Lecture 5 Polynomial Approximation

- Polynomial Interpolation
  - Example
  - Limitation
- Polynomial Approximation (fitting)
  - Line fitting
  - Quadratic curve fitting
  - Polynomial fitting

Interpolation Vs approximation

- An interpolating polynomial is expected to satisfy all constraints of paired data
- An interpolating polynomial is unable to retrieve an original target function when noise paired data are provided
- For noise paired data, the goal of polynomial fitting is revised to minimize the mean square approximating error

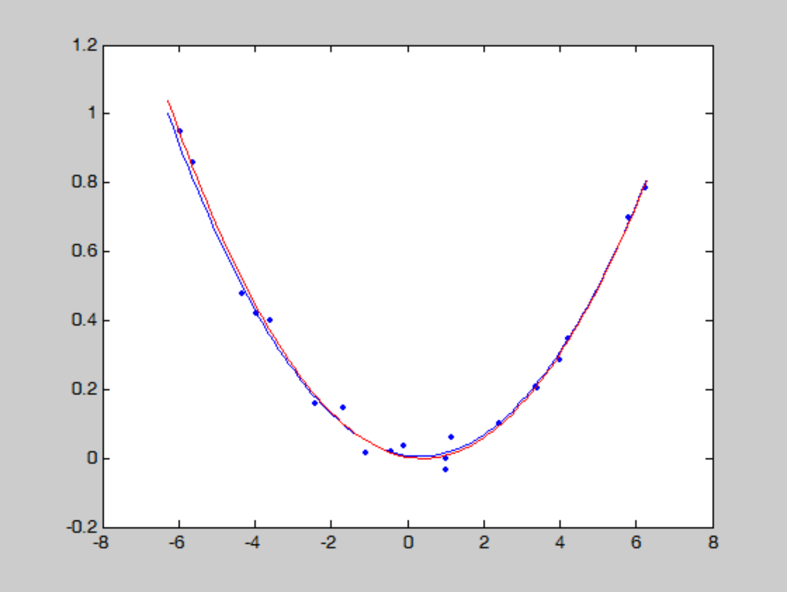
Polynomial approximation

- Given paired data, (x_i,y_i), $i=1,\ldots,n$, the approximating polynomial is required to minimize the mean square error of approximating y_i by $f(x_i)$

$$E = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

Polynomial fitting

```
fstr=input('input a function:
x.^2+cos(x) :','s');
f=inline(fstr);
x=linspace(-2*pi,2*pi);
y=f(x); plot(x,y); hold on;
N=100;
x=rand(1,N)*2*2*pi-2*pi;
ns=rand(1,N)*0.1-0.05;
y=f(x)+ns;
plot(x,y,'.');
m=input('polynomial degree:');
p=polyfit(x,y,m)
y_hat=polyval(p,x);
E=mean((y_hat-y).^2)

z=linspace(-2*pi,2*pi);
z_hat=polyval(p,x);
plot(z,z_hat,'r')
```



Line fitting

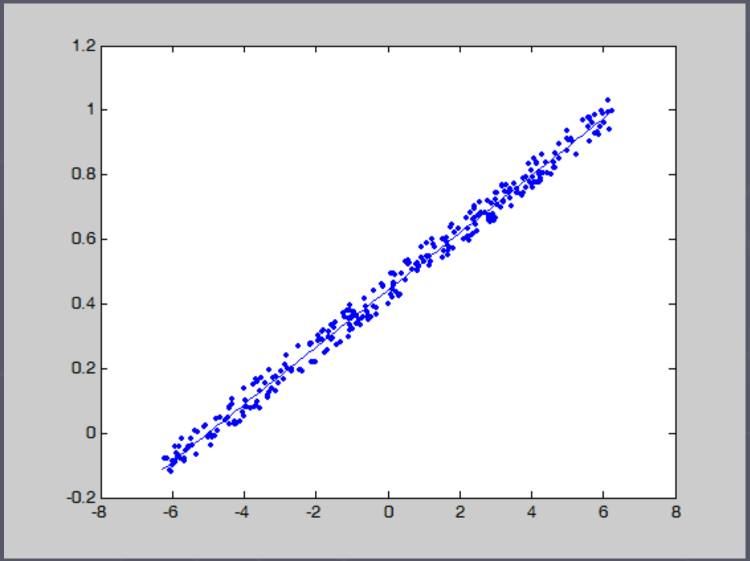
- Minimizaing the mean square approximating error

