Polynomial Interpolation

<u>Polynomial Interpolation</u>

Questions

i. What is the Algorithm?

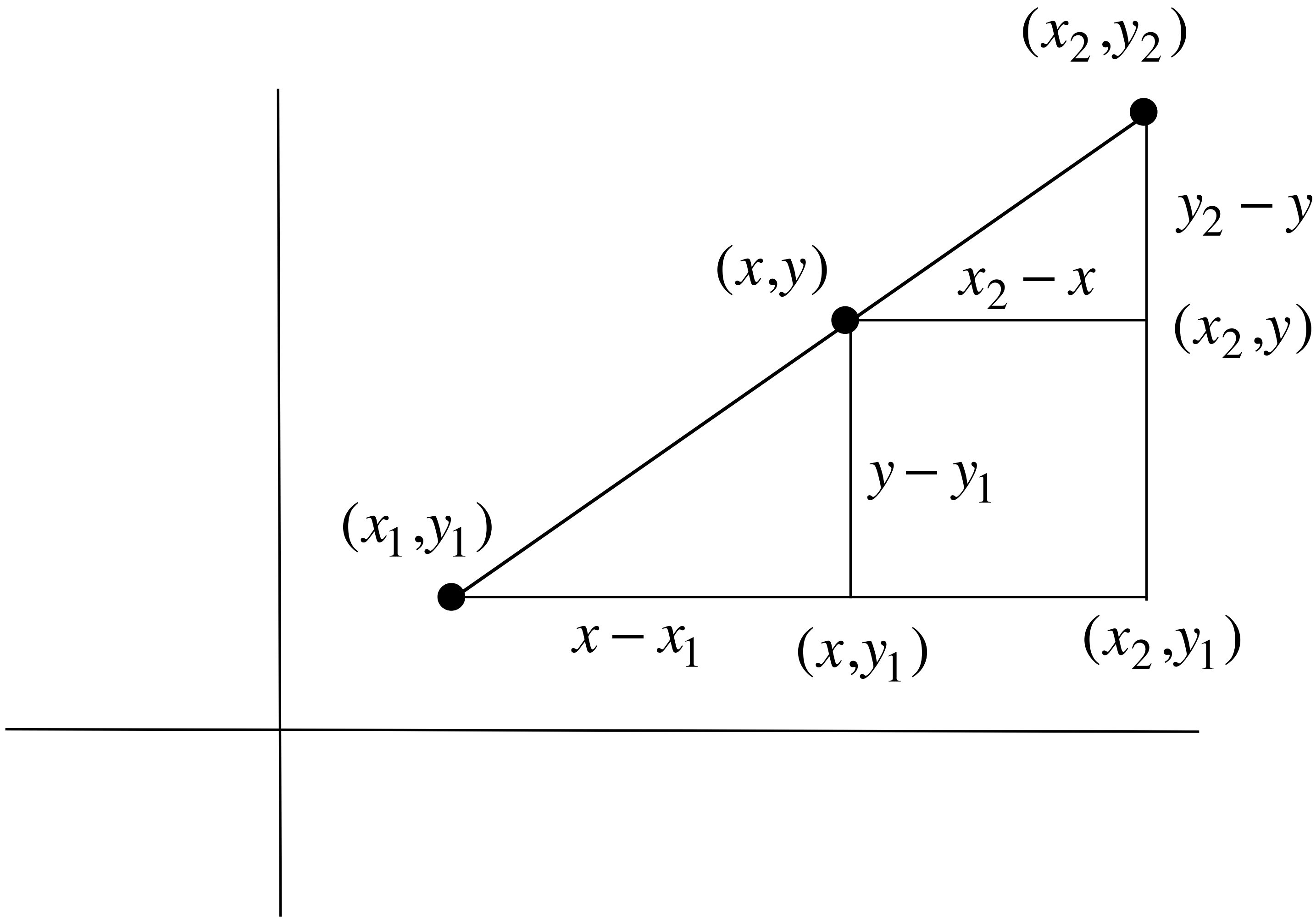
ii. Prove that the Algorithm Always Works.

iii. Prove that the Algorithm Generates Polynomials of Degree $\leq N$ for $N+1$ data points.

iv. How Fast is the Algorithm?

v. What is the Best Way to Program the Algorithm?

Linear Interpolation



Similar Triangles

Slope

$$\frac{y_2 - y}{x_2 - x} = \frac{y - y_1}{x - x_1}$$

$$(y_2 - y)(x - x_1) = (y - y_1)(x_2 - x)$$

$$y_2 x - y_2 x_1 - yx + yx_1 = yx_2 - yx - y_1 x_2 + y_1 x$$

$$(x - x_1)y_2 + yx_1 = yx_2 + (x - x_2)y_1$$

$$(x - x_1)y_2 + (x_2 - x)y_1 = y(x_2 - x_1)$$

$$\frac{x - x_1}{x_2 - x_1}y_2 + \frac{x_2 - x}{x_2 - x_1}y_1 = y$$

Neville's Algorithm

Linear Interpolation (Base Case)

$$I(x) = \frac{x_2 - x}{x_2 - x_1} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$$

- $I(x_1) = y_1$
- $I(x_2) = y_2$

Neville's Algorithm

Linear Interpolation (Base Case)

$$I_{D_2}(x) = \frac{x_2 - x}{x_2 - x_1} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$$

Recursion

$$I_{D_n}(x) = \frac{x_n - x}{x_n - x_1} I_{D_n^-}(x) + \frac{x - x_1}{x_n - x_1} I_{D_n^+}(x)$$

- $D_n \quad = (x_1, y_1), \ldots, (x_n, y_n)$
- $D_n^- = (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$
- $D_n^+ = (x_2, y_2), \ldots, (x_n, y_n)$

Polynomial Interpolation and Neville's Algorithm

Questions

i. Prove that Neville's Algorithm Always Works.

ii. Prove that Neville's Algorithm Generates Polynomials of Degree $\leq N$ for $N+1$ data points.

iii. How Fast is Neville's Algorithm?

iv. What is the Best Way to Program Neville's Algorithm?

Neville's Algorithm

Theorem 1: Neville's Algorithm interpolates the data:

$$I_{D_n}(x_k) = y_k \quad k = 1,\ldots,n.$$

Proof:

1. Base Case

$$I_{D_2}(x) = \frac{x_2 - x}{x_2 - x_1} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$$

$$I_{D_2}(x_1) = y_1 \quad \text{and} \quad I_{D_2}(x_2) = y_2$$

2. Inductive Step

$$I_{D_n}(x) = \frac{x_n - x}{x_n - x_1} I_{D_n^-}(x) + \frac{x - x_1}{x_n - x_1} I_{D_n^+}(x)$$

Assume: $I_{D_n}(x_k) = y_k \quad k = 1,\ldots,n.$ (Works for n data points.)

Must Show: $I_{D_{n+1}}(x_k) = y_k \quad k = 1,\ldots,n+1.$ (Works for $n+1$ data points.)

2. Inductive Step (continued)

By the inductive hypothesis

- $I_{D_{n+1}^{-}}(x_k) = y_k \qquad k = 1,\ldots,n \qquad \{n \text{ data points}\}$
- $I_{D_{n+1}^{+}}(x_k) = y_k \qquad k = 2,\ldots,n+1 \qquad \{n \text{ data points}\}$

Consider

- $I_{D_{n+1}}(x) = \dfrac{x_{n+1} - x}{x_{n+1} - x_1} I_{D_{n+1}^{-}}(x) + \dfrac{x - x_1}{x_{n+1} - x_1} I_{D_{n+1}^{+}}(x)$

-- $I_{D_{n+1}}(x_1) = I_{D_{n+1}^{-}}(x_1) = y_1$

-- $I_{D_{n+1}}(x_{n+1}) = I_{D_{n+1}^{+}}(x_{n+1}) = y_{n+1}$

-- $I_{D_{n+1}}(x_k) = \dfrac{x_{n+1} - x_k}{x_{n+1} - x_1} I_{D_{n+1}^{-}}(x_k) + \dfrac{x_k - x_1}{x_{n+1} - x_1} I_{D_{n+1}^{+}}(x_k)$

$$= \frac{x_{n+1} - x_k}{x_{n+1} - x_1} y_k + \frac{x_k - x_1}{x_{n+1} - x_1} y_k$$

$$= y_k \qquad\qquad k = 2,\ldots,n$$

<u>**Neville's Algorithm**</u>

Theorem 2: For n data points, Neville's Algorithm generates a polynomial $I_{D_n}(x)$, *and* $\deg\left(I_{D_n}(x)\right) \leq n-1$.

Proof:

1. Base Case

$$I_{D_2}(x) = \frac{x_2 - x}{x_2 - x_1} y_1 + \frac{x - x_1}{x_2 - x_1} y_2 \quad \text{is a polynomial of degree 1.}$$

2. Inductive Step

Assume: $I_{D_n}(x_k)$ is a polynomial of degree at most $n-1$.

