My Hobby: Extrapolating

As you can see, by late next month you’ll have over four dozen husbands. Better get a bulk rate on wedding cake.

EECS 16A
Least Squares Algorithm
Last lecture: Trilateration

Finding my 2D position by calculating distances to 3 satellites with known positions:
Case 1: No noise gives unique solution

Correct measurements:
- B1: 5m
- B2: 5m
- B3: 5m
- B4: 5m

Least squares estimate: (0,0)
Case 2: Noisy measurements

B1: 5.3m  
B2: 4.5m  
B3: 5.1m  
B4: 4.8m

All measurements have error: Least squares estimate:  
(-0.28, 0.26)  
Estimate has some error, but will get smaller with more measurements (if error is random)
Case 3: Some noisy measurements

All measurements have error:

- B1: 6.8m
- B2: 5m
- B3: 5m
- B4: 5m

Least squares estimate:

(0, 1.04)

Error is not spread evenly (random), if I knew 3 were correct, I would have gotten answer correct...
Trilateration:

Let's find my coordinates in 2D world, \( \mathbf{z} = [z_1, z_2] \), from known distances \( d_A, d_B, d_C \) to 3 satellites with known positions \( \mathbf{a}, \mathbf{b}, \mathbf{c} \).

\[
\begin{align*}
||\mathbf{z} - \mathbf{a}||^2 &= d_A^2 \\
||\mathbf{z} - \mathbf{b}||^2 &= d_B^2 \\
||\mathbf{z} - \mathbf{c}||^2 &= d_C^2
\end{align*}
\]

3 equations, 2 unknowns. Problem: not linear!

Some math tricks:

\[
\begin{pmatrix}
-2a_1 + 2b_1 \\
-2a_2 + 2b_2 \\
-2c_1 + 2c_2
\end{pmatrix}
= \begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
\]

2 equations, 2 unknowns. Linear 😊

Now we want to solve this \( A\mathbf{z} = \mathbf{b} + \mathbf{e} \) problem in the presence of measurement noise, and possibly for many satellite measurements (>3):

\[
A\mathbf{z} = \mathbf{b} + \mathbf{e}
\]

Error due to noise (unknown)

\( A\mathbf{z} = \mathbf{b} \) might have more equations than unknowns:

\[
\begin{pmatrix}
A & \mathbf{b}
\end{pmatrix}
\]

Overdetermined System

\( A^T A \) least-squares solve effectively does averaging

Least Squares Algorithm → finds the best estimate \( \hat{\mathbf{z}} \) such that \( A\hat{\mathbf{z}} \) is as close as possible to \( \mathbf{b} \) (i.e. minimizes \( \mathbf{e} \))

Want \( \min \| \mathbf{e} \|^2 = \| \mathbf{b} - \hat{\mathbf{b}} \|^2 = \| \mathbf{b} - A\hat{\mathbf{z}} \|^2 = \| A\hat{\mathbf{z}} - \mathbf{b} \|^2 \)

Estimate of \( \mathbf{z} \)

Solution will be given by projection of \( \mathbf{b} \) onto \( \mathbf{a} \) (for 2D case)

\( \hat{\mathbf{b}} = (\mathbf{a}^T \mathbf{a}) \mathbf{a} \)

Find \( \mathbf{a} \)!

Key idea in ML/ISP, used for classification, etc.

Projections → find the component along a particular direction

What does it have to do with inner product?

Looking for collinear component (largest inner product)

or perpendicular component (smallest inner product)

Orthogonal
project \( \mathbf{b} \) onto subspace spanned by \( \mathbf{a} \) 

The projected vector \( \hat{\mathbf{b}} \) is collinear with \( \mathbf{a} \) \( (\hat{\mathbf{b}} = \alpha \mathbf{a}) \) and perpendicular to \( \mathbf{e} = \mathbf{b} - \hat{\mathbf{b}} \) \( (\mathbf{e} \perp \hat{\mathbf{b}}, \mathbf{e} \perp \mathbf{a}) \) 

