Interpolation - Extrapolation

Numerical Methods for IT

Problem

For set of points $\mathrm{T} = \{(x_1, y_1), \dots, (x_N, y_N)\}, x_1 < \cdots < x_N \in [x_1, x_N]$.
Question 1: given $x \in [x_1, x_N]$. Find the y value corresponding to $x : y = y(x) = f(x) = ?$.
Question 2: given $x \notin [x_1, x_N]$. Find the y value corresponding to $x : y = y(x) = f(x) = ?$.

It is highly desirable that an interpolation or extrapolation routine should provide an estimate of its own error.

Conceptually, the interpolation process has 2 stage:
(**1**) Fit (once) an interpolating function to the data point provided.
(**2**) Evaluate that interpolating function at a target point x.

However, this 2-stage method is usually not the best way to proceed in practice. We can modify to another 2-stage method:
(**1**) Find the right staring position in the table T: x_1 or i.
(**2**) Perform the interpolating using M nearby values.

In the new 2-stage method, we can use the binary-search routine.

Polynomial Interpolation and Extrapolation

Lagrange's idea: through any 2 point there is a unique line. Through any 3 points there is a unique a quadratic. …
Interpolating polynomial of degree $M-1$ throught M points $y_1 = f(x_1), \ldots, y_M = f(x_M)$ is given explicitly by Lagrange's formular:

$$P(x) = \frac{(x-x_2)(x-x_3)\ldots(x-x_M)}{(x_1-x_2)(x_1-x_3)\ldots(x_1-x_M)} y_1 + \cdots + \frac{(x-x_1)(x-x_2)\ldots(x-x_{M-1})}{(x_M-x_1)(x_M-x_2)\ldots(x_M-x_{M-1})} y_M.$$

The resulting algorithm gives no error estimate, and it is also somewhat awkward to program. A much better algorithm is Nevill's algorithm.
Nevill's algorithm closely related to and sometimes confused with Aitken algorithm.

Algorithm (Nevill)

$$P_{i(i+1)\ldots(i+m)} = \begin{cases} P_i, & i = 1, \ldots, M \\ \frac{(x-x_{i+m})P_{i(i+1)\ldots(i+m-1)} + (x_i - x)P_{(i+1)(i+2)\ldots(i+m)}}{x_i - x_{i+m}}, & else \end{cases}.$$

Let $C_{m,i} \equiv P_{i\ldots(i+m)} - P_{i\ldots(i+m-1)}$; $D_{m,i} \equiv P_{i\ldots(i+m)} - P_{(i+1)\ldots(i+m)}$ then one can derive the above formular the relation: $D_{m+1,i} = \frac{(x_{i+m+1}-x)(C_{m,i+1} - D_{m,i})}{x_i - x_{i+m+1}}$; $C_{m+1,i} = \frac{(x_i - x)(C_{m,i+1} - D_{m,i})}{x_i - x_{i+m+1}}$.

Cubic spline interpolation

Given a tabulated function $y_i = y(x_i), i = 1, \dots, N$.

Focus on one particularly interval, between $x_i\ and\ x_{i+1}$, and let $y = ay_j + by_{j+1}$ (*),

where $a = \frac{(x_{j+1} - x)}{x_{j+1} - x_j}, b = \frac{x - x_j}{x_{j+1} - x_j}$.

Since it is (piecewise) linear, it has zero second derivative in the interior of each interval and an undefined, or infinite, second derivative at the abscissas x_j.

The goal of cubic spline interpolation is to get an interpolation formular that is smooth in the first derivative and continuous in the second derivative, both within an interval and its boundaries.

Suppose that there is also tabulated values of $y_i'', i = 1, \dots, N$. Then we can rewrite (*):

$y = ay_j + by_{j+1} + cy_j'' + dy_{j+1}''$ (**), where

$c = \frac{1}{6}(a^3 - a)\left(x_{j+1} - x_j\right)^2; d = \frac{1}{6}(b^3 - b)\left(x_{j+1} - x_j\right)^2$.

