Monotone Cubic Interpolation

Aresh Pourkavoos

July 11, 2022

Given a set of data points $(x_1, y_1), \ldots, (x_n, y_n)$ in which both coordinates strictly increase moving down the list (i.e. $x_1 < \ldots < x_n$ and $y_1 < \ldots < y_n$), how can we define a function which passes through all points and which has positive slope everywhere? In other words, we want to find $f : [x_1, x_n] \to \mathbb{R}$ such that $f(x_i) = y_i$ for all $i \in \{1, \ldots, n\}$ and $f'(x) > 0$ for all $x \in [x_1, x_n]$.

Drawing line segments between adjacent points almost works, but the derivative is generally undefined at each point, since the slopes of the segments on either side may be different. The apparent solution would be to use cubic interpolation instead, which allows for the derivative at each point to be set, and cubic curves between each pair of adjacent points may be found.

(Hermite polynomials)

This raises the question of what the derivatives should be.

A common approach to cubic interpolation is, for every point, to connect the points on its left and right and use the slope of the resulting segment. If the point is on the far left or right, the slope of the segment between it and its adjacent point is used. However, this approach does not guarantee monotonicity for monotone inputs.

(Example)

On the other hand, the derivative could be 0 at every data point, which guarantees that the function is increasing. However, this solution is not ideal because the derivative should be positive everywhere. It is possible to choose a very small positive slope at each point, which still avoids the decreasing segments seen previously. But this does not produce a straight line if the points are collinear, which an ideal solution would.

Instead of checking different formulas for the function, we can think about finding it as an optimization problem. In other words, given a function that passes through the points, we can define a way to evaluate how "well" it interpolates between them, and select the function which perform the best. A common approach for these types of problems is to define the "energy" for a given curve, which usually measures deviation from a straight line. The name comes from the fact that an elastic rod stores mechanical potential energy when bent and tries to minimize its energy by straightening when released.

One of the ways to define a straight line is that its derivative is constant, i.e. its second derivative is 0 everywhere. Thus, we might try to keep the second derivative of our function as close to 0 as possible, to prevent it from changing slope too quickly. A natural way to measure distance from 0 is to square the second derivative, disincentivizing both negative and positive values. Since every point along the graph of the function should be taken into account, we can integrate the squared second derivative over the entire interval:

$$J(f) = \int_{x_1}^{x_n} f''(x)^2 dx.$$

This is a problem in variational calculus, which is like multivariable calculus except that rather than functions whose inputs are vectors, variational calculus is concerned with functionals, i.e. functions whose inputs are themselves functions. J defined above is a functional, since it accepts a function f and returns its energy. Here, we will restrict f to be C^1, or continuously differentiable. This means that f must have a derivative on the open interval (x_1, x_n) (as well as one-sided derivatives at the endpoints x_1 and x_n), and f' must itself be a continuous function.

The set of such functions that passes through an arbitrary set of points $(x_1, y_1), \ldots, (x_n, y_n)$ is difficult to parameterize. Instead, we take advantage of the definition of C^1 to break the problem into more manageable

parts. Suppose we already know $f'(x_1), \ldots, f'(x_n)$, i.e. the derivative of the function at each data point. Call these values $c_1, \ldots, c_n$. Then f may be split into pieces $f_1, \ldots, f_{n-1}$, where the domain of f_i is the interval between adjacent data points $[x_i, x_{i+1}]$ for all $i \in \{1, \ldots, n-1\}$. It follows that each f_i is also C^1 and

$$\begin{aligned} f_i(x_i) &= y_i, \\ f_i(x_{i+1}) &= y_{i+1}, \\ f_i'(x_i) &= c_i, \\ f_i'(x_{i+1}) &= c_{i+1}, \end{aligned}$$

where the endpoints have one-sided derivatives. The important thing, though, is that once the derivatives are set, the different possible curves of each piece f_i may be swapped out independently of one another. This is because the derivatives of adjacent pieces will always meet up at the endpoints, preserving the continuity of f' across the entire domain $[x_1, x_n]$. This means that whatever the derivatives c_i happen to be, f must necessarily be made of segments which are individually optimal for those c_i. If f had a suboptimal segment in it, then that segment could be replaced with one with a lower value of J, bringing down the total value for f. Thus f can be optimized as follows:

- Let the derivatives $c_1, \ldots, c_n$ be arbitrary.
- For each interval $[x_i, x_{i+1}]$, find the optimal C^1 function f_i with endpoint values y_i and y_{i+1} and endpoint derivatives c_i and c_{i+1}.
- Assemble f using these f_i and calculate $J(f)$ in terms of the various c_i.
- Since J now depends only on a handful of values c_i, it may be minimized using multivariable calculus.