	5.066		.566	5.009	
	LLR_{M2}	.119		(2.557)	.159		(2.897)	+
	MRR2		.879	4.909		.763	5.503	
	PLR	.1185		4.997	.155		5.285	+
Data4	OLS			.340			.340	
	LLR	.086		.352	.049		.471	–
	MRR1		.479	.271		.996	.468	
	LLR_{M2}	.140		(.087)	.061		(.205)	–
	MRR2		.961	.277		1	.395	
	PLR	.1405		.281	.0575		.418	–
Data5	OLS			5.003			5.003	
	LLR	.050		.566	.051		.566	
	MRR1		1	.566		1	.566	
	LLR_{M2}	.057		(.313)	.052		(.320)	
	MRR2		1	.503		1	.511	
	PLR	.060		.487	.050		.510	

[[1] denotes criterion is globally minimized at $h = 1$]

studied more in the future, but for the current work more emphasis is placed on selecting h. Notice, though, that for the few cases where h is chosen appropriately, λ is also chosen appropriately. This is a good sign that λ may be easier to select than h, but more work is needed on this.

7.D PRESS** Results

A criterion that maintains the penalty for variance, but also includes a penalty for bias (large h's) is PRESS**. PRESS** is described in section 3.B.3 and is expressed as equation (3.B.22). This criterion is designed to still prevent the selection of h's close to zero (as PRESS* does), while at the same time penalizing a little more as h starts to get larger. This should usually provide choices of h that are at least a little smaller than those chosen by PRESS* (and λ's that are a little larger than those from PRESS*, for comparable h's). This balancing of penalties for both bias and variance is but one of the two main advantageous properties of PRESS**. The second is the virtual elimination of the problem of selecting h=1. In other words, PRESS** (almost always) corrects for the structure shown in Figures 7.C.1 (c) and (d) by increasing the values of the curve for larger h's. This is an important consideration when executing the simulations in the next chapter. There, search routines are used to find h and λ for 500 data sets per simulation, and it is not practical to look at PRESS* or PRESS** curves (plotted vs. h) for each of these data sets. Thus, it is difficult to control the selection of h=1 if the curves are of the forms in Figures 7.C.1 (c) and (d). Wise selection of the starting values of h in the search routine (explained below) can overcome problems like figure (c), but figure (d) curves are much more problematic. This problem is *much* less prevalent in PRESS** than in PRESS*, as shown later. Also, it should by noted that PRESS** is on rare occasions minimized by an inappropriately small h (e.g., .015, .03, ...). In these cases, the PRESS** curve starts out at one or two small values (for small h), but then abruptly changes into a curve the shape of those in Figures 7.C.1 (a) or (b). An example is shown in Figure 7.D.1, which shows the PRESS** curve (as a function of h) for PLR for Data4. The initial small

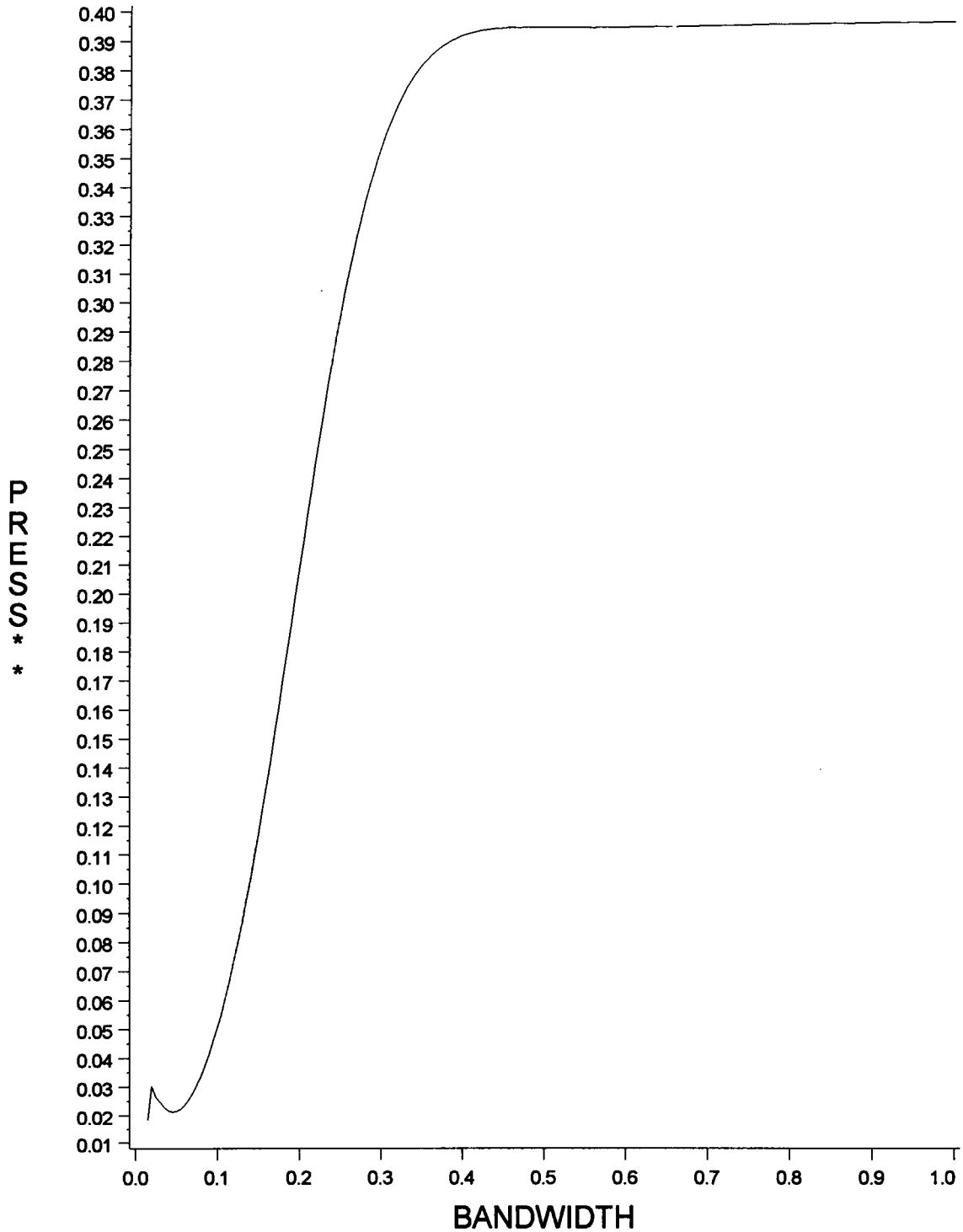


Figure 7.D.1. Plot of PRESS** curve as a function of bandwidth for PLR fit of Data4.

value is a result of the PRESS value in the numerator of PRESS**, due to the fact that PRESS may give unusual values for extremely small h's. The denominator n–tr(**H**) in PRESS* eliminates this problem by approaching zero for small h, but the second penalty term in PRESS** prevents its denominator from going to zero and thus does not eliminate this problem. However, proper choice of starting values in the search routine for h does easily prevent this choice of small h. (The strategy for these starting values in the search is to begin with the bandwidth values .08, .10, and .12 so that the search will move in only very small increments (.02 max.), finding the appropriate local minimum, and not reaching the location of the inappropriate global minimum. This strategy also eliminates the problem in graphs such as that in Figure 7.C.1 (c) (for PRESS* or possibly PRESS**) by selecting the bandwidth at the first local minimum).

The diagnostics for PRESS** are given in Table 7.D.1. For data sets 1, 2, 3, and 5, the values resulting from PRESS** generally are relatively close to the optimal values. In these examples, the advantages of using the model-robust procedures are not lost (as they often were with PRESS*). Also, in many cases the PRESS** fits are much better than the corresponding PRESS* fits, most noticeable in Data1 and Data2. Notice the smaller h's and larger λ's for PRESS** compared to PRESS*. This results in some cases where PRESS* slightly outperforms PRESS** (e.g., Data5). Unfortunately Data4 is also fit poorly by PRESS**. However, in considering all of the examples, PRESS** is much more consistent than PRESS*, and PRESS** has *no* h's chosen as 1. It appears that PRESS** (or future modifications of it) has potential as a useful data-driven selector of h and λ.

7.E Other Criteria

This section mentions some other data-driven criteria that have been studied, but found to not perform as well as PRESS** (or even PRESS* in several cases). These criteria are the usual MSE, generalized cross-validation, a standardized PRESS*, and an "average" of PRESS and PRESS*.

Table 7.D.1. Comparing h, λ, and AVEMSE values from PRESS to optimal values.** Final column denotes whether the particular h or λ is too large or too small.

					PRESS**			+ if High
		h_o	λ_o	$AVEMSE_o$	h	λ	AVEMSE	– if Low
Data1	OLS			9.885			9.885	
	LLR	.115		9.848	.120		9.868	
	MRR1		.503	8.341		.770	8.830	
	LLR_{M2}	.152		(3.828)	.145		(3.840)	
	MRR2		.713	8.386		.780	8.348	
	PLR	.153		8.711	.165		8.736	
Data2	OLS			5.385			5.385	
	LLR	.065		.922	.040		.999	-
	MRR1		.957	.912		.615	1.320.	-
	LLR_{M2}	.073		(.573)	.100		(.820)	
	MRR2		1	.873		.915	1.274	
	PLR	.0725		.875	.100		1.124	
Data3	OLS			7.485			7.485	
	LLR	.099		5.575	.080		5.957	
	MRR1		.686	5.066		.890	5.464	
	LLR_{M2}	.119		(2.557)	.080		(3.407)	-
	MRR2		.879	4.909		.940	5.536	
	PLR	.1185		4.997	.075		6.143	-
Data4	OLS			.340			.340	
	LLR	.086		.352	.040		.584	–
	MRR1		.479	.271		1	.584	
	LLR_{M2}	.140		(.087)	.050		(.274)	– –
	MRR2		.961	.277		1	.464	
	PLR	.1405		.281	.045		.517	– –
Data5	OLS			5.003			5.003	
	LLR	.050		.566	.040		.622	-
	MRR1		1	.566		1	.622	
	LLR_{M2}	.057		(.313)	.040		(.413)	-
	MRR2		1	.503		1	.604	
	PLR	.060		.487	.040		.603	-

MSE, GCV

The first of these alternatives is to choose h and λ to minimize the classical MSE formula,

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n - \mathrm{tr}(\mathbf{H})} = \frac{\mathrm{SSE}}{n - \mathrm{tr}(\mathbf{H})} .$$

This selector results in always choosing the bandwidth too small, which is due to having SSE in the numerator instead of PRESS (which is in PRESS*). A second alternative for selecting h and λ is generalized cross-validation, expressed as

$$GCV = \frac{\mathrm{PRESS}}{[n - \mathrm{tr}(\mathbf{H})]^2}$$

(Myers (1990)). This is just PRESS* with the penalty for small h's entering as a squared term. This criterion results in bandwidths that are always larger than those for PRESS*, which results in worse fits most of the time. Also, GCV has more problems with choosing h=1.

*Standardized PRESS**

A third criterion, which has performed better than the MSE or GCV techniques just described, is a standardized PRESS*. This is simply Standardized PRESS (described below), divided by the penalty $n - \mathrm{tr}(\mathbf{H})$. Standardized PRESS is defined as the sum of the standardized PRESS residuals, which are the regular PRESS residuals each divided by its standard deviation (or estimated standard deviation). It can be shown that the PRESS residuals for each fitting technique in this work may be expressed as

$$e_{i,-i} = y_i - \hat{y}_{i,-i} = \frac{e_i}{1 - h_{ii}} ,$$

where "-i" denotes "without using the i^{th} observation," and e_i is the usual residual from the regression fit. Then standardized PRESS residuals are defined as

$$\text{Std(PRESS}_i) = \frac{e_{i,-i}}{\sqrt{\text{Var}(e_{i,-i})}} = \frac{e_i}{(1-h_{ii})\sqrt{\text{Var}\left(\frac{e_i}{1-h_{ii}}\right)}} =$$

$$= \frac{e_i}{(1-h_{ii})\sqrt{\frac{1}{(1-h_{ii})^2}\text{Var}(e_i)}} = \qquad (7.E.1)$$

$$= \frac{e_i}{\frac{(1-h_{ii})}{(1-h_{ii})}\sqrt{\text{Var}(e_i)}} = \frac{e_i}{\sqrt{\text{Var}(e_i)}} \quad ,$$

the regular standardized residuals. Letting $\hat{\mathbf{y}}^{(\bullet)} = \mathbf{H}^{(\bullet)}\mathbf{y}$ for "$\bullet$" = OLS, LLR, MRR1, or PLR, so that $\mathbf{e}^{(\bullet)} = \mathbf{y} - \hat{\mathbf{y}}^{(\bullet)} = (\mathbf{I} - \mathbf{H}^{(\bullet)})\mathbf{y}$, it is clear that $\text{Var}(e_i) = \sigma^2[(\mathbf{I} - \mathbf{H}^{(\bullet)})(\mathbf{I} - \mathbf{H}^{(\bullet)})']_{ii}$ for each of these fitting procedures. For MRR2, there is an extra step involved in finding $\text{Var}(e_i)$. Letting $\mathbf{r} = (\mathbf{I} - \mathbf{H}^{(\text{ols})})\mathbf{y}$ be the residuals from the initial OLS fit, and then defining $\mathbf{e} = (\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})\mathbf{r}$ to be the residuals from the MRR2 local linear fit to $\mathbf{r}$, one obtains the following:

$$\begin{aligned}\text{Var}(\mathbf{e}) &= (\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})\text{Var}(\mathbf{r})(\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})' = \\ &= (\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})\,\sigma^2(\mathbf{I} - \mathbf{H}^{(\text{ols})})(\mathbf{I} - \mathbf{H}^{(\text{ols})})'\,(\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})' \\ &= \sigma^2(\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})(\mathbf{I} - \mathbf{H}^{(\text{ols})})(\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})' \quad ,\end{aligned}$$

and $\text{Var}(e_i) = [\sigma^2(\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})(\mathbf{I} - \mathbf{H}^{(\text{ols})})(\mathbf{I} - \mathbf{H}_2^{(\text{LLR})})']_{ii}$ for MRR2. Substituting the appropriate $\text{Var}(e_i)$ into equation (7.E.1) and summing these standardized PRESS residuals gives Standardized PRESS for each fitting technique, which facilitates the calculation of *Standardized PRESS**.

The diagnostics for Standardized PRESS* are included in Table 7.E.1. These results are very similar to those from PRESS*. There are some areas of improvement (e.g., Data1), and some areas where the fits are worse (e.g., Data3). Also, there is still a problem with choosing h=1 (three times), and one h (for PLR for Data1) is chosen too high at an inappropriate minimum of the Std. PRESS* curve. On the whole it does not

Table 7.E.1. Values of h, λ, and AVEMSE from Standardized PRESS* and AVEPRESS selection criteria.

		Standardized PRESS*			AVEPRESS		
		h	λ	AVEMSE	h	λ	AVEMSE
Data1	OLS			9.885			9.885
	LLR	.130		10.035	.115		9.848
	MRR1		.100	9.401		.745	8.694
	LLR_{M2}	.180 [1]		(3.945)	.190 [1]		(4.025)
	MRR2		.625	8.716		.575	8.864
	PLR	.190		8.897	.210		9.074
Data2	OLS			5.385			5.385
	LLR	.120		2.503	.105		1.810
	MRR1		0	5.385		.530	2.726
	LLR_{M2}	.135 [1]		(1.507)	.110 [1]		(.988)
	MRR2		0	5.385		.830	1.666
	PLR	.135 [1]		1.812	.115 [1]		1.387
Data3	OLS			7.485			7.485
	LLR	.065		6.912	.075		6.217
	MRR1		.900	6.179		.840	5.428
	LLR_{M2}	.070		(4.091)	.080		(3.407)
	MRR2		.950	6.161		.910	5.419
	PLR	.070		6.507	.080		5.831
Data4	OLS			.340			.340
	LLR	.050		.466	.040		.584
	MRR1		.995	.464		1	.584
	LLR_{M2}	.055		(.238)	.045		(.323)
	MRR2		1	.428		1	.513
	PLR	.050		.469	.045		.517
Data5	OLS			5.003			5.003
	LLR	.050		.566	.040		.622
	MRR1		1	.566		1	.622
	LLR_{M2}	.050		(.327)	.040		(.413)
	MRR2		1	.517		1	.604
	PLR	.050		.510	.040		.603

[[1] denotes criterion is globally minimized at $h = 1$]

appear that Std. PRESS* is noticeably better than PRESS*, and it definitely is not performing as well as PRESS**.

AVEPRESS

The final alternative for choosing h and λ has as its origin the desire to decrease the size of the h's chosen by PRESS*. This is done by averaging PRESS* with PRESS (which is known to give small h's) to form an "average PRESS," denoted *AVEPRESS*. This criterion can be expressed as

$$\begin{aligned}\text{AVEPRESS} &= (\text{PRESS} + \text{PRESS}^*)/2 \\ &= \frac{1}{2}\left(\frac{[n - \text{tr}(\mathbf{H})]\ (\text{PRESS})}{n - \text{tr}(\mathbf{H})} + \text{PRESS}^*\right) \\ &= \tfrac{1}{2}\left([n - \text{tr}(\mathbf{H})]\ \text{PRESS}^* + \text{PRESS}^*\right) \\ &= \tfrac{1}{2}[n - \text{tr}(\mathbf{H}) + 1]\text{PRESS}^* .\end{aligned}$$

Performance diagnostics are given in Table 7.E.1. As expected (by design), the h's are smaller for AVEPRESS than for PRESS*. This results in several improvements for AVEPRESS, but of course makes the fits for Data4 even worse. Also, using AVEPRESS does not solve the problem of selecting h=1 (or another inappropriately high value), and the fits are not quite as good overall as those from PRESS**.

In summary, it appears that PRESS** has the best potential as a data-driven selector of h and λ for the fitting techniques developed in this research. It does not solve every problem, but does usually give adequate fits that maintain the advantages of the model-robust procedures over the individual parametric and nonparametric procedures. Of course, there is much more work needed to more thoroughly investigate PRESS** (and PRESS*), and possibly make improvements or even find better methods altogether. A brief simulation study is performed on PRESS* and PRESS** in the next chapter in order to substantiate the findings reported thus far.

