Polynomial Interpolation (25 points)

Name: Yuhao Wu
ID Number: 260711365

2018-11-30

The Vandermonde Approach

Given $(n+1)$ points, $(x_0, y_0), (x_1, y_1), \dots (x_n, y_n)$, there is a **unique** polynomial p of degree $\leq n$ such that $P(x_i) = y_i$

Proof. Let $p(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_n x^n$, then we have $Ac = y$

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

As A is called the *Vandermonde Matrix* and

$$\det(A) = \prod_{0 \leq i \leq j \leq n} (x_j - x_i) \neq 0$$

Thus A is non-singular and $Ac = y$ has a unique solution $c = A^{-1} y$ $\square$

Algorithm:

1 Form the linear system $Ac = y$

2 Solve $Ac = y$ by GEPP

Cost Analysis:

1 For First Step, for each line of the matrix, we need $n-1$ multiplication based on $x_0, x_1, \dots, x_n$, thus, we need total $(n-1) \times (n+1) \approx n^2$ flops

2 For GEPP, we need $\dfrac{2}{3} n^3$ flops, due to A has some special structures, we can cost as low as $O(n \cdot \log^2(n))$

Evaluating $p(x)$ (2n flops):

$$\begin{aligned} p(x) &= c_0 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_n \cdot x^n \\ &= c_0 + x\Big(c_1 + x\Big(c_2 + \dots + x(c_{n-1} + x \cdot c_n)\Big)\Big) \end{aligned}$$

$p \leftarrow c_n$
for $i = n-1 : -1 : 0$
$p \leftarrow c_i + x \cdot p$
end

The Lagrange Approach:

The Lagrange form of the interpolating polynomial:

$$p(x) = \sum_{i=0}^{n} \ell_i(x) \cdot y_i$$

where $\ell_i(x)$ is the cardinal polynomial defined as:

$$\ell_i(x) = \frac{(x - x_0) \cdot (x - x_1) \dots (x - x_{i-1}) \cdot (x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \cdot (x_i - x_1) \dots (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \dots (x_i - x_n)} \quad \ell_i(x_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

We can rewrite $p(x)$:

$$p(x) = \sum_{i=0}^{n} \ell_i(x) \cdot y_i = \sum_{i=0}^{n} \frac{y_i}{\prod_{j=0, j\neq i}^{n} (x_i - x_j)} \cdot \frac{\prod_{j=0}^{n} (x - x_j)}{x - x_i} = q(x) \cdot \sum_{i=0}^{n} \frac{c_i}{x - x_i}$$

$$where\ q(x) = \prod_{j=0}^{n} (x - x_j) \qquad c_i = \frac{y_i}{\prod_{j=0, j\neq i}^{n} (x_i - x_j)}$$

Cost of Finding $c_0, c_1, \dots, c_n$
For each i, computing c_i needs 1 division, n subtraction, $n - 1$ multiplication, a total $2n$ flops.
So, computing all c_i needs $2n \times (n + 1) \approx 2n^2$ flops.

Cost of Evaluating $p(x)$**:**
Computing $q(x)$ needs $(2n + 1)$ flops (n+1 subtraction, n multiplication)
Computing each $\frac{c_i}{x - x_i}$ needs 2 flops for each i, total $2 \times (n + 1)$
Adding them together, need n flops.
Thus, we need total $(2n + 1) + 2 \times (n + 1) + n \approx 5n$

The Newton Approach

Idea: Suppose a polynomial $p_k(x)$ of degree at most k has been found to interpolate $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$.
Then, what we want to do is to find $p_{k+1}(x)$ of degree at most k+1 to interpolate $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k), (x_{k+1}, y_{k+1})$

Set $p_{k+1} = p_k(x) + a_{k+1}(x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_k)$, where a_{k+1} is to be determined.

