EECS16A Summer 2020 Review Session: OMP

August 12, 2020

Moses Won

Concept Dependencies

Least Squares

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

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

Inner Product and properties

$$\langle\vec{v},\vec{w}\rangle = \vec{v}^T\vec{w} = ||\vec{v}||||\vec{w}||\cos\theta$$

What's OMP?

Orthogonal Matching Pursuit (OMP) is a ***solution finding algorithm***.

For the problem: $$A\vec{x} = \vec{b}$$

Finds a $\hat{x}$ such that $A\hat{x} \approx \vec{b}$

What can OMP solve?

$$\mathbf{A}\hat{x} = \vec{b}$$

OMP addresses problems where the equation typically has ***wide matrix*** and the ***solution is sparse***. Wide matrices are guaranteed to have linearly dependent columns => Can’t use least squares directly.

Why Orthogonal *Matching Pursuit*

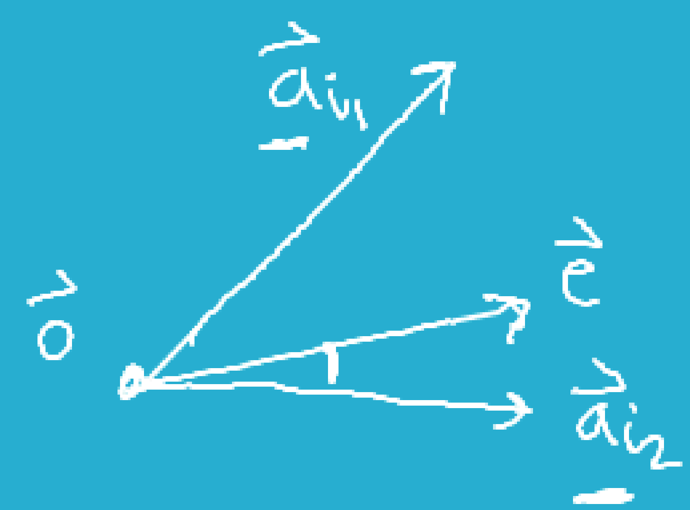
OMP matches or approximates $\vec{b}$ with the columns of $\mathbf{A}$ with a ***greedy*** strategy to make the error as ***small as possible*** as quickly as possible.

Quickly/Greedy: Choose $\vec{a}_i$ (column) with largest $|\langle \vec{a}_i, \vec{e} \rangle|$

