Natural B-Spline Construction and Evaluation on Irregular Grids

February 20, 2017

Primitives

We have a grid with N elements $g = \{g_i\}$ where $g_1 < g_2 < ... < g_{N-1} < g_N$. Also we know the values of $f(g_i), \forall i \in 1...N$. We are going to use $M - th$ order B-Spline to interpolate f.

Construction

Using De Boor's algorithm, we need a starting grid with dimention $N + 2M$. A "natural" way of constructing this auxillary grid is $g^{\text{aux}} = \left\{ \underbrace{g_1 ... g_1}_{M+1}, g_2, g_3, ..., g_{N-2}, g_{N-1}, \underbrace{g_N ... g_N}_{M+1} \right\}$. (The boundary condition it indicates is a bunch of higher order derivatives at both ends to be 0. To be specific, if $M = 3$ it suggests $f''(g_1) = f''(g_N) = 0$.) De Boor's algorithm reads

$$N_i^0(g) = \begin{cases} 1 & g \in \left(g_i^{\text{aux}}, g_{i+1}^{\text{aux}}\right) \\ 0 & \text{else} \end{cases}$$

and

$$N_i^n(g) = \frac{g - g_i^{\text{aux}}}{g_{i+n}^{\text{aux}} - g_i^{\text{aux}}} N_i^{n-1}(g) + \frac{g_{i+n+1}^{\text{aux}} - g}{g_{i+n+1}^{\text{aux}} - g_{i+1}^{\text{aux}}} N_{i+1}^{n-1}(g)$$

then in a $M - th$ order B-Spline approximation for any g,

$$f(g) = \sum_{i=1}^{N} \theta_i N_i^M(g)$$

Observations:

1. $N_i^M(g)$ is a $M - th$ order polynomial of g. For example, linear spline is a linear function of g.

2. $\forall g \in [g_1, g_N]$, there will be at most $M + 1$ is such that $N_i^M(g)$ is nonzero. For example, with a linear spline interpolation, for each point of g there will be at most two points that is non-trivially related to the value of $f(g)$.

3. Although g is a continuous variable, $N_i^M(g)$ only differs from interval to interval. For $a, b \in [g_i, g_{i+1}]$, the polynomial coefficient of $N_i^M(a)$ is exactly the same as $N_i^M(b)$. For example, to linearly interpolate any variable $x \in [1, 2]$, we always know that $f(x) = f(1)\frac{2-x}{2-1} + f(2)\frac{x-1}{2-1}$ no matter what value of x is.

Therefore, I construct and store all possible coefficients in a $N + 1 \times M + 1 \times M + 1$ matrix - we only have $N + 1$ intervals (think about extrapolation as two intervals). Within each interval, at most $M + 1$ spline objects are non-zero. For each non-zero spline object, it is a $M - th$ order polynomial so we only need to store $M + 1$ numbers.

(For sure, we can construct Jacobian similarly)

Evaluation

After we store all the coefficients, for a given g to interpolate,

1. Calculate $1, g, g^2, g^3, ..., g^M$.

2. (Binary) search which interval does g lie in.

3. Evaluate all $M+1$ spline objects, each of them is a $M-th$ order polynomial. After we do step 1, step 3 is just a matrix multiplication. And we get all $N_i^M(g)$ we need.

4. $f(g) = \sum_{i=1}^{N} \theta_i N_i^M(g)$. As I said in step 3, there are at most $M+1$ spline objects. Therefore there are only $M+1$ non-zero items here.

Time and Memory Complexity

Using float number basic calculation as measure. $(+, *)$
Construction: TC: $O\left(NM^2\right)$, MC: $O\left(NM^2\right)$,
Evaluation K points: TC: $O\left(KM^2 \log N\right)$, MC: $O\left(KM^2\right)$.