2. Polynomial Response Surfaces

2.1. Introduction

Most optimization algorithms that are in use for solving analytical engineering optimization problems are sequential in nature. That is, the objective function and constraints are evaluated at one point at a time, and the values at that point, as well as previous design points, contribute to a decision on where in the design space to move to for the next evaluation.

When the objective functions and (or) the constraints are evaluated by experiments rather than by analytical/computational evaluations, there is usually an incentive to perform the experiments in batches rather than sequentially. One reason for batching the experiments is that most require setup time, advanced planning and reservations of experimental facilities or technicians. Another reason is that experimental errors make it difficult to interpret the results of a single experiment. When a batch of experiments is performed, errors in one or two experiments tend to stand out. Duplicating experiments for identically nominal conditions permits us to estimate the magnitude of experimental scatter due to errors and variability in the properties of the tested designs. Finally, some of the experimental scatter can be averaged out by performing a large number of experiments.

Because of these advantages of running experiments in batches, experimental optimization has followed a different route than analytical optimization. The standard approach is to use an optimization strategy that is based on the results of a batch of experiments. On the basis of the experiments, we construct approximations to objective functions and/or constraints and perform optimization based on these approximations. In most cases, the optimum obtained is then tested, and if satisfactory results are obtained the design procedure is terminated. In some cases, the optimum is used as the central design point for a new batch of experiments, and the process is repeated once or twice. This process is sometimes called sequential approximate optimization. However, because of the cost and time associated with conducting experiments, it is rare that the process is iterated to convergence.

When analytical calculations were mostly based on closed-form solutions or numerical models that required minimal modeling and computations, the difference between analysis and experiments was very clear. However, today numerical evaluations of objective functions and constraints often share many of the properties of experimental evaluations. First, numerical models such as finite element structural models require a substantial investment of time to set up and debug. Furthermore, the evaluation of such models may require large computational resources, so that the cost of numerical simulation may be comparable to the cost of experiments. Second, with analytical simulations based on complex numerical models, many sources of noise are often found in the results of numerical simulations. These include round-off errors as well as errors due to incomplete convergence of iterative processes. Additionally, numerical simulations are usually based on the discretization of continua, and the accuracy of this discretization depends on the shape of the domains being discretized. For example, when stress analysis is performed in an elastic body using finite element discretization, the discretization error does not change smoothly with the shape of the body, because the number of finite elements can vary only in integer increments.

These growing similarities between analytical simulations and experiments create incentives to run analytical simulations in batches and use sequential approximate optimization. Additionally, the growing availability of parallel computers also provides incentives for running analytical simulations in batches. Finally, numerical simulations are often run with software packages that are difficult to connect directly to optimization programs. Approximate sequential optimization provides a mechanism for running these software packages in a stand-alone mode and connecting the optimization programs to the approximation.

The polynomial response surface (PRS) is the oldest form of a surrogate model where it approximates discrete experimental data (output) with different test parameters (input) using simple polynomials. PRS was invented in the 1920s to characterize crop yields in terms of inputs such as water and fertilizer. It was called the response surface approximation. The algebraic function to fit data is called surrogate, metamodel, or approximation. The term "surrogate" captures the purpose of the fit: using it instead of experiments for prediction. It is often important when data is expensive and noisy, especially for optimization and uncertainty quantification.

The process of identifying the relationship between the input parameters and output quantity of interest (QoI) is statistically referred to as regression. The term comes from the paper by Galton [10], where regression was used that happened to be about a phenomenon called regression towards the mean. He found that children of tall parents tended to be shorter than their parents, while children of short parents tended to be taller than their parents.

Through regression, a seemingly complicated behavior of QoI is approximated by simple functions. Normally linear or quadratic polynomials are used for approximation because the main goal is to find the trend of the QoI as test parameters change [4]. Since most experimental data include measurement noise, the PRS surrogate is designed to compensate for the noise. Therefore, the PRS surrogate normally does not pass through the data points. Instead, the differences between PRS predictions and measure data are assumed to be randomly distributed. In fact, the differences between PRS predictions and measured data are used to estimate the level of noise in the measurement.

In the modern form of PRS, the surrogate approximates the trend of data using a linear combination of polynomials. In this form, each monomial term is referred to as 'basis', and the constants that are multiplied with the bases are called unknown coefficients. The major process of building a PRS surrogate is to find the unknown coefficients by minimizing errors between the data and the surrogate predictions at the data points. In statistics, this is called linear regression, where the output QoI is a linear function of unknown coefficients—the basis functions can be nonlinear. In particular, when a single QoI is modeled as a linear combination of basis functions, it is called simple linear regression, compared to multiple linear regression where multiple QoIs are modeled simultaneously [11]. This is different from multivariate linear regression, where multiple 'correlated' QoIs are predicted, rather than a single QoI [12]. In the following discussion, we will only consider simple regression.

In the regression process, it is assumed that the numbers and orders of basis functions are predetermined, and only the unknown coefficients are to be determined. Therefore, the following questions naturally arise: (a) how to determine the numbers and orders of basis functions, (b) how to determine the unknown coefficients, and (c) how many data are required to build the surrogate model.

The second question of how to determine the unknown coefficients is the easiest part of surrogate modeling, which will be discussed in Section 2.2 in detail. In general, the unknown coefficients are determined by minimizing errors between the given data and surrogate predictions. It is obvious that the outcome will be different depending on how to define the errors and how to minimize them. Section 2.3 will discuss how to estimate the unknown coefficients using linear regression.

The first question of how to determine the numbers and orders of basis functions is subjective and requires domain knowledge of the specific application. Since the success of PRS surrogate modeling depends on the selection of basis functions, this step is critically important, and yet, it is hard to choose an appropriate set of basis functions. For example, when real physics behaves like a sinusoidal function, it would be hard to approximate the function using a set of polynomials. If the domain of input variables is not too wide, the Taylor series expansion can be used to justify the polynomial-based approximation. In general, when the model form of PRS is unknown, stepwise regression can be used, which will be

discussed in Section 2.5. This method starts with enough numbers and orders of basis functions, and gradually removes those basis functions that are not important or significant for the approximation.

The last question of how many data are required is related to the complexity of the surrogate model and the accuracy of prediction. Since the main usage of samples is to determine the unknown coefficients, the number of samples should be larger than that of unknown coefficients. Theoretically, m numbers of samples should be good enough to determine m numbers of coefficients. However, this corresponds to solving a linear system of m equations, not regression. In general, the required number of samples is at least two or three folds of that of the unknown coefficients. Not only the number of samples but also the location of samples is important in building a good surrogate model. It is always good that the sample locations are populated in the entire input space while maintaining a small distance between them. However, as shown in Figure 1-5, it is difficult to cover the entire input space with a small number of samples, and it is inevitable to deal with a large extrapolation region, which is indeed the major challenge in surrogate modeling. Since selecting samples is an important topic, Chapter 3 is dedicated to the design of experiments. Therefore, in this chapter, only the first two questions will be addressed.

2.2. Curve fitting

Before discussing the details of PRS, it would be beneficial to discuss curve fitting first since it is the simplest version of PRS. Consider that experimental data y_i are measured at different values of parameter x_i. Then we want to fit a curve that matches the data best in some sense. When we fit a curve to data, it is necessary to ask the following questions: (a) what is the error metric for the best fit? and (b) what is more accurate, the data or the fit? Although the answer to the first question is a matter of choice, the second one is subtle and needs to be understood thoroughly. In the following example, we start with the following assumptions: (a) data are noisy, (b) the functional form of the true function is known, and (c) data are dense enough to allow us to filter noise in data.

In order to explain the accuracy between the data and the fit, let us generate samples from a linear function $y = x$ at $x = 1, 2, \cdots, 30$. In order to make the data noisy, we can add random noise from a normal distribution $\sim N(0, 1^2)$ to the data. Using the randomized data, we can fit a linear polynomial using the `polyfit` function of Matlab, which determines the unknown regression coefficients. Once the `polyfit` function calculates the unknown coefficients, `p`, they can be used in the `polyval` function to evaluate the surrogate at any prediction point, `x`. The following Matlab script generates samples, fit the samples with a linear polynomial, and calculate the root-mean-square error of the data and that of the fit.

```
noise=randn(1,30);
x=1:1:30; y=x+noise;
[p,s]=polyfit(x,y,1);
yfit=polyval(p,x);
plot(x,y,'+',x,x,'r',x,yfit,'b');
legend('Data','True function','Fitted function');
RMS_data=sqrt(sum((x-y).^2)/30)
RMS_fit=sqrt(sum((x-yfit).^2)/30)
mean(noise)
```

Due to the randomness of noise, the above Matlab code yields different results at different trials. In this particular case, the fitted polynomial function turned out to be $\hat{y}(x) = -0.2954 + 0.9997x$. Figure 2-1 shows the data (plus markers), fitted function (blue line), and the true function (red line). By visual inspection of the figure, the fitted function seems more accurate than the data. This can be checked numerically by calculating the errors of the fit at the data points using either the root-mean-squared (RMS) error or the maximum error. The following equation is used for the RMS error:

$$e_{RMS} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y(x_i) - \hat{y}(x_i)\right)^2} \tag{2.1}$$

where $y(x_i)$ is the value of the true function at x_i, and $\hat{y}(x_i)$ is that of the fitted function. The RMS error of data can be calculated by replacing $\hat{y}(x_i)$ with y_i. For the given example, e_{RMS} of data was 0.9367, which is close to the standard deviation of noise $\sigma_{noise} = 1$. On the other hand, e_{RMS} of the fitted function was 0.3006. Therefore, the fitted function is more accurate than the data. This happened because the fitting process is based on minimizing the RMS error between the data and the fitted function. When the functional form is accurate, regression serves to filter out noise with dense data.

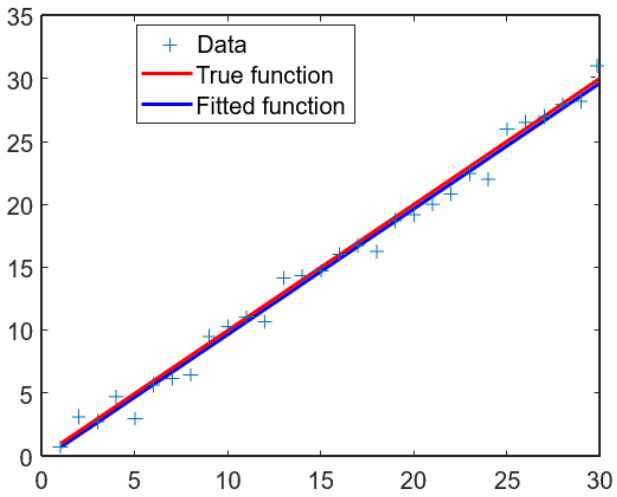


Figure 2-1: Fitting noisy data with a linear polynomial function.

Since the synthetic data are generated by adding a random noise $\sim N(0, 1^2)$ to the true function, it is questionable why the fitted function is not exact if regression is supposed to filter out noise. This can be explained by the imperfection of the noise samples. Even if the noise samples are randomly generated from the population distribution, $N(0, 1^2)$, the actual samples may not follow it. In fact, the mean of noise samples was $\mu_{noise} = -0.3005$. This is the culprit of $e_{RMS} = 0.3006$ of the fitted function. That is, the fitted function passes through the mean of the noise. If the same fitting process is repeated with the zero-mean noise, `noise = noise - mean(noise)`, e_{RMS} of data is close to the standard deviation of the samples and e_{RMS} of the fitted function is close to zero. Since the distribution of samples converges to that of the population as the number of samples increases, it is expected that the fitted function becomes accurate as the number of samples increases.

In this elementary example, it is assumed that the true function $y = x$ is knowns and synthetic data are generated by adding random noise. In reality, however, the true function is unknown and only data are given without knowing the distribution of noise. Therefore, the RMS error of the data and that of the fitted function are not available. The only available information is the error between the data and the fitted function. For the given samples, e_{RMS} between the data and the fitted function was 0.8872, which is the combined effect of the noise and the error in the fitted function. This happened to be smaller than e_{RMS} of data because the mean of noise was negative. This discrepancy would be reduced as the mean of noise converges to zero, which can happen as the number of samples increases.

The fact that the fitted function converges to the true function and e_{RMS} converges to the standard deviation of the noise is true only when the functional form of the fitted function is the same as that of the true function. The difference in the functional form between the true function and the fitted function is called the model-form error. When a model-form error is present, the fitting process treats the model-form error as if it is noise because there is no way to distinguish the model-form error from the noise. Therefore, it is critically important to choose a proper model form for successful curve fitting.

In order to investigate the effect of model-form error, 31 equally-spaced samples are generated in $x \in [0, 3]$ from a true function of $y(x) = x^2$. The measurement environment is simulated by adding normally distributed noise $\sim N(0,0.3^2)$ to the samples. The bias in the noise samples is removed by making the mean of noise samples to be zero. These samples are fitted using a linear PRS with two unknown coefficients, $\hat{y}(x) = a_0 + a_1 x$ using `polyfit` function in Matlab. In this particular case, the fitted function turned out to be $\hat{y}(x) = -1.4131 + 2.9754x$. Figure 2-2 compares the data samples (plus markers), the true function (red curve), and the fitted function (blue curve). It is clear that the regression process determines its coefficients such that the fitted linear function approximates the quadratic trend the best. Under the presence of model-form error, however, the error metrics can be difficult to explain. For example, e_{RMS} of data is 0.2932, which is close to the standard deviation of noise, $\sigma_{noise} = 0.3$. However, e_{RMS} of the fitted function is 0.7148. Therefore, under the presence of model-form error, it is possible that the data can be more accurate than the fit. Without knowing the true function, e_{RMS} between the data and the fitted function is 0.7963, which is a combined effect of model-form error and noise in data.

```
noise=0.3*randn(1,31); noise=noise-mean(noise);
x=linspace(0,3,31); ytrue=x.^2; y=ytrue+noise;
[p,s]=polyfit(x,y,1);
yfit=polyval(p,x);
plot(x,y,'+',x,ytrue,'r',x,yfit,'b');
legend('Data','True function','Fitted function');
RMS_data=sqrt(sum((ytrue - y).^2)/31)
noise_std=std(noise)
RMS_fit=sqrt(sum((ytrue - yfit).^2)/31)
RMS_disc=sqrt(sum((y - yfit).^2)/31)
```

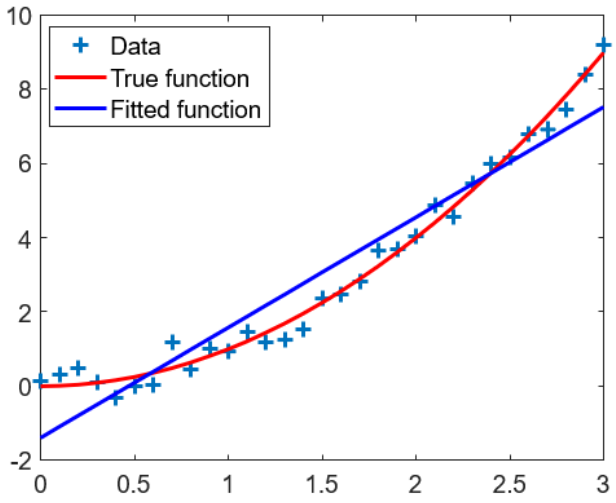


Figure 2-2: Fitting data with model-form error.

Even if it is difficult to identify the model-form error without knowing the true function, there is still a way to understand the presence of model-form error. Since the nature of the noise is random, the errors

between the samples and the fitted function are supposed to be randomly distributed. In Figure 2-2, for example, the errors between the data and the true function are indeed randomly distributed. Some errors are positive, while others are negative, and their locations are well-mixed. On the other hand, however, the errors between the data and the fitted function are grouped. That is, most errors are positive when $x \leq 0.5$ or $x \geq 2.5$, while they are mostly negative when $0.5 \leq x \leq 2.5$. This indicates that the fitted function underestimates the data in the ranges of $x \leq 0.5$ and $x \geq 2.5$, while overestimate in the range of $0.5 \leq x \leq 2.5$. Therefore, when a cluster of errors is in the same sign, it is a good indicator that there is a model-form error in the fitted function.

The major source of error in the fitted function in Figure 2-2 is due to the fact that a lower-order polynomial is used for the fitted function. However, increasing the degree of the polynomial fit using `polyfit` does not always result in a better fit. This is because high-order polynomials can be oscillatory between the sample points, leading to a poorer prediction capability between samples. A good example was shown in Figure 1-11, where a quadratic polynomial fits better than a quintic polynomial even if both models pass through all samples. If the level of error is similar, a lower-order polynomial is considered better because it tends to be smoother between samples.

One interesting question would be that if a higher-order polynomial includes the lower-order polynomial, will the fitting results be the same? That is, consider the following quintic polynomial function:

$$\hat{y}^{(5)}(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 \tag{2.2}$$

The question is if samples are generated from a quadratic polynomial, will the higher-order coefficients, $a_3, a_4,$ and a_5, be zero after the fitting process? Unfortunately, the answer to this question is no because (a) since samples have noise, the quadratic polynomial function may have a larger error than the quintic polynomial function, and (b) in general more degrees-of-freedom can yield a smaller error in the fitting process. Therefore, it would be necessary to introduce a fitting strategy to penalize the higher-order polynomials against lower-order ones.

Polynomials are unbounded and oscillatory functions by nature. Therefore, they are not well-suited to extrapolating bounded data or monotonic (increasing or decreasing) data. The curve fitting process shown in Figure 2-2 only takes into account the accuracy within the range of samples. However, if the accuracy is measured beyond the range of samples, the error in PRS can be significantly increased. In Figure 2-2, for example, it can be easily estimated that the error between the true function and the fitted function can significantly increase when $x < 0$ or $x > 3$.

2.3.Linear regression

Selecting the sample locations in the design space where experiments are to be performed is possibly the most important part of obtaining a good approximation to QoI that may be used in optimization and uncertainty quantification. However, this sample selection turns out to be a difficult optimization problem itself and will be discussed in the next chapter. In this section, the second question of how to determine the unknown coefficients of PRS is discussed first.

Polynomial response surface

In the following explanations, the goal is to approximate the true function $y(\mathbf{x})$ using PRS. In order to determine the unknown coefficients, it is assumed that the functional form of PRS is already determined, and n_y numbers of samples are available. The samples are prepared in the form of pairs of input variables and output QoI: $(\mathbf{x}_i, y_i), i = 1, \cdots, n_y$, where $\mathbf{x}_i = \{x_{1i}, x_{2i}, \cdots, x_{ni}\}^T$ is the vector of input variables at ith sample and y_i is the measured/simulated QoI at $\mathbf{x}_i$; i.e., $y_i = y(\mathbf{x}_i)$. For example, in the case of the

cantilevered beam problem in Figure 1-2, the input variable is $\mathbf{x} = \{F, L, b, h\}^T$ with $n = 4$ and output QoI is $y = \sigma_{max}$.

We want to approximate the true QoI $y(\mathbf{x})$ using a polynomial function $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$, where $\boldsymbol{\beta} = \{\beta_1, \beta_2, \cdots, \beta_{n_\beta}\}^T$ is the vector of parameters that need to be determined. The relationship between the true function and approximate function can be given as

$$y(\mathbf{x}) = \hat{y}(\mathbf{x}, \boldsymbol{\beta}) + \epsilon(\mathbf{x}) \tag{2.3}$$

where $\epsilon(\mathbf{x})$ is the approximation error. In this book, we refer to $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$ as a surrogate model. Different surrogate models can be defined depending on how to define the functional form of $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$ and how to determine the unknown model parameters. The vector of unknown parameters $\boldsymbol{\beta}$ often does not have any physical meaning. Rather, we select a functional representation for $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$, with $\boldsymbol{\beta}$ representing some coefficients to be determined so as to fit the data well.

The goal of surrogate modeling is to determine the unknown coefficient $\boldsymbol{\beta}$ so that the approximation error $\epsilon(\mathbf{x})$ is minimized. Therefore, natural questions are (a) what is the form of the approximation function $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$ and (b) what measure is used to minimize the error $\epsilon(\mathbf{x})$. To address these questions, it is assumed that the true function $y(\mathbf{x})$ is unknown, but we can evaluate it as discrete points. In experiments, for example, even if the functional relationship between input variables and output QoI may not be known, we still can perform experiments to measure QoI by changing input variables. The same is true for complex numerical simulations, where multiple simulations can be performed by changing input variables. Therefore, the first step in surrogate modeling is to perform n_y numbers of experiments to obtain samples: $(\mathbf{x}_i, y_i), i = 1, \cdots, n_y$. Then, Eq. (2.3) can be written as each sample location as

$$y_i = \hat{y}(\mathbf{x}_i, \boldsymbol{\beta}) + e_i, \qquad i = 1, \cdots, n_y \tag{2.4}$$

where $e_i = \epsilon(\mathbf{x}_i)$ is the error at ith sample location. Then, the requirement of minimizing error $\epsilon(\mathbf{x})$ is relaxed to minimize the errors at the sample locations.

