Linear Algebra I

lsq.m

```
T = readmatrix("lsq.csv");

% Get the data out of the table.
X = T(:,1);
Y = T(:,2);

% Construct the linear approximation.
P = [ones(size(X)) X];
p = (P'*P)\(P'*Y);

function y = l(x, p)
    y = p(2)*x + p(1);
end

% Construct the quadratic approximation.
A = [ones(size(X)) X X.^2];
b = (A'*A)\(A'*Y);

function y = q(x, b)
    y = b(3)*x.^2 + b(2)*x + b(1);
end

% Find the normal matrix (A^T)A.
B = A.'*A;

scatter(X, Y);
hold on
```

lsq.py

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import inv

D = np.loadtxt("lsq.csv", delimiter=",")
X, Y = D[:,0], D[:,1]

# Linear approximation.
P = np.array([np.ones(len(D)), X]).T
p = inv(P.T @ P) @ (P.T @ Y)

def l(x, p):
  return p[1]*x + p[0]

# Quadratic approximation.
A = np.array([
  np.ones(len(D)), X, X**2
]).T

b = inv(A.T @ A) @ (A.T @ Y)

def q(x, b):
  return b[2]*x**2 + b[1]*x + b[0]

# Find the normal matrix.
B = A.T@A

plt.scatter(X, Y, facecolors="none", edgecolors="tab:blue")
```

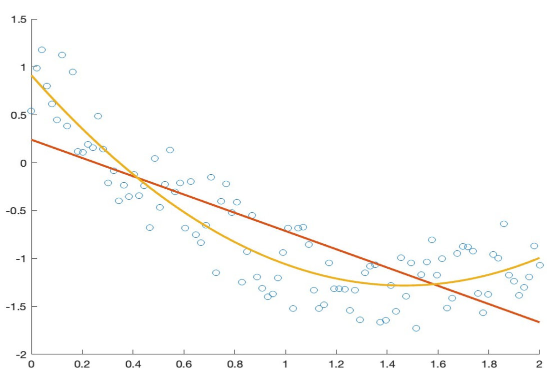
We construct a system of equations called the *normal equations*: given our matrix A, the expression $(A^\top A)^{-1}(A^\top y)$ computes the orthogonal projection from y onto the column span of Ax. Equivalently, we are minimizing $||Ab-y||^2$ — this minimizer is the vector b of squared differences between each entry of y and the closest point in Ax. We're minimizing something quadratic, so it has a unique solution given by

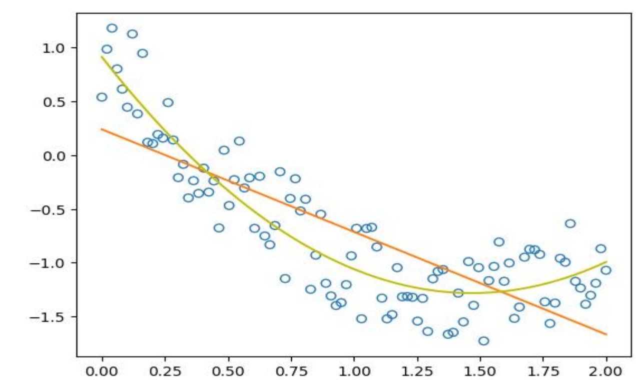
$$\begin{aligned} Ab &= y, \\ A^\top Ab &= A^\top y, \\ b &= (A^\top A)^{-1}(A^\top y). \end{aligned}$$

We can construct A with $n+1$ columns to get a degree-n polynomial fit. In this case, the data above were generated by setting an objective function $\tilde{f}(x) = x^2 - 3x + 1$ and constructing a noised dataset

$$D = \left\{ \left(x_i, \tilde{f}(x_i) \pm \mathcal{N}(0, 1/4) \right) : x_i = 2i/20,\ i \in \{0, \ldots, 20\} \right\},$$

where $\mathcal{N}(0, 1/4)$ is a normal distribution with standard deviation $\sigma = 1/4$. The results look like this:


1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


1.0
0.5
0.0
-0.5
-1.0
-1.5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00