Lec20 Applications of Linear Systems.

INTRO: We have focused on solving linear systems. You might wonder why, since systems you are likely to meet in practice are nonlinear. That is true, but often we can develop approaches to such problems that result in the solution of linear eq^{n}s.

Another common scenario is that a linear system has no exact sol^{n}, but we nevertheless want an approx sol^{n}. This problem can be cast as another linear problem (w/ a sol^{n}).

LINEAR REGRESSION Suppose we have a dataset of house prices $\{y_i\}$ and "features" of each house, $\{\vec{x}^{(i)}\}$, that may dictate the house's price.

For example:

i	x_1 size	x_2 # rooms	y price
1	2000	3	500K
2	1000	2	120K
3	2000	4	600K
⋮	⋮	⋮	⋮

n houses

Suppose, further, that we wanted to predict house prices given a house's features $\vec{x}$. How could we do it?

A common approach is to model the relationship between $\vec{x}$ and y in the dataset by a function $h: \mathbb{R}^2 \to \mathbb{R}$ s.t. $y_i = h(\vec{x}^{(i)})$.

In parametric regression, we additionally assume that we know the structure of h ahead of time. For example, suppose that h is linear:

$$h(x) = \theta_1 x_1 + \theta_2 x_2$$

The data in the table can then be written in the form:

$$\left.\begin{aligned}
\theta_1 x_1^{(1)} + \theta_2 x_2^{(1)} &= y^{(1)} \\
\theta_1 x_1^{(2)} + \theta_2 x_2^{(2)} &= y^{(2)} \\
&\vdots \\
\theta_1 x^{(n)} + \theta_2 x_2^{(n)} &= y^{(n)}
\end{aligned}\right\} (*)$$

Contrary to our earlier notation, $Ax=b$, the unknowns here are the θ's, not the x's. Thus:

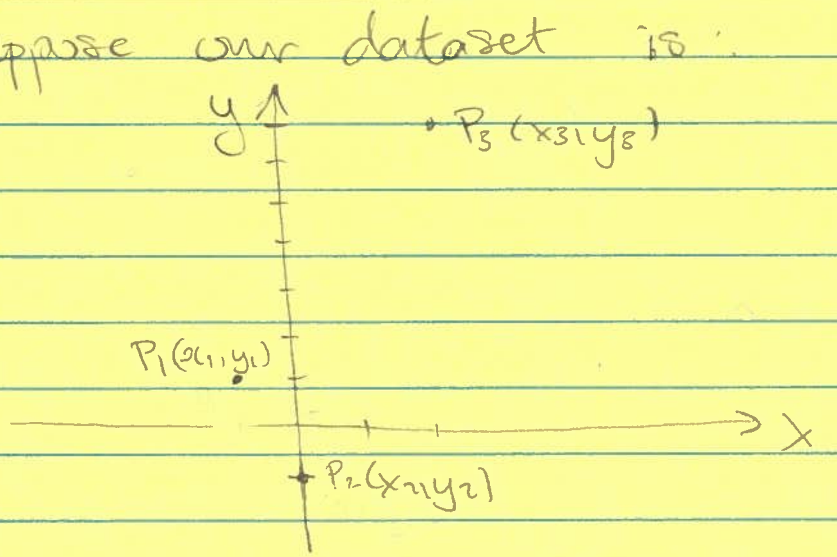
$$\underbrace{\begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \vdots & \\ x_1^{(n)} & x_2^{(n)} \end{bmatrix}}_{\text{we are given these.}} \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}}_{\text{we compute these.}} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} \qquad (**)$$

We have arrived @ a linear system of eq^{n}s.

Of course, all this is predicated on the assumption that $h(\vec{x})$ is linear. But what if it is nonlinear?

Remarkably, it turns out that the process of obtaining the interpolating $h(\vec{x})$ is still a linear problem! Let me illustrate w/ an example.

