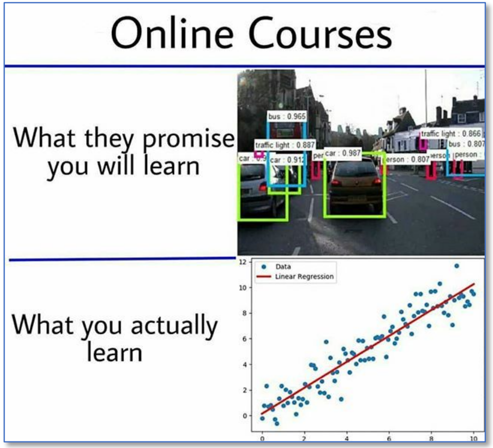
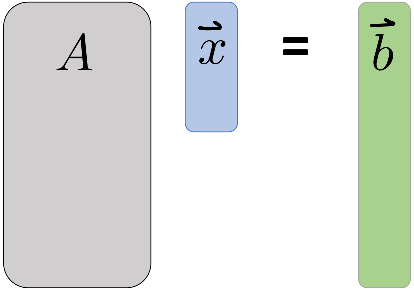


EECS 16A

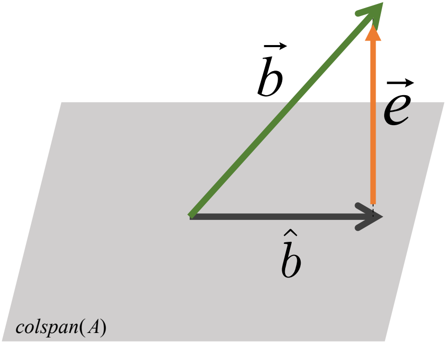
More Least Squares

Last lecture: Least Squares

- Consider the overdetermined linear system:

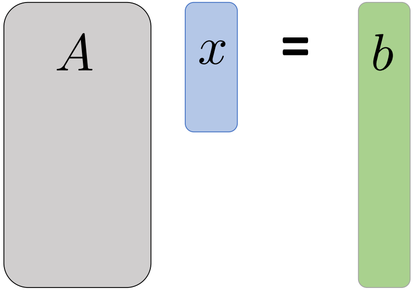


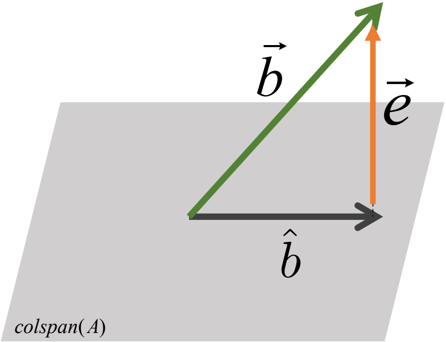
$$\vec{a}_1 x_1 + \vec{a}_2 x_2 + \ldots \vec{a}_n x_n = \vec{b}$$



- If b is not in colspan(A), there is no solution
- the least-squares solution “minimally perturbs” b

Last lecture: Least Squares





$$\min_{\vec{x}} \|\vec{e}\|^2 = \|\vec{b}-\hat{b}\|^2 = \|\vec{b}-\mathbf{A}\hat{x}\|^2$$

