# 1 Polynomial Interpolation (Section 6.1)

Suppose we're given $m+1$ data points,

$$(x_i, y_i), \quad 0 \le i \le m,$$

and we want to seek a polynomial $P$ of the *lowest possible degree* for which

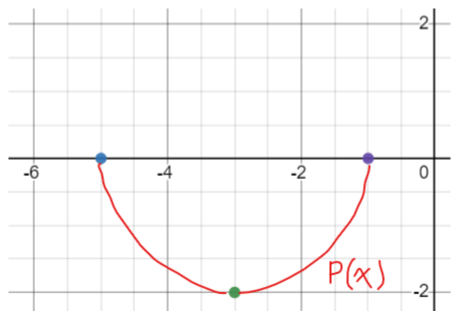
$$P(x_i) = y_i, \quad 0 \le i \le m.$$

Such a polynomial $P$ is said to **interpolate** the data.

(Example.) Suppose we have the following data points.

| $x$ | -5 | -3 | -1 |
|---|---|---|---|
| $y$ | 0 | -2 | 0 |

Drawing the points and the associated curve,




schematically, we can represent the data points above with the polynomial

$$P(x) = \frac{1}{2}(x+3)^2 - 2.$$

We don't care about what the curve looks like beyond those points. The interpolation condition just ensures that the curve associated with the polynomial goes through the *given* points.

**Remarks:**

- If we were just given a single point, then the lowest-degree polynomial we can create is a constant polynomial, $P(x) = y_0$.
- If we had two points, then we can draw a line through the points and the lowest-degree polynomial we can create is a linear equation.

The theorem that governs this problem is shown below.

**Theorem 1.1: Polynomial Interpolation**

If $x_0, x_1, \ldots, x_m$ are distinct real numbers, then for arbitrary values $y_0, y_1, \ldots, y_m$, there exists a unique polynomial $P_m$ of degree at most $m$ such that

$$P_m(x_i) = y_i \quad (0 \le i \le m)$$

*Proof.* We'll show both aspects of the theorem.

- Uniqueness: Suppose we have two interpolating polynomials $P_m(x_i) = y_i$ and $Q_m(x_i) = y_i$, both of which are degree $m$. Then, their difference, $P_m(x) - Q_m(x)$, also has at most degree $m$. This means that this difference polynomial has at most $m$ zeros/roots. But, as both $P_m$ and $Q_m$ are interpolating polynomials, $P_m(x_i) - Q_m(x_i) = y_i - y_i = 0$ has $m+1$ zeros. Thus, $P_m(x) - Q_m(x) = 0$ has zeros everywhere and thus $P_m(x) = Q_m(x)$.

- Existence: Suppose, for $k \geq 1$, $P_{k-1}(x)$ has degree $k-1$. In other words, $P_{k-1}(x_i) = y_i$ for $0 \leq i \leq k-1$. Suppose we want to construct the next higher-degree polynomial, degree $k$, such that $P_k(x_i) = y_i$ for $0 \leq i \leq k$. Then,
$$P_k(x) = P_{k-1}(x) + c(x - x_0)(x - x_1) \dots (x - x_{k-1}).$$
Then,
$$P_k(x_i) = P_{k-1}(x_i) + c \cdot 0 = P_{k-1}(x_i) = y_i, \quad (0 \leq i \leq k-1).$$
So, set $P_k(x_k) = y_k$ and then solve for $c$. More specifically,
$$P_k(x_k) = y_k = P_{k-1}(x_k) + c(x_k - x_0)(x_k - x_1)(x_k - x_2) \dots (x_k - x_{k-1}).$$
By solving for $c$, we have
$$c = \frac{y_k - P_{k-1}(x_k)}{(x_k - x_0)(x_k - x_1)(x_k - x_2) \dots (x_k - x_{k-1})}.$$

This concludes the proof. □

## 1.1 Polynomial Representation

There are different ways we can represent these polynomials, although keep in mind that they all represent the same function.

### 1.1.1 Newton's Form

Newton's Form is

$$P_m(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_m(x - x_0)(x - x_1) \dots (x - x_{m-1}) \tag{1}$$

Note that this form models $m + 1$ data points ($0 \leq i \leq m$). Notice, however, that we never include $x_m$ in our final equation. The first few cases of the above equation are

$$\begin{aligned} P_0(x) &= c_0 \\ P_1(x) &= c_0 + c_1(x - x_0) \\ P_2(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) \end{aligned}$$

(Example.) Suppose we have the polynomial $P_m(t)$, for $t \in \mathbb{R}$ (i.e., we're evaluating this polynomial with value $t$). We can write an algorithm similar to Horner's method for evaluating this polynomial. Then, we'll have the inputs $x_i$, $c_i$ for $0 \leq i \leq m$, and $t \in \mathbb{R}$.

**Algorithm 1** Finding the Polynomial

```
p ← c_m
for k ← m − 1 to 0 step −1 do
    p ← (t − x_k) · p + c_k
end for
```

To find the coefficients $c_k$, we have

$$c_k = \begin{cases} y_0 & k = 0 \\ \frac{y_k - P_{k-1}(x_k)}{(x_k - x_0)(x_k - x_1)(x_k - x_2) \dots (x_k - x_{k-1})} & k \geq 1 \end{cases}. \tag{2}$$

To compute the coefficients, we can make use of the following algorithm. Given

- $x_i \quad (0 \leq i \leq m)$

- $y_i \quad (0 \le i \le m)$

this algorithm should output $c_i$ for $0 \le i \le m$.

**Algorithm 2** Computing $c_i$

$c_0 \leftarrow y_0$<br>
**for** $k \leftarrow 1$ to $m$ **do**<br>
    $d \leftarrow (x_k - x_{k-1})$<br>
    $p \leftarrow c_{k-1}$<br>
    **for** $i \leftarrow k-2$ to $0$ step $-1$ **do**<br>
        $d \leftarrow d(x_k - x_i)$ ▷ Denominator<br>
        $p \leftarrow p(x_k - x_i) + c_i$ ▷ $P_{k-1}(x_k)$<br>
    **end for**<br>
    $c_k \leftarrow (y_k - p)/d$<br>
**end for**

(Example.) Suppose we have the data points

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

Newton's form for the polynomial looks like

$$P(x) = c_0 + c_1(x-5) + c_2(x-5)(x+7) + c_3(x-5)(x+7)(x+6).$$

Then, we can compute each of the $c_i$ for $0 \le i \le m = 3$.

- $i = 0$: we know that
$$c_0 = y_0 = 1.$$
- $i = 1$: we have
$$c_1 = \frac{y_1 - P_0(x_1)}{(x_1 - x_0)} = \frac{-23 - 1}{(-7-5)} = 2.$$
- $i = 2$: we have
$$c_2 = \frac{y_2 - P_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} = \frac{-54 - (c_0 + c_1(x_2 - x_0))}{(-6-5)(-6-(-7))} = \frac{-54 - (1 + 2(-6-5))}{(-6-5)(-6-(-7))} = 3.$$
- $i = 3$: by the same process as above, we find that $c_3 = 4$.