Project discussion, 22 May: Mandatory but ungraded.

Thanks for doing this

TOMMOROW!

June 4, 6pm deadline for submitting poster for printing (pdf preferred). TAs have to print 43 posters. Dropbox link or email TA

https://www.dropbox.com/request/XGqCV0qXm9LBYz7J1msS

June 5, 5-8pm Atkinson Hall: Poster and Pizza. Easels available.

June 15, 8am deadline for submitting report and code. (we have 43 reports to read in 3 days!) use dropbox link or email TA

https://www.dropbox.com/request/XGqCV0qXm9LBYz7J1msS

Evaluation

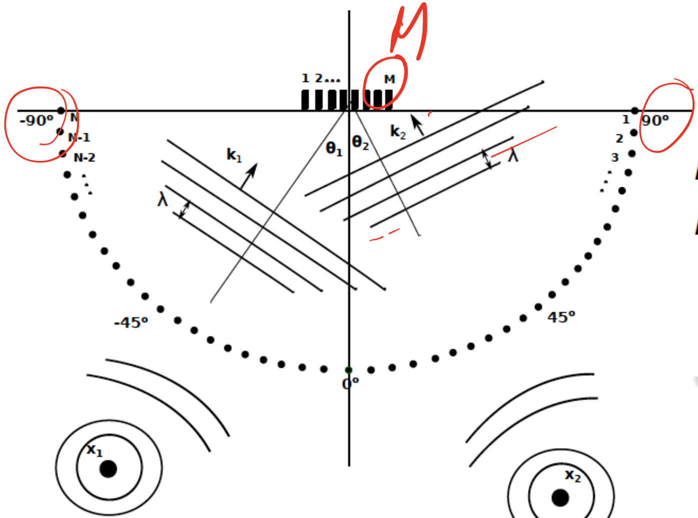
Report:30%

Poster: 10% (as displayed)

Code: 10% (should run automatically)

Beamforming / DOA estimation

DOA estimation with sensor arrays



$p_1(\mathbf{r},t) = x_1 e^{j(\omega t - \mathbf{k}_1 \mathbf{r})}$ $\quad$ $p_2(\mathbf{r},t) = x_2 e^{j(\omega t - \mathbf{k}_2 \mathbf{r})}$

$$x \in \mathbb{C},\ \theta \in [-90^\circ, 90^\circ]$$

$$\mathbf{k} = -\frac{2\pi}{\lambda}\sin\theta,\ \lambda\text{:wavelength}$$

$$y_m = \sum_n x_n e^{j\frac{2\pi}{\lambda} r_m \sin\theta_n}$$

$m \in [1, \cdots, M]$: sensor

$n \in [1, \cdots, N]$: look direction

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

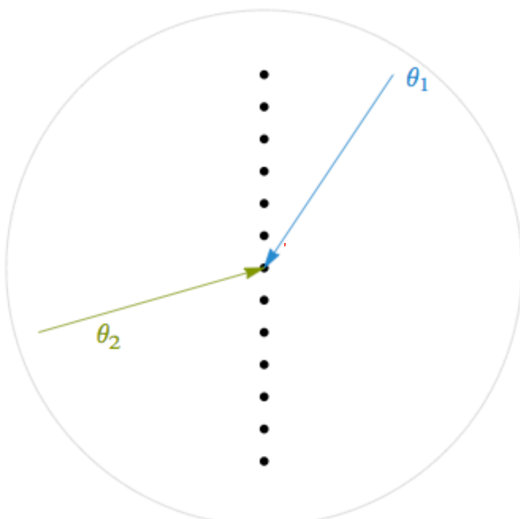
$$\mathbf{y} = [y_1, \cdots, y_M]^T, \quad \mathbf{x} = [x_1, \cdots, x_N]^T$$

$$\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N]$$

$$\mathbf{a}_n = \frac{1}{\sqrt{M}}[e^{j\frac{2\pi}{\lambda} r_1 \sin\theta_n}, \cdots, e^{j\frac{2\pi}{\lambda} r_M \sin\theta_n}]^T$$

The DOA estimation is formulated as a linear problem

Direction of arrival estimation



Plane waves from a source/interferer impinging on an array/antenna

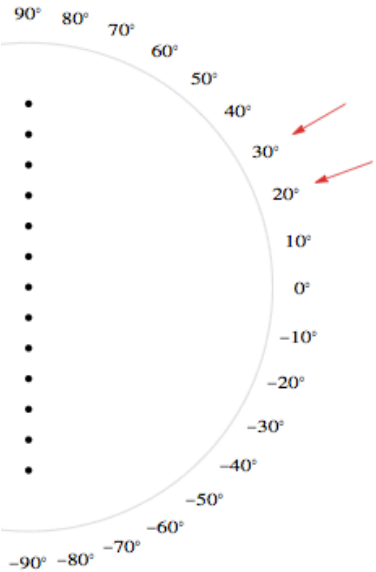
True DOA is sparse in the angle domain

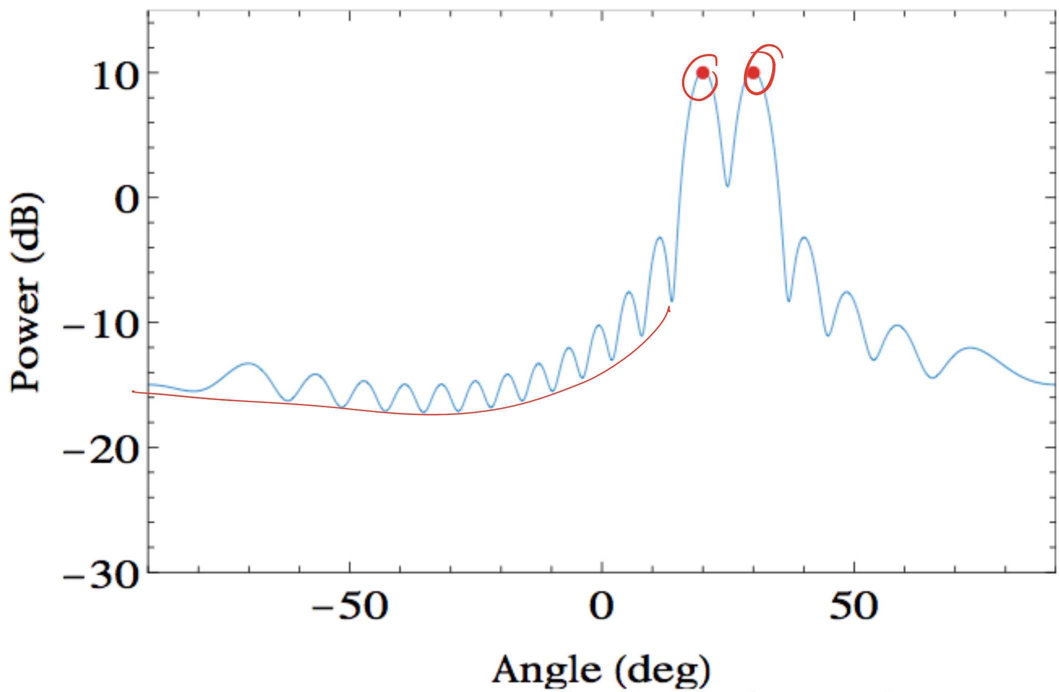
$$\boldsymbol{\Theta} = \{0, \cdots, 0, \theta_1, 0, \cdots, 0, \theta_2, 0, \cdots, 0\}$$

Conventional beamforming

Plane wave weight vector $\mathbf{w}_i = [1, e^{-\imath \sin(\theta_i)}, \cdots, e^{-\imath (N-1)\sin(\theta_i)}]^T$

$$\mathcal{B}(\theta) = |\mathbf{w}^H(\theta)\mathbf{b}|^2$$



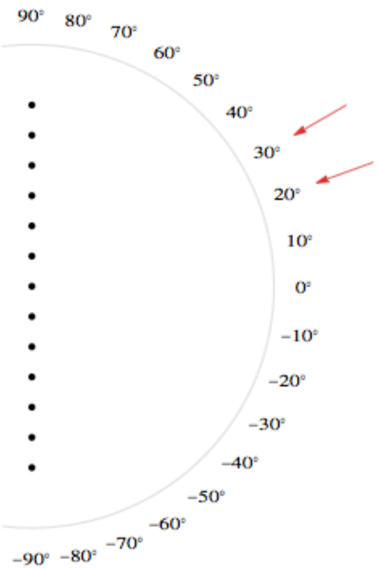


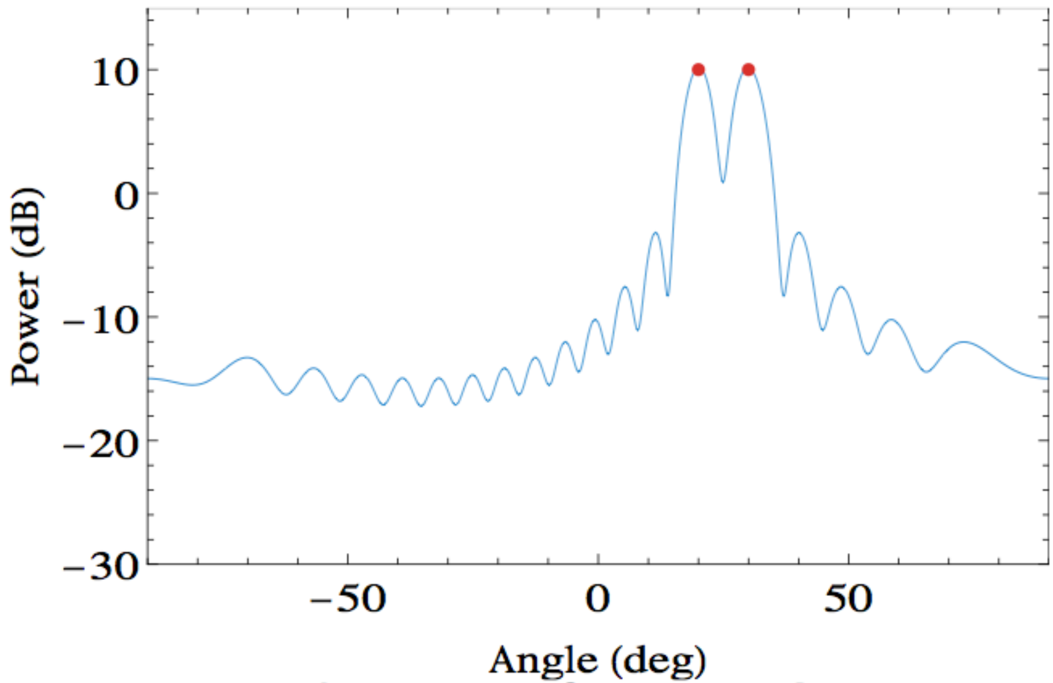
ULA, half-wavelength spacing, $N = 20$ sensors, $\theta_1 = 20°$, $\theta_2 = 30°$,

Conventional beamforming

Equivalent to solving the ℓ_2 problem with $\mathbf{A} = [\mathbf{w}_1, \cdots, \mathbf{w}_M]$, $M > N$.

$$\min \|\mathbf{x}\|_2 \text{ subject to } \mathbf{Ax} = \mathbf{b}$$



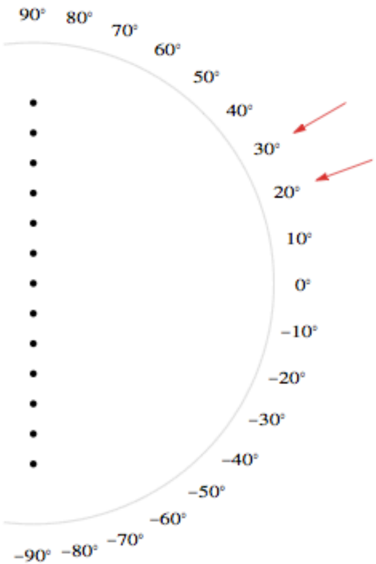


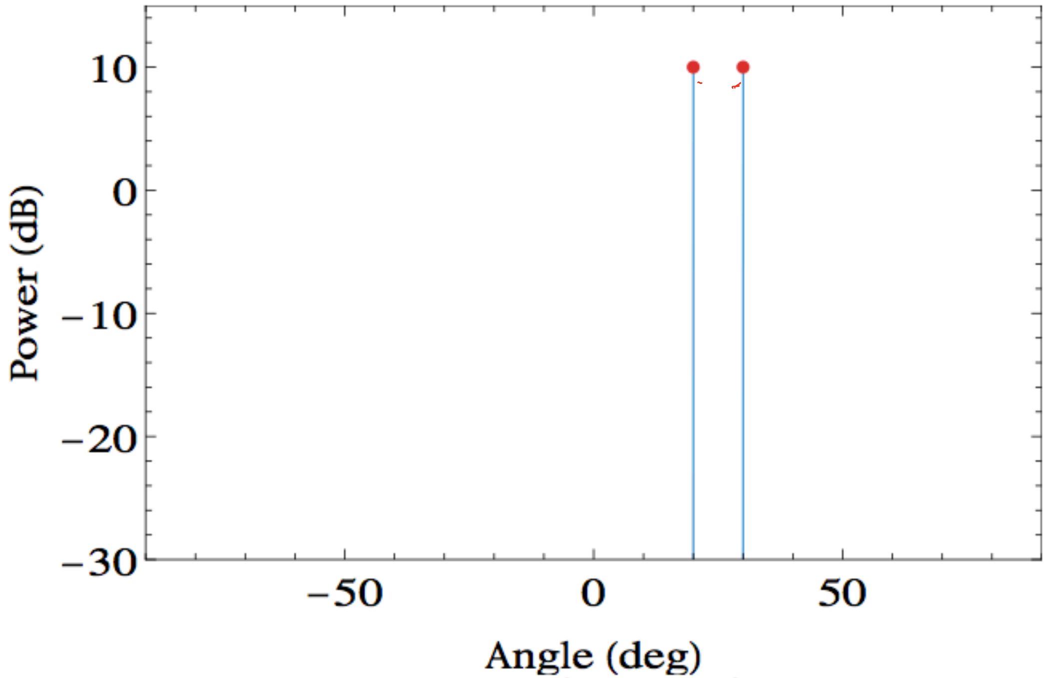
A is an overcomplete dictionary of candidate DOA vectors. Columns span $-90°$ to $90°$ in steps of $1°$ ($M = 181$).

ℓ_1 minimization

In contrast ℓ_1 minimization provides a sparse solution with exact recovery:

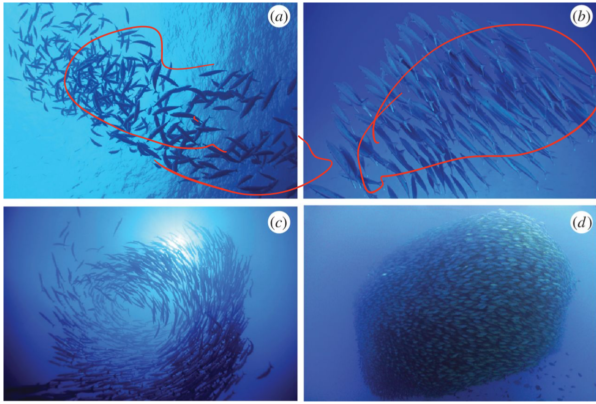
$$\min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{A}\mathbf{x} = \mathbf{b}$$



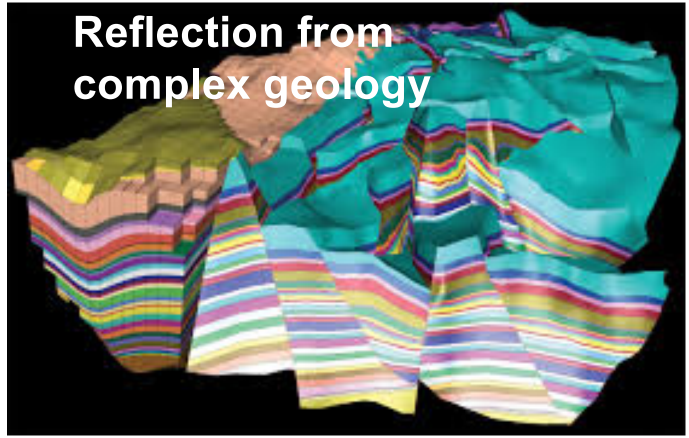


Columns of **A** span −90° to 90° in steps of 1° ($M = 181$).