$$E(a,b) = \frac{1}{n}\sum_{i=1}^{n}(ax_i + b - y_i)^2$$

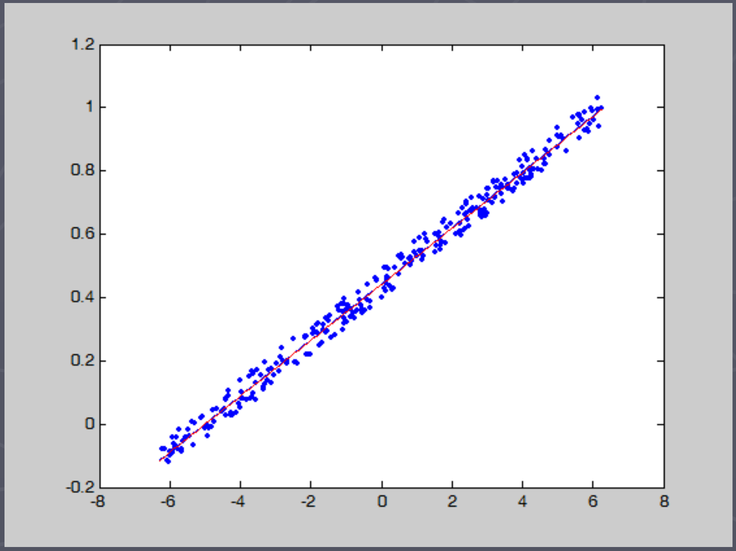
$$y = ax + b$$

Line fitting

```
>> fa1d_polyfit
input a function: x.^2+cos(x) :x+5
keyin sample size:300
polynomial degree:1

E =

  8.3252e-004
```