Chapter 8: Simulation Results

8.A Introduction

This chapter makes use of Monte Carlo simulations to further study the various fitting techniques and to substantiate the many observations made in previous chapters. All results thus far have been based on *single* data sets and on derived "theoretical" formulas (for MSE values). Thus, there are two main areas that need to be checked for accuracy. First, a comparison needs to be made of the theoretical MSE values based on equations derived in section 6.B to the simulated MSE values based on fitting *many* data sets (for a given underlying model). (For all simulations presented here, S=500 simulated data sets are used for each Monte Carlo example). These comparisons will actually be a check on the accuracy of the INTMSE values for several examples, where INTMSE is based on the MSE formulas being verified. Recall that the MSE formulas (and INTMSE) do not depend on the generated data, but only on the true underlying model. However, INTMSE is estimating the true integrated MSE that would be based on the MSE calculations of all possible data sets. Thus, Monte Carlo simulations of many data sets are needed to study MSE, even though the formulas being checked do not themselves depend on the data. The second topic studied through these simulations involves the quantities and diagnostics that are actually determined by the data. These include the bandwidths and mixing parameters chosen by data-driven methods, and the diagnostics such as df_{model} and PRESS that are evaluated based on the data. The conclusions reached in previous chapters, although informative and very useful as preliminary studies, are only based on one of an infinite number of possible data sets for a given underlying model. This makes the results at least a little suspect, and more reliable results are needed. These are obtained by running Monte Carlo simulations with 500 simulated data sets and calculating the average values of the quantities in question across all 500 data sets. This technique provides a much better idea of the values of the diagnostics and data-driven selection quantities that one expects to see on average for a particular example.

8.A.1 Examples Used (Data)

The simulation examples studied here are based on a model closely related to model (6.C.1) of Example 1 in Chapter 6. The actual underlying model used here to generate the Monte Carlo data sets is

$$y = 2(X - 55)^2 + 5X + \gamma\left[10 \sin\left(\frac{\pi(X-1)}{225}\right)\right] + \varepsilon \,, \qquad (8.A.1)$$

where $\varepsilon \sim N(0,16)$, and γ is the *misspecification parameter*, which is taken to range from 0 to 1. This model is a quadratic model with a certain amount of deviation introduced by the sine function term. The amount of deviation is controlled by γ. Assuming that a quadratic OLS model is fit for all of these examples, γ is actually determining the amount of misspecification present in the chosen model. γ is varied for the simulations as 0, 0.25, 0.5, 0.75, and 1.0 in order to give a wide range of misspecifications to be studied. (Taking γ=3.5 yields model (6.C.1) of Example 1). Note that γ=0 results in no misspecification, and OLS should perform best. As γ increases, OLS performs more poorly, and nonparametric techniques (LLR) should prove more beneficial. The main interest, though, is in how the model-robust procedures perform across this range of misspecifications. It is shown shortly that they perform very well. A plot of the various true curves, determined by γ, is given in Figure 8.A.1. In addition to varying the misspecification level, the sample size n is also varied for the simulations. The three sample sizes used are n=6, 10, and 19, providing for a range of small to moderately large data sets for comparisons. For all examples, X-values are taken at evenly spaced locations from 1 to 10 (in increments of 1.8 for n=6, increments of 1.0 for n=10, and increments of 0.5 for n=19).

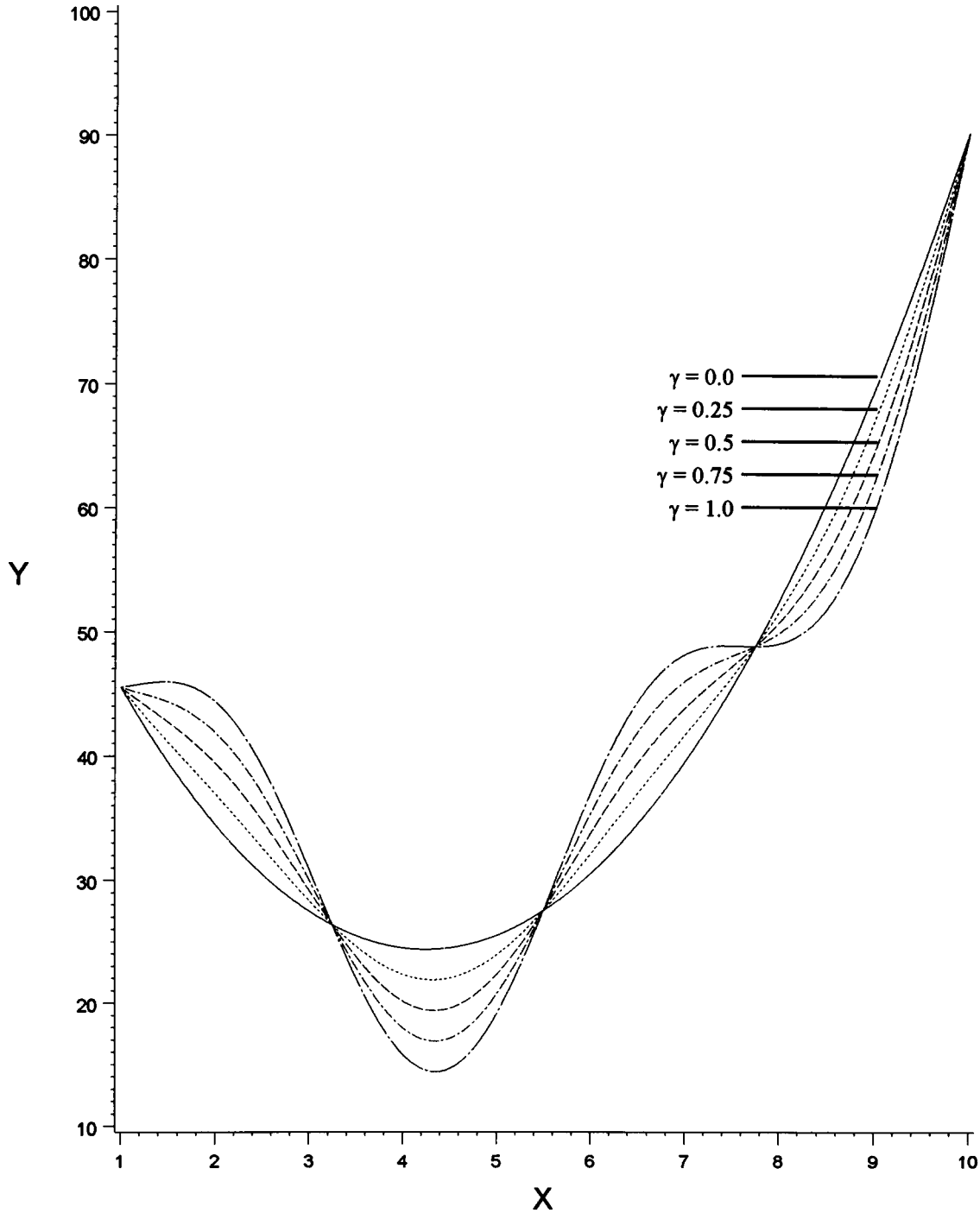


Figure 8.A.1. Underlying curves from model (8.A.1), as a function of γ.

[—— $\gamma = 0.0$ ····· $\gamma = 0.25$ ----- $\gamma = 0.5$ -·-·-· $\gamma = 0.75$ —·—·— $\gamma = 1.0$]

8.A.2 Progression of Study

The Monte Carlo simulation study is carried out in the following order. In section 8.B, the theoretical MSE formulas are compared to simulated MSE's and checked for accuracy. These results are based on the optimal bandwidths and λ's (from minimizing AVEMSE). Then in section 8.C, these optimal fits are used in comparing the various fitting techniques. Performance diagnostics and confidence interval results are presented for these fits. Section 8.D then presents simulation results for the data-driven bandwidth and λ selection methods of PRESS* and PRESS**. Values for h, λ, and INTMSE are given for comparison with the optimal values in section 8.C. Confidence interval results are also given for the better performing PRESS** method. Conclusions are presented in section 8.E.

8.B Accuracy of Theoretical MSE Formulas

Recall that INTMSE is the average of the MSE values of the fitted values at many (1000, thus far) locations across the entire range of the data. That is, INTMSE is an estimate of the integrated MSE of the regression curve. All preliminary results which have favored the model-robust procedures (MRR2 in particular) in previous chapters have been based mainly on INTMSE values. Based on comparisons with the "simulated MSE" (*SIMMSE*) of 500 data sets per simulation, it is shown here that these derived theoretical formulas appear to be *extremely* accurate. All results are based on using the optimal h_o's and λ_o's for the bandwidths and mixing parameters to determine the fits. These optimal values for the simulation examples are given in Table 8.B.1. Before proceeding, a few observations should be pointed out regarding the h_o and λ_o values. First, as n increases (for a fixed γ), h_o decreases and λ_o generally increases. (The exception is at γ=0, where λ_o actually decreases slightly, but remains close to zero, always resulting in fits close to OLS). These properties are due to LLR performing better (fits with smaller variance, allowing smaller h_o) when there is more data, and thus being used as a larger component of the model-robust fits. Second, as the misspecification (γ) increases, h_o decreases and λ_o

Table 8.B.1. Optimal bandwidths and mixing parameters for the model-robust fitting procedures. The optimal h_o for MRR1 is also h_o for LLR.

		MRR1		MRR2		PLR
n	γ	h_o	λ_o	h_o	λ_o	h_o
6	0	.146	.020	1	.016	1
	.25	.139	.256	1	.016	1
	.5	.126	.656	.140	.754	.140
	.75	.115	.851	.120	.890	.120
	1	.108	.924	.110	.939	.110
10	0	.130	.013	1	.016	1
	.25	.122	.301	.226	.478	1
	.5	.105	.751	.118	.884	.118
	.75	.091	.946	.095	.996	.095
	1	.082	1	.083	1	.083
19	0	.113	.009	1	.031	1
	.25	.104	.458	.157	.739	.158
	.5	.089	.892	.099	.996	.099
	.75	.077	1	.080	1	.080
	1	.068	1	.069	1	.070

increases. This reflects the improvement in LLR over OLS as the data departs more and more from quadratic, and the need for smaller bandwidths to pick up extra structure. Also note that λ_o is close to zero for γ=0 (thus using mainly the optimal fitting OLS), and λ_o is close to one for γ=1 (using mainly the better fitting LLR). These are very promising results, because they show that the model-robust techniques are properly mixing the individual parametric and nonparametric techniques. As seen in Table 8.B.2, this proper mixing allows the model-robust procedures to perform as well as OLS when there is no model misspecification. When there is large misspecification they perform at least as well as LLR. For small to moderate misspecification, h_o and λ_o are used by the model-robust procedures to give improved fits over OLS and LLR.

The simulated (average) mean squared error for a particular fitting procedure is calculated according to the following steps. For each of the S=500 simulated data sets, first determine the fitted values at "many" x_o locations (on the transformed scale of [0,1]). For these simulation results, 250 x_o locations are fit instead of 1000, as was done in previous chapters (the results are very similar, and using 250 gives much faster Monte Carlo runs, which still provide very adequate results). After obtaining these fits ($\hat{y}_i$'s), next compute the *average squared error* (*ase*) given by

$$ase = \frac{\sum_{i=1}^{250}\left(\mathrm{E}(y_i) - \hat{y}_i\right)^2}{250}$$

for each of the 500 simulations (i.e., get ase_j, for j=1, 2, ..., 500), where E(y_i) is the true value from the underlying function (without the error term). The Monte Carlo (simulated) average mean squared error is then given by

$$\mathrm{SIMMSE} = \frac{\sum_{j=1}^{500} ase_j}{500} . \qquad (8.B.1)$$

Table 8.B.2 gives the results for the simulated MSE values along with the theoretical INTMSE values for the simulation examples. The values are very similar to

Table 8.B.2. Simulated mean squared error values for optimal fits from 500 Monte Carlo runs. Theoretical INTMSE values are in bold.

n	γ	OLS	LLR	MRR1	MRR2	PLR
6	0	6.069 **6.384**	11.254 **11.789**	6.059 **6.382**	6.069 **6.384**	6.080 **6.391**
	.25	8.876 **9.295**	12.054 **12.695**	8.470 **8.985**	8.876 **9.295**	8.891 **9.301**
	.5	17.505 **18.028**	14.510 **15.220**	13.436 **14.143**	13.084 **13.596**	13.223 **13.659**
	.75	31.956 **32.583**	18.391 **19.190**	18.478 **19.293**	17.646 **18.229**	17.377 **17.907**
	1	52.228 **52.959**	23.723 **24.630**	24.206 **25.127**	23.231 **23.915**	22.761 **23.408**
10	0	3.987 **4.105**	7.430 **7.689**	3.985 **4.104**	3.987 **4.105**	3.992 **4.110**
	.25	6.721 **6.818**	8.015 **8.243**	6.172 **6.300**	6.388 **6.490**	6.729 **6.825**
	.5	14.881 **14.956**	9.264 **9.456**	9.105 **9.262**	8.772 **8.884**	8.688 **8.867**
	.75	28.466 **28.520**	10.555 **10.721**	10.660 **10.819**	10.282 **10.403**	10.283 **10.450**
	1	47.477 **47.509**	11.736 **11.883**	11.736 **11.883**	11.562 **11.675**	11.571 **11.722**
19	0	2.274 **2.314**	4.504 **4.622**	2.273 **2.314**	2.273 **2.314**	2.270 **2.316**
	.25	4.937 **4.973**	4.885 **4.971**	4.064 **4.104**	4.016 **4.041**	4.067 **4.080**
	.5	12.918 **12.951**	5.631 **5.695**	5.635 **5.680**	5.324 **5.348**	5.362 **5.381**
	.75	26.218 **26.247**	6.376 **6.430**	6.376 **6.430**	6.225 **6.251**	6.261 **6.283**
	1	44.837 **44.861**	7.041 **7.089**	7.041 **7.089**	6.953 **6.979**	6.988 **7.010**

each other. In fact, even the INTMSE values with the largest discrepancy from SIMMSE are less than 5% different. For moderate to large sample sizes, the values are extremely close to each other. These results are very beneficial, since they provide evidence that the MSE formulas derived in Chapter 6 (for fixed h_o, λ_o) are indeed accurate. So all results reached earlier that were based on INTMSE seem appropriate, which gives support to the benefits of the model-robust procedures.

8.C Comparisons of Procedures Based on Optimal Fits

By using the optimal h_o and λ_o, the best possible fits for the various procedures are obtained. Presented here are performance diagnostics and confidence interval results for each of these "best" fits of each procedure. From these values, conclusions can be reached as to how well the "best" fit of one technique compares to the "best" fits of the other techniques.

8.C.1 Performance Diagnostics

Table 8.C.1 displays the data-dependent diagnostics of $df_{model} = tr(\mathbf{H})$ and PRESS for each fitting technique, averaged across the 500 Monte Carlo simulations. Also included for reference are the INTMSE values. The values df_{model} and PRESS measure the complexity and adequacy, respectively, of the particular fits. It is desired to have both of these values small, which signifies a fit that is not very complex (or variable) and is not overly dependent on individual data points when fitting at those particular locations. Also, one hopes that a fit having these properties will also have a low INTMSE, which measures the relative "theoretical" performances of the fitting techniques. Unfortunately, it is shown shortly that this is not always achieved (PRESS and INTMSE often are not in agreement when measuring model adequacy).

Several observations can be made from Table 8.C.1. For no misspecification (γ=0), df_{model} for each of the model-robust procedures is close to OLS. This is due to the model-robust procedures (correctly) obtaining fits similar to OLS. For large

Table 8.C.1. Diagnostics for fitting techniques based on optimal h_o and λ_o. Data-dependent diagnostics are df_{model} and PRESS, but INTMSE is also included as a reference as to which techniques are "theoretically" best.

n	γ		OLS	LLR	MRR1	MRR2	PLR
6	0	df_{model}	3	5.05	3.04	3.00	3.01
		PRESS	291.79	826.23	291.72	291.78	291.71.
		INTMSE	*6.384*	*11.789*	*6.382*	*6.384*	*6.391*
	.25	df_{model}	3	5.19	3.56	3.00	3.01
		PRESS	331.41	926.71	333.13	331.41	331.52
		INTMSE	*9.295*	*12.695*	*8.985*	*9.295*	*9.301*
	.5	df_{model}	3	5.45	4.61	4.67	5.23
		PRESS	455.64	1242.86	481.11	492.56	992.05
		INTMSE	*18.028*	*15.220*	*14.143*	*13.596*	*13.659*
	.75	df_{model}	3	5.65	5.25	5.30	5.59
		PRESS	664.46	1775.24	743.40	763.10	1762.08
		INTMSE	*32.583*	*19.190*	*19.293*	*18.229*	*17.907*
	1	df_{model}	3	5.76	5.55	5.58	5.75
		PRESS	957.89	2523.13	1110.12	1144.22	2843.23
		INTMSE	*52.959*	*24.630*	*25.127*	*23.915*	*23.408*

n	γ		OLS	LLR	MRR1	MRR2	PLR
10	0	df_{model}	3	5.82	3.04	3.00	3.01
		PRESS	249.78	438.03	249.77	249.78	249.83
		INTMSE	*4.105*	*7.689*	*4.104*	*4.105*	*4.110*
	.25	df_{model}	3	6.09	3.93	3.55	3.01
		PRESS	300.07	497.45	298.68	311.26	299.92
		INTMSE	*6.818*	*8.243*	*6.300*	*6.490*	*6.825*
	.5	df_{model}	3	6.76	5.82	5.87	6.29
		PRESS	448.59	660.10	422.12	442.12	680.90
		INTMSE	*14.956*	*9.456*	*9.262*	*8.884*	*8.867*
	.75	df_{model}	3	7.49	7.24	7.26	7.31
		PRESS	695.37	921.86	581.95	763.39	1007.05
		INTMSE	*28.520*	*10.721*	*10.819*	*10.403*	*10.450*
	1	df_{model}	3	8.09	8.09	8.00	8.02
		PRESS	1040.39	1287.96	1287.96	1314.16	1454.45
		INTMSE	*47.509*	*11.883*	*11.883*	*11.675*	*11.722*

(cont...)