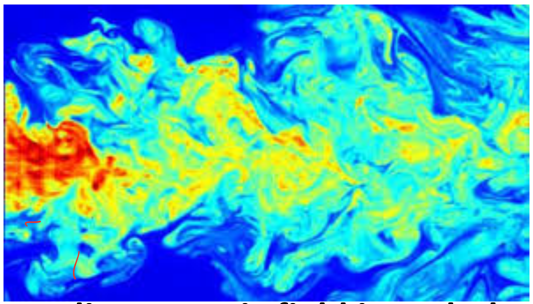
We can’t model everything…



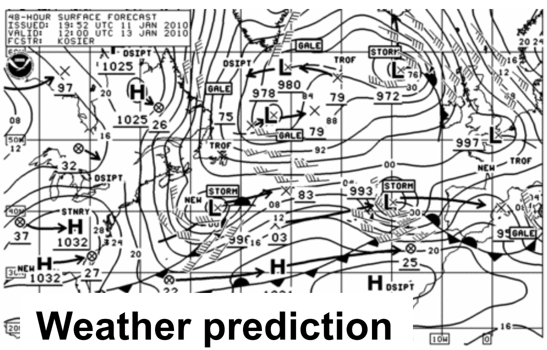
Back scattering from fish school



Reflection from complex geology



Predict acoustic field in turbulence



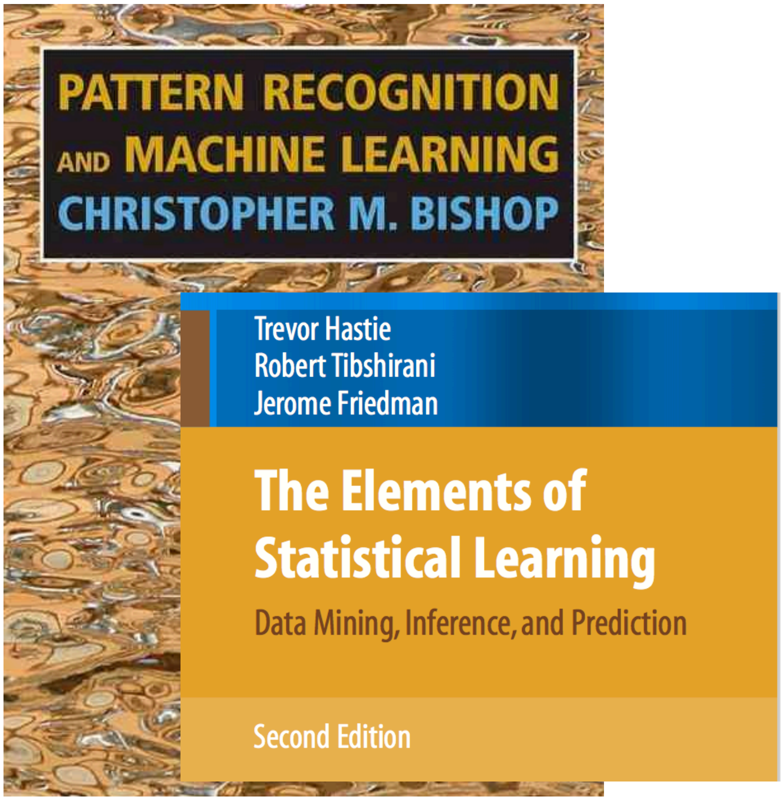
Weather prediction

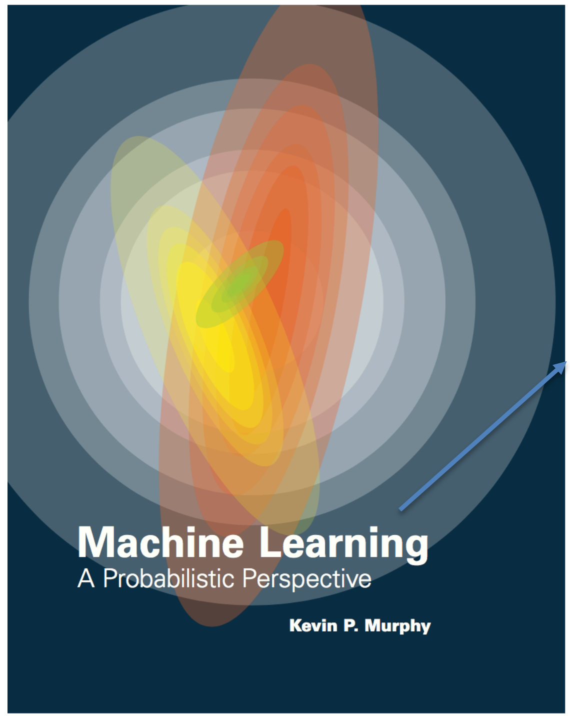
Detection of mines. Navy uses dolphins to assist in this.

Dolphins = real ML!

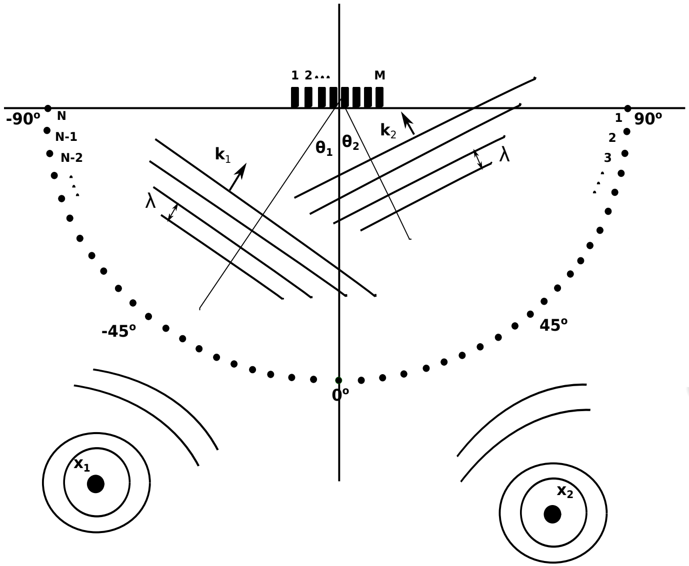
Machine Learning for physical Applications
noiselab.ucsd.edu

Murphy: "…**the best way to make machines that can learn from data is to use the *tools of probability theory*, which has been the mainstay of statistics and engineering for centuries**."





DOA estimation with sensor arrays



$p_1(\mathbf{r},t) = x_1\, e^{j(\omega t - \mathbf{k}_1 \mathbf{r})}$ $\quad$ $p_2(\mathbf{r},t) = x_2\, e^{j(\omega t - \mathbf{k}_2 \mathbf{r})}$

$$y_m = \sum_n x_n e^{j\frac{2\pi}{\lambda} r_m \sin\theta_n}$$

$m \in [1, \cdots, M]$: sensor

$n \in [1, \cdots, N]$: look direction

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

$$\mathbf{y} = [y_1, \cdots, y_M]^T, \quad \mathbf{x} = [x_1, \cdots, x_N]^T$$

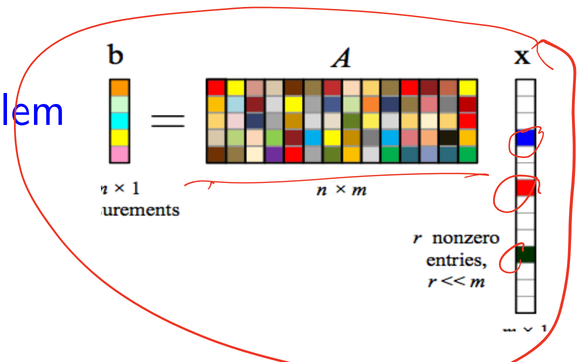
$$\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N]$$

$$\mathbf{a}_n = \frac{1}{\sqrt{M}}[e^{j\frac{2\pi}{\lambda} r_1 \sin\theta_n}, \cdots, e^{j\frac{2\pi}{\lambda} r_M \sin\theta_n}]^T$$

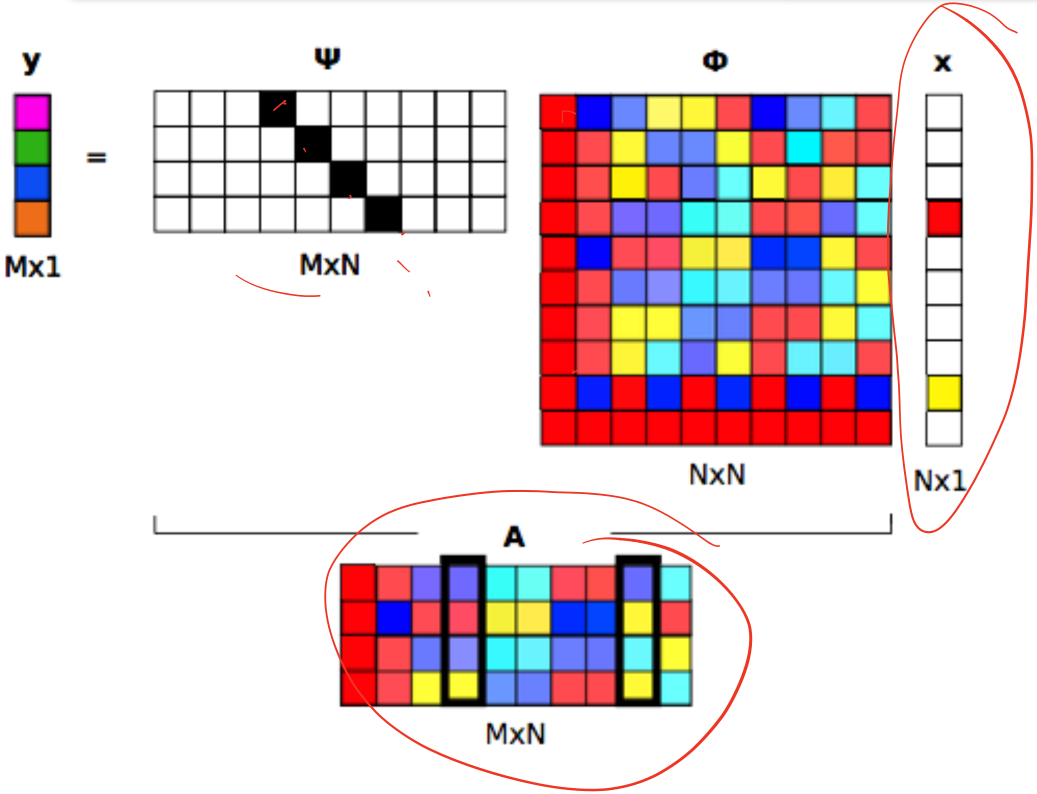
$$x \in \mathbb{C}, \ \theta \in [-90^\circ, 90^\circ]$$

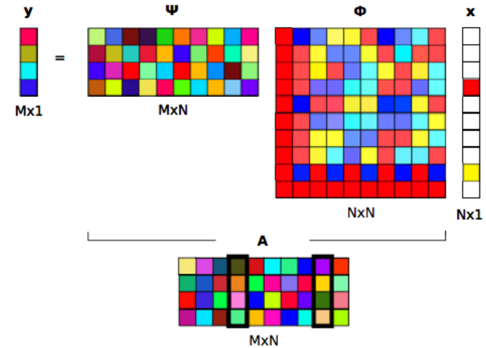
$$\mathbf{k} = -\frac{2\pi}{\lambda}\sin\theta, \ \lambda\text{:wavelength}$$

The DOA estimation is formulated as a linear problem



Compressive beamforming





$\boldsymbol{y}$: measurement vector

$\boldsymbol{\Phi}$: Transform matrix

x: desired sparse vector

$\boldsymbol{\psi}$: selection matrix

A: measurement matrix

Sparse: N>K

Often N>>M

In compressive beamforming $\boldsymbol{\psi}$ is given by sensor position

$$\min \|\boldsymbol{x}\|_0 \ \text{ subject to } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\| < \varepsilon$$

[Edelman,2011; Xenaki 2014; Fortunati 2014; Gerstoft 2015]

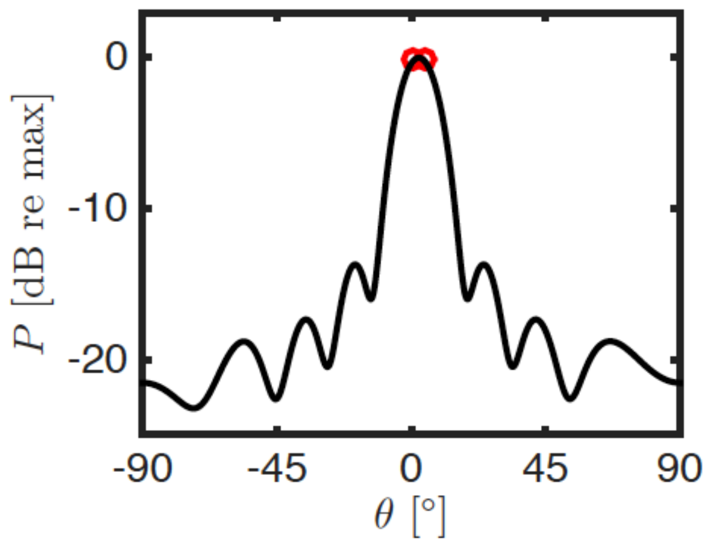
Conventional Beamforming

Solving

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ $\quad\quad\quad$ $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_N]$ $\quad\quad\quad$ $\boldsymbol{a}_1^{\mathrm{H}}\boldsymbol{a}_1 = 1$

Gives

$$\boldsymbol{x} = \boldsymbol{A}^{+}\boldsymbol{y} = (\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}})^{-1}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{y} \approx \boldsymbol{A}^{\mathrm{H}}\boldsymbol{y} = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{H}}\boldsymbol{y} \\ \vdots \\ \boldsymbol{a}_N^{\mathrm{H}}\boldsymbol{y} \end{bmatrix}$$



With L snapshots we get the power

$$x_n^2 = \boldsymbol{a}_1^{\mathrm{H}}\mathbf{C}\boldsymbol{a}_1$$

With the sample covariance matrix

$$\boldsymbol{C} = \frac{1}{L}\sum_{l=1}^{L}\boldsymbol{y}_l\boldsymbol{y}_l^H$$

More advanced beamformers exists that

Beamfroming *vs* Compressive sensing

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \qquad M < N$$

$$\mathbf{n} \in \mathbb{C}^M, \; \text{SNR} = 20log_{10}\frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{n}\|_2}, \; \|\mathbf{n}\|_2 \leq \epsilon$$

Conventional beamforming (CBF)

simplified ℓ_2-norm minimization ($\mathbf{A}\mathbf{A}^H = \mathbf{I}_M$)

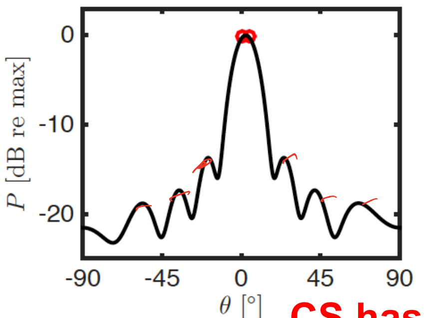
$$\hat{\mathbf{x}} = \mathbf{A}^H\mathbf{y} = \mathbf{A}^H\mathbf{A}\mathbf{x} + \mathbf{A}^H\mathbf{n}$$

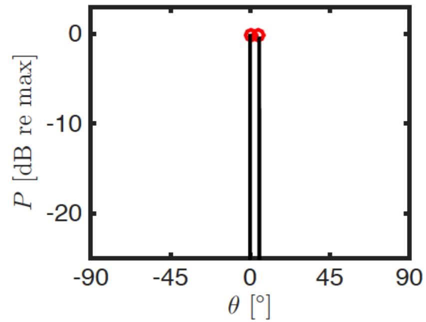
Compressive sensing (CS)

ℓ_1-norm minimization

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{x}\|_1 \text{ s.t. } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon$$

ULA $M = 8$, $\frac{d}{\lambda} = \frac{1}{2}$, $\{\theta_1, \theta_2\} = \{0^\circ, 5^\circ\}$, SNR $= 20$ dB





CS has no side lobes!