<u>Example</u> Suppose our dataset is:



We assume $h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$.
Since $h(x)$ must interpolate P_1, P_2, P_3 we have:

$$P_1: \theta_0 + \theta_1(-1) + \theta_2(-1)^2 = 1$$

$$P_2: \theta_0 + \theta_1(0) + \theta_2(0)^2 = -1$$

$$P_3: \theta_0 + \theta_1(2) + \theta_2(2)^2 = 7$$

Notice that, by analogy with (*), there are three "features" for each point (x, y): $1, x, x^2$.

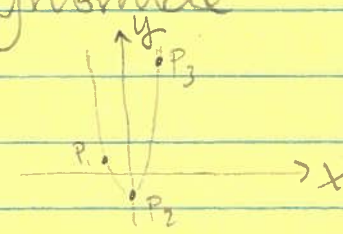
Notice also that, by analogy with (**), we have a linear problem.

$$\begin{array}{c} \text{features} \\ \text{examples} \left\{ \begin{bmatrix} 1 & -1 & (-1)^2 \\ 1 & 0 & 0 \\ 1 & 2 & (2)^2 \end{bmatrix} \right. \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \Theta_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 7 \end{bmatrix} \end{array}$$

... EVEN THOUGH $h(x)$ is nonlinear! The reason, of course, is that $h_{\vec{\Theta}}(x)$ is nonlinear in x, but linear in $\underline{\Theta}$ $= \langle \Theta_0, \Theta_1, \Theta_2 \rangle$. The process of using hypotheses $h_\Theta(x)$ that are linear in Θ is called "linear regression".

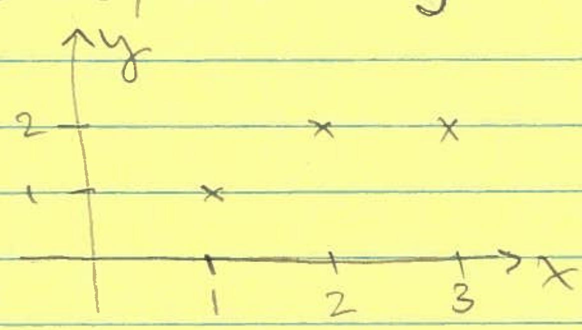
B.T.W. GE applied to this system shows that $\vec{\Theta} = \langle -1, 0, 2 \rangle$, corresponding to the polynomial

$$h(x) = -1 + 2x^2.$$



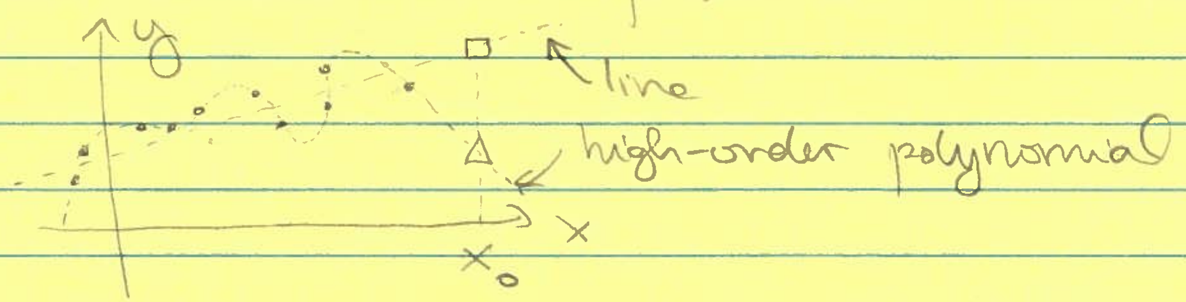
LEAST-SQUARES

In the example above, we were asked to fit a quadratic curve to three points. Clearly this can be done exactly. But what if we were presented w/ the following data:



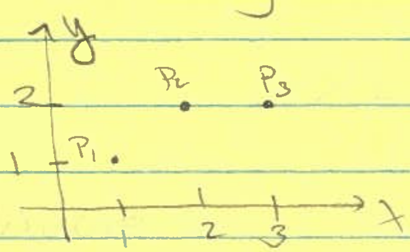
and asked to fit a straight line, $h(x) = \theta_0 + \theta_1 x$. Though no line goes thru' all three pts, there is a best-fit line.