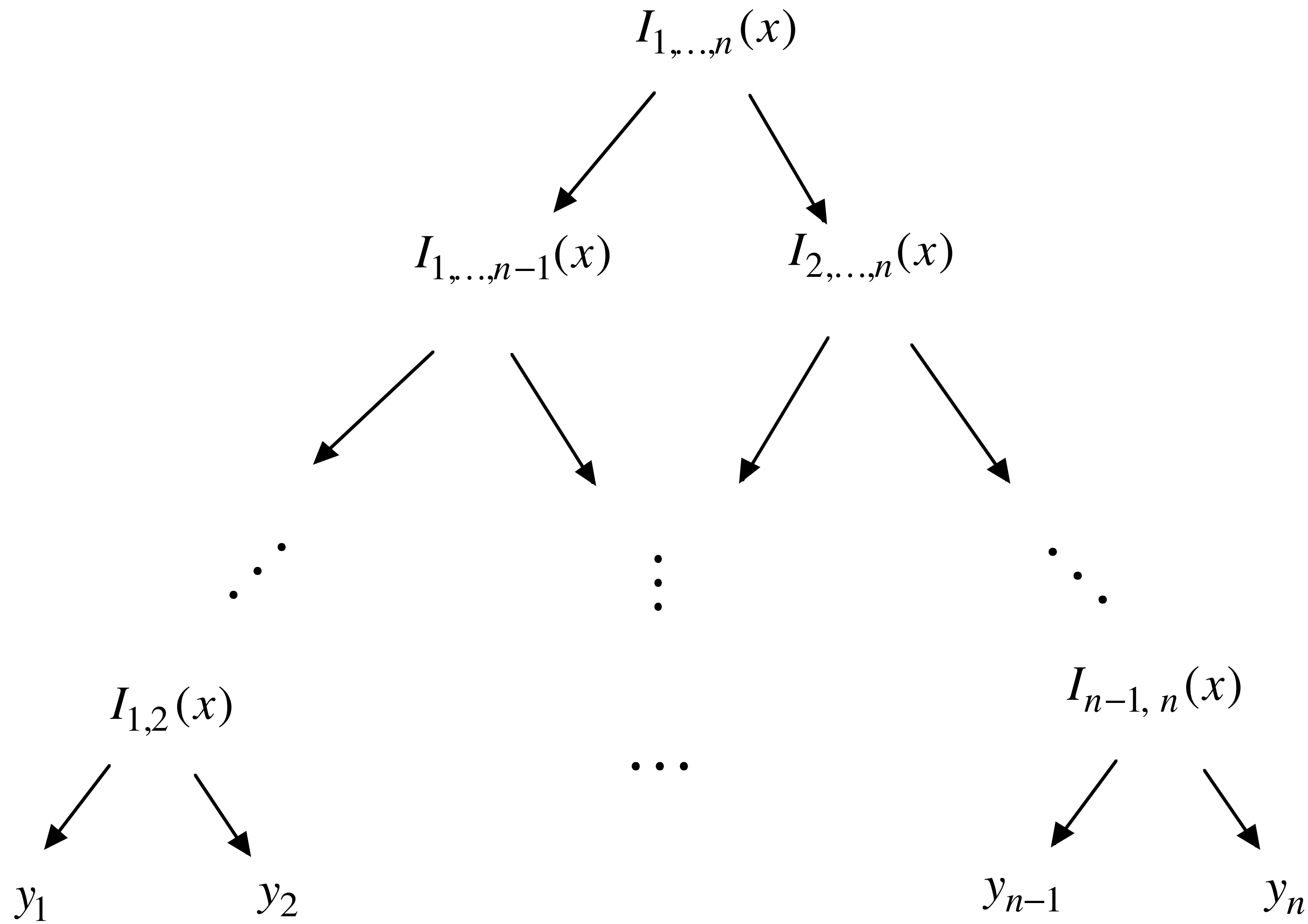
Must Show: $I_{D_{n+1}}(x)$ is a polynomial of degree at most n.

$$I_{D_{n+1}}(x) = \frac{x_{n+1} - x}{x_{n+1} - x_1} I_{D_{n+1}^-}(x) + \frac{x - x_1}{x_{n+1} - x_1} I_{D_{n+1}^+}(x)$$

But this result follows easily, since by the inductive hypothesis

$I_{D_n^-}(x)$, $I_{D_n^+}(x)$ are both polynomials of degree at most $n-1$.

Neville's Algorithm -- Recursive Implementation



$n-1$ *Levels* $\leftrightarrow$ $2^n - 2$ *multiplications*