CS provides high-resolution imaging

$\|Ax - y\|_2 + \lambda \|x\|_1$

Off-the-grid versus on-the-grid

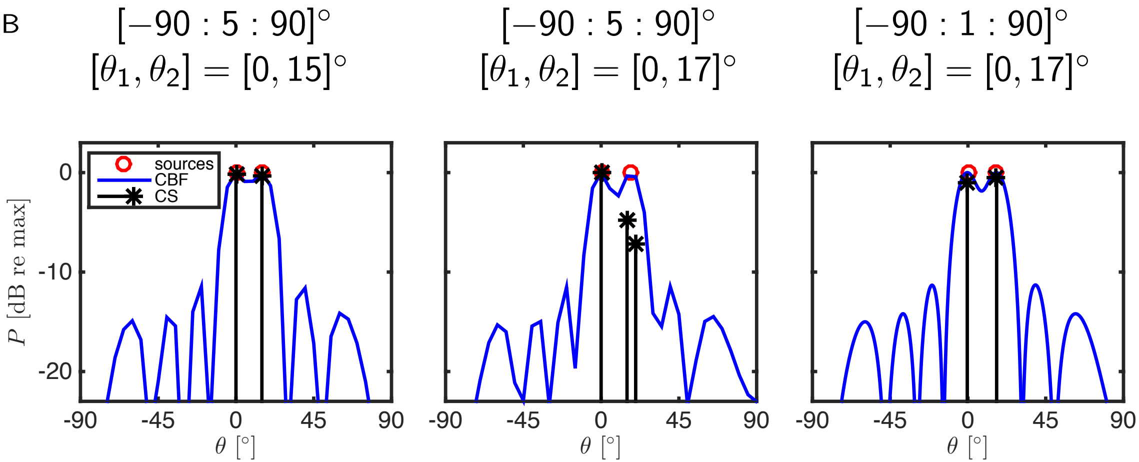
Physical parameters $\boldsymbol{\theta}$ are often continuous

Discretize

$$\boldsymbol{y} = \boldsymbol{A}(\theta)\boldsymbol{x} + \boldsymbol{n} \longrightarrow \boldsymbol{y} \approx \boldsymbol{A}_{\text{grid}}\boldsymbol{x} + \boldsymbol{n}$$

Grid-mismatch effects: Energy of an off-grid source is spread among

ULA $M = 8$, $\frac{d}{\lambda} = \frac{1}{2}$, SNR=20dB

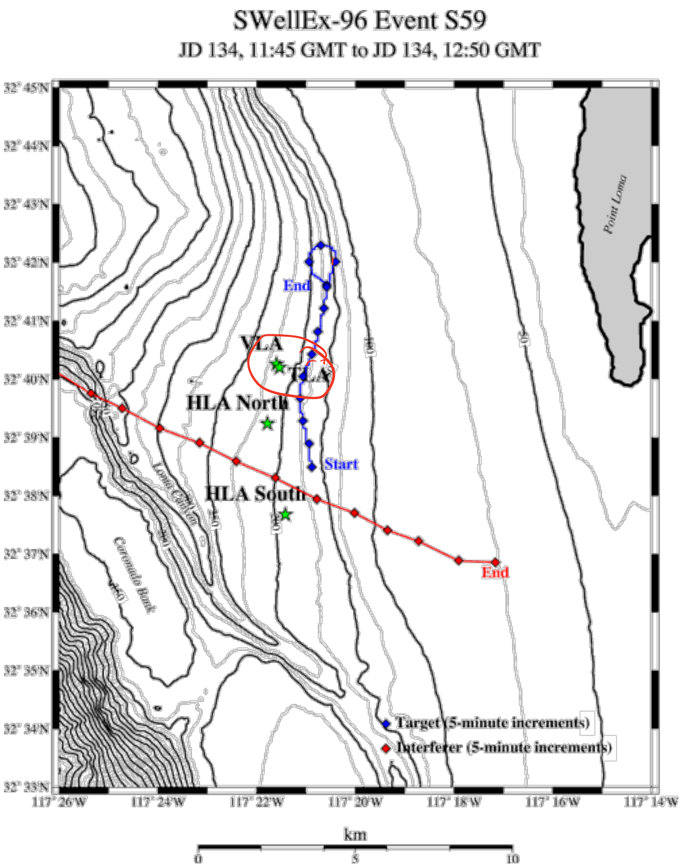


A fine angular resolution can ameliorate this problem

Continuous grid methods are being developed

=>[Angeliki Xenaki; Yongmin Choo; Yongsung Park]

[Xenaki, JASA, 2015]



SWellEx-96 Event S59:

Source 1 (S1) at 50 m depth (blue)

Surface Interferer (red)

14*3=42 processed frequencies:

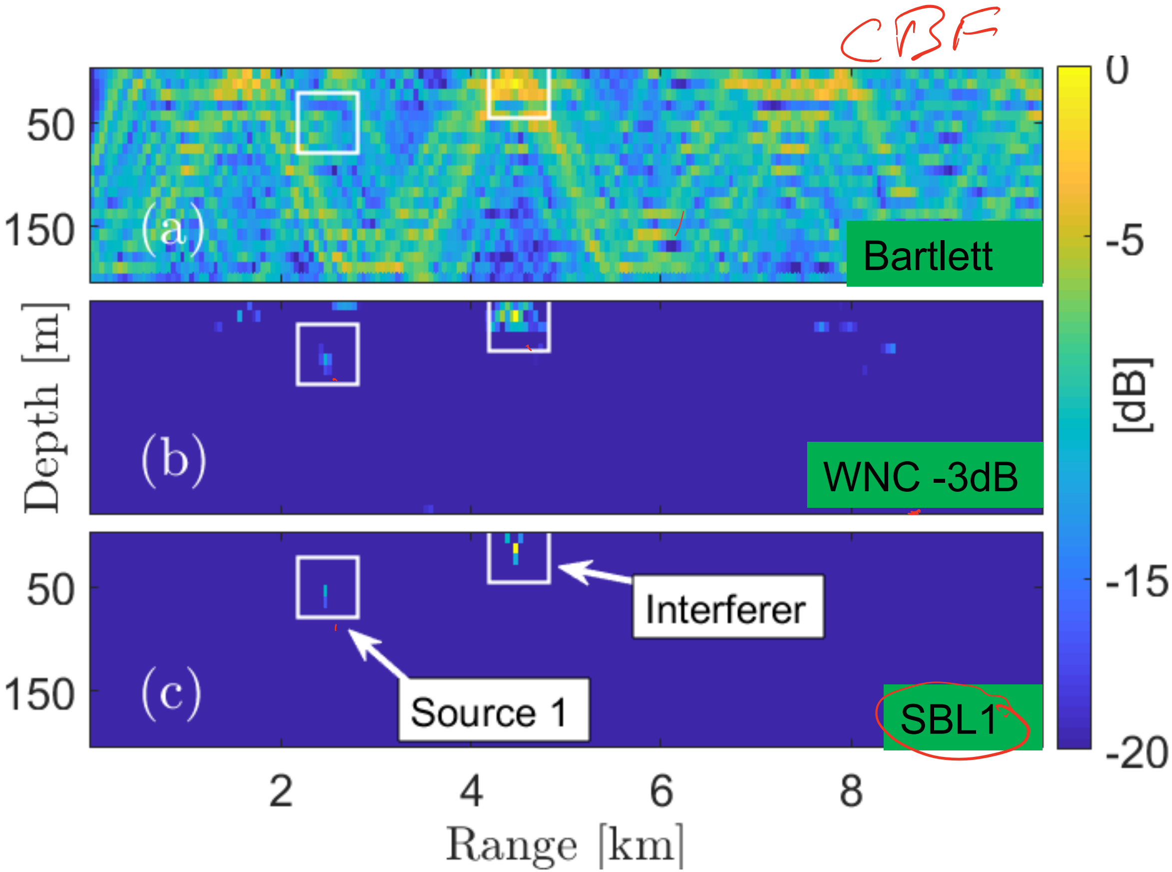
- 166 Hz (S1 SL at 150 dB re 1 μPa)
- 13 freq. ranging from 52-391 Hz (S1 SL at 122-132 dB re 1 μPa)
- +/- 1 bin each

FFT Length: 4096 samples rec. at 1500 Hz

21 Snapshots @ 50% overlap

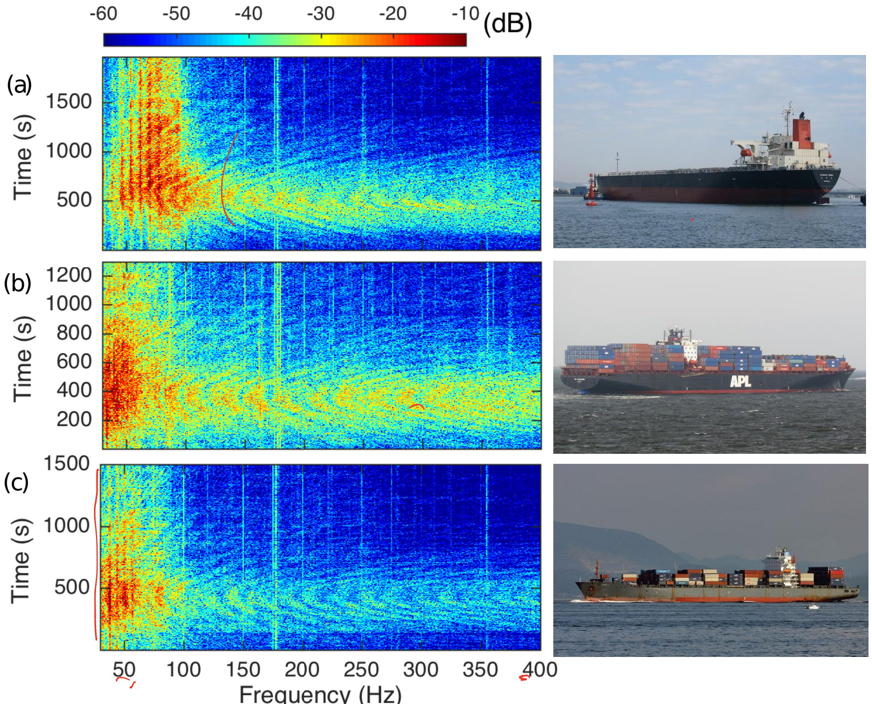
135 segments

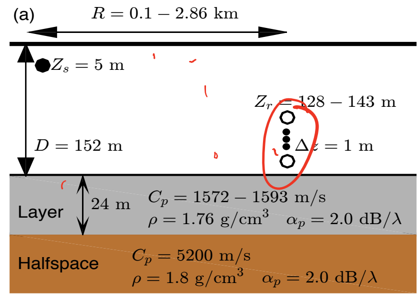
Experiment site (near San Diego) with Source (blue) and Interferer (red) track.



- Simulation
- Source 1 (50 m)
- Surface Interferer
- Freq. = 204 Hz
- SNR = 10 dB
- Int/S1 = 10 dB
- Stationary noise

Ship localization using machine learning





Ship range is extracted underwater noise from array

Sample covariance matrix (SCM) has range-dependent signature

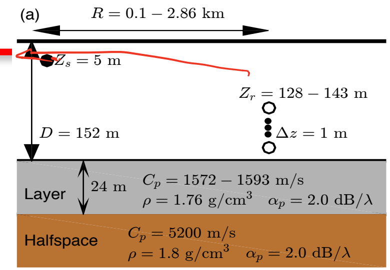
Averaging SCM overcomes noisy environments

***Old method:* Matched-Field Processing or (MFP)**

Need environmental parameters for prediction

Niu 2017a, JASA
Niu 2017b, JASA

tched-Field Processing on test data 1

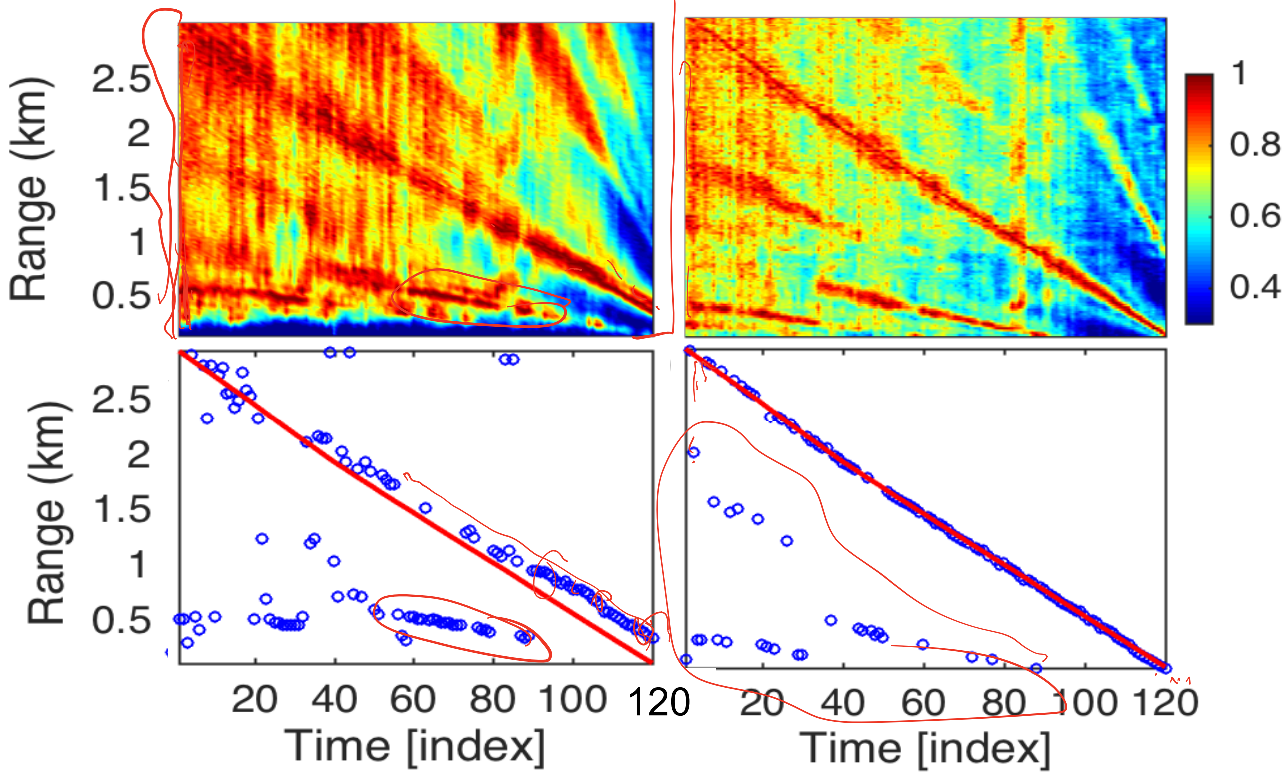


Frequencies [300:10:950]Hz

$$E_{\text{MAPE}} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{Rp_i - Rg_i}{Rg_i} \right|$$

$$B = \mathbf{p}^H \mathbf{C} \mathbf{p}$$

synthetic replicas. measured replicas



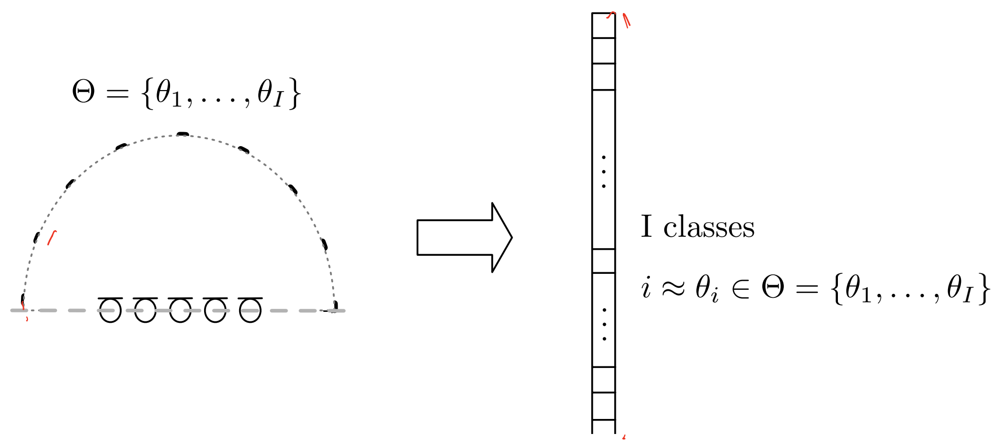
Mean Absolute Percentage Error error of MFPs: ***55%*** and ***19%***

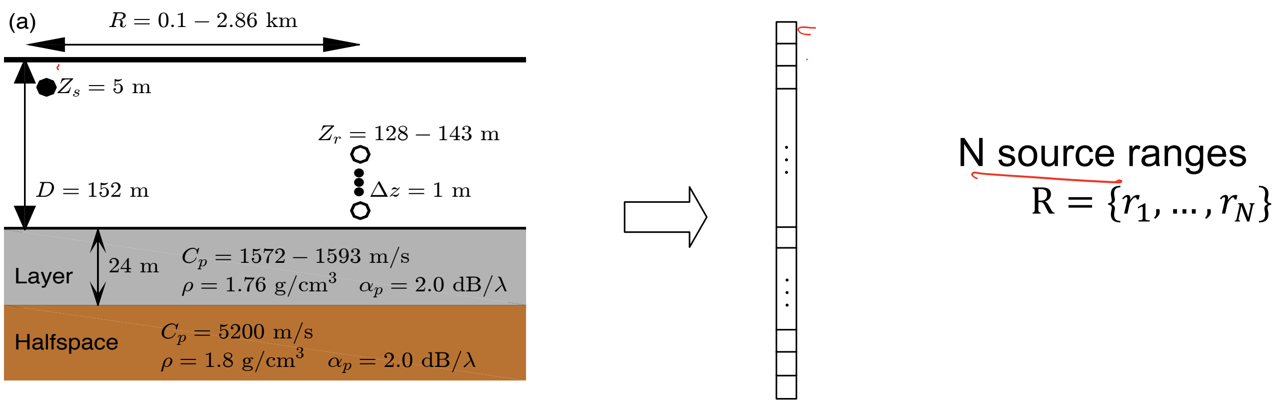
DOA estimation as a classification problem

DOA estimation can formulated as an classification with I classes

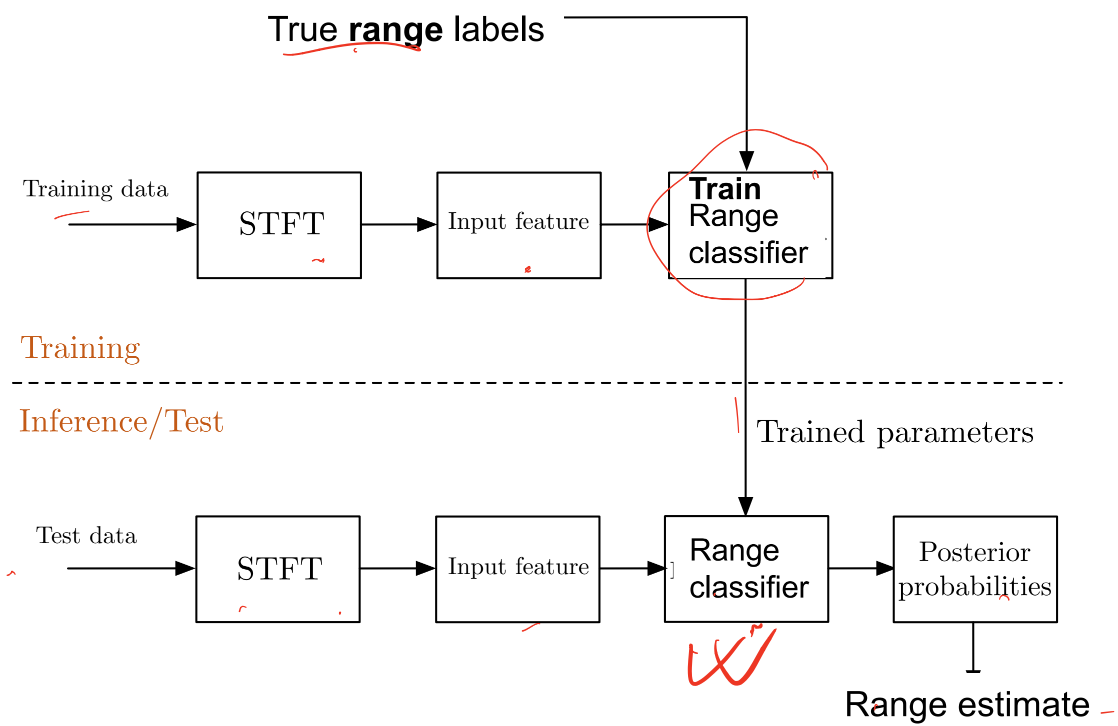
Discretize the whole DOA into a set I discrete values $\Theta = \{\theta_1, \dots, \theta_N\}$

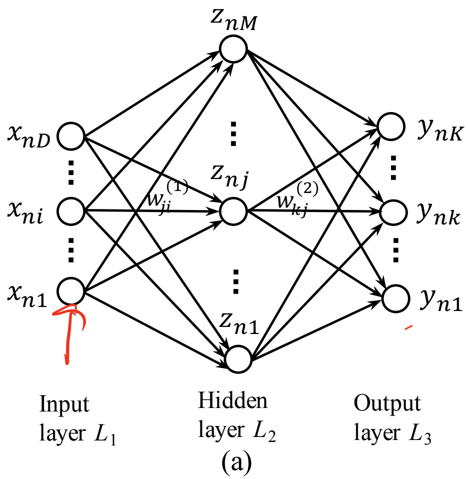
Each class corresponds to a potential DOA.