$$\Downarrow$$

$$\langle \mathbf{A}^T, \vec{e}\rangle = 0$$

$$\boxed{\vec{\hat{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b}}$$

Least-Squares sol'n

Linear regression

unknowns

known

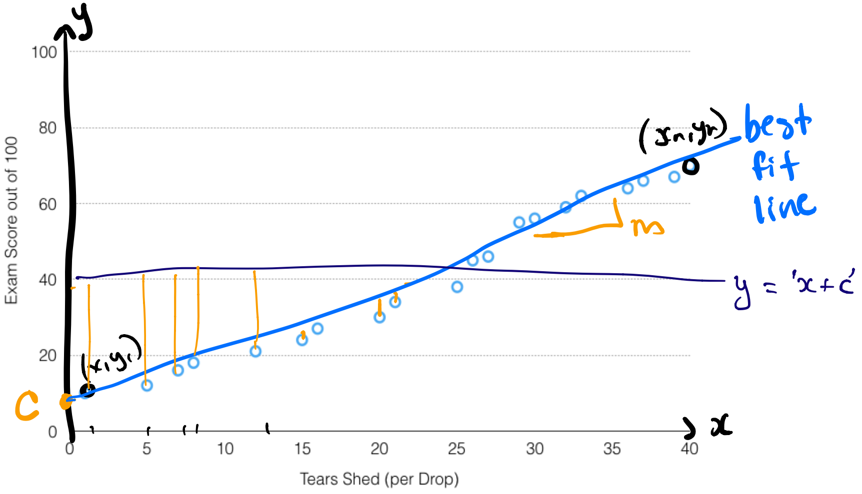
$$y = mx + c$$

have/known: $(x_1, y_1)(x_2, y_2)\dots(x_n, y_n)$

$\hookrightarrow y_1 = mx_1 + c$

$y_n = mx_n + c$

unknowns: $\begin{bmatrix} m \\ c \end{bmatrix} = \vec{w}$



$$\underset{A}{\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \\ x_n & 1 \end{bmatrix}}_{n\times 2}} \; \underset{\vec{w}}{\underbrace{\begin{bmatrix} m \\ c \end{bmatrix}}_{2\times 1}} = \underset{\vec{b}}{\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_{n\times 1}} \checkmark$$

solve for me!

least squares sol'n

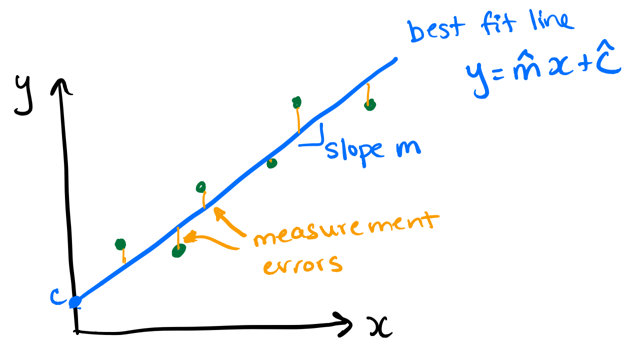
$$\hat{\vec{w}} = (A^T A)^{-1} A^T \vec{b}$$

Least-squares is building block for all of signal processing / machine learning / pattern matching

Example: $\boxed{\text{Linear regression}}$

↳ fit to a line (i.e. project data onto an allowed line)
↳ But, i don't know the line!

$$\underbrace{y}_{\text{known}} = \underbrace{m}_{\text{unknowns}} \underbrace{x}_{\text{known}} + \underbrace{c}_{\text{unknowns}}$$



We measure y for multiple x values:

Known: (x_1, y_1), (x_2, y_2), (x_3, y_3), $\vdots$, (x_n, y_n)

treat as known coeffs. A ; treat as measurements $\vec{b}$

unknown: $\begin{bmatrix} m \\ c \end{bmatrix}$ ← treat as $\vec{w}$ in $A\vec{w} = \vec{b}$

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_{n\times 2}^{A} \underbrace{\begin{bmatrix} m \\ c \end{bmatrix}}_{2\times 1}^{\vec{w}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_{n\times 1}^{\vec{b}}$$

solve for me!

Best estimate for $\hat{\vec{w}} = (A^T A)^{-1} A^T \vec{b}$ least squares sol'n

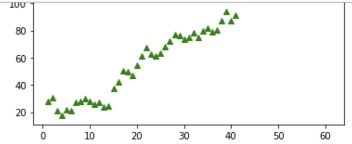
$\boxed{\text{* go to Python notebook demo 'Linear-regression-Facebook.ipynb'}}$

↳ Fits fairly well to line
↳ but that may not hold for extrapolation

Demo: fitting Facebook stock data to a line

Launcher | Linear-regression-Facebook.i | Linear-regression_Snapchat_

Download GitHub Binder Code

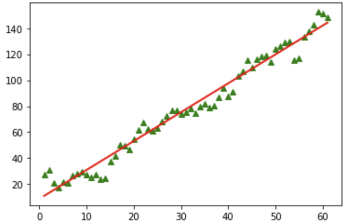



```
[5]: a1 = (fb_stock_data[:,0]).reshape([number_of_data_points,1])
     a2 = np.ones([number_of_data_points,1])
     A = np.hstack((a1,a2))
     b = (fb_stock_data[:,1]).reshape([number_of_data_points,1])
```