Table 8.C.1 (continued)

n	γ		OLS	LLR	MRR1	MRR2	PLR
19	0	df_{model}	3	6.52	3.03	3.00	3.01
		PRESS	364.94	487.92	364.94	364.94	364.94
		INTMSE	*2.314*	*4.622*	*2.314*	*2.314*	*2.316*
	.25	df_{model}	3	6.93	4.80	4.61	5.19
		PRESS	426.90	510.18	415.10	421.77	445.55
		INTMSE	*4.973*	*4.971*	*4.104*	*4.041*	*4.080*
	.5	df_{model}	3	7.87	7.35	7.25	7.31
		PRESS	621.88	564.64	508.16	533.02	543.31
		INTMSE	*12.951*	*5.695*	*5.680*	*5.348*	*5.381*
	.75	df_{model}	3	8.88	8.88	8.61	8.65
		PRESS	949.88	623.63	623.63	609.22	620.20
		INTMSE	*26.247*	*6.430*	*6.430*	*6.251*	*6.283*
	1	df_{model}	3	9.78	9.78	9.62	9.64
		PRESS	1410.91	677.73	677.73	669.34	682.12
		INTMSE	*44.861*	*7.089*	*7.089*	*6.979*	*7.010*

misspecification (starting at γ=.75, or γ=.5 for n=19), df_{model} for the model-robust procedures is much closer to that of LLR. This is necessary (and correct) since the data being fit is more complex. The OLS df_{model} is always equal to 3, but INTMSE shows that OLS is very inadequate when misspecification is present. The real benefits of the model-robust procedures show up in the cases of small to moderate misspecification (γ=.25 and often γ=.5). Table 8.C.1 illustrates that df_{model} for the model-robust procedures remains rather low (much lower usually than LLR), but these procedures still make use of the nonparametric fits to reduce INTMSE. For example, this shows up very clearly for the case where n=19 and γ=.25. The LLR df_{model} is approximately 7, while the model-robust procedures have df_{model} values around 5 or less. The resulting INTMSE values are much lower for the model-robust procedures than either LLR or OLS (which maintains df_{model}=3). In conclusion, the model-robust procedures seem to be performing well in the sense of obtaining the best fits possible to capture the structure in the data, while at the same time remaining as simple (smooth) as possible. This is precisely the result of maintaining both low bias and low variance.

The PRESS values also provide for some interesting observations. For γ=0, the model-robust procedures are once again close to OLS, with LLR being much larger. In fact, the LLR PRESS remains large for all γ-values. As γ increases, the PRESS values for MRR1 and MRR2 get larger, but remain smaller than that for LLR. The difference in these values is very large for n=6 and becomes less as n increases to 10 and 19. The reason for this is that when fitting at a point x_o, the smaller the sample size, the more emphasis LLR places on that particular x_o. This characteristic is what inflates PRESS for small n. That is, the fit changes significantly when removing x_o and recalculating the fit. The model-robust procedures MRR1 and MRR2 make use of OLS to alleviate some of this weight placed on the point being fit, and this shows up in smaller PRESS values. Do note, however, that PRESS for PLR is often very large, especially as γ gets larger. This is an artifact of PLR always using the entire LLR fit in its development, which has already been mentioned as a drawback of the PLR procedure. In relation to OLS, the PRESS

values for MRR1 and MRR2 are slightly larger for n=6 (but as γ get larger, OLS gives poor fits based on INTMSE). This supports the contention made earlier that PRESS is not always in agreement with INTMSE in terms of diagnosing fits, and should not be relied on entirely for making such assessments. For larger sample sizes, MRR1 and MRR2 PRESS values are a little more reliable and tend to be lower than OLS. In summary, PRESS gives some support to the use of MRR1 or MRR2, and provides some reservations as to the overall performance of PLR.

PRESS* could also be considered as a diagnostic to compare the fits of the different techniques. PRESS* values have been obtained for all of the simulations described above. These values, while different in magnitude from PRESS, give the same information as PRESS in regard to ordering the model-robust procedures and LLR in terms of performance. One change when using PRESS* is that the values remain very low for OLS, even when γ becomes large. However, it is clear from INTMSE that these OLS fits (for γ large) are very poor. This misleading representation of performance in PRESS* (as well as the drawbacks of PRESS given above) emphasizes that data-dependent comparisons should be based on several diagnostics (not just one) in order to provide more confidence as to the true behavior of the fits.

8.C.2 Confidence Intervals

The final method for comparing the performances of the different fitting techniques is to observe the validity of the respective confidence intervals. The expressions for the confidence intervals studied here are those given in section 6.D by equation (6.D.2). In Chapter 6, the average C.I. widths for each fitting technique were compared for three examples, with the model-robust procedures performing well. To better understand the effectiveness of the C.I.'s, one needs to determine not only these widths, but also the coverage probabilities that are obtained by each procedure. This is accomplished through the 500 Monte Carlo simulations described above, where the fits are based on the optimal h_o and λ_o given in Table 8.B.1.

Table 8.C.2 supplies the diagnostics of interest when forming *95%* confidence intervals for each of the sample size (n) and misspecification (γ) combinations. To facilitate the study, three X-values have been chosen based on the curves in Figure 8.A.1, and confidence intervals have been constructed for each of these values. As seen in Table 8.C.2, these values are at the locations x_o = 2, 4, and 7. These values have been chosen due to their locations at points where there is much change in the underlying curves as γ is varied. In fact, these points have been chosen because it should be more difficult to obtain adequate fits at these locations than at most other locations, especially for large γ values. The diagnostics presented to compare the fits at these points across fitting procedures are the true y-value ($E(y_o)$) at the particular x_o, the (mean) fitted value (across the 500 Monte Carlo runs) at x_o, the (mean) confidence interval width at x_o, and the observed coverage probability of these C.I.'s at x_o. The observed coverage probability is the percentage of the C.I.'s in the 500 simulations that contain the true y-value at x_o. Of course, it is desired to have C.I. widths as small as possible (for precision), while still maintaining close to 95% coverage (for accuracy). Wide intervals with large coverage probabilities, as well as narrow intervals with small coverage probabilities are indications of a need for improving the procedures. Based on all of these considerations, many interesting conclusion can be drawn from the information in Table 8.C.2, and some of these are discussed below.

First, note that the C.I. widths for a given fitting technique (OLS, LLR, MRR1, MRR2, or PLR) are identical for the locations x_o= 4 and 7. This occurs because the variance portion of the confidence intervals (the $\mathbf{h}_o'\mathbf{h}_o$ term in (6.D.2)) happens to be the same for each of these points (also, the t-value and $\hat{\sigma}$ used in the C.I.'s for any X-values are always the same within a given fitting technique). An illustration of this occurrence is supplied by Figures 6.C.10 (c) and 6.C.11 (c) in Chapter 6. These figures show the variance curves (the same variances as used in the confidence intervals) for the model used in these simulations with γ = .35. For all fitting techniques, the variances are equal at the locations x_o= 4 and 7. With the variance portions equal, the only other values that determine C.I. widths are the t-value and $\hat{\sigma}$. For a given fitting technique, these values are

Table 8.C.2 (a)-(o). Confidence interval diagnostics for the various optimal fits for the 500 Monte Carlo runs. Values are reported for three X-locations: 2, 4, 7.

(a) $[n = 6 \quad \gamma = 0]$

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	34.5	34.55	15.086	.938
	4	24.5	24.42	13.833	.942
	7	39.5	39.31	13.833	.948
LLR	2	34.5	36.29	79.335	1
	4	24.5	26.16	80.801	1
	7	39.5	41.27	80.801	1
MRR1	2	34.5	34.58	15.102	.938
	4	24.5	24.46	13.831	.944
	7	39.5	39.34	13.831	.946
MRR2	2	34.5	34.55	15.086	.938
	4	24.5	24.42	13.834	.942
	7	39.5	39.31	13.834	.948
PLR	2	34.5	34.54	15.111	.938
	4	24.5	24.42	13.848	.944
	7	39.5	39.31	13.848	.948

(b) $[n = 6 \quad \gamma = .25]$

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	36.96	34.77	17.432	.932
	4	22.33	24.52	15.984	.944
	7	41.67	39.21	15.984	.920
LLR	2	36.96	36.99	125.619	1
	4	22.33	25.08	129.033	1
	7	41.67	42.10	129.033	1
MRR1	2	36.96	35.34	18.362	.958
	4	22.33	24.66	16.813	.940
	7	41.67	39.95	16.813	.948
MRR2	2	36.96	34.77	17.432	.932
	4	22.33	24.52	15.985	.944
	7	41.67	39.21	15.985	.920
PLR	2	36.96	34.77	17.473	.932
	4	22.33	24.51	16.013	.944
	7	41.67	39.21	16.013	.920

Table 8.C.2. (continued)

(c) [$n = 6$ $\gamma = .5$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	39.42	35.00	23.280	.938
	4	20.17	24.61	21.347	.936
	7	43.83	39.11	21.347	.916
LLR	2	39.42	37.74	577.503	1
	4	20.17	23.79	600.149	1
	7	43.83	43.06	600.149	1
MRR1	2	39.42	36.80	38.215	.998
	4	20.17	24.08	36.987	.990
	7	43.83	41.70	36.987	1
MRR2	2	39.42	35.77	40.957	.988
	4	20.17	22.96	41.523	.992
	7	43.83	40.78	41.523	.990
PLR	2	39.42	36.01	142.443	1
	4	20.17	22.41	148.570	1
	7	43.83	41.32	148.570	1

(d) [$n = 6$ $\gamma = .75$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	41.89	35.23	30.866	.956
	4	18.00	24.71	28.303	.956
	7	46.00	39.02	28.303	.942
LLR	2	41.89	38.54	8847.372	1
	4	18.00	22.57	9244.707	1
	7	46.00	44.14	9244.707	1
MRR1	2	41.89	38.04	175.033	1
	4	18.00	22.89	177.014	1
	7	46.00	43.37	177.014	1
MRR2	2	41.89	36.72	214.386	.998
	4	18.00	21.52	226.091	.998
	7	46.00	42.22	226.091	1
PLR	2	41.89	36.89	3168.754	1
	4	18.00	21.12	3338.471	1
	7	46.00	42.61	3338.471	1

Table 8.C.2. (continued)

(e) [$n = 6$ $\gamma = 1$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	44.35	35.45	39.198	.978
	4	15.84	24.81	35.944	.972
	7	48.16	38.92	35.944	.976
LLR	2	44.35	39.35	460385	1
	4	15.84	21.41	481998	1
	7	48.16	45.24	481998	1
MRR1	2	44.35	39.06	1953.368	1
	4	15.84	21.67	2011.022	1
	7	48.16	44.76	2011.022	1
MRR2	2	44.35	37.60	2658.005	1
	4	15.84	20.26	2841.556	1
	7	48.16	43.48	2841.556	1
PLR	2	44.35	37.73	255239	1
	4	15.84	19.96	269552	1
	7	48.16	43.77	269552	1

Table 8.C.2. (continued)

(f) [$n = 10$ $\gamma = 0$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	34.5	34.36	9.661	.950
	4	24.5	24.43	8.089	.954
	7	39.5	39.48	8.089	.950
LLR	2	34.5	35.62	12.409	.962
	4	24.5	25.80	12.017	.962
	7	39.5	40.85	12.017	.962
MRR1	2	34.5	34.37	9.648	.950
	4	24.5	24.45	8.073	.956
	7	39.5	39.50	8.073	.948
MRR2	2	34.5	34.36	9.661	.950
	4	24.5	24.43	8.089	.954
	7	39.5	39.48	8.089	.950
PLR	2	34.5	34.36	9.667	.952
	4	24.5	24.43	8.090	.956
	7	39.5	39.48	8.090	.952

(g) [$n = 10$ $\gamma = .25$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	36.96	34.93	10.745	.906
	4	22.33	24.68	8.997	.854
	7	41.67	39.24	8.997	.820
LLR	2	36.96	36.98	13.175	.984
	4	22.33	24.43	12.892	.932
	7	41.67	41.90	12.892	.982
MRR1	2	36.96	35.54	10.541	.924
	4	22.33	24.60	8.931	.844
	7	41.67	40.04	8.931	.890
MRR2	2	36.96	35.11	10.616	.914
	4	22.33	24.43	9.050	.880
	7	41.67	39.47	9.050	.862
PLR	2	36.96	34.92	10.754	.902
	4	22.33	24.67	9.000	.856
	7	41.67	39.24	9.000	.820

Table 8.C.2. (continued)

(h) [$n = 10$ $\gamma = .5$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	39.42	35.49	13.535	.850
	4	20.17	24.92	11.333	.648
	7	43.83	38.99	11.333	.652
LLR	2	39.42	38.62	15.613	.980
	4	20.17	22.51	15.499	.946
	7	43.83	43.23	15.499	.984
MRR1	2	39.42	37.84	13.745	.956
	4	20.17	23.11	12.920	.878
	7	43.83	42.18	12.920	.952
MRR2	2	39.42	37.23	13.373	.926
	4	20.17	22.29	13.128	.932
	7	43.83	41.65	13.128	.918
PLR	2	39.42	37.49	14.342	.946
	4	20.17	21.96	13.909	.946
	7	43.83	42.04	13.909	.942

(i) [$n = 10$ $\gamma = .75$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	41.89	36.06	17.253	.852
	4	18.00	25.16	14.446	.488
	7	46.00	38.75	14.446	.480
LLR	2	41.89	40.66	20.133	.992
	4	18.00	20.31	20.096	.980
	7	46.00	44.99	20.096	.994
MRR1	2	41.89	40.41	19.168	.986
	4	18.00	20.56	18.951	.970
	7	46.00	44.65	18.951	.986
MRR2	2	41.89	39.83	18.462	.974
	4	18.00	19.85	18.646	.976
	7	46.00	44.15	18.646	.974
PLR	2	41.89	39.86	19.199	.978
	4	18.00	19.84	18.780	.978
	7	46.00	44.20	18.780	.976

Table 8.C.2. (continued)

(j) [$n = 10$ $\gamma = 1$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	44.35	36.63	21.423	.868
	4	15.84	25.41	17.938	.386
	7	48.16	38.51	17.938	.396
LLR	2	44.35	42.98	27.821	.998
	4	15.84	17.99	27.809	.998
	7	48.16	47.02	27.809	1
MRR1	2	44.35	42.98	27.821	.998
	4	15.84	17.99	27.809	.998
	7	48.16	47.02	27.809	1
MRR2	2	44.35	42.39	26.046	.998
	4	15.84	17.60	26.243	.996
	7	48.16	46.43	26.243	.996
PLR	2	44.35	42.41	27.006	.998
	4	15.84	17.61	26.564	.996
	7	48.16	46.46	26.564	.996

Table 8.C.2. (continued)

(k) [$n = 19$ $\gamma = 0$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	34.5	34.44	6.754	.958
	4	24.5	24.47	5.279	.958
	7	39.5	39.49	5.279	.958
LLR	2	34.5	35.21	7.746	.950
	4	24.5	25.54	7.394	.918
	7	39.5	40.51	7.394	.924
MRR1	2	34.5	34.44	6.750	.956
	4	24.5	24.48	5.274	.958
	7	39.5	39.50	5.274	.958
MRR2	2	34.5	34.44	6.754	.958
	4	24.5	24.47	5.279	.958
	7	39.5	39.49	5.279	.958
PLR	2	34.5	34.44	6.758	.956
	4	24.5	24.47	5.280	.956
	7	39.5	39.50	5.280	.956

(l) [$n = 19$ $\gamma = .25$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	36.96	35.20	7.350	.860
	4	22.33	24.80	5.745	.620
	7	41.67	39.17	5.745	.588
LLR	2	36.96	36.82	7.961	.954
	4	22.33	23.99	7.711	.866
	7	41.67	41.77	7.711	.956
MRR1	2	36.96	35.94	7.181	.902
	4	22.33	24.43	5.949	.692
	7	41.67	40.36	5.949	.866
MRR2	2	36.96	35.62	7.123	.872
	4	22.33	23.97	6.123	.806
	7	41.67	40.01	6.123	.794
PLR	2	36.96	35.76	7.324	.884
	4	22.33	23.68	6.580	.868
	7	41.67	40.30	6.580	.866

Table 8.C.2. (continued)

(m) [$n = 19$ $\gamma = .5$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	39.42	35.96	8.979	.722
	4	20.17	25.12	7.018	.146
	7	43.83	38.84	7.018	.162
LLR	2	39.42	38.64	8.513	.922
	4	20.17	22.00	8.410	.862
	7	43.83	43.28	8.410	.962
MRR1	2	39.42	38.35	8.323	.904
	4	20.17	22.34	7.989	.810
	7	43.83	42.80	7.989	.930
MRR2	2	39.42	37.86	8.107	.870
	4	20.17	21.59	8.063	.888
	7	43.83	42.41	8.063	.898
PLR	2	39.42	37.86	8.343	.876
	4	20.17	21.57	8.043	.890
	7	43.83	42.42	8.043	.910

(n) [$n = 19$ $\gamma = .75$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	41.89	36.71	11.214	.586
	4	18.00	25.45	8.765	.010
	7	46.00	38.52	8.765	.018
LLR	2	41.89	40.72	9.144	.908
	4	18.00	19.87	9.109	.866
	7	46.00	45.09	9.109	.948
MRR1	2	41.89	40.72	9.144	.908
	4	18.00	19.87	9.109	.866
	7	46.00	45.09	9.109	.948
MRR2	2	41.89	40.17	8.925	.878
	4	18.00	19.49	8.972	.894
	7	46.00	44.52	8.972	.910
PLR	2	41.89	40.16	9.151	.882
	4	18.00	19.48	8.956	.888
	7	46.00	44.52	8.956	.912

Table 8.C.2. (continued)

(o) [$n = 19$ $\gamma = 1$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	44.35	37.47	13.763	.508
	4	15.84	25.78	10.757	0
	7	48.16	38.19	10.757	0
LLR	2	44.35	42.95	9.735	.908
	4	15.84	17.72	9.724	.872
	7	48.16	47.04	9.724	.930
MRR1	2	44.35	42.95	9.735	.908
	4	15.84	17.72	9.724	.872
	7	48.16	47.04	9.724	.930
MRR2	2	44.35	42.52	9.593	.882
	4	15.84	17.40	9.648	.896
	7	48.16	46.61	9.648	.914
PLR	2	44.35	42.52	9.788	.886
	4	15.84	17.40	9.639	.898
	7	48.16	46.60	9.639	.916

the same for any x_0. Thus, the C.I. widths are identical for x_0= 4 and 7. However, since the true y-values and the (mean) fitted values differ for the different locations, the coverage probabilities may also differ, and still need to be considered.

