

## POLYNOMIAL INTERPOLATION

The purpose of this note is to clarify questions and statements in class, and to provide detailed procedures to find polynomial interpolations.

### 1. LAGRANGE INTERPOLATION

We start with *Lagrange polynomials*. Given a continuous function  $f(x)$ , we would like to find a polynomial of degree  $n$  which interpolates  $f$  at given distinct nodes  $\{x_i\}_{i=0}^n$ .

We denote  $\mathbb{P}_n$  as the space of polynomials of degree  $n$  (or less). It is a  $n + 1$  dimensional functional spaces, and therefore could be represented as the span of  $n + 1$  basis functions  $\{\phi_j(x)\}_{j=0}^n$ . A most natural set of basis function in  $\mathbb{P}_n$  is  $\{1, x, x^2, \dots, x^n\}$ .

Our target polynomial  $p_n$  could be expressed (uniquely) as an linear combination of basis functions, namely

$$p_n(x) = \sum_{j=0}^n a_j \phi_j(x).$$

Given the basis, the coefficients  $\{a_j\}_{j=0}^n$  determines the polynomial.

We shall find the coefficients such that  $p_n$  interpolates  $f$  at nodes  $\{x_i\}_{i=0}^n$ ,

$$p_n(x_i) = \sum_{j=0}^n a_j \phi_j(x_i) = f(x_i), \quad \forall i = 0, \dots, n.$$

This is a linear system which  $n + 1$  equations and  $n + 1$  unknowns ( $\{a_j\}_{j=0}^n$ ). It can be written as a matrix form

$$\begin{bmatrix} \phi_0(x_0) & \cdots & \phi_n(x_0) \\ \vdots & \ddots & \vdots \\ \phi_0(x_n) & \cdots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}$$

If the matrix is non-singular, then there exists a unique solution  $\{a_j\}_{j=0}^n$  which solves the equation. Hence, we find the unique interpolating polynomial  $p_n(x)$ .

This general approach is true for any choice of basis functions. We will cover 3 particular choices. To better understand the procedure, we shall take the following example for all 3 approaches.

**Example 1** (Lagrange polynomial). *Find a polynomial  $p_2(x)$  of degree 2 that interpolates  $f$  at  $x = 0, 1, 2$ , where  $f(0) = 1, f(1) = 1, f(2) = 3$ .*

The first approach is to use the natural basis  $\{1, x, \dots, x^n\}$ . In this case, the matrix looks like

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix}$$

The matrix is called *Vandermonde matrix*, and it is non-singular as long as  $x_i$ 's are all distinct. Therefore, by plugging in the information and solve the linear system, we get the interpolating polynomial.

To apply to the example, we have the system

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Hence, we obtain  $p_2(x) = 1 - x + x^2$ .

One disadvantage of this approach is that we have to solve a linear system, which is usually costly. The second approach is designed to avoid solving the linear system.

The idea is very simple. We will find a special set of basis, denoting by  $\{L_j(x)\}_{j=0}^n$  such that

$$L_j(x_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.$$

By doing so, the matrix becomes an identity matrix. Therefore, we get  $a_j = f(x_j)$  directly, and  $p_n(x) = \sum_{j=0}^n f(x_j)L_j(x)$ .

Note that we do not need to solve a nontrivial linear system. Instead, we move the difficulty to finding an appropriate set of basis functions. The (unique) set of basis functions are given as follows. (One can check that it satisfies the condition above.)

$$L_j(x) = \frac{\prod_{k \neq j} (x - x_k)}{\prod_{k \neq j} (x_j - x_k)}.$$

To apply to the example, we get the basis function from the formula

$$\begin{aligned} L_0(x) &= \frac{(x-1)(x-2)}{(0-1)(0-2)} = \frac{1}{2}(x-1)(x-2), \\ L_1(x) &= \frac{(x-0)(x-2)}{(1-0)(1-2)} = -x(x-2), \\ L_2(x) &= \frac{(x-0)(x-1)}{(0-2)(1-2)} = \frac{1}{2}x(x-1). \end{aligned}$$

Therefore, we obtain

$$p_2(x) = L_0(x) + L_1(x) + 3L_2(x) = \frac{1}{2}(x-1)(x-2) - x(x-2) + \frac{3}{2}x(x-1) = x^2 - x + 1.$$

Note that the polynomial  $p_2$  we get from different approaches match with each other.

The third approach, which is usually called *Newton's representation* (or Newton polynomial) uses the following basis  $\{1, (x-x_0), (x-x_0)(x-x_1), \dots, (x-x_0)\cdots(x-x_{n-1})\}$ . This makes the matrix to be lower triangular, and hence easier to solve compare with the first approach. Meanwhile, the basis is easily constructed by given nodes. So it has an advantage

compare with the second approach (especially in the more general interpolating polynomials which we will discuss later).

