6 Polynomial Regression: Making Linear Models Go Nonlinear

6.1 From Straight Lines to Curved Fits

6.1.1 Why Linear Models Fall Short

In the multivariable linear regression we discussed earlier, the independent variables x_1, x_2, … , x_D are typically raw features that can be directly measured, such as height, weight, or age. However, in practical modeling, we are not limited to these original linear features.

Sometimes, a single feature is insufficient to capture complex nonlinear relationships in the data, as illustrated in Figure 1. In such cases, we can "engineer" new features from the original ones to help the model capture these patterns. Polynomial regression is one common approach, where higher-order nonlinear features are added to a linear regression model.

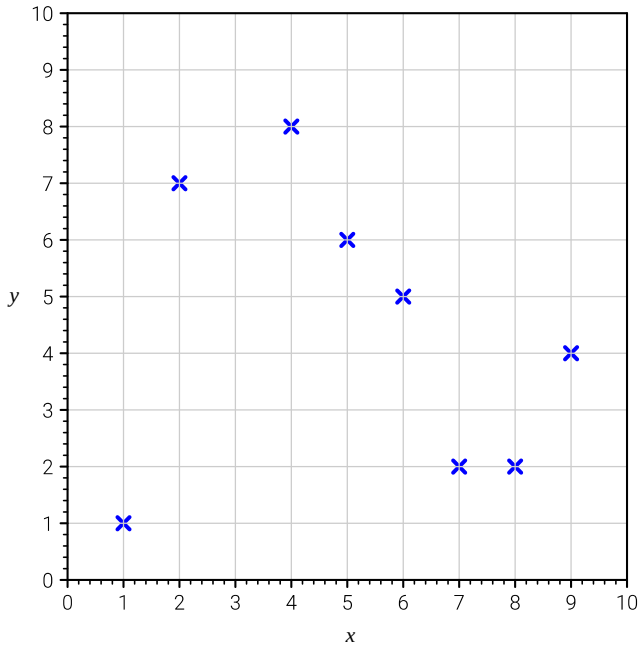

Figure 1. Scatter plot showing a nonlinear relationship between the original feature and the target

6.1.2 Creating Nonlinear Features from Linear Inputs

Even when we have only a single variable x, we can create new features such as x^2, x^3, and so on, expanding our feature set to 1, x , x^2, x^3 (Figure 2). This gives rise to polynomial regression. A univariate polynomial regression model of degree m m can be written as:

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \cdots + b_m x^m \qquad (1)$$

Here, each term x^k for $k \geqslant 2$ is a "constructed" feature derived from the original variable. When $m = 1$, this reduces to standard univariate (simple) linear regression, showing that linear regression is just a special case of polynomial regression.

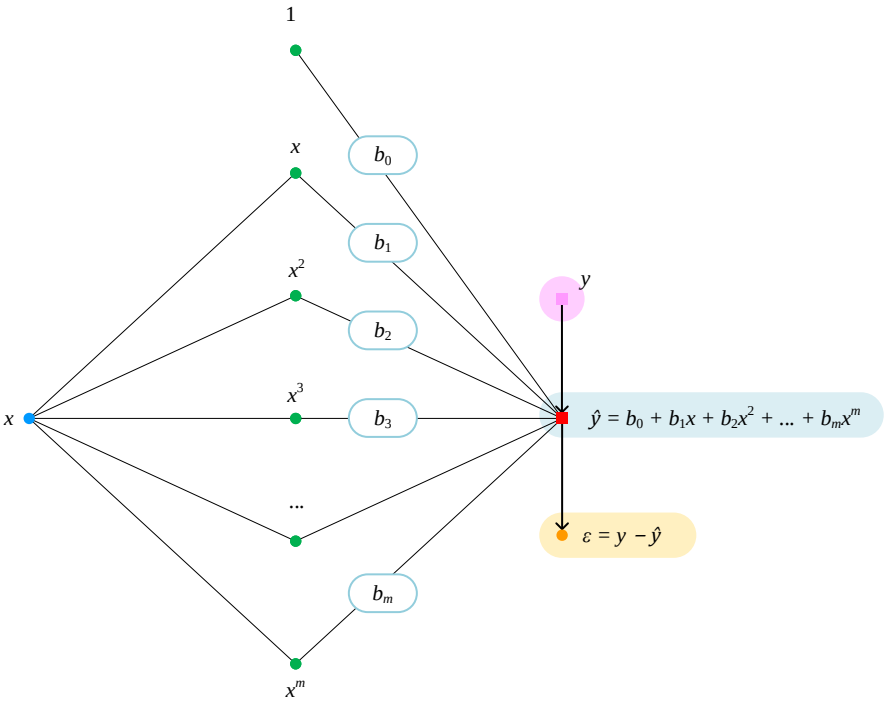

Figure 2. Univariate polynomial regression using engineered features x , x^2, x^3… to fit nonlinear data

Increasing the polynomial degree allows the model to fit more complex patterns in the data, but a degree that is too high can easily lead to overfitting, where the model performs very well on training data but poorly on new, unseen data. This trade-off between flexibility and overfitting will be discussed later in this chapter.