Now for the interesting comparisons and conclusions. To begin with, consider the examples with n = 6. For these examples, the points x_0= 2, 4, and 7 are *not* at locations of actual data points (as they are for n = 10 and 19). There is a serious problem fitting these points when the misspecification (γ) increases. The problem originates with the nonparametric fitting technique, and is compounded by the fact that n = 6 provides few data points (little information) with large gaps between them. All of these factors together result in nonparametric fits that have rather large variances. This is due to the nonparametric technique placing almost all of the local weight used in obtaining fits on the actual data points being fit. Thus, different possible data sets from the same underlying model (with some error) will result in vastly different fits (especially if the errors are large). This local weighting problem also causes the trace of the nonparametric hat matrix to get larger (closer to n), and thus n-tr(**H**) gets closer to zero. Since n-tr(**H**) is the degrees of freedom for the t-value in the C.I.'s, this t-value may become extremely large. The large variances and large t-values cause wide confidence intervals. This phenomenon is clear in parts (a) and (b) of Table 8.C.2, where there is zero or small misspecification. In both of these cases, the model-robust procedures use entirely or mostly the parametric fit of OLS (PLR does this by choosing h_0= 1). The C.I. widths are seen to be small for OLS, MRR1, MRR2, and PLR, and the resulting coverage probabilities are very close to 95%. Comparing these values to LLR, one sees a tremendous difference. The local fitting problem mentioned above causes the C.I. widths for LLR to be much too large, and the resulting coverage probabilities are 100%. For moderate to large misspecification (γ = .5, .75, 1), the model-robust procedures use a little more LLR to achieve better fits. In doing so, the confidence intervals become much too wide, which is especially evident in PLR, which uses a complete LLR fit to its residuals. The OLS procedure maintains good coverage probabilities for larger γ's, but needs rather wide C.I.'s to overcome its poor fits.

These widths look fine compared to the other procedures, but even so, still need to be improved upon. More work is needed in this area to find appropriate confidence intervals for the nonparametric and model-robust techniques when the data available is sparse. Do note, though, that for small misspecification (part (b) of table), the model-robust procedures are doing as well as OLS. There is some evidence that the problems mentioned here are alleviated when moving to a moderate sample size. In other words, the problem here has more to do with n than it does with the x_0-values not being data points. Some support (albeit for a single data set) is given in Figure 6.D.1. Here the C.I.'s for all fitting techniques are shown for data from a model equivalent to the model used for the simulations, with $\gamma = .35$. Notice that even for the nonparametric LLR procedure, the confidence interval at *any* X-location has a reasonable width (somewhere between 10 and 15). That is, locations between data points show no sign of causing special problems. Thus, it appears sample size is of more importance than the location being estimated in terms this C.I. problem.

Parts (f)-(o) of Table 8.C.2 provide the C.I. diagnostics for $n = 10$ and $n = 19$. Clearly, with everything else held constant, the confidence interval widths drop dramatically from $n = 6$ to $n = 10$ and are also quite a bit lower for $n = 19$ than for $n = 10$. This result is expected since more information should provide for more precise estimates. Now consider the effects of varying γ, with other quantities held constant. For $\gamma = 0$ (no misspecification with the quadratic model), OLS and the model-robust procedures perform extremely well for any n, as seen in the small widths and accurate coverage probabilities of tables (f) and (k). LLR C.I.'s are still a little wide with slightly high coverage probabilities for $n = 10$, and for $n = 19$ are a little wide, but with coverages a little low. The first observation to be made as γ increases is that the OLS coverage probabilities become too small. For $n = 10$, this is apparent when $\gamma = .5$ (where the coverages are .850, .648, and .652 (in table (h)). For $n = 19$, the coverage probabilities drop to .860, .620, and .588 as early as $\gamma = .25$. For larger γ values, the OLS coverages are extremely poor, actually equaling 0 for $x_0 = 4$ and 7 when $n = 19$ and $\gamma = 1$. This is all

a reflection of the poor OLS fits, as seen in the mean $\hat{y}_o$'s compared to the true y's. The key part of these observations is the fact that OLS may perform poorly for even very small misspecification. These are precisely the cases where the user would probably be led to use OLS even after observing the data and possibly performing lack-of-fit tests. Recall that the model-robust procedures were first conceived as an attempt to overcome this problem. As seen in table (h) (n =10, γ = .5) and in table (l) (n = 19, γ = .25), the model-robust procedures consistently outperform OLS. This improvement is a little more obvious in this case for MRR2 and PLR over MRR1. A disappointing coverage probability appears for MRR1 when n = 19, γ = .25, and x_o= 4 (in table (l)). This 69% value is clearly the lowest of all model-robust values in Table 8.C.2, and results from a fit that uses a large portion of OLS (λ_o= .458) (other than this .69 value, MRR1 performs just as well as the other model-robust procedures). It is also seen in table (l) that for this scenario, LLR is slightly outperforming the model-robust procedures. The C.I. widths for LLR are a little larger, but not by a significant amount, and the coverage probabilities are a bit closer to 95%. Even though this C.I. advantage is present here, one sees from Table 8.B.2 that in terms of INTMSE, LLR is performing noticeably worse than the model-robust procedures.

The key observation that should be made here is that the model-robust procedures very rarely are worse in terms of confidence intervals than LLR, and the case pointed out above is one of these rare occurrences. The following observations lend support to this contention. For n = 10 and γ = .25 (table (g)), LLR has much wider C.I.'s that give coverage probabilities too large for x_o= 2 and 4. For the same case, the model-robust techniques are slightly low in coverage probability (mid 80's to low 90's), but use much narrower intervals. In moving to γ = .5 the model-robust C.I.'s remain narrower than those for LLR, and are much closer to 95% in coverage (whereas LLR remains too high). For the larger misspecifications, the model-robust C.I.'s get closer in form to the LLR intervals and all of these become a little too wide and result in coverage probabilities that are too high. This phenomenon is similar to what happened for n = 6, but not nearly as

serious. Still, improvements could be made in the confidence intervals in the future to try and resolve this problem. These high coverage probabilities are not present when $n = 19$. For γ in the range of .5 to 1, the model-robust procedures become similar to LLR, and have coverage probabilities consistently in the upper 80's to lower 90's. These coverages are slightly low, but with the small widths of the C.I.'s here, these C.I.'s are considered to be quite adequate. Also, do not forget that the points x_0= 2, 4, and 7 were specially picked in areas that should be difficult to fit, so achieving coverages around 90% for these values is actually a very good result.

In summary, the model-robust confidence intervals have been shown to outperform OLS in most situations, and to perform just as well as OLS even when there is no misspecification present. The exception here is for very small sample sizes ($n = 6$) and larger misspecification ($\gamma = .5, .75, 1$), where LLR and the model-robust procedures give extremely wide C.I.'s. The OLS procedure looks good in comparison, but it too gives wider intervals than desired. Possible improvements need to be researched in the future in order to provide adequate model-robust confidence intervals for these situations (where the model-robust procedures already greatly outperform OLS based on INTMSE, as seen in Table 8.B.2). It has also been shown above that except in rare situations, the model-robust C.I.'s tend to be better (narrower, with adequate coverage probabilities) than LLR, with LLR often giving intervals that are much too wide. Thus, the model-robust techniques, which have been established as having better fits based on INTMSE, also provide adequate confidence intervals across different sample size and misspecification combinations. References for possible improvements in these confidence intervals (say, for small sample sizes) are briefly outlined in section 6.D.

8.D Simulation Results for Data-Driven *h* and λ Selection

All simulation results presented so far have been based on the optimal values for bandwidths and mixing parameters. These results are considered the "best" of each fitting procedure, and show that the model-robust procedures (led by MRR2) have the ability to

significantly outperform the individual parametric and nonparametric methods. Addressed in this section is the effectiveness of data-driven methods in choosing h and λ such that these advantages are maintained. To determine the extent to which this is accomplished, the bandwidths and mixing parameters chosen by the data-driven methods are compared in value to h_o and λ_o. Also compared are the INTMSE values resulting from the data-driven fits to the INTMSE values for the optimal fits. Based on the preliminary bandwidth and λ selection study given in Chapter 7, the methods chosen to be analyzed here are PRESS* and PRESS**.

The setup for the simulations parallels that of the previous sections in this chapter. The same underlying function is used, with the same misspecification (γ) levels and sample sizes (n = 6, 10, 19). Five hundred Monte Carlo runs are executed for each scenario. The difference now is that h and λ are determined by data-driven methods, instead of being defined as the optimal h_o and λ_o. These h and λ values are averaged over the 500 runs to give the (mean) bandwidth and (mean) mixing parameter to be associated with each fitting technique. In addition to the direct comparisons of h and λ to h_o and λ_o, the INTMSE values resulting from using the (mean) h and λ are compared to the optimal INTMSE values.

8.D.1 Simulation Results for PRESS*

The first results given here are based on using PRESS* as the data-driven selection criterion. Recall that PRESS* is just the usual PRESS statistic, penalized for small bandwidths. As discussed in section 7.3, PRESS* (as a function of h) may often be minimized at $h = 1$ when selecting h for the MRR2 or PLR procedures. Examples of this behavior in PRESS* are shown in Figures 7.C.1 (c) and (d). It has also been pointed out that for cases such as these, the proper method of selecting h would be to obtain the graph of PRESS* vs. h and choose h where the curve starts leveling off (or at the first local minimum). Unfortunately, this is not practical for the 500 Monte Carlo runs, and the bandwidth chosen for each of these runs is the h chosen through a search routine to find

the value corresponding to the minimum PRESS*. The starting values for this search, as described in section 7.C, are chosen in such a way that the bandwidth corresponding to the first local minimum of PRESS* should be the one selected. This (usually) alleviates the problem represented by Figure 7.C.1 (c). However, often PRESS* still results in choosing a bandwidth of one, which results in poor fits for that particular procedure.

Table 8.D.1 contains the h's and λ's chosen for each fitting technique for each of the different simulation examples. Also shown for comparison are the optimal h_o and λ_o values (in bold). A final column that has been included for each model-robust procedure gives the number of times out of the 500 Monte Carlo runs that the bandwidth for the particular procedure was chosen to be 1 (labeled # h=1). The numbers in these "# h=1" columns provide the most obvious conclusion about PRESS*: the bandwidths for MRR2 and PLR are chosen much too often to be one, and the bandwidth is never chosen to be one for MRR1. This phenomenon has two main implications. The "good news" implication is that for the small misspecification cases ($\gamma = 0$ or sometimes .25) where the optimal bandwidths are large for MRR2 and PLR, PRESS* does an adequate job of selecting h. This is seen in any row of Table 8.D.1 with $\gamma = 0$. To measure, in terms of fitting performance, the closeness of h and λ chosen by PRESS* to the optimal h_o and λ_o, Table 8.D.2 displays the INTMSE for h and λ along with the optimal INTMSE. Note for $\gamma = 0$ that the model-robust fits based on PRESS* are close to optimal. (It appears that there are discrepancies in λ for these cases for MRR2: λ larger than λ_o; however, $h = 1$ gives a nearly constant linear fit through zero for the nonparametric residual fit, and *any* proportion of this added back to the parametric fit causes no real change to the parametric fit). The h and λ chosen by PRESS* for MRR1 are also adequate for $\gamma = 0$, because λ is chosen close to the small optimal value (close to zero). The bandwidth is chosen a little large for MRR1 (actually for LLR), but $\lambda = 0$ compensates for this in MRR1. LLR is not compensated, and the larger resulting INTMSE values are apparent in Table 8.D.2, especially for the ($n = 6$, $\gamma = 0$ or .25) cases. Even with this being the case for LLR,

Table 8.D.1. Bandwidths and mixing parameters chosen by PRESS* for the model-robust fitting procedures for 500 Monte Carlo simulations. Optimal h_o and λ_o are in bold. The column "# h=1" gives the number of times the bandwidth was chosen to be 1. (The h for MRR1 is also h for LLR).

		MRR1			MRR2			PLR	
n	γ	h	λ	# h=1	h	λ	# h=1	h	# h=1
6	0	.259 **.146**	.046 **.020**	*0*	.937 **1**	.515 **.016**	*456*	.781 **1**	*310*
	.25	.270 **.139**	.060 **.256**	*0*	.943 **1**	.518 **.016**	*460*	.801 **1**	*324*
	.5	.299 **.126**	.082 **.656**	*0*	.969 **.140**	.470 **.754**	*478*	.852 **.140**	*359*
	.75	.335 **.115**	.090 **.851**	*0*	.986 **.120**	.387 **.890**	*490*	.905 **.120**	*405*
	1	.369 **.108**	.098 **.924**	*0*	.996 **.110**	.304 **.939**	*497*	.944 **.110**	*444*
10	0	.185 **.130**	.075 **.013**	*0*	.898 **1**	.434 **.016**	*432*	.741 **1**	*279*
	.25	.188 **.122**	.068 **.301**	*0*	.898 **.226**	.459 **.478**	*433*	.768 **1**	*293*
	.5	.200 **.105**	.062 **.751**	*0*	.886 **.118**	.485 **.884**	*428*	.788 **.118**	*302*
	.75	.223 **.091**	.056 **.946**	*0*	.870 **.095**	.486 **.996**	*421*	.794 **.095**	*323*
	1	.261 **.082**	.051 **1**	*0*	.851 **.083**	.489 **1**	*412*	.807 **.083**	*349*
19	0	.145 **.113**	.065 **.009**	*0*	.886 **1**	.404 **.031**	*430*	.750 **1**	*294*
	.25	.140 **.104**	.161 **.458**	*0*	.722 **.157**	.477 **.739**	*333*	.606 **.158**	*217*
	.5	.125 **.089**	.452 **.892**	*0*	.432 **.099**	.693 **.996**	*170*	.368 **.099**	*99*
	.75	.110 **.077**	.756 **1**	*0*	.185 **.080**	.866 **1**	*39*	.173 **.080**	*20*
	1	.098 **.068**	.910 **1**	*0*	.117 **.069**	.944 **1**	*9*	.114 **.070**	*3*

Table 8.D.2. INTMSE values for fits based on PRESS* from 500 Monte Carlo runs. Optimal INTMSE values are in bold.

n	γ	OLS	LLR	MRR1	MRR2	PLR
6	0	6.384	42.170	6.403	6.387	6.398
		6.384	**11.789**	**6.382**	**6.384**	**6.391**
	.25	9.295	45.198	9.336	9.297	9.306
		9.295	**12.695**	**8.985**	**9.295**	**9.301**
	.5	18.028	58.289	18.136	18.027	18.031
		18.028	**15.220**	**14.143**	**13.596**	**13.659**
	.75	32.583	78.839	32.730	32.579	32.576
		32.583	**19.190**	**19.293**	**18.229**	**17.907**
	1	52.959	105.466	53.169	52.952	52.941
		52.959	**24.630**	**25.127**	**23.915**	**23.408**
10	0	4.105	7.926	4.117	4.106	4.119
		4.105	**7.689**	**4.104**	**4.105**	**4.110**
	.25	6.818	9.535	6.688	6.820	6.826
		6.818	**8.243**	**6.300**	**6.490**	**6.825**
	.5	14.956	14.427	14.530	14.960	14.961
		14.956	**9.456**	**9.262**	**8.884**	**8.867**
	.75	28.520	25.556	27.860	28.527	28.523
		28.520	**10.721**	**10.819**	**10.403**	**10.450**
	1	47.509	61.576	47.116	47.517	47.514
		47.509	**11.883**	**11.883**	**11.675**	**11.722**
19	0	2.314	5.589	2.322	2.315	2.737
		2.314	**4.622**	**2.314**	**2.314**	**2.316**
	.25	4.973	5.557	4.609	4.969	5.917
		4.973	**4.971**	**4.104**	**4.041**	**4.080**
	.5	12.951	6.348	8.390	12.542	12.738
		12.951	**5.695**	**5.680**	**5.348**	**5.381**
	.75	26.247	7.230	9.331	15.558	13.287
		26.247	**6.430**	**6.430**	**6.251**	**6.283**
	1	44.861	8.067	9.261	12.234	10.881
		44.861	**7.089**	**7.089**	**6.979**	**7.010**

PRESS* does perform well for the model-robust procedures when the misspecification present is low ($\gamma = 0$ and sometimes $\gamma = .25$).

The negative implication of the tendency of PRESS* to result in h's = 1 is that for cases of misspecification in the model, where a small optimal bandwidth is desired, the bandwidths chosen by PRESS* are much too large for MRR2 and PLR. This discrepancy is clear from any case in Table 8.D.1 when h_o is not one. The INTMSE's in Table 8.D.2 resulting from these large h's are greatly inflated, and the benefits of using the model-robust procedures are lost. For example, notice the drastic loss in performance for the case where $n = 10$ and $\gamma = .5$. Thus, MRR2 and PLR are obviously hampered by using these selected h's from PRESS*. In observing the results for MRR1, one sees that this procedure is adversely affected also. The h's are chosen consistently large, but the main problem is with the selection of extremely small λ's (until $\gamma = .5$ for $n = 19$). Recall that PRESS* is penalizing for variance, and this characteristic affects MRR1 by choosing a small λ which prevents much of the (more variable) LLR fit from being used. The consistently high bandwidths also cause increases in the INTMSE of LLR, as seen in Table 8.D.2.

For n=19, the model-robust procedures perform a little better, but still suffer from bandwidths that are too large (still have h's =1 for MRR2 and PLR). The model-robust procedures are now quite a bit better than OLS, but still are not as good as LLR. MRR1 shows the most improvement for the model-robust procedures, but still needs to be improved upon even more.

Thus, the conclusions reached from these observations is that the h's and λ's chosen through the minimization of PRESS* are very inadequate, except for the case of no misspecification. So if using PRESS*, one would definitely need to use the technique of graphing PRESS* as a function of h and selecting the proper bandwidth according to the slope of this plot. However, it is conjectured here that the h's and λ's chosen by PRESS* would still be inadequate. This statement is supported by the MRR1 results given above (h's too large even without h's equal 1, and λ's much too small), and by the

preliminary "single data set" examples in Chapter 7. A possible alternative to PRESS* is PRESS**, which is studied now to determine if it provides sufficient improvements.