Two simple examples of the approximate function $\hat{y}(\mathbf{x}_i, \boldsymbol{\beta})$ are

$$\hat{y}(x, \boldsymbol{\beta}) = \beta_1 + \beta_2 x + \beta_3 x^2 \tag{2.5}$$

$$\hat{y}(x, \boldsymbol{\beta}) = \frac{\beta_1}{x + \beta_2} \tag{2.6}$$

The functional form in Eq. (2.5) is a quadratic PRS, while that of Eq. (2.6) is a rational function. It is noted that the approximate function in Eq. (2.5) is a linear function of coefficients, while that in Eq. (2.6) is a nonlinear function of coefficients. Identifying unknown coefficients that are in a linear relationship with the approximate function is called linear regression as in Eq. (2.5), while that in a nonlinear relationship is called nonlinear regression as in Eq. (2.6). In this chapter, we will only handle those functional forms that yield linear regression. In particular, when all basis functions are monomials, the surrogate model is referred to as a polynomial response surface (PRS).

In general, the vector of unknown coefficients, $\boldsymbol{\beta}$, is found by minimizing the error so that the approximate function is the best fit. As mentioned before, the approximation error is defined at every sample location. In order to determine the best fit, it is necessary to define a scalar measure of error from the individual errors at sample locations. The following three error measures can be used:

$$e_{RMS} = \sqrt{\frac{1}{n_y} \sum_{i=1}^{n_y} \left(y_i - \hat{y}(\mathbf{x}_i, \boldsymbol{\beta})\right)^2} \tag{2.7}$$

$$e_{av} = \frac{1}{n_y}\sum_{i=1}^{n_y}|y_i - \hat{y}(\mathbf{x}_i, \boldsymbol{\beta})| \tag{2.8}$$

$$e_{max} = \max_{n_y}|y_i - \hat{y}(\mathbf{x}_i, \boldsymbol{\beta})| \tag{2.9}$$

The RMS error is most popular because it is a smooth function of $\boldsymbol{\beta}$, but it tends to ignore small errors and emphasize large errors too much. The average error weighs small and large errors equally, while the maximum error considers the largest error only. The average error and maximum error are a non-smooth function of $\boldsymbol{\beta}$, which is the main reason that they are not used often. The errors in Eqs. (2.7)-(2.9) are in fact norms in mathematics. For example, they are L_2-norm, L_1-norm, and L_∞-norm, respectively. It is noted that the RMS error in Eq. (2.7) is different from the one in Eq. (2.1). The RMS error in Eq. (2.1) is the true error between the true function and surrogate model, while the one in Eq. (2.7) is the discrepancy between the samples and surrogate model. Unfortunately, the true RMS error cannot be found because we do not know the true function in most cases.

When the approximate function is given as a linear combination of unknown coefficients (e.g., the one in Eq. (2.5)), we can rewrite the surrogate model by focusing on the coefficients as

$$\hat{y}(\mathbf{x}, \boldsymbol{\beta}) = \sum_{i=1}^{n_\beta}\beta_i\xi_i(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T\boldsymbol{\beta} \tag{2.10}$$

where $\boldsymbol{\xi}(\mathbf{x}) = \{\xi_1(\mathbf{x}), \xi_2(\mathbf{x}), \cdots, \xi_{n_\beta}(\mathbf{x})\}^T$ is the vector of basis functions. For example, the quadratic polynomial function in Eq. (2.5) has three basis functions: $\xi_1(x) = 1, \xi_2(x) = x$, and $\xi_3(x) = x^2$. In general, the number of terms n_β and all basis functions are assumed to be given. Therefore, unknown coefficients, $\boldsymbol{\beta}$, are the only undetermined terms in Eq. (2.10).

The exact coefficients, $\boldsymbol{\beta}$, can be found when the number of sample points n_y goes to infinity. That is, when sample points are defined at all the points in the design space, which is impossible because the input variables are real numbers. With a finite number of samples, we can only find the estimate of $\boldsymbol{\beta}$, which will be referred to $\mathbf{b}$ in the following derivations. From the expression $\hat{y}(\mathbf{x}_i, \boldsymbol{\beta}) = \boldsymbol{\xi}(\mathbf{x}_i)^T\mathbf{b}$, Eq. (2.4) can be rewritten in matrix-vector notation as

$$\begin{Bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{n_y} \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_y} \end{Bmatrix} - \begin{bmatrix} \xi_1(\mathbf{x}_1) & \xi_2(\mathbf{x}_1) & \cdots & \xi_{n_\beta}(\mathbf{x}_1) \\ \xi_1(\mathbf{x}_2) & \xi_1(\mathbf{x}_2) & \cdots & \xi_{n_\beta}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_1(\mathbf{x}_{n_y}) & \xi_1(\mathbf{x}_{n_y}) & \cdots & \xi_{n_\beta}(\mathbf{x}_{n_y}) \end{bmatrix} \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_\beta} \end{Bmatrix} \tag{2.11}$$

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{b}$$

In Eq. (2.11), the $(n_y \times n_\beta)$ design matrix $\mathbf{X}$ is defined using the vectors of basis functions at all sample locations. Also, it is often called 'regressor' in statistics community.

Once the errors at sample locations are defined, the regression process determines the unknown coefficients, $\mathbf{b}$, by minimizing these errors. Since $\mathbf{b}$ is determined through the regression process, it is often referred to as regression coefficients. For that purpose, we need a scalar measure of error, instead of the vector of errors. Among the scalar measures of errors given in Eqs. (2.7)-(2.9), we will use the RMS error here. It is noted that the magnitude of e_{RMS} is not the focus her. Instead, the coefficients $\mathbf{b}$ that minimizes e_{RMS} is what we want. Using the definition of e_{RMS} in Eq. (2.7) and that of errors in Eq. (2.11), the RMS error is redefined as

$$e_{RMS} = \sqrt{\frac{1}{n_y}\sum_{i=1}^{n_y} e_i^2} = \sqrt{\frac{1}{n_y}\mathbf{e}^T\mathbf{e}} \tag{2.12}$$

Therefore, minimizing e_{RMS} is equivalent to minimizing $\mathbf{e}^T\mathbf{e}$. The square sum of errors can be expressed as

$$\mathbf{e}^T\mathbf{e} = (\mathbf{y} - \mathbf{X}\mathbf{b})^T(\mathbf{y} - \mathbf{X}\mathbf{b}) = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{b} - \mathbf{b}^T\mathbf{X}^T\mathbf{y} + \mathbf{b}^T\mathbf{X}^T\mathbf{X}\mathbf{b} \tag{2.13}$$

In the above equation, it is noted that $\mathbf{y}^T\mathbf{X}\mathbf{b} = \mathbf{b}^T\mathbf{X}^T\mathbf{y}$ because both are scalar.

The square sum of errors in Eq. (2.13) is a quadratic function of $\mathbf{b}$. Therefore, if the matrix $\mathbf{X}^T\mathbf{X}$ is positive definite, the function is convex, and a local minimum will be the global minimum. In elementary polynomial function with a single variable, this corresponds to a quadratic function ax^2 with $a > 0$. The minimum of a quadratic function can be found from the condition that the derivative (i.e., gradient) becomes zero. In the same way, the coefficient vector $\mathbf{b}$ that satisfies the first-order Kuhn-Tucker optimality condition will be the optimal coefficients:

$$\frac{d}{d\mathbf{b}}(\mathbf{e}^T\mathbf{e}) = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{0}$$

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y} \tag{2.14}$$

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Equation (2.14) is often referred to as the normal equation of linear regression equation, and $\mathbf{X}^T\mathbf{X}$ is referred to as the information matrix or moment matrix. The Solving Eq. (2.14) is computationally efficient because the dimension of the moment matrix is $(n_\beta \times n_\beta)$, and the number of unknown coefficient n_β is much smaller than the number of samples n_y. This can be considered as the major advantage of linear regression, where the global optimum coefficients can be obtained by solving a single matrix equation. In the case of nonlinear regression, a nonlinear optimization problem needs to be solved iteratively.

The normal equation has a unique solution $\mathbf{b}$ when the moment matrix $\mathbf{X}^T\mathbf{X}$ is positive definite. When monomial basis functions are used, the moment matrix is ill-conditioned for a large n_β, which means that the moment matrix is almost singular. To avoid some of the effects of the ill-conditioning, we can formulate the regression problem in a slightly different form. Consider the system of equations:

$$\mathbf{X}\mathbf{b} = \mathbf{y} \tag{2.15}$$

If we could solve this equation exactly, then from Eq. (2.13) we see that the RMS error will be zero. However, this system has n_y equations for n_β unknowns, with n_y in general larger than n_β, so that in general, we cannot find an exact solution. That is, any vector $\mathbf{b}$ will not satisfy Eq. (2.15) exactly, but instead there will be a vector of residuals (differences between the left-hand side and the right-hand side of the equation). The solution of the normal equation is the vector $\mathbf{b}$ that minimizes the sum of the squares of the residuals. However, instead of solving the normal equations, there are numerical methods, such as the QR decomposition, that solve for Eq. (2.15) directly for the least-squares solution, and these are usually more numerically stable than directly solving the normal equations. To improve numerical stability, it is also recommended to translate and scale all the variables so that each variable changes in the range $(-1, 1)$ or $(0, 1)$ [13].

Example 2-1

Using three samples $(x_i, y_i) = (0,0), (1,1), (2,0)$, fit a linear PRS $\hat{y}(x, \mathbf{b}) = b_1 + b_2 x$. Use different error metrics in Eqs. (2.7)-(2.9) and compare them.

Solution:

The three samples are applied to the linear PRS model to calculate the error in Eq. (2.11) as

$$\left.\begin{aligned} e_1 &= y(0) - b_1 - b_2 \cdot 0 \\ e_2 &= y(1) - b_1 - b_2 \cdot 1 \\ e_3 &= y(2) - b_1 - b_2 \cdot 2 \end{aligned}\right\} \rightarrow \underbrace{\begin{Bmatrix} e_1 \\ e_2 \\ e_3 \end{Bmatrix}}_{\mathbf{e}} = \underbrace{\begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}}_{\mathbf{y}} - \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix}}_{\mathbf{b}}$$

Then, the matrix and vector that are required in the regression equation in Eq. (2.14) can be obtained as

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{y} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$$

Therefore, the regression equation, $\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y}$, can be solved for regression coefficients as

$$\begin{cases} 3b_1 + 3b_2 = 1 \\ 3b_1 + 5b_2 = 1 \end{cases} \rightarrow \begin{cases} b_1 = \dfrac{1}{3} \\ b_2 = 0 \end{cases}$$

Therefore, the fitted linear PRS becomes a constant function; i.e., $\hat{y}(x, \mathbf{b}) = 1/3$. Figure 2-3(a) compares the fitted PRS with the three sample points when the RMS error is used. It is clear that the fitting process selected the coefficients such that the positive error and negative errors are equally distributed. At all sample locations, the errors are $e_1 = e_3 = -1/3$ and $e_2 = 2/3$, respectively. Then the RMS error becomes

$$e_{RMS} = \sqrt{\frac{1}{3}\left[\left(-\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2 + \left(-\frac{1}{3}\right)^2\right]} = 0.47$$

In general, it is difficult to find the PRS that minimizes the maximum error and the average error. For this simple example, however, it can be easily done. We expect that like the line that minimizes the RMS error, the lines that minimize the other two errors would be horizontal lines of the form $\hat{y}(x, \mathbf{b}) = b_1, 0 \leq b_1 \leq 1$. This is expected as samples are symmetric with respect to $x = 1$. To minimize the maximum error, it is obvious that we must have $b_1 = 0.5$ such that all errors are of the same magnitude. This results in a maximum error as

$$e_{max} = max(|0 - b_1|, |1 - b_1|, |0 - b_1|) = 0.5$$

In this case, this is also the average error and the RMS error because all three points have the same 0.5 error. To minimize the average error, we note that for the range $b_1 \in [0, 1]$, the average error will be

$$e_{av} = \frac{1}{3}(|0 - b_1| + |1 - b_1| + |0 - b_1|) = \frac{1 + b_1}{3}$$

The average error becomes minimum when $b_1 = 0$, at which the minimum average error becomes $e_{av} = 1/3$. Figure 2-3(b) compares all three linear PRSs. As can be seen in the figure, even if the same samples are used, the fitted surrogate models can be different with different error metrics.

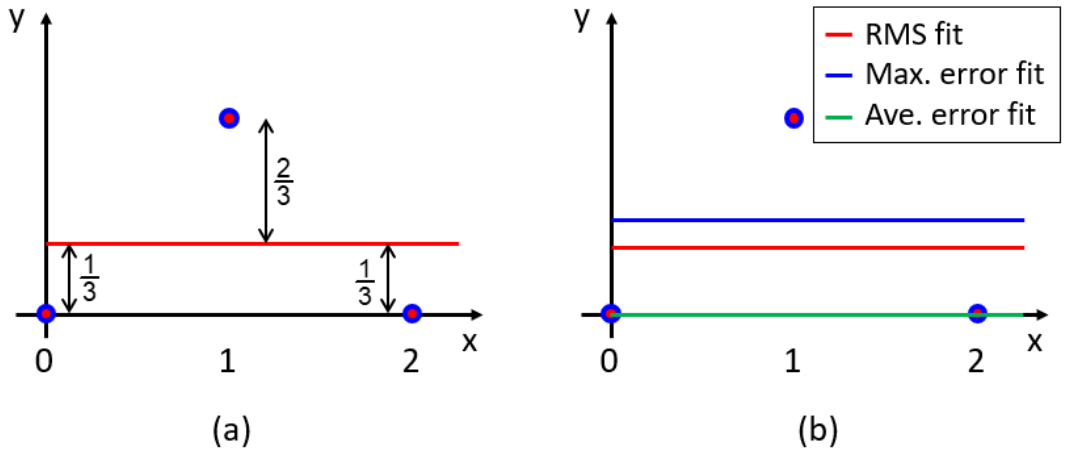


Figure 2-3: Fitting a linear PRS with three samples (a) with RMS error and (b) with average error and maximum error.

At first glance, it looks like the maximum and RMS error metrics yield a better fit than the average error metric, but the quality of a fit depends on the error metric used as well. For example, Table 2-1 evaluates the three PRS fits with the three different error metrics. It turns out that an error is a minimum when a surrogate is fitted with the same error metric.

Table 2-1: Comparison of the three linear PRS fits with the three different error metrics.

	RMS fit $\hat{y} = 1/3$	Ave. error fit $\hat{y} = 0$	Max. error fit $\hat{y} = 0.5$
e_{RMS}	0.471	0.577	0.5
e_{av}	0.444	0.333	0.5
e_{max}	0.667	1.0	0.5

Polynomial response surface in multi-dimension

So far, we only discussed a PRS in one-dimensional input variables. The `polyfit` and `polyval` Matlab functions in Section 2.2 can only be used for one-dimensional input variables with a fixed degree of polynomials. However, the linear regression is flexible enough such that it can be applied to multi-dimensional input variables with an arbitrary degree of polynomials. In fact, the basis functions do not have to be monomials. As long as the basis functions are linearly independent and the QoI has a linear relationship with the regression coefficients, the linear regression is well-defined. In Matlab, multiple linear regression function `regress` is used for this purpose.

Example 2-2

A soft-drink company wants to determine the size of a can such that it can maximize its profit. The dimensions of the can are determined by input variables: diameter $D \in [1.5, 3.5]$ in. and height $H \in [3.0, 7.0]$ in. The true profit is an explicit function of these two variables:

$$p(D,H) = 0.2361\pi D^2 H - 3.858 \times 10^{-4}\pi^2 D^4 H^2 - 0.1\pi(0.5D^2 + DH)$$

However, it is assumed that the true profit is unknown. Instead, the company conducted field tests and obtained nine samples as shown in the table.

Sample No.	D	H	p	p_{true}
1	1.8	3.6	5.9	5.6
2	2.4	4.8	13.1	13.1
3	3.0	6.0	22.0	21.9
4	1.5	4.6	4.7	4.7
5	2.1	5.8	12.0	12.0
6	2.7	7.0	20.9	20.9
7	1.4	5.6	4.9	4.9
8	2.0	6.8	12.5	12.5
9	2.6	8.0	21.4	21.4

(a) Fit a quadratic polynomial to the profit per can using the experimental nine samples and plot the profit contours with samples. (b) Repeat the fitting process using the true samples that are generated from the true profit equation (use the last column of the table). Compare the two PRS and explain the reason for the difference.

Solution:

The quadratic PRS with two input variables has the following form with six unknown coefficients:

$$\hat{p} = b_1 + b_2 D + b_3 H + b_4 D^2 + b_5 DH + b_6 H^2$$

By comparing with the true function, it is obvious that the quadratic PRS cannot be exact because the true function includes higher-order polynomial terms of input variables. And yet, it is difficult to use a higher-order PRS because then the number of unknown coefficients can be more than that of the number of samples. It is also possible that the PRS can include specific terms, such as D^2, DH, D^2H, and D^4H^2. However, it is only possible if the model form is already known.

The design matrix can be calculated using all nine samples as

$$\mathbf{X} = \begin{bmatrix} 1 & 1.8 & 3.6 & 3.24 & 6.48 & 12.96 \\ 1 & 2.4 & 4.8 & 5.76 & 11.52 & 23.04 \\ 1 & 3.0 & 6.0 & 9.00 & 18.0 & 36.0 \\ 1 & 1.5 & 4.6 & 2.25 & 6.9 & 21.16 \\ 1 & 2.1 & 5.8 & 4.41 & 12.18 & 33.64 \\ 1 & 2.7 & 7.0 & 7.29 & 18.9 & 49.0 \\ 1 & 1.4 & 5.6 & 1.96 & 7.84 & 31.36 \\ 1 & 2.0 & 6.8 & 4.00 & 13.6 & 46.24 \\ 1 & 2.6 & 8.0 & 6.76 & 20.8 & 64.0 \end{bmatrix}$$

(a) The unknown coefficients can be calculated from the regression equation in Eq. (2.14). Then the PRS approximate of the profit can be written as

$$\hat{p} = -7.1827 + 2.8835D + 0.3022H + 0.3576D^2 + 1.0024DH - 0.0707H^2$$

(b) In comparison, if the true values in the last column of the table are used, the PRS approximate becomes

$$\hat{p}_{true} = -7.9801 + 2.5672D + 0.6756H + 0.2956D^2 + 1.0898DH - 0.1150H^2$$

Note that four of the six coefficients are fairly close, whose difference is less than 10 percent of each other. However, the coefficients of H and of H^2 are quite different, indicating that these coefficients are less important for fitting the samples. This does not mean, however, that these coefficients may not affect predictions at other points besides the sample points. We are thus warned that these coefficients may need special treatment. The profit-per-can plot based on the experimental samples is shown in Figure 2-4: one

from a quadratic PRS using experimental samples and the other using the true samples. Both PRSs are obviously quite similar to each other. The following Matlab code is used for plotting Figure 2-4.

```
% Quadratic PRS for profit-per-can
x=[1.8 3.6; 2.4 4.8; 3.0 6.0; 1.5 4.6; 2.1 5.8; 2.7 7.0; 1.4 5.6; 2.0
6.8; 2.6 8.0];
p=[5.9 13.1 22.0 4.7 12.0 20.9 4.9 12.5 21.4]';
D=x(:,1); H=x(:,2);
ptrue=0.2361*pi*D.^2.*H-3.858E-4*pi^2*D.^4.*H.^2-0.1*pi*(0.5*D.^2+D.*H);
%
X=[ones(9,1) D H D.^2 D.*H H.^2];
A=X'*X;
Btrue=X'*ptrue;
B=X'*p;
btrue=A\Btrue;
b=A\B;
%
d=linspace(1.5,3.5,10);
h=linspace(3.0,7.0,10);
[D,H]=meshgrid(d,h);
P=b(1)+b(2)*D+b(3)*H+b(4)*D.^2+b(5)*D.*H+b(6)*H.^2;
Ptrue=btrue(1)+btrue(2)*D+btrue(3)*H+btrue(4)*D.^2+btrue(5)*D.*H+btrue(6)
*H.^2;
surf(D, H, P); hold on; surf(D, H, Ptrue);
```

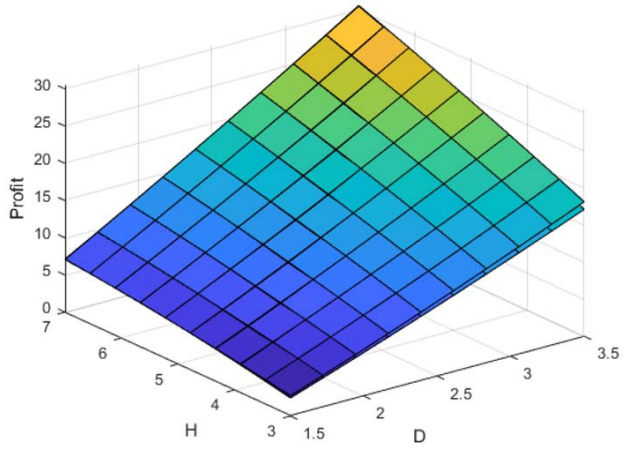


Figure 2-4: Quadratic polynomial response surface model for the profit-per-can.

