CMPSC/Math 451, Numerical Computation

Wen Shen

Department of Mathematics, Penn State University

Chapter 2: 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 $i = 0, 1, 2, \cdots, n$, with distinct x_i, not necessarily sorted, we want to find a polynomial of degree n,

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

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

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

The goal is to determine the coefficients $a_n, a_{n-1}, \cdots, a_1, a_0$.

NB! The total number of data points is 1 larger than the degree of the polynomial.

Why should we do this? Here are some reasons:

- 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, integration etc.

Example 1. 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

$$P_2(x) = a_2x^2 + a_1x + a_0$$

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

$$\begin{aligned} x = 0,\ y = 1 \quad &: \quad P_2(0) = a_0 = 1 \\ x = 1,\ y = 0 \quad &: \quad P_2(1) = a_2 + a_1 + a_0 = 0 \\ x = 2/3,\ y = 0.5 \quad &: \quad P_2(2/3) = (4/9)a_2 + (2/3)a_1 + a_0 = 0.5 \end{aligned}$$

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

The equations:

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

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:

$$a_2 = -3/4, \quad a_1 = -1/4, \quad a_0 = 1.$$

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:

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

Putting this in matrix-vector form

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

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

$\mathbf{X}$: $(n+1)\times(n+1)$ matrix, given
(It's called the van der Monde matrix)
$\vec{a}$: unknown vector, with length $(n+1)$
$\vec{y}$: given vector, with length $(n+1)$

Theorem: If x_i's are distinct, then $\mathbf{X}$ is invertible, therefore $\vec{a}$ has a unique solution.

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

Bad news: $\mathbf{X}$ has very large condition number for large n, therefore not effective to solve if n is large.

Other more efficient and elegant methods include

- Lagrange polynomials
- Newton's divided differences