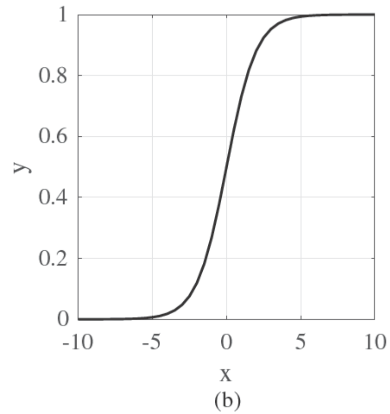




Supervised learning framework







Sigmoid function $f(a) = \sigma(a) = \frac{1}{1+e^{-a}}$

Input: preprocessed sound pressure data

Output (softmax function): probability distribution of the possible ranges

Connections between layers: Weights and biases

From layer1 to layer2: $a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \quad j = 1, \cdots, M, \Rightarrow z_j = f(a_j).$

Output layer: $a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \quad k = 1, \cdots, K \Rightarrow y_k(\mathbf{x}, \mathbf{w}) = \frac{\exp(a_k(\mathbf{x}, \mathbf{w}))}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x}, \mathbf{w}))}, \quad k = 1, \cdots, K$

Softmax

Pressure data preprocessing

Sound pressure

$$\mathbf{p}(f) = S(f)\mathbf{g}(f, \mathbf{r}) + \mathbf{n},$$

$S(f)$ Source term

Normalize pressure to reduce the effect of $|S(f)|$

$$\tilde{\mathbf{p}}(f) = \frac{\mathbf{p}(f)}{\sqrt{\sum_{l=1}^{L} |p_l(f)|^2}} = \frac{\mathbf{p}(f)}{\|\mathbf{p}(f)\|_2}$$

L Number of sensors

Sample Covariance Matrix to reduce effect of source phase

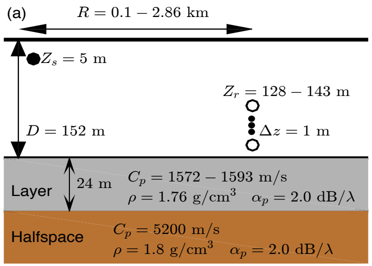
$$\mathbf{C}(f) = \frac{1}{N_s} \sum_{s=1}^{N_s} \tilde{\mathbf{p}}_s(f) \tilde{\mathbf{p}}_s^H(f)$$

N_s Number of snapshots

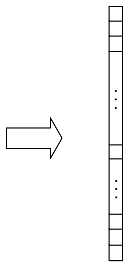
SCM is a conjugate symmetric matrix.

Input vector X: the real and imaginary parts of the entries of diagonal and upper triangular matrix in $\mathbf{C}(f)$

Classification versus regression

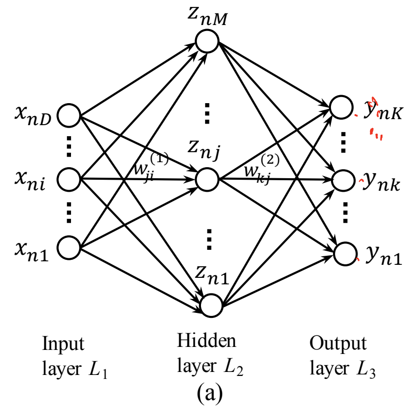


Classification:

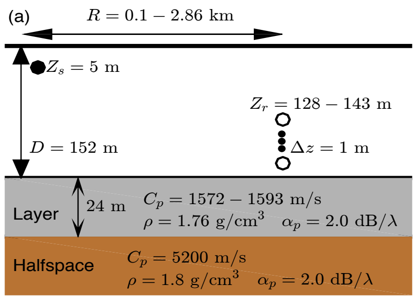


N potential source ranges

$$R = \{r_1, \ldots, r_N\}$$

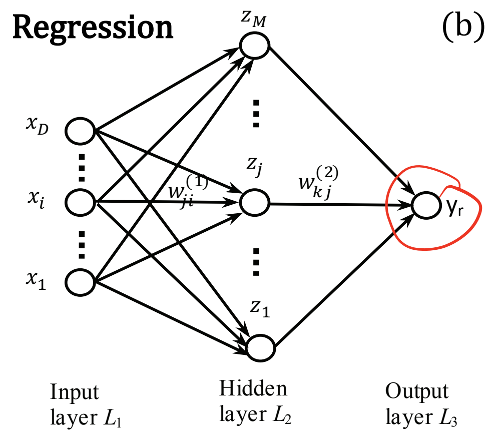


Regression:



one source continuous range

Regression is harder



Number of parameters

MFP: O(10)

ML: 400*1000+ 1000*1000+1000*100

= O(1000000)

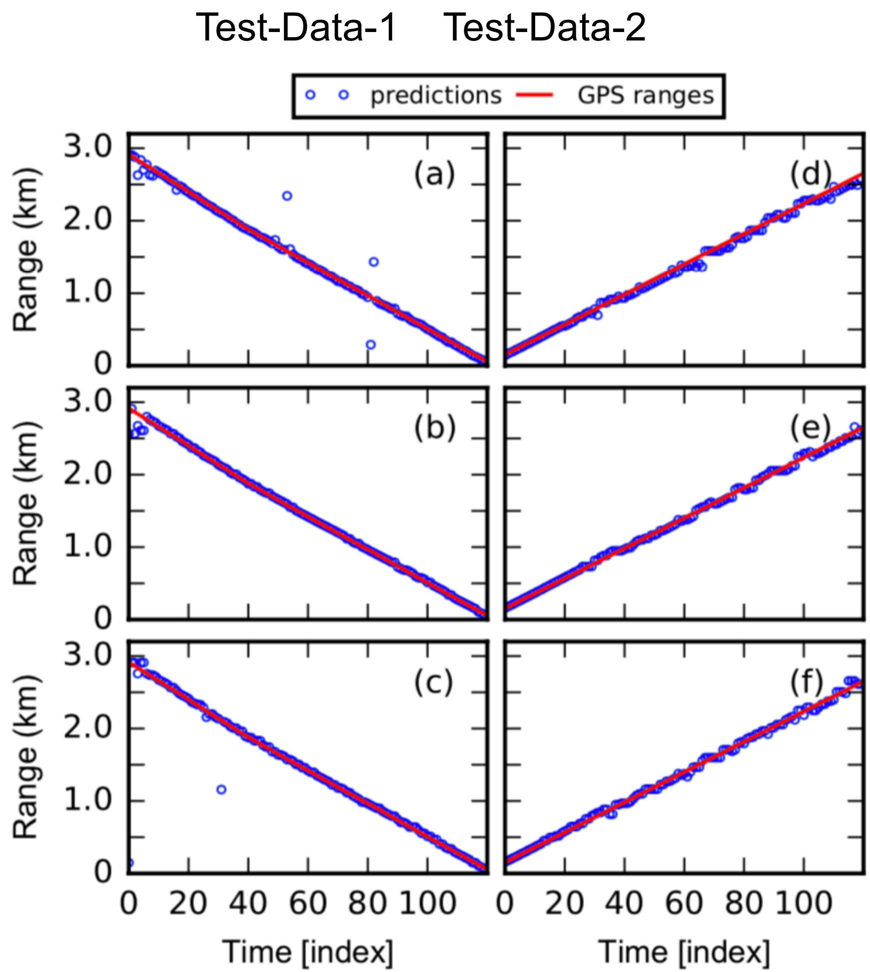
ML source range classification

Range predictions on Test-Data-1 (a, b, c) and Test-Data-2 (d, e, f) by FNN, SVM and RF for 300–950Hz with 10Hz increment, i.e., 66 frequencies.

(a),(d) FNN classifier,

(b),(e) SVM classifier,

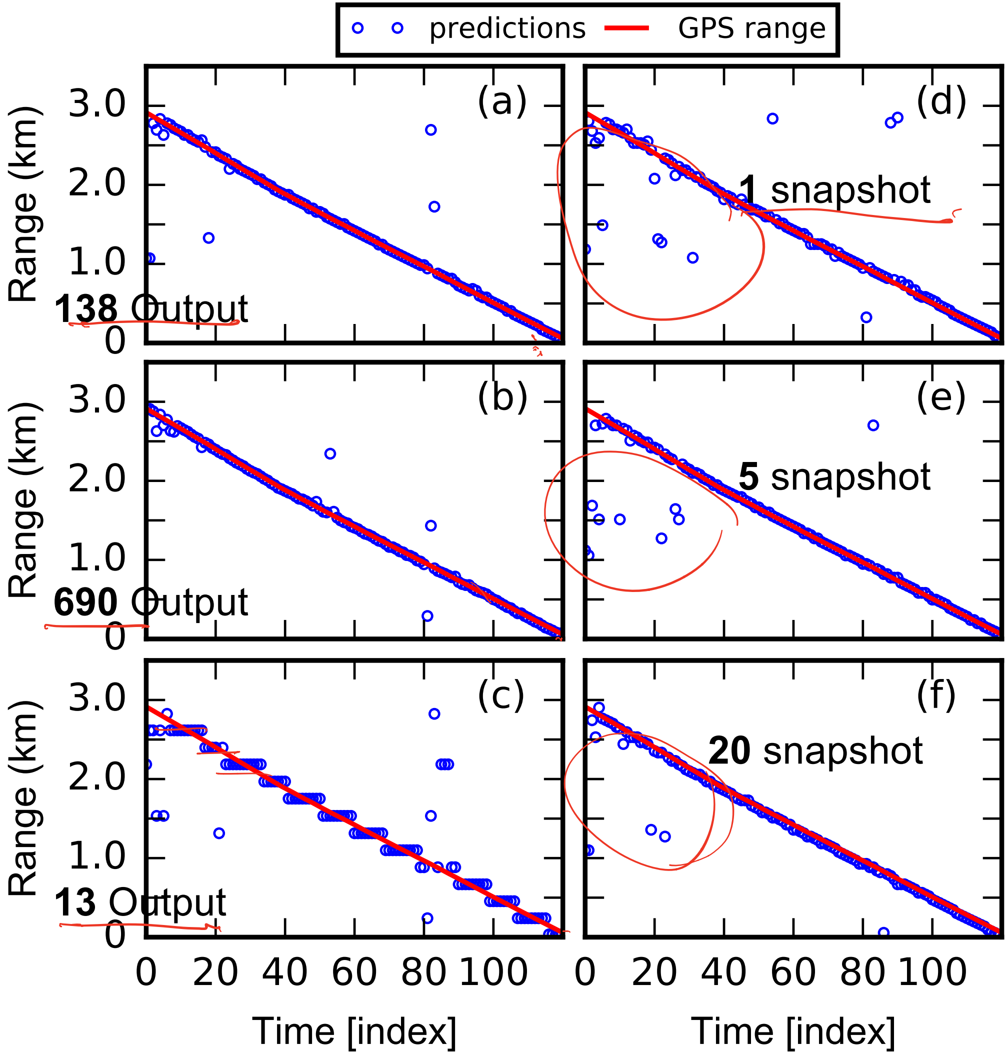
(c),(f) RF classifier.



Other parameters: FNN

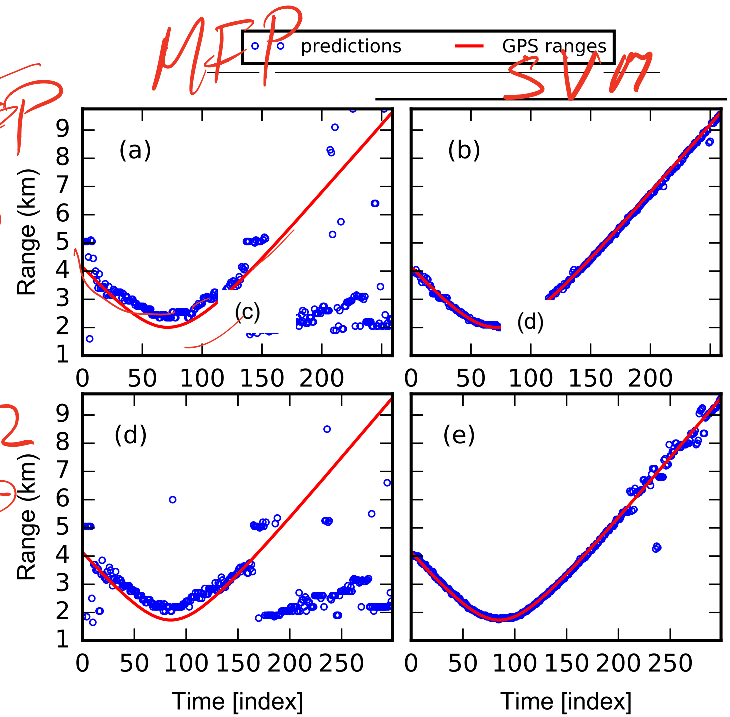
Conclusion

- Works better than MFP
- Classification better than regression
- FNN, SVM, RF works.
- Works for:
    - multiple ships,
  - Deep/shallow water
  - Azimuth from VLA



So far…

- Can machine learning learn a nonlinear noise-range relationship?
  - **Yes:** *Niu et al. 2017, "Source localization in an ocean waveguide using machine learning."*
- We can use different ships for training and testing ?
  - **Yes**: *Niu et a. 2017, "Ship localization in Santa Barbara Channel using machine learning classifiers."* (see figure)



Ship range localization using (a,c) ***MFP*** *and (b,d)* ***SVM*** *(rbf kernel).*

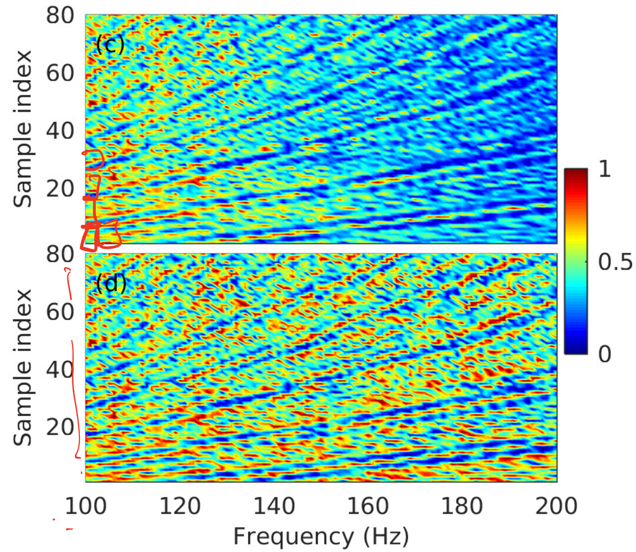
NN, SVM, and random forest
Perform about similar

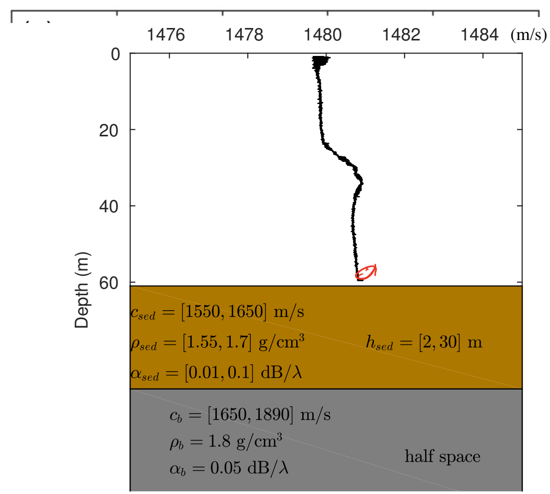
60s Science
Scientfic Am

Can we use CNN instead of FNN?

