Mustafa Sadiq

HW2 for CS323 Fall 2020

Answer to question 1

$$p(t) = -2 + 7t - 3t^2 + 5t^3$$

$$\boldsymbol{p(t) = -2 + t(7 + t(-3 + 5t))}$$

Answer to question 2 part (a)

$\boldsymbol{n - 1}$ multiplications [$O(n)$]

Answer to question 2 part (b)

$\boldsymbol{n(2n - 1)}$ multiplications [$O(n^2)$]

Answer to question 2 part (c)

$\boldsymbol{n}$ multiplications [$O(n)$]

Answer to question 3 part (a)

t	1	2	3	4
y	11	29	65	125

For interpolating the 4 points with a degree 3 polynomial:

$$y = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

The Vandermonde system will look like:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 29 \\ 65 \\ 125 \end{bmatrix}$$

Solving the matrix equation:

$$a_0 + a_1 + a_2 + a_3 = 11$$

$$a_0 + 2a_1 + 4a_2 + 8a_3 = 29$$

$$a_0 + 3a_1 + 9a_2 + 27a_3 = 65$$

$$a_0 + 4a_1 + 16a_2 + 64a_3 = 125$$

Solving system of equations:

$$a_0 = 5, a_1 = 2, a_2 = 3, a_3 = 1$$

Polynomial interpolation using monomial basis:

$$\boldsymbol{y = 5 + 2t + 3t^2 + t^3}$$

Answer to question 3 part (b)

Lagrange Interpolation:

$$P_n(x) = y_1 l_1(x) + y_2 l_2(x) + \cdots + y_{n+1} l_{n+1}(x)$$

Using given data:

$$y = 11 l_1(t) + 29 l_2(t) + 65 l_3(t) + 125 l_4(t)$$

The first Lagrange polynomial can be computed as follows:

$$l_1 = \frac{(t - t_2)(t - t_3)(t - t_4)}{(t_1 - t_2)(t_1 - t_3)(t_1 - t_4)} = \frac{(t-2)(t-3)(t-4)}{(1-2)(1-3)(1-4)} = -\frac{(t-2)(t-3)(t-4)}{6}$$

The second Lagrange polynomial can be computed as follows:

$$l_2 = \frac{(t - t_1)(t - t_3)(t - t_4)}{(t_2 - t_1)(t_2 - t_3)(t_2 - t_4)} = \frac{(t-1)(t-3)(t-4)}{(2-1)(2-3)(2-4)} = \frac{(t-1)(t-3)(t-4)}{2}$$

The third Lagrange polynomial can be computed as follows:

$$l_3 = \frac{(t - t_1)(t - t_2)(t - t_4)}{(t_3 - t_1)(t_3 - t_2)(t_3 - t_4)} = \frac{(t-1)(t-2)(t-4)}{(3-1)(3-2)(3-4)} = -\frac{(t-1)(t-2)(t-4)}{2}$$

The fourth Lagrange polynomial can be computed as follows:

$$l_4 = \frac{(t - t_1)(t - t_2)(t - t_3)}{(t_4 - t_1)(t_4 - t_2)(t_4 - t_3)} = \frac{(t-1)(t-2)(t-3)}{(4-1)(4-2)(4-3)} = \frac{(t-1)(t-2)(t-3)}{6}$$

Entire polynomial can be computed as:

$$\begin{aligned} y = 11\left(-\frac{(t-2)(t-3)(t-4)}{6}\right) + 29\left(\frac{(t-1)(t-3)(t-4)}{2}\right) \\ + 65\left(-\frac{(t-1)(t-2)(t-4)}{2}\right) + 125\left(\frac{(t-1)(t-2)(t-3)}{6}\right) \\ = \boldsymbol{t^3 + 3t^2 + 2t + 5} \end{aligned}$$

The resulting polynomial is equivalent to that obtained in part (a).

Answer to question 3 part (c)

Newton Interpolation

$$P_n(x) = a_0\pi_1(x) + a_1\pi_2(x) + \cdots + a_n\pi_{n+1}(x)$$

For interpolating the 4 points with a degree 3 polynomial:

$$y = a_0 + a_1(t - t_1) + a_2(t - t_1)(t - t_2) + a_3(t - t_1)(t - t_2)(t - t_3)$$

$$y = a_0 + a_1(t - 1) + a_2(t - 1)(t - 2) + a_3(t - 1)(t - 2)(t - 3)$$

Newton polynomial interpolant using triangular matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 2-1 & 0 & 0 \\ 1 & 3-1 & (3-1)(3-2) & 0 \\ 1 & 4-1 & (4-1)(4-2) & (4-1)(4-2)(4-3) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 29 \\ 65 \\ 125 \end{bmatrix}$$

Solving this system gives us coefficients:

$$a_0 = 11, a_1 = 18, a_2 = 9, a_3 = 1$$

Thus, the interpolating polynomial is:

$$y = 11 + 18(t - 1) + 9(t - 1)(t - 2) + 1(t - 1)(t - 2)(t - 3)$$

$$\boldsymbol{y = t^3 + 3t^2 + 2t + 5}$$

The resulting polynomial is equivalent to that obtained in the previous two parts.