Small Error: Minimize $||\vec{e}|| = ||\mathbf{A}\hat{x} - \vec{b}||$

Why Orthogonal *Matching Pursuit*

Vector picture of similarity

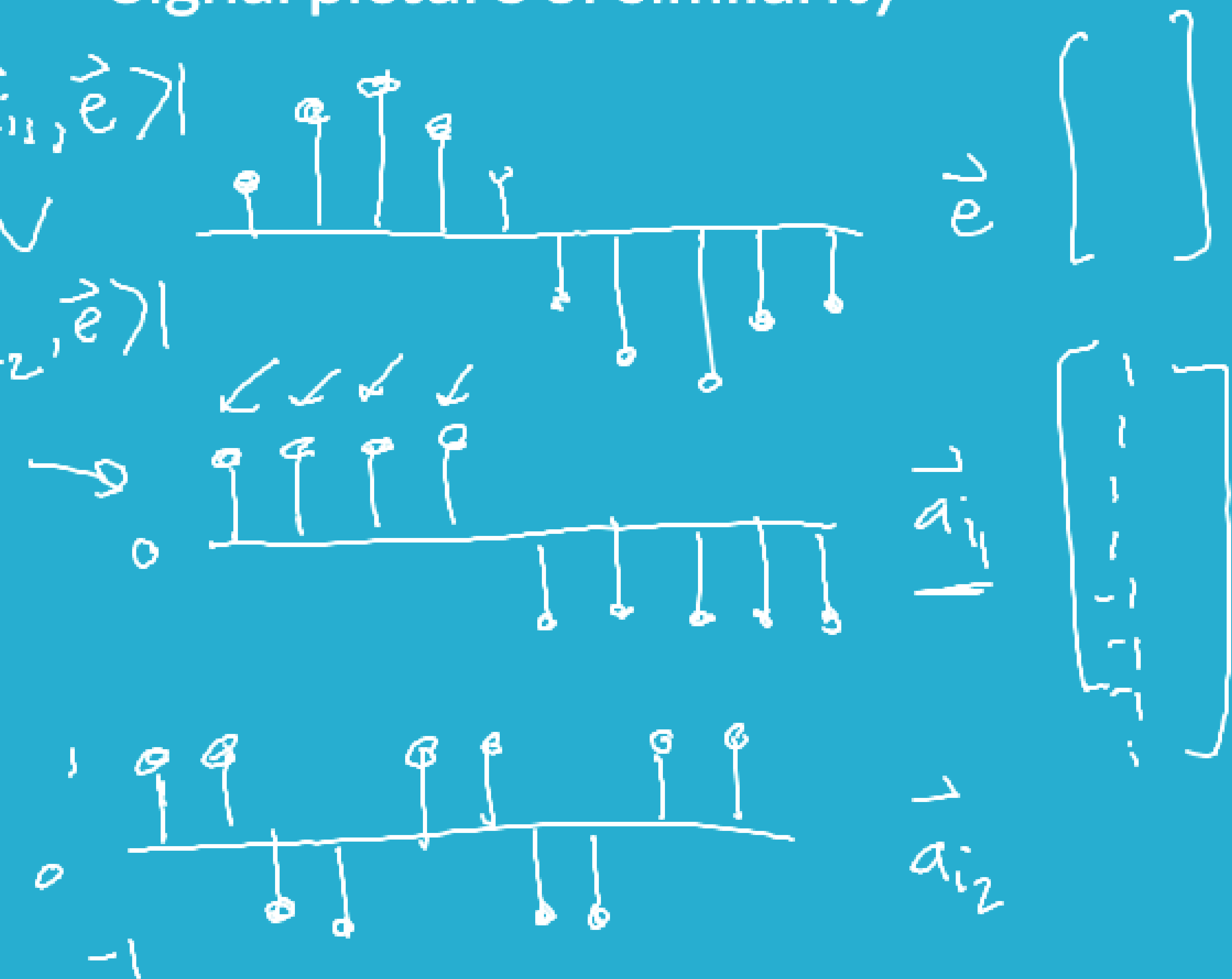


$$|\langle \vec{a}_{i_2}, \vec{e}\rangle| > |\langle \vec{a}_{i_1}, \vec{e}\rangle|$$

$\vec{a}_{i_2}$ is most similar to $\vec{e}$

Signal picture of similarity

$$|\langle \vec{a}_{i_1}, \vec{e}\rangle|$$
$$\vee$$
$$|\langle \vec{a}_{i_2}, \vec{e}\rangle|$$



Why *Orthogonal* Matching Pursuit

$\langle \vec{e}, \vec{a}_{i_5} \rangle = 0$

$\langle \vec{e}, \underbrace{10\vec{a}_{i_5}}_{\cancel{\vec{a}_{i_{10}}}} \rangle = 0$

Error is orthogonal to the $\vec{a_i}$ you chose to match $\vec{b}$ with:

$$\vec{e} \perp \vec{a_i} \iff \langle \vec{e}, \vec{a_i} \rangle = 0$$

→ ***OMP is iterated least squares***

(projection leaves error as orthogonal component)

Why *Orthogonal* Matching Pursuit

Consequence: OMP doesn't fail where least squares would.

