CS412: Lecture #10

Mridul Aanjaneya

February 19, 2015

Newton Interpolation

The Newton basis functions can be derived by considering the problem of building a polynomial interpolant *incrementally* as successive new data points are added. Here is the basic idea:

- **Step 0:** Define a degree-0 polynomial $\mathcal{P}_0(x)$ that just interpolates (x_0, y_0). Obviously, we can achieve that by simply selecting

$$\mathcal{P}_0(x) = y_0$$

- **Step 1:** Define a degree-1 polynomial $\mathcal{P}_1(x)$ that now interpolates both (x_0, y_0) and (x_1, y_1). We also want to take advantage of the previously defined $\mathcal{P}_0(x)$ by constructing $\mathcal{P}_1$ as

$$\mathcal{P}_1(x) = \mathcal{P}_0(x) + \mathcal{M}_1(x)$$

where $\mathcal{M}_1(x)$ is a degree-1 polynomial and it needs to satisfy

$$\underbrace{\mathcal{P}_1(x_0)}_{=y_0} = \underbrace{\mathcal{P}_0(x_0)}_{=y_0} + \mathcal{M}_1(x_0) \Rightarrow \mathcal{M}_1(x_0) = 0$$

Thus, $\mathcal{M}_1(x) = c_1(x - x_0)$. We can determine c_1 using:

$$\mathcal{P}_1(x_1) = \mathcal{P}_0(x_1) + c_1(x_1 - x_0) \Rightarrow c_1 = \frac{\mathcal{P}_1(x_1) - \mathcal{P}_0(x_1)}{x_1 - x_0} = \frac{y_1 - \mathcal{P}_0(x_1)}{x_1 - x_0}$$

- **Step 2:** Now construct $\mathcal{P}_2(x)$ which interpolates the three points (x_0, y_0), $(x_1, y_1), (x_2, y_2)$. Define it as:

$$\mathcal{P}_2(x) = \mathcal{P}_1(x) + \mathcal{M}_2(x)$$

where $\mathcal{M}_2(x)$ is a degree-2 polynomial. Once again we observe that

$$\left.\begin{array}{rcl} \underbrace{\mathcal{P}_2(x_0)}_{=y_0} &=& \underbrace{\mathcal{P}_1(x_0)}_{=y_0} + \mathcal{M}_2(x_0) \\ \underbrace{\mathcal{P}_2(x_1)}_{=y_1} &=& \underbrace{\mathcal{P}_1(x_1)}_{=y_1} + \mathcal{M}_2(x_1) \end{array}\right\} \Rightarrow \mathcal{M}_2(x_0) = \mathcal{M}_2(x_1) = 0$$

Thus, $\mathcal{M}_2(x)$ must have the form:

$$\mathcal{M}_2(x) = c_2(x - x_0)(x - x_1)$$

Substituting $x \leftarrow x_2$, we get an expression for c_2

$$\begin{aligned} y_2 = \mathcal{P}_2(x_2) &= \mathcal{P}_1(x_2) + c_2(x_2 - x_0)(x_2 - x_1) \\ \Rightarrow &\quad c_2 = \frac{y_2 - \mathcal{P}_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

- **Step k:** In the previous step, we constructed a degree-$(k-1)$ polynomial that interpolates $(x_0, y_0), \ldots, (x_{k-1}, y_{k-1})$. We will use this $\mathcal{P}_{k-1}(x)$ and now define a degree-k polynomial $\mathcal{P}_k(x)$ such that all of $(x_0, y_0), \ldots, (x_k, y_k)$ are interpolated. Again,

  $$\mathcal{P}_k(x) = \mathcal{P}_{k-1}(x) + \mathcal{M}_k(x)$$

  where $\mathcal{M}_k(x)$ is a degree-k polynomial.