Curse of dimensionality

As shown in the previous example, the number of regression coefficients for the quadratic PRS is six for two-dimensional input variables. The number of regression coefficients rapidly increases as the order of PRS increases and the dimension of the input variable increases. Considering the fact that the required number of samples is about two- or three-fold of the number of regression coefficients, the required number of samples rapidly increases along with the dimension of input variables and the order of PRS, which is referred to as the curse of dimensionality. In the case of a quadratic and a cubic PRS, the numbers of regression coefficients are, respectively,

$$n_\beta^{(2)} = \frac{1}{2}(n+1)(n+2) \tag{2.16}$$

$$n_\beta^{(3)} = \frac{1}{6}(n+1)(n+2)(n+3) \tag{2.17}$$

where n is the dimension of input variables. For example, when $n = 10$, $n_\beta^{(2)} = 66$ and $n_\beta^{(3)} = 286$. Therefore, in 10-dimensional input variables, we need more than 800 samples to fit the PRS, which becomes impractical for many applications with expensive simulations or experiments.

Although the rapid growth of the number of required samples is a part of the curse of dimensionality, the major difficulty comes from the fact that when the dimensionality increases, the volume of the design space increases so fast that the available samples become sparse. In order to obtain a reliable result, the number of samples needed often grows exponentially with dimensionality. For example, let us assume that all input variables are normalized by a unit length. If we generate 10 evenly spaced samples in a one-dimensional domain, then the distance between the samples would be 0.1. An equivalent sampling of a 10-dimensional unit hypercube with a lattice that has a spacing of 0.1 between adjacent samples would require 10^{10} sample points, which is practically impossible to use. Therefore, in a high-dimensional design space, samples are sparse and the distances between samples are quite large. When the distance between samples is large, it would be difficult to capture the trend of the function accurately. In addition, sparse sampling inevitably causes a large extrapolation region as shown in Figure 1-5, where the volume covered by samples (interpolation region) is dramatically reduced as the dimension increases.

Based on the above discussion, it would be challenging to build an accurate and reliable surrogate model with more than 10 design variables. Unfortunately, it is common that many engineering applications have more than 10 design variables. Instead of trying to make a surrogate model that is accurate in high-dimensional input variables with a small number of samples, it would make more sense to reduce the number of variables. It is hard to imagine that a QoI strongly depends on many variables simultaneously; such a system would be very sensitive and volatile for a small change in variables. Instead, many engineering systems strongly depend on only a handful number of variables. Therefore, it would make sense to identify those design variables that significantly affect the QoI and fix all other variables that are not significant. Then, the surrogate model is built using only those significant variables. This process of identifying significant input variables and fixing insignificant variables is called sensitivity analysis [14]. If 100 samples are affordable, it would be better to build an accurate surrogate with four significant input variables, rather than building an inaccurate surrogate with all ten variables.

It is also possible to combine multiple variables together instead of using individual variables, which is called dimensional analysis. A good example is nondimensionalization in fluid mechanics, where multiple variables are combined together to describe a physically important phenomenon together [15]. Another method of dimension reduction is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Principle component analysis [16] or proper orthogonal decomposition (POD) is one of dimension reduction, which performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance matrix of the samples is constructed and the eigenvectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data.

Assumptions in linear regression

Although the process of regression analysis in Eq. (2.14) is simple, the accuracy analysis of regression is quite comprehensive. Before we discuss regression accuracy, it would be useful to summarize the assumptions in the regression analysis.

(1) The regression analysis assumes that the input variables can be fully controllable; that is, we can select the input variables without any errors associated with them. This might sound obvious but in practice, it could be hard to control the input variables precisely during experiments. When a sample is given in the form of $(\mathbf{x}_i, y_i)$, the regression analysis assumes that there is random noise in y_i, but $\mathbf{x}_i$ is precise.

(2) The functional form of PRS is a linear combination of unknown coefficients and basis functions. Therefore, the only limitation is the relationship between the QoI and unknown coefficients is linear. This is flexibility rather than limitation. Because of this flexibility, PRS often shows "too much power", in that it tends to overfit the data. Using a too-small number of samples, too many basis functions and/or too-high order of monomial basis functions can cause overfitting. There are several techniques available to prevent overfittings, such as ridge regression, lasso regression and Bayesian linear regression.

(3) The noise in each sample is independent. More specifically, the noise in samples is statistically uncorrelated. For example, the noise in two samples is nothing related to the distance between the two samples. In fact, the linear regression allows to have samples at the same location; i.e., $\mathbf{x}_i = \mathbf{x}_j$, but the output QoI samples may not be the same; i.e., $y_i \neq y_j$ because the noise in the two samples is independent. When we generate noise samples from $\sim N(0, \sigma^2)$, we automatically apply the independence assumption.

(4) The variance of the noise is assumed to be constant throughout the entire design space. That is, the noise is normally distributed with a zero mean and a constant standard deviation σ. Based on this assumption, minimizing the RMS error can provide an unbiased fit. This seems a reasonable assumption, but in practice, the noise in experiments often depends on the value of measurement. When the QoI does not vary rapidly over the design space, this assumption can be reasonable. When the QoI varies over orders of magnitude, however, this assumption may not represent the reality. For example, when we measure stress in a component, it might be reasonable to say that the measurement error is 10MPa when the level of stress is 500MPa, which corresponds to a 2% error. However, the same measurement error of 10MPa may not be acceptable when the level of stress is 10MPa because that would imply the actual stress could vary anywhere between 0 and 20 MPa. In many engineering experiments, the measurement noise is often proportional to the magnitude of data. In such a case, it makes more sense to assume that the coefficient of variation (CoV) remains the same, not the standard deviation. If we want to check if samples satisfy this assumption, the residuals at all sample locations can be plotted. If residuals are randomly scattered about the horizontal midline at 0 with a similar bound, it means that this assumption is satisfied. If samples do not satisfy this assumption, it will result in an overall "average" estimate of variance being used instead of one that takes into account the true variance structure. This leads to less precise parameter estimates and biased standard errors, resulting in misleading tests and interval estimates.

(5) The moment matrix $\mathbf{X}^T\mathbf{X}$ is positive definite such that the regression analysis yields a unique solution $\mathbf{b}$. In order to have a positive definite matrix $\mathbf{X}^T\mathbf{X}$, the design matrix $\mathbf{X}$ must have a full column rank of n_β. Rank deficiency can happen when a linear relationship exists between two or more basis functions or when a smaller number of samples are used than the number of coefficients to be estimated. It is noted that monomial basis functions are linearly independent. In order to prevent a near singular matrix, it is recommended that the number of samples should be two or three folds larger than that of the coefficients.

2.4. Goodness of fit

The goal of the surrogate fitting process, in particular the linear regression in the previous section, is to find the best fit that can approximate the true function. In order to determine the quality of the approximation, it would be necessary to define the measure of goodness. In the case of the curve fitting in Section 2.2, the RMS error at sample locations was used as an error measure, which was minimized to determine regression coefficients. As shown in Section 2.3, different error measures would yield different fitting results. Different from the error measures to fit a surrogate, the measure of goodness can be considered post-processing in a way. After fitting a surrogate by minimizing an error measure, the measure of goodness characterizes the fidelity of the surrogate for predicting the behavior in future simulations. It is possible that a measure can be local at a point or in a region, but in this section, we limit ourselves to global measures, which are a single number that characterizes the overall fidelity of the surrogate model in the entire design space.

However, achieving this goal is often doomed because of a simple reason: we do not know the true function, and therefore, we cannot compare the accuracy of the surrogate model against the true function. Although samples represent the true function, as we discussed in the previous section, they include random noise (or, at least we assume it). Therefore, fitting the samples exactly does not guarantee that the surrogate model is accurate compared to the true function. A good example is the one shown in Figure 1-8, where the quintic polynomial passes through all samples with zero error, but it is not an accurate surrogate model except for the sample locations. Therefore, the biggest challenge in assessing the accuracy of a surrogate model is how to evaluate its accuracy without knowing the true function. All information that we have (or we assume) is that the noise is normally distributed with respect to the true function, but we still do not know the variance of the distribution. Therefore, the goal of assessing the accuracy of a surrogate model to the true function is difficult to achieve.

Instead, it is possible to evaluate the accuracy of a surrogate model against the samples. That is, how well the surrogate fits the samples, which is referred to as ‘goodness of fit.’ In evaluating the goodness of fit, it is important to remember that errors are minimized at sample locations during the fitting process. For example, if the number of samples is the same as the number of regression coefficients; i.e., $n_y = n_\beta$, theoretically, it is possible that the regression equation can be solved such that the surrogate model can pass through all sample points. That is, the errors at all sample points are zero. However, this does not mean that the surrogate model is accurate at unsampled points. Therefore, the goodness of fit does not mean simply measuring the error at sample points.

In general, the goodness of fit is evaluated in two ways: (a) equivalence between the surrogate and samples and (b) prediction accuracy of the surrogate. Equivalence measures include the coefficient of multiple determination and the adjusted coefficient of multiple determination. They measure the equivalence between the surrogate and the samples in terms of variability. The first provides the fraction of the variability in the samples captured by the surrogate. The second adjusts it in an attempt to estimate the fraction that will be captured by using the surrogate to predict values at other points. Good fidelity will be reflected in these coefficients being close to one. The second category of prediction accuracy measures estimates what will be the RMS error in predictions based on the surrogate and includes cross-validation error and standard error. The cross-validation error is a measure that can be applied to any surrogate, while the standard error applies only to linear regression with specific assumptions on the noise in the samples. A surrogate model is considered to be good when these error measures are small compared to the average value of the samples.

Estimation of noise in samples

In most applications, the PRS approximation we construct based on given data is intended for the prediction of QoI at other design points, typically for improving the design. Therefore, the ultimate test of the PRS is how well it predicts the QoI at other points of interest. However, if the PRS does not approximate the response well even at the data points, we cannot expect it to approximate well other points in the design space. The most immediate measures of the accuracy of the fit to the samples are the various errors discussed earlier, the RMS error, the average error and the maximum error.

Since the samples are assumed to include both the true function value and noise, it is important to estimate the level of noise accurately so that the surrogate can approximate the true function, not the noise. After fitting a surrogate $\hat{y}(\mathbf{x}, \boldsymbol{\beta}) = \boldsymbol{\xi}(\mathbf{x})^T \mathbf{b}$, the model predictions at sample points become $\hat{y}(\mathbf{x}_i, \boldsymbol{\beta}) = \boldsymbol{\xi}(\mathbf{x}_i)^T \mathbf{b}, i = 1, \cdots, n_y$. These n_y predictions can be written in matrix-vector notation as $\hat{\mathbf{y}} = \mathbf{X} \cdot \mathbf{b}$, and the regression errors at the sample points become $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$. Then, the RMS error in Eq. (2.7) can be rewritten as

$$e_{RMS} = \sqrt{\frac{SSe}{n_y}} \tag{2.18}$$

where the square-sum-error is defined as $SSe = \mathbf{e}^T\mathbf{e} = \mathbf{y}^T\mathbf{y} - 2\mathbf{b}^T\mathbf{X}^T\mathbf{y} + \mathbf{b}^T\mathbf{X}^T\mathbf{X}\mathbf{b}$. However, this calculation of RMS error is quite misleading if we wanted to use it to assess the accuracy of PRS. This becomes clear if we note that the number of sample points n_y is equal to the number of coefficients, n_β, then the PRS will pass through the sample points, and the error will be zero. We certainly do not expect that the error will be zero at other points, not included in the data. In fact, fitting a PRS to the same number of points equal to the number of coefficients (known as saturated design) is known to often lead to poor approximation, especially when there is noise in the data.

An impressive body of theoretical work has been done for the case where the noise in the samples is random with normal distribution with zero mean and standard deviation of σ, and where the noise at one point is uncorrelated with the noise at other data points (e.g., Myers and Montgomery [4]).

Since the surrogate fitting process minimizes the RMS error, it is minimum at sample points but underestimates the error at unsampled (prediction) points. Therefore, it is not a good measure to assess the accuracy of the surrogate model. However, the square-sum-error can be used to estimate the variance of noise in samples. The variance of random samples y_i is defined as

$$Var = \frac{1}{n_y - 1}\sum_{i=1}^{n_y}\left(y_i - \mu_y\right)^2 \tag{2.19}$$

where μ_y is the mean of samples, and the denominator $n_y - 1$ is used for unbiased variance, which represents the number of degrees-of-freedom. It is reminded that PRS assumes that the model is accurate but the samples have a random noise. The surrogate $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$ is considered a mean prediction, and the difference between the samples and the mean predictions is considered noise. Therefore, in the viewpoint of noise, its mean is zero, and the unbiased estimate of the variance of the noise from the samples is

$$\hat{\sigma}^2 = \frac{SSe}{n_y - n_\beta} \tag{2.20}$$

The square root of Eq. (2.20), $\hat{\sigma}$, is called the standard error of noise.

Since the surrogate model has n_y samples with n_β model parameters, the denominator $n_y - n_\beta$ is represents the number of degrees-of-freedom. If this estimate of the RMS error is larger than we can tolerate for predicting values of the response at candidate design points, we conclude that the PRS is inadequate. In this case, we must change the form of the response surface to try to fit the samples better.

The simplest approach is to add terms to the polynomial approximation. If we used a linear polynomial to start with, we may want to go to a quadratic. If we used a quadratic, we may want to use a cubic.

Unfortunately, while we can always improve the fit to the samples by increasing the number of terms in the PRS, it is not clear that these gains will translate into gains in predicting the PRS at other points. As we add coefficients, we run the danger of 'overfitting' the samples. This danger is particularly acute when the samples contain a substantial amount of noise. As we increase the number of coefficients, we may be fitting the noise rather than the underlying response. This danger is captured by Eq. (2.20). As we add more terms, we expect to decrease the numerator, but the denominator will also decrease.

There are several assumptions under the noise estimate in Eq. (2.20). Firstly, it is assumed that the true function is described by the model form of the surrogate. If there is a model form error, it is embedded in the estimated noise. Secondly, the samples include noise that is normally distributed with a zero mean and the same standard deviation at every sample. Lastly, the noises in different samples are not correlated. Under these assumptions, the standard error, $\hat{\sigma}$, is an estimate of the standard deviation of the noise. The standard error will be used to estimate the prediction error in Section 2.6. That is the error between the true function and the surrogate prediction.

Example 2-3

The following five equally-spaced samples are generated from a true function $y = x$ and added noise $\sim N(0,1^2)$.: $(x_i, y_i) = (0, 1.5326), (2.5, 1.7303), (5.0, 5.3714), (7.5, 7.2744), (10.0, 11.1174)$. Fit the samples using a quartic PRS (5 coefficients) and see if the surrogate fits the trend or the noise.

Solution:

The five samples can be fitted using `polyfit` Matlab function. Since the quartic PRS has five coefficients, the number of samples and the number of coefficients are the same; i.e., $n_y = n_\beta$. Therefore, the regression equation is equivalent to solving a linear system of equations. In this case, the linear system of equations has a unique solution that makes the errors zero. The fitted quartic PRS becomes

$$\hat{y}(x) = 1.5326 - 2.1864x + 1.3397x^2 - 0.1970x^3 + 0.0095x^4$$

Figure 2-5 compares the quartic PRS with samples and the true function. It is noted that the quartic PRS passes through all samples with zero errors. However, there is a significant error at unsampled locations. In this case, the PRS fits noise rather than the trend of the true function. This happens when the number of samples is not enough to determine the mean of the trend of samples. In order to have a meaningful regression performance, it is often recommended that the number of samples should be two- or three-fold larger than that of the regression coefficients. The following Matlab code can be used to plot Figure 2-5.

```
x=[0 2.5 5 7.5 10]';
y=[1.5326 1.7303 5.3714 7.2744 11.1174]';
xp=0:0.5:10; ytrue=xp;
[b,s]=polyfit(x,y,4);
yfit=polyval(b,xp);
plot(x,y,'+',xp,ytrue,'r',xp,yfit,'b');
legend('Samples','True function','Fitted function');
```

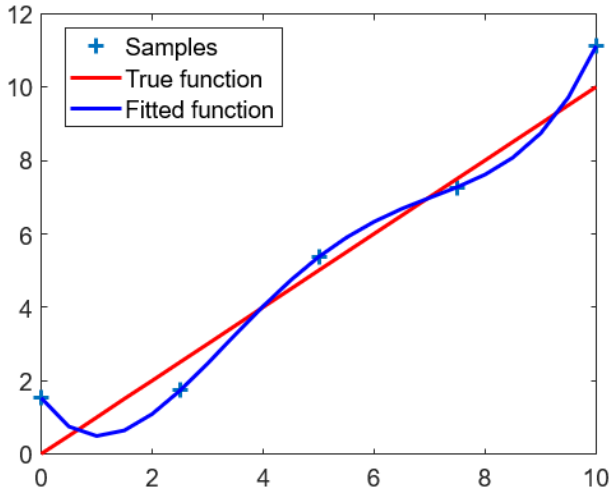


Figure 2-5: Fitted quartic polynomial response surface of noisy samples from $y = x$.

As mentioned before, the PRS assumes that the model form is correct, but samples have noise. Because of this assumption, the estimate of noise variance in Eq. (2.20) is reasonable when the model form is correct. If the model form has an error, it will be included in the estimated noise variance. The following example shows how the model form error can contribute to the noise variance estimate.

Example 2-4

Generate 20 equally-spaced samples in $x \in [0, 10]$ from a true function $y = x^2$and added noise $\sim N(0,1^2)$. Fit the samples using linear and quadratic PRS and estimate the noises from the two surrogates.

Solution:

In order to prevent different random samples, `rng default` Matlab command is used, which will reset the random number generator sequence. In addition, the noise samples are shifted to have a zero mean by the following command: `noise=noise-mean(noise)`. In this specific case, the standard deviation of noise samples is 1.4797, which is different from the population distribution $\sim N(0,1^2)$. The two fitted surrogate models become

$$\hat{y}_L(x) = -16.2473 + 10.0916x$$

$$\hat{y}_Q(x) = -0.5748 + 0.1657x + 0.9926x^2$$

The estimated standard deviations from the two surrogate models are, respectively,

$$\hat{\sigma}_L = \sqrt{\frac{SSe_L}{20-2}} = 8.7153, \quad \hat{\sigma}_Q = \sqrt{\frac{SSe_Q}{20-3}} = 1.5336$$

Compared to the actual noise standard deviation of 1.4797, the estimate from the quadratic PRS is close to the actual one, but the estimate from the linear PRS is quite different from the actual one. This is because the linear PRS has a significant level of model error. Figure 2-6(a) shows the two PRS predictions. The quadratic PRS follows the trend of samples well, while the linear PRS approximates the trend linearly. As shown in Figure 2-6(b), the errors of the quadratic PRS are randomly distributed with respect to the zero

line, while the errors of the linear PRS are all positive in $x < 2$ or $x > 8$ and all negative in $2 < x < 8$. This is good indicator of the presence of model form error. The following Matlab code is used to fit the surrogate and plot the results.

```
rng default;
x=linspace(0,10,20);
ytrue=x.^2;
noise=randn(1,20); noise=noise-mean(noise);
y=ytrue+noise;
bL=polyfit(x,y,1);
yL=polyval(bL,x);
sigmaL=sqrt((y-yL)*(y-yL)'/(20-2))
bQ=polyfit(x,y,2);
yQ=polyval(bQ,x);
sigmaQ=sqrt((y-yQ)*(y-yQ)'/(20-3))
plot(x,y,'+',x,x.^2,'r',x,yL,'b',x,yQ,'k');
legend('Samples','True function','Linear PRS','Quadratic PRS');
figure(2);
plot(x,y-yL,'+',x,y-yQ,'o');
legend('e_L','e_Q');
```

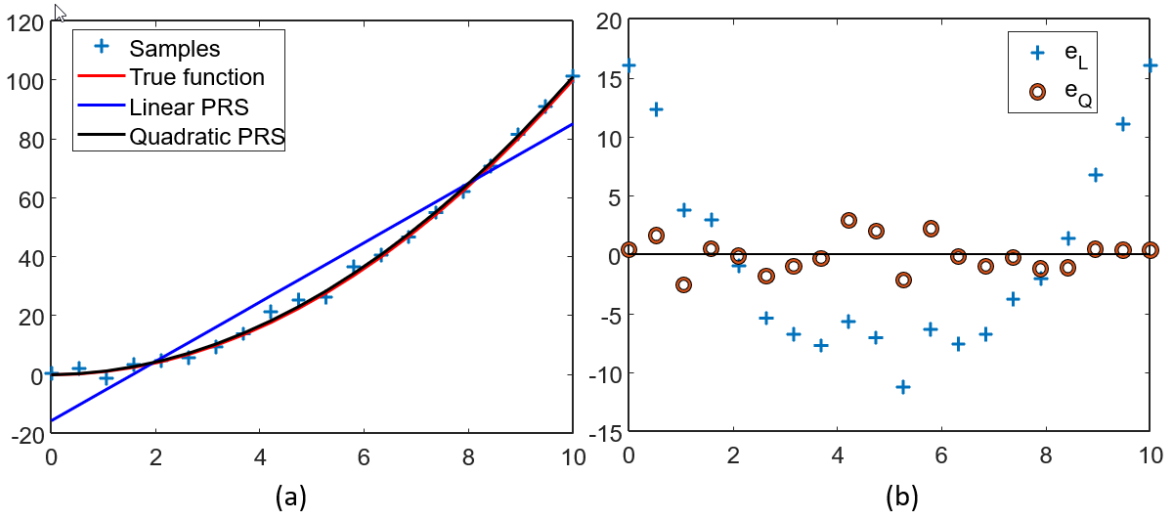


Figure 2-6: Fitted linear and quadratic polynomial response surface of noisy samples from $y = x^2$.

