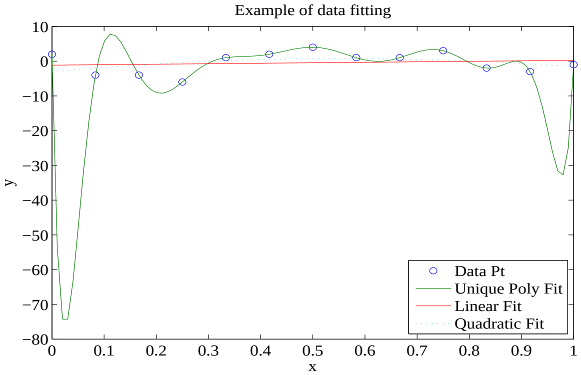
Math 215 Project 2: Least Squares Minimization Problem

Written by: Paul Dostert

Problem Description: Consider the problem of fitting a function to a large data set. If we have N data points in $2D$, then we can clearly fit it uniquely with an $N-1$ degree polynomial. Generally this is not terribly useful, since polynomials of high degree will have large fluctuations. For example, in the below plot we have 13 data points. We fit this with a 12^{th} degree polynomial, a linear polynomial, and a quadratic polynomial. Notice the unique fit (the 12^{th} degree polynomial) approximates the data very poorly in between data points.



To fit a set of data with N points with a polynomial of degree $d \ll N$ we use the technique of least squares. The general idea is that we wish to minimize the sum of the squares of the difference between our fit and the data. Suppose we wish to fit data given by (x_i, y_i), $i = 1, \ldots, N$ with a polynomial of degree d given by $p(x) = p_1 + p_2 x + \cdots + p_{d+1} x^d$. We would like to minimize

$$R = \sum_{i=1}^{N} \left(p(x_i) - y_i\right)^2 .$$

To minimize this, we can take each partial derivative with respect to p_k, set them equal to zero, and solve the resulting equations. We would find:

$$\begin{aligned} \frac{\partial R}{\partial p_k} &= \sum_{i=1}^{N} 2\left(p(x_i) - y_i\right) \cdot x_i^{k-1} = \sum_{i=1}^{N} \left(p(x_i) - y_i\right) \cdot x_i^{k-1} \\ &= \sum_{i=1}^{N} \left(p_1 + p_2 x_i + \cdots + p_{d+1} x_i^d - y_i\right) \cdot x_i^{k-1} \\ &= \sum_{i=1}^{N} \left(p_1 x_i^{k-1} + p_2 x_i^k + \cdots + p_{d+1} x_i^{k-1+d} - y_i x_i^{k-1}\right) = 0 \end{aligned}$$

which we can write as

$$p_1 \sum_{i=1}^{N} x_i^{k-1} + p_2 \sum_{i=1}^{N} x_i^k + \cdots + p_{d+1} \sum_{i=1}^{N} x_i^{k-1+d} = \sum_{i=1}^{N} y_i x_i^{k-1}.$$

We have $d+1$ equations of this form (one equation for each p_k), leading to a matrix equation

$$\begin{bmatrix} \sum_{i=1}^{N} 1 & \sum_{i=1}^{N} x_i & \cdots & \sum_{i=1}^{N} x_i^d \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \cdots & \sum_{i=1}^{N} x_i^{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} x_i^{d-1} & \sum_{i=1}^{N} x_i^d & \cdots & \sum_{i=1}^{N} x_i^{2d} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{d+1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} y_i x_i^1 \\ \vdots \\ \sum_{i=1}^{N} y_i x_i^{d-1} \end{bmatrix}.$$

Solving this matrix equation gives us the coefficients of the polynomial of degree d that best fits the data, in the least squares sense.

Luckily, we can use some linear algebra tricks to make the derivation of this matrix much simpler. Let us think of exactly we would like to have. We want a polynomial of a given degree to EXACTLY fit our data. Now, this is rarely going to happen, but none the less it is what we would love to have. So we want $A\mathbf{p} = \mathbf{y}$ where A is $N \times d$

$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^{d-1} & x_1^d \\ 1 & x_2 & \cdots & x_2^{d-1} & x_2^d \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{N-1} & \cdots & x_{N-1}^{d-1} & x_{N-1}^d \\ 1 & x_N & \cdots & x_N^{d-1} & x_N^d \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_d \\ p_{d+1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix}.$$

Since we have many more equations than unknowns, this will likely be an inconsistent equation. How can we turn this into a consistent equation? We multiply both sides by A^T , since A^TA is a $d \times d$ matrix, and A^Ty will be a size d vector. We have that A^TA is