  Now we have for any $i \in \{0, 1, \ldots, k-1\}$

  $$\underbrace{\mathcal{P}_k(x_i)}_{=y_i} = \underbrace{\mathcal{P}_{k-1}(x_i)}_{=y_i} + \mathcal{M}_k(x_i) \Rightarrow \mathcal{M}_k(x_i) = 0$$

  Thus, the degree-k polynomial $\mathcal{M}_k$ must have the form

  $$\mathcal{M}_k(x) = c_k(x - x_0) \ldots (x - x_{k-1})$$

  Substituting $x \leftarrow x_k$ gives

  $$\begin{aligned} y_k &= \mathcal{P}_k(x_k) = \mathcal{P}_{k-1}(x_k) + c_k(x_k - x_0) \ldots (x_k - x_{k-1}) \\ \Rightarrow c_k &= \frac{y_k - \mathcal{P}_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} \end{aligned}$$

Every polynomial $\mathcal{M}_i(x)$ in this process is written as

$$\mathcal{M}_i(x) = c_i \mathcal{N}_i(x) \quad \text{where} \quad \mathcal{N}_i(x) = \prod_{j=0}^{i-1}(x - x_j)$$

After n steps, the interpolating polynomial $\mathcal{P}_n(x)$ is then written as:

$$\mathcal{P}_n(x) = c_0\mathcal{N}_0(x) + c_1\mathcal{N}_1(x) + \ldots + c_n\mathcal{N}_n(x)$$

where

$$\begin{aligned} \mathcal{N}_0(x) &= 1 \\ \mathcal{N}_1(x) &= x - x_0 \\ \mathcal{N}_2(x) &= (x - x_0)(x - x_1) \\ &\vdots \\ \mathcal{N}_k(x) &= (x - x_0)(x - x_1) \ldots (x - x_{k-1}) \end{aligned}$$

These are the *Newton polynomials* (compare with the Lagrange polynomials $l_i(x)$). Note that the x_i's are called the *centers*.

We illustrate the incremental Newton interpolation by building the Newton interpolant incrementally as the new data points are added. We begin with the first data point $(x_0, y_0) = (-2, -27)$, which is interpolated by the constant polynomial

$$\mathcal{P}_0(x) = y_0 = -27$$

Incorporating the second data point $(x_1, y_1) = (0, -1)$, we modify the previous polynomial so that it interpolates the new data point as well:

$$\begin{aligned}
\mathcal{P}_1(x) &= \mathcal{P}_0(x) + \mathcal{M}_1(x) = \mathcal{P}_0(x) + c_1(x - x_0) \\
&= \mathcal{P}_0(x) + \frac{y_1 - \mathcal{P}_0(x)}{x_1 - x_0}(x - x_0) \\
&= -27 + \frac{-1 - (-27)}{0 - (-2)}(x - (-2)) \\
&= -27 + 13(x + 2)
\end{aligned}$$

Finally, we incorporate the third data point $(x_2, y_2) = (1, 0)$, modifying the previous polynomial so that it interpolates the new data point as well:

$$\begin{aligned}
\mathcal{P}_2(x) &= \mathcal{P}_1(x) + \mathcal{M}_2(x) = \mathcal{P}_1(x) + c_2(x - x_0)(x - x_1) \\
&= \mathcal{P}_1(x) + \frac{y_2 - \mathcal{P}_1(x_2)}{(x_2 - x_0)(x_2 - x_1)}(x - x_0)(x - x_1) \\
&= -27 + 13(x + 2) + \frac{0 - 12}{(1 - (-2))(1 - 0)}(x - (-2))(x - 0) \\
&= -27 + 13(x + 2) - 4(x + 2)x
\end{aligned}$$

So far, we saw two ways of computing the Newton interpolant, triangular matrix and incremental interpolation. There is, however, another efficient and systematic way to compute them, called *divided differences*. A divided difference is a function defined over a set of sequentially indexed centers, e.g.,

$$x_i, x_{i+1}, \ldots, x_{i+j-1}, x_{i+j}$$

The divided difference of these values is denoted by:

$$f[x_i, x_{i+1}, \ldots, x_{i+j-1}, x_{i+j}]$$

The value of this symbol is defined recursively as follows. For divided differences with one argument,

$$f[x_i] \equiv f(x_i) = y_i$$

With two arguments:

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

With three:

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

With $j+1$ arguments:

$$f[x_i, x_{i+1}, \ldots, x_{i+j-1}, x_{i+j}] = \frac{f[x_{i+1}, \ldots, x_{i+j}] - f[x_i, \ldots, x_{i+j-1}]}{x_{i+j} - x_i}$$

The fact that makes divided differences so useful is that $f[x_i, \ldots, x_{i+j}]$ can be shown to be the coefficient of the *highest power of* x in a polynomial that interpolates through

$$(x_i, y_i), (x_{i+1}, y_{i+1}), \ldots, (x_{i+j-1}, y_{i+j-1}), (x_{i+j}, y_{i+j})$$

Why is this so useful?

Remember, the polynomial that interpolates

$$(x_0, y_0), \ldots, (x_k, y_k)$$

is

$$\mathcal{P}_k(x) = \underbrace{\mathcal{P}_{k-1}(x)}_{\text{highest power}=x^{k-1}} + \underbrace{c_k(x - x_0)\ldots(x - x_{k-1})}_{=c_k x^k + \text{lower powers}}$$

Thus, $c_k = f[x_0, x_1, x_2, \ldots, x_k]$! Or, in other words,

$$\begin{aligned}
\mathcal{P}_n(x) &= f[x_0] \\
&+ f[x_0, x_1](x - x_0) \\
&+ f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
&\vdots \\
&+ f[x_0, x_1, \ldots, x_n](x - x_0)\ldots(x - x_{n-1})
\end{aligned}$$

So, if we can quickly evaluate the divided differences, we have determined $\mathcal{P}_n(x)$! Let us see a specific example:

$$
\begin{aligned}
(x_0, y_0) &= (-2, -27) \\
(x_1, y_1) &= (0, -1) \\
(x_2, y_2) &= (1, 0) \\
f[x_0] &= y_0 = -27 \\
f[x_1] &= y_1 = -1 \\
f[x_2] &= y_2 = -0 \\
f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} &= \frac{-1 - (-27)}{0 - (-2)} = 13 \\
f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} &= \frac{0 - (-1)}{1 - 0} = 1 \\
f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} &= \frac{1 - 13}{1 - (-2)} = -4
\end{aligned}
$$