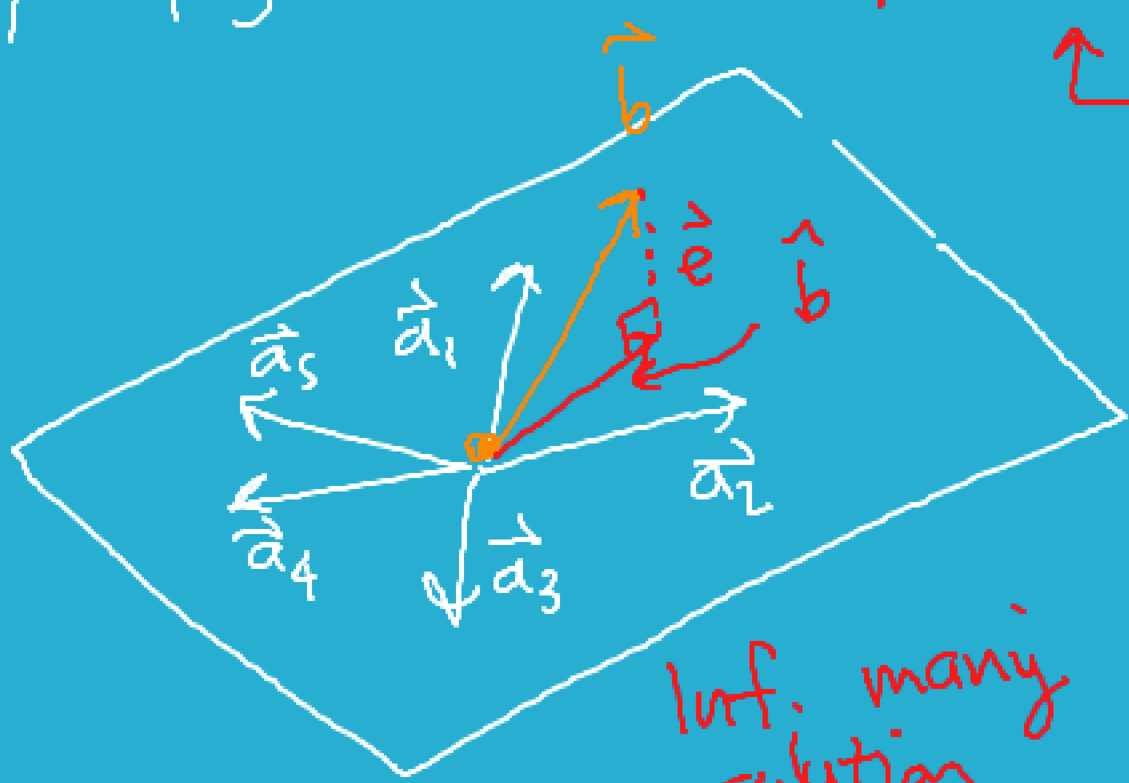
Least squares picture

$$A = \begin{bmatrix} | & & | \\ \vec{a}_1 & \cdots & \vec{a}_5 \\ | & & | \end{bmatrix}$$