Newton polynomial interpolant using incremental interpolation:

Initially define the polynomial as:

$$P_0(t) = y_0 = 11$$

Adding $(t_1, y_1) = (2{,}29)$, we modify the polynomial as:

$$P_1(t) = P_0(t) + M_1(t) = P_0(t) + c_1(t - t_0)$$

$$= P_0(t) + \frac{y_1 - P_0(t)}{t_1 - t_0}(t - t_0)$$

$$= 11 + \frac{29 - 11}{2 - 1}(t - 1)$$

$$= -7 + 18t$$

Adding $(t_2, y_2) = (3, 65)$, we modify the polynomial as:

$$P_2(t) = P_1(t) + M_2(t)$$

$$= P_1(t) + c_2(t - t_0)(t - t_1)$$

$$= P_1(t) + \frac{y_2 - P_1(t_2)}{(t_2 - t_0)(t_2 - t_1)}(t - t_0)(t - t_1)$$

$$= -7 + 18t + \frac{65 - 47}{(3 - 1)(3 - 2)}(t - 1)(t - 2)$$

$$= 9t^2 - 9t + 11$$

Adding $(t_3, y_3) = (4, 125)$, we modify the polynomial as:

$$P_3(t) = P_2(t) + M_3(t)$$

$$= P_2(t) + c_3(t - t_0)(t - t_1)(t - t_2)$$

$$= P_2(t) + \frac{y_3 - P_2(t_3)}{(t_3 - t_0)(t_3 - t_1)(t_3 - t_2)}(t - t_0)(t - t_1)(t - t_2)$$

$$= 9t^2 - 9t + 11 + \frac{125 - 119}{(4 - 1)(4 - 2)(4 - 3)}(t - 1)(t - 2)(t - 3)$$

$$= \boldsymbol{t^3 + 3t^2 + 2t + 5}$$

The resulting polynomial is equivalent to that obtained in the previous two parts.

Newton polynomial interpolant using divided differences:

$$y = f[t_0] + f[t_o, t_1](t - t_o) + f[t_o, t_1, t_2](t - t_o)(t - t_1) + f[t_o, t_1, t_2, t_3](t - t_o)(t - t_1)(t - t_2)$$

$$f[t_0] = 11$$

$$f[t_o, t_1] = \frac{29 - 11}{2 - 1} = 18$$

$$f[t_o, t_1, t_2] = \frac{f[t_1, t_2] - f[t_0, t_1]}{3 - 1} = \frac{36 - 18}{2} = 9$$

$$f[t_o, t_1, t_2, t_3] = \frac{f[t_1, t_2, t_3] - f[t_o, t_1, t_2]}{4 - 1} = \frac{12 - 9}{3} = 1$$

$$y = 11 + 18(t - 1) + 9(t - 1)(t - 2) + 1(t - 1)(t - 2)(t - 3) = \boldsymbol{t^3 + 3t^2 + 2t + 5}$$

The resulting polynomial is equivalent to that obtained in the previous two parts.

Answer to question 4 part (a)

$$\frac{d}{dt}\prod_{i=1}^{n}(t - t_i) = \prod_{i=1}^{n}(t - t_i)'\prod_{j \neq i}(t - t_i)$$

$$= \prod_{i=1}^{n} 1 \prod_{j \neq i}(t - t_i)$$

Substituting t_j for t

$$\pi'(t_j) = \prod_{j \neq i}(t_j - t_i)$$

$$\boldsymbol{\pi'(t_j) = (t_j - t_1) \dots (t_j - t_{j-1})(t_j - t_{j+1}) \dots (t_j - t_n)}$$

<u>Answer to question 4 part (b)</u>

$$\pi'(t_j) = (t_j - t_1) \dots (t_j - t_{j-1})(t_j - t_{j+1}) \dots (t_j - t_n)$$

Lagrange basis function

$$\frac{(t - t_1) \dots (t - t_{j-1})(t - t_{j+1}) \dots (t - t_n)}{\pi'(t_j)}$$

Multiplying by $\frac{(t-t_j)}{(t-t_j)}$

$$\frac{(t - t_1) \dots (t - t_{j-1})(t - t_{j+1}) \dots (t - t_n)(t - t_j)}{\pi'(t_j)(t - t_j)}$$

jth Lagrange basis function can be expressed as

$$\boldsymbol{l_j(t) = \frac{\pi(t)}{\pi'(t_j)(t - t_j)}}$$

Answer to question 5

```
import numpy as np

print("Mustafa Sadiq (ms3035)")
print("Answer to question 5")
print("--------------------\n")

def horners (poly, n, t):
    answer = poly[n]
    derivative = 0
    for i in range(0, n):
        derivative = answer + t * derivative
        answer = poly[n - 1 - i] + t * answer
    return answer, derivative

print ("p(t) = -1 + 2t - 6t^2 + 2t^3, t0 = 3")
poly = np.array([-1, 2, -6 , 2])
n = len(poly) - 1
t = 3
result, derivative = horners (poly, n, t)
print ("p(t0) =", result) #5
print ("p'(t0) =", derivative) #20

print("\n")

print ("p(t) = 1 + 3t + 0t^2 + 2x^3, t0 = 2")
poly = np.array ([1, 3, 0, 2])
n = len(poly) - 1
t = 2
result, derivative = horners (poly, n, t)
print ("p(t0) =", result) #23
print ("p'(t0) =", derivative) #27
```