Machine learning and applications to ocean acoustics

Peter Gerstoft,

http://noiselab.ucsd.edu/. Slides and 42-page review paper [Bianco 2019]

With help from Mike Bianco, Emma Ozanich, Haiqiang Niu, Kay Gemba, James Traer, Christoph Mecklenbrauker, Eliza Michalopoulou

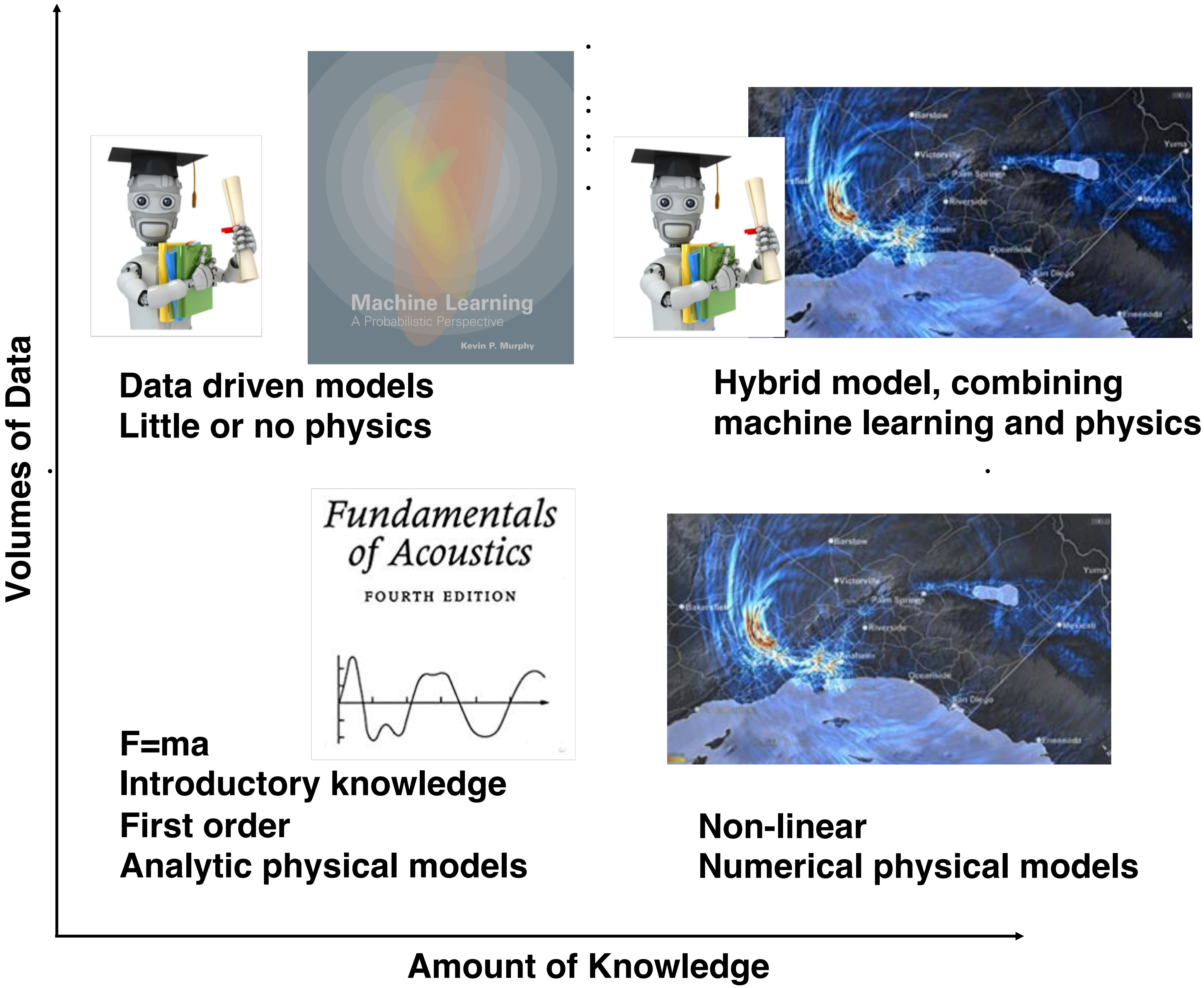
Machine leaning contains the mathematical tools we need to do **data science**

Can Machine Learning

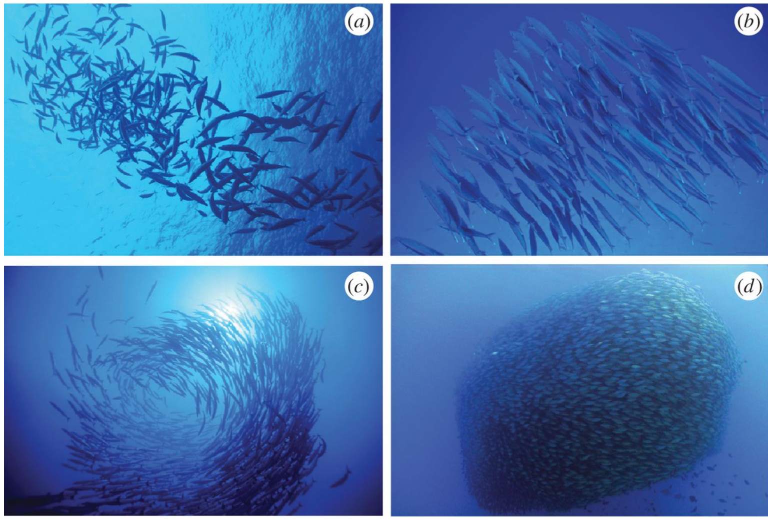
- Replace CTBTO/SONAR processing chain?
- Discover PDE (Partial differential equation) in video?
- Find sea mines?
- Design metamaterials?
- Predict earthquakes?
- Source location in the ocean waveguide w/o training?
- Replace 50 years of array processing?
- Learn the physical model (sound speed, temperature…)

Machine learning versus knowledge based

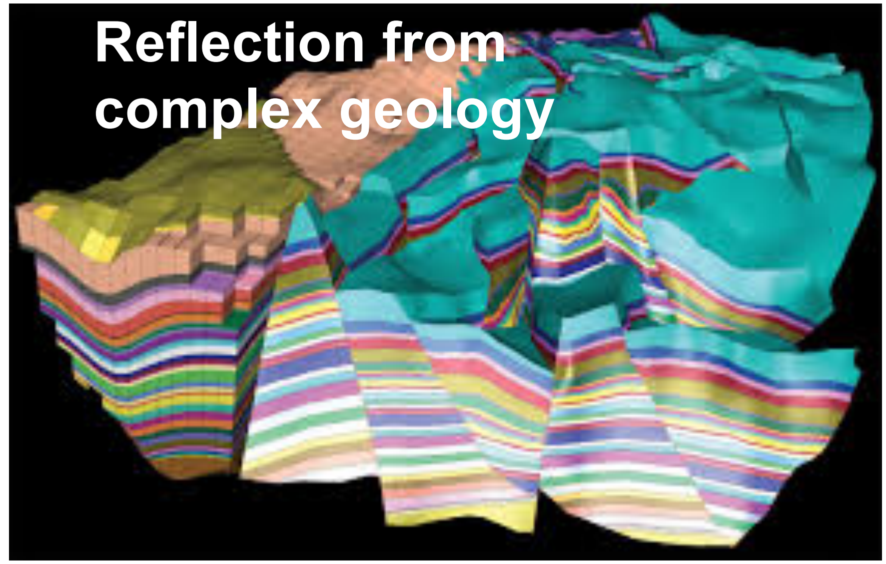
Acoustic insight can be improved by leveraging the strengths of both physical and ML-based, data-driven models.



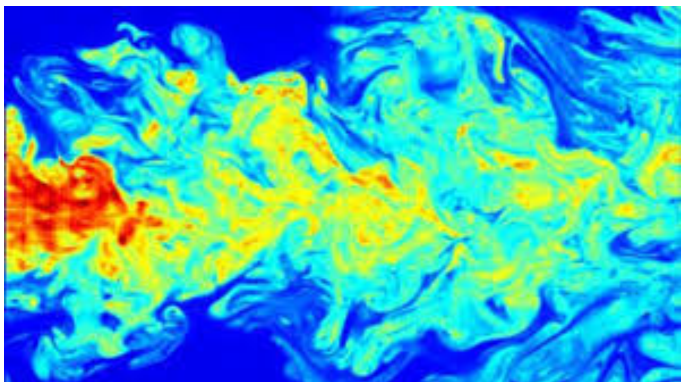
We can't model everything…



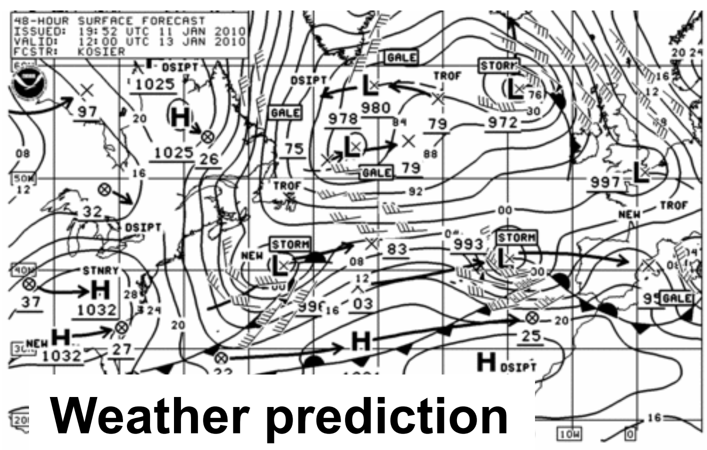
Back scattering from fish school



Reflection from complex geology



Predict acoustic field in turbulence



Weather prediction

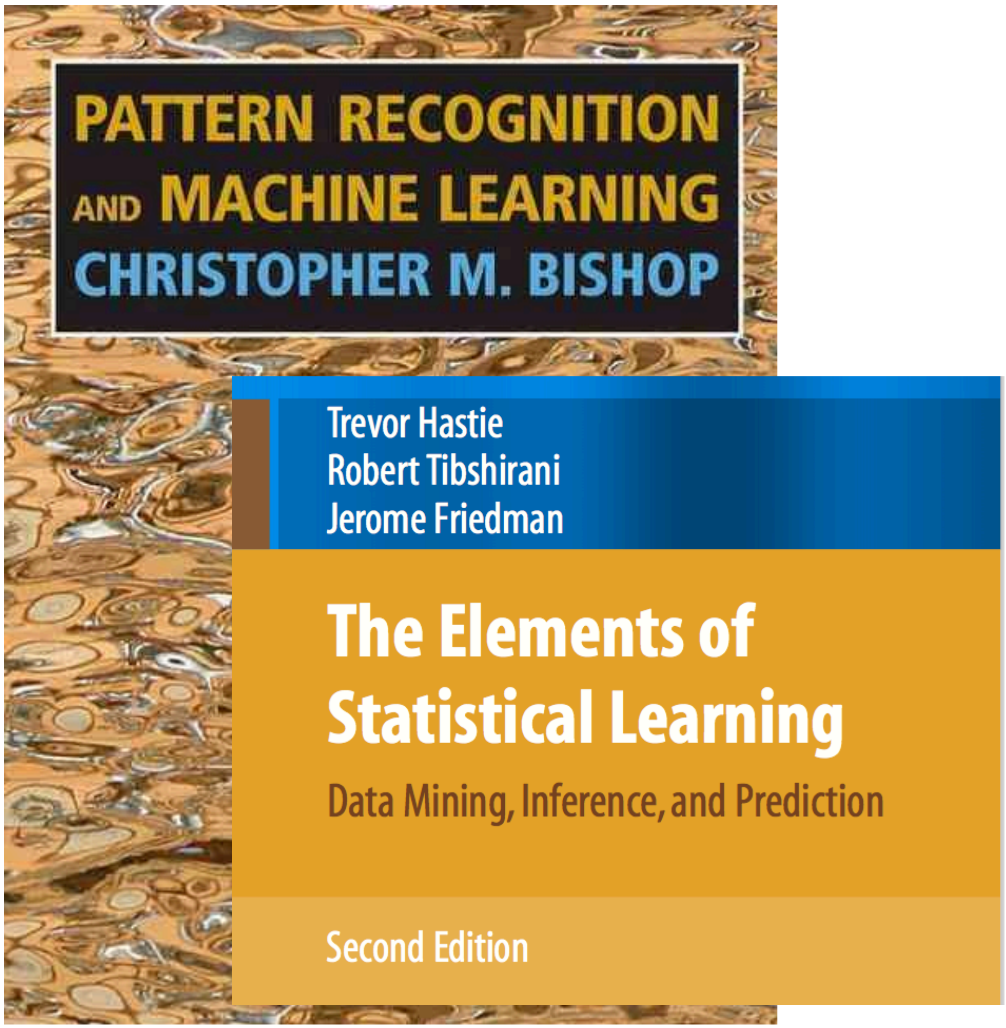
Detection of mines. Navy uses dolphins to assist in this.