$$A^T A \tilde{x} = A^T \vec{b}$$

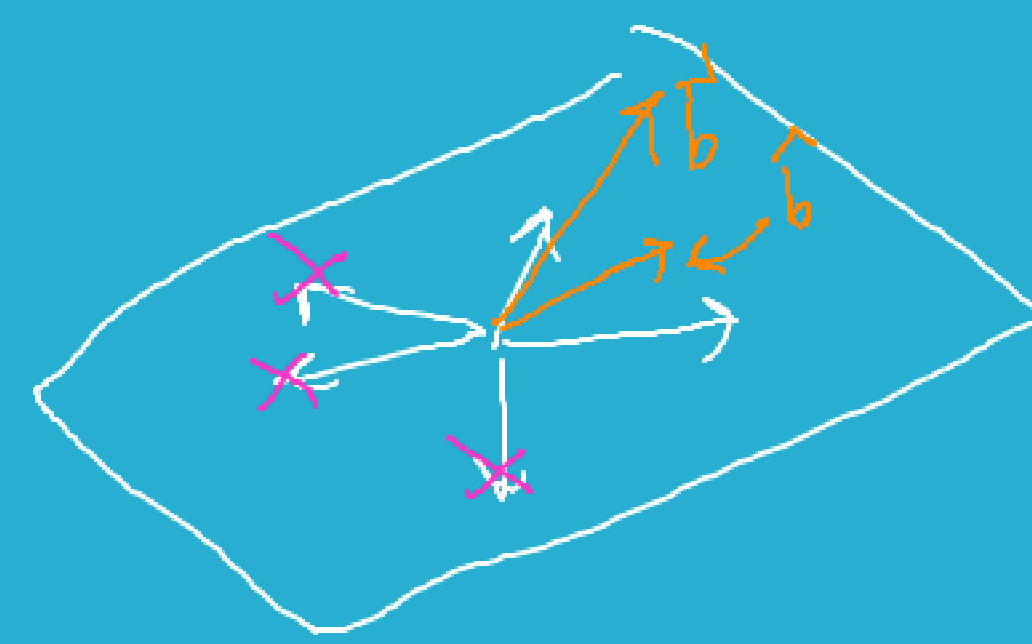
always has solution

not necessarily unique



Inf. many solution

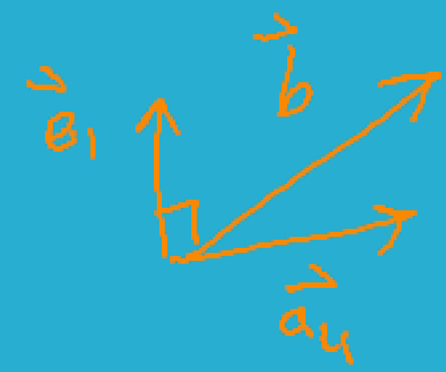
OMP picture



OMP: Iterated Least Squares

$|\langle \vec{a}_{i_1}, \vec{e}_0 \rangle| > |\langle \vec{a}_j, \vec{e}_0 \rangle|$

$A\vec{x}_1 = \hat{b} = \mathrm{proj}_{C(A_1)} \vec{b}$



$\hat{x}_1$

Update error

$$\vec{e}_0 = \vec{b}$$

Update A matrix

$$\mathbf{A_1} = [\vec{a_{i_1}}]$$

Update error

Least Squares

Compute signal estimate

Compute solution estimate

$$\vec{e}_1 = \vec{b} - \mathbf{A}_1(\mathbf{A}_1^{\mathbf{T}}\mathbf{A}_1)^{-1}\mathbf{A}_1^T\vec{b}$$

Update A matrix

$$\mathbf{A_2} = [\vec{a_{i_1}}, \vec{a_{i_2}}]$$

Update error

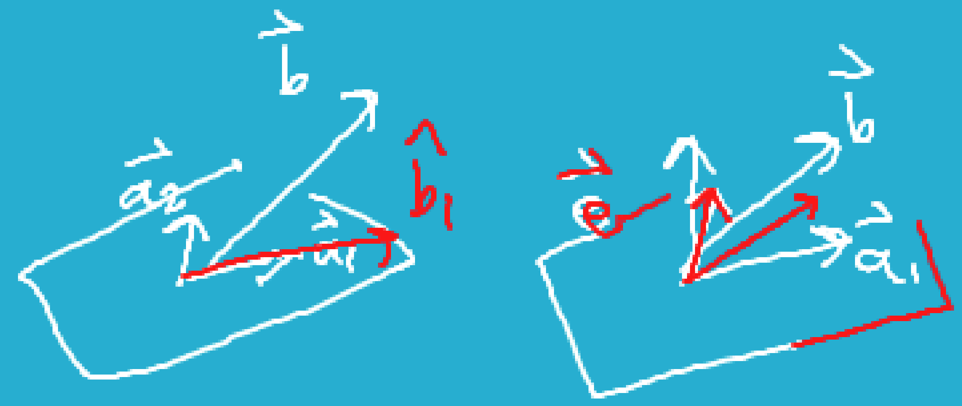
$$\vec{e}_2 = \vec{b} - \mathbf{A}_2(\mathbf{A}_2^{\mathbf{T}}\mathbf{A}_2)^{-1}\mathbf{A}_2^T\vec{b}$$

$|\langle \vec{a}_{i_2}, \vec{e}_1 \rangle| > |\langle \vec{a}_j, \vec{e}_1 \rangle|$

$\langle \vec{a}_{i_1}, \vec{e}_1 \rangle = 0$

Stopping Condition: when ||error|| is small enough, or you know you've hit the number of columns (signals) you expected to see (sparsity level - how many nonzero signals).