Using the property that perpendicular vectors have 0 inner product: 
\[
\langle \mathbf{e}, \mathbf{a} \rangle = 0 \\
\langle \mathbf{b} - \hat{\mathbf{b}}, \mathbf{a} \rangle = 0 \\
\langle \mathbf{b}, \mathbf{a} \rangle - \langle \hat{\mathbf{b}}, \mathbf{a} \rangle = 0 \\
\langle \mathbf{b}, \mathbf{a} \rangle = \langle \alpha \mathbf{a}, \mathbf{a} \rangle \\
\langle \mathbf{b}, \mathbf{a} \rangle = \alpha \langle \mathbf{a}, \mathbf{a} \rangle \\
\langle \mathbf{b}, \mathbf{a} \rangle = \alpha \frac{\mathbf{b} \cdot \mathbf{a}}{\| \mathbf{a} \|^2}
\]

\[
\hat{\mathbf{b}} = \frac{\langle \mathbf{b}, \mathbf{a} \rangle}{\| \mathbf{a} \|^2} \mathbf{a}
\]

**Theorem:** Consider matrix \( A \), vector \( \mathbf{y} \in \text{colspan}(A) \). Then, consider vector \( \mathbf{z} \) 
\[
\begin{align*}
\langle \mathbf{z}, \mathbf{a}_1 \rangle &= 0 \\
\langle \mathbf{z}, \mathbf{a}_2 \rangle &= 0 \\
&\vdots \\
\langle \mathbf{z}, \mathbf{a}_n \rangle &= 0
\end{align*}
\]
\( \mathbf{z} \) is orthogonal to all vectors in \( \text{colspan}(A) \) 

\[
\begin{align*}
\mathbf{y} &= c_1 \mathbf{a}_1 + c_2 \mathbf{a}_2 + \ldots + c_n \mathbf{a}_n \\
= \mathbf{A} \mathbf{c} \\
\end{align*}
\]

**Proof:** we know \( \mathbf{y} \in \text{colspan}(A) \), so it's a lin.combo. of cols: 

\[
\mathbf{y} = c_1 \mathbf{a}_1 + c_2 \mathbf{a}_2 + \ldots + c_n \mathbf{a}_n
\]

we want \( \langle \mathbf{z}, \mathbf{y} \rangle = 0 \) 

\[
\begin{align*}
\langle \mathbf{z}, \mathbf{y} \rangle &= \langle \mathbf{z}, c_1 \mathbf{a}_1 + c_2 \mathbf{a}_2 + \ldots + c_n \mathbf{a}_n \rangle \\
&= c_1 \langle \mathbf{z}, \mathbf{a}_1 \rangle + c_2 \langle \mathbf{z}, \mathbf{a}_2 \rangle + \ldots + c_n \langle \mathbf{z}, \mathbf{a}_n \rangle \\
&= c_1 \langle \mathbf{z}, \mathbf{a}_1 \rangle + c_2 \langle \mathbf{z}, \mathbf{a}_2 \rangle + \ldots + c_n \langle \mathbf{z}, \mathbf{a}_n \rangle = 0 \quad \text{yay!}
\end{align*}
\]
Ok, but we need to find $\hat{z}$ from $\hat{b} = A\hat{z}$

Least Squares: minimize $\|A\hat{z} - \hat{b}\| = \|\hat{e}\|$

First, write $A$ in terms of column view $A = \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \cdots & \hat{a}_n \end{bmatrix}$

$A\hat{z}$ is in $\text{col}(A)$

Let search for $\hat{b} = A\hat{z}$ ← should be in $\text{col}(A)$, even tho $\hat{b}$ is not

$\hat{b} + \hat{e} = \hat{b}$, $\hat{e} = \hat{b} - \hat{b}$

Sine $\hat{e} \perp \text{col}(A)$:

\[
\begin{align*}
\langle \hat{a}_1, \hat{e} \rangle &= 0 \\
\langle \hat{a}_2, \hat{e} \rangle &= 0 \\
&\vdots \\
\langle \hat{a}_n, \hat{e} \rangle &= 0 \\
\end{align*}
\]

\[
\begin{bmatrix}
\hat{a}_1^T \\
\hat{a}_2^T \\
\vdots \\
\hat{a}_n^T
\end{bmatrix}
\begin{bmatrix}
\hat{e}
\end{bmatrix}
= 0
\]

