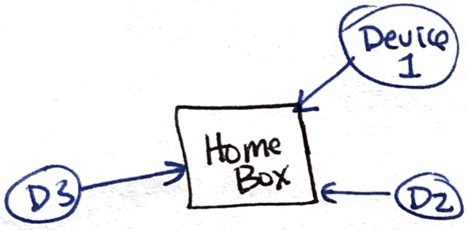
M3-Lec 6

Last time:
- IOT problem setup
- OMP algorithm development ← This time!

IOT → Internet of Things.



New: at most k devices 'ON' at once

→ may not care what shifts are (if not locationing)

→ but want value of song strength from each device → "message"

e.g. Temperature Value from smart thermometer

encoded by $\vec{s} \rightarrow \alpha \vec{s}$ ← temp.

Looked @ example situation w $\boxed{2{,}000}$ devices (n) → each has unique song $\vec{s}_0, \vec{s}_1, \ldots \vec{s}_{1999}$ ← device ID

of length $\boxed{400}$ (m) samples → $\vec{s}_i \in \mathbb{R}^{400}$ ← dim of song

Want to: ① Identify 'on' devices
② Estimate message values

Recieved signal at homebox is sum of shifted & scaled songs. How to find them? cross-corr., but in particular way.

Recap of OMP:

Unknowns: $\vec{x} \in \mathbb{R}^{n}$

$\boxed{m<n}$

Songs: $S = \{\underbrace{\vec{s}_0, \vec{s}_1, \ldots \vec{s}_{n-1}}_{\text{all the songs}}\} \in \mathbb{R}^m$

device ID

Messages: $\alpha_0, \alpha_1, \ldots \alpha_{n-1}$ ← but a lot of these are zero

Sparsity: K ← max. # devices talking @ once

↳ device IDs for 'ON' devices: $i_1, i_2, \ldots i_k$

↳ shifts for 'ON' devices: $N_{i1}, N_{i2}, \ldots N_{ik}$

Measurement: (received signal @ Home Box) $\vec{b} \in \mathbb{R}^{m}$

$$\vec{b} = \sum_i \underset{\text{msg}}{\alpha_i} \underset{\text{song}}{\vec{s}_i}^{(N_i)} + \underset{\text{noise}}{\vec{e}}$$

shift

When to use OMP? to recover estimate of a 'sparse' unknown signal (mssgs)

#unknowns > #measurements

Let's us recover answer from fewer meas. than unknowns!

Why? sparsity is a 'prior' that adds extra information..sort

What do I need? songs are all approx. orthogonal.

ithm development:

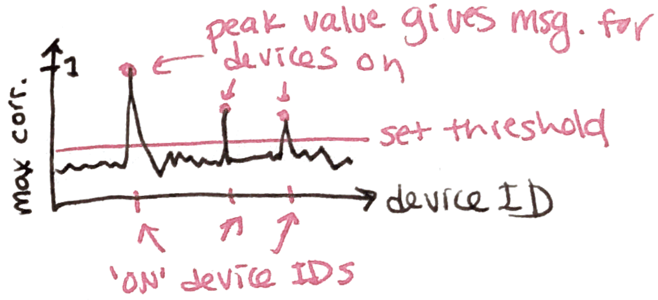
- this is a stretch for 16A (i.e. hard!)
- ~~dont~~ helps understand algorithm to develop it step-by-step.
- also, shows you how to think to design your own algorithms in future
  - ↳ cause you are leaders, don't want to just use existing algos, but to invent better ones!

Algo V1. 2, 3 → cross-corr $\vec{b}$ w all songs

→ pick out peaks to find 'on' devices

CONS: non-orthog. means small msg. gets drowned by big one! can't find small stuff.

Fix: subtract out one at a time.



Problem: Noise

Fix: use LS to solve

$$\underset{\text{meas}}{\vec{b}} = \underbrace{\overbrace{\begin{bmatrix} | & | & & | \\ \vec{z}_1 & \vec{z}_2 & \cdots & \vec{z}_k \\ | & | & & | \end{bmatrix}}^{Z}}_{\text{set of shifted songs}} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix}$$

messages

SOLVE FOR ME!

where $\vec{z}_k = \vec{S}_{ik}^{(N_{ik})}$

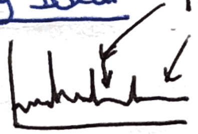
correctly shifted song

by LS: $\boxed{\hat{\alpha} = (Z^T Z)^{-1} Z^T \vec{b}}$

LS solution for msgs

'pealing' songs off one-by-one

Key Idea: This 'grass' between peaks is not necessarily noise, but rather due to non-orthogonality! (if orthogonal → zero, no problem!)



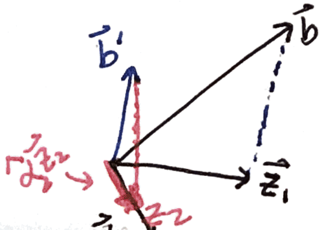
To subtract out loudest song: → pick largest peak i_1 device ID & value α_{i_1} of msg.

→ 'residue' → $\vec{b}' = \vec{b} - \alpha_{i_1} \vec{z}_1$ what's left of meas. after effect of 1st device is removed

→ then do this over and over until done!

