MA579H Scientific Computing

Polynomial Interpolation-I

Rafikul Alam
Department of Mathematics
IIT Guwahati

Lecture outline

- Vandermonde interpolating polynomial
- Lagrange interpolating polynomial
- Barycentric Lagrange interpolation

Polynomial interpolation

Problem: Given a data set $(x_0, f_0), \ldots, (x_n, f_n)$ consisting of

$$\text{distinct nodes:} \qquad [x_0, x_1, \ldots, x_n]$$
$$\text{and values:} \qquad [f_0, f_1, \ldots, f_n],$$

construct a polynomial $p(x)$ of lowest degree such that $p(x_j) = f_j$ for $j = 0 : n$.

Polynomial interpolation

Problem: Given a data set $(x_0, f_0), \ldots, (x_n, f_n)$ consisting of

$$\begin{array}{rl} \text{distinct nodes:} & [x_0, x_1, \ldots, x_n] \\ \text{and values:} & [f_0, f_1, \ldots, f_n], \end{array}$$

construct a polynomial $p(x)$ of lowest degree such that $p(x_j) = f_j$ for $j = 0 : n$.

Questions:

- Does an interpolating polynomial exist? If it exists, is it unqiue? How to compute an interpolating polynomial?

Polynomial interpolation

Problem: Given a data set $(x_0, f_0), \ldots, (x_n, f_n)$ consisting of

$$\text{distinct nodes:} \quad [x_0, x_1, \ldots, x_n]$$
$$\text{and values:} \quad [f_0, f_1, \ldots, f_n],$$

construct a polynomial $p(x)$ of lowest degree such that $p(x_j) = f_j$ for $j = 0 : n$.

Questions:

- Does an interpolating polynomial exist? If it exists, is it unqiue? How to compute an interpolating polynomial?
- Does the choice of nodes have any impact on the behaviour of an interpolating polynomial? If yes, how to choose optimal nodes?

Example

Consider the data set $(x_0, f_0), (x_1, f_1), (x_2, f_2)$. Then there is a unique interpolating polynomial $p(x) = a_0 + a_1 x + a_2 x^2$ of degree ≤ 2.

Example

Consider the data set $(x_0, f_0), (x_1, f_1), (x_2, f_2)$. Then there is a unique interpolating polynomial $p(x) = a_0 + a_1 x + a_2 x^2$ of degree ≤ 2.

The conditions $p(x_0) = f_0, p(x_1) = f_1, p(x_2) = f_2$ gives

$$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}.$$

Example

Consider the data set $(x_0, f_0), (x_1, f_1), (x_2, f_2)$. Then there is a unique interpolating polynomial $p(x) = a_0 + a_1 x + a_2 x^2$ of degree ≤ 2.

The conditions $p(x_0) = f_0, p(x_1) = f_1, p(x_2) = f_2$ gives

$$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}.$$

For the data set $(-2, -27), (0, -1), (1, 0)$, we have

$$\begin{bmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -27 \\ -1 \\ 0 \end{bmatrix}.$$

Example

Consider the data set $(x_0, f_0), (x_1, f_1), (x_2, f_2)$. Then there is a unique interpolating polynomial $p(x) = a_0 + a_1 x + a_2 x^2$ of degree ≤ 2.

The conditions $p(x_0) = f_0, p(x_1) = f_1, p(x_2) = f_2$ gives

$$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}.$$

For the data set $(-2, -27), (0, -1), (1, 0)$, we have

$$\begin{bmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -27 \\ -1 \\ 0 \end{bmatrix}.$$

Solving the system, we have $[a_0, a_1, a_2] = [-1, 5, -4]$ so that the interpolating polynomial is given by

$$p(x) = -1 + 5x - 4x^2.$$

Vandermonde interpolating polynomial

Theorem Consider the nodes $[x_0, \ldots, x_n]$ and the values $[f_0, \ldots, f_n]$. There exists a unique polynomial of degree at most n such that $p(x_j) = f_j$, $j = 0, \ldots, n$.

Vandermonde interpolating polynomial

Theorem Consider the nodes $[x_0, \ldots, x_n]$ and the values $[f_0, \ldots, f_n]$. There exists a unique polynomial of degree at most n such that $p(x_j) = f_j$, $j = 0, \ldots, n$.

Proof: Consider the polynomial $p(x) = a_0 + a_1 x + \cdots + a_n x^n$. Then $p(x_j) = f_j$, $j = 0 : n$ gives the Vandermonde system

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

which has a unique solution (Why?).

Vandermonde interpolating polynomial

Theorem Consider the nodes $[x_0, \ldots, x_n]$ and the values $[f_0, \ldots, f_n]$. There exists a unique polynomial of degree at most n such that $p(x_j) = f_j, \, j = 0, \ldots, n.$

Proof: Consider the polynomial $p(x) = a_0 + a_1 x + \cdots + a_n x^n$. Then $p(x_j) = f_j, \; j = 0 : n$ gives the Vandermonde system

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

which has a unique solution (Why?). Therefore there exists a unique polynomial $p(x)$ of degree at most n passing through the points $(x_j, f_j), \, j = 0, \ldots, n.$ ■

Vandermonde interpolating polynomial

Theorem Consider the nodes $[x_0, \ldots, x_n]$ and the values $[f_0, \ldots, f_n]$. There exists a unique polynomial of degree at most n such that $p(x_j) = f_j$, $j = 0, \ldots, n$.