Coefficient of multiple determination

As mentioned before, the accuracy at sample points is different from the accuracy at prediction points. Therefore, instead of measuring accuracy at sample points, it would be possible to measure the equivalence between the samples and PRS predictions in terms of variability. That is how well a surrogate captures the variability in samples. The coefficient of multiple determination (R^2) is a numerical index that reflects the degree to which variation in the samples is accounted for by the predictions. If a surrogate passes through all sample points, then the variability of the predictions at sample points would be the same as that of the samples. In that case, the entire variability in the samples is explained by the variability of the model predictions. On the other hand, if the model prediction is a constant at the mean of the samples, then none of the variability of the samples can be explained by the model predictions. In such a case, the entire variability becomes the square-sum-error SS_e in Eq. (2.18).

Since we do not know the true function, the coefficient of multiple determination is defined as variation with respect to the mean of samples, which is defined as

$$\bar{y} = \frac{1}{n_y}\sum_{i=1}^{n_y} y_i \tag{2.21}$$

The vector version of the mean can be defined as $\bar{\mathbf{y}} = \{\bar{y}, \bar{y}, \cdots, \bar{y}\}^T$ whose dimension is $n_y \times 1$. The variation of samples from the mean is defined as

$$SS_y = \sum_{i=1}^{n_y} (y_i - \bar{y})^2 \tag{2.22}$$

This is also known as the total variation. Similarly, the variation of the PRS $\hat{y}$ from the same mean is defined as

$$SS_r = \sum_{i=1}^{n_y} (\hat{y}_i - \bar{y})^2 \tag{2.23}$$

where $\hat{y}_i = \hat{y}(\mathbf{x}_1)$. This is also known as the explained variation. That is, the variation can be explained by the fitted model. If samples were generated by adding random noise to a true function and if the fitted PRS model is close to the true function, then SS_y is larger than SS_r because of the variation from the random noise in the samples. In fact, the variation from the random noise corresponds to the sum of square errors in Eq. (2.18), which is also known as the unexplained variation. In linear regression, these three variations have the following relationship:

$$SS_y = SS_r + SS_e \tag{2.24}$$

That is, the total variation in samples is the sum of the explained variation by the model and the unexplained variation by noise. The relationship in Eq. (2.24) holds if and only if $\mathbf{y}^T\bar{\mathbf{y}} = \hat{\mathbf{y}}^T\bar{\mathbf{y}}$. Since $\bar{y}$ is a constant, the condition means that the sum of samples equals to the sum of predictions at sample points, or equivalently, the sum of residuals $e_i = y_i - \hat{y}_i$ is zero. Since the first column of the design matrix $\mathbf{X}$ is all ones, the first element of $\mathbf{X}^T\mathbf{e}$ is the sum of residuals. From the regression analysis, it can be shown that

$$\mathbf{X}^T\mathbf{e} = \mathbf{X}^T[\mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]\mathbf{y} = [\mathbf{X}^T - \mathbf{X}^T]\mathbf{y} = \mathbf{0} \tag{2.25}$$

Therefore, the condition $\mathbf{y}^T\bar{\mathbf{y}} = \hat{\mathbf{y}}^T\bar{\mathbf{y}}$ holds for linear regression of PRS. The proof of Eq. (2.24) is left as an exercise problem.

The ratio of Eq. (2.23) over Eq. (2.22), denoted by R^2, measures the fraction of the variation in the samples is captured by the PRS:

$$R^2 = \frac{SS_r}{SS_y} = 1 - \frac{SS_e}{SS_y} \tag{2.26}$$

R^2 is a measure of the goodness of fit of a PRS model [17], often interpreted as the proportion of sample variation "explained" by the model prediction. If the fitted model $\hat{y}$ comes from linear regression, it is expected that $0 \le R^2 \le 1$. An $R^2 = 1$ indicates that the fitted model explains all variability in samples $\mathbf{y}$, while $R^2 = 0$ indicates no 'linear' relationship between the samples and predictions. An interior value such as $R^2 = 0.7$ means that 70% of variability in the samples may come from the surrogate predictions and the remaining 30% may come from the noise in samples.

It is interesting to note that R^2 monotonically increases as the PRS includes additional basis functions; i.e., increasing the number of coefficients n_β. This can be considered a major drawback of R^2

to be used as a criterion for assessing a goodness-of-fit of a surrogate model. A good example was shown in Figure 2-5, where the order of PRS is increased to a quintic polynomial, the PRS passes through all the samples, and thus, $R^2 = 1$. However, this does not mean that the PRS is accurate in prediction. Therefore, R^2 may not represent the accuracy of a surrogate especially when the number of coefficients is close to the number of samples.

This leads to the alternative approach of looking at the adjusted R^2, namely R_a^2, which penalizes R^2 as extra basis functions are included in the model. The penalization is based on the degrees-of-freedom of the estimate of the variance around the mean. First, the square-sum-error SS_e is penalized by the degrees-of-freedom $n_y - n_\beta$ because it uses n_y samples but n_β coefficients have been used. In the same way, the variance of samples SS_y is penalized by the degrees-of-freedom $n_y - 1$ because the mean is fixed. After using these penalizations on the original definition of R^2, the adjusted coefficient of multiple determination R_a^2 is given as

$$R_a^2 = 1 - \frac{SS_e/(n_y - n_\beta)}{SS_y/(n_y - 1)} = 1 - (1 - R^2)\left(\frac{n_y - 1}{n_y - n_\beta}\right) \tag{2.27}$$

If the adjusted value, R_a^2, decreases as we increase the number of coefficients, it is a warning that we may be fitting the samples better, but losing predictive capability.

R_a^2 can be negative and always less than or equal to R^2. When adding an additional basis, R_a^2 increases only when the basis contributes to the prediction accuracy. Therefore, R_a^2 can be a good indicator if an additional basis should be introduced or not. Let us assume that a series of additional basis functions are available in the order of their significance. Then each basis function is introduced in the linear regression and calculate R_a^2 each time. The level at which R_a^2 reaches a maximum, and decreases afterward, would be the regression with the ideal combination of having the best fit without fitting the noise. Therefore, R_a^2 is more appropriate when evaluating the PRS fit and comparing alternative models.

Example 2-5

A QoI y is a function of a single variable x, and has eventually been identified to follow a simple relationship $y_{true} = x$. However, the first set of measurements that were taken are given as four (x, y) sample pairs: $(-2, -1.5), (-1, -1.5), (1, 1.25), (2, 1.75)$. From theoretical considerations, you also know that $y(0) = 0$. Fit a linear function and a quadratic function to the samples, compare the two fits and see how they model the true function.

Solution:

Linear PRS: The PRS is of the form $\hat{y}(x) = b_1 + b_2 x$, and using the point $(0,0)$ as another sample point we can define the following matrices and vectors for regression:

$$\mathbf{X} = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{y} = \begin{Bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{Bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{y} = \begin{Bmatrix} 0 \\ 9.25 \end{Bmatrix}$$

By solving the normal equation in Eq. (2.14), the unknown regression coefficients are identified as $b_1 = 0$ and $b_2 = 0.925$. Therefore, the following linear PRS and residuals at sample locations can be obtained:

$$\hat{y}^{(1)}(x) = 0.925x, \quad \mathbf{e} = \begin{Bmatrix} 0.35 \\ -0.575 \\ 0.0 \\ 0.325 \\ -0.1 \end{Bmatrix}$$

The coefficient of multiple determination can be calculated from Eqs. (2.21) - (2.27) as

$$\bar{y} = 0,\ \ SS_y = 9.125,\ \ SS_r = 8.5563,\ \ R^2 = 0.9377,\ \ R_a^2 = 0.9169$$

Therefore, based on the coefficient of multiple determination, the linear PRS performs well in predicting the trend of samples. In fact, the linear PRS appears to be satisfactory in terms of error measures as well:

$$e_{RMS} = 0.337, e_{av} = 0.27, e_{max} = 0.575$$

Lastly, the estimate of the standard deviation of noise in Eq. (2.20) becomes

$$\hat{\sigma}^2 = \frac{SS_e}{n_y - n_\beta} = 0.1896, \quad \Rightarrow \quad \hat{\sigma} = 0.4354$$

which is close to the true standard deviation of noise $\sigma = 0.3953$, which can be obtained by comparing it with y_{true} and the samples.

Quadratic PRS: The PRS is of the form $\hat{y}(x) = b_1 + b_2 x + b_3 x^2$, and using point (0,0) as another sample point we can define the following matrices and vectors for regression:

$$\mathbf{X} = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}, \quad \mathbf{y} = \begin{Bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{Bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 5 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{y} = \begin{Bmatrix} 0 \\ 9.25 \\ 0.75 \end{Bmatrix}$$

By solving the normal equation in Eq. (2.14), the unknown regression coefficients are identified as $b_1 = -0.1071, b_2 = 0.925$, and $b_3 = 0.0536$. Therefore, the following quadratic PRS and residuals at sample locations can be obtained:

$$\hat{y}^{(2)}(x) = -0.1071 + 0.925x + 0.0536x^2, \quad \mathbf{e} = \begin{Bmatrix} 0.243 \\ -0.521 \\ 0.107 \\ 0.379 \\ -0.207 \end{Bmatrix}$$

The coefficient of multiple determination can be calculated from Eqs. (2.21) – (2.27) as

$$\bar{y} = 0,\ \ SS_y = 9.125,\ \ SS_r = 8.5964,\ \ R^2 = 0.9421,\ \ R_a^2 = 0.8841$$

We see that there is only a small improvement in R^2, and that R_a^2 is actually poorer, which is an indication that we have not gained any predictive capabilities by adding the quadratic terms. The same conclusion can be obtained based on the error metrics as

$$e_{RMS} = 0.325, e_{av} = 0.291, e_{max} = 0.521$$

These errors have not changed much compared to the linear model, with small improvements in the RMS and maximum errors, and a small increase in the average error. The estimate of the standard deviation of the noise in the samples from Eq. (2.20) becomes

$$\hat{\sigma}^2 = \frac{SS_e}{n_y - n_\beta} = 0.2643, \quad \Rightarrow \quad \hat{\sigma} = 0.5141$$

which is a small increase over the linear case. This increase is another indication that the quadratic approximation is not better than the linear approximation.

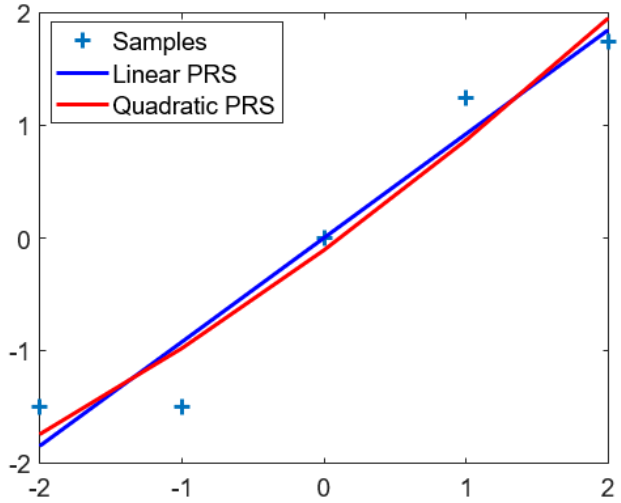


Figure 2-7: Comparison of linear and quadratic polynomial responses

---

We discussed before that PRS starts from the assumption that the model form is accurate, but samples have random noise that follows a normal distribution with zero mean. In reality, however, we do not know the correct model form. We do not know if the true functional form is polynomials either. From Taylor's expansion theorem, it is possible that any continuous and continuous function can be approximated using polynomials. Therefore, we can accept the idea that the true function is in the form of polynomials. Then the remaining question is what orders of polynomials should be used. As shown in the previous example, a higher-order polynomial does not mean a better approximation. Therefore, in order to find an appropriate model form, we start from lower-order polynomials and gradually increase the order until the approximation becomes worse, or we start from higher-order polynomials and gradually remove basis functions that do not contribute significantly. In Section 2.5, we will discuss the second approach, which is called backward elimination.

Example 2-6

Compare the average error and the coefficient of multiple determination of two PRS surrogates that are generated from two true functions $y^{(1)}(x) = x$ and $y^{(2)}(x) = 0.1x$. In order to build the surrogates, generate 31 equal-interval samples from the design space $x \in [0, 30]$ and add random noise from the normal distribution $\sim N(0,1^2)$. Check if e_{av} and R^2 can be used to assess the accuracy of surrogates.

Solution:

In order to generate the same sequence of random numbers, the `rng default` command is used before generating random numbers in Matlab. After generating samples of true functions $y_{true}^{(1)}$ and $y_{true}^{(2)}$, the same random noises are added to the true functions. Since the input variable is one dimension, `polyfit` and `polyval` Matlab functions are used for building the linear PRS. The two PRS surrogates are, respectively,

$$\hat{y}^{(1)}(x) = 0.5728 + 0.9993x$$

$$\hat{y}^{(2)}(x) = 0.5728 + 0.0993x$$

Both PRSs have the same average error $e_{av} = 0.5628$. However, the coefficients of multiple determination are quite different. R^2 for $\hat{y}^{(1)}$ is 0.9806, while that of $\hat{y}^{(2)}$ is 0.3325. Therefore, even if the

average error of the two surrogates is the same, R^2 of the first surrogate looks much better than the second. This happens because even if the noises are the same $y^{(1)}$ varies much faster than $y^{(2)}$. As shown in Figure 2-8(a), the difference between samples and PRS surrogate seems small, while the difference looks significant in Figure 2-8(b). Therefore, R^2 does not reflect the accuracy of a surrogate model. It shows the ratios of the variation of PRS surrogate to the variation of samples. The following Matlab code is used to plot Figure 2-8.

```
rng default;
x=linspace(0,30,31); noise=randn(1,31);
y1true=x; y1=y1true + noise;
y2true=0.1*x; y2=y2true + noise;
b1=polyfit(x,y1,1);
b2=polyfit(x,y2,1);
y1p=polyval(b1,x);
y2p=polyval(b2,x);
eva1= sum(abs(y1p-y1true))/31;
eva2= sum(abs(y2p-y2true))/31;
meany1=mean(y1);
meany2=mean(y2);
SSy1= sum((y1-meany1).^2);
SSy2=sum((y2-meany2).^2);
SSr1=sum((y1p-meany1).^2);
SSr2=sum((y2p-meany2).^2);
R1=SSr1/SSy1;
R2=SSr2/SSy2;
figure(1)
plot(x,y1,'+',x,y1true,'b',x,y1p,'r');
legend('Samples','True function','Linear PRS');
figure(2)
plot(x,y2,'+',x,y2true,'b',x,y2p,'r');
legend('Samples','True function','Linear PRS');
```

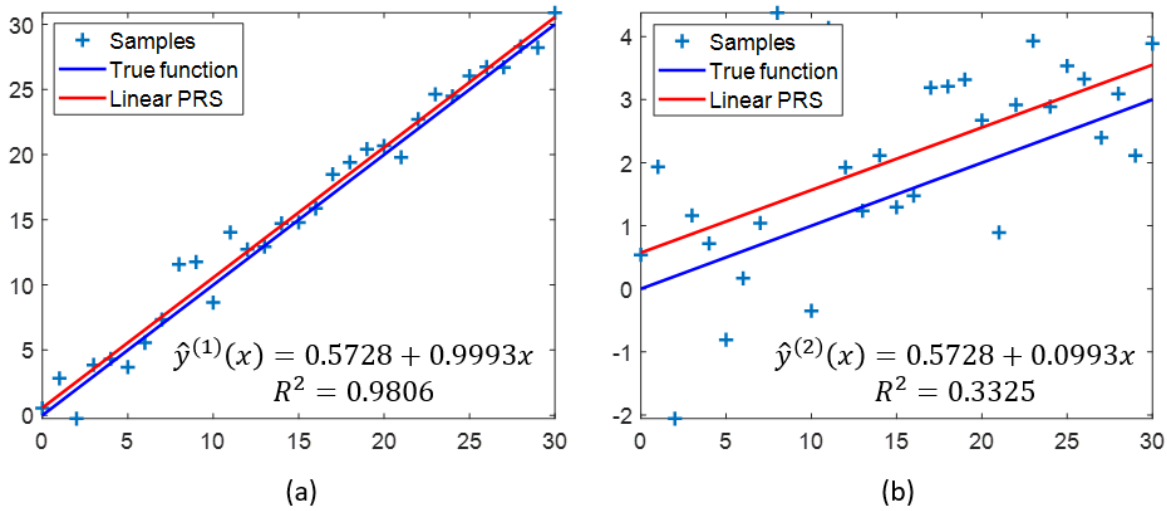


Figure 2-8: Fitted linear polynomial response surface from (a) $y = x$ and (b) $y = 0.1x$.

The Matlab function `regress` can calculate the coefficient of multiple determination as a part of its model statistics. For example, the following code can be used to calculate R^2:

```
[b,~,~,~,stats]=regress(y, X);
R2=stats(1)
```

Cross-validation

The only true test of the predictive capabilities of the response surface is evaluating it at points not used in its construction. In that aspect, the coefficient of multiple determination cannot be a good measure to evaluate the goodness-of-fit. In order to evaluate the prediction capability of a surrogate, it would be necessary to have a separate set of samples that are not used in the fitting process. Therefore, it would be necessary to divide the entire set of samples into a training set and a validation set. The samples in the training set are used for fitting a surrogate, and the samples in the validation set are used for evaluating the accuracy or prediction capability of the surrogate. This may be considered wasteful because we do not use all the samples for fitting the best possible surrogate. Because additional tests or numerical evaluations of y for validation are often expensive, it is worthwhile to look for a way of checking predictive capability without performing additional evaluations at new points.

Cross-validation is a resampling method that uses different portions of the samples to train and validate a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. The goal of cross-validation is to test the model's ability to predict new samples that were not used in estimating it, in order to flag problems like overfitting or selection bias [18] and to give an insight into how the model will generalize to a new dataset. Cross-validation combines (averages) measures of fitness in prediction to derive a more accurate estimate of model prediction performance [19].

k-fold cross-validation: In k-fold cross-validation, the original samples are randomly partitioned into k equal-sized subsets. Of the k subsets, a single subset is retained as the validation sample set for testing the model, and the remaining $k-1$ subsets are used as training samples. The cross-validation process is then repeated k times, with each of the k subsets used exactly once as the validation set. The k results can then be averaged to produce a single estimation. In this method, all samples are used for both training and validation, and each sample is used for validation exactly once. Figure 2-9 shows an example when $k = 5$. The entire samples are divided into five equal-sized subsets. Although the division looks sequential in the figure, in practice subsets are generated randomly. The first four subsets are used to fit a surrogate, and the last subset is used to validate the surrogate. The outcome of the validation is in terms of prediction error e_{p1}. This process is repeated for other subsets and yields five predictions errors: $\mathbf{e}_p = \{e_{p1}, e_{p2}, e_{p3}, e_{p4}, e_{p5}\}^T$.

Figure 2-9: Five-fold cross-validation.