There is an equivalent but more practical ways to process Newton's representation, using divided differences. The procedure is given in the following table (take  $n = 3$  for simpler illustration).

$$\begin{array}{l|l}
 x = x_0 & f_0 = f(x_0) \\
 x = x_1 & f_1 = f(x_1) \\
 x = x_2 & f_2 = f(x_2) \\
 x = x_3 & f_3 = f(x_3)
 \end{array}
 \begin{array}{l}
 f_{01} = \frac{f_1 - f_0}{x_1 - x_0} \\
 f_{12} = \frac{f_2 - f_1}{x_2 - x_1} \\
 f_{23} = \frac{f_3 - f_2}{x_3 - x_2} \\
 f_{012} = \frac{f_{12} - f_{01}}{x_2 - x_0} \\
 f_{123} = \frac{f_{23} - f_{12}}{x_3 - x_1} \\
 f_{0123} = \frac{f_{123} - f_{012}}{x_3 - x_0}
 \end{array}$$

After calculating divided differences from left to right, the interpolating polynomial is given by

$$p_3(x) = f_0 + f_{01}(x - x_0) + f_{012}(x - x_0)(x - x_1) + f_{0123}(x - x_0)(x - x_1)(x - x_2).$$

Note that  $(f_0, f_{01}, f_{012}, f_{0123})$  are coefficients for the interpolating polynomial with respect to the basis.

We now apply the procedure to the example.

$$\begin{array}{l|l}
 x = 0 & f_0 = 1 \\
 x = 1 & f_1 = 1 \\
 x = 2 & f_2 = 3
 \end{array}
 \begin{array}{l}
 f_{01} = \frac{1-1}{1-0} = 0 \\
 f_{12} = \frac{3-1}{2-1} = 2 \\
 f_{012} = \frac{2-0}{2-0} = 1
 \end{array}$$

And  $p_2(x) = 1 + 0 \cdot (x - 0) + 1 \cdot (x - 0)(x - 1) = x^2 - x + 1$ .

One more advantage of this approach compared with the second approach is that it is more flexible when adding or deleting an interpolating nodes. For example, if we would like to interpolating  $f$  in the example only at points  $x_1 = 1$  and  $x_2 = 2$ , then the interpolating polynomial is  $p_1(x) = f_1 + f_{12}(x - x_1) = 1 + 2(x - 1) = 2x - 1$ . However, the second approach need a completely new set of basis functions.

## 2. HERMITE INTERPOLATION

Hermite polynomial is different from Lagrange polynomial in the sense that not only the value of the polynomial has to match with the function, but also the derivatives. As there are  $2n + 2$  equations ( $n + 1$  for function values and  $n + 1$  for derivatives), we are considering with polynomials in  $\mathbb{P}_{2n+1}$ .

Similar as Lagrange polynomial, we express the interpolating function  $p_{2n+1}(x)$  by basis functions  $\{\phi_j(x)\}_{j=0}^{2n}$ , and the  $(2n + 2)$ -by- $(2n + 2)$  system reads

$$p_{2n+1}(x_i) = \sum_{j=0}^{2n+1} a_j \phi_j(x_i) = f(x_i), \quad p'_n(x_i) = \sum_{j=0}^n a_j \phi'_j(x_i) = f'(x_i), \quad \forall i = 0, \dots, n.$$

Or we write the matrix form

$$\begin{bmatrix} \phi_0(x_0) & \cdots & \phi_{2n+1}(x_0) \\ \phi'_0(x_0) & \cdots & \phi'_{2n+1}(x_0) \\ \vdots & \ddots & \vdots \\ \phi_0(x_n) & \cdots & \phi_{2n+1}(x_n) \\ \phi'_0(x_n) & \cdots & \phi'_{2n+1}(x_n) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_{2n+1} \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f'(x_0) \\ \vdots \\ f(x_n) \\ f'(x_n) \end{bmatrix}$$

We use the same 3 approaches to find the Hermite polynomial with the following example.

**Example 2** (Hermite polynomial). *Find a polynomial  $p_3(x)$  of degree 3 that interpolates  $f$  at  $x = 0, 1$ , where  $f(0) = 1, f'(0) = 0, f(1) = 2, f'(1) = 3$ .*

First, if we use  $\{1, x, x^2, x^3\}$  as basis, the system becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

So,  $p_3(x) = 1 + 0 \cdot x + 0 \cdot x^2 + x^3 = x^3 + 1$ .

Second, if we would like to avoid solving the linear system, we would like to find basis  $\{\phi_j\}_{j=0}^{2n+1}$  such that the matrix is identity matrix. Here is the unique way to make it happen:

$$\begin{aligned} \phi_{2j}(x) &= H_j(x) := [L_j(x)]^2(1 - 2L'_j(x_j)(x - x_j)), \\ \phi_{2j+1}(x) &= K_j(x) := [L_j(x)]^2(x - x_j), \quad j = 0, \dots, n. \end{aligned}$$

Here,  $\{L_j(x)\}_{j=0}^n$  are basis functions in Lagrange interpolation. One can check that

$$H_j(x_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad H'_j(x_i) = 0, \quad \forall i, \quad K'_j(x_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad K_j(x_i) = 0, \quad \forall i,$$

and this ensures that the matrix is an identity matrix. Therefore, the interpolating polynomial is

$$p_{2n+1} = \sum_{j=0}^{2n+1} a_j \phi_j(x) = \sum_{j=0}^n f(x_j) H_j(x) + \sum_{j=1}^n f'(x_j) K_j(x).$$