8.D.2 Simulation Results for PRESS**

Bandwidth, λ, INTMSE

Recall that PRESS** (defined in equation 3.B.22) is designed to control (reduce) the size of h chosen by PRESS*. This is accomplished by a penalty term for large h being added to the denominator of PRESS*. It is hoped that PRESS** would prevent selection of $h = 1$ when a smaller bandwidth is desired. Tables 8.D.3 and 8.D.4 supply the results of using PRESS** as the data-driven selection criterion for the simulation examples being studied. Table 8.D.3 contains the values of the selected h's and λ's (with optimal values in bold), along with the number of h's chosen to be one. Table 8.D.4 contains the INTMSE's for the chosen bandwidths and mixing parameters, and the INTMSE's for the optimal fits (in bold). These values can be used to measure how "close" the chosen h's and λ's are to h_o and λ_o in terms of fitting performance.

The first observance from Table 8.D.3 is the major reduction (from PRESS*) in the number of bandwidths chosen to be one for MRR2 and PLR. Except for the cases (n=19, $\gamma = 0$ or .25) the largest number of bandwidths chosen to be one (out of the 500 runs) for any particular case is 23 (which is only 4.6% of the time). This means that in practice, one can be very confident that h will not be chosen as 1 when using PRESS**. Unfortunately, there is one problem with this tendency to shy away from $h = 1$. For $\gamma = 0$ (no misspecification), the optimal bandwidth *should* be one for MRR2 and PLR (h_o should also be one in a few cases where $\gamma = .25$). As seen in Table 8.D.3, the chosen h values are far below one for these cases. This would not be a problem at all, though, if λ were chosen to be close to zero for these cases (giving OLS). Unfortunately, this does not happen, and λ is actually chosen rather large for the small γ cases (seen in Table 8.D.3). These large λ values result from PRESS** penalizing for bias, and consequently desiring

Table 8.D.3. Bandwidths and mixing parameters chosen by PRESS** for the model-robust fitting procedures for 500 Monte Carlo simulations. Optimal h_o and λ_o are in bold. The column "# h=1" gives the number of times the bandwidth was chosen to be 1. (The h for MRR1 is also h for LLR).

		MRR1			MRR2			PLR	
n	γ	h	λ	# h=1	h	λ	# h=1	h	# h=1
6	0	.191 **.146**	.647 **.020**	*0*	.145 **1**	.863 **.016**	*0*	.118 **1**	*0*
	.25	.201 **.139**	.625 **.256**	*0*	.140 **1**	.824 **.016**	*0*	.127 **1**	*0*
	.5	.229 **.126**	.590 **.656**	*0*	.132 **.140**	.747 **.754**	*0*	.126 **.140**	*0*
	.75	.270 **.115**	.566 **.851**	*0*	.127 **.120**	.692 **.890**	*0*	.125 **.120**	*0*
	1	.309 **.108**	.440 **.924**	*0*	.125 **.110**	.661 **.939**	*0*	.123 **.110**	*0*
10	0	.156 **.130**	.378 **.013**	*0*	.138 **1**	.755 **.016**	*14*	.123 **1**	*5*
	.25	.154 **.122**	.432 **.301**	*0*	.141 **.226**	.784 **.478**	*18*	.131 **1**	*8*
	.5	.149 **.105**	.555 **.751**	*0*	.133 **.118**	.844 **.884**	*14*	.139 **.118**	*15*
	.75	.143 **.091**	.665 **.946**	*0*	.121 **.095**	.878 **.996**	*10*	.139 **.095**	*16*
	1	.136 **.082**	.742 **1**	*0*	.109 **.083**	.899 **1**	*5*	.116 **.083**	*5*
19	0	.128 **.113**	.338 **.009**	*0*	.486 **1**	.670 **.031**	*205*	.376 **1**	*110*
	.25	.122 **.104**	.497 **.458**	*0*	.338 **.157**	.765 **.739**	*123*	.268 **.158**	*63*
	.5	.108 **.089**	.782 **.892**	*0*	.144 **.099**	.908 **.996**	*23*	.123 **.099**	*5*
	.75	.093 **.077**	.928 **1**	*0*	.089 **.080**	.954 **1**	*0*	.089 **.080**	*0*
	1	.082 **.068**	.972 **1**	*0*	.078 **.069**	.976 **1**	*0*	.079 **.070**	*0*

Table 8.D.4. INTMSE values for fits based on PRESS** from 500 Monte Carlo runs. Optimal INTMSE values are in bold.

n	γ	OLS	LLR	MRR1	MRR2	PLR
6	0	6.384	16.448	10.477	8.597	10.453
		6.384	**11.789**	**6.382**	**6.384**	**6.391**
	.25	9.295	18.814	12.656	9.778	11.029
		9.295	**12.695**	**8.985**	**9.295**	**9.301**
	.5	18.028	28.350	20.728	13.439	13.669
		18.028	**15.220**	**14.143**	**13.596**	**13.659**
	.75	32.583	50.968	37.320	20.055	18.015
		32.583	**19.190**	**19.293**	**18.229**	**17.907**
	1	52.959	80.155	57.096	29.787	23.969
		52.959	**24.630**	**25.127**	**23.915**	**23.408**
10	0	4.105	7.826	4.621	5.197	6.592
		4.105	**7.689**	**4.104**	**4.105**	**4.110**
	.25	6.818	8.234	6.321	6.358	7.027
		6.818	**8.243**	**6.300**	**6.490**	**6.825**
	.5	14.956	10.048	10.529	9.321	9.275
		14.956	**9.456**	**9.262**	**8.884**	**8.867**
	.75	28.520	12.733	15.498	12.569	13.251
		28.520	**10.721**	**10.819**	**10.403**	**10.450**
	1	47.509	16.024	20.672	15.469	14.854
		47.509	**11.883**	**11.883**	**11.675**	**11.722**
19	0	2.314	5.079	2.614	2.322	2.756
		2.314	**4.622**	**2.314**	**2.314**	**2.316**
	.25	4.973	5.078	4.190	4.750	5.095
		4.973	**4.971**	**4.104**	**4.041**	**4.080**
	.5	12.951	5.691	5.839	6.724	5.856
		12.951	**5.695**	**5.680**	**5.348**	**5.381**
	.75	26.247	6.435	6.590	6.675	6.480
		26.247	**6.430**	**6.430**	**6.251**	**6.283**
	1	44.861	7.089	7.222	7.441	7.278
		44.861	**7.089**	**7.089**	**6.979**	**7.010**

more of the nonparametric fit to be used. This problem with λ (as with the problem of λ too small from PRESS*) is not really addressed in the current work, and is left for possible future research. This is done for two reasons. First, most of the λ values chosen by PRESS** are indeed adequate (for larger n and some misspecification present). Secondly, if the bandwidth could be improved upon (made closer to 1) (which may effect λ at the same time), the whole problem could be eliminated and it would not matter what λ was chosen to be. Thus, the discussion here centers on the bandwidth. For the large sample size case (n = 19), PRESS** does result in (mean) bandwidths of .486 and .376 (at γ = 0). These h's are large enough to give close to optimal fits, as seen in the INTMSE values of Table 8.D.4 for (n = 19, γ = 0). For the (n = 19, γ = .25) case, the (mean) bandwidths remain large for MRR2 and PLR (with "# h=1" of 123 and 63, respectively), but they remain a little *too* large. However, the optimal bandwidths are somewhat large also (relative to higher γ cases), and the resulting fits from PRESS** are not too far from optimal. The MRR2 INTMSE is smaller than those for OLS or LLR, thus maintaining the beneficial properties of this model-robust procedure. Thus, for larger sample sizes it appears that the model-robust procedures are not overly impacted by these discrepancies in bandwidth selection. But what about small to moderate sample sizes? For the cases of n = 6 and 10 where h_o= 1 for MRR2 and PLR, the chosen bandwidths are .145, .140, .118, .127, .138, .123, and .131, as seen in various locations in Table 8.D.3. This appears to be a huge misspecification that would result in fits far from optimal. However, as seen in Table 8.D.4 for the cases (n = 6, γ = 0 or .25) and (n = 10, γ = 0 or .25 for PLR), the fits are actually not greatly different from optimal. To get an idea of a fit that would be considered "greatly different" from optimal, one only needs to look at Table 8.D.2 (the PRESS* INTMSE table). Fits such as those for LLR, MRR1, MRR2, and PLR for the case (n = 10, γ = .75) (or even (n = 10, γ = .5)) are what is meant by this expression. Differences in INTMSE's like "28.527 vs. 10.403" represent extremely poor fits, and differences like "14.960 vs. 8.884" also show significant problems. (These numbers are from MRR2 for the cases (n = 10, γ = .75) and (n = 10, γ = .5), respectively). Now, for

the PRESS** results, the largest difference in INTMSE's for MRR2 or PLR is for PLR in the case ($n = 6$, $\gamma = 0$). Here the difference in values is 10.453 vs. the optimal 6.391. This is the *only* location in the entire Table 8.D.4 where one of the MRR2 or PLR fits (based on PRESS**) might be considered to be rather poor, but still it probably should not be considered "greatly different" from optimal. (The MRR2 fit for ($n = 6$, $\gamma = 1$) is a bit away from its optimum, but it still strongly outperforms either OLS or LLR). The MRR2 for the ($n = 6$, $\gamma = 0$) case is actually not too bad at all and is much better than LLR. In this situation, MRR2 uses a significant portion of the LLR residual fit, but the underlying adequate OLS fit keeps the INTMSE at a somewhat low value. This is a case where fitting the residuals (containing less structure), rather than the data, with the nonparametric fit is beneficial. This is illustrated by the difference in INTMSE's of MRR2 (8.597) compared to MRR1 (10.477). For all cases shown in Table 8.D.4, other than (n=6, γ=0), the INTMSE values for the fits based on PRESS** remain relatively close to optimal.

The following discussion provides the key points to be made about the fits resulting from using PRESS** as the selector of h and λ. First, nothing as of yet has been said about the performance of MRR1. While the bandwidths are now somewhat smaller than they were for PRESS* (which were quite a bit too large), they are still consistently a little large. Also, the λ's chosen for MRR1 are somewhat erratic, especially for smaller sample sizes. These characteristics result in many of the MRR1 fits being quite poor when based on PRESS**, as seen in the INTMSE values of Table 8.D.4. MRR1 does perform very well for $n = 19$, but the problems for smaller n damage the reliability of MRR1 in terms of performing well in a general setting. However, the most important point to be made here is about the good, consistent performance of MRR2 and PLR when based on PRESS**. Notice from Table 8.D.4 that for the no misspecification examples, MRR2 and PLR have INTMSE values a little larger than OLS, but they are not far off (the possible exception being PLR for ($n = 6$, $\gamma = 0$ discussed above), and are much better than LLR. Also, for large misspecification examples, the MRR2 and PLR INTMSE's are just a little

larger than those for LLR for the cases (n = 19 and γ = .5, .75, or 1), and are lower everywhere else. Also, the INTMSE's are much lower than those for OLS, which fits poorly. For small to moderate misspecification, MRR2 and PLR often give results much better than either OLS or LLR, thus establishing the advantages of these model-robust procedures over the individual parametric and nonparametric procedures. In conclusion then, if the user does not know how much model misspecification may be present in a certain situation, and wants to protect against all possibilities, then using MRR2 (or PLR) based on h and λ from PRESS** is the appropriate method to use. This would provide adequate fits at either extreme of misspecification, and would perform better than either OLS or LLR for anything in between. If OLS were used, then large problems could arise if there happens to be misspecifications present. A similar statement applies for using LLR, where much performance is lost if there happens to be no misspecification present. These considerations are precisely why model-robust methods have been developed in this work, and the previous discussion above shows that PRESS** can make the methods work in practice.

Confidence Intervals

The final concern as to the actual effectiveness of using PRESS** as the selection criterion is whether or not adequate confidence intervals can be obtained for the various fitting techniques. Of main interest at this point are the performances of MRR2 and PLR, which have been shown above to have the best potential for providing adequate fits based on PRESS**. Table 8.D.5 contains the 95% confidence interval diagnostics for the simulations being studied, where the fits of the various procedures are based on h and λ chosen by PRESS**. The information in Table 8.D.5 parallels that of Table 8.C.2 for the optimal fits. Namely, the three x_0 locations 2, 4, and 7 are selected to be studied since they are located at points where there is much change in the underlying curve as γ is varied. The diagnostics reported at each of these x_0's are the true y-value ($E(y_0)$), the (mean) fitted value (across the 500 Monte Carlo runs), the (mean) C.I. width, and

Table 8.D.5 (a)-(o). Confidence interval diagnostics for the various fits based on PRESS** for the 500 Monte Carlo runs.

(a) [$n = 6 \quad \gamma = 0$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	34.5	34.55	15.086	.938
	4	24.5	24.42	13.833	.942
	7	39.5	39.31	13.833	.948
LLR	2	34.5	36.78	39.290	1
	4	24.5	27.30	37.313	1
	7	39.5	42.38	37.313	1
MRR1	2	34.5	36.00	27.210	.990
	4	24.5	26.22	25.437	.986
	7	39.5	41.27	25.437	.996
MRR2	2	34.5	34.51	29.819	.980
	4	24.5	24.32	30.929	.988
	7	39.5	39.41	30.929	.992
PLR	2	34.5	34.48	9.64×10^{23}	.996
	4	24.5	24.29	1.02×10^{24}	.990
	7	39.5	39.39	1.02×10^{24}	.996

(b) [$n = 6 \quad \gamma = .25$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	36.96	34.77	17.432	.932
	4	22.33	24.52	15.984	.944
	7	41.67	39.21	15.984	.920
LLR	2	36.96	37.32	40.720	1
	4	22.33	27.19	38.030	1
	7	41.67	43.07	38.030	1
MRR1	2	36.96	36.35	28.734	.996
	4	22.33	26.13	26.499	.982
	7	41.67	41.56	26.499	.998
MRR2	2	36.96	35.14	35.901	.988
	4	22.33	23.55	37.081	.986
	7	41.67	40.19	37.081	.980
PLR	2	36.96	35.27	3.77×10^{23}	.996
	4	22.33	23.28	3.99×10^{23}	.992
	7	41.67	40.43	3.99×10^{23}	.990

Table 8.D.5. (continued)

(c) [$n = 6$ $\gamma = .5$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	39.42	35.00	23.280	.938
	4	20.17	24.61	21.347	.936
	7	43.83	39.11	21.347	.916
LLR	2	39.42	37.85	44.462	1
	4	20.17	27.91	39.592	1
	7	43.83	44.10	39.592	1
MRR1	2	39.42	36.66	33.445	.988
	4	20.17	26.58	29.725	.978
	7	43.83	42.03	29.725	.992
MRR2	2	39.42	35.74	46.033	.990
	4	20.17	22.88	46.989	.992
	7	43.83	40.86	46.989	.990
PLR	2	39.42	36.07	1.36×10^{23}	1
	4	20.17	22.24	1.44×10^{23}	.998
	7	43.83	41.48	1.44×10^{23}	.998

(d) [$n = 6$ $\gamma = .75$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	41.89	35.23	30.866	.956
	4	18.00	24.71	28.303	.956
	7	46.00	39.02	28.303	.942
LLR	2	41.89	38.26	49.799	1
	4	18.00	29.36	41.623	.996
	7	46.00	45.24	41.623	1
MRR1	2	41.89	36.95	40.221	.986
	4	18.00	27.45	34.494	.970
	7	46.00	42.47	34.494	.988
MRR2	2	41.89	36.33	57.829	.998
	4	18.00	22.29	58.487	.994
	7	46.00	41.45	58.487	.996
PLR	2	41.89	36.87	1.92×10^{22}	1
	4	18.00	21.18	2.03×10^{22}	1
	7	46.00	42.54	2.03×10^{22}	1

Table 8.D.5. (continued)

(e) [$n = 6 \quad \gamma = 1$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	44.35	35.45	39.198	.978
	4	15.84	24.81	35.944	.972
	7	48.16	38.92	35.944	.976
LLR	2	44.35	38.52	56.505	1
	4	15.84	30.87	44.894	.980
	7	48.16	46.24	44.894	1
MRR1	2	44.35	36.87	46.425	.992
	4	15.84	27.48	39.776	.976
	7	48.16	42.07	39.776	.992
MRR2	2	44.35	36.90	67.659	1
	4	15.84	21.73	67.794	.998
	7	48.16	42.01	67.794	1
PLR	2	44.35	37.68	2111.801	1
	4	15.84	20.12	2222.729	1
	7	48.16	43.61	2222.729	1

Table 8.D.5. (continued)

(f) [$n = 10$ $\gamma = 0$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	34.5	34.36	9.661	.950
	4	24.5	24.43	8.089	.954
	7	39.5	39.48	8.089	.950
LLR	2	34.5	35.76	12.593	.986
	4	24.5	26.51	11.462	.946
	7	39.5	41.54	11.462	.934
MRR1	2	34.5	34.88	9.775	.940
	4	24.5	25.06	8.435	.946
	7	39.5	40.06	8.435	.920
MRR2	2	34.5	34.47	10.303	.942
	4	24.5	24.42	9.721	.938
	7	39.5	39.53	9.721	.952
PLR	2	34.5	34.59	2.81×10^{22}	.964
	4	24.5	24.43	2.81×10^{22}	.950
	7	39.5	39.62	2.81×10^{22}	.956

(g) [$n = 10$ $\gamma = .25$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	36.96	34.93	10.745	.906
	4	22.33	24.68	8.997	.854
	7	41.67	39.24	8.997	.820
LLR	2	36.96	36.98	13.303	.986
	4	22.33	25.51	12.167	.846
	7	41.67	42.43	12.167	.984
MRR1	2	36.96	35.88	10.791	.942
	4	22.33	24.65	9.394	.840
	7	41.67	40.58	9.394	.906
MRR2	2	36.96	35.98	11.424	.920
	4	22.33	23.38	10.886	.916
	7	41.67	40.58	10.886	.914
PLR	2	36.96	36.28	4.16×10^{20}	.954
	4	22.33	23.12	4.16×10^{20}	.958
	7	41.67	40.95	4.16×10^{20}	.956