Leave-one-out cross-validation: Leave-one-out cross-validation involves using one sample as the validation set and the remaining $n_y - 1$ samples as the training set. That is, $n_y - 1$ samples are used to fit a surrogate, and the error between the surrogate prediction and the validation sample is calculated as a prediction error e_{pi}. This process is repeated in all n_y samples to yield n_y prediction errors. This is equivalent to k-fold cross-validation with $k = n_y$. The process looks similar to a jackknife; however, with cross-validation one computes a statistic on the left-out sample(s), while with jackknifing one computes a statistic from the kept samples only. In this section, we will only explain leave-one-out cross-validation.

If the number of samples used for the fit is substantially larger than the number of coefficients, $n_y \gg n_\beta$, leaving out one sample will not change the quality of the fit significantly. We can therefore leave out one sample, fit the PRS to the remaining samples and check the error at the point that was left out. Let us assume that we leave out the ith sample. The surrogate that is built without ith sample is denoted by $\hat{y}^{(i)}$. Then, the prediction error at the ith sample is

$$e_{pi} = y_i - \hat{y}^{(i)}(\mathbf{x}_i) \tag{2.28}$$

Note that the prediction error e_{pi} is different from the regression error: $e_i = y_i - \hat{y}(\mathbf{x}_i)$. We can then repeat the procedure at each sample to obtain the vector of prediction errors $\mathbf{e}_p^T = \{e_{p1}, e_{p2}, \cdots, e_{pn_y}\}$. Since we need a single scalar measure to evaluate the prediction accuracy of a surrogate, the square sum of the prediction errors can be used, which is known as PRESS (predicted residual error sum of squares). The PRESS can be defined as

$$PRESS = \sum_{i=1}^{n_y} e_{pi}^2 = \sum_{i=1}^{n_y} \left[y_i - \hat{y}^{(i)}(\mathbf{x}_i)\right]^2 \tag{2.29}$$

The PRESS statistic provides a summary measure of the fit of a model to a sample of observations that were not themselves used to estimate the model [20]. The PRESS statistic can be calculated for a number of surrogate models for the same set of samples, with the lowest values of PRESS indicating the best surrogate. Surrogate models that are over-fitted samples would tend to give small residuals for samples included in the model-fitting but large residuals for samples that are excluded.

Another advantage of the PRESS statistic is that it is independent of surrogate models. That is, it can be used in any surrogate model. Therefore, it is versatile. The only bottleneck of the PRESS statistic is that it can be expensive to fit surrogate models n_y times. As n_y increases, this can be expensive. However, the baseline assumption of surrogate modeling is that the major cost comes from obtaining samples, which means performing experiments or running expensive computer simulations. Therefore, the fitting cost of the surrogate can be ignorable.

In the case of linear regression, however, the requirement of fitting surrogates n_y times can be removed. The PRESS statistics can be evaluated with one surrogate fitting. First, we derive the linear relationship between the surrogate predictions and samples. By evaluating the surrogate predictions at all the sample points, we have

$$\hat{\mathbf{y}} = \mathbf{X} \cdot \mathbf{b} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{E} \cdot \mathbf{y} \tag{2.30}$$

where the $n_y \times n_y$ symmetric matrix $\mathbf{E}$ is an idempotent matrix. An important property of the idempotent matrix is that $\mathbf{E}^k = \mathbf{E}$. Using this relationship, the regression errors at sample locations in Eq. (2.11) can be written as

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = [\mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]\mathbf{y} = [\mathbf{I} - \mathbf{E}]\mathbf{y} \tag{2.31}$$

where $[\mathbf{I}-\mathbf{E}]$ is also an idempotent matrix. In particular, the diagonal term $1-E_{ii}$ represents the importance of sample y_i on error e_i. It would be difficult to derive, but it can be shown that the relationship between the regression error e_i and the prediction error e_{pi} can be written as

$$e_{pi}=\frac{e_i}{1-E_{ii}} \tag{2.32}$$

Basically, the prediction error at the ith sample is the regression error scaled by the diagonal term $1-E_{ii}$. Therefore, the PRESS statistic can be calculated from a single surrogate fit as

$$PRESS=\sum_{i=1}^{n_y}\left(\frac{e_i}{1-E_{ii}}\right)^2 \tag{2.33}$$

The root-mean-squared error of the PRESS statistic can be defined as

$$e_{PRESS}=\sqrt{\frac{1}{n_y-1}\sum_{i=1}^{n_y}\left(\frac{e_i}{1-E_{ii}}\right)^2} \tag{2.34}$$

The e_{PRESS} is preferred over $PRESS$ because the former has the same unit as QoI.

It should be noted that the matrix $\mathbf{X}^T\mathbf{X}$ is often ill-conditioned, especially for large problems, and then the calculation of the matrix $\mathbf{E}$ from Eq. (2.30) is not very reliable. However, even the direct calculation of $\mathbf{e}_p$, by performing the n_y times of surrogate fits, is usually less expensive than carrying out additional experiments in order to test the accuracy of the surrogate.

Example 2-7

Generate 30 equal-interval samples from the true function $y=x, x\in[1,30]$ with randomly distributed noise $\sim N(0,1^2)$. Fit a linear PRS and compare (a) the constant term b_1 with the mean of noise, (b) the standard error with the standard deviation of noise samples, and (c) the standard error and the PRESS statistic.

Solution:

We used `rng default` command to keep the random number sequence the same. The mean of noise is $\mu_{noise}=0.5519$ and the standard deviation of the noise is $\sigma_{noise}=1.3$. Fitting the 30 samples with a linear PRS yields the following expression

$$\hat{y}(x)=0.5981+0.997x$$

It is noted that the constant term $b_1=0.5981$ is the same as the mean of noise. This is because the linear regression is unbiased fitting. That is, even if samples were generated from the true function since the noise samples were biased by 0.5519, the PRS surrogate is also shifted by a similar level of bias. This is not identical to the mean of noise because some errors are shifted to the linear coefficient b_2.

The standard error is calculated based on the square root of Eq. (2.20): $\hat{\sigma}=1.3228$, which is close to the standard deviation of noise, 1.30. In addition, the root-mean-squared error of the PRESS statistic, e_{PRESS} is calculated using Eq. (2.34): $e_{PRESS}=1.3907$, which is also close to the standard error.

The actual RMS error between $\hat{y}(x)$ and the true function is 0.5619, which is much smaller than the standard error. This is because a large number of samples filtered the noise. As we discussed in Section 2.2, the surrogate is more accurate than the samples.

The following Matlab code is used for the example:

```
rng default;
x=[1:30]'; noise=randn(30,1);y=x+noise;
```

```
X=[ones(30,1) x];
b=regress(y,X)
yfit=b(1)+b(2)*x;
error=y-yfit;
sigma=sqrt(error'*error/28)
mean(noise)
std(noise)
M=X'*X; E=X*inv(M)*X';
d=diag(E)
ep=error./(1-d);
PRESS=ep'*ep
ePRESS=sqrt(PRESS/29)
```

Example 2-8

Consider the quadratic PRS in **Example 2-5**, calculate the PRESS statistic by repeating the fitting process by 5 times, and compare it with the PRESS statistic from a single fit in Eq. (2.33).

Solution:

First, we calculate the PRESS statistic from a single fit. From **Example 2-5**, the idempotent matrix can be obtained as

$$\mathbf{E} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \begin{bmatrix} 0.8857 & 0.2671 & -0.0857 & -0.1429 & 0.0857 \\ 0.2571 & 0.3714 & 0.3429 & 0.1714 & -0.1429 \\ -0.0857 & 0.3429 & 0.4857 & 0.3429 & -0.0857 \\ -0.1429 & 0.1714 & 0.3429 & 0.3714 & 0.2571 \\ 0.0857 & -0.1429 & -0.0857 & 0.2571 & 0.8857 \end{bmatrix}$$

Therefore, the regression errors and the prediction errors from a single fit can be obtained from Eq. (2.32) as

$$\mathbf{e} = \begin{Bmatrix} 0.243 \\ -0.521 \\ 0.107 \\ 0.379 \\ -0.207 \end{Bmatrix}, \qquad \mathbf{e}_p = \left\{\frac{e_i}{1-E_{ii}}\right\} = \begin{Bmatrix} 2.125 \\ -0.8295 \\ 0.2083 \\ 0.6023 \\ -1.8125 \end{Bmatrix}$$

It is noted that the prediction error significantly increased in the first and last samples. This is because E_{11} and E_{55} are close to one. Physically it means that when the first and the last sample is dropped, they belong to the extrapolation region after fitting the surrogate.

Second, we demonstrate the case when the third sample (0,0) is dropped. In this case, the following matrices and vectors can be defined:

$$\mathbf{X} = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}, \quad \mathbf{y} = \begin{Bmatrix} -1.5 \\ -1.5 \\ 1.25 \\ 1.75 \end{Bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{y} = \begin{Bmatrix} 0 \\ 9.25 \\ 0.75 \end{Bmatrix}$$

By solving the normal equation in Eq. (2.14), the unknown regression coefficients are identified as $b_1 = -0.2083, b_2 = 0.925$, and $b_3 = 0.0833$. Therefore, the following quadratic PRS can be obtained:

$$\hat{y}^{(3)}(x) = -0.2083 + 0.925x + 0.0833x^2$$

Now the prediction error is calculated at the dropped sample location (0,0), which is

$$e_{p3} = y_3 - \hat{y}^{(3)}(0) = 0 + 0.2083 = 0.2083$$

This process is repeated for every sample position to obtain the following prediction errors:

$$\mathbf{e}_p = \{y_i - \hat{y}^{(i)}(x_i)\} = \begin{Bmatrix} 2.125 \\ -0.8295 \\ 0.2083 \\ 0.6023 \\ -1.8125 \end{Bmatrix}$$

Therefore, the prediction error from Eq. (2.32) is identical to the prediction error from the definition of cross-validation. The following Matlab code is used to calculate the prediction errors:

```
X=[1 -2 4;1 -1 1;1 0 0;1 1 1;1 2 4];
y=[-1.5 -1.5 0 1.25 1.75]';
for i=1:5
 Xt=X; Xt(i,:)=[];
 yt=y;yt(i)=[];
 bt=inv(Xt'*Xt)*Xt'*yt;
 ept(i)=y(i)-X(i,:)*bt;
end
```

Since the PRESS statistic measures the prediction accuracy of a surrogate, a surrogate is considered to be accurate when the PRESS statistic is small. When multiple surrogates are compared, the surrogate whose PRESS statistic is the smallest can be considered the best surrogate to fit the samples. In the case of PRS surrogates, however, the PRESS static will not converge to zero. This is expected because the PRS starts from the assumption that the samples have random noise and a good surrogate is supposed to fit the trend of samples, not the noise. Therefore, if a surrogate is identical to the true function, the diagonal term of the idempotent matrix $E_{ii} = 0$ and the PRESS statistic in Eq. (2.33) becomes the square-sum-error SS_e.

2.5. Confidence of coefficients and backward elimination

In **Example 2-5** we compared a linear PRS surrogate to a quadratic PRS surrogate. However, we do not have to limit ourselves to PRS surrogates that have all the terms up to a particular order. For example, we can consider a quadratic model without the constant term, $\hat{y}(x) = b_1 x + b_2 x^2$. Similarly, with two variables, it is common for people to consider a model of the form $\hat{y}(\mathbf{x}) = b_1 + b_2 x_1 + b_3 x_2 + b_4 x_1 x_2$. This model does not include quadratic terms in x_1 and x_2, but it includes the 'interaction' term $x_1 x_2$. We can stipulate such a partial model based on some knowledge of the behavior of the true function. However, most often such partial models are created by discarding terms with coefficients that cannot be accurately estimated based on the samples. Such coefficients do not have much effect on the accuracy of the fit of the given samples, and leaving them in the model can reduce its predictive quality for design points where these coefficients have a large effect on the prediction.

The first straightforward idea is to remove those basis functions whose coefficients are small compared to other coefficients. For example, let us consider the following quadratic PRS surrogate:

$$\hat{y}(x_1, x_2) = 10.5 + 0.01x_1 + 9.87x_2 - 5x_1^2 + 7.6x_1 x_2 + 0.02x_2^2 \tag{2.35}$$

If we assume that both input variables x_1 and x_2 are normalized $x_1, x_2 \in [0, 1]$, then it is clear that the linear x_1 term and quadratic x_2^2 term are not important as their coefficients are small compared to other coefficients. That is, even if x_1 and x_2^2 vary, the surrogate prediction will not change much because of their small coefficients. Therefore, these two basis functions can be removed to yield

$$\hat{y}(x_1, x_2) = 10.5 + 9.87x_2 - 5x_1^2 + 7.6x_1x_2 \tag{2.36}$$

However, such a heuristic approach may cause a problem if other basis functions with a large coefficient vary significantly with a small change in samples. Therefore, it would be necessary to develop a systematic approach to determine which basis functions or which coefficients should be removed from the surrogate model.

As we discussed several times, PRS surrogates start from the assumption that there is a true function, and samples include random noise to the true function. We try to build a surrogate that can filter out random noise and approximate the true function as closely as possible. Because of this randomness in noise, however, different samples can be obtained by different realizations of the noise, which end up identifying different coefficients. Therefore, it is fair to say that the identified coefficients of the PRS are random as well, whose distribution we want to characterize. A straightforward way of identifying the distribution of coefficients is to generate many sets of samples by adding random noise to the true function. By fitting these multiple sets of samples, we can obtain multiple sets of coefficients, from which we can estimate the distribution. However, this approach is only possible when we know the true function and we can generate multiple sets of samples. In reality, it is already expensive to generate one set of samples. Therefore, it would be necessary to come up with a method of identifying the distribution of coefficients with one set of samples.

We can identify the distribution of these coefficients without changing the samples by estimating the standard deviation of the coefficients. That is, we assume that the coefficients are normally distributed with the values obtained from linear regression as a mean. Therefore, the standard deviation is the only unknown information needed to be estimated. Since a change in a sample can affect multiple coefficients simultaneously, it would be necessary to estimate the covariance of the coefficients. Let us first introduce the covariance matrix $\mathbf{\Sigma_b}$ of the vector of random coefficients $\mathbf{b}$, which is defined as

$$\mathbf{\Sigma_b} = [\mathbf{b} - E(\mathbf{b})][\mathbf{b} - E(\mathbf{b})]^T \tag{2.37}$$

where $E(\cdot)$ is the expected value of a random variable. That is, due to random noise we get different vectors $\mathbf{b}$ by fitting the samples from different realizations. $E(\mathbf{b})$ is the expected value (or average over a very large number of experiments) of $\mathbf{b}$, and $\mathbf{b} - E(\mathbf{b})$ is the deviation of $\mathbf{b}$ from its expected value. Then the covariance matrix is the expected value of products of various components of the deviations. In particular, the diagonal terms of the matrix are by definition the variance of the components of $\mathbf{b}$, while the off-diagonal terms are a measure of the correlation between components. It is possible to show that

$$\mathbf{\Sigma_b} = \sigma^2[\mathbf{X}^T\mathbf{X}]^{-1} \tag{2.38}$$

That is, due to the randomness in noise, the estimated coefficients from the regression are also random, whose randomness follows a normal distribution $\sim N(\mathbf{b}, \mathbf{\Sigma_b})$.

Note that σ is the standard deviation of random noise, which is unknown. Instead, we can use its estimate $\hat{\sigma}$ derived in Eq. (2.20). Now, we can estimate the standard deviation of individual coefficients as

$$\sigma_{b_i} = \hat{\sigma}\sqrt{[(\mathbf{X}^T\mathbf{X})^{-1}]_{ii}} \tag{2.39}$$

which is referred to as the standard error of the coefficient. A useful measure of the accuracy of a component of $\mathbf{b}$ is the estimate of the coefficient of variation, $\mathbf{c}$, which is the ratios of the standard deviation to the absolute value of the component. The coefficient of variation of the ith component c_i is defined as

$$c_i = \frac{\sigma_{b_i}}{|b_i|} \tag{2.40}$$

If $c_i > 1$, it means that the uncertainty (i.e., the standard deviation) is larger than the coefficient itself, which means that the identified coefficient has very low confidence.

The coefficient $|b_i|$ and its coefficient of variation c_i can be used to identify insignificant basis functions of the PRS. An important basis function should have a large magnitude of the coefficient such that the surrogate prediction strongly depends on the term. Also, if the coefficient of variation is small, it means that the coefficient is well-identified. On the other hand, when the coefficient cannot be identified accurately for given samples, it means that the coefficient does not have much effect on the accuracy of the PRS.

In many PRS procedures, the quantity which is used to assess the need for a coefficient is the inverse of c_i, which is called the test statistic, or the t-statistic. The coefficient of variation is used in a strategy called backward elimination (Myers and Montgomery, p. 650 [4]). In this strategy, we eliminate the coefficient with the largest coefficient of variation. After removing the basis function, the surrogate is performed with the reduced basis function with the same samples. This process is repeated until the coefficient of variation of the remaining terms is small enough, or we can use a measure such as R_a^2 or e_{PRESS} to indicate when eliminating additional terms reduces the accuracy of the PRS. This is illustrated by applying this strategy to the quadratic PRS in **Example 2-5**.

Example 2-9

Use backward elimination, starting with the quadratic model of **Example 2-5**, to find the model with the highest value of R_a^2. Confirm your conclusion by recalculating the coefficients for a slight perturbation of the sample at $x = 1$ to $y(1) = 1.35$.

Solution:

The quadratic PRS that we obtained from **Example 2-5** was $\hat{y}(x) = -0.1071 + 0.925x + 0.0536x^2$, which has the adjusted coefficient of multiple determination $R_a^2 = 0.8841$ and the estimated noise standard deviation $\hat{\sigma} = 0.5141$. In order to determine which coefficient should be removed, the inverse of the moment matrix and the covariance matrix of the coefficients are calculated as

$$(\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} \frac{17}{35} & 0 & -\frac{1}{7} \\ 0 & \frac{1}{10} & 0 \\ -\frac{1}{7} & 0 & \frac{1}{14} \end{bmatrix}, \quad \boldsymbol{\Sigma_b} = \begin{bmatrix} 0.1284 & 0 & -0.03776 \\ 0 & 0.02643 & 0 \\ -0.03776 & 0 & 0.01888 \end{bmatrix}$$

Using these matrices, the coefficients of variation of all the coefficients can be calculated as

$$c_1 = \frac{\sqrt{0.1284}}{0.1071} = 3.35, \quad c_2 = \frac{\sqrt{0.02643}}{0.925} = 0.176, \quad c_3 = \frac{\sqrt{0.01888}}{0.0536} = 2.56$$

It turns out that the first coefficient b_1 has the largest coefficient of variation, and thus, we can remove the constant term. Therefore, the reduced PRS form becomes $\hat{y}(x) = b_2x + b_3x^2$. Note that we cannot use b_2 and b_3 from the initial surrogate because the reduced surrogate has different basis functions. Therefore, we need to fit the reduced surrogate again with the same samples, with the following matrices and vectors:

$$\mathbf{X} = \begin{bmatrix} -2 & 4 \\ -1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 2 & 4 \end{bmatrix}, \quad \mathbf{y} = \begin{Bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{Bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 10 & 0 \\ 0 & 34 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{y} = \begin{Bmatrix} 9.25 \\ 0.75 \end{Bmatrix}$$

The normal equation yields $b_2 = 0.925$ and $b_3 = 0.0221$. That is, $\hat{y}(x) = 0.925x + 0.0221x^2$. Therefore, the linear coefficient does not change, but the quadratic coefficient changes significantly from 0,0536 to 0.0221. The reduced surrogate has the following regression errors:

$$\mathbf{e} = \{-0.262, 0.597, 0.0, -0.3038, 0.188\}^T$$

And the various error metrics are $SS_y = 9.125$, $SS_e = 0.5522$, $R^2 = 0.9394$, $R_a^2 = 0.9193$, $\hat{\sigma} = 0.429$. Compared to R_a^2 of the initial quadratic PRS, R_a^2 of the reduced PRS is increased from 0.8841 to 0.9193. Therefore, we can say that eliminating the constant term improves the prediction capability.

In order to check if a further elimination improves the surrogate, the coefficients of variation are calculated again. First, the following matrices are defined:

$$(\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} \frac{1}{10} & 0 \\ 0 & \frac{1}{34} \end{bmatrix}, \quad \mathbf{\Sigma_b} = \begin{bmatrix} 0.0184 & 0 \\ 0 & 0.00541 \end{bmatrix}$$

Using these matrices, the coefficients of variation of all the coefficients can be calculated as

$$c_2 = \frac{\sqrt{0.0184}}{0.925} = 0.147, \quad c_3 = \frac{\sqrt{0.00541}}{0.0221} = 3.33$$

Based on the three coefficients of variation, the last coefficient b_3 has the largest coefficient of variation, and thus, we can remove the quadratic term. Therefore, the reduced PRS form becomes $\hat{y}(x) = b_2 x$. Fitting this linear PRS yields $\hat{y}(x) = 0.925x$ with error metrics $R_a^2 = 0.9169$, $\hat{\sigma} = 0.4354$. Therefore, R_a^2 is slightly decreased from the previous PRS surrogate, which means the previous PRS surrogate is better than this one. Therefore, we stop the elimination process and conclude that the best PRS surrogate to fit the samples is $\hat{y}(x) = 0.925x + 0.0221x^2$.