① proj. $\vec{y}$ onto $\vec{z}_1$

② then proj. $\vec{b}'$ onto $\vec{z}_2$



CONS: if $\vec{z}_1, \vec{z}_2$ (songs) not orthogonal, can't just do one at a time & add components

Fix: need to project onto entire subspace of $\vec{z}_1, \vec{z}_2$! (subtract both together!)

us us finally to Orthogonal Matching Pursuit $\boxed{\text{(OMP) Algorithm:}}$

→ at each step, keep track of identified 'ON' songs (+ shifts)

then remove effect of 'found' devices & repeat...

e.g. $A = \begin{bmatrix} | & | \\ \vec{z}_1 & \vec{z}_2 \\ | & | \end{bmatrix}$, solve $\vec{b} = A\hat{\vec{\alpha}}$ using least squares:

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

at each iter. add a new col from 'found' loudest song

project meas. onto entire 'known' (so far) subspace to est. msg.

What if less devices (<K) are on?

↳ Stop early by checking if less than some small threshold value.

So, Find peaks one-by-one, but remove all together at each step!

ONP Algo pseudocode:

songs, meas, sparsity, threshold for stopping

$$\text{OMP}\,(\underbrace{s, \vec{b}, k, th}_{\text{inputs}})\ \{$$

index of iteration

init w zeros

Initialization: $\vec{y} = \vec{b}$ start w meas. $F = [\text{empty}]$ Found Devices IDs $A_0 = [\ ]$ Songs found so far $j = 1$ iter # $\vec{x} = \vec{0}$ Device messages (SOLUTION)

Algorithm: $\text{while } (j \leq k) \text{ and } \underbrace{(\|\vec{y}\| > th)}_{\text{threshold to stop}}\ \{$

device ID, shift

$$\underbrace{[i, N_i]}_{\text{outputs}} = \underset{\text{function}}{\text{find-song}}\,(\underbrace{s, \vec{y}}_{\text{inputs}})$$

some fcn. that finds loudest song

Newly discovered: $\vec{z}_j = \vec{s}_i^{\,(N_i)}$ Song of identified device w proper shift. Does it have to be shifted? Yes, to remove properly.

union

$F = F \cup \{i\}$ add identified device to list of ON devices

'Augment' matrix $A_j = [A_{j-1} \,\vdots\, \vec{z}_j]$ add new song to set

some proxy to $\vec{x}$ $\vec{\alpha}_j = [A_j^TA_j]^{-1}A_j^T\vec{b}$ projection of meas. onto subspace of all the songs we know on so far...

.. meas. that is EXPLAINED: $\vec{b}_j = A_j a_j = A_j \left[A_j^T A_j \right]^{-1} A_j^T \vec{b}$

part of meas. that is NOT EXPLAINED: $\vec{y} = \vec{b} - \vec{b}_j$

When does $\vec{b}_j = \vec{b}$? When I found everything!

So $\vec{y} \to 0$ and $\| \vec{y} \| < th$

threshold test for stopping

```
    $\vec{x}[F] = a_j$     update msg. values for 'on' devices
                           with their current estimates. (rest stay zero)
    $j = j + 1$   }        update iteration counter.
```

Am I done? Yes! $\vec{x}$ is solution! Yay!

But what was 'find_song' function? Finds device ID & shift of loudest song.

```
find_song($S$, $\vec{y}$) {
    max_correl = zeros (# songs, 1)  }  initialize
    shifts = zeros (# songs, 1)      }
               ← iter index
    for ($i = 1$ : # songs)
               from 1 to # songs
        $\rho$ = corr ($\vec{y}$, $\vec{s}_i$)
                                  ↙ pick out max corr (for each song)
        max_correl[i] = max(abs($\rho$))
        shifts[i] = argmax(abs($\rho$))  ← find it's shift

    ~~j = argmax(m~~
    $j$ = argmax (max_correl)
    $N_j$ = shifts[j]
}
```

return these as outputs

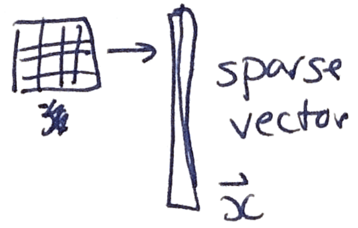
What if msg = −1? Large negative corr., so just use abs to ignore sign!

Iteration	Song Found	A_i	Explained	Unexplained	Measurement
0	[ ]	[ ]	$\vec{0}$	$\vec{b}$	$\vec{b}$
1	$\vec{z}_1$	$A_1 = [\vec{z}_1]$	$\vec{b}_1 = A_1(A_1^TA_1)^{-1}A_1^T\vec{b}$	$\vec{b} - \vec{b}_1$	$\vec{b}$
2	$\vec{z}_2$	$A_2 = \begin{bmatrix} \vert & \vert \\ \vec{z}_1 & \vec{z}_2 \\ \vert & \vert \end{bmatrix}$	$\vec{b}_2 = A_2(A_2^TA_2)^{-1}A_2^T\vec{b}$	$\vec{b} - \vec{b}_2$	$\vec{b}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
j	$\vec{z}_j$	$A_j = \begin{bmatrix} \vert & \vert & & \vert \\ \vec{z}_1 & \vec{z}_2 & \cdots & \vec{z}_j \\ \vert & \vert & & \vert \end{bmatrix}$	$\vec{b}_j = A_j(A_j^TA_j)^{-1}A_j^T\vec{b}$	$\vec{b} - \vec{b}_j$	$\vec{b}$

Recall: Imaging Lab

$$A\underset{\uparrow}{\vec{x}} = \vec{b}$$

Now say $\vec{x}$ is sparse image



$\boxed{\text{What can OMP do for you here?}}$

We can take fewer measurements!

But how to make orthogonal?$^{\nearrow \text{ish}}$ does it work?