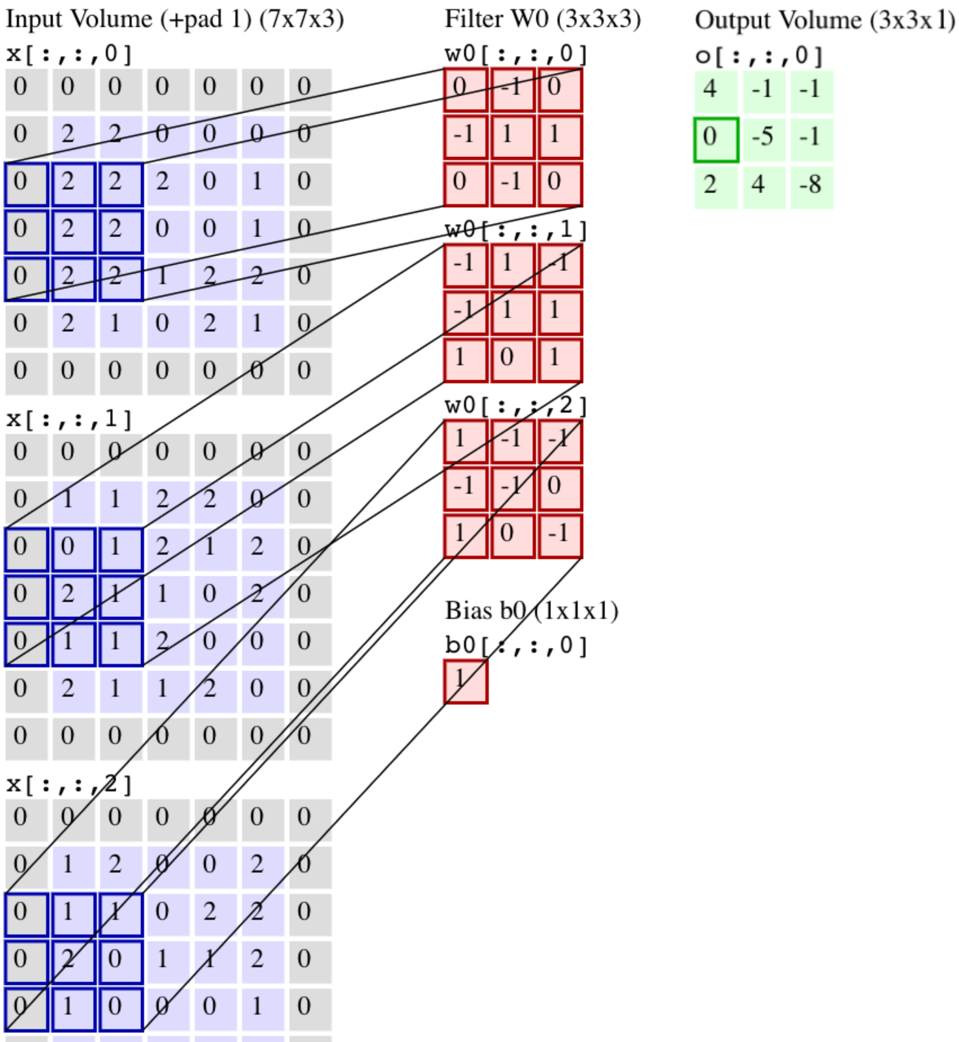
CNN uses much less weights!

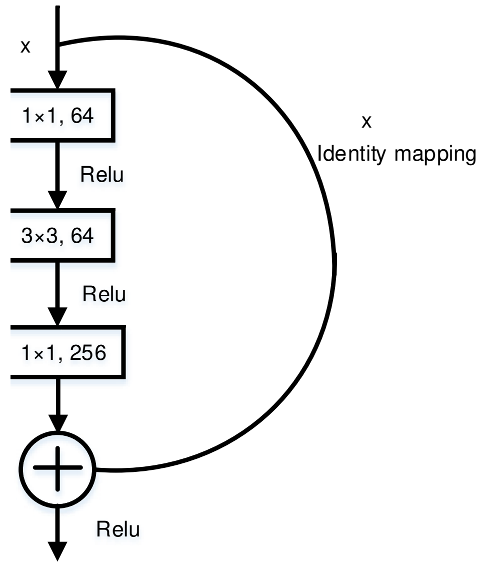
CNN relies on local features

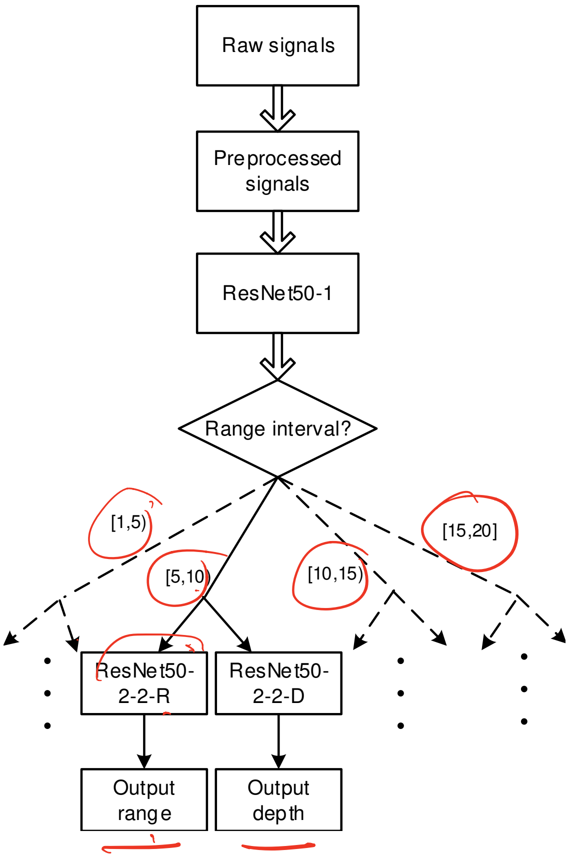




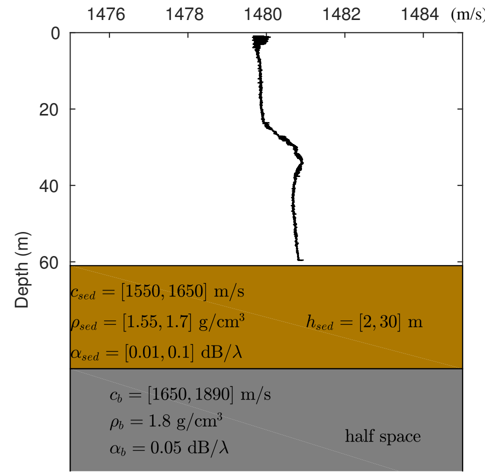
Rsnet and CNN for range estimation



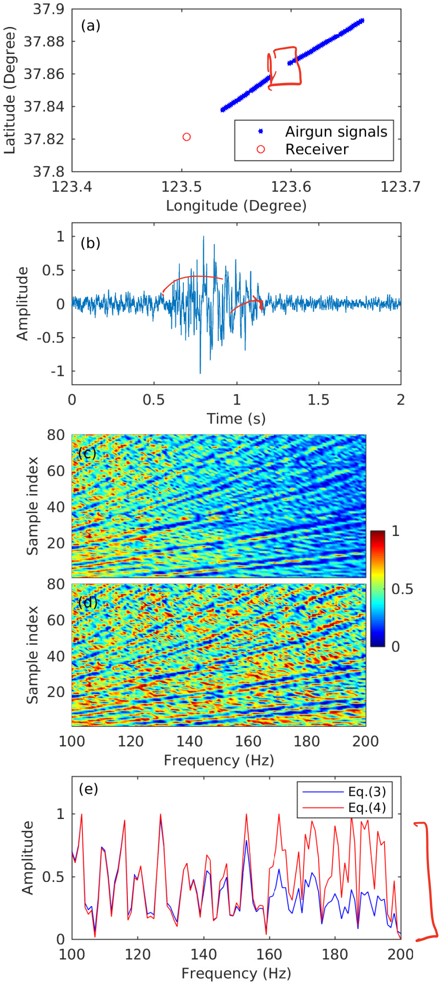


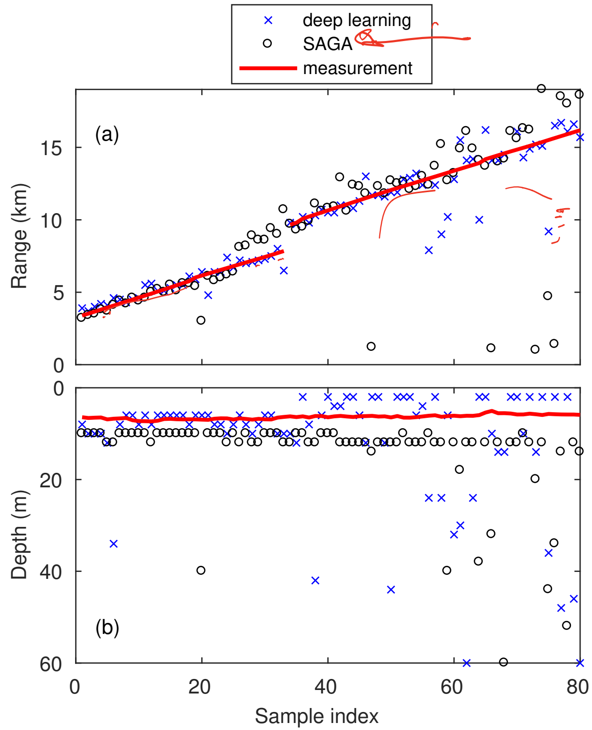


Raw signals
Preprocessed signals
ResNet50-1
Range interval?
[1,5)
[5,10)
[10,15)
[15,20]
ResNet50-2-2-R
ResNet50-2-2-D
Output range
Output depth



1476
1478
1480
1482
1484
(m/s)
0
20
40
60
Depth (m)
$c_{sed} = [1550, 1650]$ m/s
$\rho_{sed} = [1.55, 1.7]$ g/cm^3
$h_{sed} = [2, 30]$ m
$\alpha_{sed} = [0.01, 0.1]$ dB/λ
$c_b = [1650, 1890]$ m/s
$\rho_b = 1.8$ g/cm^3
$\alpha_b = 0.05$ dB/λ
half space


(a)
Latitude (Degree)
Longitude (Degree)
37.9
37.88
37.86
37.84
37.82
37.8
123.4
123.5
123.6
123.7
Airgun signals
Receiver
(b)
Amplitude
Time (s)
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
(c)
(d)
Sample index
80
60
40
20
1
0.5
0
100
120
140
160
180
200
Frequency (Hz)
(e)
Eq.(3)
Eq.(4)
Amplitude
Frequency (Hz)


deep learning
SAGA
measurement
(a)
Range (km)
15
10
5
0
(b)
Depth (m)
0
20
40
60
0
20
40
60
80
Sample index

Conventional Beamforming

snapshots

$$B(\theta_m) = \sum_{l=1}^{L} |\mathbf{w}^H(\theta_m)\mathbf{p}_l|^2 = \sum_{l=1}^{L} \mathrm{Tr}\{\mathbf{w}^H(\theta_m)\mathbf{p}_l\mathbf{p}_l^H\mathbf{w}(\theta_m)\}$$

- Linearize:

$$= \sum_{l=1}^{L} \mathrm{Tr}\{\mathbf{w}(\theta_m)\mathbf{w}^H(\theta_m)\mathbf{p}_l\mathbf{p}_l^H\} = L\,\mathrm{Tr}\{\mathbf{W}^H\mathbf{P}\}.$$

$$B(\theta_k) = Tr\left\{(\mathbf{W}^R)^T\mathbf{P}^R + (\mathbf{W}^I)^T\mathbf{P}^I\right\} = tr\{W P\}$$

$$= vec(\mathbf{W}^R)^T vec(\mathbf{P}^R) + vec(\mathbf{W}^I)^T vec(\mathbf{P}^I)$$

$$B(\theta_k) = \mathbf{w}_{eff}(\theta_k)^T \mathbf{p}_{eff}$$

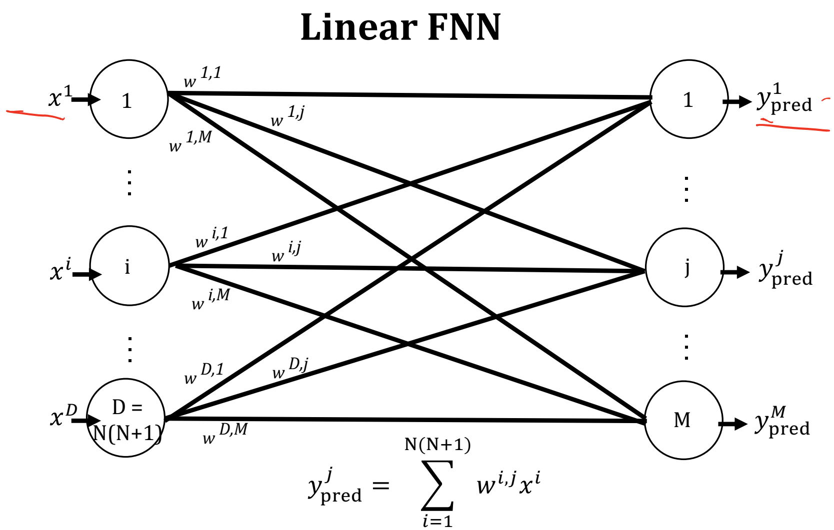
- Real, linear beam-function with vector inputs

$$\mathbf{w}_{eff}(\theta_k) = \left[vec(\mathbf{W}^R), vec(\mathbf{W}^I)\right]$$

$$\mathbf{p}_{eff}(\theta_k) = \left[vec(\mathbf{P}^R), vec(\mathbf{P}^I)\right]$$

Beamforming is now a linear problem in **weights**
with real-valued input from sample covariance matrix

Machine Learning: Feed-forward neural network



- $\boldsymbol{x}_i = \boldsymbol{p}_{eff}(\theta_i)$ (data covariance)
- $y^j{}_{i,true} = \begin{cases} 1, & j = m \\ 0, & j \neq m \end{cases}, \quad j = 1, \ldots, M.$
- $y^j{}_{i,true}$: output for class *m*
- FNN linear model: $y^j{}_{i,pred} = \boldsymbol{w}_m^T \boldsymbol{x} = \sum_{n=1}^{2N^2} w_{nm}^T x_{i,n}$
- No hidden layer

Machine Learning: Feed-forward neural network

- Encourage similarity between true and predicted outputs

- Recall,

$$y^j{}_{i,true} = \begin{cases} 1, & j = m \\ 0, & j \neq m \end{cases}, \qquad j = 1, \ldots, M.$$

- Cost function:

$$\underset{\boldsymbol{w}_i}{\operatorname{argmin}} - \sum_{i=1}^{T} \sum_{m=1}^{M} y_{i,true}^{m} y_{i,pred}^{m}$$
$$= \underset{\boldsymbol{w}_i}{\operatorname{argmin}} - \sum_{i=1}^{T} \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x}_i$$

Machine Learning and Conventional Beamforming

- Compare:

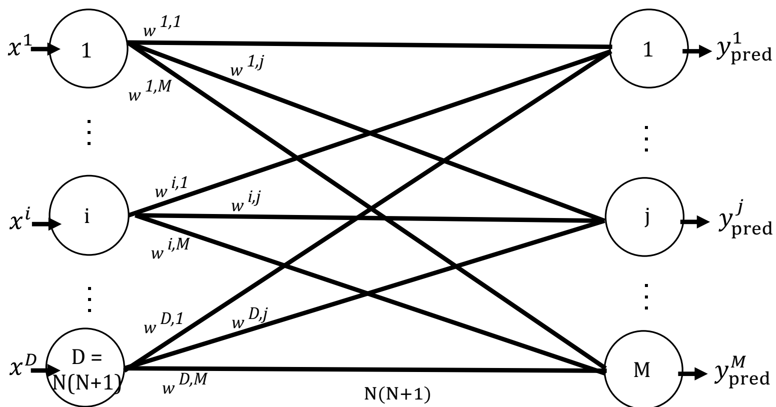
CBF → ***FNN***

$$\underset{\boldsymbol{w}_{eff}}{\operatorname{argmax}} -\boldsymbol{w}_{eff}(\theta_m)^T \boldsymbol{p}_{eff}$$

$$\underset{\boldsymbol{w}_i}{\operatorname{argmin}} -\boldsymbol{w}_i^T \boldsymbol{x}_i \quad \forall \boldsymbol{i}$$

- Assume $\boldsymbol{x}_i = \boldsymbol{p}_{eff}$ $\quad \boldsymbol{w}_i = \boldsymbol{w}_{eff}(\theta_m)$
- **Thus, linear FNN converges to CBF if trained on plane waves**