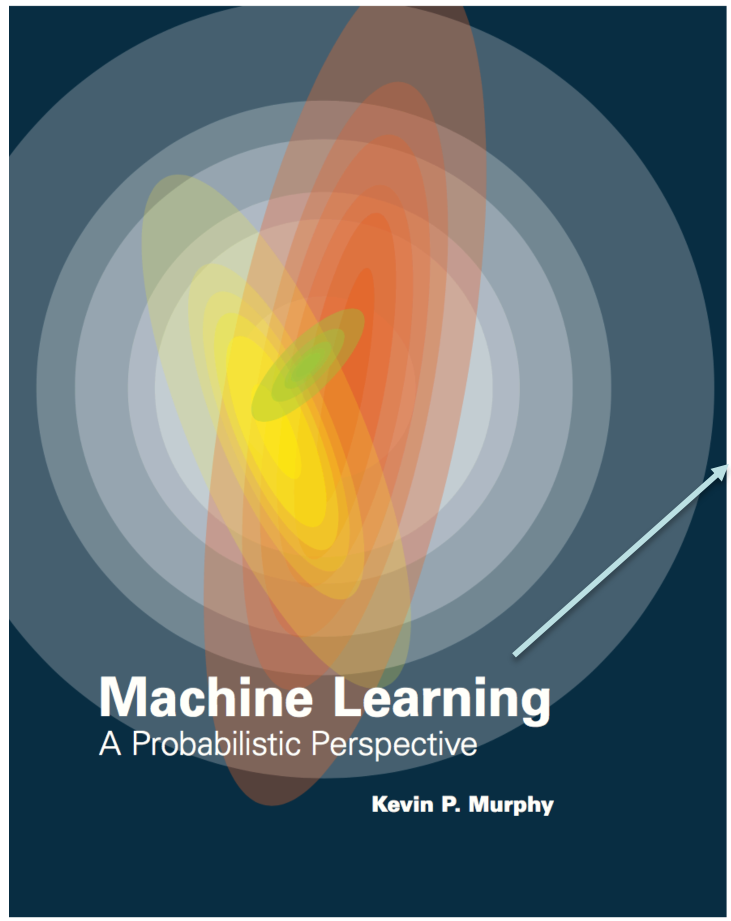
Dolphins = real ML!

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**."





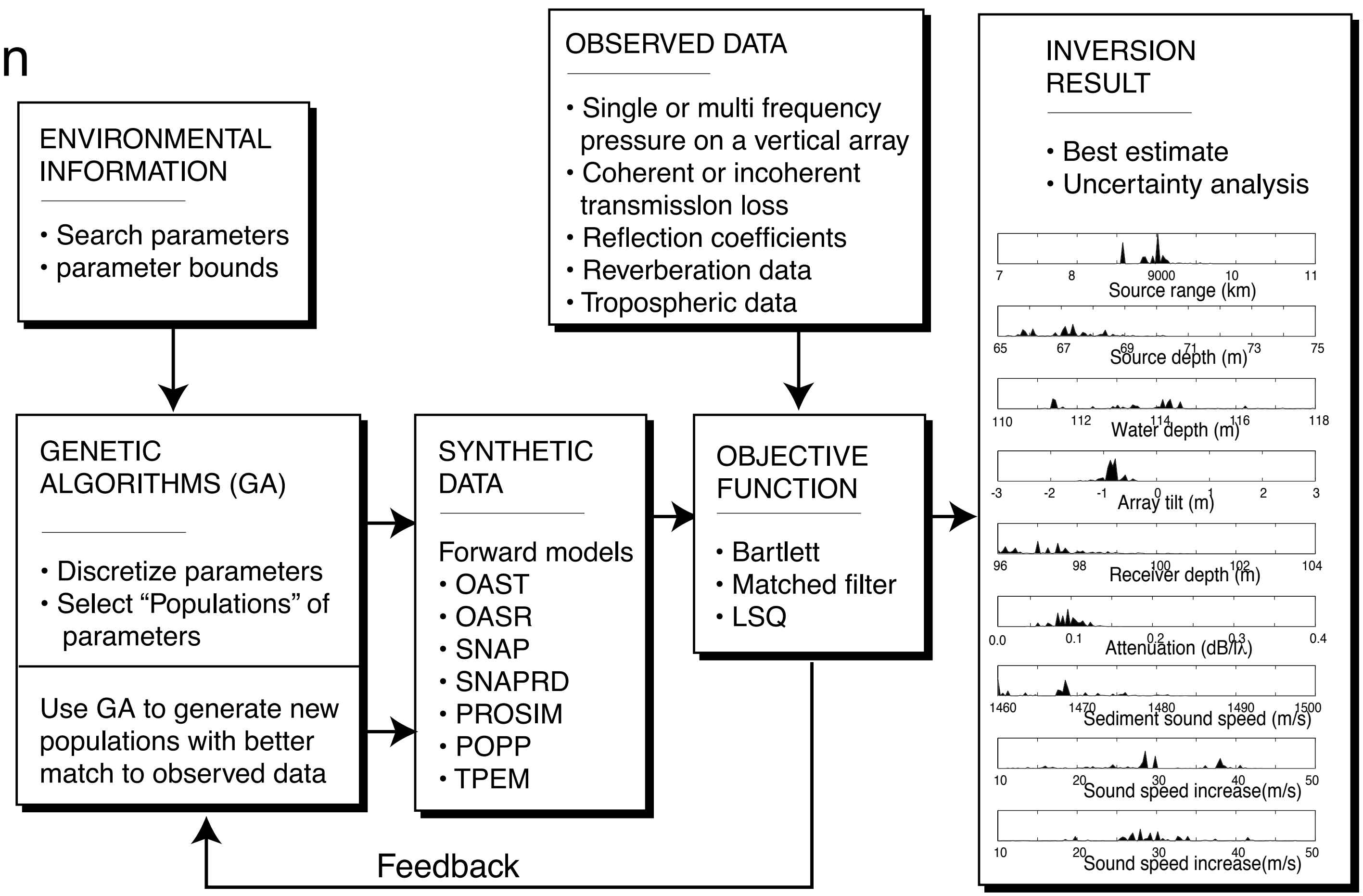
SAGA (NURC 1992-97) is also ML

SAGA has the features that characterize a ML approach:

- Data-driven
- Model based
- Gaussian based likelihoods.
- Bayesian posterior probabilities

Also later additions

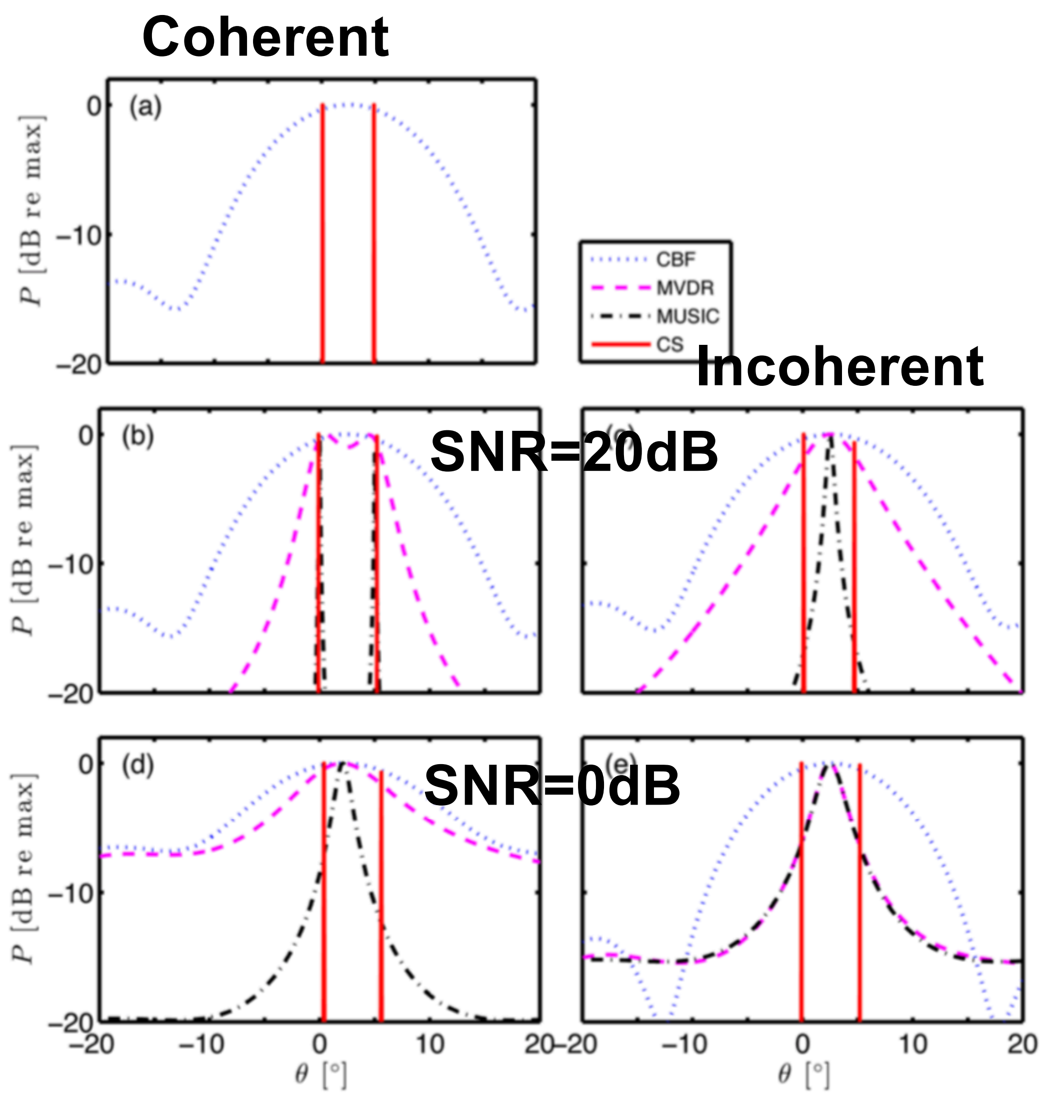
- Sequential estimation
- Particle Filtering

