**EAST WEST UNIVERSITY**

**LAB – 7**

**Polynomial Regression**

**Course Code: ICE470**

**Course Title: Applied Numerical Methods**

**Section – 01**

**Submitted To:**

Muhammad Suhail Najeeb  
Lecturer

Dept. of ECE

East West University

**Submitted By:**

MD Imran Ahmed

ID: 2015-2-50-018

**Date of Submission**: 28-07-2019

**Objective**: Use polynomial regression to find the best polynomial fit of the following values

**MATLab Code:**

clc;

clearall;

closeall;

m = 4; % degree of polynomila

n = 10 %num of total data

x = [0 1 2 3 4 5 6 7 8 9];

y = [2.1 7.7 13.6 27.2 40.9 61.1 65.2 70.3 75.8 80.9];

xm = mean(x);

ym = mean(y);

XI = [];

forj = 1:2\*m+1

sum = 0;

fori = 1:n

sum = sum + power(x(i),j-1);

end

XI(j) = sum;

end

XIYI = []

forj = 1: m+1

sum = 0;

fori = 1:n

sum = sum + power(x(i),j-1)\*y(i);

end

XIYI(j) = sum;

end

A = zeros(m+1,m+1); %creating a matrix

idx = 1;

fori = 1:m+1

forj = 1: m+1

A(idx) = XI(i+j-1);

idx = idx+1;

end

end

B = transpose(XIYI)

%-------------------------------------------

%Gauss elimination to solve the system

n = m+1;

%forward elimination

fork = 1: n-1

fori = k+1:n

factor = A(i,k)/A(k,k);

forj = 1:n

A(i,j) = A(i,j)-factor\*A(k,j);

end

B(i) = B(i) - factor\*B(k);

end

end

%back substitution

X = zeros(1,n);

X(n) = B(n)/A(n,n);

fori = n-1: -1:1

sum = B(i);

forj = i+1:n

sum = sum - A(i,j)\*X(j);

end

X(i) = sum/A(i,i);

end

display(X);

%-----------------------------------------------

%polynomial equation

f = @(x) 3.3115-5.1417\*x +7.4161\*x.^2 - 1.1055\*x.^3 + 0.0501\*x.^4;

yy = f(x);

plot(x,y,'\*');

holdon;

plot(x,yy,'r');

%-----------------------------------------------------

**%Error analysis of the polynomial fit**

st = 0;

sr = 0;

sumx = 0;

sumy = 0;

for i = 1:n

sumx = sumx + x(i);

sumy = sumy + y(i);

end

fprintf('Computations for an error analysis of the polynomial fit: \n');

fprintf('x \t y \t \t \t st \t \t \t sr \n');

for i = 1:n

st0 = (y(i) - ym)^2;

sr0 = (y(i) - 3.3115 + 5.1417\*x(i) -7.4161\*x(i)^2 + 1.1055\*x(i)^3 - 0.0501\*x(i)^4)^2;

st = st + st0;

sr = sr + sr0;

fprintf('%d \t %.5f \t %.5f \t %.5f \n',x(i),y(i),st0,sr0);

end

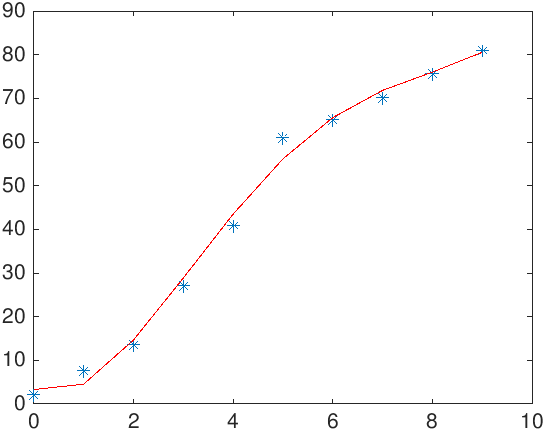
fprintf('%d \t %.5f \t %.5f \t %.5f \n\n',sumx,sumy,st,sr);

syx = (sr/(n-2))^(0.5);

r2 = (st - sr)/st;

fprintf('sy/x = %.5f \t & \t r2 = %.5f',syx,r2);

**Plotting the data points along with the polynomial:**



**Output:**

X = [3.3115

-5.1417

7.4161

-1.1055

0.0501]

**Computations for an error analysis of the polynomial fit:**

Computations for an error analysis of the polynomial fit:

x y st sr

0 2.10000 1796.06440 1.46773

1 7.70000 1352.76840 10.04573

2 13.60000 953.57440 1.10271

3 27.20000 298.59840 2.69255

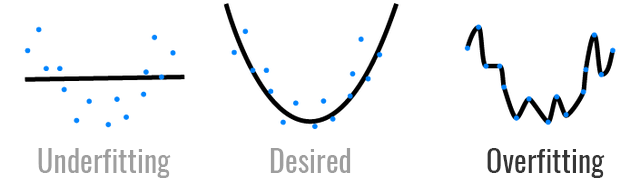
4 40.90000 12.81640 6.63526

10 91.50000 4413.82200 21.94399

sy/x = 2.70456 & r2 = 0.99503>>

**Discussion:**

If we add high order polynomial the polynomial regression line overfits in our data. But uses of lower order polynomial causes under fits.



In our case, when we tried to add 7 or 8th order polynomial line through our data the overfitting takes the place.

**Error Analysis:**

**Standard deviation:**

Standard deviation for the regression line

S(y/x) = = 2.70456

This is called the *standard error of the estimate*. The subscript notation (y/x) designates that the error is for a predicted value of y corresponding to a particular value of x. also we divide by n-2 because two data-derived estimates a0 and a1were used to compute Sr

**Coefficient of determination:**

r2 = = 0.99503

R squared, also known as the coefficient of determination, is a measure to indicate how close the data is to the fitted regression line. The value of the R-Squared is the percentage of variation of the response variable(y) that is explained by a linear model. R-squared is always between 0 and 100%. 0% indicates that the model explains none of the variability of the response data around its mean.100% indicates that the model explains all the variability of the response data around its mean.

We can say that **99.503%** of the variation of the salary is explained by this simple linear model