OMP: Iterated Least Squares



Update error

$$\vec{e_0} = \vec{b}$$

Update A matrix

$$\mathbf{A_1} = [\vec{a_{i_1}}]$$

Least Squares

Compute signal estimate

Compute solution estimate

Update error

$$\vec{e_1} = \vec{b} - \boxed{\mathbf{A}_1 \boxed{(\mathbf{A_1^T A_1})^{-1} \mathbf{A}_1^T \vec{b}}}$$

Update A matrix

$$\mathbf{A_2} = [\vec{a_{i_1}}, \vec{a_{i_2}}]$$

Update error

$$\vec{e_2} = \vec{b} - \mathbf{A}_2 (\mathbf{A_2^T A_2})^{-1} \mathbf{A}_2^T \vec{b}$$

Stopping Condition: when ||error|| is small enough, or you know you've hit the number of columns (signals) you expected to see (sparsity level - how many nonzero signals).

OMP Details

$$\begin{bmatrix} \vec{a}_5 \, \vec{a}_6 \; \vec{a}_{10} \end{bmatrix} \begin{bmatrix} \hat{x} \end{bmatrix} \approx \vec{b}$$

a

Error at the beginning is $\vec{b}$ because we've not taken anything out yet.

$$\vec{e} = \vec{b} - \vec{0}$$

$$\underline{\hat{x}} = \begin{bmatrix} \hat{x}_{31} \\ \hat{x}_{32} \end{bmatrix}$$

5th
6th
10th

Your solution at the end is $\boxed{\hat{x}}$ not $\boxed{\hat{b}}$

(Keep track of what formula you're using)

$\hat{x}_1 = \hat{x}_1 \longrightarrow$ only used one col.

$\hat{x}_2 = \begin{bmatrix} \hat{x}_{21} \\ \hat{x}_{22} \end{bmatrix} \rightarrow$ used 2 cols from A

$\hat{x}_3 = \begin{bmatrix} \hat{x}_{31} \\ \vdots \\ \hat{x}_{32} \end{bmatrix} \rightarrow$ used 3 cols

$\vec{a}_5 \; \vec{a}_6, \vec{a}_{10}$

OMP Details: Possible Mistakes!

- Subtraction from the previous error iteration, rather than removing the projection of $\vec{b}$ from $\vec{b}$:

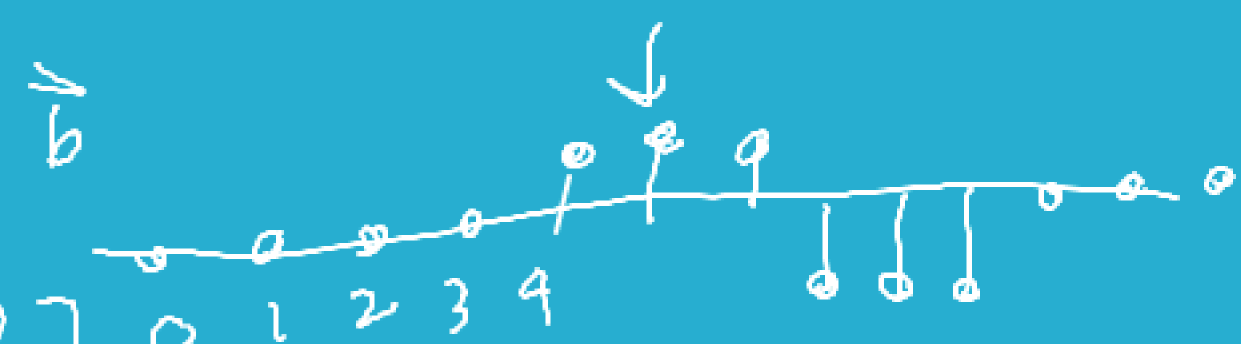
$$\vec{e_2} = \cancel{\vec{e_1}} - \mathbf{A}_1(\mathbf{A}_1^T\mathbf{A}_1)^{-1}\mathbf{A}_1^T\vec{b}$$