Table 8.D.5. (continued)

(h) [$n = 10$ $\gamma = .5$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	39.42	35.49	13.535	.850
	4	20.17	24.92	11.333	.648
	7	43.83	38.99	11.333	.652
LLR	2	39.42	38.25	15.211	.987
	4	20.17	24.35	14.060	.796
	7	43.83	43.38	14.060	.984
MRR1	2	39.42	37.14	13.364	.926
	4	20.17	23.98	11.911	.744
	7	43.83	41.46	11.911	.864
MRR2	2	39.42	37.56	14.039	.932
	4	20.17	22.07	13.619	.900
	7	43.83	41.90	13.619	.910
PLR	2	39.42	37.82	17.037	.962
	4	20.17	21.72	16.519	.934
	7	43.83	42.36	16.519	.938

(i) [$n = 10$ $\gamma = .75$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	41.89	36.06	17.253	.852
	4	18.00	25.16	14.446	.488
	7	46.00	38.75	14.446	.480
LLR	2	41.89	39.63	17.914	.982
	4	18.00	23.01	16.753	.800
	7	46.00	44.47	16.753	.986
MRR1	2	41.89	38.59	16.644	.942
	4	18.00	22.96	15.189	.740
	7	46.00	42.64	15.189	.876
MRR2	2	41.89	39.21	17.270	.958
	4	18.00	20.59	16.989	.938
	7	46.00	43.40	16.989	.934
PLR	2	41.89	39.40	18.884	.970
	4	18.00	20.26	18.307	.942
	7	46.00	43.76	18.307	.934

Table 8.D.5. (continued)

(j) [$n = 10$ $\gamma = 1$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	44.35	36.63	21.423	.868
	4	15.84	25.41	17.938	.386
	7	48.16	38.51	17.938	.396
LLR	2	44.35	41.14	21.041	.986
	4	15.84	21.46	19.946	.838
	7	48.16	45.72	19.946	.992
MRR1	2	44.35	40.18	20.186	.968
	4	15.84	21.64	18.791	.794
	7	48.16	43.99	18.791	.888
MRR2	2	44.35	40.87	20.778	.978
	4	15.84	19.12	20.574	.964
	7	48.16	44.86	20.574	.956
PLR	2	44.35	41.13	21.932	.984
	4	15.84	18.70	21.342	.972
	7	48.16	45.31	21.342	.962

Table 8.D.5. (continued)

(k) [$n = 19$ $\gamma = 0$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	34.5	34.44	6.754	.958
	4	24.5	24.47	5.279	.958
	7	39.5	39.49	5.279	.958
LLR	2	34.5	35.27	7.739	.930
	4	24.5	25.89	7.067	.862
	7	39.5	40.88	7.067	.868
MRR1	2	34.5	34.66	6.734	.944
	4	24.5	24.82	5.538	.914
	7	39.5	39.82	5.538	.926
MRR2	2	34.5	34.39	6.803	.920
	4	24.5	24.47	5.839	.924
	7	39.5	39.47	5.839	.936
PLR	2	34.5	34.37	2.51×10^{22}	.918
	4	24.5	24.46	2.51×10^{22}	.922
	7	39.5	39.45	2.51×10^{22}	.930

(l) [$n = 19$ $\gamma = .25$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	36.96	35.20	7.350	.860
	4	22.33	24.80	5.745	.620
	7	41.67	39.17	5.745	.588
LLR	2	36.96	36.87	7.906	.948
	4	22.33	24.55	7.305	.720
	7	41.67	42.02	7.305	.954
MRR1	2	36.96	36.12	7.182	.904
	4	22.33	24.29	6.101	.692
	7	41.67	40.66	6.101	.808
MRR2	2	36.96	35.88	7.174	.866
	4	22.33	23.62	6.386	.784
	7	41.67	40.34	6.386	.770
PLR	2	36.96	36.00	1.19×10^{21}	.878
	4	22.33	23.41	1.19×10^{21}	.810
	7	41.67	40.54	1.19×10^{21}	.800

Table 8.D.5. (continued)

(m) [$n = 19$ $\gamma = .5$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	39.42	35.96	8.979	.722
	4	20.17	25.12	7.018	.146
	7	43.83	38.84	7.018	.162
LLR	2	39.42	38.58	8.351	.918
	4	20.17	22.73	7.931	.682
	7	43.83	43.27	7.931	.944
MRR1	2	39.42	38.12	8.098	.882
	4	20.17	22.94	7.359	.588
	7	43.83	42.43	7.359	.784
MRR2	2	39.42	37.77	7.961	.826
	4	20.17	22.00	7.542	.748
	7	43.83	42.00	7.542	.758
PLR	2	39.42	37.92	2.95×10^{20}	.850
	4	20.17	21.74	2.95×10^{20}	.818
	7	43.83	42.26	2.95×10^{20}	.818

(n) [$n = 19$ $\gamma = .75$]

Method	x_o	True y	Mean $\hat{y}_o$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	41.89	36.71	11.214	.586
	4	18.00	25.45	8.765	.010
	7	46.00	38.52	8.765	.018
LLR	2	41.89	40.51	8.924	.892
	4	18.00	20.65	8.677	.722
	7	46.00	44.85	8.677	.934
MRR1	2	41.89	40.28	8.905	.884
	4	18.00	20.88	8.504	.670
	7	46.00	44.45	8.504	.870
MRR2	2	41.89	39.90	8.716	.834
	4	18.00	20.01	8.544	.804
	7	46.00	44.00	8.544	.812
PLR	2	41.89	40.02	8.909	.854
	4	18.00	19.79	8.620	.836
	7	46.00	44.21	8.620	.854

Table 8.D.5. (continued)

(o) [$n = 19$ $\gamma = 1$]

Method	x_0	True y	Mean $\hat{y}_0$	Mean C.I. Width	Obs. Covrg. Probability
OLS	2	44.35	37.47	13.763	.508
	4	15.84	25.78	10.757	0
	7	48.16	38.19	10.757	0
LLR	2	44.35	42.61	9.529	.882
	4	15.84	18.48	9.394	.766
	7	48.16	46.67	9.394	.920
MRR1	2	44.35	42.49	9.566	.872
	4	15.84	18.64	9.356	.750
	7	48.16	46.46	9.356	.882
MRR2	2	44.35	42.15	9.407	.840
	4	15.84	17.95	9.348	.836
	7	48.16	46.07	9.348	.840
PLR	2	44.35	42.23	9.573	.856
	4	15.84	17.79	9.348	.866
	7	48.16	46.22	9.348	.866

observed coverage probabilities of these C.I.'s. Several conclusions can be derived from this table and are described below.

First, note that often the (mean) C.I. width for PLR is reported as an extremely large number (9.64×10^{23}, for example). These values should not be interpreted to mean that PLR will usually result in extremely wide C.I.'s. The actual cause of these large values is the tendency on rare occasions for PLR (based on PRESS**) to select a very small bandwidth (such as .035 or .05). This results in small (close to zero) degrees of freedom ($n - \text{tr}(\mathbf{H})$) for the t-value in the confidence interval, which results in a huge t-value, which leads to wide C.I.'s. As mentioned above, this is a rare occurrence, but even getting just one of these values out of the 500 simulations would result in a large (mean) width. Thus, the large C.I. width values for PLR in Table 8.D.5 are really misleading, and one should keep in mind that most of the 500 individual widths are not so large. In several cases (tables (h), (i), (j), (n), (o)), no extremely low bandwidths were chosen for PLR, and the (mean) width values are accurate. Whether or not the (mean) C.I. width measurements are accurate, the observed coverage probability values can be interpreted as being accurate. This is because the individual huge C.I. width problem only occurs in rare data sets, and the coverage probabilities are barely affected, if at all.

For comparisons of the fitting techniques, consider first the small sample cases ($n = 6$), where the x_0-values are not actual data points. Recall for the optimal fits (data in Table 8.C.2), the C.I.'s became much too wide for $\gamma \geq .5$, and coverage probabilities (except for OLS) were around .99 or larger. For OLS, these coverages were much more adequate, but the C.I. width was still wider than desired. For $\gamma = 0$ or .25 (for $n = 6$), the robust procedures performed very well by selecting fits close to OLS (gave small widths and accurate coverages). Similar results (for $n = 6$) hold for using PRESS** to choose h and λ, except that the C.I.'s are too wide even for small γ values. For these small γ, the coverage probabilities are around .99 on average, compared to .94 to .95 for the optimal fits. This drop in accuracy is due to the model-robust procedures based on PRESS** selecting more of the LLR fit than the OLS fit, as was pointed out in the previous

subsection. This results in much larger variances, and thus the wider C.I.'s. The conclusion here (for small sample sizes) is that, while the fits are fine, improved methods for confidence intervals are needed. This was also the conclusion from the optimal fits.

Now consider the cases of moderate to larger sample sizes (n = 10, 19). For these cases, the performance of PRESS** is very adequate and the results look promising for this selection criterion. In fact, for n = 10 the fits from using PRESS** often provide confidence intervals that have even better properties than the optimal fits. This is seen in the (g) tables of Tables 8.C.2 and 8.D.5 for γ = .25, where MRR2 gives slightly higher coverage probabilities that are closer to .95 and PLR gives much more accurate coverages ($\approx$ .95 compared to coverages ranging from .82 to .90 for optimal fits). The greatest improvement, though, can be found in the (i) and (j) tables for large γ values. For these cases, PRESS** yields C.I.'s that are significantly narrower and have much better coverage probabilities, especially for the model-robust procedures. For example, the coverage probabilities at γ = 1 for the model-robust procedures based on optimal fits were all larger than .99. These values range from .956 to .984 for MRR2 and PLR from fits based on PRESS**. The C.I. widths drop from around 26.5 on average (for optimal fits) to around 21.0 on average (for PRESS** fits). MRR1 confidence intervals are not consistently as good as those of MRR2 and PLR. These MRR1 intervals from PRESS** still show improved behavior over those from optimal fits, but tend to often have somewhat low coverage probabilities at x_0 = 4 and 7. This property of MRR1, coupled with the problem of PLR on occasion selecting a bandwidth too small (resulting in wide C.I.'s), leads to the conclusion that MRR2 based on PRESS** as the selection criterion seems to be the most promising model-robust technique to be used in practice (at least based on n = 10).

For n = 19, the C.I. coverage probabilities for the fitting technique based on PRESS** are always a little lower than those based on the optimal h and λ values, and thus a little further from .95. This is illustrated in any of tables (k)-(o) of Tables 8.C.2 and 8.D.5. It is important to observe, however, that all of the coverage probabilities are still

very close to those from the optimal fits, along with C.I. widths being very close (slightly wider for $\gamma = 0, .25$, and noticeably narrower for $\gamma = .5, .75, 1$). Observing the PRESS**-based MRR2 results more closely (since this is the most advantageous technique thus far), one sees that most of the coverage probabilities range from the upper 70%'s to the lower mid 90%'s. These values are only slightly below the values for the optimal fits for MRR2, and still provide "acceptable" coverages, especially considering again that the x_o-values were chosen at locations difficult to fit. The two coverage probabilities of .748 and .758 in the case $\gamma = .5$ (for $n = 19$ for PRESS**) may be considered "undesirably" low, but they are not unacceptable (as are values such as .146 and .162 for OLS). With these being the lowest coverages for MRR2, it appears that MRR2 based on PRESS** also performs well enough for $n = 19$ to be useful in practice. This is an important statement because it has been shown in prior discussions that using PRESS** (in particular, for MRR2) maintains the benefits of the fits of the model-robust procedures over the individual OLS and LLR fits. These benefits are also apparent here. Clearly from tables (l)-(o), as the misspecification increases, the OLS coverage probabilities become extremely low (approaching zero). The model-robust procedures avoid this problem and hold a distinct advantage over OLS. The benefits over LLR are not as clear for this case of $n = 19$, where LLR provides mostly adequate results. In fact, LLR only shows one slight problem: when fitting to $x_o = 4$, the coverage probabilities are sometimes rather low (.720, .682, .722). The model-robust procedures improve upon this situation by giving consistently higher coverage probabilities at this location. For the scenarios involving $x_o = 2$ or 7, LLR often provides better coverage probabilities than MRR2. However, the MRR2 confidence intervals are always narrower than those for LLR, even when the coverage probabilities are higher (and more accurate) for MRR2. Thus, the choice between LLR and MRR2 is rather difficult for $n = 19$, but it does appear that MRR2 would be slightly more reliable, all cases considered. The smaller sample sizes are what really hinder LLR as a general technique to be used in practice. This and other conclusions from the simulation study presented here are summarized in the next section.

8.E Conclusions

Based on the results of the simulations presented above, several general conclusions can be made about the model-robust procedures developed in this work. First, the model-robust procedures (MRR1, MRR2, and PLR) all have the ability to outperform the individual procedures of OLS and LLR. This is supported by the simulation results where the optimal fits were obtained. Namely, in section 8.C when the optimal h and λ (those that minimize AVEMSE) were used to obtain the various fits, the INTMSE values for the model-robust procedures were lower than (or $\approx$ equal to) those for OLS and LLR. With no misspecification present in the chosen model, the model-robust procedures performed as well or better than the ordinarily used OLS procedure. At the other extreme, when the model was greatly misspecified, the model-robust procedures performed as well as (or better than) the ordinarily used LLR procedure. In small to moderate cases of misspecification, the model-robust procedures were consistently better than OLS and LLR. Several other examples, presented in Chapter 6, also showed the advantages of the model-robust procedures for single data sets. These results were based on theoretical INTMSE values, and were validated in section 8.B when it was shown that the theoretical MSE formulas for each fitting procedure were very accurate (close to the simulated MSE's). A study of confidence intervals for the optimal fits revealed two main conclusions. First, for small sample sizes, it appears that some additional work is needed to greatly decrease the widths of the LLR and model-robust C.I.'s to make them better than the OLS intervals, which were also too wide, but look better than the others. For larger sample sizes, however, it appears that the C.I.'s for the model-robust fitting techniques (based on optimal fits) provide adequate results. While on occasion giving slightly low coverage probabilities (for the three points at locations difficult to fit), the various confidence intervals were shown to usually have very sufficient coverages (probabilities in upper 80%'s to lower 90%'s) while maintaining appropriately small widths. All of the conclusions just mentioned establish that the model-robust procedures definitely have the potential to be very beneficial fitting techniques.

With the potential established, the only remaining question is whether or not this potential can be reached in practice. This issue was addressed in section 8.D by studying data-driven selectors of h and λ , and very promising results were found. PRESS* was shown to provide inadequate results (large h and small λ problems), but PRESS** produced much improved fits. In comparing the fits based on PRESS** to the optimal fits based on h_o and λ_o, the PRESS** fits were found to perform relatively close to optimal in almost all of the cases. These comparisons were made by observing the chosen h and λ values, the INTMSE values, and the confidence interval diagnostics. Actually, the MRR2 procedure was found to be the most consistent technique when using PRESS**, with LLR, MRR1, and PLR each having some type of problem with their fits. (LLR and MRR1 had large bandwidth problems (especially for small to moderate sample sizes), while PLR on occasion had problems with choosing a bandwidth too small, resulting in extremely wide confidence intervals). The most important conclusion to come out of this study is that the benefits of using a model-robust procedure over an individual parametric or nonparametric procedure can be maintained in practice, with the best method appearing to be MRR2 with h and λ based on PRESS**.

Chapter 9: Future Research

Each of the fitting techniques described in the preceding chapters involves only one of many variations for that particular technique. For example, local linear regression is just one of the many nonparametric fitting techniques described in chapter 3, and the LLR procedure itself can be altered by changing the method of choosing the bandwidth. This chapter briefly mentions some of the future work needed to determine if the forms of the techniques proposed in this current research are appropriate or can be improved upon. Also mentioned are some extensions and further developments of the techniques.

9.A Nonparametric Portion (Bandwidth Choice)

The most important component of nonparametric regression is the choice of the bandwidth h. As seen in previous chapters, an incorrect bandwidth can significantly affect the performance of any fitting technique that is dependent on this choice of h. The preliminary study presented in this paper studied a variety of possible data-driven bandwidth selectors before deciding on PRESS** as a promising candidate. Simulation results show this criterion to work relatively well, but there is still room for large improvements. One such need is more consistency across sample sizes. Possible improvements may include adjusting the current form of PRESS**, making use of current popular bandwidth selectors in the recent literature (Ruppert (1995), for example), or developing new procedures altogether. A possible approach in terms of adjusting the current form of PRESS** may be to somehow combine it with PRESS*; the idea being to somehow weight PRESS* more when there is little or no misspecification (when PRESS* performs well) and to weight PRESS** more when there is significant misspecification. Many alternatives to these ideas exist, and hopefully one can be found that consistently performs well.

9.B Model Robust Techniques

Choice of λ

The main area of future research in terms of combining the separate parametric and nonparametric fits is in choosing the mixing parameter λ. The ideas here follow closely those for further work on choosing the bandwidth. Studies are needed to better determine the effectiveness of PRESS**, and whether or not adjustments to PRESS**, or even totally different criteria, are needed. This information may be gathered by fixing the bandwidth at h_0 and then observing the performances of various selection criteria (in particular, PRESS**) in terms of how close they select λ to λ_0.

Error Variance, Confidence Intervals

More work is needed in developing better confidence intervals for the various procedures. A particular need is a method of accounting for high variances of the fits in small sample size cases, thus providing for a more consistent method of constructing C.I.'s. Also of interest is to find a method about as simple as those used in the current work that gives a little bit higher coverage probabilities for LLR and the model-robust techniques. Several more complicated techniques were introduced in section 6.D. Also of interest, in addition to forming C.I.'s for just the three "difficult to fit" x_0-values that were used in this paper, would be to study the C.I.'s formed for other types of points (boundary points, or points in smooth areas of the true underlying curve that should be fit easily).

Multiple Regression

Important for any regression technique is its ability to handle the multiple regressor situation. Future work needs to involve extensions of the model-robust procedures to this situation. Of interest are comparisons on the ease at which each procedure may be extended and studies of the performances of the fits themselves. The key step here is extending the nonparametric portions of the fits to multiple regression. For instance,

kernel regression may be extended by replacing the usual one-dimensional distance measure $(X_i - X_j)$ with an appropriate multi-dimensional measure $\| \mathbf{x}_i - \mathbf{x}_j \|$. Local polynomial regression may be extended by using regular weighted multiple regression, where the weights are based on kernel weights achieved by the multiple regression extension just mentioned above. These types of extensions need to be incorporated into the model-robust procedures, allowing for the development of a vast amount of other multiple regression techniques (such as variable selection methods).