6.2 Expanding the Feature Space

6.2.1 Building the Polynomial Design Matrix

From a data perspective, polynomial regression involves not only using the original values of the independent variable but also including its higher-order powers as additional features. This allows the model to better capture nonlinear patterns in the data, as illustrated in Figure 3.

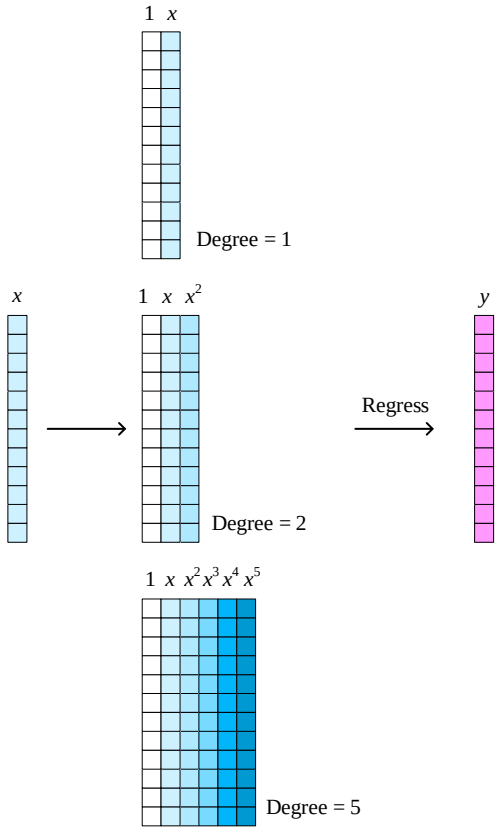

Figure 3. Design matrix for univariate polynomial regression, including higher-order features

For example, in a univariate fifth-degree polynomial regression, the design matrix $\boldsymbol{X}$ can be written as:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{x} & \boldsymbol{x} \odot \boldsymbol{x} & \cdots & \boldsymbol{x} \odot \boldsymbol{x} \odot \boldsymbol{x} \odot \boldsymbol{x} \odot \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} 1 & x^{(1)} & \left(x^{(1)}\right)^2 & \cdots & \left(x^{(1)}\right)^5 \\ 1 & x^{(2)} & \left(x^{(2)}\right)^2 & \cdots & \left(x^{(2)}\right)^5 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{(n)} & \left(x^{(n)}\right)^2 & \cdots & \left(x^{(n)}\right)^5 \end{bmatrix} \tag{2}$$

Here, terms like $\boldsymbol{x} \odot \boldsymbol{x}$ represent the element-wise product, also called the Hadamard product. This operation multiplies corresponding elements of two matrices (or vectors) of the same shape, producing a matrix of the same shape as the output. Recall that vectors can be considered as special cases of matrices.

With this design matrix, the target vector y y can be expressed as a linear combination of the columns of $\boldsymbol{X}$ and the coefficient vector $\boldsymbol{b}$:

$$\underbrace{\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} 1 & x^{(1)} & \left(x^{(1)}\right)^2 & \cdots & \left(x^{(1)}\right)^5 \\ 1 & x^{(2)} & \left(x^{(2)}\right)^2 & \cdots & \left(x^{(2)}\right)^5 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{(n)} & \left(x^{(n)}\right)^2 & \cdots & \left(x^{(n)}\right)^5 \end{bmatrix}}_{\boldsymbol{X}} \underbrace{\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_5 \end{bmatrix}}_{\boldsymbol{b}} + \underbrace{\begin{bmatrix} \varepsilon^{(1)} \\ \varepsilon^{(2)} \\ \vdots \\ \varepsilon^{(n)} \end{bmatrix}}_{\boldsymbol{\varepsilon}} \tag{3}$$

6.2.2 Viewing Polynomial Regression as a Sum of Curves

From a functional perspective, a univariate polynomial regression model can be viewed as a sum of several curves, each corresponding to a different power of x. Figure 4 illustrates this concept: a fifth-degree polynomial function can be seen as the combination of six curves, including the constant term and powers of x from one to five. This visualization helps to understand how polynomial regression builds flexible models by stacking multiple nonlinear features linearly.