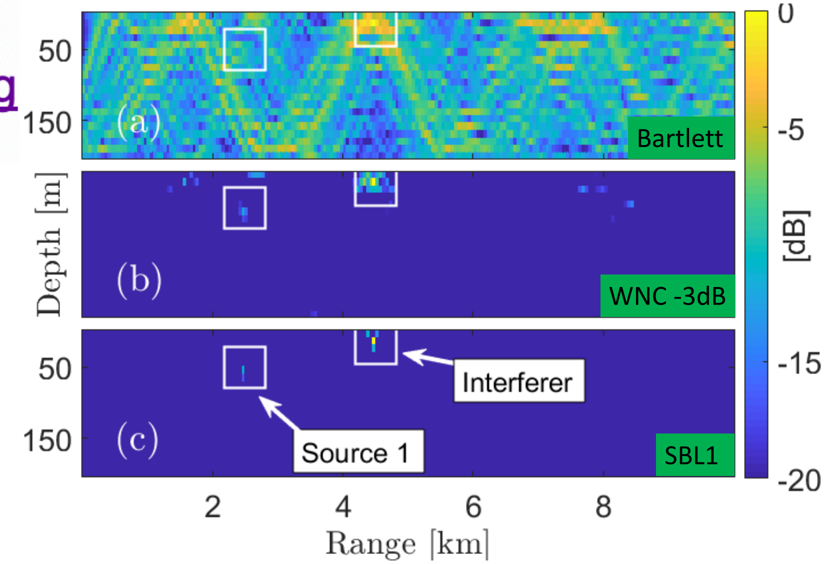


Gerstoft, 1994

Compressive beamforming is also ML

Compressive beamforming
A **Xenaki**, P **Gerstoft**, K Mosegaard - JASA, 2014 Cited by 142
Multiple and single snapshot compressive beamforming
P **Gerstoft**, A **Xenaki**, CF Mecklenbrauker- JASA, 2015 Cited by 78





CS beamforming:
- **single or multiple snapshots**
- **coherent or incoherent**

Xenaki 2014, 2015, Gemba 2017

Machine learning in acoustics: a review

Michael J. Bianco,[1,a)] Peter Gerstoft,[1] James Traer,[2] Emma Ozanich,[1] Marie A. Roch,[3] Sharon Gannot
Charles-Alban Deledalle,[5] and Weichang Li[6]
[1] *Scripps Institution of Oceanography, University of California San Diego, La Jolla, CA 92037, USA*
[2] *Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139*
[3] *Department of Computer Science, San Diego State University, San Diego, CA 92182, USA*
[4] *Faculty of Engineering, Bar-Ilan University, Ramat-Gan 5290002, Israel*
[5] *Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92037, USA*
[6] *Aramco Research Center-Houston, Aramco Services Company, Houston, TX 77084*

- **42-page** JASA review of ML theory. Available on **arXiv** or **http://noiselab.ucsd.edu/**. (Pdf of talk is also there)
- Sections:
  - Machine learning principles
    - Supervised/ Unsupervised learning
  - Deep learning
  - Source localization in speech processing
  - Source localization in ocean acoustics
  - Bioacoustics
  - Seismic exploration
  - Perception of everyday sounds
    - Reverberation
    - Environmental sounds

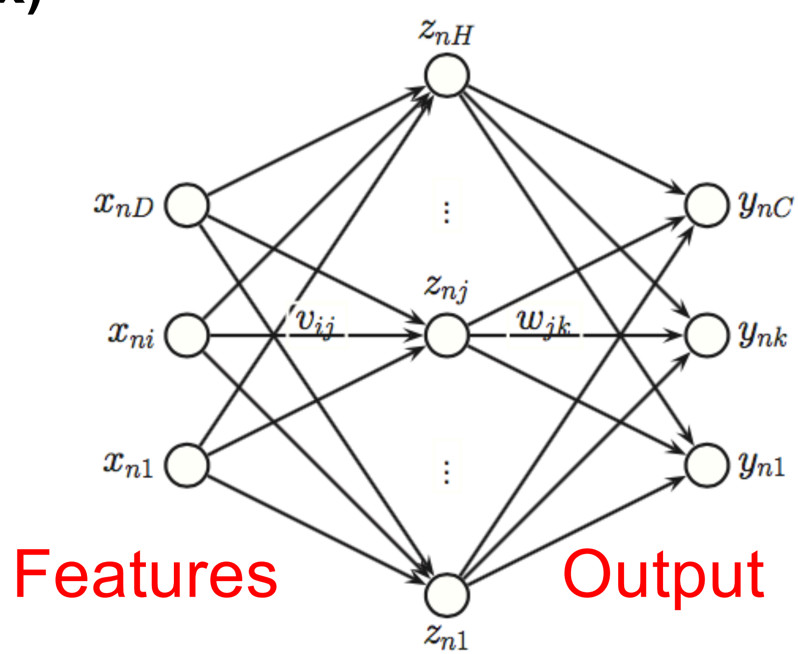
ML Principles

In ML, we are often interested in training a model to produce a desired output given inputs,

$$\mathbf{y=f(x)} + \epsilon$$

- Input $\mathbf{x} \in \mathbb{R}^N$, N features
- output $\mathbf{y} \in \mathbb{R}^P$, P outputs

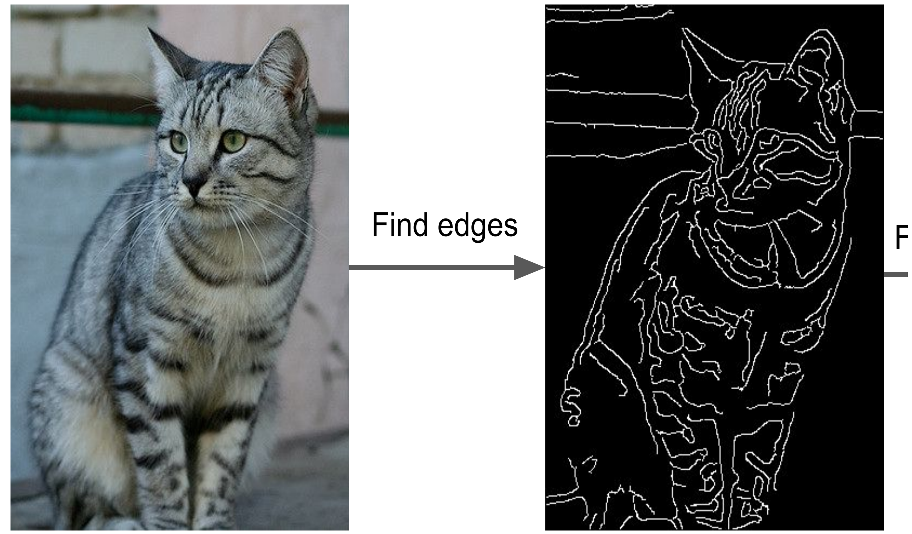
- **Supervised learning**: the P outputs have labelled examples (response variables **y**)
- **Unsupervised learning**: there are no labels. The goal is to find interesting properties from **x,** as an autoencoder $\tilde{\mathbf{x}}=\mathbf{f(x)}$
- ….. and we *train* the model



Two ways to make computers do what you want:

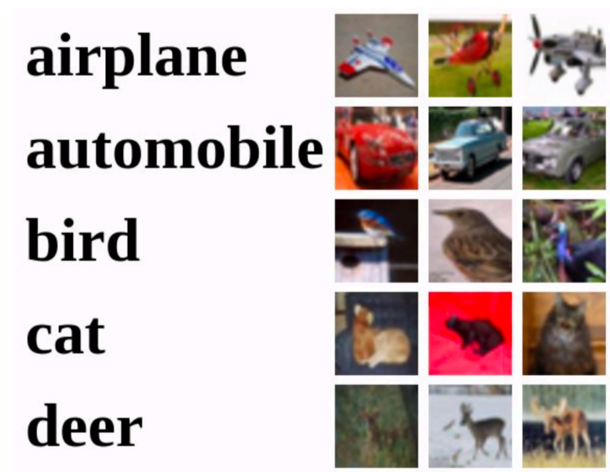
In Image processing this has been done:

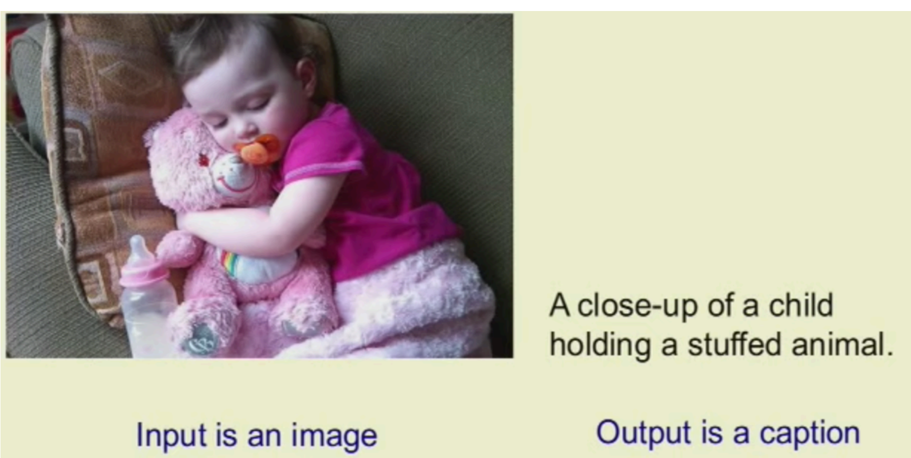
1) Hand-engineered design: Consciously figure out exactly how to manipulate symbolic representations to perform the task and then tell the computer in detail what to do.



2) Learning: Show computers lots of examples of input with desired outputs. Let the computer learn how to map inputs to outputs using **general purpose** learning procedure

Example training set



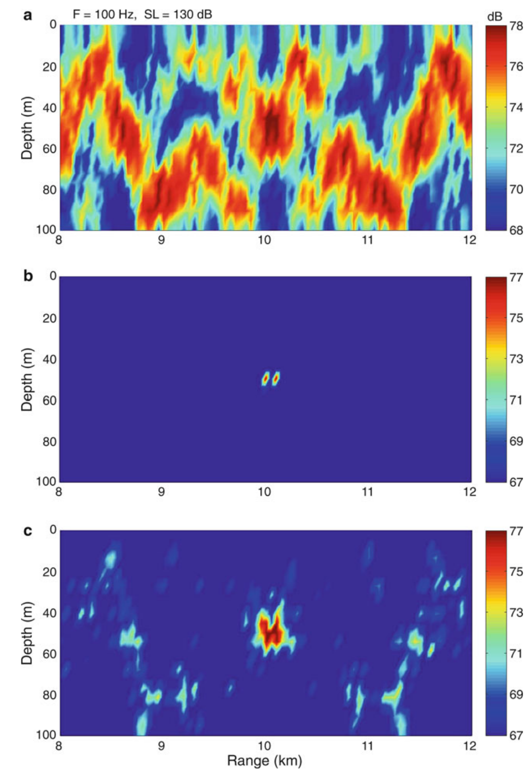


Two ways to make computers do what you want:

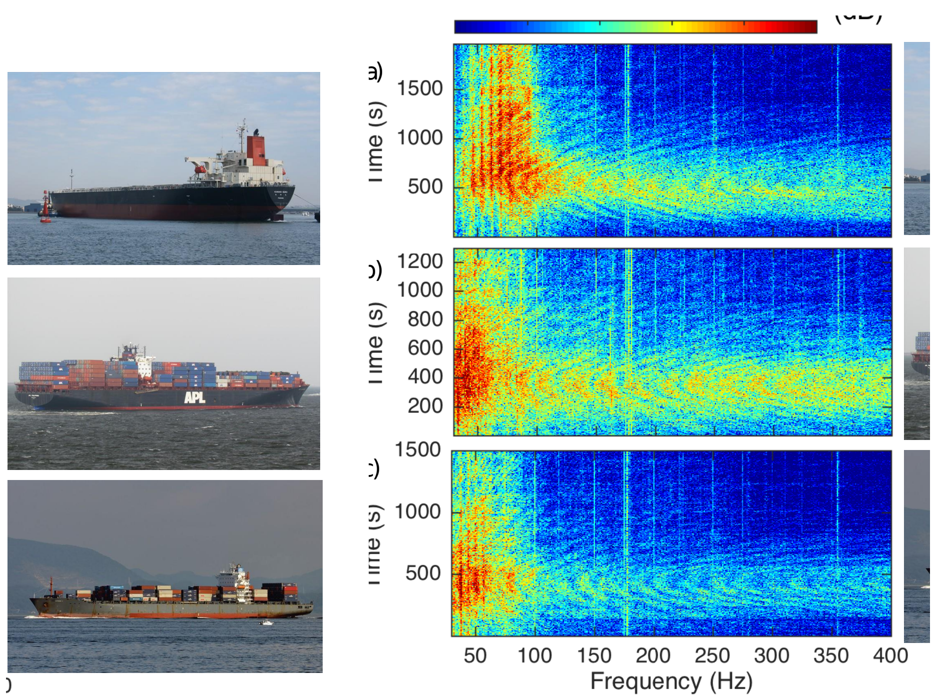
In Ocean acoustics:

1) Hand-engineered design: See the **1000 papers** on Match Field Processing! Sometimes it works…

=> **Old School**



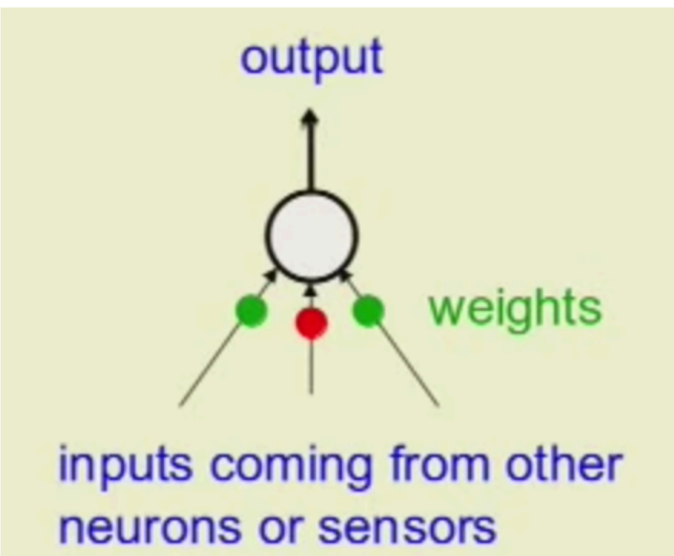
2) Learning: Show computers lots of examples of input with desired outputs. Let the computer learn how to map inputs to outputs using **general purpose** learning procedure



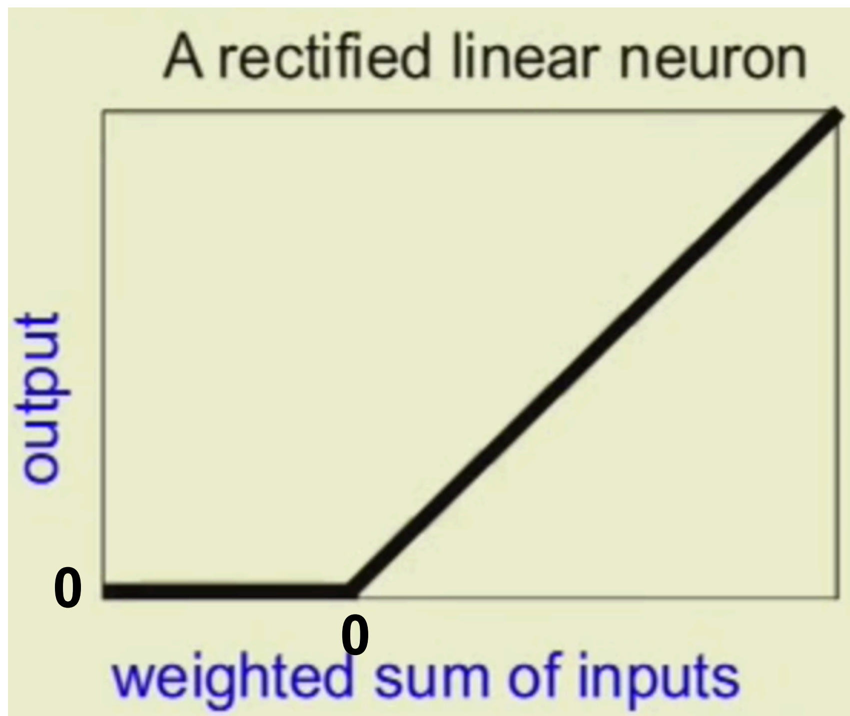
What is an artificial neuron?

We simplify a real neuron to investigate how neurons can do computations that are too difficult to program as

- Converting image pixel intensity into string of words describing it



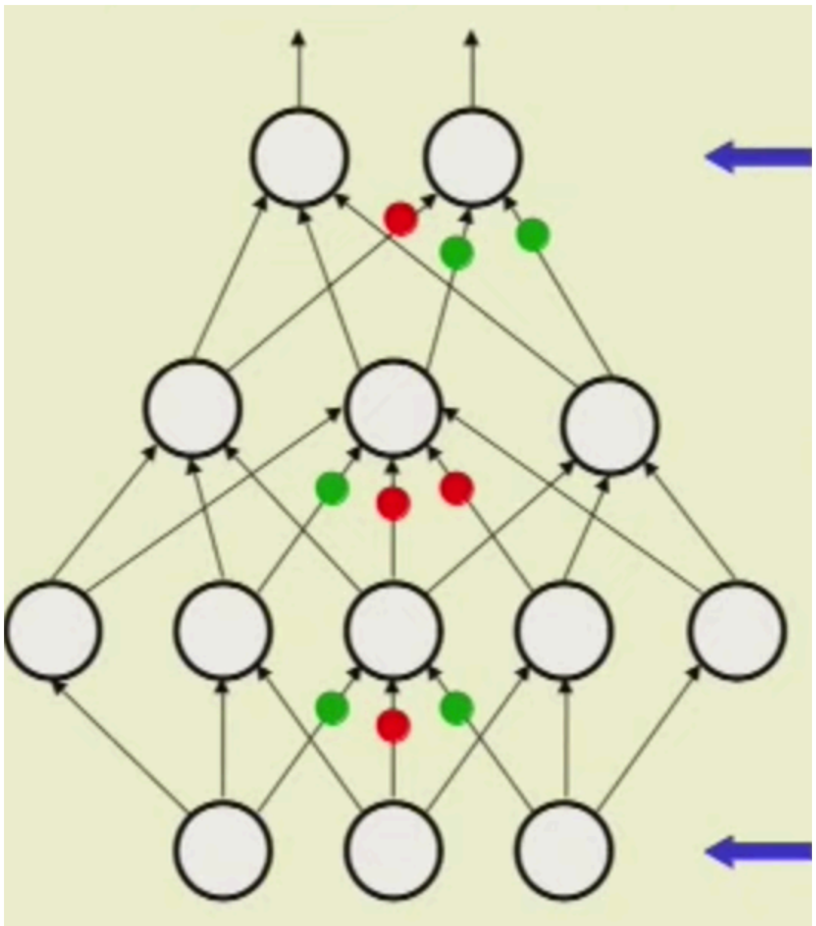
ReLu



What is artificial neural network

Connecting neurons in layers with no cycles gives a feed-forward neural net (FNN).

$$a_j = \mathrm{ReLu}(\boldsymbol{w}^T \boldsymbol{x}) = \mathrm{ReLu}(\sum_{n=1}^{N} w_n x_n)$$



Output neurons

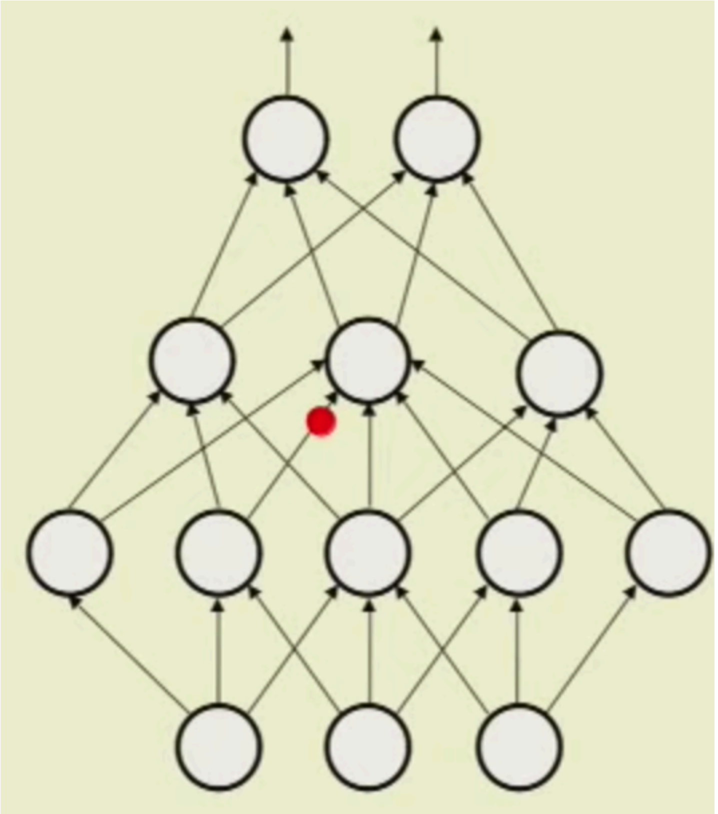
Multiple layers of hidden neurons
Hidden layers

Input neurons

Supervised training vs backpropagatoin

Supervised training is inefficient:

- Take a few of the training cases and measure the NN output. (called **stochastic sampling**)
- Change one weight slightly.
- If NN output improved, **keep it.**

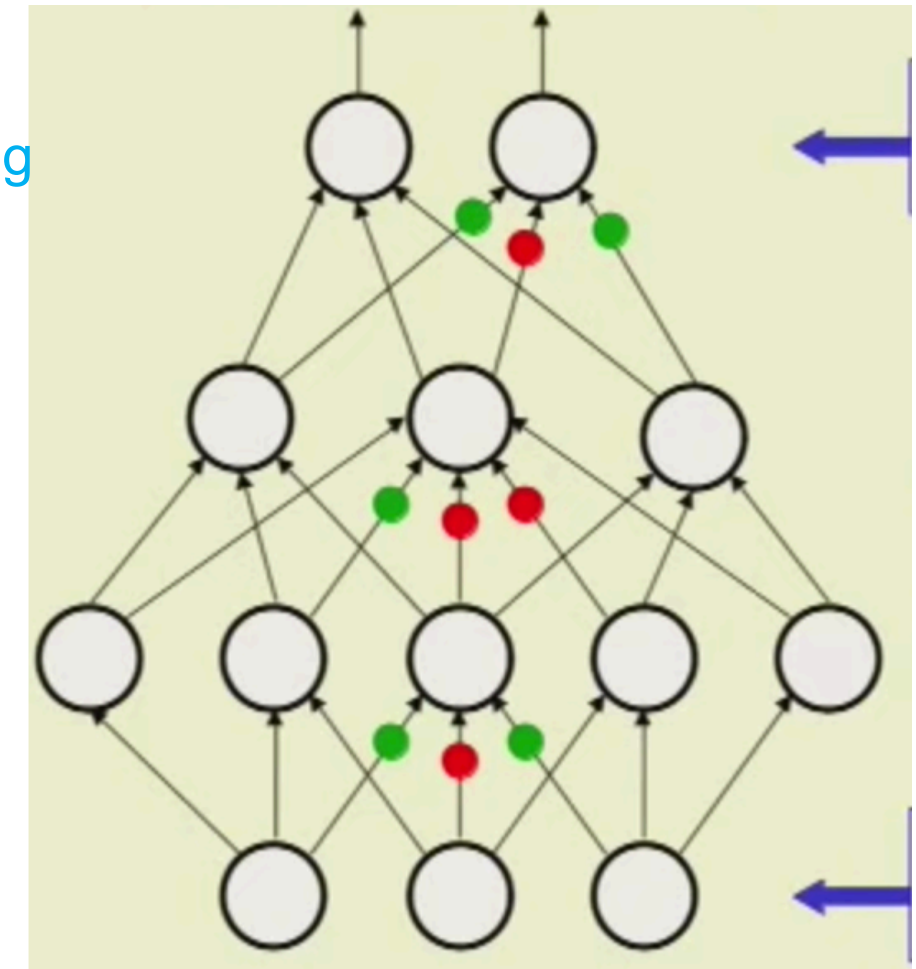


- **Backpropagation** efficiently compute how a change in weight effects the NN output.
- The error gradients for all of the weights is obtained at once. The chain rule dictates how the NN output change for each weight.