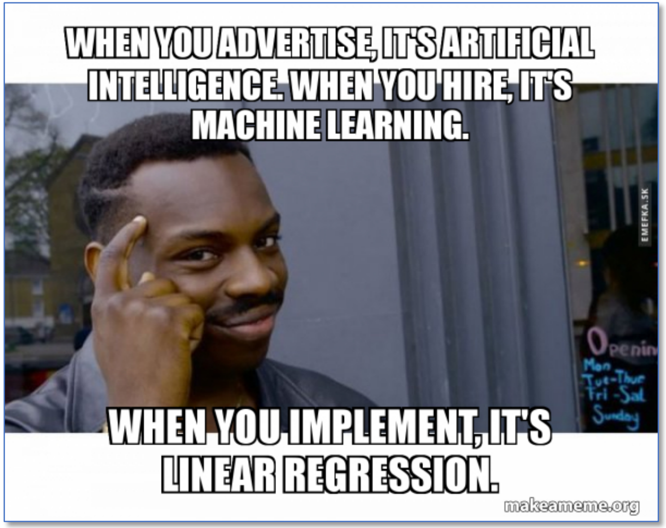
```
[6]: x = least_squares(A,b)
     m = x[0]
     c = x[1]
```

```
[7]: y_new = m*(fb_stock_data[:,0]) + c
```

```
[8]: plt.plot(fb_stock_data[:,0], fb_stock_data[:,1], 'g^')
     plt.plot(fb_stock_data[:,0], y_new, 'r', linewidth=2.0)
     plt.show()
```



Lazy



BUT, not everything fits to a line!?!

Reminder: Most things aren't linear!

Least squares is first attributed to Gauss (1800s)

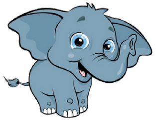
↳ scientist Piazzi tracked a bright spot b/w orbits of Mars & Jupiter, thinking it might be a new planet. (it was Ceres, not a full planet, in asteroid belt)

↳ he missed a few days when he got sick, lost some days due to sun obscured

↳ so he published data, and others tried to calculate future position from existing data

↳ Gauss won the competition by inventing least squares

BUT, not everything fits to a line!?!



"Using a term like nonlinear science is like referring to the bulk of zoology as the study of non-elephant animals." – S. Ulam

How did Gauss find Ceres? fit to Kepler's Laws (elliptical orbits)

$$ax^2 + by^2 + cxy + dx + ey = 1$$ Ellipse eq'n

since squared terms are known → finding coeffs is linear problem

position (knowns!)

coefficients (unknowns!)

How to set up least squares problem?

Let's put unknowns into a vector: $\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}$

Write some equations for measured position (x_1, y_1): $ax_1^2 + by_1^2 + cx_1y_1 + dx_1 + ey_1 = 1$

↳ There are 22 measurements in dataset, so let's put in a matrix:

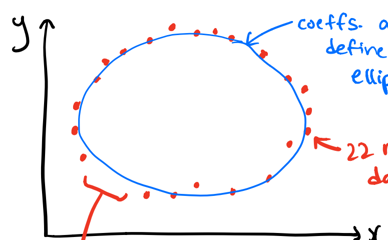
$$\underbrace{\begin{bmatrix} x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 \\ x_2^2 & y_2^2 & x_2y_2 & x_2 & y_2 \\ & & \vdots & & \\ x_{22}^2 & y_{22}^2 & x_{22}y_{22} & x_{22} & y_{22} \end{bmatrix}}_{22 \times 5} \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}}_{5 \times 1} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{22 \times 1}$$

solve for me!

22 equations, 5 unknowns

'Overdetermined'

in machine learning, cols are called 'features'



*Gauss did this by hand! We have Jupyter notebooks so can be lazy ☺ yay!

Demo: 'Ceres-orbit.ipynb'

*go to demo 'Linear-regression_Snapchat_new.ipnb', load 'SNAP.csv'

↳ the fit gets better for higher-order polynomials (y_new = predict_with_degree(1))

↳ BUT, you might be "fitting to noise"

↳ want to use the simplest model that most accurately represents the real world! KISS!

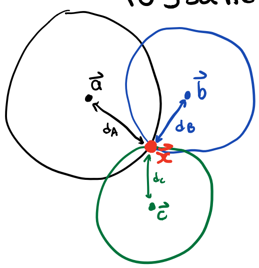
"Everything should be made as simple as possible, and no simpler." - Einstein?