- Doing only one dimensional projections (onto a single vector), not the full least squares projection:

$$\vec{e_3} = \vec{b} - \cancel{\mathrm{proj}_{\vec{a_{i_3}}}}\vec{b}$$

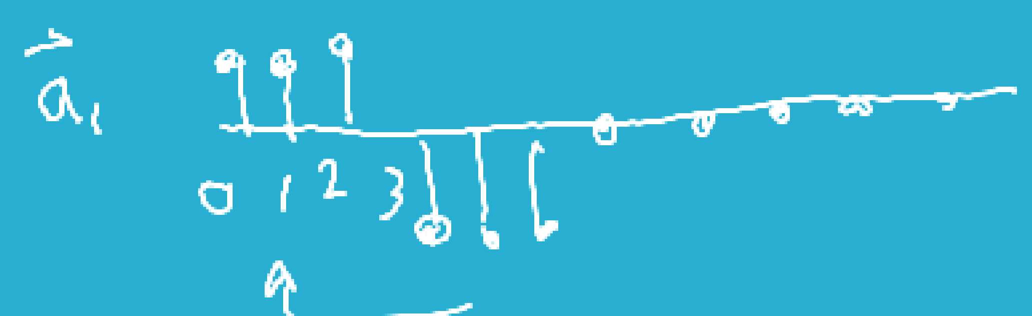
OMP Details: Correlation and Time Delay

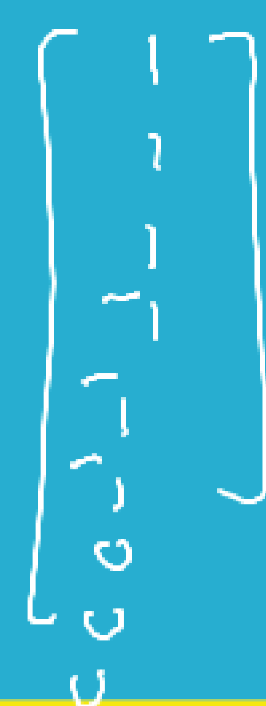
Cross correlation only needed if you know your signals can appear shifted in time.

