Data interpolation

Eugeniy E. Mikhailov

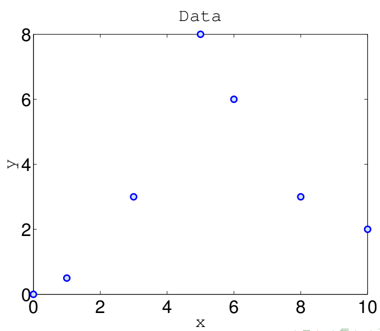
The College of William & Mary

Lecture 22

Notes

Data interpolation - filling the voids

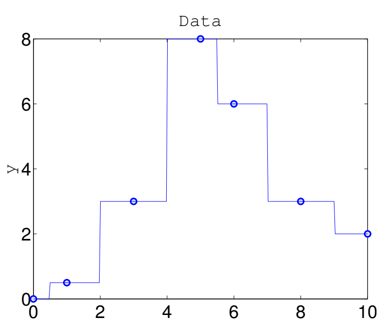
Very rarely there is enough data. Often taking a data point takes a lot of time or it is expensive. But we would like to have some presentation of the system in the voids.



Notes

Nearest neighbor interpolation

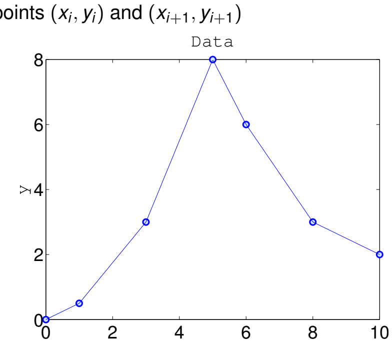
The name says it all. For each interpolated point $x_{interpolated}$ find its nearest neighbor along the x_i axis in the data set and use its y_i value.



Notes

Linear interpolation

We will split our data set with N points to $N-1$ intervals and interpolate the values in the given interval as a line passing through the border points (x_i, y_i) and (x_{i+1}, y_{i+1})



Notes

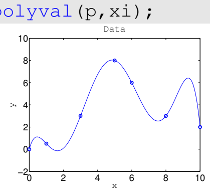
Polynomial fit

You can always find a polynomial of $N-1$ degree passing through N data points.

$$y(x) = p_1 x^n + p_2 x^{n-1} + \dots + p_n x + p_{n+1}$$

Matlab has the 'polyfit' function which returns the polynomial coefficient.

```
% calculate coefficients
p=polyfit(xdata, ydata, (length(xdata)-1) );
%  interpolate
yi=polyval(p,xi);
```



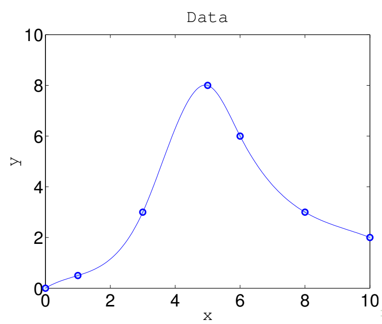
Interpolated values tend to oscillate for the polinomial of a high degree.
Do not use it!
Unless you know what you are doing.

Notes

Cubic spline interpolation

We will interpolate N data points by a polynomial of 3rd degree for each i_{th} interval between data point

$$f_i(x) = p_{1_i} x^3 + p_{2_i} x^2 + p_{3_i} x + p_{4_i}, x \in [x_i, x_{i+1}]$$



Notes

Cubic spline interpolation demystified

We will interpolate N data points by a polynomial of 3rd degree for each i_{th} interval between data point

$$f_i(x) = p_{1_i} x^3 + p_{2_i} x^2 + p_{3_i} x + p_{4_i}, x \in [x_i, x_{i+1}]$$

Interpolation must pass through data points

$$f_i(x_i) = y_i$$
$$f_i(x_{i+1}) = y_{i+1}$$

Two data points (at the interval border) are not enough to set equations for 4 polynomial coefficients.

So we will request twice continuous differentiable

$$f_i'(x_{i+1}) = f_{i+1}'(x_{i+1})$$
$$f_i''(x_{i+1}) = f_{i+1}''(x_{i+1})$$

At the end points second derivative must be set to 0 (so called natural cubic spline)

$$f_1''(x_1) = 0$$
$$f_{N-1}''(x_N) = 0$$

Notes

Matlab built in interpolation

Use matlab `interp1(xdata, ydata, xi, method)` for some of above methods

Where `method` could be

- 'nearest' Nearest neighbor interpolation
- 'linear' Linear interpolation (default)
- 'spline' Cubic spline interpolation
- other see more in help

Notes

Extrapolation - voids outside the measurements region

Do not extrapolate!!!

Unless you have a physical model of the process.

Notes

Notes

Notes

Notes