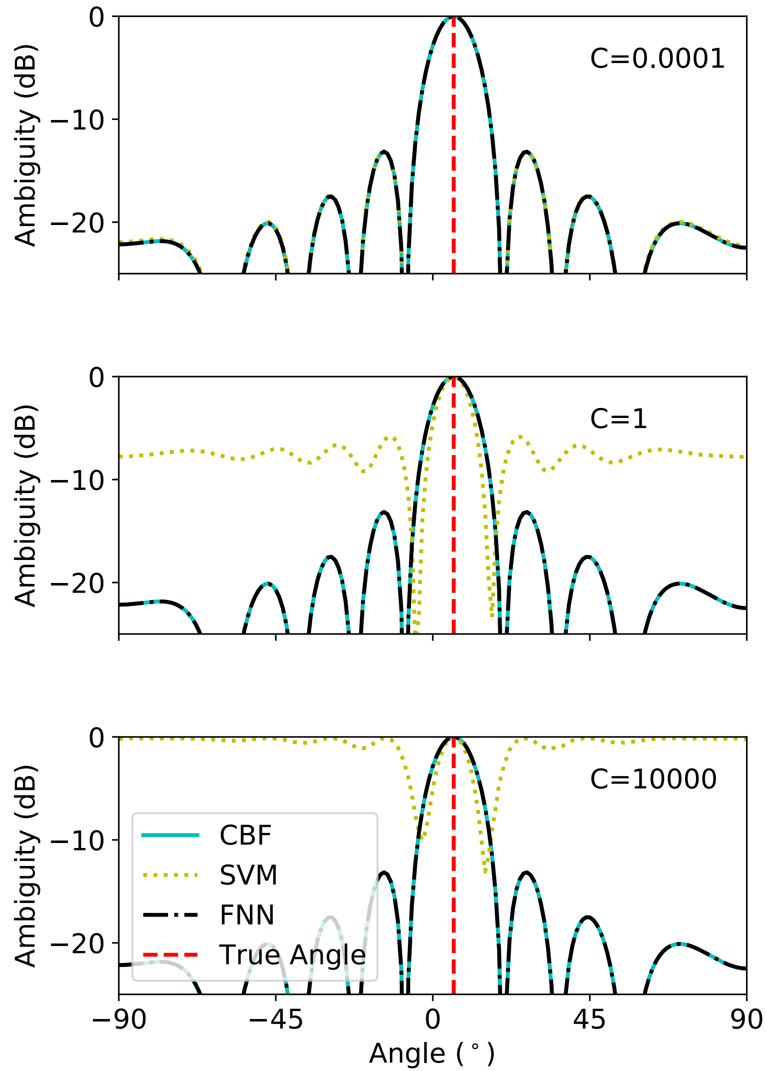
Linear FNN



$$y^j_{\text{pred}} = \sum_{i=1}^{N(N+1)} w^{i,j} x^i$$

Conventional Beamforming and Machine Learning

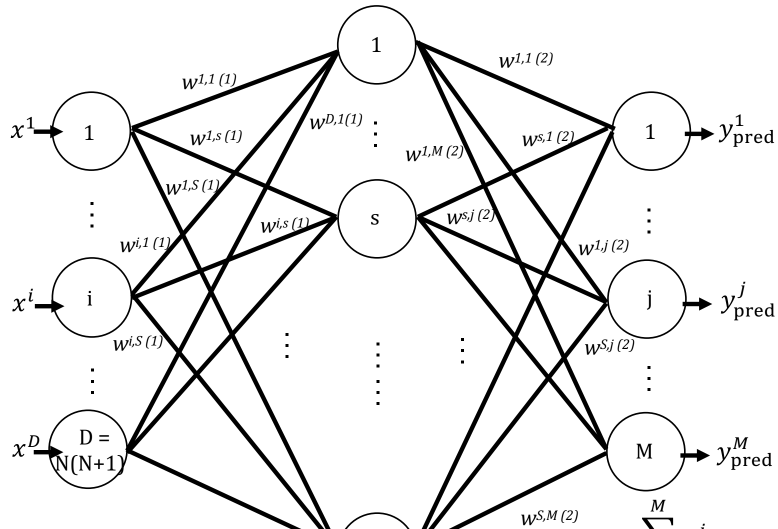
- Conventional beamforming (CBF) is written as linear function
- 2-Layer Feed-forward neural network (FNN), same linear function
- Support Vector Machine (SVM) is a linear classifier, differs from CBF



Ozanich 2019?

Fully connected FNN

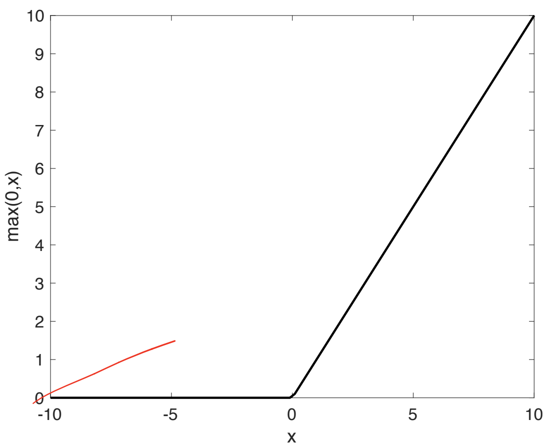
Nonlinear FNN



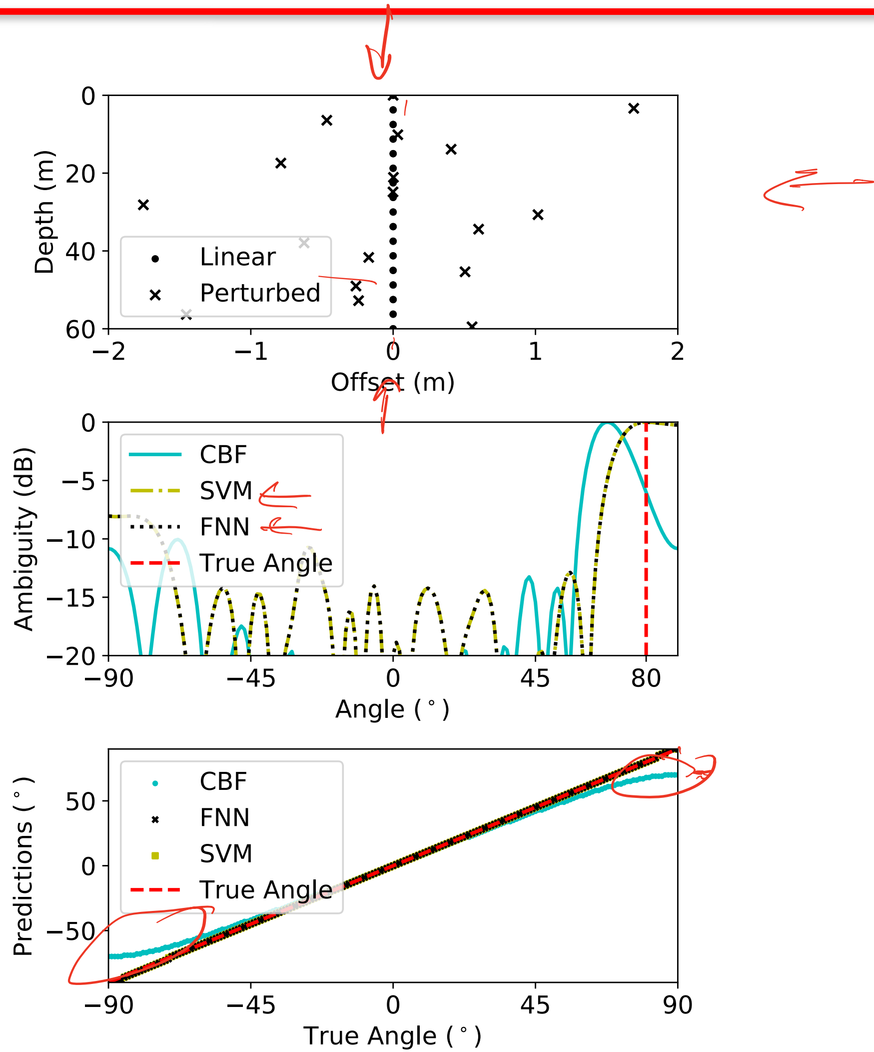
$$\sum_{j=1}^{M} y^j_{\text{pred}} = 1$$

$$a^s = \max\left(0, \sum_{i=1}^{N(N+1)} w^{i,s\,(1)} x^i\right) \qquad y^j_{\text{pred}} = \frac{e^{\sum_{s=1}^{S} w^{s,j\,(2)} a^s}}{\sum_{j=1}^{M} e^{\sum_{s=1}^{S} w^{s,j\,(2)} a^s}}$$

ReLu

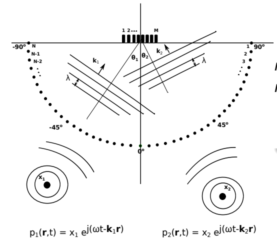


Perturbed array



Coherent vs incoherent sources

Two DOAs θ_1, θ_2
With sources $S_k = |S_k| e^{i\phi_k}$
Coherent source $\Delta\phi = \phi_2 - \phi_1 = 0$
Incoherent $\Delta\phi = \phi_2 - \phi_1 = \mathrm{U}(-\pi, \pi)$



$x \in \mathbb{C},\ \theta \in [-90°, 90°]$

$\mathbf{k} = -\frac{2\pi}{\lambda}\sin\theta,\ \lambda$:wavelength

$$y_m = \sum_n x_n e^{j\frac{2\pi}{\lambda} r_m \sin\theta_n}$$

$m \in [1, \cdots, M]$: sensor
$n \in [1, \cdots, N]$: look direction

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

$$\mathbf{y} = [y_1, \cdots, y_M]^T, \quad \mathbf{x} = [x_1, \cdots, x_N]^T$$

$$\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N]$$

$$\mathbf{a}_n = \frac{1}{\sqrt{M}}[e^{j\frac{2\pi}{\lambda} r_1 \sin\theta_n}, \cdots, e^{j\frac{2\pi}{\lambda} r_M \sin\theta_n}]^T$$

Forming the sample covariance matrix

$$\boldsymbol{P} = \boldsymbol{y}\boldsymbol{y}^H = \boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{A}^H$$

Or

$$P^{\mathrm{R}}_{n,m} = \frac{1}{N}\Big[\sum_{k=1}^{2} |S_k|^2 \cos\Big(\frac{\omega}{c}(n-m)\ell\sin(\theta_k)\Big) + 2S_1S_2\cos\Big(\frac{\omega}{c}(n\sin(\theta_1) - m\sin(\theta_2))\ell + \Delta\phi\Big)\Big]$$

$$P^{\mathrm{I}}_{n,m} = \frac{1}{N}\Big[\sum_{k=1}^{2} |S_k|^2 \sin\Big(\frac{\omega}{c}(n-m)\ell\sin(\theta_k)\Big)\Big] \qquad (64)$$

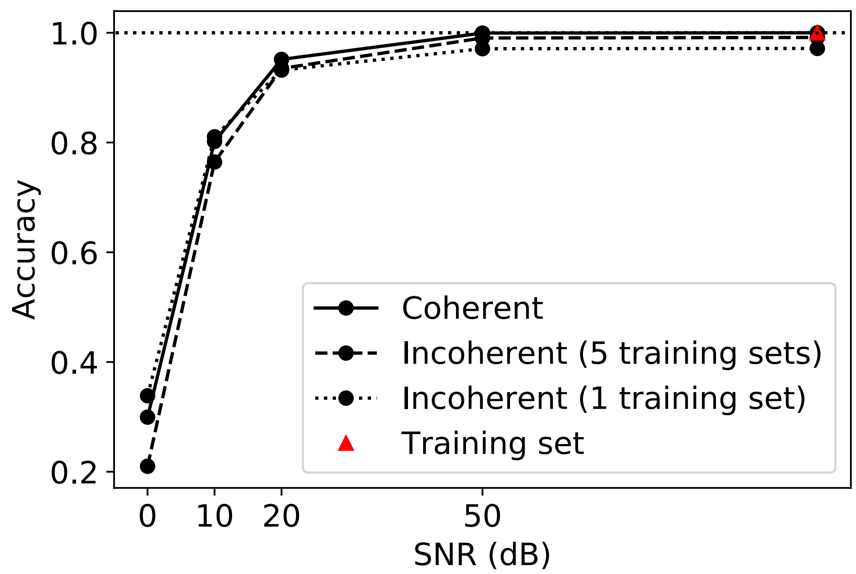
Coherent vs incoherent sources

$$P^{\mathrm{R}}_{n,m} = \frac{1}{N}\Big[\sum_{k=1}^{2}|S_k|^2\cos\Big(\frac{\omega}{c}(n-m)\ell\sin(\theta_k)\Big)+$$

$$\sum_{i=1}^{L}2S_1S_2\cos\Big(\frac{\omega}{c}(n\sin(\theta_1)-m\sin(\theta_2))\ell+\Delta\phi_i\Big)\Big]$$

$$P^{R}_{n,m} \rightarrow \frac{1}{N}\Big[\sum_{k=1}^{2}|S_k|^2\cos\Big(\frac{\omega}{c}(n-m)\ell\sin(\theta_k)\Big)\Big], \qquad (65)$$

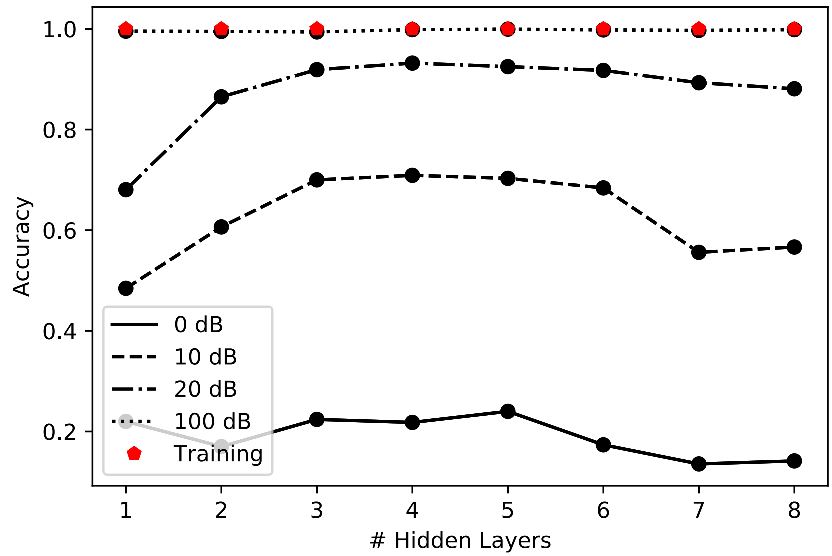
$$L\rightarrow\infty, \quad \Delta\phi_i\in\mathcal{U}\{\pi,\pi\}.$$



FNN hidden layers

- Two DOAs θ_1, θ_2: 0-180 deg.
- Training all combinations
- Validation 1000 Uniformly random DOA

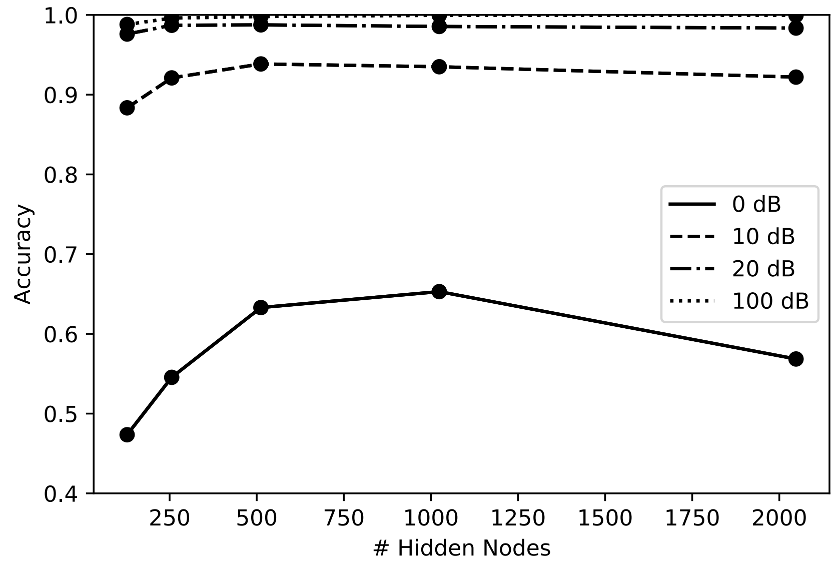
- Each Hidden layers add S(S+1)
- 512 nodes in each layer



FNN hidden layers

- Two DOAs θ_1, θ_2: 0-180 deg.
- Training all combinations
- Validation 1000 Uniformly random DOA

- Each Hidden layers add O(S)