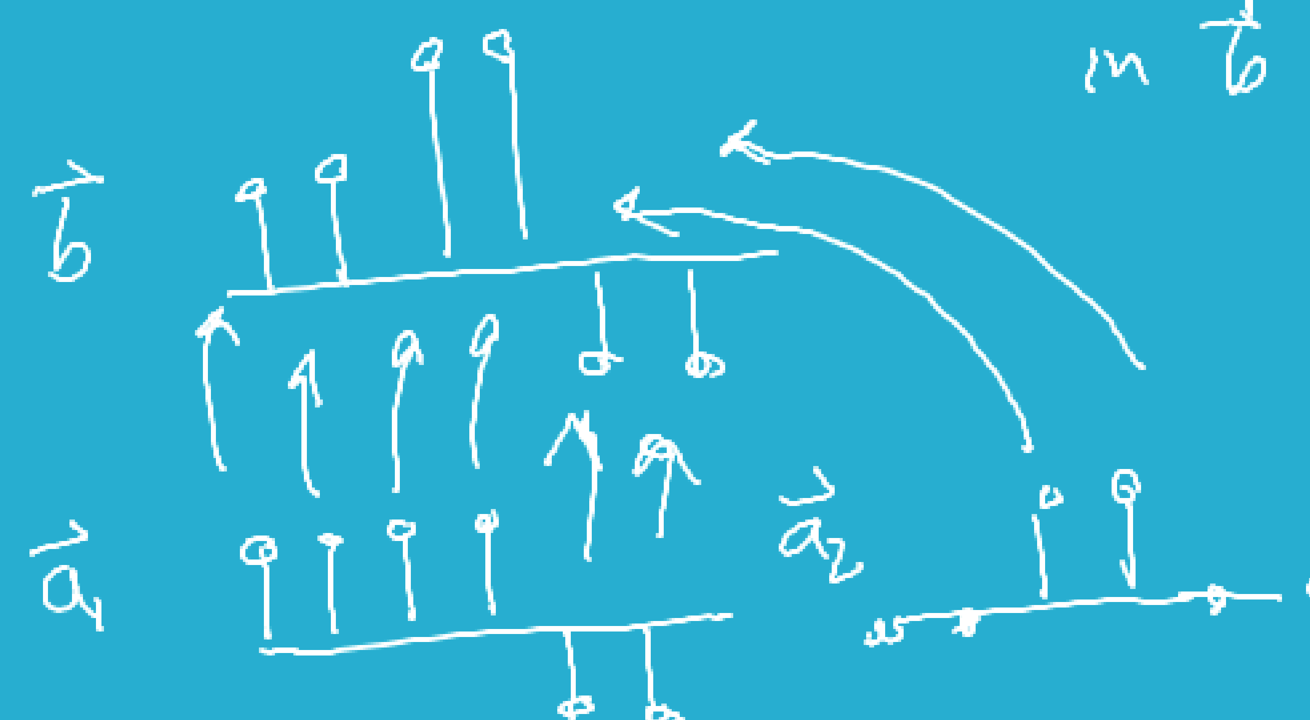


$\vec{b}$ starts @ $t = 4$

$\langle \vec{a}_{shifted}, \vec{b} \rangle$ = large → can tell that $\vec{a}$ is present in $\vec{b}$







$\langle \vec{a}_1, \vec{b} \rangle$ = small

np.corr

OMP Details: Correlation and Time Delay

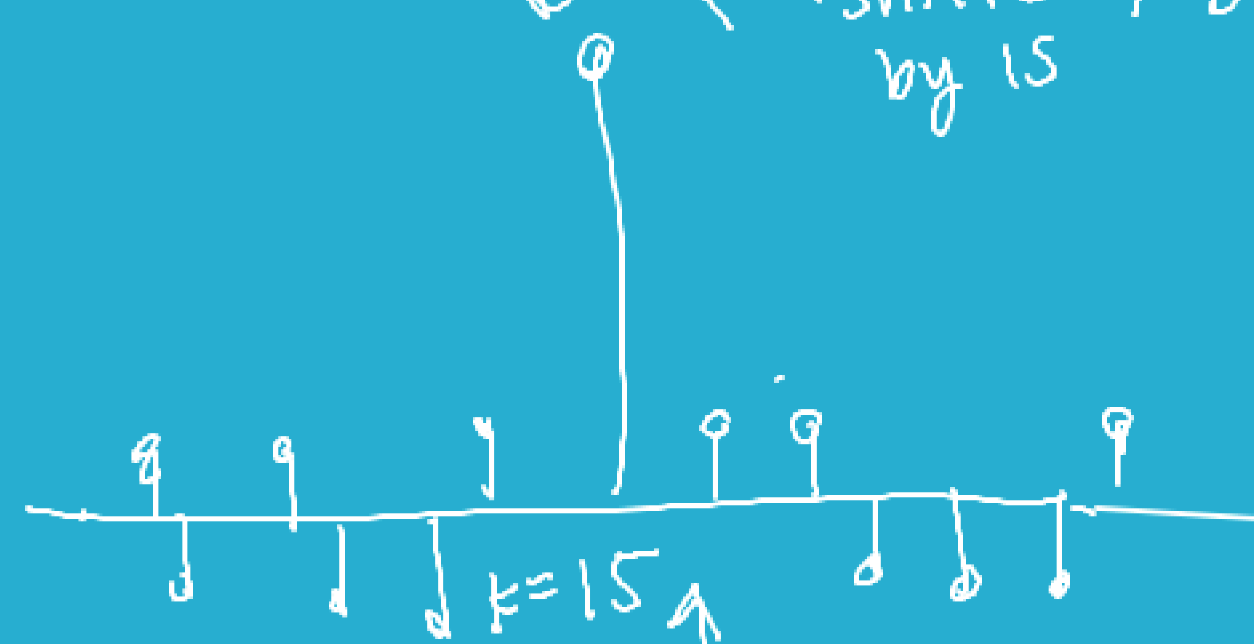
Cross correlation only needed if you know your signals can appear shifted in time.