Take derivative (**) with respect to x: $\frac{dy}{dx} = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{3a^2 - 1}{6}\left(x_{j+1} - x_j\right)y_j'' + \frac{3b^2 - 1}{6}\left(x_{j+1} - x_j\right)y_{j+1}''$

(***) and $\frac{d^2y}{dx^2} = ay_j'' + by_{j+1}''$. And $\frac{x_j - x_{j-1}}{6}y_{j-1}'' + \frac{x_{j+1} - x_{j-1}}{3}y_j'' + \frac{x_{j+1} - x_j}{6}y_{j+1}'' = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}$.

For a unique solution, we need:

. Set one or both $y_1'' = 0, y_N'' = 0$ (natural cubic spline),

. Set either $y_1''\ and\ y_N''$ to values calculate from (***).

Rational function interpolation/extrapolation

Some functions not well approximated by polynomials but are well approximated by rational functions, that is quotients of polynomials.

Let $R_{i(i+1)\dots(i+m)}$: a rational function passing to (m+1) point $(x_{i+1}, y_{i+1}), \dots, (x_{i+m}, y_{i+m})$.

Suppose:$R_{i(i+1)\dots(i+m)} = \frac{P_\mu(x)}{Q_v(x)} = \frac{p_0+p_1x+\cdots+p_\mu x^\mu}{q_0+q_1x+\cdots+q_v x^v}$ with $m+1 = \mu + v + 1$.

The algorithm is summarized by the recursive relation:

$$R_{i(i+1)\dots(i+m)} = R_{(i+1)\dots(i+m)} + \frac{R_{(i+1)\dots(i+m)} - R_{i\dots(i+m-1)}}{\left(\frac{x-x_i}{x-x_{i+m}}\right)\left(1 - \frac{R_{(i+1)\dots(i+m)} - R_{i\dots(i+m-1)}}{R_{(i+1)\dots(i+m)} - R_{(i+1)\dots(i+m-1)}}\right) - 1}.$$

This recursive generates the rational functions through $(m+1)$ points. It is started with: $R_i = y_i$
And with $R = [R_{i(i+1)\dots(i+m)}\ with\ m = -1] = 0$.

We can convert to one involving only small differences:
$C_{m,i} = R_{i\dots(i+m)} - R_{i\dots(i+m-1)}$; $D_{m,i} = R_{i\dots(i+m)} - R_{(i+1)\dots(i+m)}$
satisfy the relation $C_{m+1,i} - D_{m+1,i} = C_{m,i+1} - D_{m,i}$.

$$D_{m+1,i} = \frac{C_{m,i+1}\left(C_{m,i+1} - D_{m,i}\right)}{\left(\frac{x-x_i}{x-x_{i+m-1}}\right) D_{m,i} - C_{m,i+1}} \ ; C_{m+1,i} = \frac{\left(\frac{x-x_i}{x-x_{i+m-1}}\right) D_{m,i}\left(C_{m,i+1} - D_{m,i}\right)}{\left(\frac{x-x_i}{x-x_{i+m-1}}\right) D_{m,i} - C_{m,i+1}}$$

Interpolation on data in multidimensions

Given a scattered set of N data points $(\boldsymbol{x}_i, y_i), \in \mathbb{R}^{n+1}, i = 1, \dots, N$.