Other Developments

To better establish model-robust regression as a basic regression tool, various other measures need to be developed. These include such measures as lack-of-fit, R^2-type measures of model adequacy, or possibly distributional results for λ (for MRR1 and/or MRR2). Lack-of-fit measures could prove very useful in providing additional support for the benefits of the model-robust procedures. In particular, it would be useful to have a measure of how far a model prescribed by the user is to a particular known true underlying model for the data structures used in this work (namely, for unreplicated data). One approach to this problem is given by Lawrence (1994). For example, it would be interesting to know how much lack-of-fit corresponds to each γ in the simulation study of this work (or how much power a lack-of-fit test would have (for each case). Such a measure would give a good indication as to how likely it is that a user would stick with his specified model, even when there is actually misspecification present. It is conjectured that many cases would arise where the user would use the specified model when the model-robust procedures would work much better. Such results could also be used to further support the main conclusion of this research. That is, to use a model-robust procedure (MRR2 seems best) for *any* regression situation where there is even the slightest hint of doubt about the validity of the specified model. If the specified model were actually correct (no lack-of-fit), then the model-robust procedure will perform as well as a parametric procedure. If the specified model is a gross misspecification (high

lack-of-fit), then the model-robust procedure will perform as well as a nonparametric procedure. And finally, if the specified model is adequate for some (or most) but not all of the data (moderate lack-of-fit), then the model-robust procedure will outperform either of the individual parametric or nonparametric procedures, possibly to a large degree.

References

Allen, D. (1974), "The Relationship Between Variable Selection and Data Augmentation and a Method for Prediction," *Technometrics*, **16**, 125-127.

Altman, N. (1992), "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *American Statistician*, **46**, 175-185.

Bishop, Y., Fienberg, S., and Holland, P. (1975), *Discrete Multivariate Analysis: Theory and Practice*. Cambridge: MIT Press.

Chiu, S. (1990), "Why Bandwidth Selectors Tend to Choose Smaller Bandwidths, and a Remedy," *Biometrika*, 77, 222-226.

Chu, C. (1989), "Some Results in Nonparametric Regression," *Ph.D. Dissertation*, Univ. North Carolina, Chapel Hill.

Chu, C. and Marron, J. (1991), "Choosing a Kernel Regression Estimator" (with discussion), *Statistical Science*, **6**, 404-436.

Cleveland, W. (1979), "Robust Locally Weighted Regression and Smoothing Scatter Plots," *Journal of the American Statistical Association*, **74**, 829-836.

Einsporn, R. (1987), "HATLINK: A Link Between Least Squares Regression and Nonparametric Curve Estimation," *Ph.D. thesis*, Virginia Polytechnic Institute and State University.

Einsporn, R. and Birch, J. (1993), "Model Robust Regression: Using Nonparametric Regression to Improve Parametric Regression Analyses," *Technical Report Number 93-5*, Dept. of Statistics, Virginia Polytechnic Institute and State University.

Epanechnikov, V. (1969), "Nonparametric Estimates of a Multivariate Probability Density," *Theory of Prob. and its Applications*, **14**, 153-158.

Eubank, R. (1988), *Spline Smoothing and Nonparametric Regression*. New York: Dekker.

Fan, J. (1992), "Design-adaptive Nonparametric Regression," *Journal of the American Statistical Association*, **87**, 998-1004.

Faraway, J. (1990), "Bootstrap Selection of Bandwidth and Confidence Bands for Nonparametric Regression," *J. Statist. Comput. Simul.*, **37**, 37-44.

Gasser, T. and Müller, H. (1979), "Kernel estimation of Regression Functions" in *Smoothing Techniques for Curve Estimation*, eds. Gasser and Rosenblatt. Heidelberg: Springer-Verlag.

Gasser, T., Müller, H., and Mammitzsch, V. (1985), "Kernels for Nonparametric Curve Estimation," *J. Royal Stat. Soc. B*, **47**, 238-252.

Green, P., Jennison, C., and Seheult, A. (1985), "Analysis of Field Experiments by Least Squares Smoothing," *J. Royal Stat. Soc. B*, **47**, 299-315.

Hall, P. and Wehrly, T. (1991), "A Geometrical Method for Removing Edge Effects from Kernel-type Nonparametric Regression Estimates," *Journal of the American Statistical Association*, **86**, 665-672.

Härdle, W. (1990), *Applied Nonparametric Regression*. New York: Cambridge University Press.

Härdle, W. and Bowman, A. (1988), "Bootstrapping in Nonparametric Regression: Local Adaptive Smoothing and Confidence Bands," *Journal of the American Statistical Association*, **83**, 102-110.

Härdle, W., Hall, P., and Marron, J. (1988), "How Far are Automatically Chosen Regression Smoothing Parameters from Their Optimum?" (with discussion), *Journal of the American Statistical Association*, **83**, 86-99.

Härdle, W. and Marron, J. (1985), "Optimal Bandwidth Selection in Nonparametric Regression Function Estimation," *Annals of Statistics*, **13**, 1465-1481.

Hastie, T. and Loader, C. (1993), "Local Regression: Automatic Kernel Carpentry" (with discussion), *Statistical Science*, **8**, 120-143.

Hastie, T. and Tibshirani, R. (1987), "Generalized Additive Models: Some Applications," *Journal of the American Statistical Association*, **82**, 371-386.

Hoaglin, D. and Welsch, R. (1978), "The Hat Matrix in Regression and ANOVA," *American Statistician*, **32**, 17-22.

Lawrance, A. J. (1994), "Testing Regression Lack of Fit Without Replication: A Tutorial Around Minitab's SLOF Procedures," *Journal of Applied Statistics*, **21**, 541-548

Messer, K. (1991), "A Comparison of a Spline Estimate to its Equivalent Kernel Estimate," *Annals of Statistics*, **19**, 817-829.

Montgomery, D. and Peck, E. (1992), *Introduction to Linear Regression Analysis*, second edition. New York: Wiley.

Müller, H. and Stadtmüller, U. (1987), "Variable Bandwidth Kernel Estimators of Regression Curves," *Annals of Statistics*, **15**, 182-201.

Myers, R. (1990), *Classical and Modern Regression with Applications*, second edition. Boston, MA: PWS-KENT.

Nadaraya, E. (1964), "On Estimating Regression," *Theory of Prob. and its Applications*, **9**, 141-142.

Olkin, I. and Spiegelman, C. (1987), "A Semiparametric Approach to Density Estimation," *Journal of the American Statistical Association*, **82**, 858-865.

Priestley, M. and Chao, M. (1972), "Nonparametric Function Fitting," *J. Royal Stat. Soc. B*, **34**, 384-392.

Reinsch, H. (1967), "Smoothing by Spline Functions," *Numerische Mathematik*, **10**, 177-183.

Rice, J. (1984a), "Bandwidth Choice for Nonparametric Regression," *Annals of Statistics*, **12**, 1215-1230.

Rice, J. (1984b), "Boundary Modification for Kernel Regression," *Communications in Statistics, Ser. A--Theory and Methods*, **13**, 893-900.

Ruppert, D. (1995), "Empirical-bias Bandwidths for Local Polynomial Nonparametric Regression and Density Estimation," Unpublished manuscript.

Shibata, R. (1981), "An Optimal Selection of Regression Variables," *Biometrika*, **68**, 45-54.

Silverman, B. (1984), "A Fast and Efficient Cross-validation Method for Smoothing Parameter Choice in Spline Regression," *Journal of the American Statistical Association*, **79**, 584-589.

Silverman, B. (1985), "Some Aspects of the Spline Smoothing Approach to Nonparametric Regression Curve Fitting" (with discussion), *J. Royal Stat. Soc. B*, **47**, 1-52.

Speckman, P. (1988), "Kernel Smoothing in Partial Linear Models," *J. Royal Stat. Soc. B*, **50**, 413-436.

Stine, R. (1989), "An Introduction to Bootstrap Methods--Examples and Ideas" in *Modern Methods of Data Analysis*, eds. Long, J. and Fox, J. Newbury Park, CA: Sage Publications.

Stone, C. (1980), "Optimal Rates of Convergence for Nonparametric Estimators," *Annals of Statistics*, **8**, 1348-1360.

Stone, C. (1982), "Optimal Global Rates of Convergence for Nonparametric Regression," *Annals of Statistics*, **10**, 1040-1053.

Stone, M. (1974), "Cross-validatory Choice and Assessment of Statistical Predictions" (with discussion), *J. Royal Stat. Soc. B*, **36**, 111-147.

Watson, G. (1964), "Smooth Regression Analysis," *Sankhya Ser. A*, **26**, 359-372.

Wong, W. (1982), "On the Consistency of Cross-validation in Kernel Nonparametric Regression," *Annals of Statistics*, **11**, 1136-1141.

Appendix

(A, B, C, D)

Appendix A: Choice of Penalizing Function

Härdle (1990) defines the general weight sequence $\{W_{hj}(x)\}_{j=1}^{n}$ for obtaining kernel predictions at location x as

$$W_{hj}(x) = \frac{h^{-1}K\left(\frac{x-x_j}{h}\right)}{\hat{g}_h(x)}, \tag{A.1}$$

where h is the bandwidth, K is the kernel, and $\hat{g}_h(\cdot)$ is the Rosenblatt-Parzen kernel density estimator of the (marginal) density of X. (Note: the Nadaraya-Watson estimate of equation (3.B.3) is achieved by defining

$$\hat{g}_h(x) = n^{-1}\sum_{j=1}^{n} h^{-1}K\left(\frac{x-x_j}{h}\right). \tag{A.2}$$

From the weight sequence in (A.1), the general kernel estimator of the true function f can be expressed as

$$\hat{f}_h(x) = \frac{\sum_{j=1}^{n} W_{hj}(x)\, y_j}{n} = \frac{\sum_{j=1}^{n} h^{-1}K\left(\frac{x-x_j}{h}\right) y_j}{n\,\hat{g}_h(x)} = \frac{n^{-1}h^{-1}\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right) y_j}{\hat{g}_h(x)}. \tag{A.3}$$

Now, the argument u of the penalizing function $\Xi(u)$ is defined to be $n^{-1}W_{hj}(x_j)$, giving the function $\Xi[n^{-1}W_{hj}(x_j)]$. To see how this results in penalizing for small h, note that

$$\Xi[n^{-1}W_{hj}(x_j)] = \Xi\left[\frac{n^{-1}h^{-1}K\left((x_j - x_j)/h\right)}{\hat{g}_h(x_j)}\right] = \Xi\left[\frac{n^{-1}h^{-1}K(0)}{\hat{g}_h(x_j)}\right], \tag{A.4}$$

a function clearly increasing as h gets smaller (since $\Xi(u)$ is defined to be increasing in u).

Appendix B: Technical Assumptions

The following are the five technical assumptions necessary for the asymptotic bias and variance expressions of equations (3.B.16-20). The first three are needed in the fixed design case, with the last two added for the random design case.

A1. f is twice continuously differentiable on a neighborhood of the point x;

A2. K is a symmetric, probability density supported on [-1, 1], bounded above 0 on [-1/2, 1/2], with a bounded derivative;

A3. $n \to \infty$, with $n^{-1+\delta} \leq h \leq n^{-\delta}$, for some $\delta \in (0, 1/2)$;

A4. the marginal density g of x_j has a bounded and continuous first derivative and is bounded above zero, on a neighborhood of x;

A5. x_j and ε_j are uncorrelated.

Appendix C: X Matrix in PLR

For partial linear regression (PLR), suppose that $\mathbf{H}^{(\text{ker})}$ (= $\mathbf{H}_{\text{P}}^{(\text{ker})}$) is the kernel hat matrix obtained from kernel smoothing on the regressor X, where the rows of $\mathbf{H}^{(\text{ker})}$ each sum to one. Defining $\tilde{\mathbf{X}} = (\mathbf{I} - \mathbf{H}^{(\text{ker})})\mathbf{X}$ and $\hat{\beta} = (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{y}}$, the following proof shows why the matrix of regressors $\mathbf{X}$ cannot contain a column of ones.

Proof: For $\mathbf{H}^{(\text{ker})} = (h_{ij}^{(\text{ker})})$, it is known that ***(1)*** $\sum_{j=1}^{n} h_{ij}^{(\text{ker})} = 1$ for $i = 1, 2, \ldots, n$ (i.e., the rows of $\mathbf{H}^{(\text{ker})}$ each sum to one).

(2) Assume that $\mathbf{X}$ *does* contain a column of ones, say this is column c.

Consider the c^{th} column of

$$\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_1\ \tilde{\mathbf{X}}_2\ \ldots\ \tilde{\mathbf{X}}_p] = (\mathbf{I} - \mathbf{H}^{(\text{ker})})[\mathbf{X}_1\ \mathbf{X}_2\ \ldots\ \mathbf{X}_p] = (\mathbf{I} - \mathbf{H}^{(\text{ker})})\mathbf{X}:$$

$$\tilde{\mathbf{X}}_c = (\mathbf{I} - \mathbf{H}^{(\text{ker})})\mathbf{X}_c$$

$$= (\mathbf{I} - \mathbf{H}^{(\text{ker})})\,\mathbf{1} \quad (\text{by } \textit{(2)})$$

$$= \mathbf{1} - \mathbf{H}^{(\text{ker})}\,\mathbf{1}$$

$$= \begin{bmatrix} 1 - \sum_{j=1}^{n} h_{1j}^{(\text{ker})} \\ 1 - \sum_{j=1}^{n} h_{2j}^{(\text{ker})} \\ \vdots \\ 1 - \sum_{j=1}^{n} h_{nj}^{(\text{ker})} \end{bmatrix} = \begin{bmatrix} 1-1 \\ 1-1 \\ \vdots \\ 1-1 \end{bmatrix} (\text{by } \textit{(1)}) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{0}.$$

Hence, $\tilde{\mathbf{X}}$ contains a column of zeros, and $\tilde{\mathbf{X}}'\tilde{\mathbf{X}}$ contains a column (and a row) of zeros. Thus, $\tilde{\mathbf{X}}'\tilde{\mathbf{X}}$ is a singular matrix, and $(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}$ does not exist. This implies that $\hat{\beta} = (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{y}}$ does not exist.

Thus, to obtain estimates for PLR, assumption ***(2)*** above must be incorrect, and so $\mathbf{X}$ cannot contain a column of ones ∎

Appendix D: Bias and Variance Derivations

** Note that all results here are for fixed bandwidths (h) and mixing parameters (λ).

Appendix D.1: Kernel Regression

Consider the general underlying model $\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon} = \mathbf{X}\boldsymbol{\beta} + \mathbf{f} + \boldsymbol{\varepsilon}$, where $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $Var(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$. The kernel fitted values are $\hat{\mathbf{y}}_{ker} = \mathbf{H}^{(ker)}\mathbf{y}$. To simplify notation, define the kernel hat matrix as $\mathbf{H}^{(ker)} = \mathbf{K}$. The bias and variance of $\hat{\mathbf{y}}_{ker}$ are then as follows:

$$\begin{aligned} Bias(\hat{\mathbf{y}}_{ker}) &= E(\hat{\mathbf{y}}_{ker}) - E(\mathbf{y}) \\ &= E(\mathbf{Ky}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) \\ &= \mathbf{K}(\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) - \mathbf{X}\boldsymbol{\beta} - \mathbf{f} \\ &= \mathbf{KX}\boldsymbol{\beta} + \mathbf{Kf} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f} \\ &= -(\mathbf{I} - \mathbf{K})\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{K})\mathbf{f} \\ &= -(\mathbf{I} - \mathbf{K})(\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) \blacksquare \end{aligned}$$

$$\begin{aligned} Var(\hat{\mathbf{y}}_{ker}) &= Var(\mathbf{Ky}) \\ &= \mathbf{K}Var(\mathbf{y})\mathbf{K}' \\ &= \mathbf{K}(\sigma^2\mathbf{I})\mathbf{K}' \\ &= \sigma^2\mathbf{KK}' \blacksquare \end{aligned}$$

These are equations (6.B.5) and (6.B.6), respectively.