Now let the sample at $x = 1$ is perturbed from $y(1) = 1.25$ to $y(1) = 1.35$, which corresponds to 8% perturbation. Then, the coefficients of the original quadratic PRS are perturbed by

$$\hat{y}_{original}(x) = -0.1071 + 0.925x + 0.0536x^2$$

$$\hat{y}_{perturbed}(x) = -0.0729 + 0.935x + 0.0465x^2$$

That is, the constant coefficient is changed by 30%, the linear coefficient by 1%, and the quadratic coefficient by 13%. Therefore, it makes sense that the constant coefficient is the most uncertain and needs to be eliminated first. In the case of the reduced PRS, the coefficients are perturbed by

$$\hat{y}_{original}(x) = 0.925x + 0.02205x^2$$

$$\hat{y}_{perturbed}(x) = 0.935x + 0.025x^2$$

The linear coefficient is changed by 1%, while the quadratic coefficient is by 13%. However, the elimination of the quadratic coefficient may decrease the prediction capability of the surrogate.

2.6. Prediction variance

In Section 2.4, we discussed cross-validation, which is a good metric to measure the prediction capability at unsampled locations. The good characteristic of cross-validation is that it is model-independent, which means that it can be applied to any kind of surrogate, not limited to linear regression PRS. However, cross-validation can provide the prediction accuracy of a surrogate in the form of prediction errors at sample locations and RMS of prediction errors. The latter is a good estimate of the standard deviation of

the noise. Therefore, cross-validation is a good metric to assess the overall accuracy of a surrogate. However, the ultimate goal of the accuracy study is to estimate the prediction accuracy of the surrogate at unsampled locations.

Prediction uncertainty

If we limit ourselves to the PRS surrogate with linear regression, it is possible to estimate the prediction accuracy at unsampled locations. As mentioned before, the PRS surrogate that we obtained from regression in Eq. (2.10) is actually the mean of the prediction. This is because the PRS surrogate starts from the assumption of random noise. Because of the random distribution of noise $\sim N(0,\sigma^2)$, the regression coefficients are also random $\sim N(\mathbf{b},\mathbf{\Sigma_b})$, and thus, the prediction at unsampled locations as well. Since the relationship between the prediction and coefficients is linear and since the basis functions are not random, the prediction at an unsampled point $\mathbf{x}_p$ will also follow a normal distribution $\sim N\left(\hat{y}(\mathbf{x}_p),\sigma_y^2(\mathbf{x}_p)\right)$. That is, the obtained PRS surrogate is a mean prediction and $\sigma_y^2(\mathbf{x}_p)$ is the prediction variance. Since this uncertainty in prediction is caused by the randomness in noise, it makes sense that the prediction uncertainty is proportional to the standard deviation σ of the noise.

Also, the prediction accuracy is usually good close to the sample locations because even if the samples include random noise, the regression process filters out the noise and the samples represent the trend of the true function. Therefore, as the prediction point moves away from sample locations, the prediction becomes inaccurate; that is, the prediction uncertainty increases. It would be better to explain the meaning of prediction uncertainty using a normal distribution. Figure 2-10 shows the probability density function of a normal distribution. The mean location corresponds to the surrogate prediction $\hat{y}(\mathbf{x}_p)$. Then the probability that the true function is located within the range $[\hat{y}(\mathbf{x}_p)-\sigma_y(\mathbf{x}_p),\hat{y}(\mathbf{x}_p)+\sigma_y(\mathbf{x}_p)]$ is 68%, $[\hat{y}-2\sigma_y,\hat{y}+2\sigma_y]$ is 95%, and $[\hat{y}-3\sigma_y,\hat{y}+3\sigma_y]$ is 99.7%. If the prediction uncertainty $\sigma_y(\mathbf{x}_p)$ is large, then the chance that the surrogate prediction is close to the true function is low. Therefore, it is important to have a low prediction uncertainty in order to have a good prediction capability. If σ_y can not be low at everywhere, it should be low at the point of interest, such as the optimum design point.

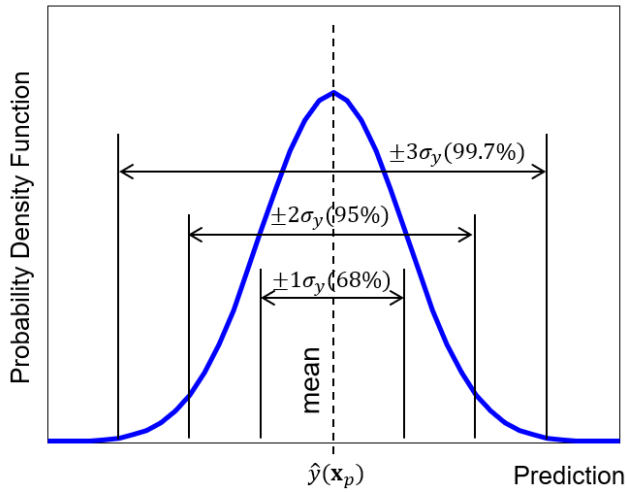


Figure 2-10: Normal distribution of surrogate prediction.

An important thing to note in prediction uncertainty is that it tends to be low in the interpolation region, while it tends to increase fast in the extrapolation region. As shown in Figure 1-5, the interpolation region is within the convex hull of the sample points, while the extrapolation region is outside of the

convex hull. The prediction in the extrapolation region is associated with large errors. Especially for a high dimensional domain, extrapolation usually cannot be avoided.

The PRS expression in Eq. (2.10) can be written as $\hat{y}(\mathbf{x}_p) = \boldsymbol{\xi}(\mathbf{x}_p)^T\mathbf{b}$. In the viewpoint of uncertainty, the vector of basis functions $\boldsymbol{\xi}(\mathbf{x}_p)$ is deterministic, while the vector of coefficient $\mathbf{b}$ is uncertain. In the probability theory, the variance of a random variable $y = ax$ with variance $V[x] = \sigma^2$ can be obtained by $V[y] = a^2V[x] = a^2\sigma^2$. When the random variable is a vector with multiple components, the covariance matrix is used instead. Therefore, the variance of the prediction $\hat{y}(\mathbf{x}_p)$ can be written as $V\left[\hat{y}(\mathbf{x}_p)\right] = \boldsymbol{\xi}(\mathbf{x}_p)^T\boldsymbol{\Sigma}_\mathbf{b}\boldsymbol{\xi}(\mathbf{x}_p) = \hat{\sigma}^2\boldsymbol{\xi}(\mathbf{x}_p)^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x}_p)$. The square root of the prediction variance is called the standard error of prediction, defined as

$$\sigma_y = \hat{\sigma}\sqrt{\boldsymbol{\xi}(\mathbf{x}_p)^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x}_p)} \tag{2.41}$$

where we used $\hat{\sigma}$ as an estimate of σ. The standard error of prediction increases proportionally to the standard deviation of the noise. Also, the term $\boldsymbol{\xi}(\mathbf{x}_p)^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x}_p)$ increases fast as the prediction point $\mathbf{x}_p$ moves away from sample points.

Sample sensitivity

Another interesting piece of information related to prediction accuracy is the sensitivity of prediction with respect to samples. That is, how much the prediction $\hat{y}(\mathbf{x}_p)$ will vary due to a change in one sample y_i? This information is particularly important if we want to find out the most important samples. Of course, the locations of samples affect significantly the prediction performance. However, this topic will be discussed separately in Chapter 3. In this section, it is assumed that the locations of all samples are fixed, and only the value of QoI y_i varies.

This information is readily available from the definition of PRS surrogate with identified coefficients

$$\hat{y}(\mathbf{x}_p) = \boldsymbol{\xi}(\mathbf{x}_p)^T\mathbf{b} = \boldsymbol{\xi}(\mathbf{x}_p)^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2.42}$$

Since sample locations are fixed, the moment matrix $\mathbf{X}^T\mathbf{X}$ is fixed. Differentiating Eq. (2.42) with respect to ith component of $\mathbf{y}$ yields

$$\frac{\partial\hat{y}(\mathbf{x}_p)}{\partial y_i} = \left\{\boldsymbol{\xi}(\mathbf{x}_p)^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\right\}_i \tag{2.43}$$

It is noted that the sample sensitivity in Eq. (2.43) only shows the effect of the sample value on the mean prediction. Its effect on the prediction uncertainty is more complicated through the standard deviation of the noise.

Example 2-10

The design space of a linear PRS $\hat{y}(x_1, x_2) = b_1 + b_2x_1 + b_3x_2$ is given by $-1 \le x_1, x_2 \le 1$. (a) When three sample locations are given as $(x_1, x_2) = (-1,-1), (-1,1), (1,-1)$, calculate the standard error of prediction in terms of noise standard deviation $\hat{\sigma}$ at all corner points and the center of the design space. (b) Find the location and value or the minimum standard error of prediction. (c) If an additional sample is given at $(1,1)$, repeat part (a).

Solution:

(a) With the three samples, Figure 2-11(a) shows the interpolation and extrapolation regions along with the samples. For linear PRS, the vector of basis functions and design matric is defined as

$$\boldsymbol{\xi} = \begin{Bmatrix} 1 \\ x_1 \\ x_2 \end{Bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}, \quad (\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{4}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

The standard error of prediction in Eq. (2.41) is given as

$$\sigma_y = \hat{\sigma}\sqrt{\boldsymbol{\xi}^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}} = \hat{\sigma}\sqrt{0.5(1 + x_1 + x_2 + x_1^2 + x_2^2 + x_1x_2)}$$

The standard error at the three sample locations is $\sigma_y = \hat{\sigma}$. That is, the prediction uncertainty at the sample location is the same as the uncertainty of the sample itself. At the center of the design space $(x_1, x_2) = (0,0)$, $\sigma_y = \hat{\sigma}/\sqrt{2}$, which is lower than that of the sample locations. On the other hand, the unsampled corner $(x_1, x_2) = (1,1)$ has the highest prediction uncertainty of $\sigma_y = \sqrt{3}\hat{\sigma}$. This is because this corner is the farthest extrapolation point.

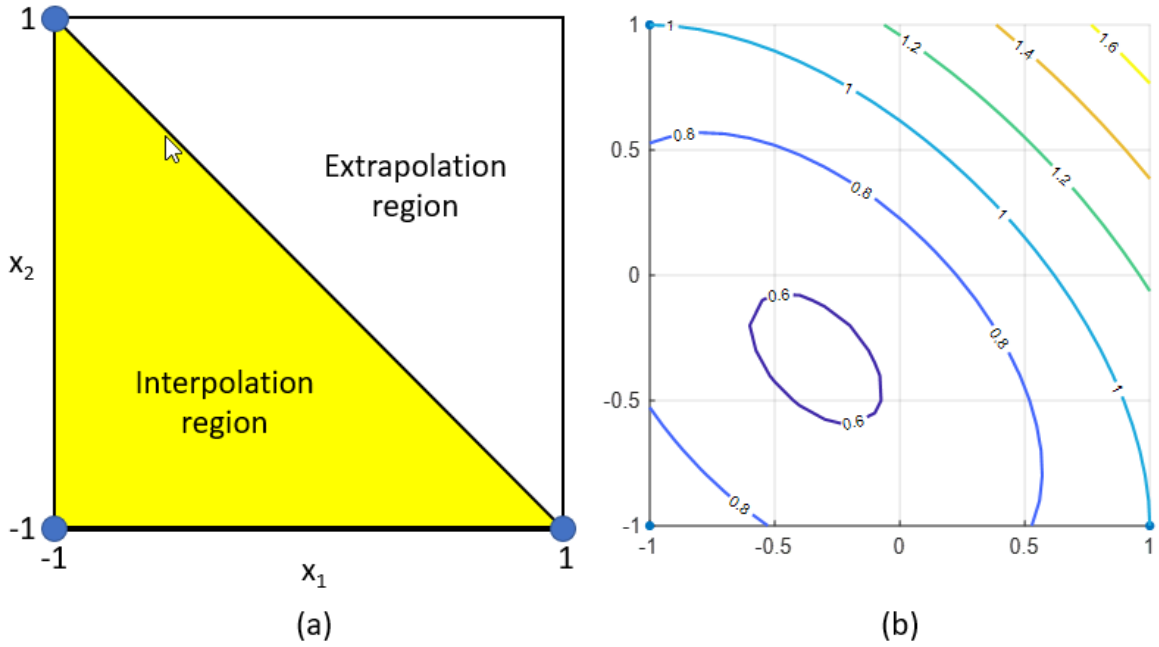


Figure 2-11: Interpolation and extrapolation regions defined by samples in **Example 2-10**.

(b) It is interesting to note that the prediction uncertainty is not minimum at sample locations. Figure 2-11(b) shows the contour plot of the prediction uncertainty. It seems that the prediction uncertainty is minimum inside the interpolation region. The minimum point can be found by differentiating the standard error of prediction in Eq. (2.41)

$$\frac{\partial \sigma_y^2}{\partial x_1} = \frac{\hat{\sigma}^2}{2}(1 + 2x_1 + x_2) = 0$$

$$\frac{\partial \sigma_y^2}{\partial x_2} = \frac{\hat{\sigma}^2}{2}(1 + 2x_2 + x_1) = 0$$

Solving the above two equations solves for $x_1 = x_2 = -1/3$. The prediction uncertainty at this point is $\sigma_y = \hat{\sigma}/\sqrt{3}$. The minimum point is in fact the centroid of the interpolation region. The following Matlab code is used to plot Figure 2-11(b).

```
x=[-1 -1 1]; y=[-1 1 -1];
[X,Y]=meshgrid(-1:.1:1, -1:.1:1);
```

```
Z=sqrt(.5*(1+X+Y+X.^2+Y.^2+X.*Y));
v=linspace(0.6,1.8,7)
scatter(x,y,'filled');
grid on; hold on
[C,h]=contour(X,Y,Z,v);
clabel(C,h)
```

(c) When an additional sample is added at (1,1), the vector of basis functions and design matric is defined as

$$\boldsymbol{\xi} = \begin{Bmatrix} 1 \\ x_1 \\ x_2 \end{Bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad (\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{4}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The standard error of prediction in Eq. (2.41) is given as

$$\sigma_y = \hat{\sigma}\sqrt{\boldsymbol{\xi}^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}} = \hat{\sigma}\sqrt{0.25(1 + x_1^2 + x_2^2)}$$

The standard error of prediction at the four corner points is $\sigma_y = \sqrt{\widehat{3}}\sigma/2$, which is lower than the standard error of noise. This means that the surrogate is more accurate than the samples. The minimum prediction uncertainty occurs at the origin of the design space (0,0), whose value is $\sigma_y = \hat{\sigma}/2$. Figure 2-12 shows the contour plot of the prediction uncertainty. It is noted that the additional sample does not improve the low prediction uncertainty, but significantly reduces the large prediction uncertainty in the extrapolation region.

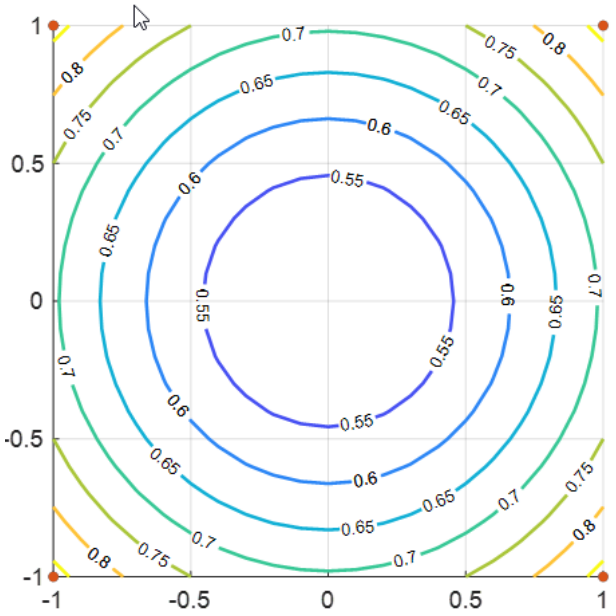


Figure 2-12: Contour of the standard error of prediction with four samples.

Prediction variance with variable noise

So far, we consider the case when noises of all samples are from an identical distribution $\sim N(0, \sigma^2)$. From this assumption, the prediction variance is determined as $V[\hat{y}(\mathbf{x}_p)] = \sigma^2\boldsymbol{\xi}(\mathbf{x}_p)^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x}_p)$. In some cases, however, noises at different locations may have different magnitudes. This is particularly true

when the magnitude of the noise is proportional to the magnitude of QoI. In such a case, the sample sensitivity in the previous subsection can be utilized to estimate the prediction variance. Let the noise variance of each sample point is σ_i^2. Then, the prediction variance can be defined as

$$V[\hat{y}(\mathbf{x}_p)] = \sum_{i=1}^{n_y} \left(\frac{\partial \hat{y}(\mathbf{x}_p)}{\partial y_i} \right)^2 \sigma_i^2 \tag{2.44}$$

Example 2-11
Consider the linear PRS in **Example 2-5**. (a) When all samples have the noise standard deviation σ, calculate the prediction variance at $x = 3$. (b) When all samples have different noise standard deviations, which sample has the most significant influence on the prediction variance at $x = 3$. Explain why.

Solution:
(a) In the case of linear PRS, the following matrices were defined:

$$\mathbf{X} = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{y} = \begin{Bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{Bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{Bmatrix} 1 \\ x \end{Bmatrix}$$

The linear PRS becomes

$$\hat{y}(x) = \boldsymbol{\xi}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = 0.925x$$

Prediction variance at $x = 3$ becomes

$$V[\hat{y}(x)] = \sigma^2\boldsymbol{\xi}^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi} = \sigma^2\{1, x\}\begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}\begin{Bmatrix} 1 \\ x \end{Bmatrix} = (0.2 + 0.1x^2)\sigma^2 = 1.1\sigma^2$$

Therefore, if all samples have the same noise standard deviation, the prediction variance at $x = 3$ is $1.1\sigma^2$.

(b) When all samples have different noise standard deviations, it would be necessary to use sample sensitivity in Eq. (2.44). First sample sensitivity at $x = 3$ can be calculated as

$$\frac{\partial \hat{y}}{\partial y_i} = \boldsymbol{\xi}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = 0.1[2 - 2x \quad 2 - x \quad 2 \quad 2 + x \quad 2 + 2x]$$
$$= 0.1[-4 \quad -1 \quad 2 \quad 5 \quad 8]$$

When all samples have the same noise standard deviation, Eq. (2.44) yields $V[\hat{y}(x)] = 1.1\sigma^2$, which is the same outcome with Part (a). If all samples have different noise standard deviation, the most influential sample to the prediction variance at $x = 3$ is y_5 whose sample sensitivity is highest. This is because the sample location $x = 2$ is closest to the prediction point $x = 3$.

2.7. Outliers

In Section 2.5, we discussed the stepwise elimination of insignificant coefficients based on their coefficient of variation. That is, if a coefficient has a large uncertainty, it means that the coefficient is difficult to identify and insignificant in prediction. The true function may not have the basis function and it justifies the removal of the coefficient. If a sample is suspicious, we may need systematic reasoning to

eliminate the sample from the fitting process. The same concept can be applied to identify wrong/erroneous samples.

Both in physical experiments and computer simulations, we have samples with large errors occasionally. In computer simulations, these may reflect failures of the solution algorithm, software implementation, or mistakes by the user of the software. Sample points with large errors are called outliers. It is important to detect them and either remove or repair them because they can have a large detrimental effect on the prediction accuracy of the surrogate.

A standard tool for detecting outliers is the Iteratively Reweighted Least Squares (IRLS) procedure. In order to understand their basis, let us consider first the weighted least square (WLS) procedure. Weighted least squares procedures minimize a weighted sum of the squares of the residuals. Linear regression in Section 2.3 corresponds to using the same weights for all the samples. There are many possible reasons for using WLS rather than standard least squares. We may have more confidence in some samples than in others. We may want to weigh more heavily points that are close to the region where we will need to predict the surrogate than points far away. The noise at some points may be known to be higher than at other points.