Write in matrix form:

\[
\begin{bmatrix}
-\hat{a}_1^T \\
-\hat{a}_2^T \\
\vdots \\
-\hat{a}_n^T
\end{bmatrix}
\begin{bmatrix}
\hat{b}
\end{bmatrix}
= 0
\]

\[
A^\top (\hat{e} - \hat{b}) = 0
\]

\[
A^\top \hat{e} = A^\top A \hat{z} = 0
\]

Sometimes we want $\hat{b} = A\hat{z}$

\[
\hat{b} = A (A^\top A)^{-1} A^\top \hat{b}
\]

Least Squares Solution!

\[
\hat{z} = (A^\top A)^{-1} A^\top \hat{b}
\]

Example:

\[
\begin{bmatrix}
2 & 1 \\
\end{bmatrix}
\begin{bmatrix}
2 \\
1
\end{bmatrix}
= \begin{bmatrix}
3 \\
1
\end{bmatrix}
\]

If we did Gauss Elim:

\[
\begin{bmatrix}
2 & 1 \\
0 & 1/2
\end{bmatrix}
\]

Inconsistent

(no soln)

So lets find best estimate $\hat{z}$ by least squares:

\[
\hat{z} = (A^\top A)^{-1} A^\top \hat{b}
\]

\[
= (\frac{1}{2}) \begin{bmatrix}
2 & 1
\end{bmatrix} \begin{bmatrix}
3 \\
1
\end{bmatrix}
\]

\[
= (\frac{1}{2}) \cdot 3 \\
= 3/2
\]

Dr, note that $\hat{b} \notin \text{col}(A)$

\[
A^\top = \begin{bmatrix}
2 & 1
\end{bmatrix}
\]

\[
A^\top A = \begin{bmatrix}
2 & 1
\end{bmatrix} \begin{bmatrix}
2 & 1
\end{bmatrix}
= 5
\]

\[
(A^\top A)^{-1} = \frac{1}{5}
\]
Example: \( A\hat{x} = \hat{b} \)
\[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
1 \\
2 \\
3
\end{bmatrix}
\]
\( x_1 = 1 \)
\( x_2 = 2 \)
\( x_3 = 3 \)

Least-squares Algorithm:
\[
\hat{x} = (A^TA)^{-1}A^T\hat{b}
\]
\[
= \begin{bmatrix}
1 & 0 \\
0 & 1/2
\end{bmatrix}
\begin{bmatrix}
1 \\
3/2
\end{bmatrix}
= \begin{bmatrix}
1 \\
0.5
\end{bmatrix}
\]

\( \hat{x} = \begin{bmatrix}
2.5
\end{bmatrix} \)

Least squares is first attributed to Gauss (1800s)

A 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 obscural

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

Gauss won the competition by inventing least squares

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 \((x, y)\):
\[a x^2 + b y^2 + c x y + dx + ey = 1\]

There are 22 measurements in dataset, so let's put in a matrix:
Least-squares is building block for all of signal processing/machine learning/pattern matching.

Linear regression

\[ y = mx + c \]

Known: \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\)

Unknown: \(\begin{bmatrix} m \\ c \end{bmatrix}\)

\[
\begin{pmatrix}
  x_1 & \cdots & x_n \\
  y_1 & \cdots & y_n \\
\end{pmatrix}
\begin{pmatrix}
  m \\
  c \\
\end{pmatrix}
= \begin{pmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_n \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
  \hat{w} \\
\end{pmatrix} = (A^T A)^{-1} A^T b
\]

Best estimate for \(\hat{w}\) using least squares.

22 equations, 5 unknowns ('Overdetermined')

\(\star\) Gauss did this by hand!
We have Jupyter notebooks so can be lazy 😊 yay!
(see slides or 'Ceres-orbit' notebook)