Radial Basis Function Interpolation – RBF Interpolation.
Imagine that every known point j "influences" tis surroundings the same way in all directions, according, according to some assumed functional form $\phi(r)$ – the radical basis function – that is a function only of radial distance $r = \left|\boldsymbol{x} - \boldsymbol{x}_j\right|$.
Let try to approximate the interpolating function by a linear combination of ϕ's, centered on all know points: $y(\boldsymbol{x}) = \Sigma_{i=1}^{N} w_i \phi(|x - x_i|)$, where w_i unknowns.
Solve a set of N linear equations: $y_i = \Sigma_{i=1}^{N} w_i \phi(|x - x_i|)$

Normalized Radial Basis Function Interpolation – RBF Interpolation.
We require the sum of the basis functions to be unity, or
$y(\boldsymbol{x}) = \frac{\Sigma_{i=1}^{N} w_i \phi(|x - x_i|)}{\Sigma_{i=1}^{N} \phi(|x - x_i|)}$ and $y_j \Sigma_{i=1}^{N} \phi\left(\left|x_j - x_i\right|\right) = \Sigma_{i=1}^{N} w_i \phi\left(\left|x_j - x_i\right|\right)$.

Interpolation by Kriging

Kriging method is also called Gauss-Markov estimation or Gauss process regression.

Let $v(\boldsymbol{x}) \sim \frac{1}{2}\langle[y(\boldsymbol{x}+\boldsymbol{r}) - y(\boldsymbol{x})]^2\rangle$, called variogram – an estimation of the mean square variation, where the average is over all $\boldsymbol{x}$ with fixed $\boldsymbol{r}$. We usually takes $v(\boldsymbol{r})$ to be a function only of the magnitude $r = |\boldsymbol{r}|$, write $v(r)$.

Let v_{ij} denote $v\left(\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|\right)$ and v_{*j} denote $v\left(\left|x_* - x_j\right|\right)$, where $\boldsymbol{x}_*$ being a point at which we want an interpolated value $y(\boldsymbol{x}_*)$.
Let $\boldsymbol{Y} = (y_1, \dots, y_N, 0)$; $\boldsymbol{V}_* = (v_{*1}, \dots, v_{*N}, 1)$ and

$$\boldsymbol{V} = \begin{pmatrix} v_{11} & \cdots & v_{1,N} \\ \vdots & \ddots & \vdots \\ v_{N,1} & \cdots & v_{NN} \\ 1 & \cdots & 1 \end{pmatrix},$$

Then the Kriging interpolation estimate $\hat{y}_* = \boldsymbol{V}_* \cdot \boldsymbol{V}^{-\mathbf{1}} \cdot \boldsymbol{Y}$ and
its variance is given by $\mathrm{V}ar(\hat{y}_*) = \boldsymbol{V}_* \cdot \boldsymbol{V}^{-\mathbf{1}} \cdot \boldsymbol{V}_*$

Laplace Interpolation

We look at a missing data, namely, how to restore missing or unmeasured values on a grid regular. One good method is Laplace interpolation or Laplace/Poisson interpolation

The general idea is to find an interpolation function y that satisfies Laplace's equation in n-dimension: $\nabla^2 y = 0$ whenever there is no data, and which satisfies $y(\boldsymbol{x}_i) = y_i$ at all measured data points.

For purpose illustration, we specialize to the case of 2-dimension, in the case of a Cartesian grid whose x_1, x_2 values evenly spaced.

If y_0 denotes the value at the free point, while y_u, y_d, y_l, r_r denote the values at neighbors, then $y_0 - \frac{1}{4}y_u - \frac{1}{4}y_d - \frac{1}{4}y_l - \frac{1}{4}y_r = 0.$

Homogeneous choices that embody natura boundary conditions are:

$y_0 - \frac{1}{2}y_u - \frac{1}{2}y_d = 0\ (left, right\ boundaries)$ $y_0 - \frac{1}{2}y_l - \frac{1}{2}y_d = 0\ (top - right\ corner)$

$y_0 - \frac{1}{2}y_l - \frac{1}{2}y_r = 0\ (top, bottom\ boundaries)$ $y_0 - \frac{1}{2}y_r - \frac{1}{2}y_u\ (bottom - left\ corner)$

$y_0 - \frac{1}{2}y_r - \frac{1}{2}y_d = 0\ (top - left\ corner)$ $y_0 - \frac{1}{2}y_l - \frac{1}{2}y_u = 0\ (bottom - right\ cornoer)$