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ x_1 & x_2 & \cdots & x_{N-1} & x_N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{d-1} & x_2^{d-1} & \cdots & x_{N-1}^{d-1} & x_N^{d-1} \\ x_1^d & x_2^d & \cdots & x_{N-1}^d & x_N^d \end{bmatrix} \begin{bmatrix} 1 & x_1 & \cdots & x_1^{d-1} & x_1^d \\ 1 & x_2 & \cdots & x_2^{d-1} & x_2^d \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{N-1} & \cdots & x_{N-1}^{d-1} & x_{N-1}^d \\ 1 & x_N & \cdots & x_N^{d-1} & x_N^d \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} 1 & \sum_{i=1}^{N} x_i & \cdots & \sum_{i=1}^{N} x_i^d \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \cdots & \sum_{i=1}^{N} x_i^{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} x_i^{d-1} & \sum_{i=1}^{N} x_i^d & \cdots & \sum_{i=1}^{N} x_i^{2d} \end{bmatrix}$$

and

$$A^Ty = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ x_1 & x_2 & \cdots & x_{N-1} & x_N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{d-1} & x_2^{d-1} & \cdots & x_{N-1}^{d-1} & x_N^{d-1} \\ x_1^d & x_2^d & \cdots & x_{N-1}^d & x_N^d \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} y_i x_i^1 \\ \vdots \\ \sum_{i=1}^{N} y_i x_i^{d-2} \\ \sum_{i=1}^{N} y_i x_i^{d-1} \end{bmatrix}.$$

This is the same exact matrix and right hand side from our previous derivation of least squares!! So, in order to solve a least squares problem, we need to only define A and y, then we can calculate everything else using Matlab. The equations in $A^TA\mathbf{p} = A^T\mathbf{y}$ are called the *normal equations* for the problem $A\mathbf{p} = \mathbf{y}$.

Example Problem: Use least squares to approximate the following data by a quadratic polynomial

x	0	$\frac{1}{2}$	1	$\frac{3}{2}$	2	$\frac{5}{2}$	3
y	-2	-1	1	1	0	-2	-1

We define $\mathbf{x} = \begin{bmatrix} 0 \\ \frac{1}{2} \\ 1 \\ \frac{3}{2} \\ 2 \\ \frac{5}{2} \\ 3 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} -2 \\ -1 \\ 1 \\ 1 \\ 0 \\ -2 \\ -1 \end{bmatrix}$. Then $A = \left[\mathbf{x^0} \,|\, \mathbf{x}^1 \,|\, \mathbf{x^2}\right]$. We can then compute (using Matlab)

$$A^TA = \begin{bmatrix} 7 & \frac{21}{2} & \frac{91}{4} \\ \frac{21}{2} & \frac{91}{4} & \frac{441}{8} \\ \frac{91}{4} & \frac{441}{8} & \frac{2275}{16} \end{bmatrix}, \quad A^T\mathbf{y} = \begin{bmatrix} -4 \\ -6 \\ -18.5 \end{bmatrix}.$$

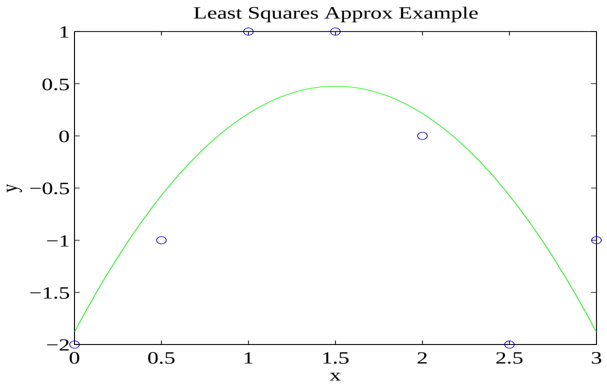
We use Gaussian elimination (the \ command) to solve $A^TA\mathbf{p} = A^T\mathbf{y}$ and get

$$\mathbf{p} \approx \begin{bmatrix} -1.8810 \\ 3.1429 \\ -1.0476 \end{bmatrix}$$

which gives us the polynomial

$$p(x) = -1.8810 + 3.1429x - 1.0476x^2.$$

Plotting the points versus the polynomial (use 100 points to plot the polynomial, not just the 7 points above) we get



Assigned Problem: Use least squares to approximate the following data by a quadratic polynomial. Show relevant Matlab commands, but do not explicitly print out your matrix (put ; after each Matlab command). Write down your quadratic polynomial, and plot it on the same figure as your data.

x	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	1	$\frac{5}{4}$	$\frac{3}{2}$	$\frac{7}{4}$	2	$\frac{9}{4}$	$\frac{5}{2}$	$\frac{11}{4}$	3
y	-5	-3	-1	2	0	-2	1	3	4	3	2	0	-1