How to learn many layers of features

Compare outputs with the correct answer to get the error signal

Back-propagate error signal to get derivatives for learning



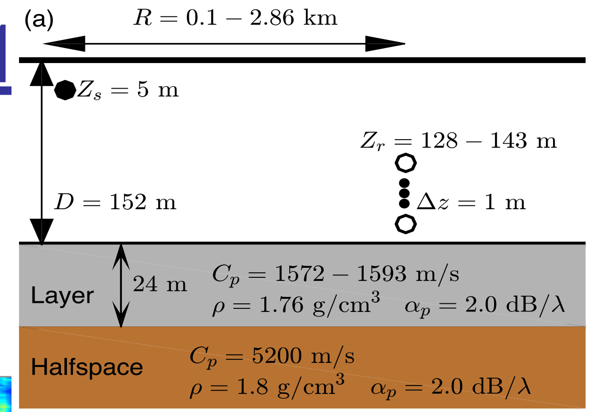
For L layers with N neurons, we have N^2L weights

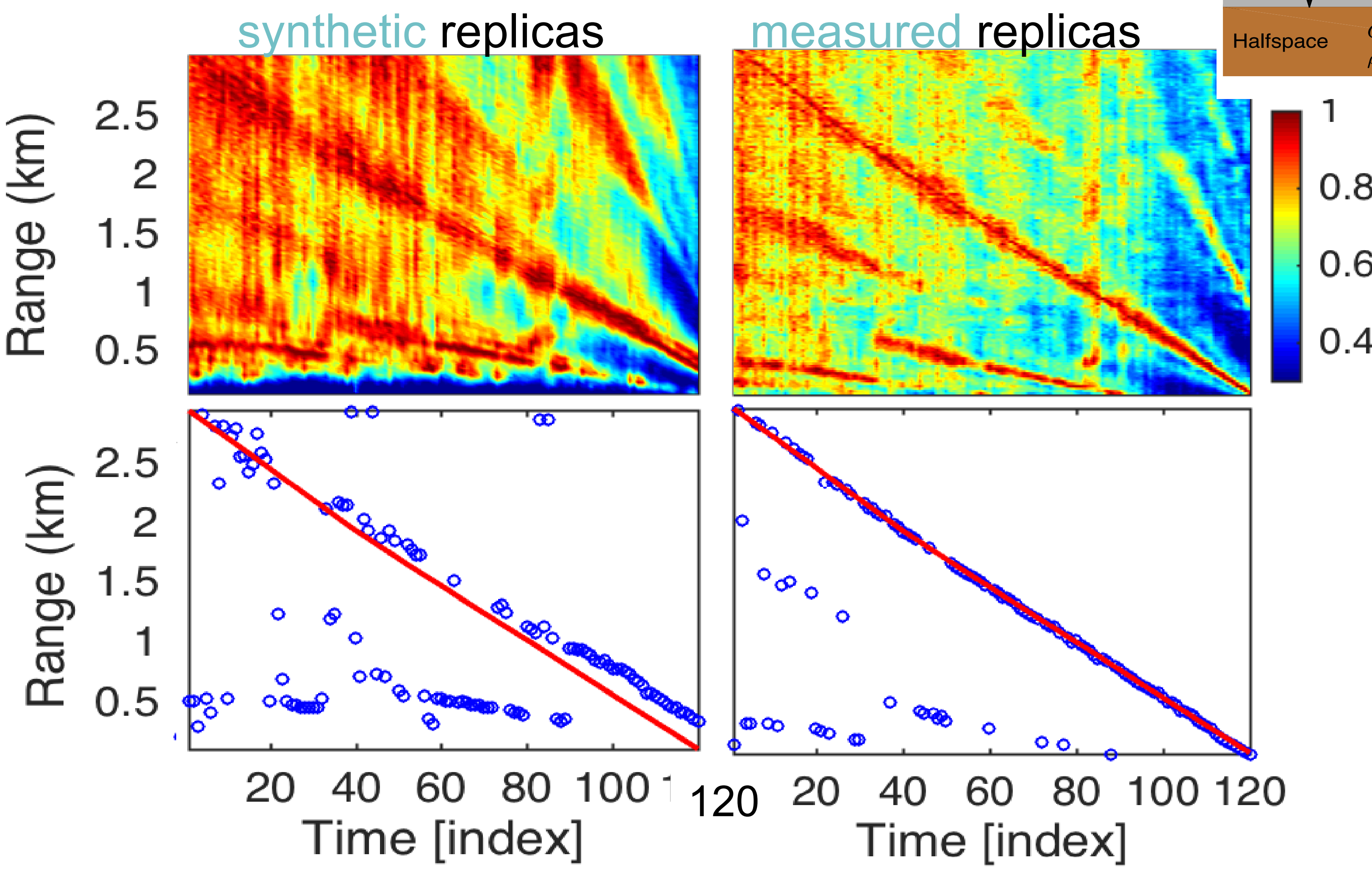
Matched-Field Processing on test data 1

Noise09 Frequencies [300:10:950]Hz

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

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





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

Niu 2017a, JASA

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

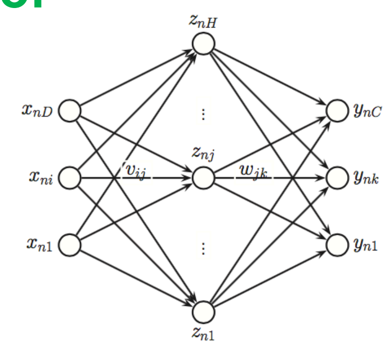
Sample Covariance Matrix to reduce effect of source phase

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

N_s Number of snapshots

SCM is a conjugate symmetric matrix.

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



ML source range classification

Array Data: 300–950Hz with 10Hz increment, i.e., 66 frequencies.
16 hydrophones with 1 m spacing

