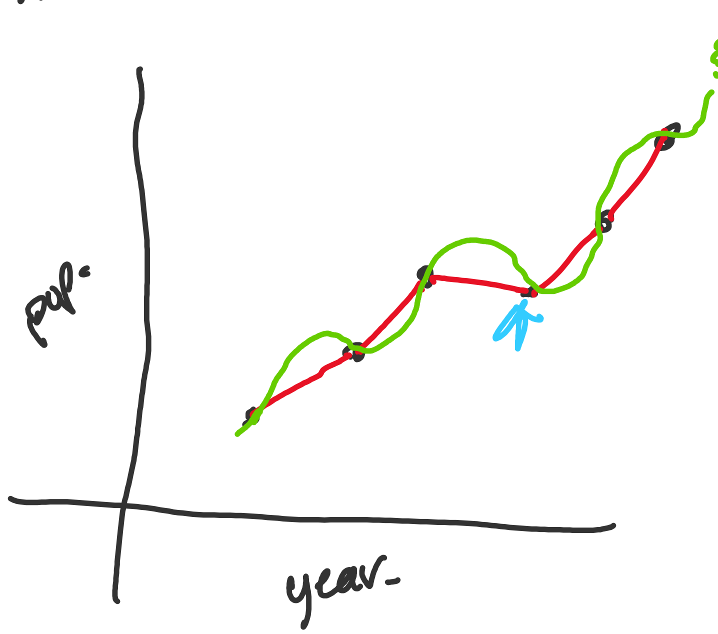
Section 3.3: Cubic Splines

Tuesday, February 16, 2021 8:37 AM

Previously (§3.1, 3.2) we discussed using a single polynomial to approximate our functn.

However, high degree polynoms can oscillate.

Instead, we can break into separate intervals each with its own low-degree approximating polynomial.



easiest idea: use a linear function between each data point.

"piecewise linear interpolation"

Q: What disadvantage does piecewise linear interp. have (esp @ nodes)

A: not differentiable at datapoints (ie, not "smooth")

Instead we consider piecewise polynomials

In particular cubic splines

How to find Eqns for cubic spline

Given a fn f on $[a,b]$ and nodes $x_0, x_1, \ldots, x_n$, a cubic spline interpolant $S(x)$ for f is a function that satisfies the following conditions:

"On each interval we have a unique cubic polynomial"

(a) $S_j(x)$ on subinterval $[x_j, x_{j+1}]$ is a cubic polynomial

$$S_j(x) = a_j + b_j(x-x_j) + c_j(x-x_j)^2 + d_j(x-x_j)^3$$

$j = 0, \ldots, n-1$

"polynoms must match data points"

(b) $S_j(x_j) = S_{j+1}(x_j) = f(x_j)$ for $j = 0, \ldots, n-1$

"continuity at datapoints"

(c) $S_j(x_j) = f(x_j)$, $S_j(x_{j+1}) = f(x_{j+1})$

"continuity of derivative at datapoints"

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ $j = 0, \ldots, n-2$

"continuity of 2nd derivative at datapoints"

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$, $j = 0, \ldots, n-2$

(f) One of the following boundary conditions is satisfied

$\boxed{S''(x_0) = S''(x_n) = 0 \quad \text{"free" "natural"}}$

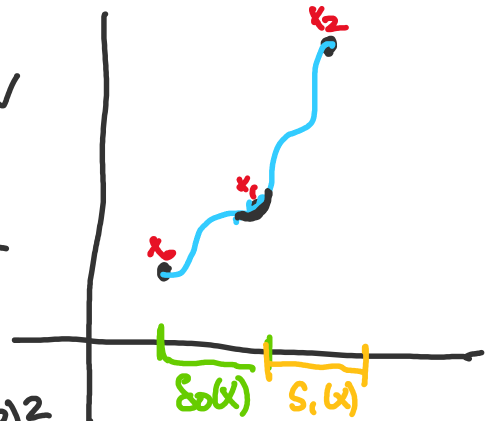
$S'(x_0) = f'(x_0)$, $S'(x_n) = f'(x_n)$ "clamped"

$\boxed{\text{"not a knot"}}$

ex Construct a natural cubic spline interpolant that passes thru $(1,2), (2,3), (3,5)$

$x = [1.2, 1.5]\ [2.6]$ $S_0(x)$ $S_1(x)$

lets follow conditions above



(a) $S_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3$ x_0

$S_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3$ x_1

⇒ we need to find 8 coefficients

(b) & (c) endpoints must match & cont @ datapoints

$S_0(x_0) = \boxed{2 = a_0}$ "plugged x_0 into $S_0(x)$"

$S_0(x_1) = 3 = a_0 + b_0(x_1-1) + c_0(x_1-1)^2 + d_0(x_1-1)^3$

$\boxed{3 = a_0 + b_0 + c_0 + d_0}$

$S_1(x_1) = \boxed{3 = a_1}$

$S_1(x_2) = 5 = a_1 + b_1(x_2-2) + c_1(x_2-2)^2 + d_1(x_2-2)^3$

$\boxed{5 = a_1 + b_1 + c_1 + d_1}$

(d) continuity of first derivative at datapoints

$S_0'(x_1) = S_1'(x_1) \Rightarrow S_0'(2) = S_1'(2)$

$S_0'(x) = b_0 + 2c_0(x-1) + 3d_0(x-1)^2$, $S_0'(2) = b_0 + 2c_0 + 3d_0$

$S_1'(x) = b_1 + 2c_1(x-2) + 3d_1(x-2)^2$, $S_1'(2) = b_1$

$\boxed{b_1 = b_0 + 2c_0 + 3d_0}$

(e) continuity of 2nd deriv at data points

$S_0''(x_1) = S_1''(x_1)$ $x_1 = 2.$

$S_0''(x) = 2c_0 + 6d_0(x-1)$, $S_0''(2) = 2c_0 + 6d_0$

$S_1''(x) = 2c_1 + 6d_1(x-2)$, $S_1''(2) = 2c_1$

$$\boxed{2c_1 = 2c_0 + 6d_0}$$

(f) Natural BC

$S_0''(x_0) = 0$ $S_0''(1) = 0 = 2c_0 \Rightarrow \boxed{c_0 = 0}$

$S_1''(x_2) = 0$ $S_1''(3) = \boxed{0 = 2c_1 + 6d_1}$

We can tediously solve all equations to

$a_0 = 2,\ b_0 = 3/4,\ c_0 = 0,\ d_0 = 1/4$

$a_1 = 3,\ b_1 = 3/2,\ c_1 = 3/4,\ d_1 = -1/4$

then our spline

$$S(x) = \begin{cases} S_0(x) = 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3, & x \in [1,2] \\ S_1(x) = 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3, & x \in [2,3] \end{cases}$$

Cubic Spline Pseudocode

vectors w/ coefficients $a_0, a_1, \ldots, b_0, b_1, \ldots$

inputs: a, b, c, d, x, datx, daty

x ← vector to interpolate over

output: y ← S evaluated at x vector.

N = length data points

for j = 1 : N-1

find x values between datx(j) & datx(j+1)

add to spline : a(j) + b(j)(x - datx(j))
+ c(j)(x - datx(j))^2 + d(j)(x - datx(j))^3

Hint: make sure spline @ x_0, x_n

end % for

plot(datx, daty, x, spline)

does my spline go thru datapoints?

Debug tip.