Red:
Approximating polynomial



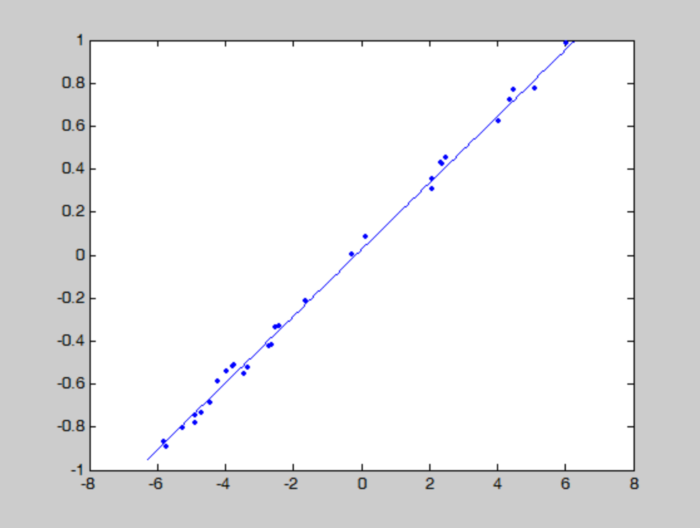
Line fitting

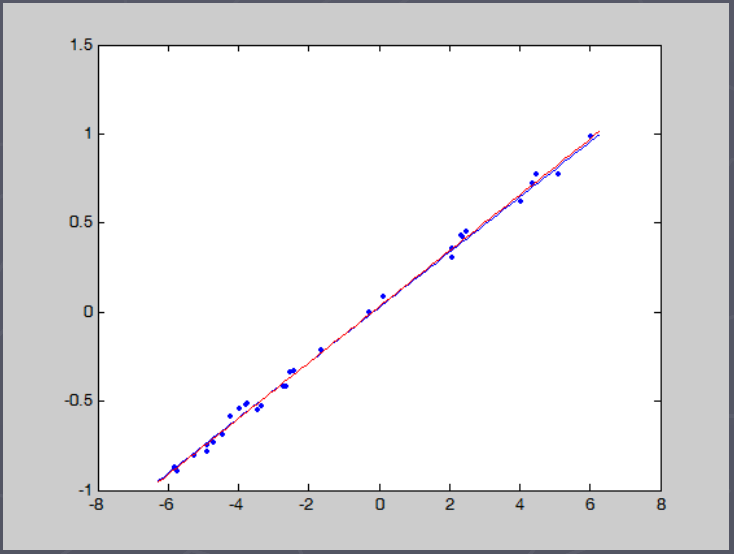
>> fa1d_polyfit
input a function: x.^2+cos(x) :3*x+1/2
keyin sample size:30
polynomial degree:1

E =

0.0010

Red:
Approximating polynomial





Objective function I

- Line fitting

$$E_1(\theta) \equiv E_{\text{Line-fitting}}(a,b) = \frac{1}{n}\sum_{i=1}^{n}(ax_i + b - y_i)^2$$

$$y = ax + b$$

- E_1 is a quadratic function of a and b
- Setting derivatives of E_1 to zero leads to a linear system

$$E_1(\theta) \equiv E_{\text{Line-fitting}}(a, b) = \frac{1}{n}\sum_{i=1}^{n}(ax_i + b - y_i)^2$$

$$\mathrm{E}_1(\theta) \equiv \mathrm{E}_{\text{Line-fitting}}(\mathrm{a},\mathrm{b}) = \frac{1}{n}\sum_{i=1}^{n}(ax_i + b - y_i)^2$$

$$\frac{dE_1}{d\theta} = 0$$

$$\frac{dE_1}{da} = \frac{2}{n}\sum_{i=1}^{n}(ax_i + b - y_i)x_i = 0$$

$$\frac{dE_1}{db} = \frac{2}{n}\sum_{i=1}^{n}(ax_i + b - y_i) = 0$$

$$\frac{dE_1}{da} = \frac{2}{n}\sum_{i=1}^{n}(ax_i + b - y_i)x_i = 0$$

$$\frac{dE_1}{db} = \frac{2}{n}\sum_{i=1}^{n}(ax_i + b - y_i) = 0$$

$$\begin{Bmatrix} \sum_{i=1}^{n}(ax_i^2 + bx_i - y_i x_i) = 0 \\ \sum_{i=1}^{n}(ax_i + b - y_i) = 0 \end{Bmatrix} \Rightarrow \begin{Bmatrix} \sum_{i=1}^{n} x_i^2 a + \sum_{i=1}^{n} x_i b = \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} ax_i + nb = \sum_{i=1}^{n} y_i \end{Bmatrix}$$

$$\begin{cases} \sum_{i=1}^{n} x_i^2 a + \sum_{i=1}^{n} x_i b = \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i a + nb = \sum_{i=1}^{n} y_i \end{cases}$$

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{pmatrix}$$

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i \, y_i \\ \sum_{i=1}^{n} y_i \end{pmatrix}$$

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n} x_i \, y_i \\ \sum_{i=1}^{n} y_i \end{pmatrix}$$

Objective function II

- Quadratic polynomial fitting

$$E_2(\theta) \equiv E_{\text{QuadraticCurveFitting}}(a, b, c)$$

$$= \frac{1}{n}\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)^2$$

$$y = ax^2 + bx + c$$

- E_2 are quadratic
- Setting derivatives of E_2 to zero leads to a linear system

$$E_2(\theta) \equiv E_{\mathrm{QuadraticCurveFitting}}(a, b, c)$$

$$= \frac{1}{n} \sum_{i=1}^{n} (a x_i^2 + b x_i + c - y_i)^2$$

$$\mathrm{E}_2(\theta) \equiv \mathrm{E}_{\text{QuadraticCurveFitting}}(\mathrm{a},\mathrm{b},\mathrm{c}) = \frac{1}{n}\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)^2$$