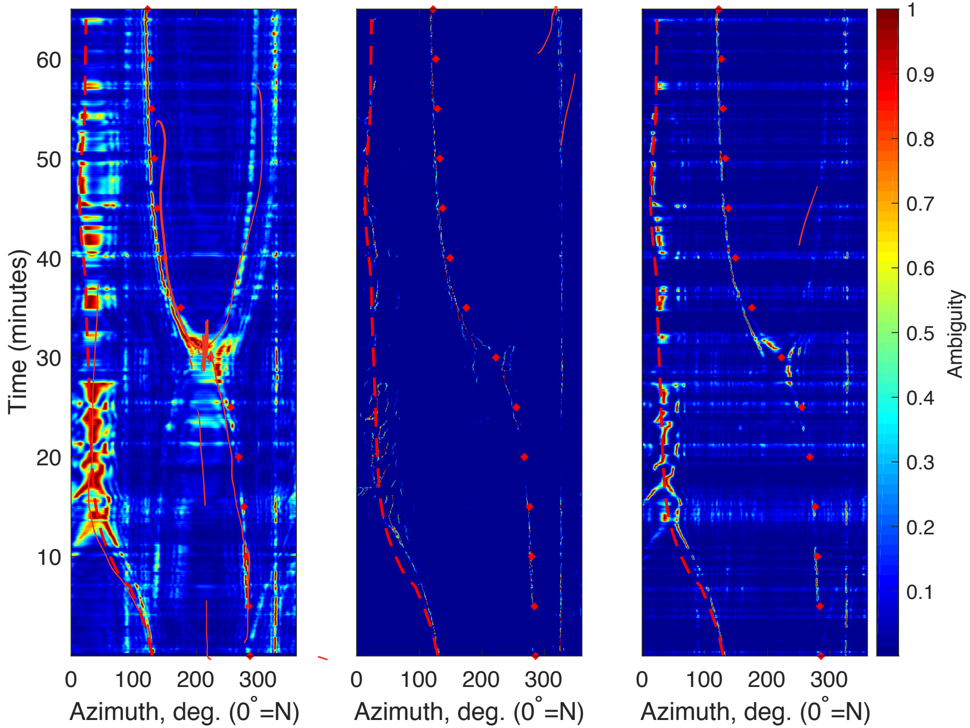
Localizing two sources from SW06

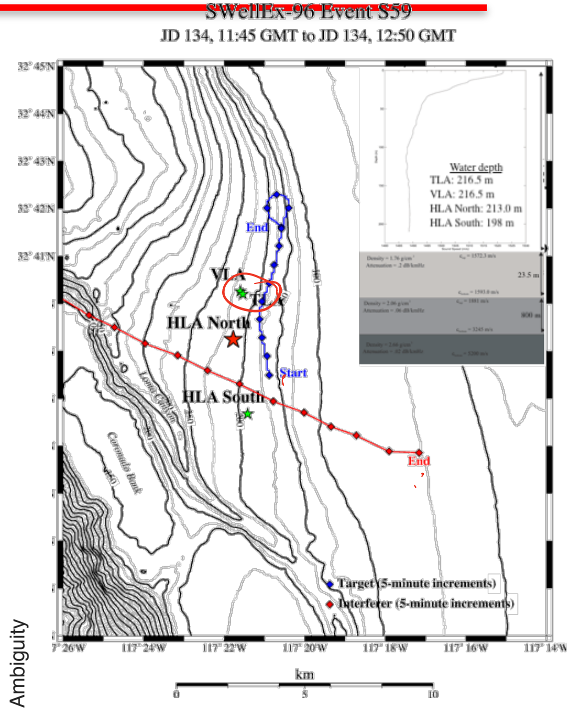
6m and 54 m source depth

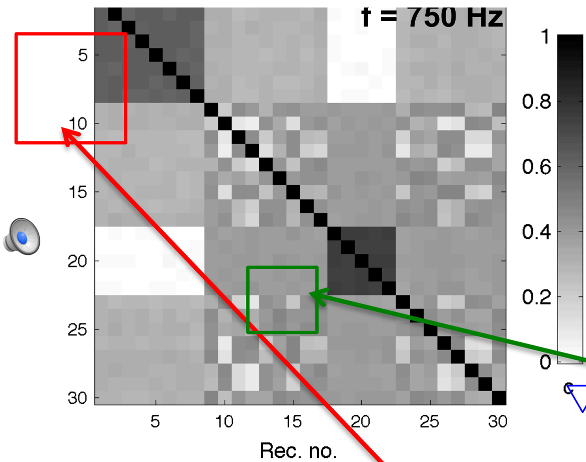
CBF

SBL

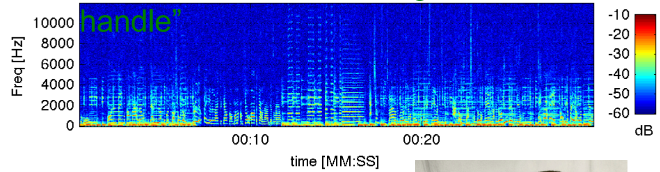
FNC



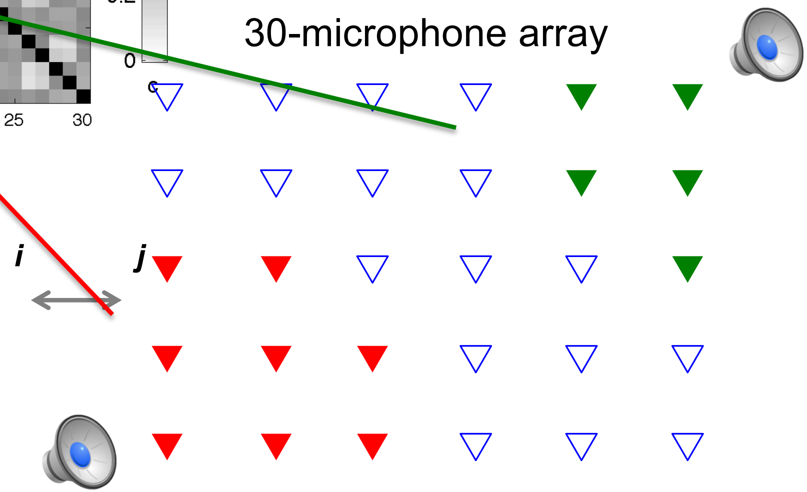




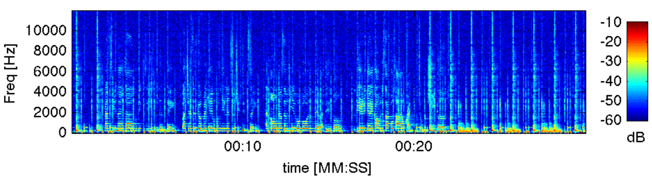
Location 1: Otis Redding - "Hard to handle"



30-microphone array



Location 1: Prince - "Sign o' the times"



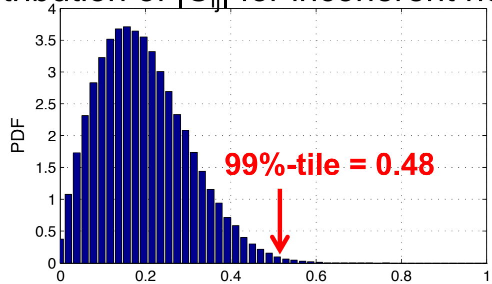
Spectral coherence between ***i*** and ***j***

$$\hat{C}_{ij}(f) = \frac{1}{N} \sum_{t=1}^{N} X_i(f,t) \cdot \bar{X}_j(f,t)$$

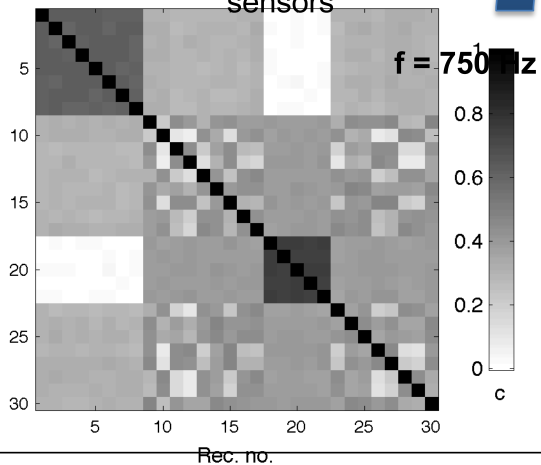
(Normalization: |X(f,t)|^2=1)

Each group is spatially coherent. But no temporal correlation between groups (i.e. different source)

Distribution of $|C_{ij}|$ for incoherent noise

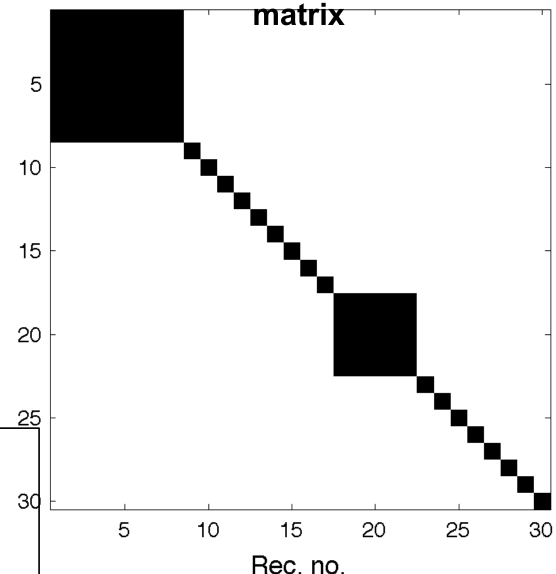


Magnitude of spectral **coherence** for 30 sensors



$|C_{ij}|$

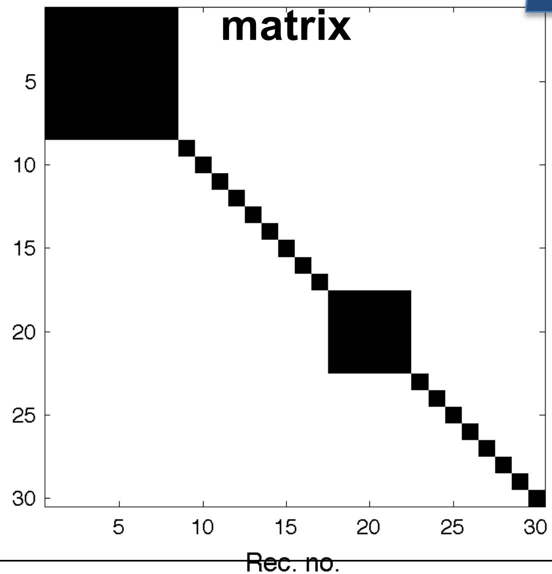
Statistically significant entries **=> Connectivity matrix**



- Each sensor is a **node** in the graph.
- If **nodes** *i* and *j* are significantly correlated $|C_{ij}|>\xi$, then they share an ***edge***.

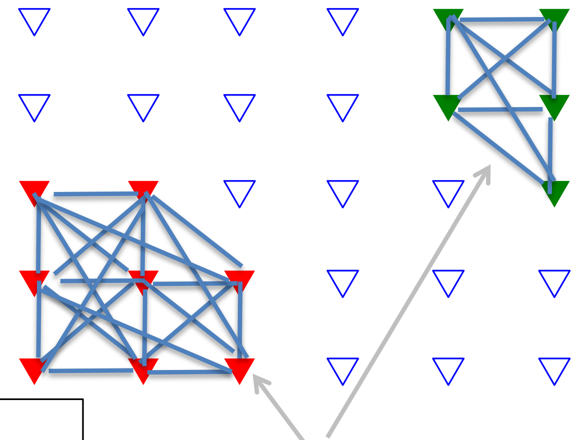
=> Two sources in the network

Statistically significant entries **=> Connectivity matrix**



- Each sensor is a **node** in the graph.
- If **nodes** *i* and *j* are significantly correlated $|C_{ij}|>\xi$, then they share an ***edge***.
- A **subgraph** has high spatial coherence across a subarray (=> likely a source nearby).

Graph with 30 nodes



Connected subgraphs:

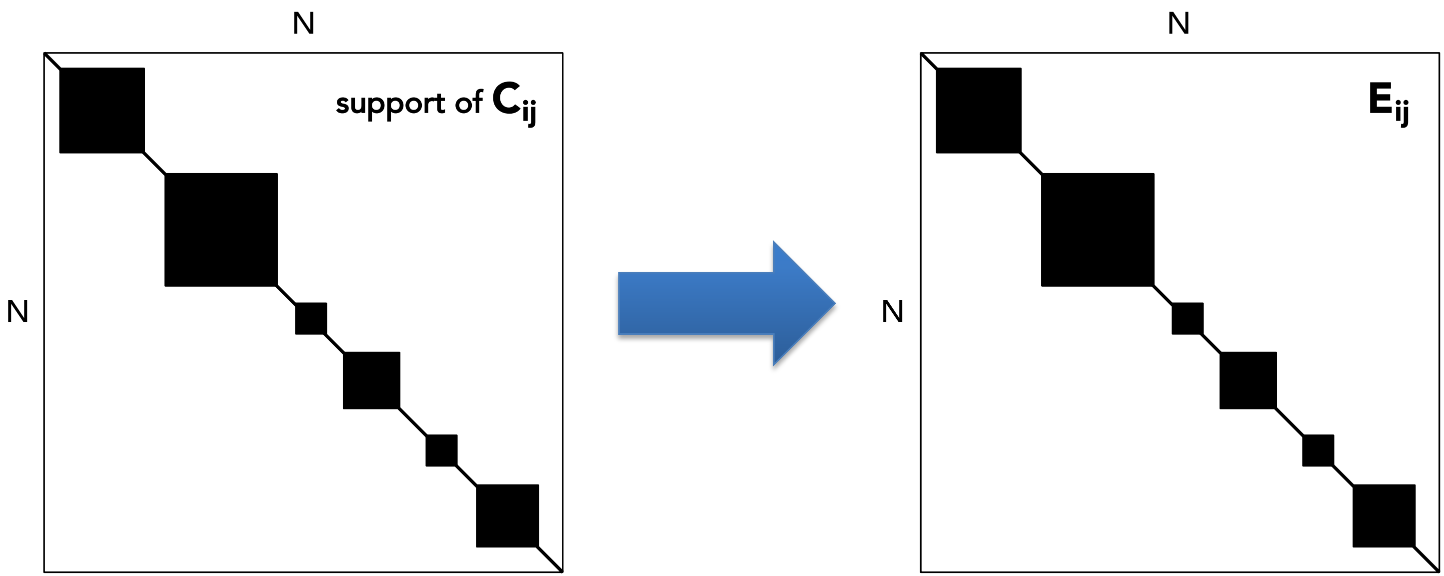
5 nodes and 9 edges

8 nodes and 20 edges

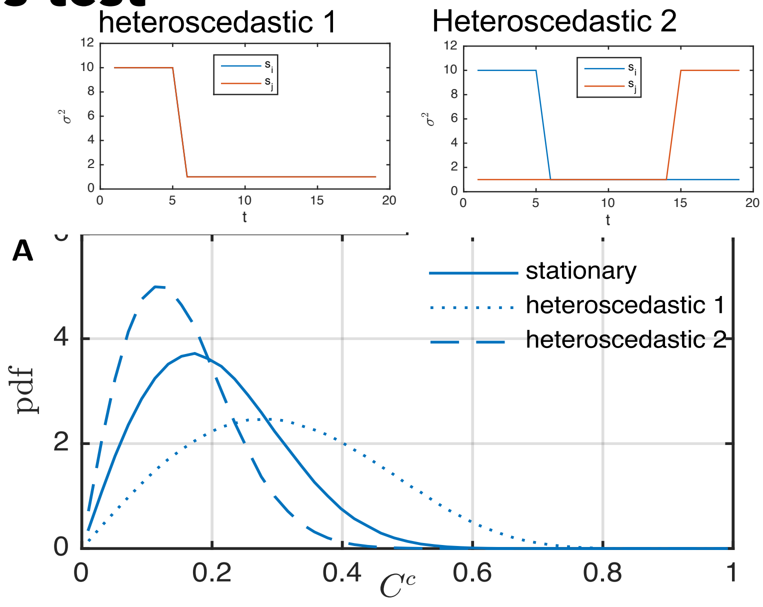
Asymptotic case

Reinterpret C_{ij} as connectivity matrix E_{ij} of network with N vertices.

$$E_{ij}^0 = \begin{cases} 1 & \text{if } \widehat{C}_{ij} > c_\alpha \\ 0 & \text{otherwise,} \end{cases}$$



Robust Coherence hypothesis test

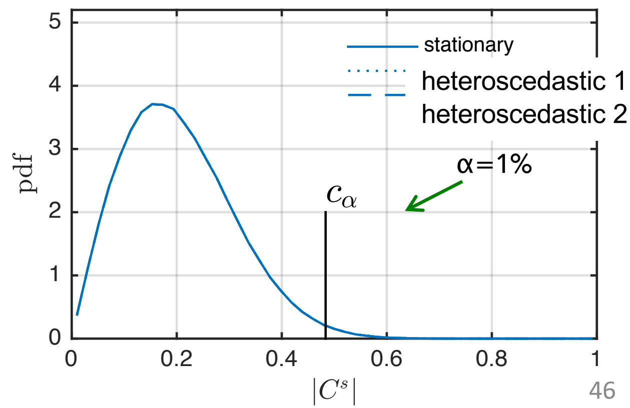


Conventional coherence

$$\widehat{C}_{ij}^{c} = \left| \frac{\frac{1}{M}\sum_{m=0}^{M-1} x_i(m) x_j^*(m)}{\left(\frac{1}{M}\sum_{m=0}^{M-1} |x_i(m)|^2\right)^{1/2}\left(\frac{1}{M}\sum_{m=0}^{M-1} |x_j(m)|^2\right)^{1/2}} \right|,$$

Phase-only coherence

$$\widehat{C}_{ij} = \left| \frac{1}{M} \sum_{m=0}^{M-1} \frac{x_i(m)}{|x_i(m)|} \frac{x_j^*(m)}{|x_j(m)|} \right|.$$

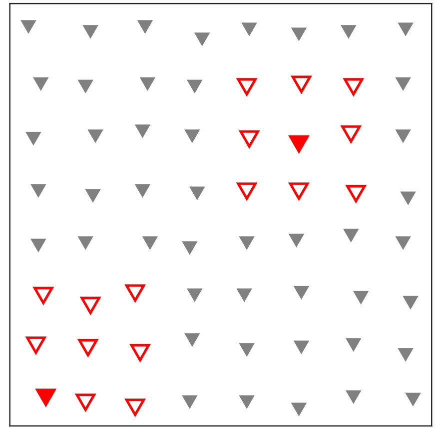


Robust to heteroscedastic noise

"Noise-only" network

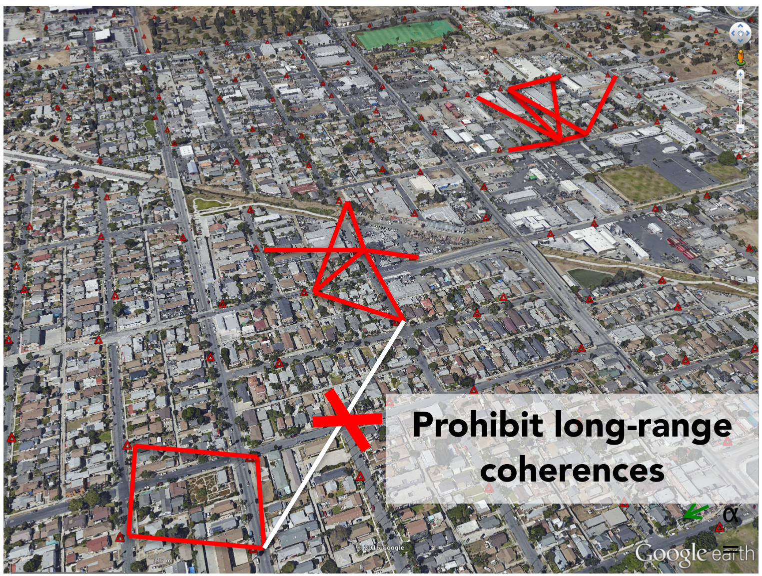
If α>2.5/(N−1) the network almost surely has a giant connected component, i.e., most sensors are linked *[Erdös & Rényi, 1959]*.

