Splines and applications

Chapter 5. of the book The Elements of Statistical Learning by the Jerome Friedman, Trevor Hastie, Robert Tibshirani.

Bágyi Ibolya
Applied Machine Learning
Master, 2006-2007

Splines and applications

Contents

Splines and applications

1. History of splines

- originally developed for ship-building in the days before computer modeling.
- Pierre Bézier

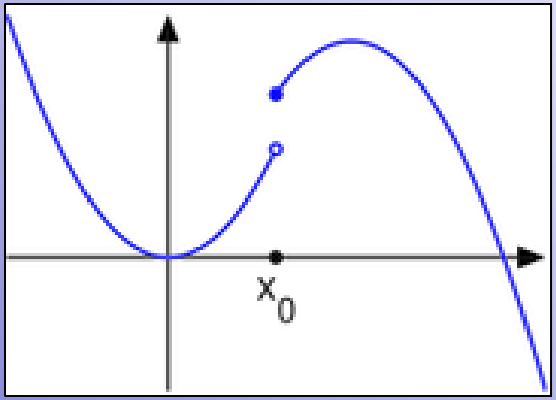
2. What is a spline?

- simply a curve
- In mathematics a spline is a special function defined piecewise by polynomials. In computer science the term spline refers to a piecewise polynomial curve.
- The solution was to place metal weights (called knots) at the control points, and bend a thin metal or wooden beam (called a spline) through the weights.

http://www.macnaughtongroup.com/spline_weights.htm

3. Piecewise Polynomial and Splines

1.) A piecewise polynomial ftn ***f(x)*** is obtained by dividing of X into contiguous intervals, and representing ***f(x)*** by a separate polynomial in each interval.



- The polynomials are joined together at the interval endpoints (knots) in such a way that a certain degree of smoothness of the resulting function is guaranteed.

http://en.wikipedia.org/wiki/Piecewise

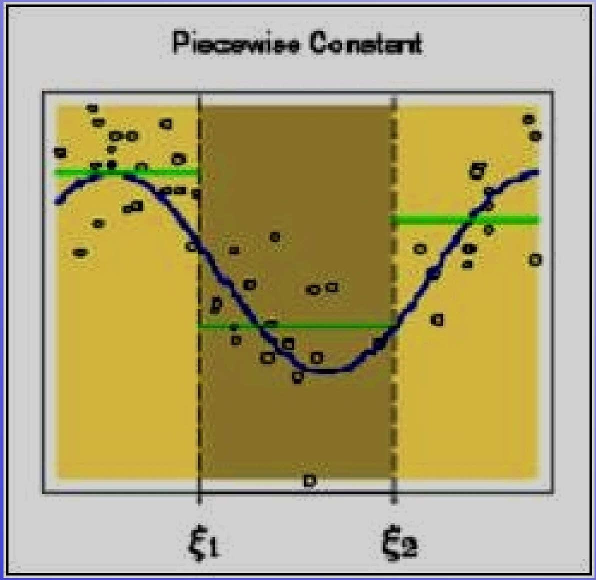
3. Piecewise Polynomial and Splines

Denote by $h_j(X) : IR^{\square} \rightarrow IR$ the jth transformation of X, $j=1\ldots M$. We then model

$$f(X) \approx \sum_{j=1}^{M} \beta_j h_j(X)$$

a linear basis expansion in X.

2.) A piecewise constant:



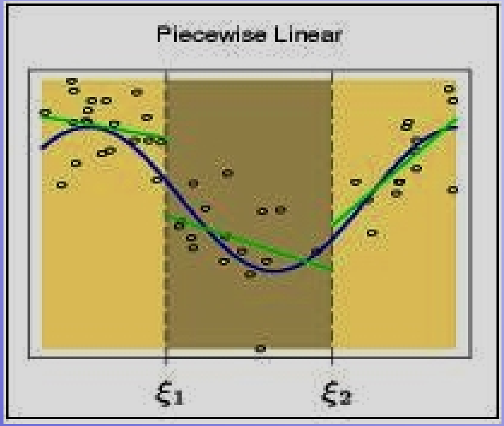
- basis function :

$$h_1(X) = 1_{X<\xi_1} \qquad h_2(X) = 1_{\xi_1 \le X < \xi_2} \qquad h_3(X) = 1_{\xi_2 \le X}$$

- *This panel shows a piecewise constant function fit to some artificial data. The broken vertical lines indicate the position of the two knots ξ_1 and ξ_2.*
- *The blue curve represents the true function.*

3. Piecewise Polynomial and Splines

3.) A piecewise linear

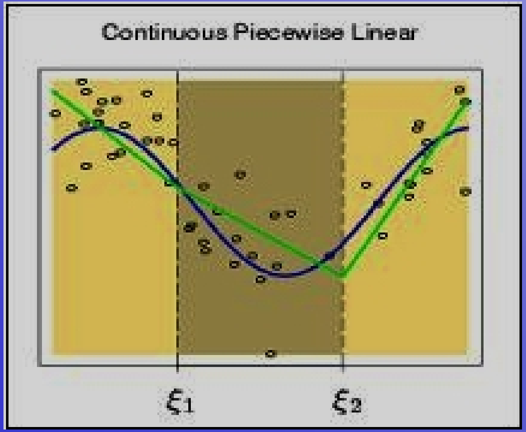


- basis function : three additional basis ftn are needed

$$\boxed{h_{m+3} = h_m(X)X, \;\; m = 1, \ldots, 3}$$

- *the panel shows piecewise linear function fit to the data.*
- *unrestricted to be continuous at the knots.*

4.) continous piecewise linear



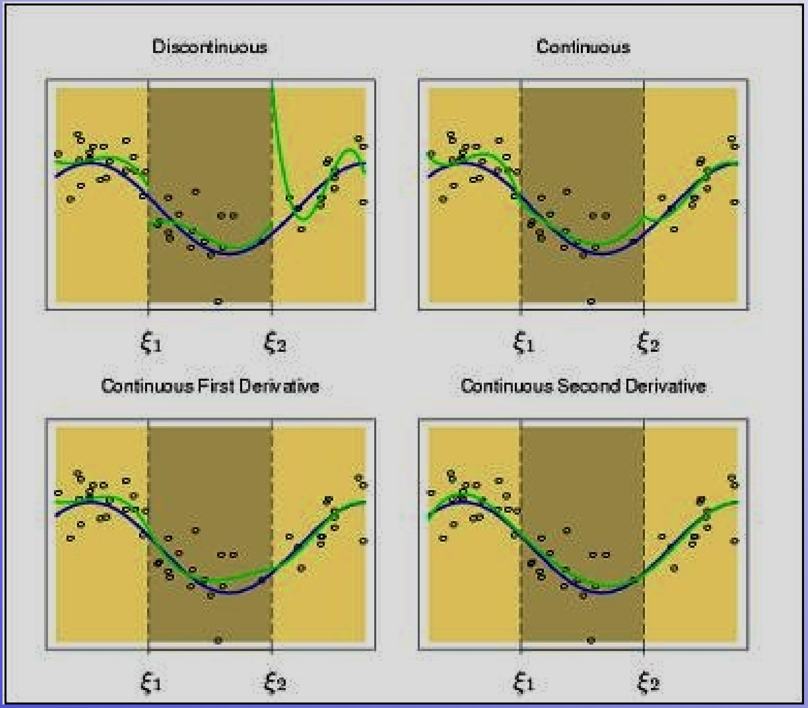
- restricted to be continuous at the two knots.
- linear constraints on the parameters:

$$\boxed{f(\xi_1^-) = f(\xi_1^+) \text{ implies that } \beta_1 + \xi_1\beta_4 = \beta_2 + \xi_1\beta_5.}$$

- *the panel shows piecewise linear function fit to the data.*
- *restricted to be continuous at the knots.*

3. Piecewise Polynomial and Splines

7.) Piecewise cubic polynomial



- \- The function in the lower right panel is continuous and has continuous first and second derivatives.
- \- It is known as a **cubic spline.**
- \- basis function:

$$
\begin{array}{lll}
h_1(X) = 1 & h_3(X) = X^2 & h_5(X) = (X - \xi_1)_+^3 \\
h_2(X) = X & h_4(X) = X^3 & h_6(X) = (X - \xi_2)_+^3
\end{array}
$$

- the pictures show a series of piecewise-cubic polynomials fit to the same data, with increasing orders of continuity at the knots.

3. Piecewise Polynomial and Splines

8.) An order-M spline with knot $\xi_j,\ j = 1, \ldots, K$ is a piecewise-polynomial of order M, and has continuous derivatives up to order M-2.

- a cubic spline has M=4. Cubic splines are the lowest-oder spline for which the knot-discontinuity is not visible to the human eye.

- the piecewise-constant function is an order-1 spline, while the continuous piecewise linear function is an order-2 spline.

In practice the most widely used orders are M=1,2 and 4.

4. Natural Cubic Splines

Natural Cubic Splines

Cubic spline is a spline constructed of piecewise third-order polynomials which pass through a set of m control points.

The second derivate of each polynomial is commonly set to zero at the endpoints, since this provides a boundary condition that completes the system of m-2 equations.

This produces a so-called “natural” cubic spline and leads to a simple tridiagonal system which can be solved easily to give the coefficients of the polynomials.

5.1 Function Spline

spline(x,y,z,d);

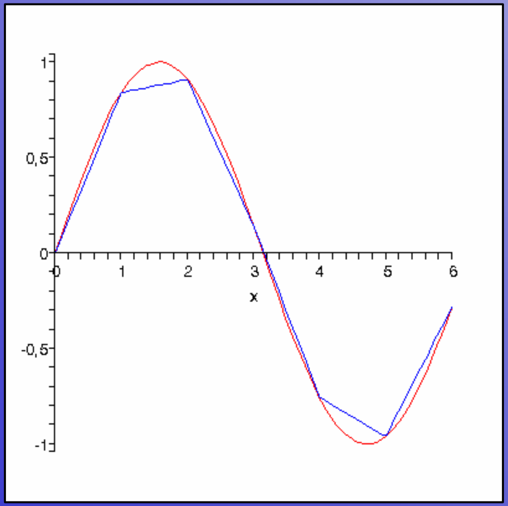
x,y - two vectors or two lists
z - name
d - (optional) positive integer or string

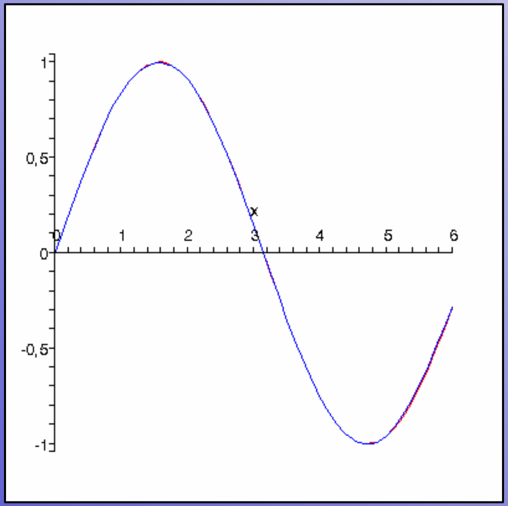
The spline function computes a piecewise polynomial approximation to the X Y data values of degree d (default 3) in the variable z. The X values must be distinct and in ascending order. There are no conditions on the Y values.

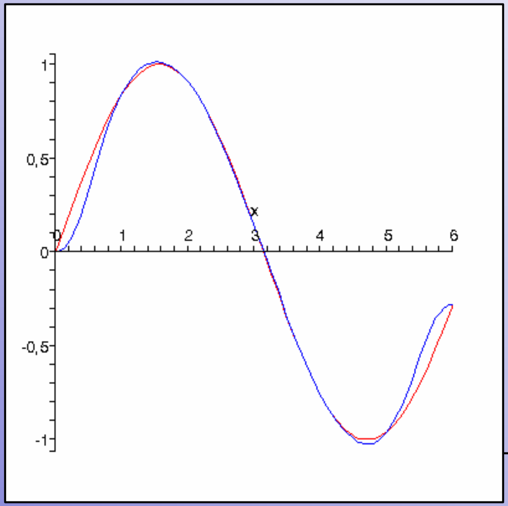
6. Applications

6.1 Maple Spline Function: y=sin(x) and x=[0,6]

```
> plot(sin(x),x=0..6);
> f:=x->sin(x);
> x1:=[0,1,2,3,4,5,6];
> fx1:=map(f,x1);
> plot([sin(x),spline(x1,fx1,x,'linear')],x=0..6,color=[red,blue],style=[line,line]);
> plot([sin(x),spline(x1,fx1,x,'cubic')],x=0..6,color=[red,blue],style=[line,line]);
> plot([sin(x),spline(x1,fx1,x,2)],x=0..6,color=[red,blue],style=[line,line]);
```







6. Applications

6.2 Interpolation with cubic spline

The function is f(x)=sin(π/2*x), x∈[-1,1]. Interpolant the function on -1, 0, 1 with cubic spline, which satisfied the following boundary conditions:

S´(-1)=f’(-1)=0

S´(1)=f’(1)=0

One seeks the cubic spline in the folowing form:

$$S(x)=\begin{cases} P_1(x)=a_1x^3+b_1x^2+c_1x+d_1, & \text{ha } x\in[-1;0] \\ P_2(x)=a_2x^3+b_2x^2+c_2x+d_2, & \text{ha } x\in[0;1] \end{cases}$$

By stating the interpolant conditions, the continuity of the spline is satisfied:

$$1)\quad S(-1)=P_1(-1)=f(-1) \Leftrightarrow -a_1+b_1-c_1+d_1=-1$$

$$2)\quad S(0)=P_1(0)=f(0) \Leftrightarrow d_1=0$$

$$3)\quad S(0)=P_2(0)=f(0) \Leftrightarrow d_2=0$$

$$4)\quad S(1)=P_2(1)=f(1) \Leftrightarrow a_2+b_2+c_2+d_2=1$$

6.2 Interpolation with cubic spline

In the same time the first and the second derivate of the spline needs to be also continous:

$$P_i'(x) = 3a_i x^2 + 2b_i x + c_i,\quad P_i''(x) = 6a_i x + 2b_i\ ,\ (i = 1,2)$$

$$\begin{array}{lll} 5) & P_1'(0) = P_2'(0) \Leftrightarrow & c_1 = c_2 \\ 6) & P_1''(0) = P_2''(0) \Leftrightarrow & b_1 = b_2 \end{array}$$

One obtains 6 equations involving 8 unknows, and in this way the Hermite condition needs to be taken into account:

$$\begin{array}{ll} 7) & P_1'(-1) = 3a_1 - 2b_1 + c_1 = 0 \\ 8) & P_2'(1) = 3a_2 + 2b_2 + c_2 = 0 \end{array}$$

Solve the system of equations. By using the equations 2), 3), 5) and 6) one can reduce the original system:

$$\begin{array}{ll} 1) & -a_1 + b_1 - c_1 = -1 \\ 4) & a_2 + b_1 + c_1 = 1 \\ 7) & 3a_1 - 2b_1 + c_1 = 0 \\ 8) & 3a_2 + 2b_1 + c_1 = 0 \end{array}$$

6. Applications

6.2 Interpolation with cubic spline

Solving this:

$$4)-1)\quad a_2+a_1+2c_1=2 \quad\Rightarrow\quad a_1+a_2=2-2c_1$$

$$7)+8)\quad 3(a_1+a_2)+2c_1=0 \quad\Rightarrow\quad 3(2-2c_1)+2c_1=6-4c_1=0 \quad\Rightarrow\quad c_1=c_2=\frac{3}{2} \text{ és } a_1+a_2=-1$$

$$1)+4)\quad a_2-a_1+2b_1=0 \quad\Rightarrow\quad a_1-a_2=2b_1$$

$$7)-8)\quad 3(a_1-a_2)-4b_1=0 \quad\Rightarrow\quad 3(2b_1)-2b_1=4b_1=0 \quad\Rightarrow\quad b_1=b_2=0\,.$$

One obtains a1-a2=0 and a1+a2=-1 => a1=a2=-1/2.

Finally the sought spline reads as follows:

$$S(x)=\begin{cases}-\frac{1}{2}x^3+\frac{3}{2}x, & \text{ha } x\in[-1;0]\\ -\frac{1}{2}x^3+\frac{3}{2}x, & \text{ha } x\in[0;1]\,.\end{cases}$$

7. Implementation of Spline

7.1 A programme for calculating spline

- procedure polynom creation

```
> creation_poly:=proc(d1,d2,x1,x2,y1,y2)
    local x,h:
    h:=x2-x1:
    unapply(y1*(x2-x)/h + y2*(x-x1)/h
    -h*h/6*d1*((x2-x)/h-((x2-x)/h)^3)
    -h*h/6*d2*((x-x1)/h-((x-x1)/h)^3),x)
    end:
```

7. Implementation of Spline

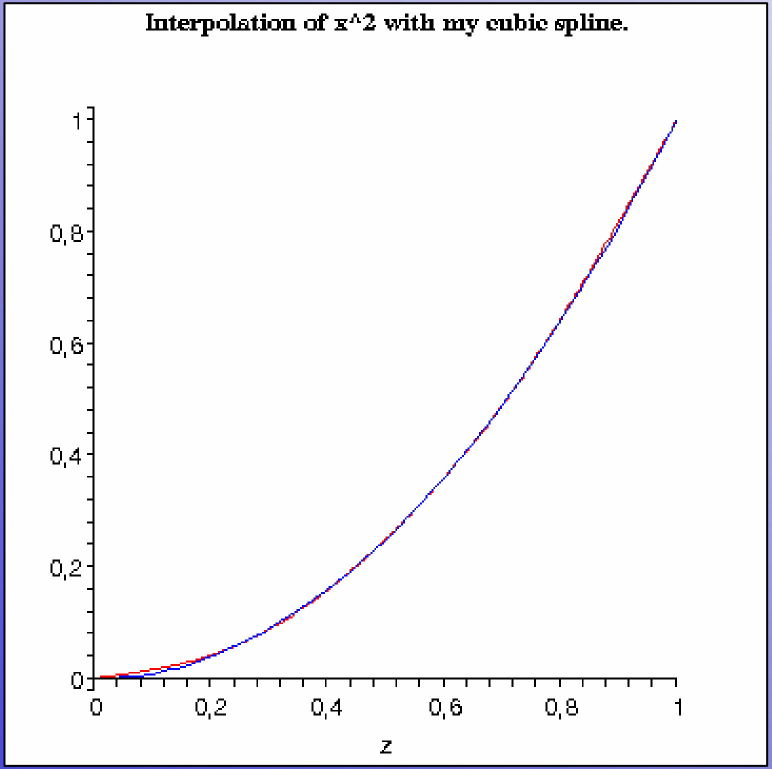
- Procedure spline

```
> s:=proc(x::list(numeric),y::list(numeric))
    local n,i,j,mat,res,sol,draw,h1,h2,pol:
    if nops(x)<>nops(y) then ERROR(„number of x and y most be equal") fi:
    n:=nops(x):
    mat:=[1,seq(0,j=1..n-1)],[seq(0,j=1..n-1),1]:
    res:=0,0:
    for i from 2 to n-1 do
    h1:=x[i]-x[i-1]:
    h2:=x[i+1]-x[i]:
    mat:=[seq(0,j=1..i-2),h1*h1,2*(h1*h1+h2*h2),h2*h2,seq(0,j=1..n-i-1)],mat:
    res:=6*(y[i+1]-2*y[i]+y[i-1]),res:
    od:
    sol:=linsolve([mat],[res]):
    draw:=NULL:
    for i to n-1 do
    pol:=creation_poly(sol[i],sol[i+1],x[i],x[i+1],y[i],y[i+1]):
    draw:=plot(pol(z),z=x[i]..x[i+1]),draw:
    od:
    eval(draw):
    end:
```

7. Implementation of Spline

7.2 Testing the programme

> **test1:=s([0,1/4,1/2,3/4,1],[0,1/16,1/4,9/16,1]):**
>
> **display(test1,plot(x^2,x=0..1,color=blue));**

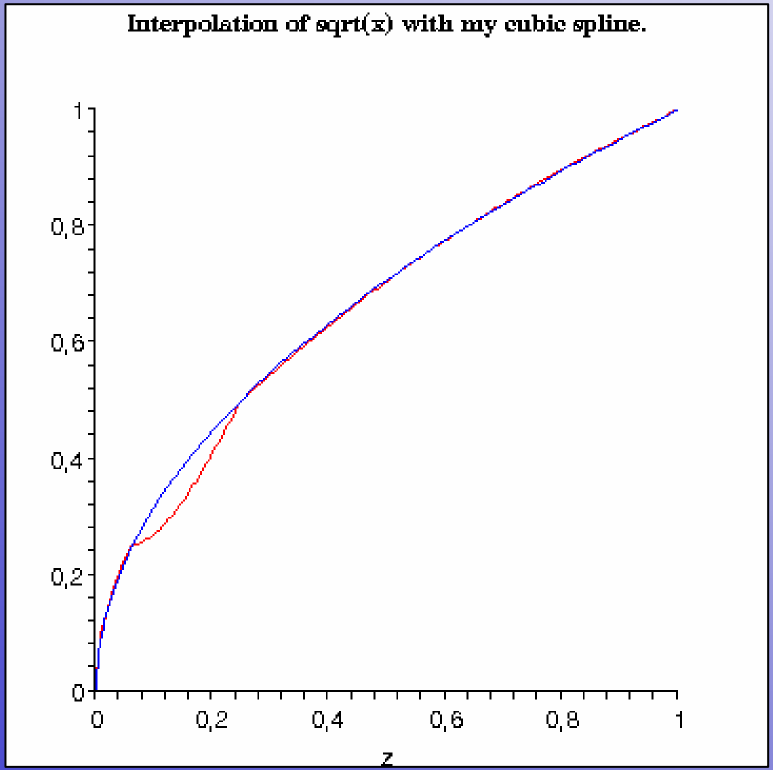


7. Implementation of Spline

7.2 Testing the programme

> **test2:=s([0,1/100,1/25,1/16,1/4,16/25,1],[0,1/10,1/5,1/4,1/2,4/5,1]):**

> **display(test2,plot(sqrt(x),x=0..1,color=blue), view=[0..1,0..1]);**



8. Glossary

piecewise: a piecewise-defined function f(x) of a real variable x is a function whose definition is given differently on disjoint intervals of its domain. A common example is the absolute value function. <-BACK

spline: in mathematics a spline is a special function defined piecewise by polynomials. In computer science the term spline refers to a piecewise polynomial curve. <-BACK

cubic spline: is a spline constructed of piecewise third-order polynomials which pass through a set of m control points. The second derivate of each polynomial is commonly set to zero at the endpoints, since this provides a boundary condition that completes the system of m-2 equations. This produces a so-called "natural" cubic spline and leads to a simple tridiagonal system which can be solved easily to give the coefficients of the polynomials. <-BACK

9. Bibliography

Jerome Friedman, Trevor Hastie, Robert Tibshirani **(2001).** *The Elements of Statistical Learning*, Basis expansion and regularization: 115-164.

Course **(2001-2002).** *Symbolic and Numerical Computation.* “Babes-Bolyai” University

Basis expansion and regularization – from site of Seoul National University
http://stat.snu.ac.kr/prob/seminar/ElementsOfStatisticalLearning/Chapter5.ppt

Spline Cubic – from Wolfram MathWorld
http://mathworld.wolfram.com/CubicSpline.html

Piecewise – Wikipedia the free encyclopedia
http://en.wikipedia.org/wiki/Piecewise

Splines – from site of University of Oregon
http://www.uoregon.edu/~greg/math352-04w/splines.pdf

Spline weight image – from site of MacNaughton Yacht Designs
http://www.macnaughtongroup.com/spline_weights.htm