$$\frac{dE_2}{d\theta} = 0$$

$$\frac{dE_2}{da} = \frac{2}{n}\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)x_i^2 = 0$$

$$\frac{dE_2}{db} = \frac{2}{n}\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)x_i = 0$$

$$\frac{dE_3}{dc} = \frac{2}{n}\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i) = 0$$

$$\begin{cases} \sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)x_i^2 = 0 \\ \sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)x_i = 0 \\ \sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i) = 0 \end{cases}$$

$$\begin{cases} \sum_{i=1}^{n}(ax_i^4 + bx_i^3 + cx_i^2 - y_i x_i^2) = 0 \\ \sum_{i=1}^{n}(ax_i^3 + bx_i^2 + cx_i - y_i x_i) = 0 \\ \sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i) = 0 \end{cases}$$

$$\begin{cases} \sum_{i=1}^{n} (ax_i^4 + bx_i^3 + cx_i^2 - y_i x_i^2) = 0 \\ \sum_{i=1}^{n} (ax_i^3 + bx_i^2 + cx_i - y_i x_i) = 0 \\ \sum_{i=1}^{n} (ax_i^2 + bx_i + c - y_i) = 0 \end{cases}$$

$$\begin{cases} \sum_{i=1}^{n} ax_i^4 + \sum_{i=1}^{n} bx_i^3 + \sum_{i=1}^{n} cx_i^2 = \sum_{i=1}^{n} y_i x_i^2 \\ \sum_{i=1}^{n} ax_i^3 + \sum_{i=1}^{n} bx_i^2 + \sum_{i=1}^{n} cx_i = \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} ax_i^2 + \sum_{i=1}^{n} bx_i + nc = \sum_{i=1}^{n} y_i \end{cases}$$

$$\begin{pmatrix} \sum_{i=1}^{n} a x_i^4 + \sum_{i=1}^{n} b x_i^3 + \sum_{i=1}^{n} c x_i^2 = \sum_{i=1}^{n} y_i x_i^2 \\ \sum_{i=1}^{n} a x_i^3 + \sum_{i=1}^{n} b x_i^2 + \sum_{i=1}^{n} c x_i = \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} a x_i^2 + \sum_{i=1}^{n} b x_i + nc = \sum_{i=1}^{n} y_i \end{pmatrix}$$

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i x_i^2 \\ \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{pmatrix}$$

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i x_i^2 \\ \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{pmatrix}$$

數值方法

Function p=my_2fit(x,y)

Start

Exit

```
A(1,1)=sum(x.^4);
A(1,2)=sum(x.^3);A(2,1)=A(1,2);
A(1,3)=sum(x.^2);A(3,1)=A(1,3);
A(2,2)=A(3,1);
A(2,3)=sum(x);A(3,2)=A(2,3);
A(3,3)=length(x);d(1)=sum(y.*(x.^2))
....
p=inv(A)*d'
```

Quadratic polynomial fitting

- Minimization of an approximating error

$$E(a,b,c) = \frac{1}{n}\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)^2$$