Obviously, we have

$$p_{k+1}(x_i) = p_k(x_i) = y_i \quad 0 \leq i \leq k$$

Setting $p_{k+1}(x_{k+1}) = y_{k+1}$, we obtain:

$$a_{k+1} = \frac{y_{k+1} - p_k(x_{k+1})}{(x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_k)}$$

The Newton form of the interpolating polynomial:

$$p_n(x) = a_0 + a_1 \cdot (x - x_0) + a_2 \cdot (x - x_0)(x - x_1) + \dots + a_n \cdot (x - x_0) \cdot (x - x_1) \dots (x - x_{n-1})$$

Evaluating: (3n) flops

$$
\begin{aligned}
p_n(x) &= a_0 + a_1 \cdot (x - x_0) + a_2 \cdot (x - x_0)(x - x_1) + \ldots + a_n \cdot (x - x_0) \cdot (x - x_1) \ldots (x - x_{n-1}) \\
&= a_0 + (x - x_0) \cdot \Big(a_1 + (x - x_1) \Big(a_2 + \ldots + (a_{n-1} + (x - x_{n-1}) a_n) \Big) \Big)
\end{aligned}
$$

Procedure for Evaluating $p_n(x)$ for some x:
$p \leftarrow a_n$
for $i = n-1 : -1 : 0$
$p \leftarrow a_i + (x - x_i) \cdot p$
end

Cost of Computing $a_1, a_2, \ldots, a_n$:

$$a_{k+1} = \frac{y_{k+1} - p_k(x_{k+1})}{(x_{k+1} - x_0)(x_{k+1} - x_1) \ldots (x_{k+1} - x_k)}$$

Cost of computing a_{k+1}:

1 Compute $p_k(x_{k+1})$ we need $3k$ flops, so for the numerator needs $3k + 1$ flops.

2 for denominator, we need k flops for multiplication, $k + 1$ flops for subtraction.

So, there are total $5k + 2 + 1$ (for division) $= 5k + 3$ flops.

Total cost

$$\sum_{k=0}^{n-1} (5k + 3) = \frac{5}{2} n^2 + \frac{1}{2} n \approx \frac{5}{2} n^2 \ flops$$

A more Efficient Method for computing $a_0, a_1, a_2, \ldots a_n$:
As we know that, $p_n(x)$ is always in this form:

$$p_n(x) = \sum_{i=0}^{n} a_i \cdot \prod_{j=0}^{i-1} (x - x_j)$$

interpolates (x_i, y_i) for $i = 0, 1, 2, \ldots, n$, we have:

$$p_n(x_i) = y_i, \quad i = 0, 1, 2, \ldots, n$$

Thus, we build the linear system $Aa = y$

$$
\begin{bmatrix}
1 & & & & & \\
1 & x_1 - x_0 & & & & \\
1 & x_2 - x_0 & \prod_{j=0}^{1}(x_2 - x_j) & & & \\
1 & x_3 - x_0 & \prod_{j=0}^{1}(x_3 - x_j) & \prod_{j=0}^{2}(x_3 - x_j) & & \\
\vdots & \vdots & \vdots & \vdots & & \\
1 & x_n - x_0 & \prod_{j=0}^{1}(x_n - x_j) & \prod_{j=0}^{2}(x_n - x_j) & \ldots & \prod_{j=0}^{n-1}(x_n - x_j)
\end{bmatrix}
\cdot
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}
=
\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}
$$

Algorithm 1 Newton's Approach

Input: nodes (x_i, y_i)

for $k = 0 : n-1$ **do**

  $a_k \leftarrow y_k$

  for $i = k+1 : n$ **do**

    $y_i \leftarrow \dfrac{y_i - y_k}{x_i - x_k}$

  end for

end for

For the inner "For-Loop", there are $3(n-k)$ flops.
Thus there are total:

$$\sum_{k=0}^{n-1} 3 \cdot (n-k) = \frac{3}{2} n \cdot (n+1) \approx \frac{3}{2} n^2$$

EXAMPLE:

x	-2	0	1	2
y	2	4	2	2

The Vandermonde approach:

Let $p(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + c_3 x^3$

$$\begin{cases} p(-2) = 2 \\ p(0) = 4 \\ p(1) = 2 \\ p(2) = 2 \end{cases} \implies \begin{cases} c_0 - 2 \cdot c_1 + 4 \cdot c_2 - 8 \cdot c_3 = 2 \\ c_0 + 1 \cdot c_1 + 1 \cdot c_2 + 1 \cdot c_3 = 2 \\ c_0 + 2 \cdot c_1 + 4 \cdot c_2 + 8 \cdot c_3 = 2 \\ c_0 = 4 \end{cases}$$