In these cases, instead of using Eq. (2.12) we use the weighted-root-mean-square error:

$$e_{wRMS} = \sqrt{\frac{1}{n_y}\sum_{i=1}^{n_y} w_i e_i^2} = \sqrt{\frac{1}{n_y}\mathbf{e}^T\mathbf{W}\mathbf{e}} \tag{2.45}$$

where, w_i is the weight associated with the ith sample, and $\mathbf{W}$ is a diagonal matrix with the weights on the diagonal. Minimizing the weighted RMS error, e_{wRMS} yields a modified set of normal equations

$$\mathbf{X}^T\mathbf{W}\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{W}\mathbf{y} \tag{2.46}$$

Iteratively reweighted least squares procedures weigh points with large residuals with small weights, with the weight decreasing with increasing magnitude of the residual. Then the WLS procedure is performed. If the point is an outlier, the surrogate model will move away from it, so its residual will increase. We will assign a smaller weight to the point and repeat the procedure. Eventually, outliers are likely to end up with zero or very low weight. There are several weighting schemes, see e.g., Myers and Montgomery, p. 671 [4]. One of the simplest is Huber's scheme, defined as

$$w_i = \begin{cases} 1 & \text{if } |e_i|/\hat{\sigma} \le 1 \\ |e_i|/\hat{\sigma} & \text{otherwise} \end{cases} \tag{2.47}$$

In practice, the procedures can be repeated until the weight of the outlier decreases enough, or once a sample is identified as an outlier, the procedures can be stopped and the samples can be removed in the regular linear regression process.

Example 2-12

Stress and strain are linearly related by $stress = E \cdot strain$, where Young's modulus E needs to be estimated based on four stress-strain measurements. For values of strains of 1, 2, 3, and 4 millistrains, stresses are measured by 9, 22, 36, and 39 ksi. Using a linear PRS model $stress = E \cdot strain$, identify Young's modulus after removing an outlier.

Solution:

Since we do not know which samples are an outlier, we perform regular linear regression with a constant weight $w_i = 1$. Denoting the stress by y as a QoI and the strain by x, the four samples are used to define the following normal equation:

$$\mathbf{X} = \begin{bmatrix}1\\2\\3\\4\end{bmatrix}, \mathbf{y} = \begin{Bmatrix}9\\22\\36\\39\end{Bmatrix}, \qquad \mathbf{X}^T\mathbf{X} = [30], \mathbf{X}^T\mathbf{y} = \{317\}, \qquad E = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = 10.569 \text{ Msi}$$

Note that there is no constant term in this PRS surrogate. The vector of residual errors is given as $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \{1.567, -0.867, 4.3, -3.267\}^T$. Therefore, the standard errors of noise and coefficient become

$$\hat{\sigma} = \sqrt{\frac{\mathbf{e}^T\mathbf{e}}{4-1}} = 3.2846, \qquad \hat{\sigma}_E = 3.2846\sqrt{\frac{1}{30}} = 0.6$$

The only residual error that is greater than $\hat{\sigma}$ is the third one. Therefore, its weight is calculated as $w_3 = 3.2846/4.3 = 0.764$.

Now, since samples have different weights, we perform weighted least-squares fitting with $\mathbf{W} = \text{diag}[1.0, 1.0, 0.764, 1.0]$ to yield

$$\mathbf{X}^T\mathbf{W}\mathbf{X} = [27.876], \mathbf{X}^T\mathbf{W}\mathbf{y} = \{295.512\}, \qquad E = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y} = 10.46 \text{ Msi}$$

Note that Young's modulus is slightly reduced. In the iterative reweighted least-squares process, the new vector of residual errors is calculated as $\mathbf{e} = \{1.457, -1.085, 4.628, 2.830\}^T$. The new linear PRS yields

$$\hat{\sigma} = \sqrt{\frac{\mathbf{e}^T\mathbf{W}\mathbf{e}}{4-1}} = 3.037, \qquad \hat{\sigma}_E = 3.037\sqrt{\frac{1}{30}} = 0.56$$

Overall, the standard error of the coefficient is reduced slightly. Among the new residual errors, only the third one is larger than the standard error of noise. Therefore, its weight is recalculated as $w_3 = 3.037/4.628 = 0.656$. It is noted that the weight of the third sample decreases further. One more iteration yields

$$\mathbf{X}^T\mathbf{W}\mathbf{X} = [26.904], \mathbf{X}^T\mathbf{W}\mathbf{y} = \{279.848\}, \qquad E = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y} = 10.40 \text{ Msi}$$

We can continue iteration until the residual errors become smaller than the standard error of noise. Or, we can be satisfied that the process has identified the third sample as an outlier and eliminated it. If we remove the third sample, we can identify $E = 9.95$ Msi, with the standard error of coefficient $\hat{\sigma}_E = 0.376$.

2.8. Statistical view of linear regression

So far, we developed the linear regression process as a minimization of an error between the samples and model predictions. This is indeed deterministic optimization and due to the quadratic nature of the sum-of-square errors, the optimization has a unique solution. At the same time, however, we also presented uncertainty associated with samples, regression coefficients, and predictions at unsampled points. These uncertainties stem from the fundamental assumption that samples include random noise. In this section, we will derive the uncertainty information from the statistical view of the linear regression process.

Although the purpose of surrogate modeling is to develop an approximate function using given samples, it would be better to express the samples in terms of the true model in order to explain their relationship from the statistical viewpoint. First, we assume that there is a true function $f(\mathbf{x}; \boldsymbol{\beta})$, which describes the behavior of QoI as a deterministic function of input variables $\mathbf{x}$ and model parameters $\boldsymbol{\beta}$. In particular, in the case of PRS surrogates, it is assumed that the true function is in the form of polynomials and the model parameters $\boldsymbol{\beta}$ become the coefficients of the polynomials. Second, it is assumed that the

PRS surrogate has an exact model form, but the coefficients are approximate. Therefore, the PRS surrogate can be represented by $f(\mathbf{x};\mathbf{b})$, where $\mathbf{b}$ is the approximate coefficients. Lastly, the samples are assumed to be generated from the true function $f(\mathbf{x};\boldsymbol{\beta})$ by adding random noise or some error defined by $\epsilon \sim N(0,\sigma^2)$. In this viewpoint, we can think of sample y at point $\mathbf{x}$ as a model given in the following form:

$$y = f(\mathbf{x};\boldsymbol{\beta}) + \epsilon \tag{2.48}$$

That is, a sample has a deterministic part (true function) and a probabilistic part (noise or error).

Since the noise is assumed to be normally distributed with a zero mean and variance σ^2, the sample in Eq. (2.48) follows a normal distribution $y|\mathbf{x} \sim N(f(\mathbf{x};\boldsymbol{\beta}),\sigma^2)$. Here, the notation, $y|\mathbf{x}$, means a conditional probability of y given $\mathbf{x}$. Let $p(y|\mathbf{x})$ be the conditional probability density function (PDF), then it can be written as

$$p(y|\mathbf{x}) = N(f(\mathbf{x};\boldsymbol{\beta}),\sigma^2) \tag{2.49}$$

In this viewpoint, each sample is a random variable whose mean is the true function and variance σ^2.

Now, the main task of surrogate modeling is to estimate the unknown coefficient $\boldsymbol{\beta}$. In the case of linear regression, the unknown coefficients are determined by minimizing the sum-of-square errors. From the statistical viewpoint, instead of minimizing a measure of error, the following question is asked: how likely is it that the samples can be obtained given the input variable $\mathbf{x}$ and parameter $\boldsymbol{\beta}$? Such likelihood of obtaining the sample is represented in the form of conditional probability $p(y|\mathbf{x})$ in Eq. (2.49). Since there are n_y numbers of samples, the likelihood of obtaining all samples can be written as

$$p\left(y_1, y_2, \cdots, y_{n_y} \middle| \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{n_y}\right) \tag{2.50}$$

This is indeed the condition joint PDF of n_y numbers of random variables.

In PRS surrogate modeling, it is assumed that all samples are uncorrelated to each other. More specifically, they are independent and identically distributed (iid). That means, the random noises in y_i and y_j come from the same distribution$\sim N(0,\sigma^2)$, but their realizations ϵ_i and ϵ_j are independent. In such a case, the conditional joint PDF in Eq. (2.50) can be obtained by the product of all conditional PDFs of individual samples. That is, the likelihood function of all samples can be written as

$$\begin{aligned} p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma) &= \prod_{i=1}^{n_y} p(y_i|\mathbf{x}_i) = \prod_{i=1}^{n_y} N(f(\mathbf{x}_i;\boldsymbol{\beta}),\sigma^2) \\ &= \left(\sigma\sqrt{2\pi}\right)^{-n_y} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n_y}\left(y_i - f(\mathbf{x}_i;\boldsymbol{\beta})\right)^2\right] \end{aligned} \tag{2.51}$$

It is important to understand the difference between the conditional joint PDF and the likelihood function. Although both have the same expression, the interpretations are different. In the case of conditional joint PDF, $p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma)$ means the PDF value of $\mathbf{y}$ for given parameters $\mathbf{X},\boldsymbol{\beta},\sigma$. In this case, $\mathbf{y}$ is varied to obtain different probability density values. On the other hand, in the case of the likelihood function, $p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma)$ means the probability density value to obtain $\mathbf{y}$ for given $\mathbf{X},\boldsymbol{\beta},\sigma$. In this case, $\mathbf{y}$ is fixed, and $\mathbf{X},\boldsymbol{\beta},\sigma$ are varied to obtain different likelihood values at different parameters.

Figure 2-13 shows the relationship between the PDF and likelihood function in the case of one sample with one parameter. Figure 2-13(a) shows three PDFs of sample y with three different parameter values β_1,β_2,β_3. The three PDFs have the same shape but with different mean values. This happens because we assume that sample y has a mean $f(\mathbf{x};\boldsymbol{\beta})$, which depends on the parameter β. If the variance σ^2 is also included as a parameter, the shape of the PDF will also change.

Now, when an actual sample turns out to be y_1 from the experiment, the PDFs have different values depending on parameter values (the three red circle markers). This is basically likelihood. When $\beta = \beta_1$, the probability density to obtain sample y_1 is A, when $\beta = \beta_2$ it is B, etc. Therefore, the likelihood of obtaining sample y_1 can be plotted by varying the parameter β as shown in Figure 2-13(b), which is the likelihood function. It is noted that sample y is a variable in PDF, while parameter β is a variable in the likelihood function. In general, the shape of PDF and that of likelihood are different.

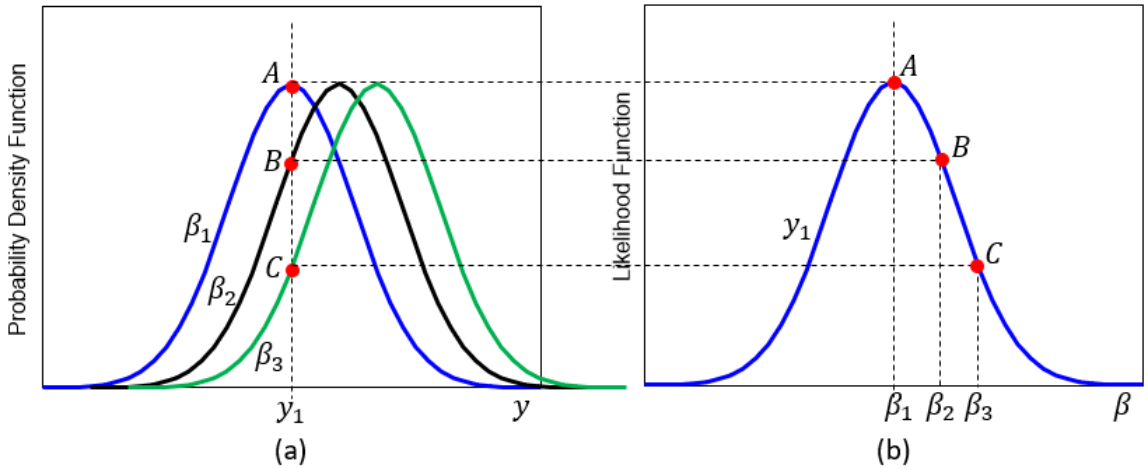


Figure 2-13: Probability density function versus likelihood function.

Different from the least-squares method in linear regression, the idea behind the likelihood function is that the likelihood function in Figure 2-13(b) is our knowledge of the unknown parameter. That is, we cannot estimate the unknown parameter deterministically but only in the form of a probabilistic distribution. Because sample y has uncertainty due to random noise, we cannot estimate the parameter exactly. Instead, the probability that the parameter is β_1 is high, while the probability that the parameter is β_3 is low. If we know sample y_1 is exact and there is no randomness; i.e., $\sigma = 0$, we can determine the unknown parameter β_1 exactly because the PDF is infinity at $y = y_1$ and zero everywhere else.

It is also interesting to note that the PDF in Figure 2-13(a) is a probability distribution of a random variable y, which is often known as aleatory uncertainty. That means, different samples will have different values every time when the sample is drawn. On the other hand, the distribution in Figure 2-13(b) does not represent any randomness. The model parameter β is not random, but uncertain. This type of uncertainty is called epistemic uncertainty. It means, the parameter is deterministic but we do not know its value exactly. The top point A in Figure 2-13(b) means that the probability that the actual parameter value being β_1 is highest, while the point C means that the probability that the actual parameter value being β_3 is low based on the proof that sample has a value of y_1. Therefore, the distribution in Figure 2-13(b) should be understood as the probability that the parameter value is an accurate one given sample y_1. If the likelihood has a narrow distribution, it means the information is accurate and uncertainty in the parameter is low. On the other hand, if the likelihood is widely distributed, it means that the information is vague and it is difficult to identify accurate parameters.

When multiple samples are present, the individual likelihood functions are multiplied together. For example, when two samples, y_1 and y_2, are present, the PDF in Figure 2-13(a) becomes a two-dimensional joint PDF (the plot will be a three-dimensional bell shape). However, the likelihood function in Figure 2-13(b) will still be a one-dimensional function of β. If two samples y_1 and y_2 are close together, the distribution in Figure 2-13(b) will be narrowed, which means the information is accurate. If

two samples are significantly different, the likelihood plot will be wide and the parameter cannot be estimated accurately. On the other hand, when multiple parameters are present in the PRS model, the likelihood function in Figure 2-13(b) will be a multi-dimensional bell-shaped plot. Therefore, the likelihood function given in Eq. (2.51) can be viewed as a mapping from n_y-dimensional joint PDF to n_β-dimensional likelihood function.

Even if the likelihood function in Figure 2-13(b) is given in the form of distribution, it is still valuable to obtain a single value of the parameter that can represent the best estimate. In fact, this estimate corresponds to the mode of the distribution in Figure 2-13(b). That is, the peak of the distribution is the best deterministic estimate of the parameter. Since the peak value corresponds to the maximum likelihood function, it is referred to as the maximum likelihood estimate. It is possible that the likelihood function in Eq. (2.51) is differentiated to obtain the maximum point, it would be better to work with a logarithm of it, which is called the log-likelihood function. This is because the product of exponential likelihood functions increases very fast and becomes a highly nonlinear function. The log-likelihood function can be defined as

$$L = \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \sigma) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n_y}|y_i - f(\mathbf{x}_i; \boldsymbol{\beta})|^2 - n_y \log \sigma\sqrt{2\pi} \tag{2.52}$$

Since logarithm is a monotonic function, even if the likelihood function in Eq. (2.51) and the log-likelihood function in Eq. (2.52) may have different values, the parameters, $\boldsymbol{\beta}$ and σ, that maximize both functions are the same.

In order to have the maximum likelihood estimate, the true function is expressed as a linear combination of basis functions as$f(\mathbf{x}_i; \boldsymbol{\beta}) = \boldsymbol{\xi}(\mathbf{x}_i)^T\boldsymbol{\beta}$. Then, the sum in the first term on the right-hand side of Eq. (2.52) becomes $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$. Then, the maximum likelihood estimate determines the model parameter by differentiating the log-likelihood function as

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \frac{1}{\sigma^2}(\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\boldsymbol{\beta}) = \mathbf{0} \tag{2.53}$$

It is interesting to note that even if σ is also considered a model parameter, the determination of parameter $\boldsymbol{\beta}$ is independent of σ. This is because σ is the same for all samples. Also, it can be seen from Figure 2-13(b) that the large σ may cause a wide distribution, but the peak value will be the same.

An important conclusion from Eq. (2.53) is that the parameters obtained from the maximum likelihood estimate are identical to the parameters obtained from linear regression in Eq. (2.14), as

$$\tilde{\mathbf{b}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2.54}$$

Here $\tilde{\mathbf{b}}$ is used instead of $\boldsymbol{\beta}$ because the right-hand side includes the random variable $\mathbf{y}$. Therefore, $\tilde{\mathbf{b}}$ is an uncertain variable with a joint PDF given as $\sim N(\mathbf{b}, \boldsymbol{\Sigma}_\mathbf{b})$. Equation (2.54) shows how aleatory uncertainty (randomness) in samples is converted into epistemic uncertainty in the regression coefficients. In addition, the variance of noise can be estimated by differentiating the log-likelihood function with respect to σ, as

$$\hat{\sigma}^2 = \frac{1}{n_y}\left(\mathbf{y} - \mathbf{X}\tilde{\mathbf{b}}\right)^T\left(\mathbf{y} - \mathbf{X}\tilde{\mathbf{b}}\right) = \frac{SS_e}{n_y} \tag{2.55}$$

which is the same as Eq. (2.20) except that the free degrees-of-freedom $n_y - n_\beta$ is used for the unbiased estimate.

The second-order derivative of the log-likelihood function with respect to parameter $\boldsymbol{\beta}$ is called the Hessian matrix or information matrix. By differentiating Eq. (2.53), the information matrix can be obtained as

$$\frac{\partial^2 L}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}} = -\frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X} \tag{2.56}$$

The uncertainty in the regression coefficients can be defined as a covariance matrix. The definition of a covariance matrix is $cov[\tilde{\mathbf{b}}] = \boldsymbol{\Sigma}_\mathbf{b} = E[\tilde{\mathbf{b}}\tilde{\mathbf{b}}^T] - E[\tilde{\mathbf{b}}]E[\tilde{\mathbf{b}}^T]$. The first term on the right-hand side can be expanded using the estimated coefficient in Eq. (2.54): $E[\tilde{\mathbf{b}}\tilde{\mathbf{b}}^T] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T E[\mathbf{y}\mathbf{y}^T]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}$. This is because only **y** is random in Eq. (2.54), all other terms can come out of the expectation operator. The mean of sample matrix term can be simplified as $E[\mathbf{y}\mathbf{y}^T] = E\left[\left(\mathbf{X}\tilde{\mathbf{b}} - \boldsymbol{\epsilon}\right)\left(\mathbf{X}\tilde{\mathbf{b}} - \boldsymbol{\epsilon}\right)^T\right] = \mathbf{X}\mathbf{b}\mathbf{b}^T\mathbf{X}^T + \sigma^2\mathbf{I}$. The second term is straightforward as $E[\tilde{\mathbf{b}}]E[\tilde{\mathbf{b}}^T] = \mathbf{b}\mathbf{b}^T$. Therefore, combining these two terms, the covariance matrix of the coefficients becomes

$$\boldsymbol{\Sigma}_\mathbf{b} = \sigma^2(\mathbf{X}^T\mathbf{X})^{-\mathbf{1}} \tag{2.57}$$

which is identical to the one given in Eq. (2.38). It is interesting to note that the covariance matrix is the negative of the inverse of the information matrix in Eq. (2.56).

The last uncertainty that we used is the prediction variance. From the statistical viewpoint, the surrogate prediction can be written as $\hat{y}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T\tilde{\mathbf{b}}$. In this expression, the prediction is also an uncertain variable because of $\tilde{\mathbf{b}}$. However, since $\boldsymbol{\xi}(\mathbf{x})^T$ is a deterministic function, the variance of $\hat{y}(\mathbf{x})$ can be written as

$$V[\hat{y}(\mathbf{x})] = \boldsymbol{\xi}(\mathbf{x})^T\boldsymbol{\Sigma}_\mathbf{b}\boldsymbol{\xi}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T\sigma^2(\mathbf{X}^T\mathbf{X})^{-\mathbf{1}}\boldsymbol{\xi}(\mathbf{x}) \tag{2.58}$$

Therefore, the standard error of prediction can be written as

$$\sigma_y(\mathbf{x}) = \hat{\sigma}\sqrt{\boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-\mathbf{1}}\boldsymbol{\xi}(\mathbf{x})} \tag{2.59}$$

which is identical to the one given in Eq. (2.41). In general, the standard error of prediction is used to find the confidence intervals of the prediction. For example, 95% confidence intervals can be written as

$$CI_{95} = \hat{y}(\mathbf{x}) \pm 2\sigma_y(\mathbf{x}) \tag{2.60}$$

Example 2-13

Generate 50 equally spaced samples in the design space $x \in [-5, 5]$ from the true function $y(x) = 5x^3 - x^2 + x$ and add random noise from a normal distribution $\sim N(0, 100^2)$. Fit the samples using linear, cubic, 6th-order, and 10th-order PRS. Plot the mean prediction with 95% confidence intervals.