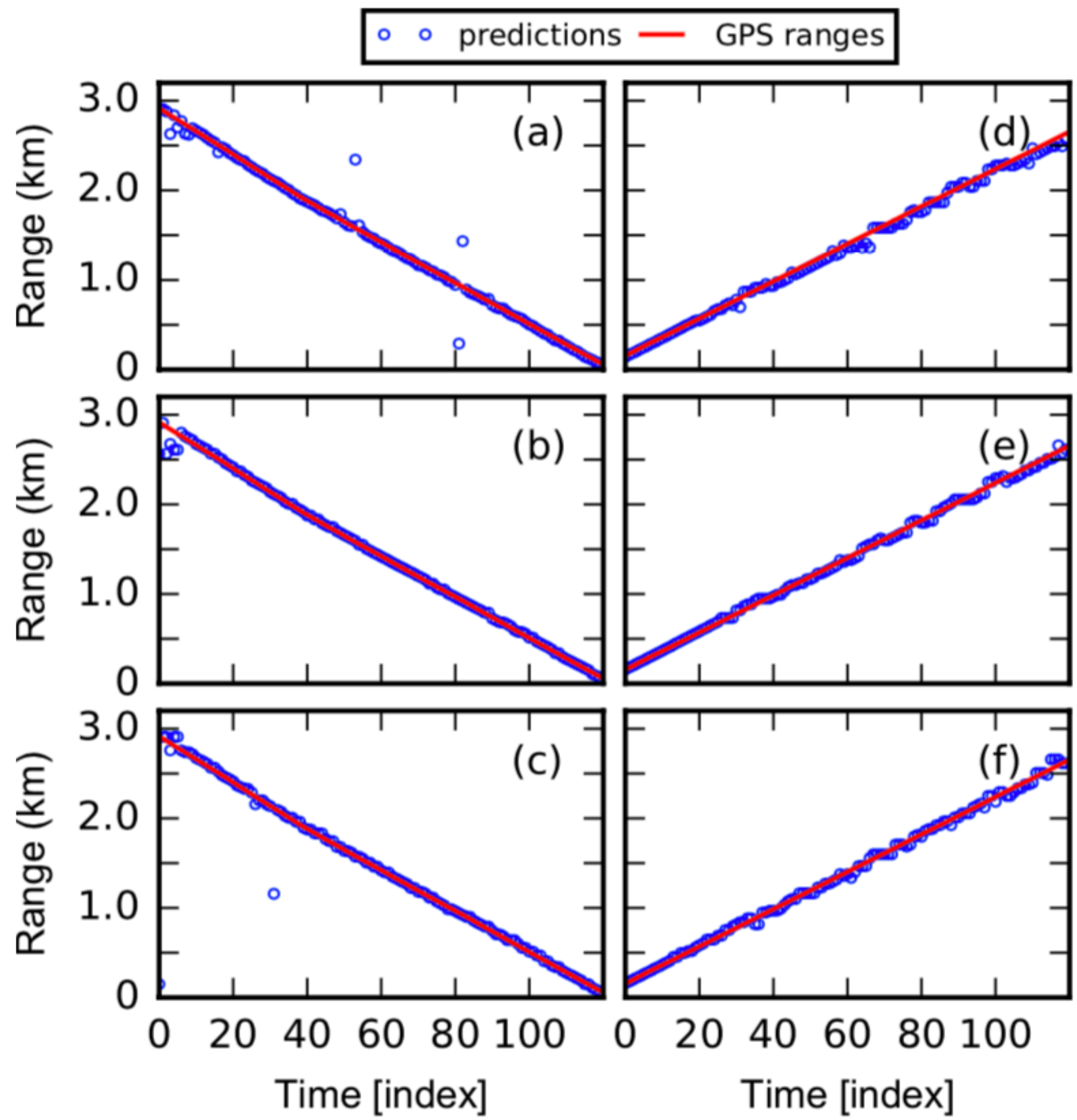
First NN is trained with one source

Test-Data-1 **Test-Data-2**

FNN
3 hidden layers with 512 nodes

SVM
Radial basis Functions

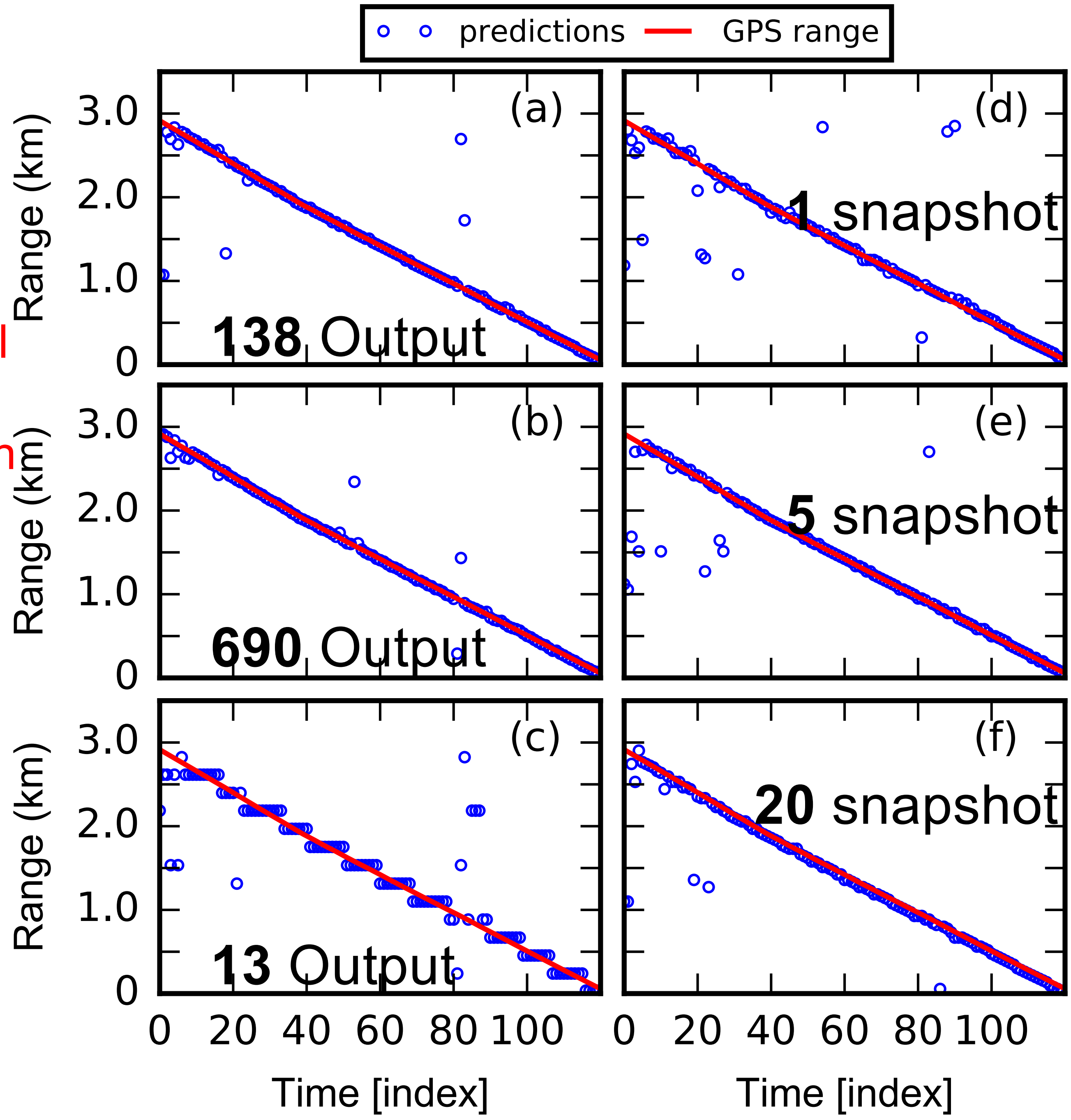
RF
Random forest



Niu 2017a, JASA

Other parameters: FNN for range classification

Conclusion
- Easier than conventional MFP
- Classification easier than regression
- **FNN, SVM, RF** works.
- Works for:
  - multiple ships,
  - Deep/shallow water



60s Science
Scientfic Am

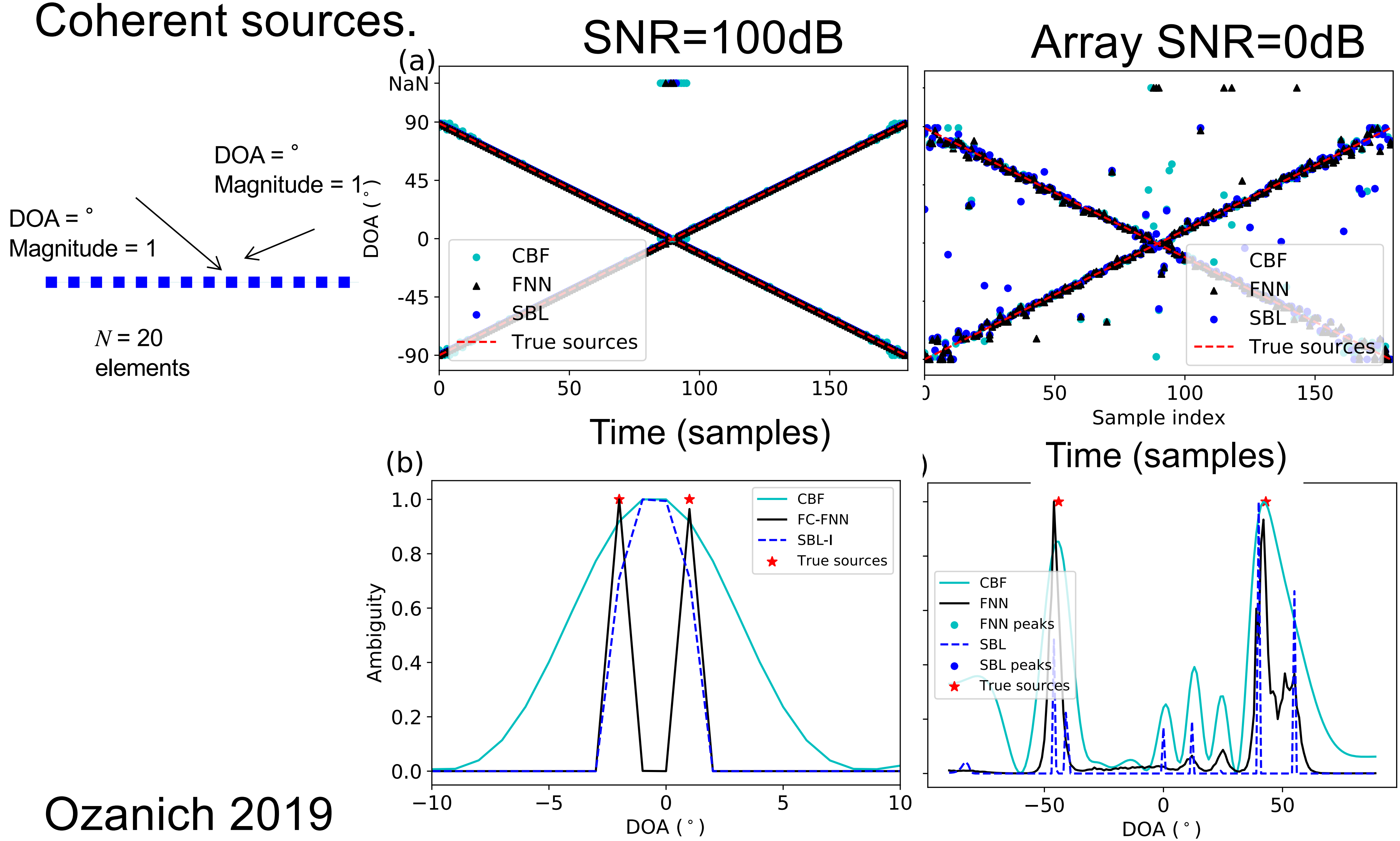
Niu 2017a, JASA

DOA estimation with Neural Networks

5 layers with 1024 nodes fully connected

20 element array at $\lambda/3$ spacing, searching for 180 DOAs

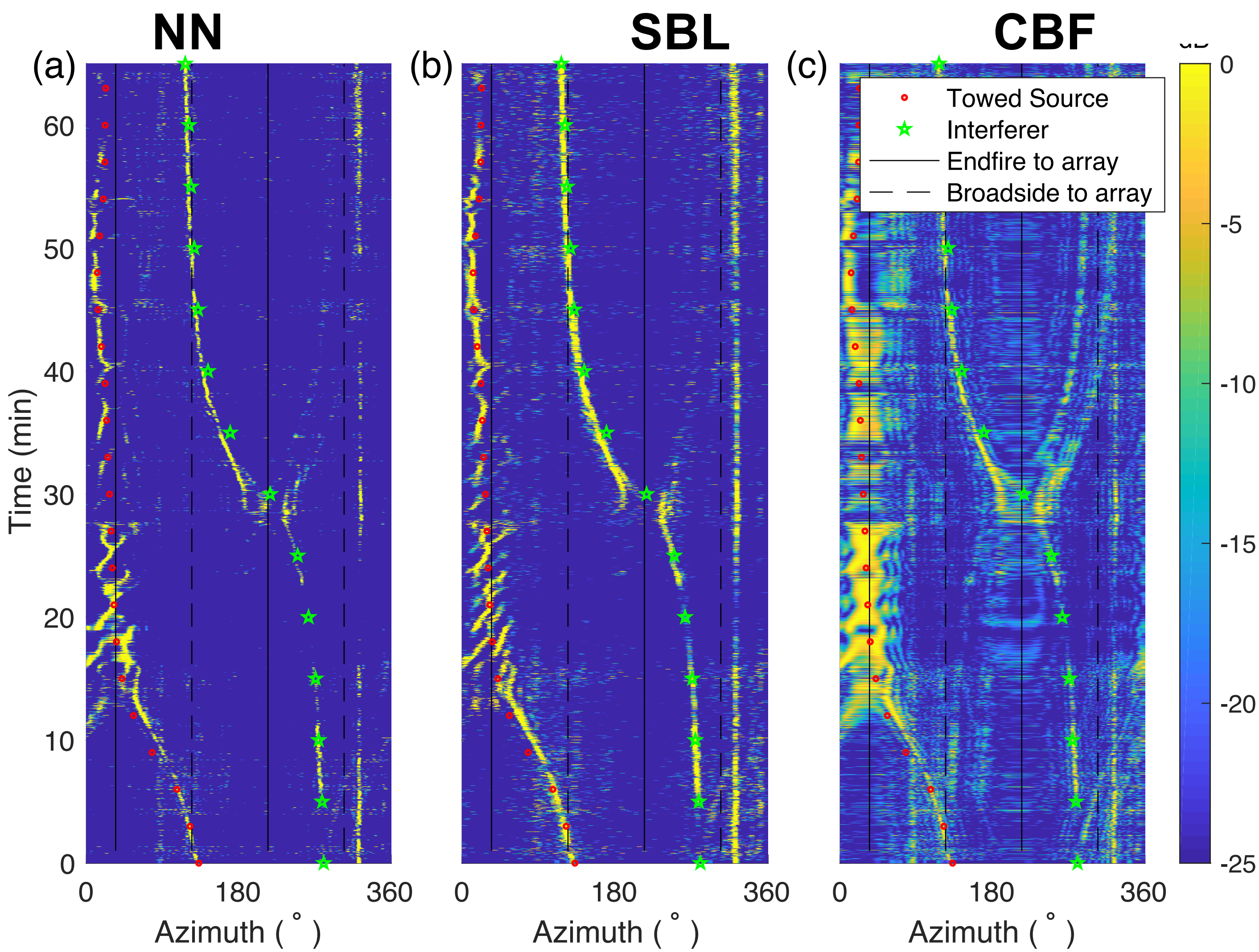
Coherent sources.

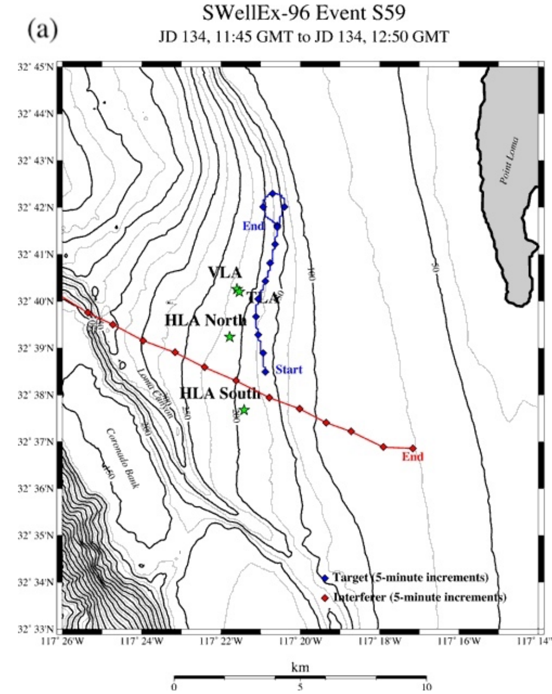


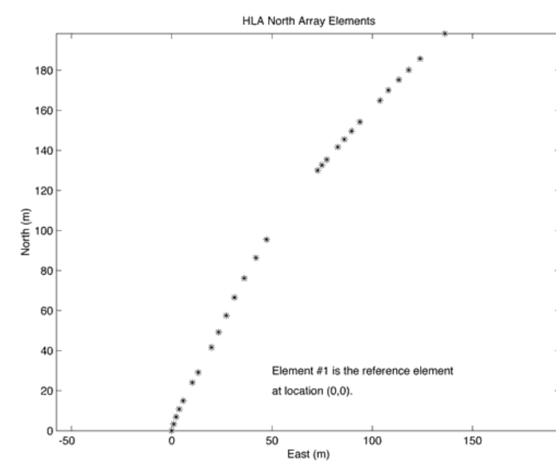
Ozanich 2019

DOA for two sources from SWELLEx96

5 layers with 1024 nodes fully connected
One frequency (79 Hz), L=1 snapshot







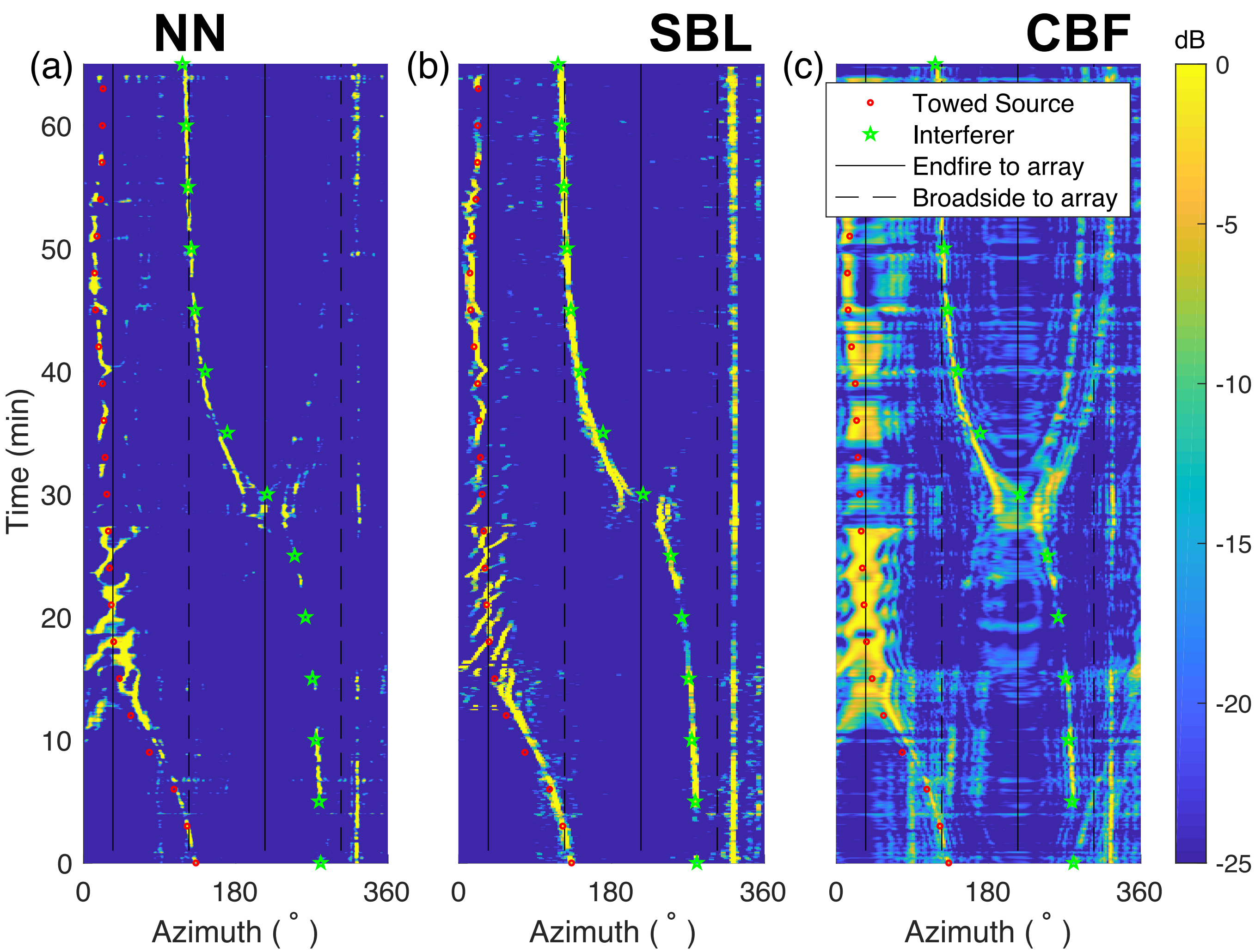
Ozanich 2019

DOA for two sources from SW06

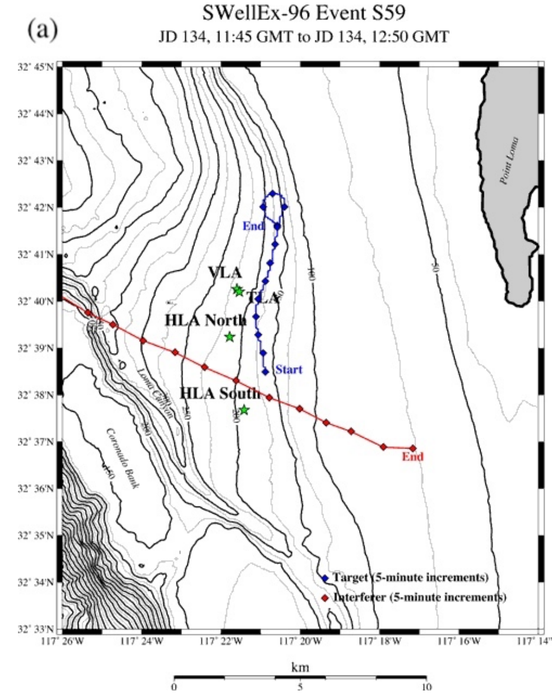
5 layers with 1024 nodes fully connected

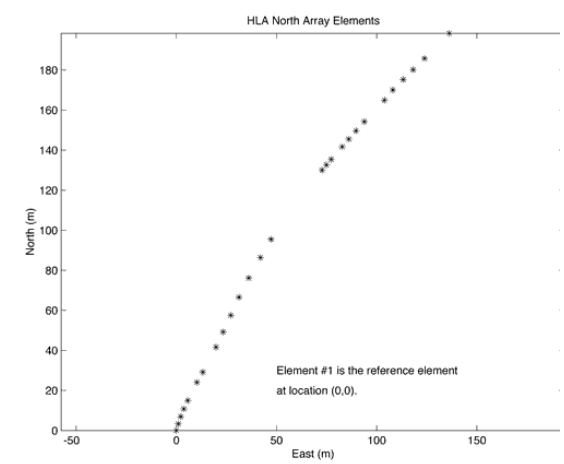
One frequency (79 Hz), **L=10** snapshot