Thus, we have

$$c_0 = 4 \quad c_1 = -2 \quad c_2 = -\frac{1}{2} \quad c_3 = \frac{1}{2}$$

Thus, we have

$$p(x) = \frac{1}{2} x^3 - \frac{1}{2} x^2 - 2x + 4$$

The Lagrange approach

We write $p(x)$ in the Lagrange Form

$$p(x) = l_0(x) y_0 + l_1(x) y_1 + l_2(x) y_2 + l_3(x) y_3 \quad \textit{where}$$

$$l_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} = \frac{(x-0)(x-1)(x-2)}{(-2-0)(-2-1)(-2-2)} = -\frac{1}{24} \cdot (x^3 - 3 \cdot x^2 + 2 \cdot x)$$

$$l_1(x) = \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} = \frac{(x+2)(x-1)(x-2)}{(0+2)(0-1)(0-2)} = \frac{1}{4} \cdot (x^3 - x^2 - 4 \cdot x + 4)$$

$$l_2(x) = \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} = \frac{(x+2)(x-0)(x-2)}{(1+2)(1-0)(1-2)} = -\frac{1}{3} \cdot (x^3 - 4 \cdot x)$$

$$l_3(x) = \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} = \frac{(x+2)(x-0)(x-1)}{(2+2)(2-0)(2-1)} = \frac{1}{8} \cdot (x^3 + x^2 - 2 \cdot x)$$

Thus, we have

$$
\begin{aligned}
p(x) &= 2 \cdot (-\frac{1}{24}) \cdot (x^3 - 3 \cdot x^2 + 2 \cdot x) \\
&+ 4 \cdot \frac{1}{4} \cdot (x^3 - x^2 - 4 \cdot x + 4) \\
&+ 2 \cdot (-\frac{1}{3}) \cdot (x^3 - 4 \cdot x) \\
&+ 2 \cdot \frac{1}{8} \cdot (x^3 + x^2 - 2 \cdot x) \\
&= \frac{1}{2}x^3 - \frac{1}{2}x^2 - 2x + 4
\end{aligned}
$$

Newton form

Suppose that

$$p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2)$$

The coefficient is calculated according to the algorithm given in class:

- $x_0 = -2 \quad y_0 = 2 \qquad a_0 : a_0 = 2$
- $x_1 = 0 \quad y_1 = 4 \qquad a_1 : \dfrac{4-2}{0-(-2)} = 1 \implies a_1 = 1$
- $x_2 = 1 \quad y_2 = 2 \qquad a_2 : \dfrac{2-2}{1-(-2)} = 0 \quad \dfrac{0-1}{1-0} = -1 \implies a_2 = -1$
- $x_3 = 2 \quad y_3 = 2 \qquad a_3 : \dfrac{2-2}{2-(-2)} = 0 \quad \dfrac{0-1}{2-0} = -\dfrac{1}{2} \quad \dfrac{-1/2-(-1)}{2-1} = \dfrac{1}{2} \implies a_3 = \dfrac{1}{2}$

So, we can say that $a_0 = 2 \quad a_1 = 1 \quad a_2 = 1 \quad a_3 = \frac{1}{2}$

Thus, we have:

$$p(x) = 2 + 1 \cdot (x+2) - 1 \cdot (x+2)(x-0) + \frac{1}{2} \cdot (x+2)(x-0)(x-1)$$

$$p(x) = \frac{1}{2}x^3 - \frac{1}{2}x^2 - 2x + 4$$

Question: Why a high degree Polynomial Interpolation is not a good idea ?

f will not be well approximate at all intermediate points as the number of nodes increases.

Take *Runge function* for example:

$$f(x) = \frac{1}{1+25x^2}, x \in [-1, 1]$$

If p_n is the polynomial that interpolates the f at $n+1$ equally spaced points on $[-1, 1]$, then

$$\lim_{n\to\infty} \max_{-1\le x\le 1} |f(x) - p_n(x)| = \infty$$