Proof: Consider the polynomial $p(x) = a_0 + a_1 x + \cdots + a_n x^n$. Then $p(x_j) = f_j$, $j = 0 : n$ gives the Vandermonde system

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

which has a unique solution (Why?). Therefore there exists a unique polynomial $p(x)$ of degree at most n passing through the points (x_j, f_j), $j = 0, \ldots, n$. $\blacksquare$

Remark:

- The Vandermonde matrix is extremely ill-conditioned for moderately large n. The computation is numerically unstable.
- The solution of the system requires $\mathcal{O}(n^3)$ operations.
- Any additional new data (x_{n+1}, f_{n+1}) requires re-computation.
- Evaluation of $p(x)$ at a given x requires $\mathcal{O}(n^2)$ operations.

Interpolating polynomial in a general basis

Let $\mathcal{P}_n$ denote the vector space of polynomials of degree at most n. Let $\phi_0(x), \ldots, \phi_n(x)$ be a basis of $\mathcal{P}_n$. Let $p(x) = a_0\phi_0(x) + \cdots + a_n\phi_n(x)$. Then $p(x_j) = f_j$ for $j = 0 : n$ yield

$$\begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \cdots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & \vdots & \cdots & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \cdots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

Interpolating polynomial in a general basis

Let $\mathcal{P}_n$ denote the vector space of polynomials of degree at most n. Let $\phi_0(x), \ldots, \phi_n(x)$ be a basis of $\mathcal{P}_n$. Let $p(x) = a_0\phi_0(x) + \cdots + a_n\phi_n(x)$. Then $p(x_j) = f_j$ for $j = 0 : n$ yield

$$\begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \cdots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & \vdots & \ldots & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \cdots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

Since the coefficient matrix is nonsingular (Check), the linear system has a unique solution. Thus the polynomial interpolation problem has a unique solution.

Lagrange interpolating polynomial

For $(x_0, f_0), (x_1, f_1), (x_2, f_2)$, define

$$p(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f_2.$$

Then we have $p(x_0) = f_0, p(x_1) = f_1$ and $p(x_2) = f_2$.

Lagrange interpolating polynomial

For $(x_0, f_0), (x_1, f_1), (x_2, f_2)$, define

$$p(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f_2.$$

Then we have $p(x_0) = f_0, p(x_1) = f_1$ and $p(x_2) = f_2$.

In general for the data set $(x_0, f_0), \ldots, (x_n, f_n)$, define
$w(x) := (x - x_0) \cdots (x - x_n) \in \mathcal{P}_{n+1}$ and

$$\ell_j(x) := \prod_{i \neq j} \frac{(x - x_i)}{(x_j - x_i)} = \frac{w(x)}{(x - x_j)w'(x_j)}, \quad j = 0 : n,$$

where $w'(x)$ is the derivative of $w(x)$.

Lagrange interpolating polynomial

For $(x_0, f_0), (x_1, f_1), (x_2, f_2)$, define

$$p(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f_2.$$

Then we have $p(x_0) = f_0, p(x_1) = f_1$ and $p(x_2) = f_2$.

In general for the data set $(x_0, f_0), \ldots, (x_n, f_n)$, define
$w(x) := (x-x_0)\cdots(x-x_n) \in \mathcal{P}_{n+1}$ and

$$\ell_j(x) := \prod_{i \neq j} \frac{(x-x_i)}{(x_j-x_i)} = \frac{w(x)}{(x-x_j)w'(x_j)}, \quad j = 0:n,$$

where $w'(x)$ is the derivative of $w(x)$.

Note that $\ell_j(x_i) = \delta_{ij}$, where δ_{ij} is the Dirac delta function. Hence

$$p(x) := f_0\ell_0(x) + \cdots + f_n\ell_n(x)$$

interpolates the data set $(x_0, f_0), \ldots, (x_n, f_n)$. The basis $\ell_0(x), \ldots, \ell_n(x)$ is called the Lagrange basis of $\mathcal{P}_n$

Lagrange interpolating polynomial

For $(x_0, f_0), (x_1, f_1), (x_2, f_2)$, define

$$p(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f_2.$$

Then we have $p(x_0) = f_0, p(x_1) = f_1$ and $p(x_2) = f_2$.

In general for the data set $(x_0, f_0), \ldots, (x_n, f_n)$, define
$w(x) := (x - x_0) \cdots (x - x_n) \in \mathcal{P}_{n+1}$ and

$$\ell_j(x) := \prod_{i \neq j} \frac{(x - x_i)}{(x_j - x_i)} = \frac{w(x)}{(x - x_j)w'(x_j)}, \quad j = 0 : n,$$

where $w'(x)$ is the derivative of $w(x)$.