For the example, we have

$$L_0(x) = \frac{x-1}{0-1} = 1-x, \quad L_1(x) = \frac{x-0}{1-0} = x,$$

and we use the formula to compute

$$\begin{aligned} H_0(x) &= (1-x)^2(1-2 \cdot (-1)(x-0)) = (x-1)^2(2x+1), \\ H_1(x) &= x^2(1-2 \cdot 1(x-1)) = x^2(-2x+3), \\ K_0(x) &= (1-x)^2(x-0) = x(x-1)^2, \\ K_1(x) &= x^2(x-1). \end{aligned}$$

Put everything together, we conclude

$$p_3(x) = 1 \cdot H_0(x) + 2 \cdot H_1(x) + 0 \cdot K_0(x) + 3 \cdot K_1(x) = x^3 + 1.$$

As one can see, the main computational load shifts from solving the linear system to finding the appropriate basis.

The third approach is to use Newton's basis

$$\{1, x - x_0, (x - x_0)^2, (x - x_0)^2(x - x_1), (x - x_1)^2, \dots, (x - x_0)^2 \cdots (x - x_{n-1})^2(x - x_n)\}.$$

Note that each nodes are counted twice as both the function value and the first derivative have to match. (Think about which quadratic function satisfies  $p_2(1) = p_2'(1) = 0$ . The answer is  $p_2(x) = (x - 1)^2$ .)

Plug in the basis to the linear system, the matrix is lower triangular (you can check it!). On the other hand, we can also use the divided differences procedure to find the coefficients. The only difference is that, the value of derivative is used to substitute divided difference in the case of 0/0.

Let us see how it works in the example. We shall relabel  $x_i$  to count each node twice.

$$\begin{array}{l|l} x_0 = 0 & f_0 = 1 \\ & f_{01} = f'(0) = 0 \\ x_1 = 0 & f_1 = 1 & f_{012} = \frac{1-0}{1-0} = 1 \\ & f_{12} = \frac{2-1}{1-0} = 1 & f_{0123} = \frac{2-1}{1-0} = 1 \\ x_2 = 1 & f_2 = 2 & f_{123} = \frac{3-1}{1-0} = 2 \\ & f_{23} = f'(1) = 3 \\ x_3 = 1 & f_3 = 2 \end{array}$$

We then collect coefficients from the table using the same way as for Lagrange interpolation.

$$p_3(x) = 1 + 0 \cdot (x - 0) + 1 \cdot (x - 0)^2 + 1 \cdot (x - 0)^2(x - 1) = x^3 + 1.$$

### 3. MIXED-TYPE INTERPOLATION

In this part, we consider more general interpolations, where each nodes may have different information. Again, we use an example to see how to apply the three approaches.

**Example 3** (Mixed-type interpolation). *Find a polynomial  $p_3(x)$  of degree 3 that interpolates  $f$  at  $x = 0, 1$ , where  $f(0) = 0, f(1) = 0, f'(1) = 2, f''(1) = 6$ .*

The first approach works with no difference. One just write  $p_3(x)$  as the linear combination of natural basis  $\{1, x, x^2, x^3\}$  and setup the linear system for the coefficients from the equations.

$$\begin{cases} f(0) = a_0 = 0 \\ f(1) = a_0 + a_1 + a_2 + a_3 = 0 \\ f'(1) = a_1 + 2a_2 + 3a_3 = 2 \\ f''(1) = 2a_2 + 6a_3 = 6 \end{cases} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 6 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \end{bmatrix}.$$

Therefore,  $p_3(x) = x^3 - x$ .

For the second approach, it is quite tricky to find basis such that the matrix becomes an identity. Note that such basis uniquely exists. But there is no formula in general. Hence, it is not recommended since to find the basis are costly.

The third approach works nicely in this general setup, and it is the preferable way since there is no need to solve the linear system. One need to count different nodes different times based on the number of derivatives to match for a node.

We solve the example using this procedure as follows.

$$\begin{array}{l|l}
x_0 = 0 & f_0 = 0 \\
x_1 = 1 & f_1 = 0 \\
x_2 = 1 & f_2 = 0 \\
x_3 = 1 & f_3 = 0
\end{array}
\begin{array}{l}
f_{01} = \frac{0-0}{1-0} = 0 \\
f_{12} = f'(1) = 2 \\
f_{23} = f'(1) = 2 \\
f_{012} = \frac{2-0}{1-0} = 2 \\
f_{123} = \frac{f''(1)}{2} = 3 \\
f_{0123} = \frac{3-2}{1-0} = 1
\end{array}$$

One important remark here is: to use derivative to substitute divided difference, one has to use  $f^{(n)}(x_i)/(n!)$  to substitute  $n$ -th divided difference.

Finally, to conclude, the interpolating polynomial reads

$$p_3(x) = 0 + 0 \cdot (x - 0) + 2 \cdot (x - 0)(x - 1) + 1 \cdot (x - 0)(x - 1)^2 = x^3 - x.$$