Interpolation

To interpolate (fill-in gaps) means to construct a function that agrees with certain given data. 

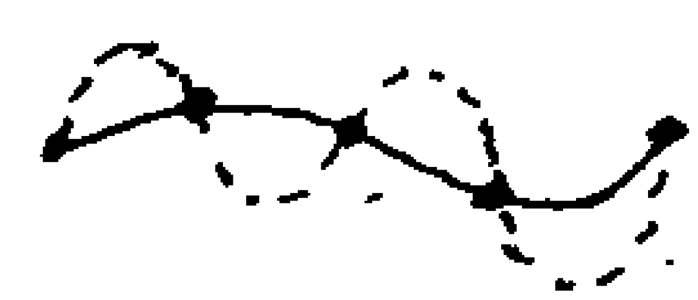
i.e. want to construct a curve (or surface) that passes thru given data pts.

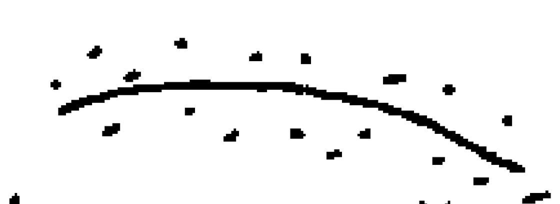
Primary uses:

1. Interpolation of table values: given a set of values $\{(x_i, y_i), i=1,\dots,n\}$ (of some quantity $y=f(x)$, unknown) want to construct a (model) function $F(x)$ that reproduces exactly the given data: $F(x_i) = y_i$, $i=1:n$

2. Representation of a (complicated) function $f(x)$ by a simpler $F(x)$

Should think of data points (x_i, y_i) as coming from a function f: $y_i = f(x_i)$ and want to interpolate f at $(x_i, f(x_i))$ by an interpolant F: $F(x_i) = f(x_i)$

Remarks: 1. There are infinitely many interpolants! 

2. Interpolation is useful only for error-free data. 
Data contaminated by errors (measurements, ...) should be approximated by "data fitting" (regression), to be discussed later...

3. Weirstrass Approximation Thm: Any $f \in \mathcal{C}[a,b]$ can be approximated uniformly by a polynomial:

$$\forall \varepsilon > 0 \ \exists \overset{\text{polynomial}}{p(x)} : \ |f(x) - p(x)| < \varepsilon \ \ \forall x \in [a,b], \text{ i.e. } \|f-p\|_\infty < \varepsilon$$

Choice of interpolants: by physical insight, by experience, recommenation, analogy, ...
and for simplicity ...

Usual approach: Choose a set of basis functions $\{ b_1(x), b_2(x), \ldots, b_m(x) \}$
and seek coefficients $c_1, \ldots, c_m$ for interpolant

$$F(x) = \sum_{j=1}^{m} c_j \, b_j(x) \quad : \quad F(x_i) = y_i \ , \ i = 1, 2, \ldots, n$$

i.e. $$\sum_{j=1}^{m} c_j \, b_j(x_i) = y_i \ , \ i = 1, \ldots, n$$ with data pts (x_i, y_i)

This is a linear $n \times m$ system: $B\vec{c} = \vec{y}$

$$B = [b_{ij}] = [\, b_j(x_i) \,] \ , \quad \vec{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} , \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Most common choices: polynomials, trig. polynomials, exponentials, rational functions

Osculatory interpolation: match values and values of derivatives when available

most common: Hermite interpolation (by cubic polynomials): $\underbrace{F(x_i) = y_i}_{\text{values}}$ and $\underbrace{F'(x_i) = d_i}_{\text{slopes}}$

Extreme case: Taylor polynomial for a function $f(x)$ at single node x_0
matches value of $f(x)$ and its derivatives at expansion point x_0.

<u>Polynomial interpolation problem</u>: Given $N+1$ data points (x_i, y_i), $i=0,\dots,N$

find polynomial $P_N(x)$ of degree $\le N$ passing through the points:

$$P_N(x_i) = y_i \quad , \quad i = 0, 1, \dots, N$$

<u>Thm</u>: If the node x_i are distinct, there is <u>unique</u> interpolating poly. $P_N(x)$

Note: ordering of data is irrelevant.

There are 3 methods to construct the unique interp. poly. $P_N(x)$:

<u>Method 1</u>: Solve the system $P_N(x_i) = y_i$ of $N+1$ equations in $N+1$ unknowns

$A\vec{c} = \vec{y}$ (basis: $\{1, x, x^2, \dots, x^N\}$)

Bad idea: for large N the coeff. matrix A is a Vandermonde matrix, ill-conditioned!

<u>Method 2</u>: <u>Lagrange form</u>: $P_N(x) = y_0 L_0(x) + \dots + y_N L_N(x)$

where $L_j(x) = \prod_{i \ne j} \frac{x - x_i}{x_j - x_i} = \frac{x-x_0}{x_j - x_0} \cdot \frac{x - x_1}{x_j - x_1} \cdots \frac{x - x_{j-1}}{x_j - x_{j-1}} \cdot \frac{x - x_{j+1}}{x_j - x_{j+1}} \cdots \frac{x - x_N}{x_j - x_N}$.

each $L_N(x)$ is a poly. of degree N (called Cardinal functions)

Note: $L_j(x_i) = \begin{cases} 0, & i \ne j \\ 1, & i = j \end{cases}$

Advantage: can be written down by formula.

