EE16B - Fall'16 - Lecture 10B Notes[1]

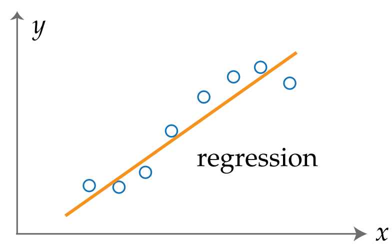
Murat Arcak

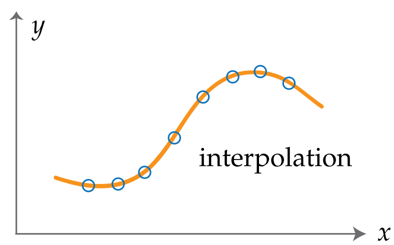
3 November 2016

Sampling and Interpolation

Regression vs. Interpolation

Given data points (x_i, y_i), $i = 1, \cdots, n$, *interpolation* is the task of finding a function that exactly matches the data points as shown in the figure below (right). This differs from *regression* whose aim is to choose an approximate fit to data from among a class of functions as in the figure (left). The *least squares* method you studied in 16A is a commonly used form of regression.





Regression is meaningful when the data are inaccurate; for example when we have noisy measurements that cluster around a line rather than lying exactly on a line.

Interpolation is preferable when the data are accurate and we believe the variation is the result of the core phenomenon that the data represents rather than noise. For example, when zooming in on a digital image, an algorithm interpolates between existing pixels to fill in the pixels between them.

Interpolation is performed using a family of functions, such as polynomials, to predict what happens between the data points.

Polynomial Interpolation

If we have two data points (x_1, y_1), (x_2, y_2) we can find a line $y = a_0 + a_1 x$ that exactly matches these points, that is

$$a_0 + a_1 x_1 = y_1, \quad a_0 + a_1 x_2 = y_2.$$

To find a_0 and a_1, rewrite the two equations above in matrix form as

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{1}$$

and note that the matrix

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix}$$

is invertible when $x_1 \neq x_2$.

For three data points (x_1, y_1), (x_2, y_2), (x_3, y_3) we can find a quadratic polynomial $y = a_0 + a_1 x + a_2 x^2$ by solving for a_0, a_1, a_2 from

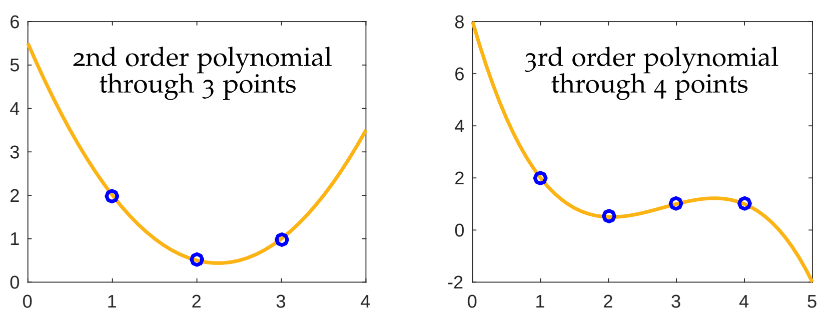
$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}. \tag{2}$$

Similarly, for four data points (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), we look for a cubic polynomial $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ and solve for a_0, a_1, a_2, a_3 from

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}. \tag{3}$$

We will see below that the matrices in the two equations above are invertible if $x_i \neq x_j$ when $i \neq j$, that is if the x_i values are distinct.

The results of this procedure are illustrated in the figure below for three data points (left) and four data points (right).



Generalizing the arguments above to n data points we reach the following conclusion:

> Given (x_i, y_i), $i = 1, \cdots, n$, where $x_i \neq x_j$ when $i \neq j$, there is a unique polynomial of order $(n-1)$ passing through these points:
> $$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}.$$

We solve for the n coefficients $a_0, a_1, \ldots, a_{n-1}$ from

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}}_{V} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{bmatrix}. \tag{4}$$

The matrix V is known as a Vandermonde matrix in linear algebra and its determinant is given by

$$\det(V) = \prod_{1 \le i < j \le n} (x_j - x_i), \tag{5}$$

which is $(x_2 - x_1)$ for $n = 2$, $(x_2 - x_1)(x_3 - x_1)(x_3 - x_2)$ for $n = 3$, and so on[2]. Since we assumed $x_i \neq x_j$ when $i \neq j$, we conclude that the determinant (5) is nonzero and, thus, V is invertible.

[2] Verify this formula for $n = 2$ and $n = 3$ by calculating the determinant of the matrices in (1) and (2).

Example: To find a cubic polynomial $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ passing through the four points $(-1, 1), (0, 0), (1, 1), (2, 4)$, set up the equation (3):

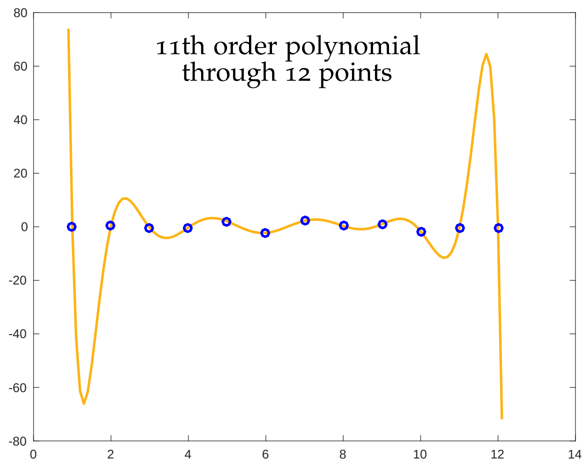
$$\underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \end{bmatrix}}_{V} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 4 \end{bmatrix}.$$

By inverting the matrix V we obtain

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \underbrace{\frac{1}{12} \begin{bmatrix} 0 & 12 & 0 & 0 \\ -4 & -6 & 12 & -2 \\ 6 & -12 & 6 & 0 \\ -2 & 6 & -6 & 2 \end{bmatrix}}_{V^{-1}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Thus, the polynomial is $y = x^2$. In general we need a third order polynomial to interpolate between four points but in this example the points lie on a quadratic curve, thus the leading term $a_3 x^3$ is zero. If we change y_4 to $4 + \varepsilon$, we get $a_3 = \varepsilon/6$, $a_2 = 1$, $a_1 = -\varepsilon/6$, $a_0 = 0$, thus the new polynomial is third order: $y = -\frac{\varepsilon}{6} x + x^2 + \frac{\varepsilon}{6} x^3$. □

A shortcoming of polynomial approximation is that, as n gets larger, the polynomial has terms with large powers (up to $n - 1$) that grow very fast. This makes the interpolation erratic and unreliable, especially near the end points. The figure below illustrates this behavior with a polynomial interpolation through $n = 12$ points.



Interpolation with Basis Functions

An alternative method is to assign to each x_i a function $\phi_i(x)$ that satisfies:

$$\phi_i(x_i) = 1 \quad \text{and} \quad \phi_i(x_j) = 0 \text{ when } j \neq i \tag{6}$$

and to interpolate between the data points (x_i, y_i) with the function:

$$y = \sum_k y_k \phi_k(x). \tag{7}$$

When $x = x_i$ this expression yields $y = y_i$ as desired, because $\phi_k(x_i) = 0$ except when $k = i$.

We refer to $\phi_i(x)$ as "basis functions" since our interpolation consists of a linear combination of these functions. Basis functions restrict the behavior of the interpolation between data points and avoid the erratic results of polynomial approximation. The figure below uses triangular basis functions which lead to a linear interpolation.