Solution:

The following MATLAB code is used to fit the PRS surrogate model and plot the prediction and confidence intervals. The code requires removing the comment symbol % in front of the design matrix **X** for the corresponding order of PRS. First, random noises are generated from a normal distribution. They are shifted such that their mean is zero. Even if the standard deviation of 100 is used to generate noise samples, the sample standard deviation turns out to be 126.3. MATLAB function `regress` is used to calculate the regression coefficients **b** as well as residuals **r**. The standard error of noise is calculated using the residuals, and the standard error of prediction is calculated at all sample points. Figure 2-14 shows the PRS predictions and 95% confidence intervals for linear, cubic, 6th-order, and 10th-order PRS. It is obvious that the linear PRS failed to follow the trend of the true function, and the estimated uncertainty is not consistent (30 samples were out of 95% confidence intervals). The cubic and 6th-order PRS seem to follow the trend of true function well, while the 10th-order PRS seems overfitting samples. In order to compare these surrogates R^2 and R_a^2 are calculated in Table 2-2. When R^2 is considered, it looks that the 10th-order PRS fits the best. But this is due to the fact that more coefficients are used.

Based on R_a^2, the cubic PRS turns out to be the best, which has the same order of polynomials with the true function.

Table 2-2: Comparison of R^2 and R_a^2 for four PRS surrogates.

	1st-order PRS	3rd-order PRS	6th-order PRS	10th-order PRS
$\hat{\sigma}$	186.3	120.8	123.5	122.3
R^2	0.57	0.8267	0.8306	0.8494
R_a^2	0.561	0.8154	0.8070	0.8108

```
rng default;                                  % Control random number sequence
x=linspace(-5,5,50)';                        % Equally spaces sample locations
ytrue=5*x.^3 - x.^2 + x;                                  % True function
noise=100*randn(50,1); noise=noise-mean(noise);     % Unbiased random noise
y=ytrue+noise;                                          % Generate samples
%X=[ones(50,1) x];
X=[ones(50,1) x x.^2 x.^3];                               % Design matrix
%X=[ones(50,1) x x.^2 x.^3 x.^4 x.^5 x.^6];
%X=[ones(50,1) x x.^2 x.^3 x.^4 x.^5 x.^6 x.^7 x.^8 x.^9 x.^10];
[b,bint,r,rint,stats]=regress(y,X);                         % Fitting PRS
yfit=X*b;                                  % Prediction at sample points
s=sqrt(r'*r/(50-size(X,2));                    % Standard error of noise
sy=s.*diag(sqrt(X*inv(X'*X)*X'));             %Standard error of prediction
plot(x,y,'+',x,yfit,'b',x,yfit-2*sy,'k',x,yfit+2*sy,'k');        % Plotting
```

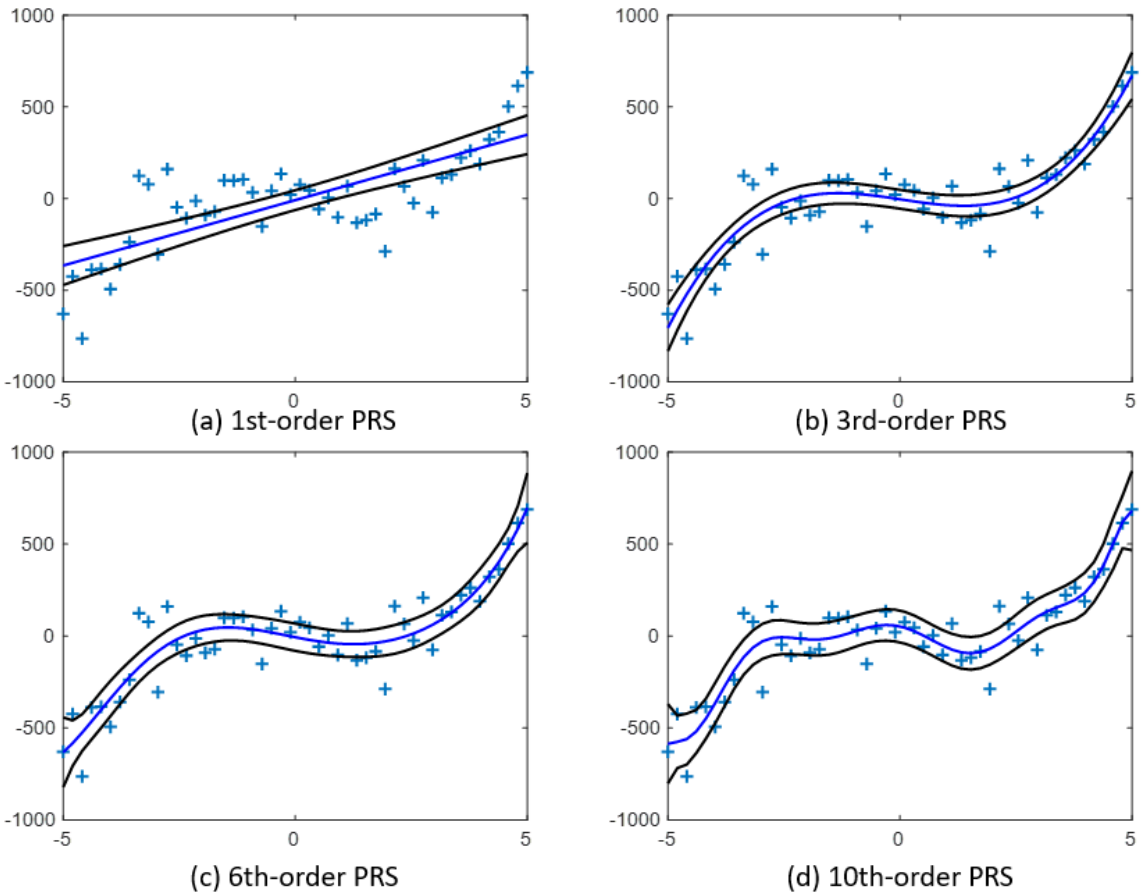


Figure 2-14: Confidence intervals of different surrogate models.

2.9.Exercise

1. Fit a linear function in Figure 2-1 with 10, 30, 50, and 100 equal-interval samples. Show that e_{RMS} of the fitted curve converges to zero as the number of samples increases. Explain why that happens using the mean of noise samples.
2. Generate 21 equal-interval samples for $x \in [0, 2]$ from the true function $y(x) = x^2$ and add random noise from the standard normal distribution $\sim N(0,0.1^2)$. Shift the noise samples such that the samples have a zero mean. Fit these samples using a linear polynomial $\hat{y}(x) = b_1 + b_2 x$ and estimate e_{RMS} of the fitted model. Discuss the reason for the difference between the standard deviation of noise and e_{RMS}.
3. In a curve-fitting example with a true function $y = x$, noisy data are fitted to a linear polynomial $\hat{y} = 1.06x$. In addition, the data at $x = 10$ was $y_{10} = 11$. What are (a) ϵ, (b) ϵ_{10}, and (c) the surrogate error at $x = 10$?
4. Generate 30 equal-interval samples for $x \in [1, 30]$ using a true function $y(x) = x$ and add random noise from $\sim N(0,1^2)$. Using the three error metrics in Eqs. (2.7)-(2.9) to fit a linear PRS. Compare the three PRSs using the three error metrics as in Table 2-1. Hint: use the `fminsearch` function in Matlab.
5. Repeat Problem 4 with the surrogate $\hat{y}(x) = bx$.
6. Repeat Problem 4 with 10 points and compare the accuracy of the fit with respect to the true function.
7. Find other error metrics for a fit besides the three discussed in Eqs. (2.7)-(2.9).
   For $p = 1$, it becomes absolute error norm, for $p = 2$, it becomes RMS error, and as p approaches infinity, the p-normal approaches the maximum norm $\|\mathbf{e}\|_\infty = \max|e_i|$. Therefore, a large p can be used to approximate the maximum error norm.
8. Check the accuracy of the quadratic PRS in **Example 2-2** in the region $1.5 \le D \le 3.5$, $3 \le H \le 7$. Find the maximum error, average error and RMS error (a) for the 9 sample points, and (b) for the entire region (using the analytical expression from **Example 2-2**). You may cover the domain with a grid of (20×20) points and calculate the error in each one of the 400 points. (c) Calculate the error using the PRESS procedure and compare it to the result of part (b)
9. Using the samples in **Example 2-2**, use backward elimination to find an incomplete quadratic PRS with the highest R_a^2. Then, check for accuracy of the fit compared to the analytical profit per can over the entire region, and compare to the results obtained in Problem 8.
10. Consider the following experimental samples $(x, y) = (1,1), (2,2), (3,5), (4,9)$. (a) Construct a linear PRS $\hat{y}(x) = b_1 + b_2 x$. (b) Estimate R_a^2 and e_{PRESS}. (c) If the quadratic PRS is $\hat{y}(x) = 1.25 - 1.05x + 0.75x^2$, compare the quality of quadratic PRS with linear PRS using e_{PRESS}. For quadratic PRS, use the following idempotent matrix:

$$E = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \begin{bmatrix} .95 & .15 & -.15 & .05 \\ .15 & .55 & .45 & -.15 \\ -.15 & .45 & .55 & .15 \\ .05 & -.15 & .15 & .95 \end{bmatrix}$$

11. A function is given at 3 data points: $y(-1) = 0$, $y(0) = 0$, and $y(1) = 3$. Fit a least square constant and linear polynomials to the data. Calculate the RMS error of the two curve fits. The linear curve fit has a smaller error but check whether it could be expected to be a better predictor of the true function.

12. A function $y = bx$ is fitted using two sample points, $y_1 = y(1) = 10$, $y_2 = y(2) = 25$. (a) Calculate the cross-validation error at the first point from the definition. (b) If the true value of the coefficient b is 11 ($y_{\text{true}} = 11x$) and the fitted value is 12, $\hat{y} = 12x$, calculate the value of ϵ, e, and the surrogate error at $x = 1$. (c) Calculate the coefficient b to check that its value is indeed 12.
13. A function $y = b/(1 + x)$ is fitted using two sample points, $y_1 = y(0) = 10$, $y_2 = y(1) = 4$. (a) Estimate the coefficient b using linear regression. (b) Calculate the cross-validation error at the first point from the definition. (b) The noise in the samples has been determined to be normally distributed with a standard deviation of 1.1. Use that to estimate the standard error in b. (c) Provide a better choice of b if the objective is to minimize the maximum error rather than the regression fit.
14. A function $y = b(1 - x)/(1 + x)$ is fitted using two sample points, $y_1 = y(0) = 0$, $y_2 = y(2) = -20$. (a) Estimate the coefficient b using linear regression. (b) Calculate the cross-validation error at $x = 2$ from definition.
15. In **Example 2-3**, calculate the RMS error between the true function and fitted function with 50 equally spaced points (a) in $x \in [0, 10]$ and (b) in $x \in [-2, 12]$. Discuss why the RMS error in (b) is larger than (a).
16. Generate 20 equally-spaced samples of the true function $y(x) = x^3 - x$ in the design space $x \in [-1.5, 1.5]$ with random noise $\sim N(0,0.5^2)$. Fit the samples using (a) linear PRS and (b) cubic PRS. Compare the two surrogates in terms of the RMS error between the predictions and samples as well as the RMS error between the predictions and the true function.
17. Prove the relationship in Eq. (2.24).
18. The sample pairs (0, 0), (1, 1), and (2,1) represent strain (millistrains) and stress (ksi) measurements. (a) estimate Young's modulus using regression, and (b) calculate the error in Young's modulus using cross-validation both from the repeated fitting and from the formula with one fitting.
    This yields $e_{p3} = y_3 - \hat{y}(2) = -1.0$. Therefore, the prediction errors $\mathbf{e}_p = \{0, 0.5, -1\}^T$ is identical to the one-fit case. Therefore, $e_{PRESS} = 0.7606$ will be the same as well.

19. The pairs $(1,1), (2,2), (4,3)$ represent strain (millistrains) and stress (ksi) measurements. The relationship is given as stress = E*strain, where E is Young's modulus. Estimate Young's modulus using the three error measures: e_{RMS}, e_{av}, and $e_{\max}$. Estimate the error in Young's modulus using cross-validation of the three error measures. Use the range of Young's modulus $0.75 \le E \le 1$.

20. Two samples (1, 1) and (4, 3) represent strain (millistrains) and stress (ksi) measurements. For the material tested $\sigma = k\sqrt{\epsilon}$, (a) estimate k using regression, (b) estimate noise in samples, (c) estimate the standard error in the estimate of k, and (d) compare the standard error to the estimate k by cross-validation.

21. Two samples $y(0) = 10.0, y(1) = 4.0$ are fitted by $\hat{y}(x) = b/(1 + x)$. (a) Estimate the coefficient b via regression. (b) The noise in the samples has been determined to be normally distributed with a standard deviation of 1.1. Use that to estimate the standard error of the coefficient. (c) If the maximum error is used for the fitting process, estimate a better coefficient than (a). Explain how you obtain the coefficient.
22. Two samples $y(0) = 0.0, y(2) = -10.0$ are fitted by $\hat{y}(x) = b(1 - x)/(1 + x)$. (a) Estimate the coefficient b via regression. (b) Calculate the cross-validation error at $x = 2$ from the definition (not using E).
23. Repeat **Example 2-7** using only data at $x = 3, 6, 9, \cdots, 30$.

24. We are trying to estimate the average cost of a gadget based on three sample points \$110, \$120, and \$130. You can think of it as a simple regression problem of fitting a polynomial $\hat{y} = b$, to three sample points. (a) Perform the linear regression fit to the samples, which is to estimate b from the regression equations. (b) Calculate the cross-validation error for the third sample from the definition (not the formula). (c) If the true average (from a very large number of samples) is \$125, calculate the values of noise ϵ, residual e, and the surrogate error for the third sample in Part (a).
25. A PRS is given in the following model form: $\hat{y}(x) = bx$. Two samples are given as $y(1) = 10, y(2) = 25$. (a) Calculate the cross-validation error at the first point from the definition, not the formula. (b) If the true value of b is 11 ($y_{true}(x) = 11x$) and the fitted value is 12, $\hat{y}(x) = 12x$, calculate the values of ϵ, e, and the surrogate error at $x = 1$. (c) Calculate b to check that its value is indeed 12.
26. The pairs (1,1) and (3, 2) represent strain (millistrains) and stress (ksi) measurements. Denoting the strain by x, the stress by y, and Young's modulus by b, we assume that the true relationship between the stress and strain is $y = bx$. (a) Calculate the linear regression fit to the samples; that is, what the estimate of b is. (b) Calculate the cross-validation errors, from the definition (not the formula). (c) If the true value of Young's modulus is 1, calculate the values of ϵ, e, and the surrogate error at $x = 3$.
27. The following table shows 15 test results with two variables. Since the magnitudes of the two variables are significantly different, it is a good idea to scale them such that $-1 \le \tilde{x}_1, \tilde{x}_2 \le 1$ where $\tilde{x}_i = \left(x_i - [x_i^{max} + x_i^{min}]/2\right)/\left([x_i^{max} - x_i^{min}]/2\right)$.
    (a) Construct a linear response surface using scaled variables: $\hat{y}(\tilde{x}_1, \tilde{x}_2) = b_1 + b_2\tilde{x}_1 + b_3\tilde{x}_2$.
    (b) Calculate root-mean-square, average, and maximum errors.
    (c) Estimate the variance of random noise, coefficient of multiple determination, adjusted multiple determination, and PRESS.
    (d) Construct a quadratic response surface: $\hat{y}(\tilde{x}_1, \tilde{x}_2) = b_1 + b_2\tilde{x}_1 + b_3\tilde{x}_2 + b_3\tilde{x}_1^2 + b_4\tilde{x}_2\tilde{x}_1 + b_5\tilde{x}_2^2$.
    (e) Evaluate the quality of the quadratic PRS compared with the linear PRS.
    (f) Perform backward elimination once for the least confident coefficient.

Observation	x_1	x_2	y
1	195	4.0	1004
2	255	4.0	1636
3	195	4.6	852
4	255	4.6	1506
5	225	4.2	1272
6	225	4.1	1270
7	225	4.6	1269
8	195	4.3	903
9	255	4.3	1555
10	225	4.0	1260
11	225	4.7	1146
12	225	4.3	1276
13	225	4.72	1225
14	230	4.3	1321

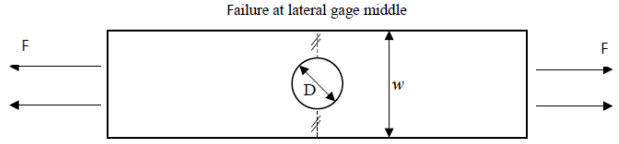
28. Repeat **Example 2-10** when the four sample locations are not at the corners but at $(\pm 0.7, \pm 0.7)$. Discuss the difference from the results in **Example 2-10**.
29. For a grid of 3×3 sample points in a two-dimensional design space $x_1, x_2 \in [-1, 1]$, compare the contours of the standard error of prediction for linear and quadratic PRS.
30. The Branin-Hoo function is a popular analytical function that is used for testing surrogate models [21]. The functional form is defined in $x_1 \in [-5, 10]$ and $x_2 \in [0, 15]$.

$$f(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi^2}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10$$

(a) Generate an equally-spaced 6×6 grid in the design space, among which use only internal $4 \times 4 = 16$ points as samples. Use the analytical function above to evaluate the samples. No need to add random noise.

(b) Fit the samples using cubic PRS with 10 coefficients. Evaluate the accuracy of the surrogate using $e_{max}, e_{av}, e_{RMS}, R^2, R_a^2$, and e_{PRESS}.

31. Tensile tests of composite plates with a hole are performed at different configurations as shown in the table. Multiple nominally identical specimens were tested for different ratios of plate width w to hole diameter D and for different fractions R_{45} of the thickness of the plate occupied by $\pm 45°$ plies. The results were collected for 12 combinations of the two variables, and the failure stress (based on the cross-section without the hole) is given in the table. The standard deviation between nominally identical specimens is given for information only. For fitting a surrogate use only the mean. (a) Fit the samples using a linear and quadratic PRS. (b) Compare the two surrogates on the basis of global measures ($R_a^2, \hat{\sigma}$, and e_{PRESS}) and the maximum prediction variance in the domain.



No.	1	2	3	4	5	6	7	8	9	10	11	12
R45	0.2	0.2	0.2	0.2	0.5	0.8	0.5	0.5	0.5	0.8	0.8	0.8
w/D	3	4	6	8	6	6	3	4	8	3	4	8
Mean(ksi)	87.35	91.46	97.44	100.12	71.07	50.30	58.11	67.62	72.39	37.68	42.68	51.84
Std(ksi)	3.55	5.42	5.20	4.90	2.84	1.68	2.48	2.70	3.28	1.34	0.85	1.33

32. Assuming a linear model; i.e., $y = \hat{y} + \epsilon = b_1 + b_2 x + \epsilon$, where $\hat{y}$ is the regression model and ϵ is the error with independent normal distribution; i.e., $\epsilon \sim N(0, \sigma^2)$. (a) Compute 95% confidence interval of $\mathbf{b}$; i.e., computer 2.5% and 97.5% percentiles of $\mathbf{b}$. (b) Compute a 95% confidence interval of $\hat{y}$ at $x = 0.2$. (c) Plot $\hat{y}$ and confident interval of $\hat{y}$ in the same graph $\in [0,1]$. Samples are given as

```
x=[0 0.2 0.4 0.6 0.8 1.0]';
y=[0.4662, 0.5844, 0.7845, 0.8007, 0.9028, 0.8995]';
```

33. In order to fit a linear polynomial $y(x) = b_1 + b_2 x$, six equally-spaced samples are generated in the design space of $x \in [0,1]$. From the true model, the samples have normally distributed noise with a mean of zero and a standard deviation of 0.1. The random number generator in Matlab generates the following noise samples $(x, \epsilon) = (0,0.119), (0.2, -0.04), (0.4,0.033), (0.6,0.017), (0.8,0.019), (1.0, -0.072)$. Add the noise samples to the function $y = f(x) = x$, and fit it to a linear polynomial. Calculate the error measures $e_{RMS}, R^2, R_a^2, e_{PRESS}$ and compare e_{RMS} and e_{PRESS} to the RMS error calculated analytically between the line $y = x$ and the fitted line.
34. Repeat Problem 33 when the true function is $y = x^2$. (a) use a linear PRS. (b) use a quadratic PRS.
35. .

36. .