Appendix D.2: MRR1

Consider the underlying model $\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon} = \mathbf{X}\boldsymbol{\beta} + \mathbf{f} + \boldsymbol{\varepsilon}$, where $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $\text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$. The MRR1 fitted values are $\hat{\mathbf{y}}_{MRR1} = \mathbf{H}^{(MRR1)}\mathbf{y} = [\lambda\mathbf{H}^{(ker)} + (1-\lambda)\mathbf{H}^{(ols)}]\mathbf{y}$. To simplify notation, define the kernel hat matrix as $\mathbf{H}^{(ker)} = \mathbf{K}$ and the OLS hat matrix as $\mathbf{H}^{(ols)} = \mathbf{H}$. Also, note that *(1)* $\mathbf{HX} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = \mathbf{X}$, *(2)* $\mathbf{H}' = \mathbf{H}$, and *(3)* $\mathbf{HH} = \mathbf{H}$. The bias and variance of $\hat{\mathbf{y}}_{MRR1}$ are then as follows:

$$
\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}_{MRR1}) &= E(\hat{\mathbf{y}}_{MRR1}) - E(\mathbf{y}) = E(\mathbf{H}^{(MRR1)}\mathbf{y}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) = \\
&= \mathbf{H}^{(MRR1)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) - \mathbf{X}\boldsymbol{\beta} - \mathbf{f} \\
&= \mathbf{H}^{(MRR1)}\mathbf{X}\boldsymbol{\beta} + \mathbf{H}^{(MRR1)}\mathbf{f} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f} \\
&= -(\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{f} \\
&= -(\mathbf{I} - \lambda\mathbf{K} - (1-\lambda)\mathbf{H})\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{f} \\
&= -\mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{K}\mathbf{X}\boldsymbol{\beta} + (1-\lambda)\mathbf{H}\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{f} \\
&= -\mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{K}\mathbf{X}\boldsymbol{\beta} + (1-\lambda)\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{f} \qquad \text{(by } \textit{(1)}\text{)} \\
&= -\mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{K}\mathbf{X}\boldsymbol{\beta} + \mathbf{X}\boldsymbol{\beta} - \lambda\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{f} \\
&= -\lambda\mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{K}\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{f} \\
&= -\lambda(\mathbf{I} - \mathbf{K})\mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR1)})\mathbf{f} \blacksquare
\end{aligned}
$$

$$
\begin{aligned}
\text{Var}(\hat{\mathbf{y}}_{MRR1}) &= \text{Var}(\mathbf{H}^{(MRR1)}\mathbf{y}) \\
&= \mathbf{H}^{(MRR1)}\text{Var}(\mathbf{y})\mathbf{H}'^{(MRR1)} \qquad (= \sigma^2\mathbf{H}^{(MRR1)}\mathbf{H}'^{(MRR1)}) \\
&= [\lambda\mathbf{K} + (1-\lambda)\mathbf{H}](\sigma^2\mathbf{I})\,[\lambda\mathbf{K} + (1-\lambda)\mathbf{H}]' \\
&= \sigma^2[\lambda^2\mathbf{K}\mathbf{K}' + (1-\lambda)\lambda\mathbf{H}\mathbf{K}' + \lambda(1-\lambda)\mathbf{K}\mathbf{H}' + (1-\lambda)^2\mathbf{H}\mathbf{H}'] \\
&= \sigma^2[\lambda^2\mathbf{K}\mathbf{K}' + (1-\lambda)\lambda\mathbf{H}\mathbf{K}' + \lambda(1-\lambda)\mathbf{K}\mathbf{H} + (1-\lambda)^2\mathbf{H}] \qquad \text{(by } \textit{(2)}\text{, } \textit{(3)}\text{)} \\
&= \sigma^2\{\,\lambda[\lambda\mathbf{K} + (1-\lambda)\mathbf{H}]\mathbf{K}' + (1-\lambda)[(1-\lambda)\mathbf{I} + \lambda\mathbf{K}]\mathbf{H}\,\} \\
&= \sigma^2\{\,\lambda\mathbf{H}^{(MRR1)}\mathbf{K}' + (1-\lambda)[\mathbf{I} - \lambda(\mathbf{I} - \mathbf{K})]\mathbf{H}\,\} \blacksquare
\end{aligned}
$$

These are equations (6.B.9) and (6.B.10), respectively.

Appendix D.3: MRR2

Consider the underlying model $\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon} = \mathbf{X}\boldsymbol{\beta} + \mathbf{f} + \boldsymbol{\varepsilon}$, where $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $Var(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$. The MRR2 fitted values are $\hat{\mathbf{y}}_{MRR2} = \mathbf{H}^{(MRR2)}\mathbf{y} = [\mathbf{H}^{(ols)} + \lambda\mathbf{H}_2^{(ker)}(\mathbf{I}-\mathbf{H}^{(ols)})]\mathbf{y}$, where $\mathbf{H}_2^{(ker)}$ is the kernel hat matrix for a kernel fit to the residuals from the parametric (OLS) fit. To simplify notation, define the kernel hat matrix as $\mathbf{H}_2^{(ker)} = \mathbf{K}$ and the OLS hat matrix as $\mathbf{H}^{(ols)} = \mathbf{H}$. Also, notice that *(1)* $\mathbf{HX} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = \mathbf{X}$, *(2)* $\mathbf{H}' = \mathbf{H}$, and *(3)* $\mathbf{HH} = \mathbf{H}$. The bias and variance of $\hat{\mathbf{y}}_{MRR2}$ are then as follows:

$$
\begin{aligned}
Bias(\hat{\mathbf{y}}_{MRR2}) &= E(\hat{\mathbf{y}}_{MRR2}) - E(\mathbf{y}) \\
&= E(\mathbf{H}^{(MRR2)}\mathbf{y}) - (\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) \\
&= \mathbf{H}^{(MRR2)}(\mathbf{X}\boldsymbol{\beta} + \mathbf{f}) - \mathbf{X}\boldsymbol{\beta} - \mathbf{f} \\
&= \mathbf{H}^{(MRR2)}(\mathbf{X}\boldsymbol{\beta}) + \mathbf{H}^{(MRR2)}\mathbf{f} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f} \\
&= [\mathbf{H} + \lambda\mathbf{K}(\mathbf{I}-\mathbf{H})](\mathbf{X}\boldsymbol{\beta}) - \mathbf{X}\boldsymbol{\beta} + \mathbf{H}^{(MRR2)}\mathbf{f} - \mathbf{f} \\
&= \mathbf{HX}\boldsymbol{\beta} + \lambda\mathbf{K}(\mathbf{I}-\mathbf{H})\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{f} \\
&= \mathbf{HX}\boldsymbol{\beta} + \lambda\mathbf{K}(\mathbf{X}\boldsymbol{\beta} - \mathbf{HX}\boldsymbol{\beta}) - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{f} \\
&= \mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{K}(\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta}) - \mathbf{X}\boldsymbol{\beta} - (\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{f} \qquad \text{(by } \textit{(1)}\text{)} \\
&= -(\mathbf{I} - \mathbf{H}^{(MRR2)})\mathbf{f} \blacksquare
\end{aligned}
$$

$$
\begin{aligned}
Var(\hat{\mathbf{y}}_{MRR2}) &= Var(\mathbf{H}^{(MRR2)}\mathbf{y}) \\
&= \mathbf{H}^{(MRR2)}Var(\mathbf{y})\mathbf{H}'^{(MRR2)} \qquad (= \sigma^2\mathbf{H}^{(MRR1)}\mathbf{H}'^{(MRR1)}) \\
&= [\mathbf{H} + \lambda\mathbf{K}(\mathbf{I}-\mathbf{H})](\sigma^2\mathbf{I})[\mathbf{H} + \lambda\mathbf{K}(\mathbf{I}-\mathbf{H})]' \\
&= \sigma^2[\mathbf{HH}' + \lambda\mathbf{K}(\mathbf{I}-\mathbf{H})\mathbf{H}' + \lambda\mathbf{H}(\mathbf{I}-\mathbf{H})'\mathbf{K}' + \lambda^2\mathbf{K}(\mathbf{I}-\mathbf{H})(\mathbf{I}-\mathbf{H})'\mathbf{K}'] \\
&= \sigma^2[\mathbf{HH} + \lambda\mathbf{K}(\mathbf{I}-\mathbf{H})\mathbf{H} + \lambda\mathbf{H}(\mathbf{I}-\mathbf{H})\mathbf{K}' + \lambda^2\mathbf{K}(\mathbf{I}-\mathbf{H})(\mathbf{I}-\mathbf{H})\mathbf{K}'] \quad \text{(by } \textit{(2)}\text{)} \\
&= \sigma^2[\mathbf{H} + \lambda\mathbf{K}(\mathbf{H}-\mathbf{H}) + \lambda(\mathbf{H}-\mathbf{H})\mathbf{K}' + \lambda^2\mathbf{K}(\mathbf{I}-\mathbf{H})\mathbf{K}'] \qquad \text{(by } \textit{(3)}\text{)} \\
&= \sigma^2[\mathbf{H} + \lambda^2\mathbf{K}(\mathbf{I}-\mathbf{H})\mathbf{K}'] \blacksquare
\end{aligned}
$$

These are equations (6.B.13) and (6.B.14), respectively.

Appendix D.3: MRR2 (*cont.*)

In this section, the bias and variance expressions ((6.B.9) and (6.B.10)) for determining the optimal bandwidth h_o for MRR2 will be developed. This bandwidth is for the kernel fit to the residuals from the OLS fit, which may be expressed as $\hat{\mathbf{r}} = \mathbf{H}_2^{(ker)}\mathbf{r}$, where $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{ols}$ and $\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon} = \mathbf{X}\boldsymbol{\beta} + \mathbf{f} + \boldsymbol{\varepsilon}$, where $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $Var(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$. Defining $\mathbf{H}_2^{(ker)} = \mathbf{K}$ and $\mathbf{H}^{(ols)} = \mathbf{H}$, the bias and variance of $\hat{\mathbf{r}}$ are as follows:

$$
\begin{aligned}
Bias(\hat{\mathbf{r}}) &= E(\hat{\mathbf{r}}) - E(\mathbf{r}) = E(\mathbf{Kr}) - E(\mathbf{r}) = \\
&= \mathbf{K}E(\mathbf{r}) - E(\mathbf{r}) = -(\mathbf{I} - \mathbf{K})E(\mathbf{r}) = \\
&= -(\mathbf{I} - \mathbf{K})[E(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{ols})] \\
&= -(\mathbf{I} - \mathbf{K})(E[(\mathbf{I} - \mathbf{H})\mathbf{y}]) \qquad (\text{since } \hat{\mathbf{y}}_{ols} = \mathbf{H}\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}}_{ols}) \\
&= -(\mathbf{I} - \mathbf{K})[(\mathbf{I} - \mathbf{H})E(\mathbf{y})] \\
&= -(\mathbf{I} - \mathbf{K})[(\mathbf{I} - \mathbf{H})(\mathbf{X}\boldsymbol{\beta} + \mathbf{f})] \\
&= -(\mathbf{I} - \mathbf{K})[(\mathbf{I} - \mathbf{H})\mathbf{X}\boldsymbol{\beta} + (\mathbf{I} - \mathbf{H})\mathbf{f}] \\
&= -(\mathbf{I} - \mathbf{K})[\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta} + (\mathbf{I} - \mathbf{H})\mathbf{f}] \qquad (\text{by } \textit{(1)}) \\
&= -(\mathbf{I} - \mathbf{K})(\mathbf{I} - \mathbf{H})\mathbf{f}\ \blacksquare
\end{aligned}
$$

$$
\begin{aligned}
Var(\hat{\mathbf{r}}) &= Var[\mathbf{K}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{ols})] \\
&= \mathbf{K}Var(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{ols})\mathbf{K}' \\
&= \mathbf{K}Var[(\mathbf{I} - \mathbf{H})\mathbf{y}]\mathbf{K}' \\
&= \mathbf{K}(\mathbf{I} - \mathbf{H})Var(\mathbf{y})(\mathbf{I} - \mathbf{H})'\mathbf{K}' \\
&= \mathbf{K}(\mathbf{I} - \mathbf{H})(\sigma^2\mathbf{I})(\mathbf{I} - \mathbf{H})'\mathbf{K}' \\
&= \sigma^2\mathbf{K}(\mathbf{I} - \mathbf{H})(\mathbf{I} - \mathbf{H})'\mathbf{K}' \\
&= \sigma^2\mathbf{K}(\mathbf{I} - \mathbf{H})\mathbf{K}'\ \blacksquare \qquad (\text{by } \textit{(3)})
\end{aligned}
$$

These are equations (6.B.15) and (6.B.16), respectively.

Appendix D.4 (a): PLR (when using kernel regression to fit residuals)

Consider the underlying model $\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon} = \mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f} + \boldsymbol{\varepsilon}$, where $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $\text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$, and $\mathbf{X}_P$ is the $\mathbf{X}$ matrix without a column of ones. The PLR fitted values are $\hat{\mathbf{y}}_{PLR} = \mathbf{H}^{(PLR)}\mathbf{y} = [\mathbf{H}_P^{(ker)} + \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{H}_P^{(ker)})]\mathbf{y}$, where $\tilde{\mathbf{X}} = (\mathbf{I} - \mathbf{H}_P^{(ker)})\mathbf{X}_P$. To simplify notation, define the kernel hat matrix as $\mathbf{H}_P^{(ker)} = \mathbf{K}$. The bias and variance of $\hat{\mathbf{y}}_{PLR}$ are then as follows:

$$
\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}_{PLR}) &= E(\hat{\mathbf{y}}_{PLR}) - E(\mathbf{y}) = E(\mathbf{H}^{(PLR)}\mathbf{y}) - (\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f}) = \\
&= \mathbf{H}^{(PLR)}E(\mathbf{y}) - \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{f} \\
&= \mathbf{H}^{(PLR)}(\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f}) - \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{f} \\
&= [\mathbf{K} + \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})]\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{H}^{(PLR)}\mathbf{f} - \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{f} \\
&= \mathbf{K}\mathbf{X}_P\boldsymbol{\beta}_P + \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})\mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= \mathbf{K}\mathbf{X}_P\boldsymbol{\beta}_P + \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{X}}\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= \mathbf{K}\mathbf{X}_P\boldsymbol{\beta}_P + \tilde{\mathbf{X}}\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= \mathbf{K}\mathbf{X}_P\boldsymbol{\beta}_P + (\mathbf{I} - \mathbf{K})\mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= \mathbf{K}\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{K}\mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= -(\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f}\ \blacksquare
\end{aligned}
$$

$$
\begin{aligned}
\text{Var}(\hat{\mathbf{y}}_{PLR}) &= \text{Var}(\mathbf{H}^{(PLR)}\mathbf{y}) \\
&= \mathbf{H}^{(PLR)}\text{Var}(\mathbf{y})\mathbf{H}'^{(PLR)} \qquad (= \sigma^2\mathbf{H}^{(PLR)}\mathbf{H}'^{(PLR)}) \\
&= \sigma^2[\mathbf{K} + \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})][\mathbf{K} + \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})]' \\
&= \sigma^2[\mathbf{K} + \mathbf{P}_{\tilde{\mathbf{X}}}(\mathbf{I} - \mathbf{K})][\mathbf{K} + \mathbf{P}_{\tilde{\mathbf{X}}}(\mathbf{I} - \mathbf{K})]' \qquad (\text{defining } \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}' = \mathbf{P}_{\tilde{\mathbf{X}}}) \\
&= \sigma^2[\mathbf{K}\mathbf{K}' + \mathbf{K}(\mathbf{I} - \mathbf{K})'\mathbf{P}_{\tilde{\mathbf{X}}}' + \mathbf{P}_{\tilde{\mathbf{X}}}(\mathbf{I} - \mathbf{K})\mathbf{K}' + \mathbf{P}_{\tilde{\mathbf{X}}}(\mathbf{I} - \mathbf{K})(\mathbf{I} - \mathbf{K})'\mathbf{P}_{\tilde{\mathbf{X}}}'] \\
&= \sigma^2[\mathbf{K}\mathbf{K}' + \mathbf{K}(\mathbf{I} - \mathbf{K})'\mathbf{P}_{\tilde{\mathbf{X}}} + \mathbf{P}_{\tilde{\mathbf{X}}}(\mathbf{I} - \mathbf{K})\mathbf{K}' + \mathbf{P}_{\tilde{\mathbf{X}}}(\mathbf{I} - \mathbf{K})(\mathbf{I} - \mathbf{K})'\mathbf{P}_{\tilde{\mathbf{X}}}]\ \blacksquare
\end{aligned}
$$

These are equations (6.B.19) and (6.B.20), respectively.

Appendix D.4 (b): PLR (when using local polyn. regression to fit residuals)

Consider the underlying model $\mathbf{y} = \mathbf{g}(x) + \boldsymbol{\varepsilon} = \mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f} + \boldsymbol{\varepsilon}$, where $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $\text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$, and $\mathbf{X}_P$ is the $\mathbf{X}$ matrix without a column of ones. The PLR fitted values are $\hat{\mathbf{y}}_{PLR} = \mathbf{H}^{(PLR)}\mathbf{y} = [\mathbf{H}_P^{(LPR)} + (\mathbf{I} - \mathbf{H}_P^{(LPR)})\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{H}_P^{(ker)})]\mathbf{y}$, where, once again, $\tilde{\mathbf{X}} = (\mathbf{I} - \mathbf{H}_P^{(ker)})\mathbf{X}_P$. To simplify notation, define the kernel hat matrix as $\mathbf{H}_P^{(ker)} = \mathbf{K}$ and the LPR hat matrix for fitting the residuals as $\mathbf{H}_P^{(LPR)} = \mathbf{K}_L$. The bias and variance of $\hat{\mathbf{y}}_{PLR}$ are then as follows:

$$
\begin{aligned}
\text{Bias}(\hat{\mathbf{y}}_{PLR}) &= E(\hat{\mathbf{y}}_{PLR}) - E(\mathbf{y}) = E(\mathbf{H}^{(PLR)}\mathbf{y}) - (\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f}) = \\
&= \mathbf{H}^{(PLR)}E(\mathbf{y}) - \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{f} \\
&= \mathbf{H}^{(PLR)}(\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{f}) - \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{f} \\
&= [\mathbf{K}_L + (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})]\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{H}^{(PLR)}\mathbf{f} - \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{f} \\
&= \mathbf{K}_L\mathbf{X}_P\boldsymbol{\beta}_P + (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})\mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= \mathbf{K}_L\mathbf{X}_P\boldsymbol{\beta}_P + (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{X}}\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= \mathbf{K}_L\mathbf{X}_P\boldsymbol{\beta}_P + (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= \mathbf{K}_L\mathbf{X}_P\boldsymbol{\beta}_P + \mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{K}_L\mathbf{X}_P\boldsymbol{\beta}_P - \mathbf{X}_P\boldsymbol{\beta}_P - (\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \\
&= -(\mathbf{I} - \mathbf{H}^{(PLR)})\mathbf{f} \blacksquare
\end{aligned}
$$

$$
\begin{aligned}
\text{Var}(\hat{\mathbf{y}}_{PLR}) &= \text{Var}(\mathbf{H}^{(PLR)}\mathbf{y}) \\
&= \mathbf{H}^{(PLR)}\text{Var}(\mathbf{y})\mathbf{H}'^{(PLR)} \qquad (= \sigma^2\mathbf{H}^{(PLR)}\mathbf{H}'^{(PLR)}) \\
&= \sigma^2[\mathbf{K}_L + (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})][\mathbf{K}_L + (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})]' \\
&= \sigma^2[\mathbf{K}_L\mathbf{K}_L' + \mathbf{K}_L(\mathbf{I} - \mathbf{K})'\tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\mathbf{X}_P'(\mathbf{I} - \mathbf{K}_L)' + \\
&\quad (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})\mathbf{K}_L' + \\
&\quad (\mathbf{I} - \mathbf{K}_L)\mathbf{X}_P(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{K})(\mathbf{I} - \mathbf{K})'\tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\mathbf{X}_P'(\mathbf{I} - \mathbf{K}_L)'] \blacksquare
\end{aligned}
$$

These are equations (6.B.21) and (6.B.22), respectively.

The vita has been removed from the scanned document