Homework 8

Math 87

1 How effective is regression?

In this exercise, you will investigate the quality of an linear least squares regression as a function of the amount of noise in your data.

- To sample from this line, one could uniformly sample x from $[0, N]$ and then apply the transformation $m * x + b$ to each sample, defining a new random variable that samples along the line $y = mx + b$. Since these samples lie exactly on the line, we should only need two samples to reconstruct the line.
- As σ increases, we will expect the required number to approximate the line to increase.
- Below is sample code to solve this problem: The code generates a chosen number of samples (here set to 100000) from the "noisy line" with a prescribed variance (we use $\sigma = 0.1$). Then the code computes the least squares regression for the sampled data. One may then plot the samples and solution lines using matplot.lib.

Sampling

```
import numpy as np
from scipy import optimize
import matplotlib.pyplot as plt
s = np.random.uniform(0,10,100000)
def generategauss(n,sigma):
    eta=np.random.normal(0,sigma,n)
    return eta

M=generategauss(len(s),0.1)

y=3*s+4+M
J=np.ones(len(s))
A=[]
for i in np.arange(len(s)):
    A.append([s[i],1])
np.linalg.solve(
np.matmul(np.transpose(A),A),np.matmul(np.transpose(A),y5))
```

One should see that as σ increases, a greater number of samples is needed to approximate the line correctly.

2 Polynomial curve fitting

Given a set of observations in the form (x_i, y_i), we can represent the problem of fitting an n-degree polynomial as a matrix equation. Let the polynomial be of the form:

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

1. The general matrix equation for fitting the observations to an n-degree polynomial is:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

   Here, m is the number of observations.

2. To test the formula, we can use Python with libraries like NumPy and Matplotlib. Below is an example code snippet:

Fitting

```
import numpy as np
import matplotlib.pyplot as plt

# Given data
x = np.array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
y = np.array([3.66711, 4.06035, 4.41999, 4.69781, 4.82905,
              4.54028, 4.38693, 3.17306, 1.40012, -1.0748])

M=len(x)

degrees = [1, 2, 3, 4, 5]
plt.scatter(x, y, label='Data', color='k')

for degree in degrees:
  # Fit polynomial
  A=np.array([[pow(x[i],k) for k in range(degree+1)]
              for i in range(M)])
  coefficients = np.linalg.lstsq(A,y, rcond=None)[0]
  # Compute predicted values
  y_pred = A@coefficients

  # Compute residual errors
  residuals = y - y_pred

  # Print results
  print(f"Degree {degree} Polynomial Coefficients: {coefficients}")
  print(f"Residual Errors: {residuals}")

  # Plot the data and the fitted polynomial
  plt.plot(x, y_pred, label=f'Degree {degree} Polynomial Fit')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
degree=1	-1.6180	-0.8081	-0.0317	0.6628	1.2107	1.3386	1.6020	0.8048	-0.5514	-2.6097
degree=2	0.4661	-0.1134	-0.3791	-0.3793	-0.1787	-0.0507	0.5599	0.4575	0.1433	-0.5256
degree=3	-0.0488	0.0582	0.0499	0.0007	-0.0316	-0.1978	0.1800	0.0284	-0.0283	-0.0107
degree=4	-0.0069	0.0071	0.0104	0.0077	0.0103	-0.1560	0.1869	-0.0111	-0.0795	0.0311
degree=5	0.0115	-0.0359	0.0135	0.0414	0.0287	-0.1744	0.1532	-0.0141	-0.0365	0.0127

Table 1: Residual error at each point

The resulting residuals are given in table 1 and coefficients are as follows:

degree=1: $a_0 = 5.7018,\ a_1 = -4.1670$

degree=2: $a_0 = 1.8810,\ a_1 = 14.9371,\ a_2 = -17.3673$

degree=3: $a_0 = 3.6339,\ a_1 = -0.6099,\ a_2 = 16.3416,\ a_3 = -20.4297$

degree=4: $a_0 = 3.3014,\ a_1 = 3.6520,\ a_2 = 0.7472,\ a_3 = 0.8795,\ a_4 = -9.6860$

degree=5: $a_0 = 2.8628,\ a_1 = 10.8827,\ a_2 = -37.2112,\ a_3 = 85.7427,\ a_4 = -94.0380,\ a_5 = 30.6735$