$$y = ax^2 + bx + c$$

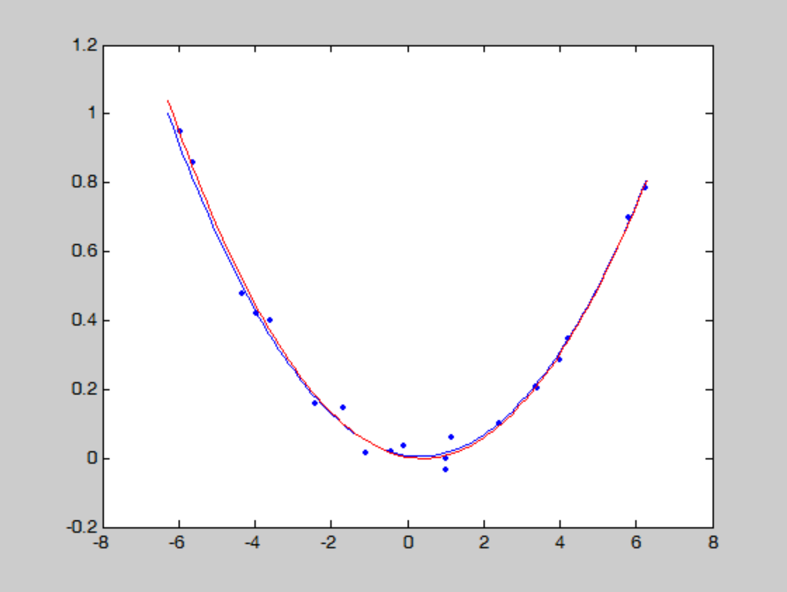
Quadratic poly fitting

input a function: x.^2+cos(x) :3*x.^2-2*x+1
keyin sample size:20
polynomial degree:2

E =

6.7774e-004

a=3
b=-2
c=1



Data driven polynomial approximation

- Minimization of Mean square error (mse)
- Data driven polynomial approximation
  - f : a polynomial p_m
  - Polynomial degree m is less than data size n

$$E = \frac{1}{n}\sum_{i=1}^{n}(p_m(x_i) - y_i)^2$$

Special case: polynomial interpolation

$$E = \frac{1}{n}\sum_{i=1}^{n}(p_m(x_i) - y_i)^2$$

$$E = 0 \Rightarrow y_i = p_m(x_i) \text{ for all i}$$

POLYFIT: Fit polynomial to data

- polyfit(x,y,m)
  - x : input vectors or predictors
  - y : desired outputs
  - m : degree of interpolating polynomial
- Use m to prevent from over-fitting
- Tolerance to noise

POLYFIT: Fit polynomial to data

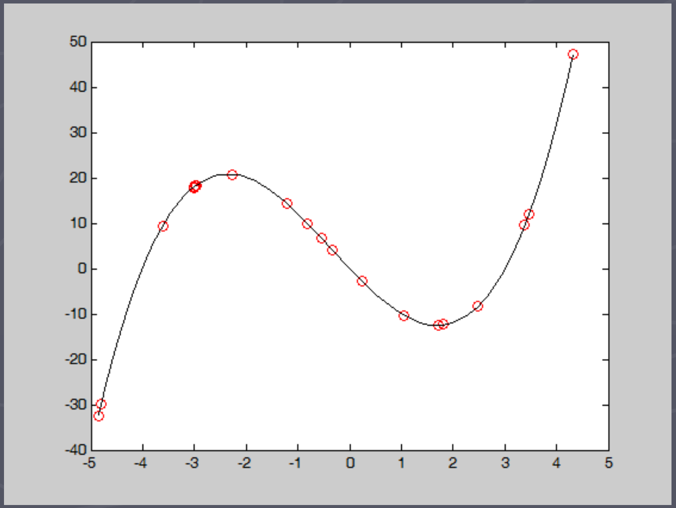
```
P=poly([-4 0 3]); n=20;m=3;
```

A polynomial determined by zeros in [-4,0,3]

```
x=rand(1,20)*10-5;
y=polyval(P,x);
nois=rand(1,20)*0.5-0.25;
plot(x,y+nois, 'ro');
```

Plot the interpolating polynomial

```
v=linspace(min(x),max(x));
p=polyfit(x,y+nois,m);hold on;
plot(v,polyval(p,v),'k');
```

