Contents

1 Examples

1.1 One Predictor Variable - Third Order

The regression model:

$$Y_i = \beta_0 + \beta_1 x_i + \beta_{11} x_i^2 + \beta_{111} x_i^3 + \varepsilon_i$$

where:

$$x_i = X_i - \bar{X}$$

is a third-order model with one predictor variable. The response function for the regression model is:

$$E\{Y\} = \beta_0 + \beta_1 x + \beta_{11} x^2 + \beta_{111} x^3$$

1.2 Two Predictor Variables - Second Order

The regression model:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_{11} x_{i1}^2 + \beta_{22} x_{i2}^2 + \beta_{12} x_{i1} x_{i2} + \varepsilon_i$$

where:

$$x_{i1} = X_{i1} - \bar{X}_1$$

$$x_{i2} = X_{i2} - \bar{X}_2$$

is a second-order model with two predictor variables. The response function is:

$$E\{Y\} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2$$

which is the equation of a conic section. Note that the regression model above contains separate linear and quadratic components for each of the two predictor variables and a cross-product term. The latter represents the interaction effect between x_1 and x_2. The coefficient β_{12} is often called the interaction effect coefficient.

1.3 Three Predictor Variables - Second Order

The second-order regression model with three predictor variables is:

$$\begin{aligned} Y_i =& \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_{11} x_{i1}^2 + \beta_{22} x_{i2}^2 + \beta_{33} x_{i3}^2 \\ &+ \beta_{12} x_{i1} x_{i2} + \beta_{13} x_{i1} x_{i3} + \beta_{23} x_{i2} x_{i3} + \varepsilon_i \end{aligned}$$

where:

$$x_{i1} = X_{i1} - \bar{X}_1$$

$$x_{i2} = X_{i2} - \bar{X}_2$$

$$x_{i3} = X_{i3} - \bar{X}_3$$

The response function for this regression model is:

$$
\begin{aligned}
E\{Y\} =& \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 \\
&+ \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3
\end{aligned}
$$

The coefficients β_{12}, β_{13}, and β_{23} are interaction effect coefficients for interactions between pairs of predictor variables.

2 Model

Polynomial regression models have two basic types of uses:

1. When the true curvilinear response function is indeed a polynomial function.
2. When the true curvilinear response function is unknown (or complex) but a polynomial function is a good approximation to the true function.

The second type of use, where the polynomial function is employed as an approximation when the shape of the true curvilinear response function is unknown, is very common.

A main danger in using polynomial regression models, as we shall see, is that extrapolations may be hazardous with these models, especially those with higher-order terms. Polynomial regression models may provide good fits for the data at hand, but may turn in overfit the training data.

Given an n dimensional column vector $\mathbf{x}$, we could construct the Vandermonde matrix

$$
V = \left[\mathbf{1}, \mathbf{x}, \mathbf{x}^2, \cdots, \mathbf{x}^m\right]
$$

which represents the data by polynomial features up to the m degree. In sklearn, the PolynomialFeatures(degree) transformer generates all monomials up to m “degree”. This gives us exactly the Vandermonde matrix with “nsamples” rows and “m degree + 1” columns.

Note: Because monomials of higher orders can lead to large values for continuous columns. It may be necessary to use feature scaling to bring the transformed data back to an acceptable range of values.

2.1 Orthogonal Polynomials

A potential drawback of polynomial regression is that serious multicollinearity may be present even when the predictor variables are centered. One approach to address this issue is to use an orthogonal polynomial basis. That is, instead of the standard monomial basis

$$
\left[\mathbf{1}, \mathbf{x}, \mathbf{x}^2, \cdots, \mathbf{x}^m\right],
$$

we use some other basis

$$[\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \cdots, \mathbf{b}_4],$$

where the vectors ($\mathbf{b}_i$) span the same subspace as the "monomial" vectors, but also form an orthogonal basis for that subspace, meaning that for all $i \neq j$ we have $\mathbf{b}_i^T \mathbf{b}_j = 0$. The change of basis is just a reparameterization, so it doesn't affect the final regression function, but now we have a set of uncorrelated predictors, and $V^T V$ is trivial to invert since it's just a diagonal matrix. R's built-in poly() function uses a QR decomposition on the Vandermonde matrix

$$V = QR$$

where Q is an $n \times n$ orthogonal matrix whose columns form an orthogonal basis for the same m-dimensional subspace spanned by the columns of V. The matrix R, which tells us how to "reconstruct" V from Q, is upper triangular, which guarantees that the first k columns of V can be represented as a linear combination of the first k columns of Q. Remember that the first k columns of V are just the monomials up to order $k-1$, so this means that the first column of Q must be a constant, the second linear (w.r.t. $\mathbf{x}$), and so on. This means that we can take the columns of Q to be our orthogonal basis, without losing any of the intuitive "polynomial-ness" of the original basis.