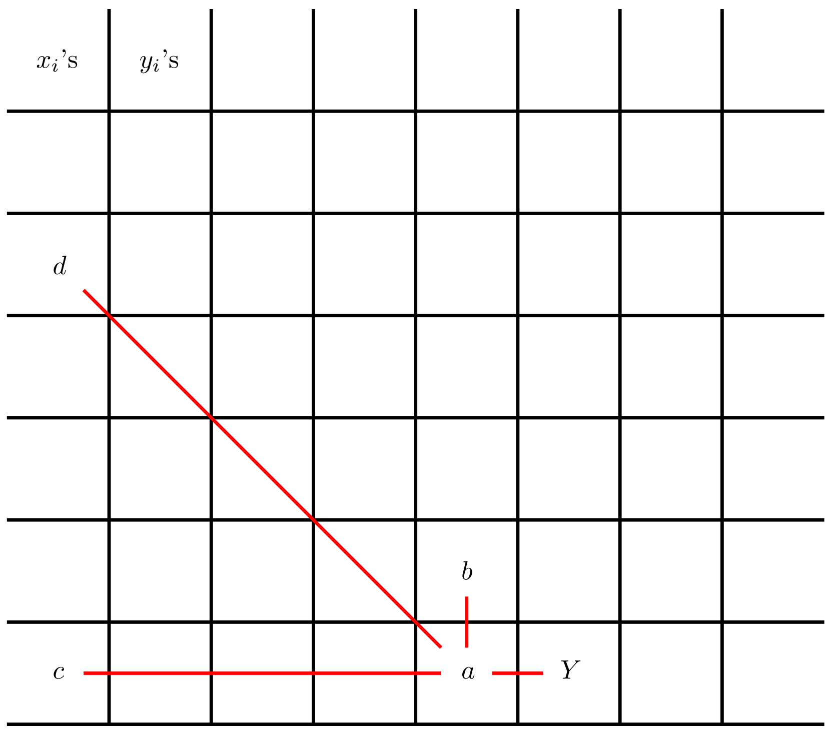
Thus,

$$
\begin{aligned}
\mathcal{P}_2(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
&= -27 + 13(x + 2) - 4(x + 2)x
\end{aligned}
$$

Divided differences are usually tabulated as follows:

	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$\ldots$
x_0	$f[x_0]$			
x_1	$f[x_1]$	$f[x_0,x_1]$		
x_2	$f[x_2]$	$f[x_1,x_2]$	$f[x_0,x_1,x_2]$	
x_3	$f[x_3]$	$f[x_2,x_3]$	$f[x_1,x_2,x_3]$	$\ldots$
x_4	$f[x_4]$	$f[x_3,x_4]$	$f[x_2,x_3,x_4]$	$\ldots$

The recursive definition can be implemented directly on the table as follows:



$$Y = \frac{a-b}{c-d}$$

For example, for the sample set $(x_0, y_0) = (-2, -27)$, $(x_1, y_1) = (0, -1)$, $(x_2, y_2) = (1, 0)$,

x_i's	y_i's			
-2	-27			
0	-1	13		
1	0	1	-4	

Easy evaluation

$$\begin{aligned}
\mathcal{P}_4(x) &= c_0 \\
&+ c_1(x - x_0) \\
&+ c_2(x - x_0)(x - x_1) \\
&+ c_3(x - x_0)(x - x_1)(x - x_2) \\
&+ c_4(x - x_0)(x - x_1)(x - x_2)(x - x_3)
\end{aligned}$$

$$= \underbrace{c_0 + (x - x_0)\underbrace{[c_1 + (x - x_1)\underbrace{[c_2 + (x - x_2)\underbrace{[c_3 + (x - x_3)\underbrace{c_4}_{\mathcal{Q}_4(x)}]}_{\mathcal{Q}_3(x)}]}_{\mathcal{Q}_2(x)}]}_{\mathcal{Q}_1(x)}}_{\mathcal{Q}_0(x)}$$

$$\mathcal{P}_4(x) = \mathcal{Q}_0(x)$$

Recursively: Define $\mathcal{Q}_n(x) = c_n$. Then

$$\mathcal{Q}_{n-1}(x) = c_{n-1} + (x - x_{n-1})\mathcal{Q}_n(x)$$

The value of $\mathcal{P}_n(x) = \mathcal{Q}_0(x)$ can be evaluated (in linear time) by iterating this recurrence n times. We also have

$$\begin{aligned}
\mathcal{Q}_{n-1}(x) &= c_{n-1} + (x - x_{n-1})\mathcal{Q}_n(x) \\
\Rightarrow \mathcal{Q}'_{n-1}(x) &= \mathcal{Q}_n(x) + (x - x_{n-1})\mathcal{Q}'_n(x)
\end{aligned}$$

Thus, once we have computed all the $\mathcal{Q}'_k$s, we can also compute all the derivatives too! Ultimately, $\mathcal{P}'_n(x) = \mathcal{Q}'_0(x)$.

Let us evaluate Newton's method, as we did with other methods:

- Cost of computing $\mathcal{P}_n(x)$: $O(n^2)$.
- Cost of evaluating $\mathcal{P}_n(x)$ for an arbitrary x: $O(n)$.

  This can be accelerated (similar to Horner's method) using the recursive scheme defined above.
- Availability of derivatives: *yes*, as discussed above.
- Allows for incremental interpolation: *yes*!