Note that $\ell_j(x_i) = \delta_{ij}$, where δ_{ij} is the Dirac delta function. Hence

$$p(x) := f_0\ell_0(x) + \cdots + f_n\ell_n(x)$$

interpolates the data set $(x_0, f_0), \ldots, (x_n, f_n)$. The basis $\ell_0(x), \ldots, \ell_n(x)$ is called the Lagrange basis of $\mathcal{P}_n$ and $p(x)$ is called the Lagrange interpolating polynomial.

Lagrange interpolating polynomial

Remark:

- Computation of $p(x)$ requires $\mathcal{O}(n^2)$ operations.
- Cannot accommodate new data (x_{n+1}, f_{n+1}). Any additional new data requires re-computation.
- Evaluation of $p(x)$ at a given x requires $\mathcal{O}(n^2)$ operations.

Example: For the data set $(-2,-27),(0,-1),(1,0)$, we have

$$\begin{aligned} p(x) &= -27\frac{(x-0)(x-1)}{(-2-0)(-2-1)} - 1\frac{(x+2)(x-1)}{(0+2)(0-1)} + 0\frac{(x+2)(x-0)}{(1+2)(1-0)} \\ &= -\frac{9}{2}x(x-1) + \frac{1}{2}(x+2)(x-1) = -1+5x-4x^2 \end{aligned}$$

Barycentric Lagrange interpolation

Recall $w(x) = \prod_{j=0}^{n}(x - x_j)$. Define $w_j = 1/w'(x_j) = 1/\prod_{i \neq j}(x_j - x_i)$ for $j = 0 : n$. Then

$$\ell_j(x) = \frac{w(x)}{(x - x_j)w'(x_j)} = w(x)\frac{w_j}{x - x_j}.$$

Hence the Lagrange interpolating polynomial can be rewritten as

$$p(x) = \frac{\sum_{j=0}^{n} \frac{w_j f_j}{x - x_j}}{\sum_{j=0}^{n} \frac{w_j}{x - x_j}}$$

which is called the Lagrange interpolating polynomial in barycentric form.

Barycentric Lagrange interpolation

Recall $w(x) = \prod_{j=0}^{n}(x - x_j)$. Define $w_j = 1/w'(x_j) = 1/\prod_{i \neq j}(x_j - x_i)$ for $j = 0 : n$. Then

$$\ell_j(x) = \frac{w(x)}{(x - x_j)w'(x_j)} = w(x)\frac{w_j}{x - x_j}.$$

Hence the Lagrange interpolating polynomial can be rewritten as

$$p(x) = \frac{\sum_{j=0}^{n} \frac{w_j f_j}{x - x_j}}{\sum_{j=0}^{n} \frac{w_j}{x - x_j}}$$

which is called the Lagrange interpolating polynomial in barycentric form.

Proof: Since $p(x) = \sum_{j=0}^{n} \ell_j(x) f_j = w(x) \sum_{j=0}^{n} \frac{w_j f_j}{x - x_j}$ and $1 = w(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j}$ (why?), diving the former by the latter yields

$$p(x) = \sum_{j=0}^{n} \frac{w_j f_j}{x - x_j} \Big/ \sum_{j=0}^{n} \frac{w_j}{x - x_j}.$$

Barycentric Lagrange interpolation

Advantages:

- Computation of w_j requires $\mathcal{O}(n^2)$ flops and is independent of f_j for $j = 0 : n$. This permits the interpolation of as many functions as desired in $\mathcal{O}(n)$ operations each once the weights w_j are known.

Barycentric Lagrange interpolation

Advantages:

- Computation of w_j requires $\mathcal{O}(n^2)$ flops and is independent of f_j for $j = 0 : n$. This permits the interpolation of as many functions as desired in $\mathcal{O}(n)$ operations each once the weights w_j are known.

- Evaluation of $p(x)$ at a given x requires only $\mathcal{O}(n)$ flops.

- Requires $\mathcal{O}(n)$ flops to incorporate a new data pair (x_{n+1}, f_{n+1}). Indeed, computation of $w_j/(x_j - x_{n+1}), j = 0 : n$, require $2(n+1)$ flops and computation of w_{n+1} requires $2n + 3$ flops.

Barycentric Lagrange interpolation

Advantages:

- Computation of w_j requires $\mathcal{O}(n^2)$ flops and is independent of f_j for $j = 0 : n$. This permits the interpolation of as many functions as desired in $\mathcal{O}(n)$ operations each once the weights w_j are known.

- Evaluation of $p(x)$ at a given x requires only $\mathcal{O}(n)$ flops.

- Requires $\mathcal{O}(n)$ flops to incorporate a new data pair (x_{n+1}, f_{n+1}). Indeed, computation of $w_j/(x_j - x_{n+1}), j = 0 : n$, require $2(n+1)$ flops and computation of w_{n+1} requires $2n + 3$ flops.

- Barycentric formula has a beautiful symmetry. The weights w_j appear in the denominator exactly as in the numerator, except without the data factors f_j. This means any common factor in all the weights w_j may be cancelled without affecting the value of $p(x)$.

***