Math 391

Worksheet 1 Q1: Notes and comments
Polynomial Interpolation

RJ Vestrum

September, 2019

Notes for Question 1

Find the polynomial the interpolates the data points

$$\Big\{(-1,-1),(0,3),(2,11),(3,27)\Big\}$$

This is a homework problem you solved in Math 211 in the first assignment. It is very easy to over think this problem.
So there is a series of data points and we want to be able to find the polynomial that interpolates those points. In Math 211 we learned tools to solve linear systems, so if we can represent this as a system of equations we already have the tools to solve this.

Recall from Linear Algebra that to get a unique solution, the most variables we can have is equal to the number of equations we have. Since we have 4 data points, we will be able to create 4 equations. So we are going to solve for 4 coefficients, or a polynomial of order 3.

The polynomial will be in the form: $c_1x^3 + c_2x^2 + c_3x^1 + c_4 = y$

With this equation we can use our x and y pairs to create a series of equations:

$$\begin{aligned}
c_1x_1^3 + c_2x_1^2 + c_3x_1^1 + c_4 &= y_1\\
c_1x_2^3 + c_2x_2^2 + c_3x_2^1 + c_4 &= y_2\\
c_1x_3^3 + c_2x_3^2 + c_3x_3^1 + c_4 &= y_3\\
c_1x_4^3 + c_2x_4^2 + c_3x_4^1 + c_4 &= y_4
\end{aligned}$$

This system might be better represented using matrix multiplication. We can also right the system of equations as:

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1^1 & x_1^0\\ x_2^3 & x_2^2 & x_2^1 & x_2^0\\ x_3^3 & x_3^2 & x_3^1 & x_3^0\\ x_4^3 & x_4^2 & x_4^1 & x_4^0 \end{bmatrix} \begin{bmatrix} c_1\\ c_2\\ c_3\\ c_4 \end{bmatrix} = \begin{bmatrix} y_1\\ y_2\\ y_3\\ y_4 \end{bmatrix}$$

If we multiply both sides by the inverse of the first matrix we get:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} x_1^3 & x_1^2 & x_1^1 & x_1^0 \\ x_2^3 & x_2^2 & x_2^1 & x_2^0 \\ x_3^3 & x_3^2 & x_3^1 & x_3^0 \\ x_4^3 & x_4^2 & x_4^1 & x_4^0 \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

This matrix can be generated via matlab, your code might look like:

```
n=3;
A=zeros(length(x),n+1)
for i = 1:length(x)
    for i = 1:(n+1)
        A(i,j)=x(i)^{(n+1-j)};
    end
end

c=inv(A)*y'
```

y' is used to transpose y from a horizontal to vertical vector. Takes the dimension from (1x4) to (4x1)

This result, c, should match the result of the matlab function polyfit(x,y,3).

```
A\y'=
 1.0000
-1.0000
 2.0000
 3.0000
```

If you are using the Matlab function Vander you should read the help file to compare how it generates the matrix compared to above.

To get the same result, you will need to use the code:

```
A = fliplr(vander(x))
```