Problem 1: Can you fit a line?

Problem Statement

Consider N data points $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, N$ obtained from an experiment , where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$. Let $N > n+1$. The goal of linear regression is to find the "best" linear fit; i.e. find $\mathbf{c} \in \mathbb{R}^n$, $d \in \mathbb{R}$ s.t.:

$$y_i \approx \mathbf{c}^T \mathbf{x}_i + d, \text{ for } i = 1, \ldots, N$$

a

Suppose each measurement is corrupted by independent Gaussian noise with identical variances and 0 mean. Find $\mathbf{A} \in \mathbb{R}^{N \times (n+1)}$ in terms of $\mathbf{x}_1, \ldots, \mathbf{x}_N$ s.t. that the maximum likelihood estimate of $\hat{\mathbf{c}}$, $\hat{d}$ is

$$\begin{pmatrix} \hat{\mathbf{c}} \\ \hat{d} \end{pmatrix} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} \mathbf{A}^T y, \text{ where } \mathbf{y} := \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

b

Consider the case with $n = 1$; i.e. $x_1, \ldots, x_N$ are scalars. Finding the best linear fit amounts to finding $(\hat{c}, \hat{d})$ tht minimizes:

$$J(c, d) := \sum_{i=1}^{N} (y_i - cx_i - d)^2$$

Compute a stationary point $(\hat{c}, \hat{d})$ by setting the derivative of J w.r.t c and d to zero.

c

Use the second derivative tests to conclude that your $(\hat{c}, \hat{d})$ computed in (b) is indeed a local minimizer of J. Can you conclude from this second derivative test alone that it is a global minimizer?

d

Consider the following (x, y) pairs:

- $(1.00, 1.10)$
- $(1.50, 1.62)$
- $(2.00, 1.98)$
- $(2.57, 2.37)$
- $(3.00, 3.23)$
- $(3.50, 3.69)$
- $(4.00, 3.97)$

Draw a scatter plot of these points; then, plot the best linear fit to these points using your formula in part (b).

Solution

a

We can rewrite the given equation to:

$$y_i = \mathbf{x}_i^T \mathbf{c} + d$$

Additionally, let: $\mathbf{1}_N$ be a column vector of 1's with N rows. Thus:

$$\mathbf{y} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} \mathbf{c} + d\mathbf{1}_N$$

We can further re-arrange this to:

$$\mathbf{y} = \begin{pmatrix} \mathbf{x}_1 & 1 \\ \mathbf{x}_2 & 1 \\ \vdots & \vdots \\ \mathbf{x}_N & 1 \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ d \end{pmatrix} \implies \mathbf{A} = \begin{pmatrix} \mathbf{x}_1 & 1 \\ \mathbf{x}_2 & 1 \\ \vdots & \vdots \\ \mathbf{x}_N & 1 \end{pmatrix}$$

b

$$\nabla J(c, d) = \begin{pmatrix} \frac{\partial J}{\partial c} \\ \frac{\partial J}{\partial d} \end{pmatrix} = 2 \begin{pmatrix} \sum_{i=1}^{N} (y_i - cx_i - d)(-x_i) \\ \sum_{i=1}^{N} (y_i - cx_i - d)(-1) \end{pmatrix}$$

Setting equal to zero and noting that the 2 can then just drop out (and push the negative sign around):

$$\mathbf{0} = \begin{pmatrix} \sum_{i=1}^{N} (cx_i + d - y_i)(x_i) \\ \sum_{i=1}^{N} (cx_i + d - y_i) \end{pmatrix}$$

Exanding the top:

$$0 = \sum_{i=1}^{N} (cx_i^2 + dx_i - y_i x_i) = c\left(\sum x_i^2\right) + d\left(\sum x_i\right) - \left(\sum x_i y_i\right)$$

And the bottom:

$$0 = \sum_{i=1}^{N} (cx_i + d - y_i) = c\left(\sum x_i\right) + d\left(\sum 1\right) - \left(\sum y_i\right)$$

Solve for c:

$$c = \left(\left(\sum y_i\right) - d\left(\sum 1\right)\right) / \left(\sum x_i\right)$$

Note that $\sum_{i=1}^{N} 1 = N$. Plug into the first partial derivative equation:

$$0 = \frac{\sum y_i - Nd}{\sum x_i} \sum x_i^2 + d \sum x_i - \sum x_i y_i$$

Solve for d:

$$\frac{\sum x_i^2 \sum y_i - Nd \sum x_i^2}{\sum x_i} + d \sum x_i - \sum x_i y_i$$

$$= \sum x_i^2 \sum y_i - Nd \sum x_i^2 + d \left(\sum x_i\right)^2 - \sum x_i \sum x_i y_i$$

$$Nd \sum x_i^2 - d \left(\sum x_i\right)^2 = \sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i$$

$$d \left(N \sum x_i^2 - \left(\sum x_i\right)^2\right) = \sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i$$

$$d = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{N \sum x_i^2 - \left(\sum x_i\right)^2}$$

which can then be plugged back in to get c. I'm lazy so I used Wolfram to simplify the expression and got:

$$c = \frac{\sum x_i \sum y_i - N \sum x_i y_i}{\left(\sum x_i\right)^2 - N \sum x_i^2}$$

c

Given the gradient, the Hessian is trivially calculable as:

$$\nabla^2 J(c, d) = \begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{pmatrix}$$

If $\nabla^2 J \succcurlyeq 0$ everywhere then J is convex and the first order condition ($\nabla J = 0$) is sufficient for both a global and local optimality. HOWEVER, this will ultimately depend on the x's chosen, as well as the quantity N chosen. For this to hold:

$$\sum x_i^2 + N \geq \sqrt{\sum x_i^2 - 2N \sum x_i^2 + 4 \left(\sum x_i\right)^2 + N^2}$$