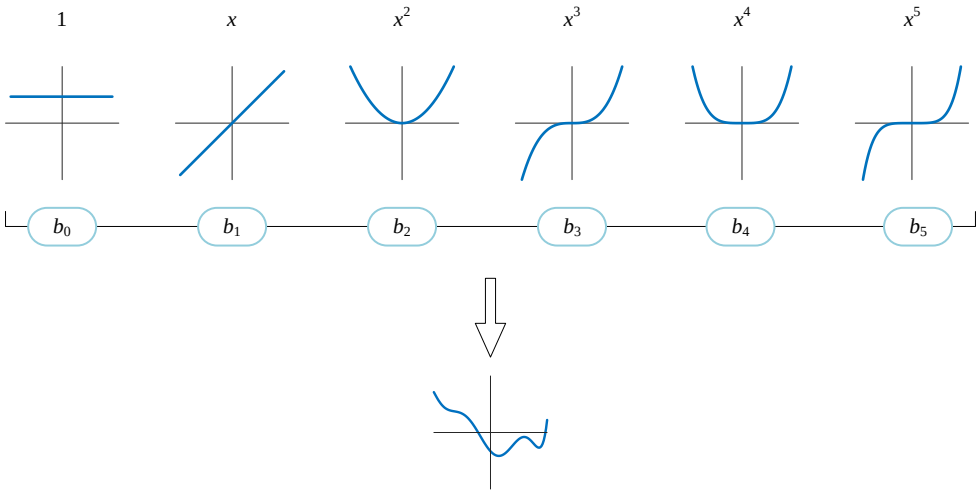

Figure 4. A fifth-degree univariate polynomial as the sum of six component curves

6.3 Solving Polynomial Regression

6.3.1 Revisiting Orthogonal Projection

The method for solving the coefficients b b in univariate polynomial regression is the same as the one used in multivariable linear regression discussed earlier. In polynomial regression, the columns of the design matrix $\boldsymbol{X}$ span a multidimensional vector space, just as in linear regression, and the predicted values $\hat{\boldsymbol{y}}$ are obtained as the orthogonal projection of $\boldsymbol{y}$ onto this space (Figure 5).

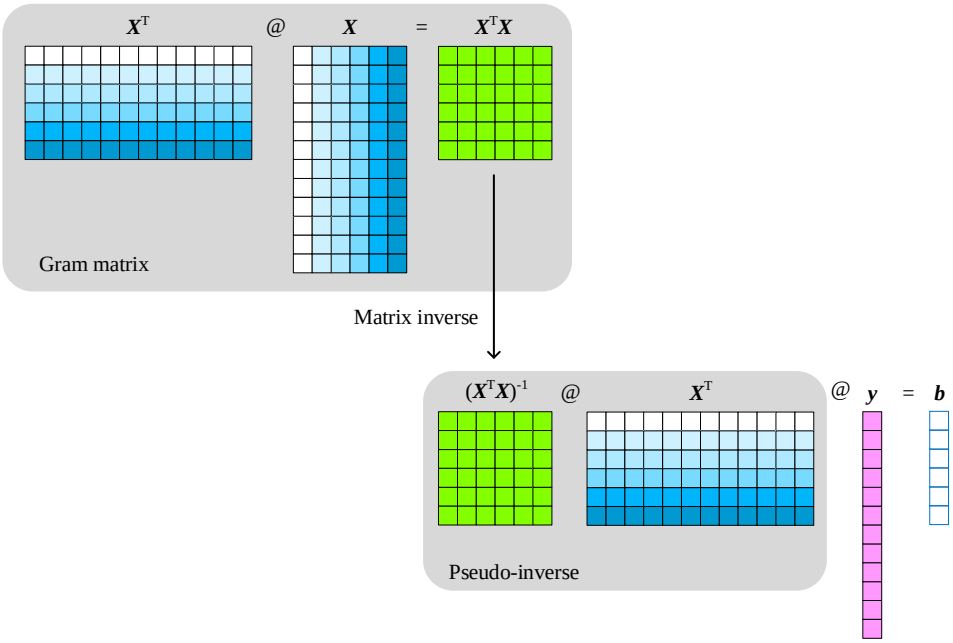

Figure 5. Computing coefficients ***b*** in univariate polynomial regression

6.3.2 Step-by-Step Example: Quadratic Regression (Second-Degree Polynomial)

Consider a simple example with a single independent variable x and dependent variable y, represented as column vectors (Figure 1):