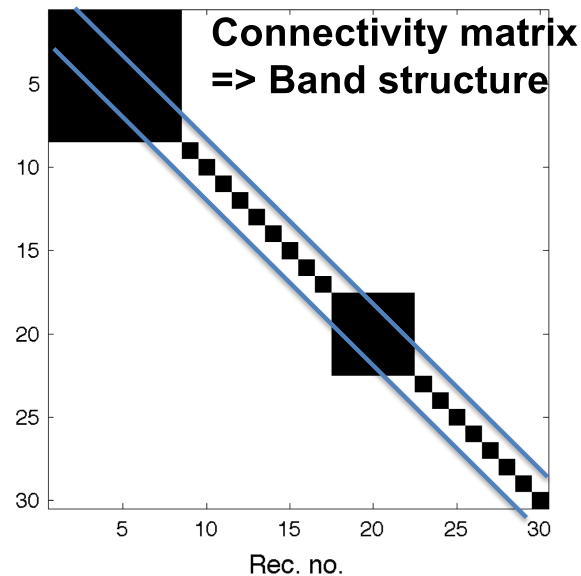
Bad for cluster search!



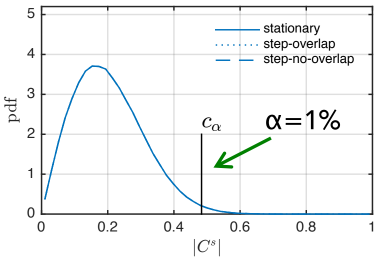
We limit this by testing just the **8-nearest neighbors:**

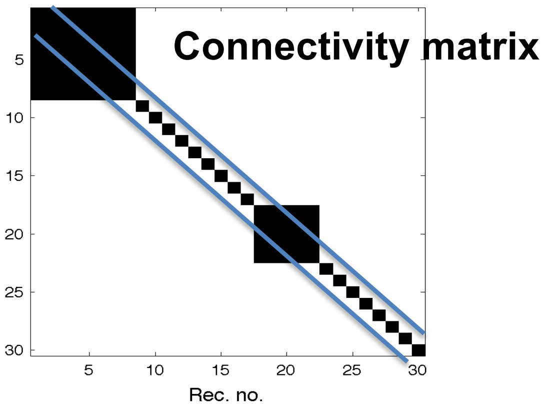
$$E_{ij} = \begin{cases} 1 & \text{if } \widehat{C}_{ij} > c_\alpha \text{ and } i \in \mathrm{N}(j) \\ 0 & \text{otherwise,} \end{cases}$$

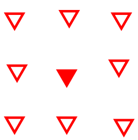


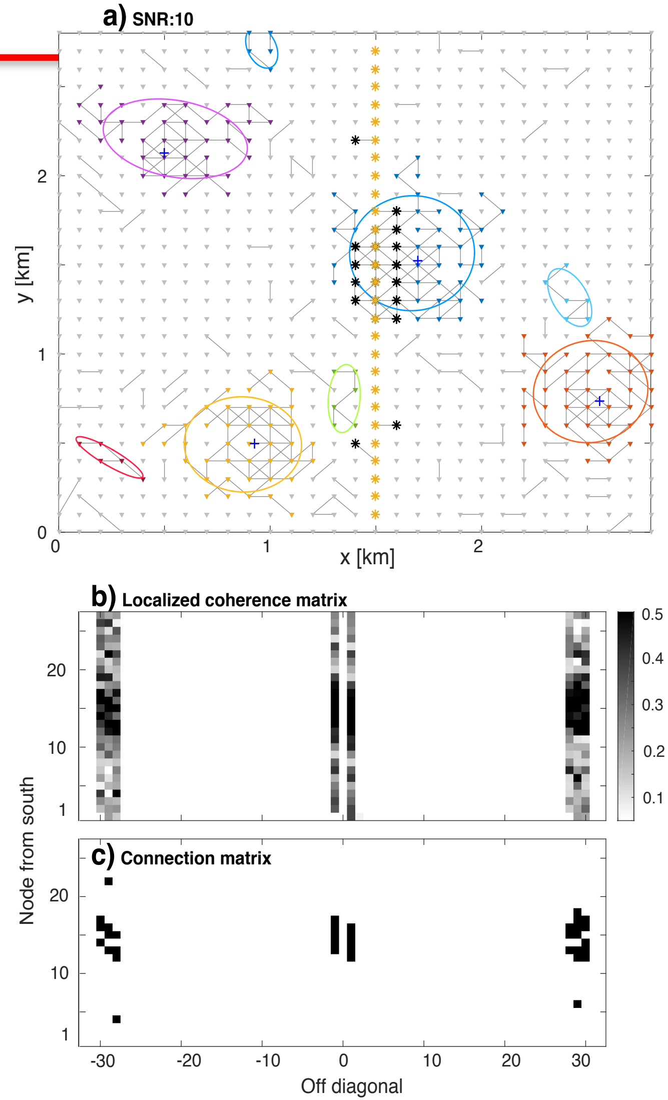


Simulation K=4, SNR=10





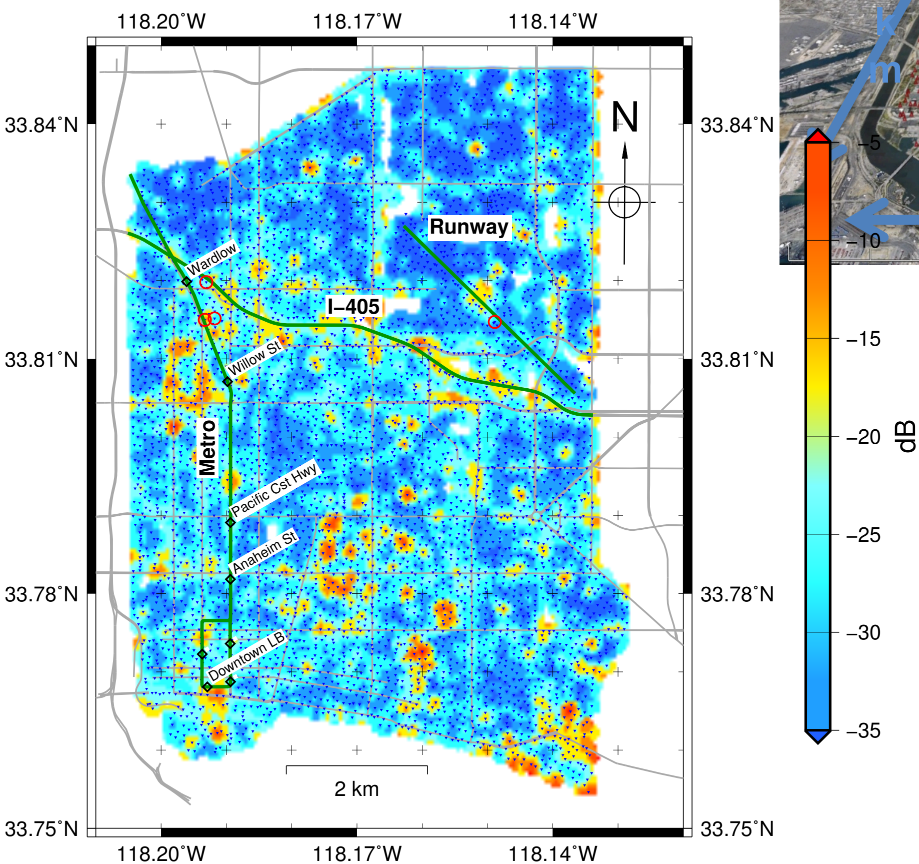


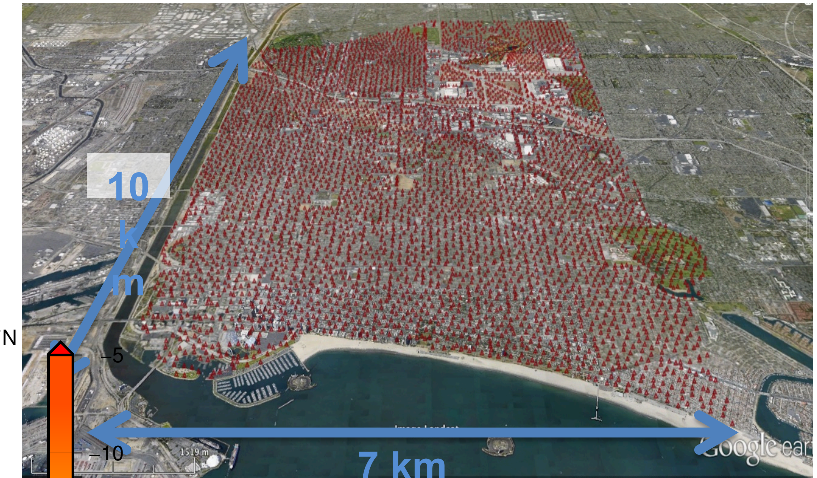


-A cluster is formed if

>4 nodes are connected with >4edges

Long Beach array





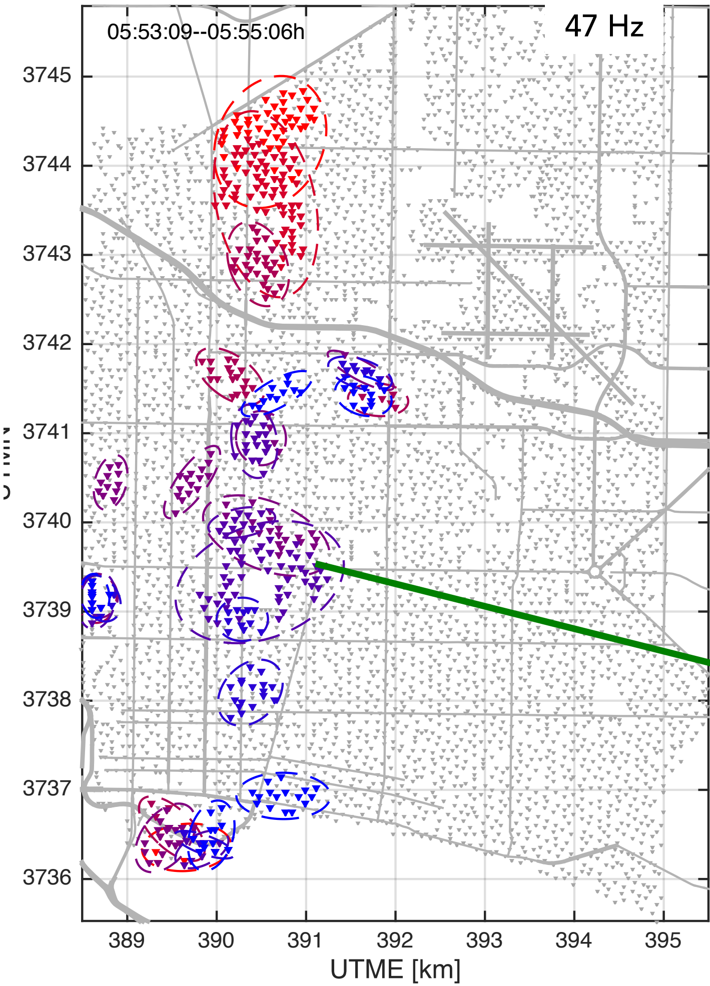
Thursday, March 10th

250 Hz sampling rate

FFT sample size 256 (≈1 sec)

Block-averaging over 19 windows

Window advances by 10 sec.



Helicopter rotor noise (seismo-acoustic coupling)

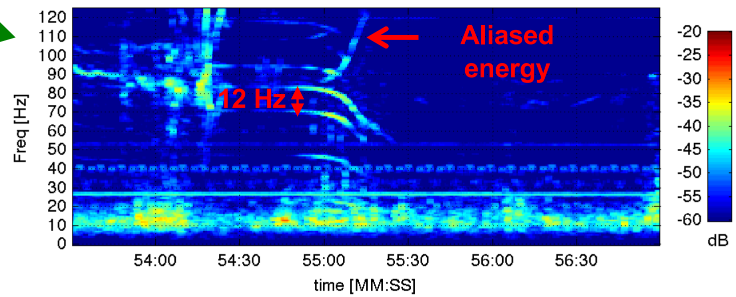
Several peaks consistent with helicopter rotor harmonics (20-100 Hz).

Doppler shift

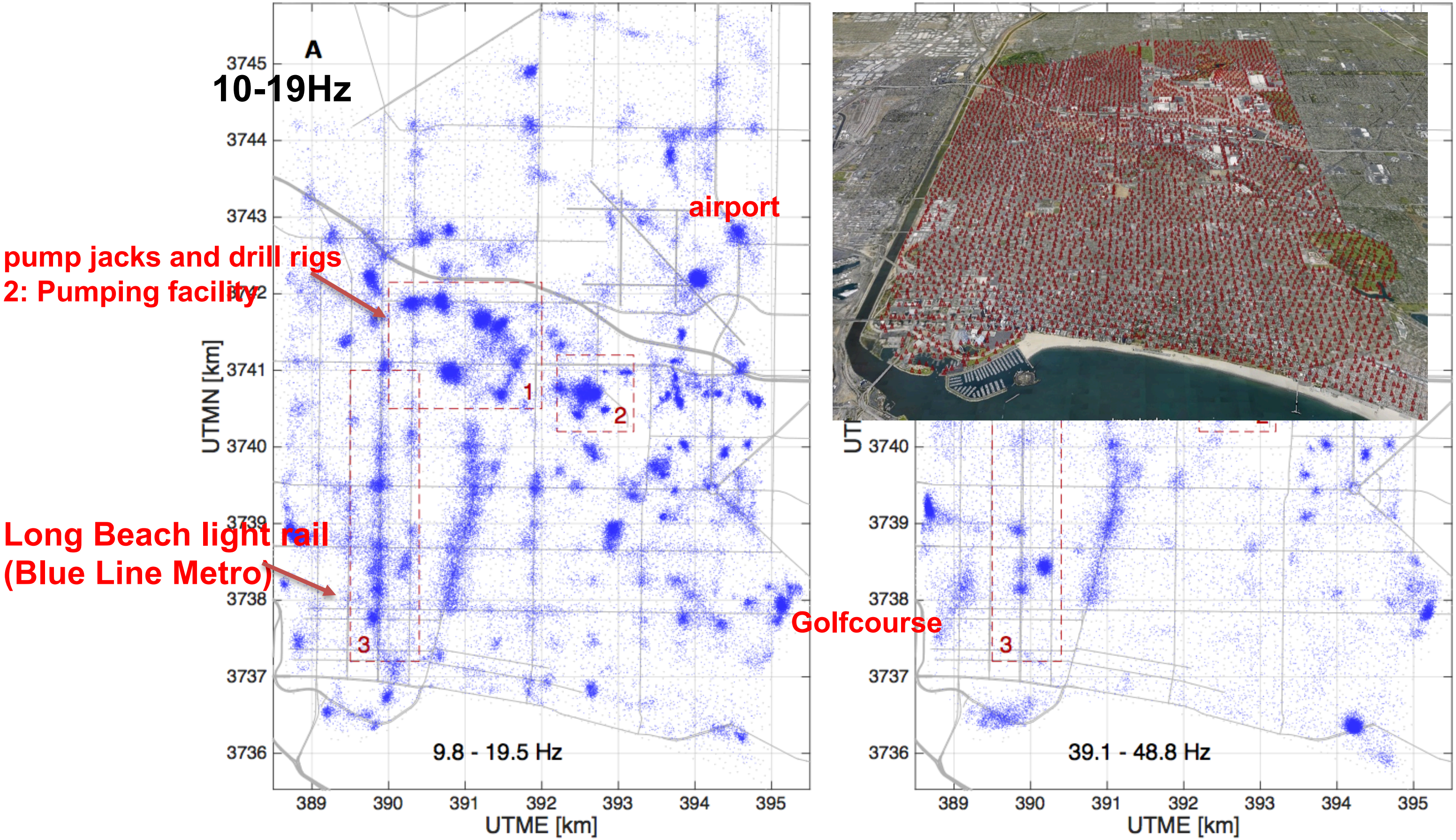
$f_{high}/f_{low}=(v_0+v)/(v_0-v)\approx1.4$ i.e. $v\approx250$ km/h

Speed over ground 7km/2min=210km/h

- ✓ Rotor frequencies
- ✓ Doppler frequency shift
- ✓ Movement in map



Clusters on March 10



Based on 9400 time windows x 10 frequency bins.
Each dot is the center of a cluster. 90% of the clusters cover <1.5% of the area.
Few false detections