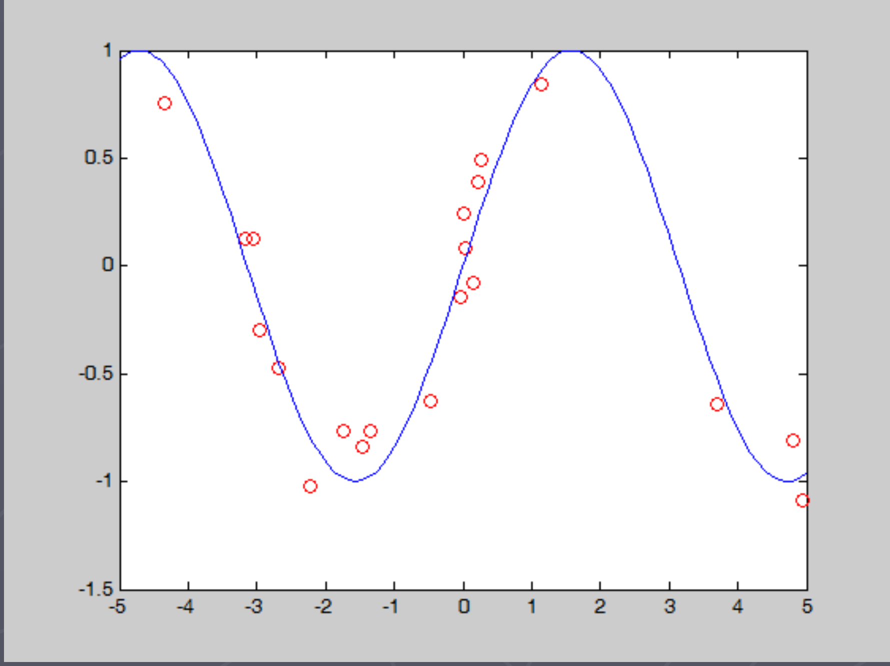


Non-polynomial

- sin

```
fx=inline('sin(x)'); n=20;m=3
```

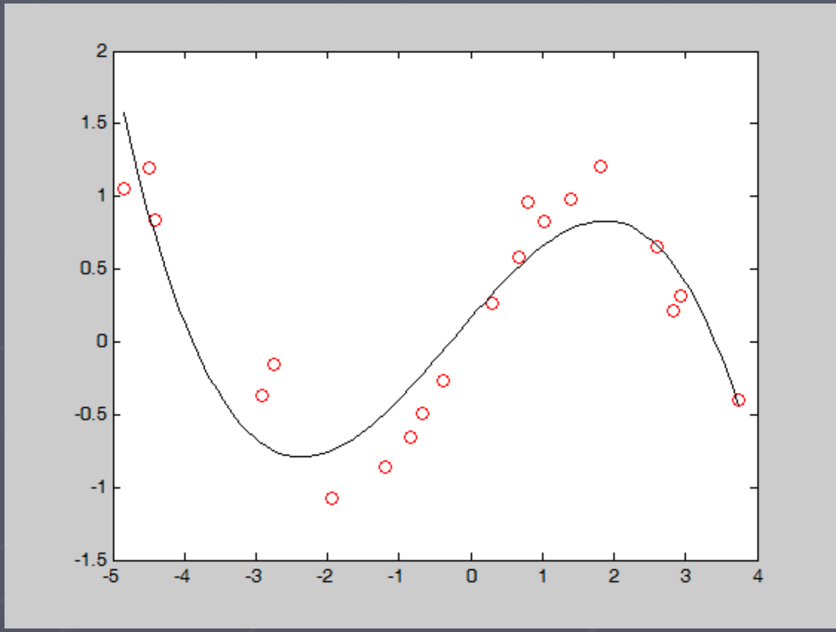
```
x=rand(1,n)*10-5;
y=fx(x);
nois=rand(1,n)*0.5-0.25;
plot(x,y+nois, 'ro');hold on
```



Under-fitting

m=3

```
v=linspace(min(x),max(x));
p=polyfit(x,y+nois,m);hold on;
plot(v,polyval(p,v),'k');
```