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 1 \\ 7 \\ 8 \\ 6 \\ 5 \\ 2 \\ 2 \\ 4 \end{bmatrix} \tag{4}$$

In regression analysis, it is generally important to have a sufficiently large number of samples to ensure that the estimated model parameters are stable and reliable. However, in this example, we intentionally use a small dataset to make the computation simple and transparent. With fewer samples, readers can easily follow the matrix operations and even verify the results by hand. This helps build an intuitive understanding of how linear regression works before scaling up to larger, real-world datasets.

A univariate quadratic regression, or second-degree polynomial regression, extends the linear model by including a squared term:

$$\hat{y} = b_0 + b_1 x + b_2 x^2 \tag{5}$$

Here, "quadratic" refers to the inclusion of the x^2 term in the model. The highest power of x in the model defines the degree of the polynomial, which in this case is 2. The design matrix for this model is:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{x} & \boldsymbol{x} \odot \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 9 & 81 \end{bmatrix} \tag{6}$$

As shown in Figure 6, the predicted vector $\hat{\boldsymbol{y}}$ is obtained by projecting $\boldsymbol{y}$ orthogonally onto the subspace spanned by $\boldsymbol{1}$, $\boldsymbol{x}$, $\boldsymbol{x} \odot \boldsymbol{x}$.

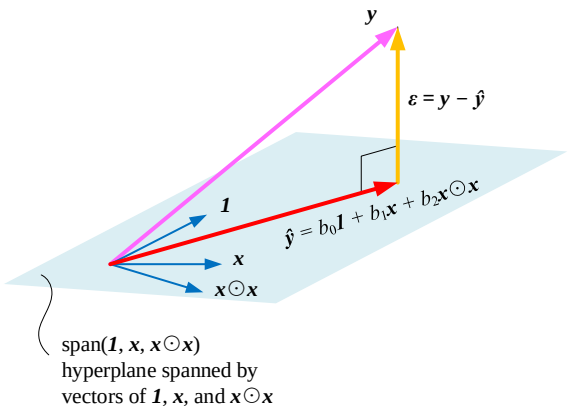

Figure 6. Orthogonal projection of $\boldsymbol{y}$ onto the subspace spanned by $\boldsymbol{1}$, $\boldsymbol{x}$, $\boldsymbol{x} \odot \boldsymbol{x}$ in quadratic regression

The coefficients $\boldsymbol{b}$ are calculated in the same way as in multivariable linear regression:

$$
\begin{aligned}
\boldsymbol{b} &= \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \\
&= \left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 9 & 81 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 9 & 81 \end{bmatrix}\right)^{-1} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 9 & 81 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 1 \\ 7 \\ 8 \\ 6 \\ 5 \\ 2 \\ 2 \\ 4 \end{bmatrix} \\
&= \begin{bmatrix} 8 & 42 & 276 \\ 42 & 276 & 1998 \\ 276 & 1998 & 15252 \end{bmatrix}^{-1} \begin{bmatrix} 35 \\ 173 \\ 1037 \end{bmatrix} \\
&= \begin{bmatrix} 1.636 & -0.670 & 0.058 \\ -0.670 & 0.344 & -0.033 \\ 0.058 & -0.033 & 0.003 \end{bmatrix} \begin{bmatrix} 35 \\ 173 \\ 1037 \end{bmatrix} \\
&= \begin{bmatrix} 1.655 \\ 1.926 \\ -0.214 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}
\end{aligned} \tag{7}
$$

Once $\boldsymbol{b}$ is determined, the quadratic regression model (red curve in Figure 7) is fully specified:

$$\hat{y} = 1.655 + 1.927x - 0.214x^2 \tag{8}$$

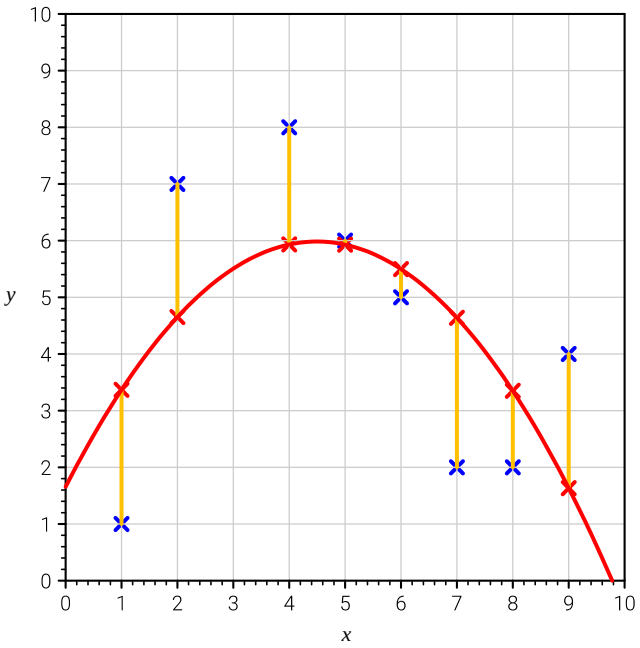

Figure 7. Fitted quadratic regression curve (red) and residuals (orange). Figure generated by Ch06_01_Polynomial_Regression.ipynb.

6.3.3 Understanding the Residual Vector and Total Error

In Figure 7, the red parabola represents this fitted quadratic function, while the orange line segments show the residuals corresponding to the vector

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \begin{bmatrix} -2.368 \\ 2.349 \\ 2.068 \\ 0.070 \\ -0.498 \\ -2.638 \\ -1.35 \\ 2.367 \end{bmatrix} \tag{9}$$

The sum of squared residuals, which measures the total error, is given by:

$$\|\boldsymbol{\varepsilon}\|_2^2 = \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = \begin{bmatrix} -2.368 \\ 2.349 \\ 2.068 \\ 0.070 \\ -0.498 \\ -2.638 \\ -1.35 \\ 2.367 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} -2.368 \\ 2.349 \\ 2.068 \\ 0.070 \\ -0.498 \\ -2.638 \\ -1.35 \\ 2.367 \end{bmatrix} = 30.040 \tag{10}$$

6.3.4 Going Beyond Quadratic: Cubic Regression

Sometimes, a low-degree polynomial is not flexible enough to capture the complexity in the data. For example, in Figure 7, the orange line segments representing residuals are still quite long. To better fit the data, we can increase the polynomial degree to three.

A univariate cubic regression, or third-degree polynomial regression, extends the quadratic model by including a cubic term:

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + b_3 x^3 \tag{11}$$

In this model, the highest power of x is three, allowing the regression curve to bend more and adapt to more complex patterns in the data. The design matrix $\boldsymbol{X}$ for a cubic regression includes columns for the constant term, x, x^2, and x^3. The coefficient vector $\boldsymbol{b}$ is computed using the same method as in linear and quadratic regression.

Figure 8 shows the fitted cubic regression curve in red, corresponding to the equation

$$\hat{y} = -8.539 + 12.211x - 2.643x^2 + 0.16x^3 \tag{12}$$

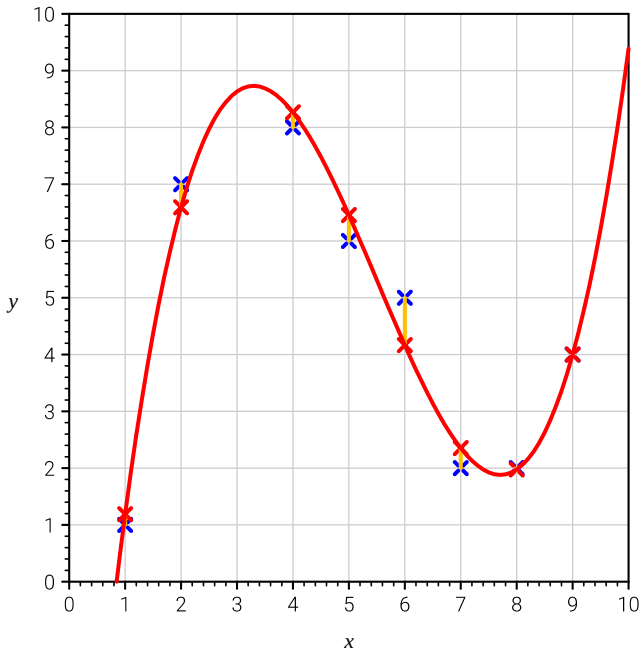

Figure 8. Fitted cubic regression curve (red) and residuals (orange) in univariate cubic polynomial regression. Figure generated by Ch06_01_Polynomial_Regression.ipynb.