↳ this is why Gauss' example worked so well → Keppler's laws were correct

Let's go back to least squares for trilateration:

Recall:

Trilateration find my coordinates $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ from known distances d_A, d_B, d_C to 3 sattelites with known positions $\vec{a}, \vec{b}, \vec{c}$



$$\|\vec{x}-\vec{a}\|^2 = d_A^2$$
$$\|\vec{x}-\vec{b}\|^2 = d_B^2$$
$$\|\vec{x}-\vec{c}\|^2 = d_C^2$$

$$\underbrace{\begin{bmatrix} -2a_1+2b_1 & -2a_2+2b_2 \\ -2a_1+2c_1 & -2a_2+2c_2 \end{bmatrix}}_{\substack{\text{knowns} \\ A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\substack{\text{to solve for} \\ \vec{x}}} = \underbrace{\begin{bmatrix} d_A^2 - d_B^2 - \|\vec{a}\|^2 + \|\vec{b}\|^2 \\ d_A^2 - d_C^2 - \|\vec{a}\|^2 + \|\vec{c}\|^2 \end{bmatrix}}_{\substack{\text{knowns} \\ \vec{b}}}$$

Linear system

Now we can use least squares: $\hat{x} = \underbrace{(A^TA)^{-1}}_{} A^T \vec{b}$

How do we know if $\underbrace{(A^TA)}_{\text{call this } Q}{}^{-1}$ exists?

Recall: invertible → trivial Null space → lin. ind. cols!

Matrix Q is invertible if and only if $Null(Q) = \vec{0}$ (the nullspace is trivial)

Theorem: $Null(A^TA) = Null(A)$

Recall: If $\|\vec{x}\| = 0$, then $\vec{x} = \vec{0}$

Proof: $\|\vec{x}\| = 0$

$\|\vec{x}\|^2 = 0$

$x_1^2 + x_2^2 + \dots x_n^2 = \langle \vec{x}, \vec{x} \rangle = 0$

all terms ≥ 0

so all terms $= 0$ → so $\vec{x} = \vec{0}$! ✓☺

Recall: Properties of transposes

$$(AB)^T = B^TA^T$$

$(n\times m \; m\times k)^T$ → $(n\times k)^T$ → $k\times n$; $k\times m \;\; m\times n$ → $k\times n$ ✓☺

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}$$

$$(AB)^T = \begin{bmatrix} \overset{\alpha}{a_{11}b_{11} + a_{12}b_{21}} \\ \underset{\beta}{a_{21}b_{11} + a_{22}b_{21}} \end{bmatrix}^T = \begin{bmatrix} \alpha & \beta \end{bmatrix}$$

$$B^TA^T = \underset{1\times 2}{\begin{bmatrix} b_{11} & b_{21} \end{bmatrix}} \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}}$$

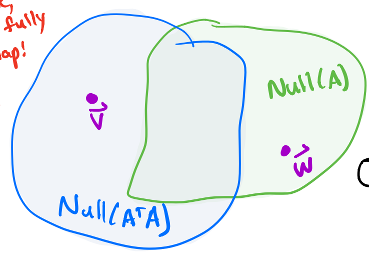
$$= \underset{1\times 2}{\begin{bmatrix} \underset{\alpha}{(b_{11}a_{11} + b_{21}a_{12})} & \underset{\beta}{(b_{11}a_{21} + b_{21}a_{22})} \end{bmatrix}} = \begin{bmatrix} \alpha & \beta \end{bmatrix}$$

Same! ✓☺

Now, back to proof...

$\text{Null}(A^TA) = \text{Null}(A)$

if true, these fully overlap!



① some vector $\vec{w} \in \text{Null}(A)$, then it should be in $\text{Null}(A^TA)$

② some vector $\vec{v} \in \text{Null}(A^TA)$, then it should be in $\text{Null}(A)$

① Known: $\vec{w} \in \text{Null}(A)$ | Want: $\vec{w} \in \text{Null}(A^TA)$

$A \cdot \vec{w} = \vec{0}$ | $(A^TA) \cdot \vec{w} = \vec{0}$

$\times A^T$

$A^T(A \cdot \vec{w}) = A^T\vec{0}$

$(A^TA)\vec{w} = \vec{0}$

$\vec{w}$ belongs to $\text{Null}(A^TA)$! ☺ ✓

more tricky! (optional)

② Known: $\vec{v} \in \text{Null}(A^TA)$ | want: $\vec{v} \in \text{Null}(A)$

$A^TA\vec{v} = \vec{0}$ | $A\vec{v} = \vec{0}$

can't divide by A^T to get rid of it

Consider
$$\begin{aligned}
\|A\vec{v}\|^2 &= \langle A\vec{v}, A\vec{v}\rangle \\
&= (A\vec{v})^T(A\vec{v}) \\
&= \vec{v}^TA^T(A\vec{v}) \\
&= \vec{v}^T(A^TA\vec{v}) \\
&= \vec{v}^T(\vec{0})
\end{aligned}$$

$\|A\vec{v}\|^2 = \vec{0} \longrightarrow \|A\vec{v}\| = \vec{0} \longrightarrow A\vec{v} = \vec{0}$ ✓ ☺