Under-fitting due to approximating non-polynomial by low-degree polynomials

Intolerant mse

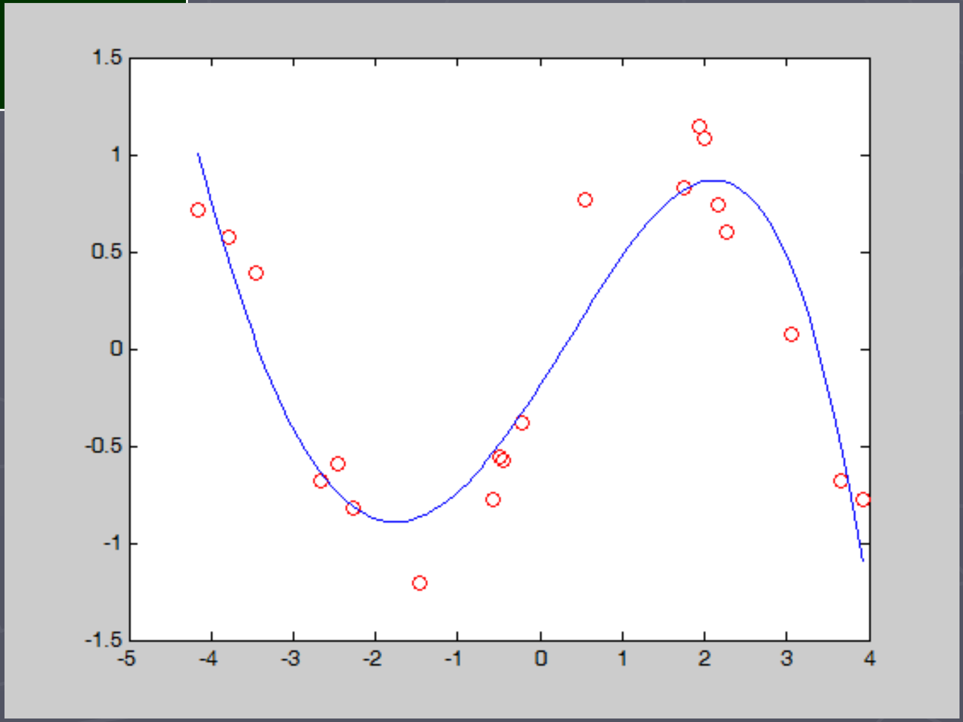
```
>> mean((polyval(p,x)-(y+nois)).^2)

ans =

    0.0898
```

Under-fitting causes intolerant mean square error

Under-fitting

```
m=4;
v=linspace(min(x),max(x));
p=polyfit(x,y+nois,m);hold on;
plot(v,polyval(p,v), 'b');
```

M=4

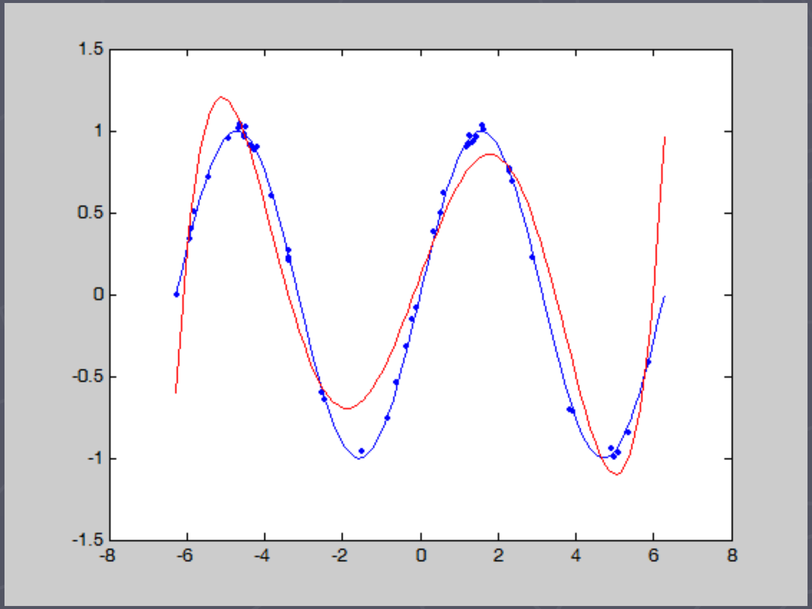


Fitting non-polynomial

>> fa1d_polyfit
input a function: x.^2+cos(x) :sin(x)
keyin sample size:50
polynomial degree:5

E =

0.0365



Fitting non-polynomial

>> fa1d_polyfit
input a function: x.^2+cos(x) :tanh(x+2)+sech(x)
keyin sample size:30
polynomial degree:5

E =

    0.0097