The orange line segments represent the residuals, forming the residual vector

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \begin{bmatrix} -0.189 \\ 0.407 \\ -0.266 \\ -0.457 \\ 0.834 \\ -0.352 \\ 0.023 \\ -0.001 \end{bmatrix} \tag{13}$$

The sum of squared residuals, which measures the total error of the model, is calculated as:

$$\left\|\boldsymbol{\varepsilon}\right\|_2^2 = \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = \begin{bmatrix} -0.189 \\ 0.407 \\ -0.266 \\ -0.457 \\ 0.834 \\ -0.352 \\ 0.023 \\ -0.001 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} -0.189 \\ 0.407 \\ -0.266 \\ -0.457 \\ 0.834 \\ -0.352 \\ 0.023 \\ -0.001 \end{bmatrix} = 1.302 \tag{14}$$

Compared with the quadratic regression, the cubic regression clearly reduces the residuals and fits the data more closely, demonstrating how increasing the polynomial degree increases the model's flexibility and its ability to capture nonlinear patterns.

6.3.5 The Double-Edged Sword: Flexibility vs. Overfitting

While increasing the degree of a polynomial can make a model more flexible, very high-degree polynomials may lead to **overfitting**. Overfitting occurs when a model fits the training data extremely well but performs poorly on new, unseen data. In polynomial regression, a model with too high a degree may capture not only the underlying pattern but also the noise and minor fluctuations in the training data. This reduces the model's **generalization capability**, meaning it may fail to make accurate predictions for new observations.

Figure 9 illustrates an eighth-degree univariate polynomial regression. The red curve passes almost perfectly through every data point, resulting in nearly zero training error

$$\hat{y} = 7.123 - 11.430x - 2.828x^2 + 15.342x^3 - 9.463x^4 + 2.599x^5 - 0.370x^6 + 0.027x^7 - 0.001x^8 \quad (15)$$

However, such a model is rarely ideal. As the degree increases, the model becomes more sensitive to the idiosyncrasies of the training set, making it likely to perform poorly on test data or in real-world applications.

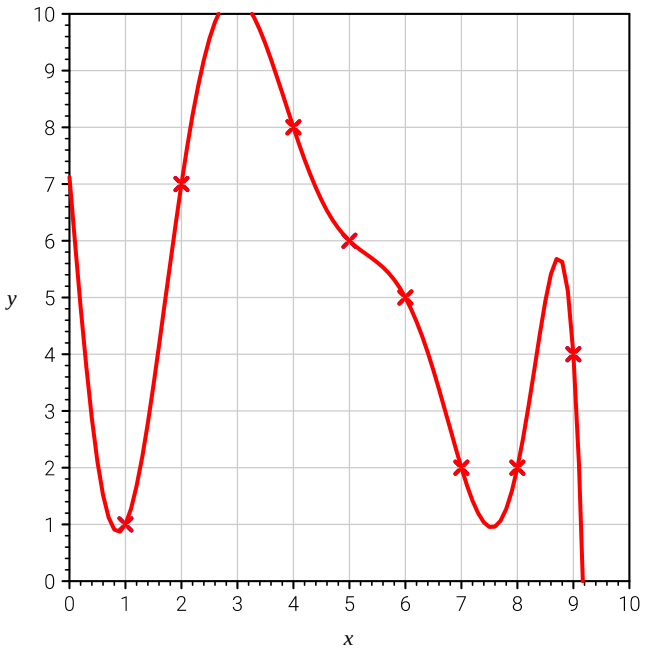

Figure 9. Eighth-degree univariate polynomial regression showing overfitting (red curve) and residuals. Figure generated by Ch06_01_Polynomial_Regression.ipynb.

6.4 Conclusion

This chapter introduces polynomial regression, a method for modeling nonlinear relationships by creating higher-order features from the original variables. Starting with univariate regression, new features such as squares, cubes, and higher powers of the original variable are added to the model, allowing it to capture more complex patterns in the data.

The coefficients are estimated in the same way as in linear regression, using a design matrix and orthogonal projection, and the predicted values are linear combinations of these engineered features. Examples with quadratic and cubic regressions illustrate how increasing the polynomial degree can reduce residual errors and improve the fit.

However, very high-degree polynomials can lead to overfitting, where the model matches the training data too closely, capturing noise and losing the ability to generalize to new data. The chapter also extends the concept

to multivariate polynomial regression, introducing interaction terms and higher-order combinations to model complex relationships between multiple variables effectively.