

Exercise 2: Regression Using Least Squares

Write a program in python (ipython-notebook) to implement the least squares solution for the polynomial curve fitting problem. Follow the instructions below:

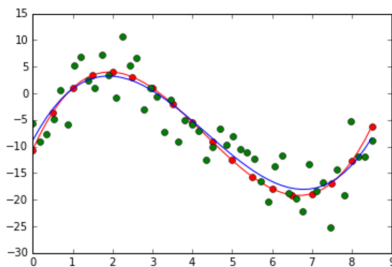
1. Generate 50 2D-data points using the following function: $y = 0.4345x^3 - 5.607x^2 + 16.78x - 10.61$
2. Add Gaussian random noise to the data. Look at the sample code here:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from scipy import linalg

def actual_distribution():
    x = np.arange(0, 9, 0.5);
    y = 0.4345*np.power(x,3) - 5.607*np.power(x,2) + 16.78*x - 10.61
    return x,y

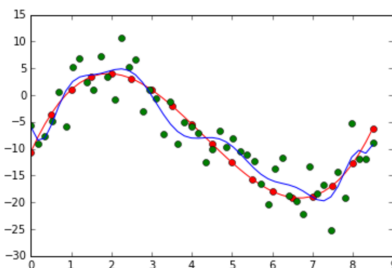
def add_noise(y_curve):
    mu = 0
    sigma=4.0
    noise = np.random.normal(mu, sigma, len(y_curve))
    y_noise = y_curve + noise
    return y_noise
```

3. Fit the generated noisy data using the least squares method based on a polynomial function. You must write two versions of the least squares solution:
(1) $\bar{w} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \bar{t}$
(2) $\bar{w} = \mathbb{X}^+ \bar{t}$ or using numpy's pinv function
4. Show the original curve line, the noisy data, and the curve line estimated from the noisy data, like a figure below:



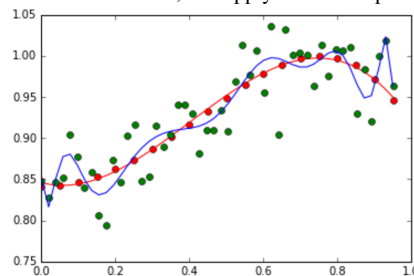
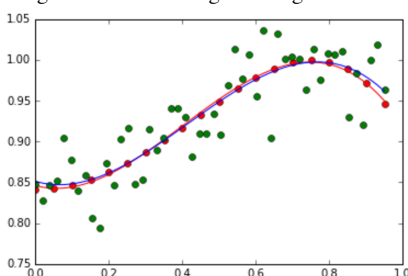
The red line is the original curve based on the equation. The green dots are the noisy data. The blue line is the estimated fitting line, by setting $M = 4$.

5. Compute and display the error value, $E(w)$
6. Display the estimated values of w
7. Experiment with your code by changing M to various values. Plot the values of M against those of $E(w)$, meaning: project them onto a 2D space, where the x-axis is M and the y-axis is $E(w)$. Try to understand the cases of underfitting and overfitting, and how they relate to the error value, $E(w)$



An example of the overfitting problem.

8. Change the noise levels, and apply the same operations above
9. Change the function for generating the data to: $y = \sin(x^2 + 1)$, and apply the same operations as above



In your solutions, aside from the code, you need to include or show the results (as instructed above) in the ipython-notebook format.