Disadvantage: adding a new point changes all L_j, must be redone.

<u>Method 3</u>: <u>Newton form</u>: $P_N(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_N(x - x_0)\cdots(x - x_{N-1})$

$c_0 = y_0$, $c_1 = 1^{st}$ divided diff., $c_2 = 2^{nd}$ divided diff., ...

$$\begin{array}{lllll} x_0 & y_0 & & & \\ & & \frac{y_1 - y_0}{x_1 - x_0} = \Delta_1 & & \\ x_1 & y_1 & & & \frac{\Delta_2 - \Delta_1}{x_2 - x_0} \\ & & \frac{y_2 - y_1}{x_2 - x_1} = \Delta_2 & & \\ x_2 & y_2 & & & \\ \vdots & \vdots & \vdots & & \vdots \end{array}$$

Advantage: builts up degree by degree, adding another point, another term

Example: <u>Straight line interpolation</u>

Given 2 pts (x_0, y_0), (x_1, y_1), $x_0 \neq x_1$, find the line joining them.

<u>Method 1</u>: (Solve the system) Find c_0, c_1 : $\boxed{P_1(x) = c_0 + c_1 x}$ passes thru the points:

$$\begin{cases} c_0 + c_1 \cdot x_0 \overset{\text{want}}{=} y_0 \\ c_0 + c_1 \cdot x_1 = y_1 \end{cases} \qquad \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}. \ \det A = x_1 - x_0 \neq 0 \therefore \text{unique sol}$$

$$\Rightarrow \quad P_1(x) = \frac{y_0 x_1 - y_1 x_0}{x_1 - x_0} + \frac{y_1 - y_0}{x_1 - x_0} \cdot x \quad \text{: line thru 2 points}$$

Here we used as basis $\{1, x\}$ for the vector space of polys of deg ≤ 1, of dim 2.
Any other basis for this vector space may be used, e.g. $\{1, x - x_0\}$:

$\boxed{\tilde{P}_1(x) = c_0 + c_1(x - x_0)}$ at x_0: $c_0 \overset{\text{want}}{=} y_0$, at x_1: $c_0 + c_1(x_1 - x_0) = y_1 \Rightarrow c_1 = \frac{y_1 - y_0}{x_1 - x_0}$

$\Rightarrow \tilde{P}_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$, the point-slope formula for line!

Note that a wiser choice of basis made solution of system trivial!

<u>Method 2</u>: <u>Lagrange form</u> of interpolant: Lagrange basis: $\left\{ b_0(x) = \frac{x - x_1}{x_0 - x_1},\ b_1(x) = \frac{x - x_0}{x_1 - x_0} \right\}$

Are they linearly independent (functions)? $\qquad b_j(x_i) = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$

Does $c_0 b_0(x) + c_1 b_1(x) = 0 \ \forall x$ imply $c_0 = c_1 = 0$?

For $x = x_0$: $c_0 \cdot 1 + c_1 \cdot 0 = 0 \Rightarrow c_0 = 0$
For $x = x_1$: $c_0 \cdot 0 + c_1 \cdot 1 = 0 \Rightarrow c_1 = 0$ yes! $\therefore$ lin. independent

Lagrange 1^{st} degree interpolating polynomial: $\boxed{P_1(x) = c_0 \cdot b_0(x) + c_1 \cdot b_1(x)}$

want $P_1(x_i) = y_i$, $i = 0, 1$ $\Rightarrow$ $P_1(x_0) = c_0 \cdot 1 + c_1 = y_0 \Rightarrow c_0 = y_0$
$P_1(x_1) = c_0 \cdot 0 + c_1 \cdot 1 = y_1 \Rightarrow c_1 = y_1$

$$\Rightarrow P_1(x) = y_0 \cdot b_0(x) + y_1 \cdot b_1(x) \equiv y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0} = \ldots \equiv \text{line above}$$

<u>Remark</u>: Choice of basis is <u>very</u> important! can make the problem easy, as above, or can make an impossible to do problem doable!

Finding a good basis arises all over, we'll see it again in quadrature, Least Squares and it is crucial in Machine Learning with neural nets!

<u>Method 3</u>: <u>Newton form</u> of interpolant: basis $\{1, x - x_0\}$

$P_1(x) = c_0 + c_1 \cdot (x - x_0)$: want $c_0 + 0 = y_0 \Rightarrow c_0 = y_0$
$c_0 + c_1 \cdot (x_1 - x_0) = y_1 \Rightarrow c_1 = \frac{y_1 - y_0}{x_1 - x_0}$

$\Rightarrow P_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} \cdot (x - x_0)$ the usual point-slope formula

$= \ldots =$ same line as in Method 1, 2