$\langle \vec{a}_{\text{shifted by 15}}, \vec{b} \rangle$

large peak: $\vec{a}$ is present

(use $\vec{a}$ for 1st sq.)

↳ shift $\vec{a}$ by 15



corr[k]

shift / delay

$\langle \vec{a}_{\text{shifted by 16}}, \vec{b} \rangle$

Finding faults with PG&E (F19 Final #6b)

Link: F19 Final

$$\vec{s}_1[n] = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^T \xrightarrow{\text{delayed by 1}} \vec{u}_1[n] = \boxed{\begin{bmatrix}0 & 1 & 1 & 1 & 0\end{bmatrix}^T}, \tag{3}$$

$$\vec{s}_2[n] = \begin{bmatrix}1 & -1 & 1\end{bmatrix}^T \xrightarrow{\text{delayed by 2}} \vec{u}_2[n] = \begin{bmatrix}0 & 0 & 1 & -1 & 1\end{bmatrix}^T, \tag{4}$$

$$\vec{s}_3[n] = \begin{bmatrix}1 & 1 & -1\end{bmatrix}^T \xrightarrow{\text{delayed by 1}} \vec{u}_3[n] = \begin{bmatrix}0 & 1 & 1 & -1 & 0\end{bmatrix}^T, \tag{5}$$

$$\vec{s}_4[n] = \begin{bmatrix}-1 & -1 & 1\end{bmatrix}^T \xrightarrow{\text{delayed by 2}} \vec{u}_4[n] = \begin{bmatrix}0 & 0 & -1 & -1 & 1\end{bmatrix}^T. \tag{6}$$

$\vec{e}_1 = \vec{r} - \vec{u}_1 \frac{5}{3}$

$\langle \vec{e}_1, \vec{u}_i \rangle = \langle \vec{r}, \vec{u}_i \rangle - \frac{5}{3}\langle \vec{u}_1, \vec{u}_i \rangle$

Determine which two unique signals are contained in the received signal $\vec{r}[n]$. What are the weights on the two signals? Show all of your work.

$\boxed{\vec{r} \approx \alpha \vec{u}_1 + \beta \vec{u}_4}$

Some calculations that might be useful:

$<\vec{r}[n], \vec{u}_1[n]>= 5$	$<\vec{u}_1[n], \vec{u}_2[n]>= 0$	$<\vec{u}_2[n], \vec{u}_3[n]>= 2$	$<\vec{u}_3[n], \vec{u}_4[n]>= 0$
$<\vec{r}[n], \vec{u}_2[n]>= 0$	$<\vec{u}_1[n], \vec{u}_3[n]>= 1$	$<\vec{u}_2[n], \vec{u}_4[n]>= 1$	
$<\vec{r}[n], \vec{u}_3[n]>= 1$	$<\vec{u}_1[n], \vec{u}_4[n]>= -2$		
$<\vec{r}[n], \vec{u}_4[n]>= -2$			

$\vec{u}_2 \to 0 - \cancel{\frac{5}{3}}\ 0$

$\vec{u}_3 \to 1 - \frac{5}{3}(1) = -\frac{2}{3}$

This might also help:

$$\begin{bmatrix}a & b\\ c & d\end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix}d & -b\\ -c & a\end{bmatrix} \tag{7}$$

$\vec{u}_4 \to -2 - \frac{5}{3}(-2) = \frac{4}{3}$

$\vec{u}_1, \vec{u}_4$

$\vec{e}_0 = \vec{r}$

Choose $\vec{u}_1$ to match $\vec{r}$ $(\vec{e}_0)$ with

$\vec{e}_1 = \vec{r} - \vec{u}_1(\vec{u}_1^T\vec{u}_1)^{-1}\vec{u}_1^T\vec{r}$

$\vec{e}_1 = \begin{bmatrix} \\ \end{bmatrix} - \vec{u}_1(3)^{-1}5 = \begin{bmatrix}1\\2\\1\\2\\1\end{bmatrix} - \begin{bmatrix}0\\1\\1\\1\\0\end{bmatrix}\frac{5}{3} = \begin{bmatrix}1\\1/3\\-2/3\\1/3\\ \end{bmatrix}$