Polynomial Interpolation

Suppose we are given a table of $(n+1)$ points.

x	x_0	x_1	$\dots$	x_n
y	y_0	y_1	$\dots$	y_n

Then polynomial interpolation consists of finding a polynomial that passes through the given points. The $x_i's$ are called the nodes.

Polynomials of degree n Consider the form of an nth degree polynomial:

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_2x^2 + a_1x + a_0$$

That nth degree polynomial has $n+1$ coefficients, because of the a_0. That means we have $n+1$ free parameters that need to be solved for. We would like a unique solution to this problem, so we'd need $n+1$ independent equations. Since each data pair (x, y) can be used to generate an equation, that means

> We need $n+1$ data points to construct a unique nth degree polynomial with $P(x_i) = f(x_i)$ for $x = 0$ to n.

Methods for determining the interpolating polynomial We will only learn one method to interpolate. That method is know as **Newton's form** of interpolating polynomials. There are other methods discussed in the book like **Lagrange form**, and a technique called **Newton's divided difference method**, which is another algorithm for producing Newton's form.

Newton's Form

Newton's form of the interpolating polynomial is the following:

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + ... + c_N(x - x_0)(x - x_1)...(x - x_{N-1})$$

where the x_i are the given $N+1$ data points $x_0...x_N$, and the c_i are coefficients we must solve for. The idea behind solving for the coefficients is this (and it's an important use for this form): not only does the Newton's form polynomial interpolate the data points, but the first term is the 0 degree polynomial interpolating the first point, the first and second terms are the 1 degree polynomial interpolating the first two points, and so on:

$$\begin{aligned}
p_0(x) &= c_0 \\
p_1(x) &= c_0 + c_1(x - x_0) \\
p_2(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) \\
&\vdots \\
p_N(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + ... + c_N(x - x_0)(x - x_1)...(x - x_{N-1})
\end{aligned}$$

At each data point, we wish for the value of the polynomial to equal the value of the function. So, at each x_i, $P(x_i) = f(x_i)$. The technique for solving for the coefficients is basically nested multiplication; we use x_0 to solve p_0 for c_0, then use c_0 and x_1 to solve P_1 for c_1, and so on:

$$\begin{aligned}
f(x_0) &= c_0 \\
f(x_1) &= c_0 + c_1(x_1 - x_0) \\
f(x_2) &= c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) \\
&\vdots \\
f(x_N) &= c_0 + c_1(x_N - x_0) + c_2(x_N - x_0)(x_N - x_1) + ... + c_N(x_N - x_0)(x_N - x_1)...(x_N - x_{N-1})
\end{aligned}$$

Example:

x	5	-7	-6	0
y	1	-23	-54	-954

$$\begin{aligned}
1 &= c_0 \\
-23 &= c_0 + c_1((-7) - (5)) \\
-54 &= c_0 + c_1((-6) - (5)) + c_2((-6) - (5))((-6) - (-7)) \\
-954 &= c_0 + c_1((0) - (5)) + c_2((0) - (5))((0) - (-7)) + ... \\
&= c_3((0) - (5))((0) - (-7))((0) - (-6))
\end{aligned}$$

$$c_0 = 1$$

$$\begin{aligned}
1 + (-12)c_1 &= -23 \\
c_1 &= 2
\end{aligned}$$

$$\begin{aligned}
1 + 2(-11) + c_2(-11)(1) &= -54 \\
c_2 &= 3
\end{aligned}$$

$$\begin{aligned}
1 + 2(-5) + 3(-5)(7) + c_3(-5)(7)(6) &= -954 \\
c_3 &= 4
\end{aligned}$$

$$P(x) = 1 + 2(x-5) + 3(x-5)(x+7) + 4(x-5)(x+7)(x+6)$$

Advantages to Newton's form

- It shows the polynomial written as a function of the data points; you can see exactly what x-values were sampled to build the thing
- Data does not need to be in order.
- This form is the best form for adding on additional (x, y) pairs. Each of the polynomials gets built up from the last one. If we wish to add in extra data points, we do not need to recalculate all the previous coefficients.

Example: Suppose our table now includes

x	5	-7	-6	0	-1
y	1	-23	-54	-954	-10

We now can form a fourth degree polynomial out of the 5 nodes. To do so, we need only to calculate the coeffcient c_4:

$$\begin{aligned}
f(x_4) &= c_0 + c_1(x_4 - x_0) + c_2(x_4 - x_0)(x_4 - x_1) + ... \\
&= c_3(x_4 - x_0)(x_4 - x_1)(x_4 - x_2) + c_4(x_4 - x_0)(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)
\end{aligned}$$

$$\begin{aligned}
-10 &= 1 + 2((-1) - (5)) + 3((-1) - (5))((-1) - (-7)) + ... \\
&= 4((-1) - (5))((-1) - (7))((-1) - (-6)) + c_4((-1) - (5))((-1) - (7))((-1) - (-6))((-1) - (0))
\end{aligned}$$

$$\begin{aligned}
&= 1 + 2(-6) + 3(-6)(6) + 4(-6)(6)(5) + c_4(-6)(6)(5)(-1) \\
&= 1 - 12 - 108 - 720 + 180c_4
\end{aligned}$$

$$c_4 = 4.6056$$

$P(x)$ now equals

$$P(x) = 1 + 2(x - 5) + 3(x - 5)(x + 7) + 4(x - 5)(x + 7)(x + 6) + 4.6056(x - 5)(x + 7)(x + 6)(x + 1)$$

This form will be the preferred form for interpolation. The only down side is it's a pain to do the calcuations by hand.