ASIDE: Before we tackle this problem, it is instructive to ask why we are not considering a quadratic function, or indeed any higher-order function capable of passing thru' all 3 pts. The answer lies at the heart of machine learning: we don't want to overfit. An example w/ more data makes the point:



If you were asked to predict y when $x = x_0$, would you use the □ or the △? I think you would use the □ because the data appear to be scattered about the line. Put another way, our hypothesis is that $y = \theta_0 + \theta_1 x + \varepsilon$, where ε is some noise ($\langle\varepsilon\rangle = 0$). Had we sampled a different set of pairs (x_i, y_i), the higher order polynomial would have been very different, but the line wouldn't have changed very much.

Returning to our problem, we have:



P_1: $\theta_0 + \theta_1 1 = 1$
P_2: $\theta_0 + \theta_1 2 = 2$
P_3: $\theta_0 + \theta_2 3 = 2$

or $\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ or $X\theta = y$ (***)

While we cannot solve (***) exactly, we can relax the problem and try to find an approx sol^n $\vec{x}$ satisfying $X\theta \simeq y$.

The most common approach is to minimize:

$$\|X\theta - y\|_2^2 = (X\theta - y)^T (X\theta - y)$$

$$= (\theta^T X^T - y^T)(X\theta - y).$$

$$= \theta^T X^T X\theta - \theta^T X^T y - y^T X\theta - y^T y$$

But

$$(y^T X\theta)^T = \theta^T X^T y$$

$$\Rightarrow y^T X\theta = (\theta^T X^T y)^T \underset{\substack{\uparrow \\ \text{since } y^T X\theta = \text{scalar}}}{=} \theta^T X^T y$$

Thus:

$$\|X\theta - y\|_2^2 = \theta^T X^T X\theta - 2\theta^T X^T y - y^T y \quad (4)$$

This is smallest when its gradient is zero

$$\vec{\nabla}_\theta \|X\theta - y\|_2^2 = 0.$$

$$\underset{(4)}{\Rightarrow} \quad 2X^T X\theta - 2X^T y = 0.$$

$$\Rightarrow \quad \boxed{X^T X\theta = X^T y.} \quad \text{Normal Equations} \quad (5)$$

In our case:

$$X^T X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix}$$

$$X^T y = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

Thus (5) $\Rightarrow$

$$\begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

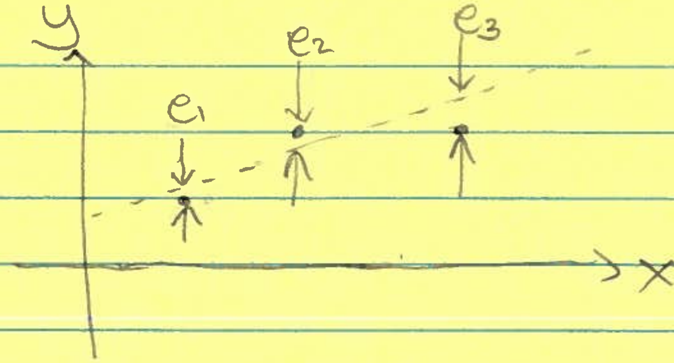
$\underset{GE}{\Rightarrow}$
$$\begin{array}{cc|c} 3 & 6 & 5 \\ 0 & 2 & 1 \end{array}$$

$\underset{\text{Back Sub.}}{\Rightarrow}$
$$2\theta_1 = 1 \Rightarrow \theta_1 = 1/2$$
$$3\theta_0 + 6\theta_1 = 5 \Rightarrow 3\theta_0 = 5 - 6\left(\tfrac{1}{2}\right) = 5 - 3 = 2$$
$$\Rightarrow \theta_0 = 2/3.$$

Recall that this line is the one that minimizes $\|X\theta - y\|_2^2$. But

$$(X\theta)_i = \theta_0 + \theta_1 x_i$$
$$= \text{"predicted" value of } y_i,$$
often denoted $\hat{y}_i$.

Thus:
$$(X\theta - y)_i = \hat{y}_i - y_i = e_i :$$



Thus least-squares tries to minimize the sum of squares of errors:

$$\|X\theta - y\|_2^2 = \sum_i e_i^2 .$$