Chapter 2: Polynomial Interpolation

Contents

1 Polynomial Interpolation

In this chapter we study how to interpolate a data set with a polynomial.

Problem Description: Given (n+1) points, say(x_i, y_i), where i = 0,1,2,. . . ,n, with distinct x_i, not necessarily sorted, we want to find a polynomial of degree n:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$$

such that it interpolates these points, i.e.,

$$P_n(x) = y_i, i = 0, 1, 2, ..., n$$

The goal is the determine the coefficients $a_n, a_{n-1}, ..., a_1, a_0$. Note the total number of data poitns is 1 larger than the degree of the polynomial.

Why should we do this?

- Find the values between the points for discrete data set
- To approximate a (probably complicated) function by a polynomial
- Then, it is easier to do computations such as derivative, integrations, etc. . .

1.1 Van Der Monde Matrix Example

Interpolate the given data set with a polynomial of degree 2:

x_i	0	1	2/3
y_i	1	0	0.5

Answer Let

$$P_2(x) = a_2 x^2 + a_1 x + a_0$$

We need to find the coefficients a_2, a_1, a_0. By the interpolating properties, we have 3 equations:

$$x = 0, y = 1 : P_2(0) = a_0 = 1$$

$$x = 1, y = 0 : P_2(1) = a_2 + a_1 + a_0 = 0$$

$$x = \frac{2}{3}, y = 0.5 : P_2(\frac{2}{3}) = (\frac{4}{9})a_2 + (\frac{2}{3})a_1 + a_0 = 0.5$$

Here we have 3 linear equations and 3 unknowns (a_2, a_1, a_0).

The equations:

$$a_0 = 1$$
$$a_2 + a_1 = 0$$
$$\frac{4}{9}a_2 + \frac{2}{3}a_1 + a_0 = 0.5$$

In matrix-vector form:

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ \frac{4}{9} & \frac{2}{3} & 1 \end{pmatrix} \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0.5 \end{pmatrix}$$

Easy to solve in Matlab, or do it by hand:

$$a_2 = -\frac{3}{4}$$
$$a_1 = -\frac{1}{4}$$
$$a_0 = 1$$

Then

$$P_2(x) = -\frac{3}{4}x^2 - \frac{1}{4}x + 1$$

1.1.1 The general case

- X: a (n+1) x (n+1) matrix, given (Van Der Monde matrix)
- a: unknown vector, with length (n + 1)
- y: given vector, with length (n + 1)

Theorem If x_i's are distinct, then X is invertible, therefore a has a unique solution.

In Matlab, the command vander(x), where x is a vector that contains the interpolation points x = $[x_1, x_2, ..., x_n]$, will generate this matrix.

Bad News: X has a very large condition number for large n, therefore not effective to solve if n is large.

The general case. For the general case with $(n+1)$ points, we have

$$P_n(x_i) = y_i, \qquad i = 0, 1, 2, \cdots, n$$

We will have $(n+1)$ equations and $(n+1)$ unknowns:

$$\begin{aligned}
P_n(x_0) = y_0 \quad &: \quad x_0^n a_n + x_0^{n-1} a_{n-1} + \cdots + x_0 a_1 + a_0 &= y_0 \\
P_n(x_1) = y_1 \quad &: \quad x_1^n a_n + x_1^{n-1} a_{n-1} + \cdots + x_1 a_1 + a_0 &= y_1 \\
\vdots \quad & \\
P_n(x_n) = y_n \quad &: \quad x_n^n a_n + x_n^{n-1} a_{n-1} + \cdots + x_n a_1 + a_0 &= y_n
\end{aligned}$$

Putting this in matrix-vector form

$$\begin{pmatrix} x_0^n & x_0^{n-1} & \cdots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \cdots & x_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^n & x_n^{n-1} & \cdots & x_n & 1 \end{pmatrix} \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

i.e.

$$\mathbf{X}\,\vec{a} = \vec{y}$$

Figure 1: