Section 4.1

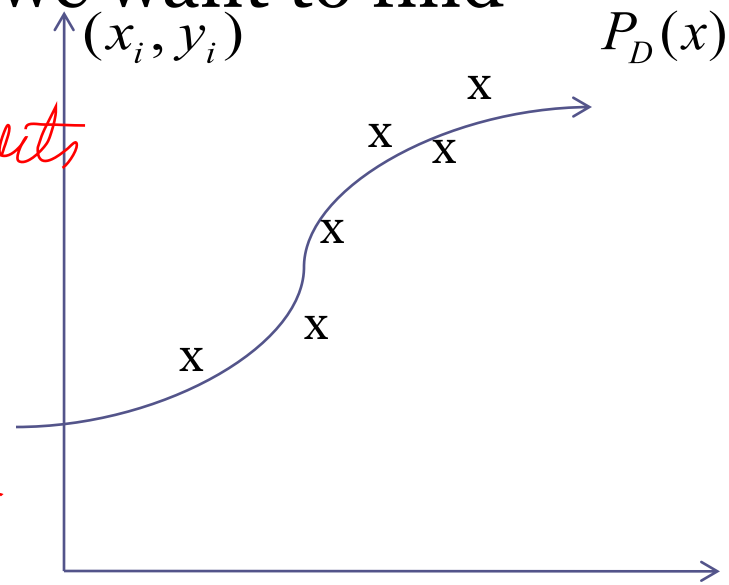
Polynomial Fits

N data points

Polynomial Fits

Given a set of data points $(x_i, y_i)_{i=1}^{N}$, and a degree D of a polynomial, we want to find

$$k_0 \ldots\ldots\ldots k_D$$ ← D+1 coefficients



such that

$$P_D(x) = k_0 + k_1 x + \ldots\ldots\ldots + k_D x^D$$

Note: D<N

$$= k_D x^D + k_{D-1} x^{D-1} + \ldots + k_1 x + k_0$$

Remarks #1

If ***D=N-1***, then one can solve for a N equations & N unknowns:

$P(x) = k_0 + k_1 x + k_2 x^2 + \dots + k_D x^D$; $y_i = P(x_i)$ for $1 \le i \le N$

known

known

$$k_0 + k_1 x_1 + \dots\dots\dots k_D x_1^D = y_1$$

$$k_0 + k_1 x_2 + \dots\dots\dots k_D x_2^D = y_2$$

$$\downarrow \qquad\qquad \downarrow$$

$$k_0 + k_1 x_N + \dots\dots\dots k_D x_N^D = y_N$$

Ex: 3[rd] degree polynomial with 4 variables

$A\vec{K} = \vec{y}$; $A = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^D \\ 1 & x_2 & x_2^2 & \dots & x_2^D \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^D \end{bmatrix}$

We need to solve for coefficients

A system of N linear eqns & N unknowns $k_0, k_1, \dots, k_D$; $\vec{K} = \begin{bmatrix} k_0 \\ k_1 \\ \vdots \\ k_D \end{bmatrix}$; $\vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$

$$\vec{A}\vec{K}=\vec{y}$$

$$\vec{K}=A^{-1}\vec{y}$$

Von DeMond Linear System

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots\ldots & x_1^D \\ 1 & x_2 & x_2^2 & \ldots\ldots & x_2^D \\ & \downarrow & & \downarrow & \\ 1 & x_N & x_N^2 & \ldots\ldots & x_N^D \end{bmatrix}}_{A} \begin{bmatrix} k_0 \\ k_1 \\ \downarrow \\ k_D \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \downarrow \\ y_N \end{bmatrix}$$

- Inverting this matrix could lead to unstable coefficients.
- Small change in the matrix $(x_i \approx x_i + \xi)$ could result in drastic change in $[k_0, k_1, \ldots. k_D]$

When N is large, do not approximate your model with the (N-1)degree polynomial
Note: When approximate N points with a (N-1) degree polynomial, the R^2 value =1
(fits in all points -> SSE ≈ 0)

Remarks #2

$(x_i\text{'s}, y_i\text{'s}, D) \rightarrow \boxed{\text{polyfit}} \rightarrow \text{coefficients } C$

$c, x \rightarrow \boxed{\text{Polyval}} \rightarrow y$

- For D<N, one can proceed with the D-variable, sum square error function & try to minimize it.

$$E(k_0,.......k_D) = \sum_{i=1}^{N} (P_D x_i) - y_i)^2$$

- Matlab has a built in function called ***"polyfit.m"*** to obtain coefficients ***c=[c(1) c(2) c(3)…c(D)]***
- Matlab built in function ***polyval.m*** to evaluate y values in fitted model.

$$y = c(1)x^D + c(2)x^{D-1} + \cdots + c(D-1)x + c(D)$$

$P(x) = C(1)\,x^{D} + C(2)\,x^{D-1} + \cdots + C(D-1)\,x + C(D)$

Matlab

$= 0 \cdot x^{D} + \cdots + 2.1000000016816 72\,x^{3}$
$0.0000004968695\,x^{2}$
$-0.99999923\,8716\,x$
0.999999995962545

Example:

>> x=[1,4,5,.....]; y=[3,8,16......]; %data points

>> D=3; %Degree

>>c=polyfit(x,y,D) %c=coefficient set

Note: $P_3(x) = c_{(1)}x^3 + c_{(2)}x^2 + c_{(3)}x + c_{(4)}$

>>xx=linspace(min(x), max(x), 100); %100 points

>>yy=polyval(c,xx) %fitted model $yy = P_3(xx)$

Mablab Example

- V=[...................................];
- P=[................................];
- D=4;
- plot(V,P,'rs','LineWidth',3);hold on;
- C=polyfit(V,P,D);
- xx=linspace(min(V),max(V),100);
- yy=polyval(C,xx); %yy=P3(xx) xx=Cofficient location
- plot(xx,yy,'-bh');hold on;
- title('.............');
- legend('Original Data','Model')
- ylabel('y');
- xlabel('x');
-

Hw §4.1
6, 9, 11

Ex: Polyfit polynomial degree of 4