CS412: Lecture #9

Mridul Aanjaneya

February 17, 2015

Lagrange Interpolation

Lagrange interpolation is an alternative way to define $\mathcal{P}_n(x)$, without having to solve expensive systems of equations. For a given set of $n+1$ points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, define the Lagrange polynomials of degree-n $l_0(x)$, $l_1(x), \ldots, l_n(x)$ as:

$$l_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Then, the interpolating polynomial is simply

$$\mathcal{P}_n(x) = y_0 l_0(x) + y_1 l_1(x) + \ldots + y_n l_n(x) = \sum_{i=0}^{n} y_i l_i(x)$$

Note that no solution of a linear system is required here. We just have to explain what every $l_i(x)$ looks like. Since $l_i(x)$ is a degree-n polynomial, with the n-roots $x_0, x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_{i+2}, \ldots, x_n$, it must have the form

$$\begin{aligned} l_i(x) &= C_i(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n) \\ &= C_i \prod_{j \neq i}(x - x_j) \end{aligned}$$

Now, we require that $l_i(x_i) = 1$, thus

$$1 = C_i \cdot \prod_{j \neq i}(x_i - x_j) \Rightarrow C_i = \frac{1}{\prod_{j \neq i}(x_i - x_j)}$$

Thus, for every i, we have

$$\begin{aligned} l_i(x) &= \frac{(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)} \\ &= \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} \end{aligned}$$

Note: This result essentially proves *existence* of a polynomial interpolant of degree n that passes through $n+1$ data points. We can also use it to prove that the Vandermonde matrix V is non-singular. If it *were* singular, a right hand side $\tilde{y} = (y_0, \ldots, y_n)$ would have existed such that $V\tilde{a} = \tilde{y}$ would have no solution, which is a contradiction!

Let's use Lagrange interpolation to compute an interpolating polynomial to the three data points $(-2,-27),(0,-1),(1,0)$.

$$\begin{aligned}\mathcal{P}_2(x) &= -27\frac{(x-0)(x-1)}{(-2-0)(-2-1)} + (-1)\frac{(x-(-2))(x-1)}{(0-(-2))(0-1)} + 0\frac{(x-(-2))(x-0)}{(1-(-2))(1-0)} \\ &= -27\frac{x(x-1)}{6} + \frac{(x+2)(x-1)}{2} = -1+5x-4x^2\end{aligned}$$

Recall form Lecture 8 that this is the same polynomial we computed using the monomial basis!

Let us evaluate the same four quality metrics we saw before for the Vandermonde matrix approach.

- Cost of determining $\mathcal{P}_n(x)$: *very easy.*

  We are essentially able to write a formula for $\mathcal{P}_n(x)$ without solving any systems. However, if we want to write $\mathcal{P}_n(x) = a_0 + a_1 x + \ldots + a_n x^n$, the cost of evaluating the a_i's would be very high! Each $l_i(x)$ would need to be expanded, leading to $O(n^2)$ operations for each $l_i(x)$ implying $O(n^3)$ operations for $\mathcal{P}_n(x)$.

- Cost of evaluating $\mathcal{P}_n(x)$ for an arbitrary x: *significant.*

  We do not really need to compute the a_i's beforehand, if we only need to evaluate $\mathcal{P}_n(x)$ at a select few locations. For each $l_i(x)$ the evaluation requires n subtractions and n multiplications, implying a total of $O(n^2)$ operations (better than $O(n^3)$ for computing the a_i's).

- Availability of derivatives: *not readily available.*

  Differentiating each $l_i(x)$ (since $\mathcal{P}'_n(x) = \sum y_i l'_i(x)$) is not trivial; the above expression has n terms each with $n-1$ products per term.

- Incremental interpolation: *not accomodated.*

Still, Lagrange interpolation is a good quality method if we can accept its limitations.

Newton Interpolation

Newton interpolation is yet another alternative, which enables *both* efficient evaluation *and* allows for incremental construction. Additionally, it allows both the coefficients $\{a_i\}$ as well as the derivative $\mathcal{P}'_n(x)$ to be evaluated efficiently.

For a given set of data points $(x_0, y_0), \ldots, (x_n, y_n)$, the Newton basis functions are given by

$$\pi_j(x) = (x - x_0)(x - x_1) \ldots (x - x_{j-1}) = \prod_{k=1}^{j-1} (x - x_k), \quad j = 0, \ldots, n$$

where we take the value of the product to be 1 when the limits make it vacuous. In the Newton basis, a given polynomial has the form

$$\mathcal{P}_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_{n-1}(x - x_0)(x - x_1) \ldots (x - x_{n-1})$$

From the definition, we see that $\pi_j(x_i) = 0$ for $i < j$, so that the basis matrix A with $a_{ij} = \pi_j(x_i)$ is *lower triangular*. To illustrate Newton interpolation, we use it to determine the interpolating polynomial for the three data points $(-2, -27), (0, -1), (1, 0)$. With the Newton basis, we have the lower triangular linear system

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & x_1 - x_0 & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

For the given data, this system becomes

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 3 & 3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -27 \\ -1 \\ 0 \end{bmatrix}$$

whose solution is $\tilde{a} = (-27, 13, -4)$. Thus, the interpolating polynomial is

$$\mathcal{P}_2(x) = -27 + 13(x + 2) - 4(x + 2)x = -1 + 5x - 4x^2$$

which is the same polynomial we computed earlier!