More snapshots give cleaner image

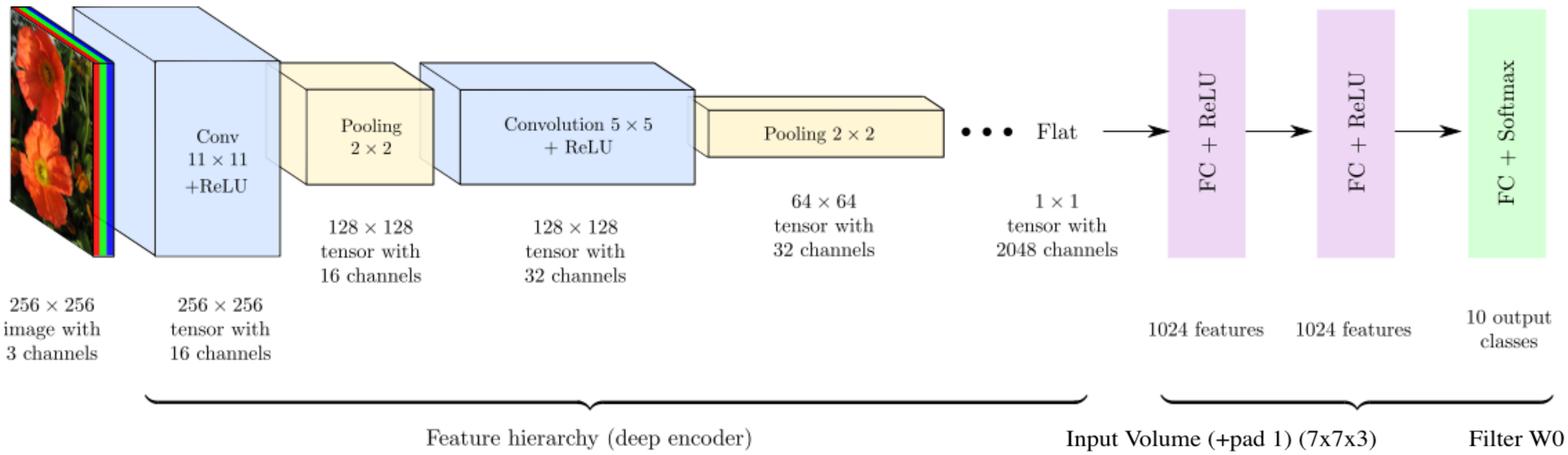


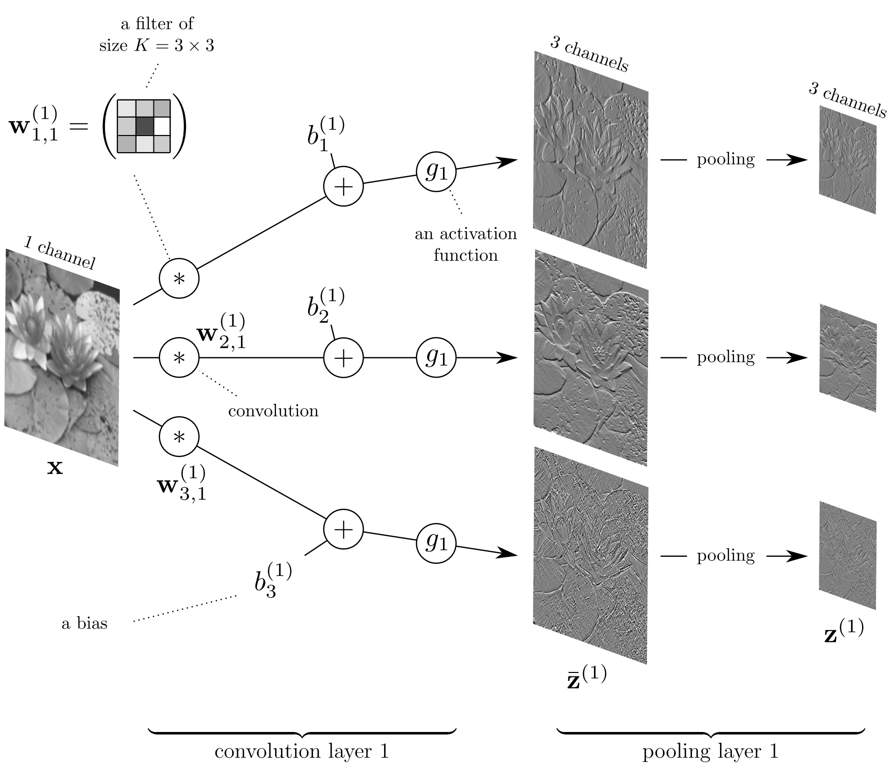
Ozanich 2019

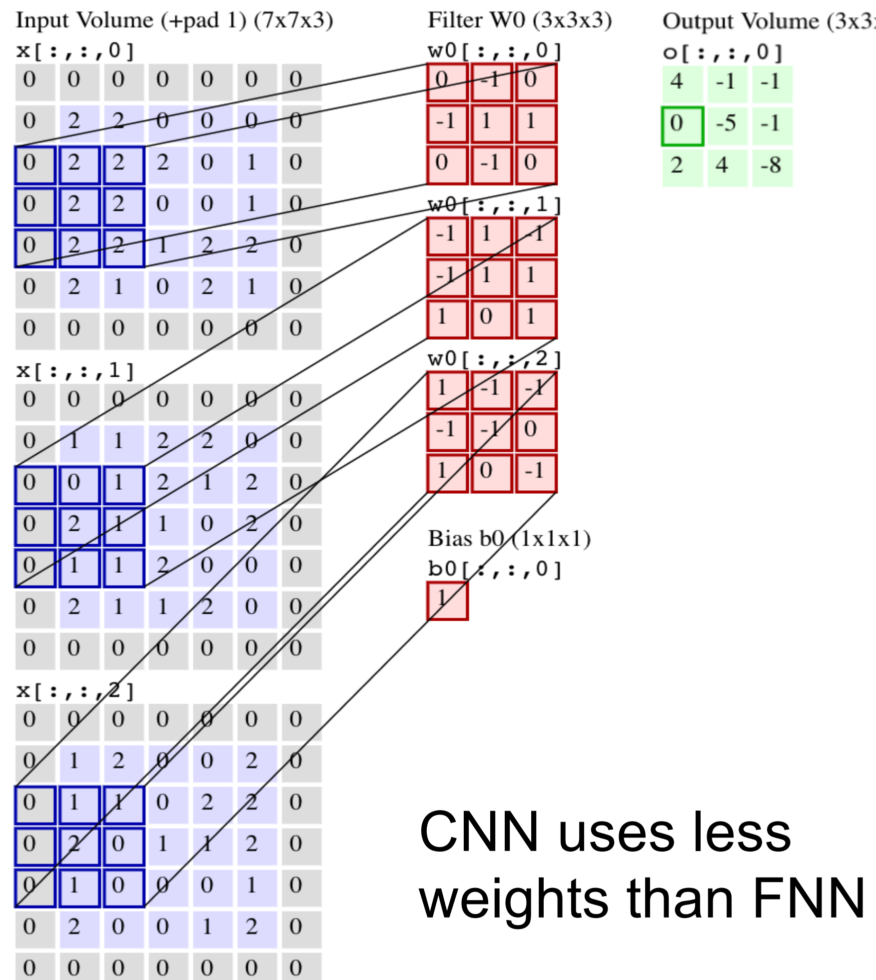




Deep Convolutional NN







Bianco 2019, Niu 2019,

CNN uses less weights than FNN

Magnitude only localization

Single receiver,
3-16 km from source
Multi-frequency 100-200 Hz,
magnitude only

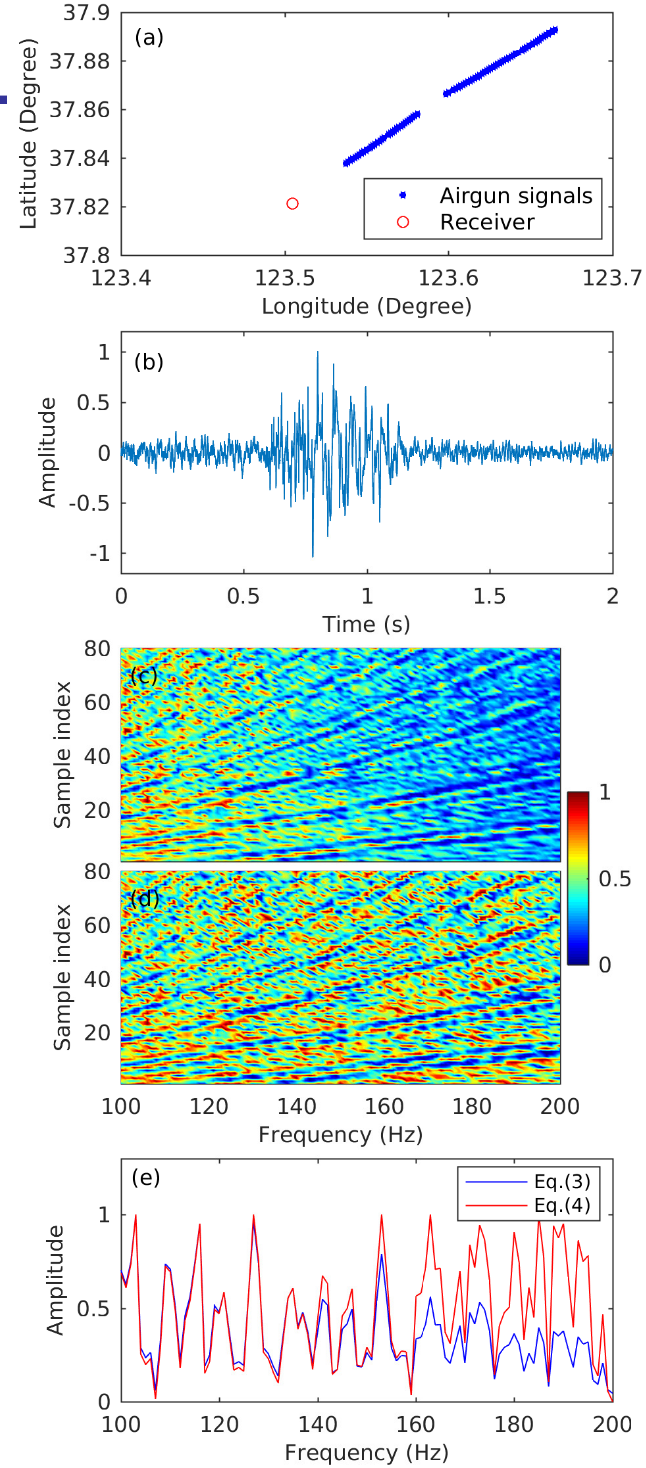
Much less input as sample covariance matrix is not needed. Magnitude is averaged directly

SAGA, multi frequency objective function

$$\phi_F(\Theta) = 1 - \frac{|\sum_{f=1}^{F} \hat{\mathbf{p}}(f)\hat{\mathbf{q}}(f,\Theta)|^2}{\sum_{f=1}^{F} |\hat{\mathbf{p}}(f)|^2 \sum_{f=1}^{F} |\hat{\mathbf{q}}(f,\Theta)|^2},$$

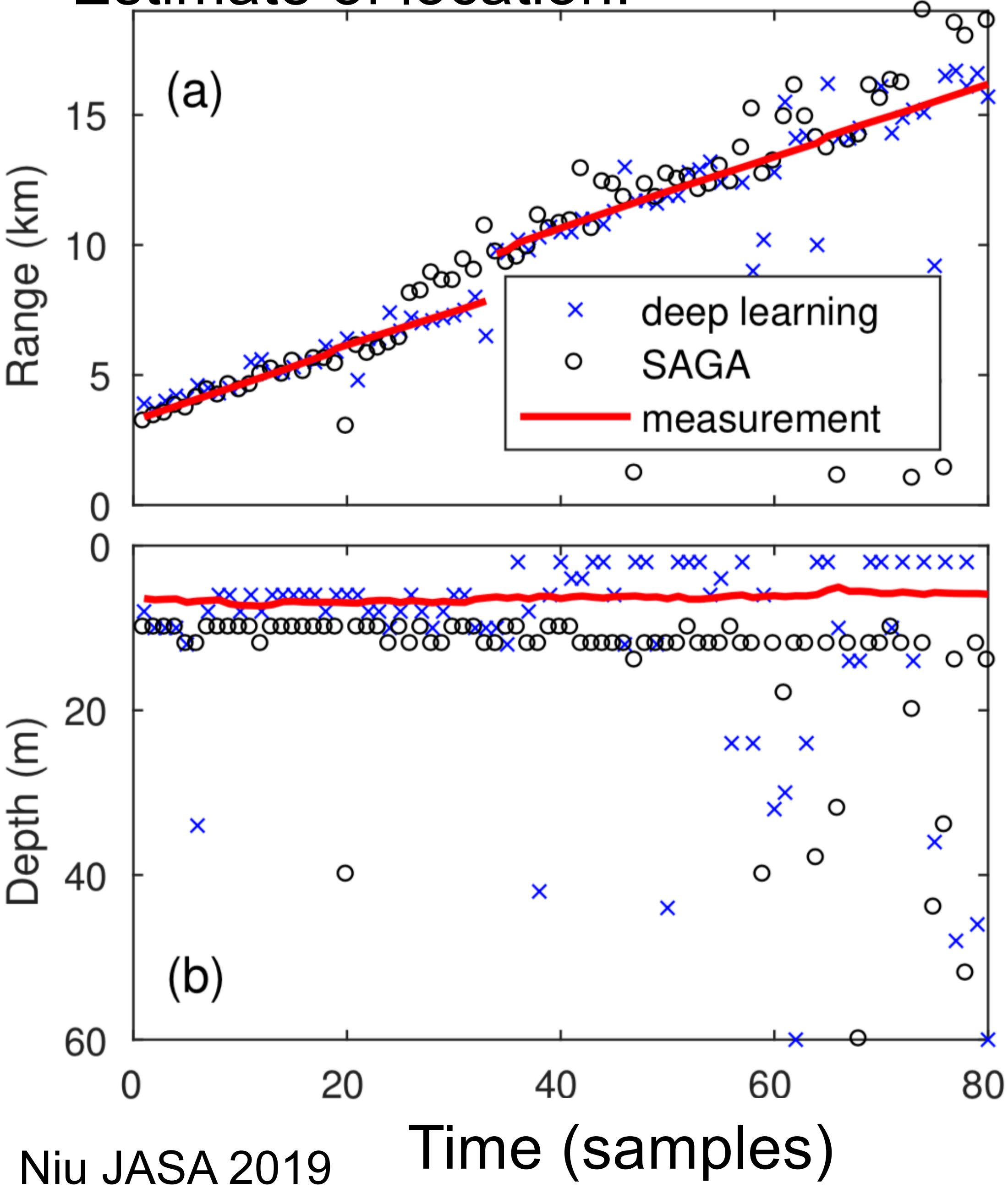
$\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ are magnitudes

Niu 2019



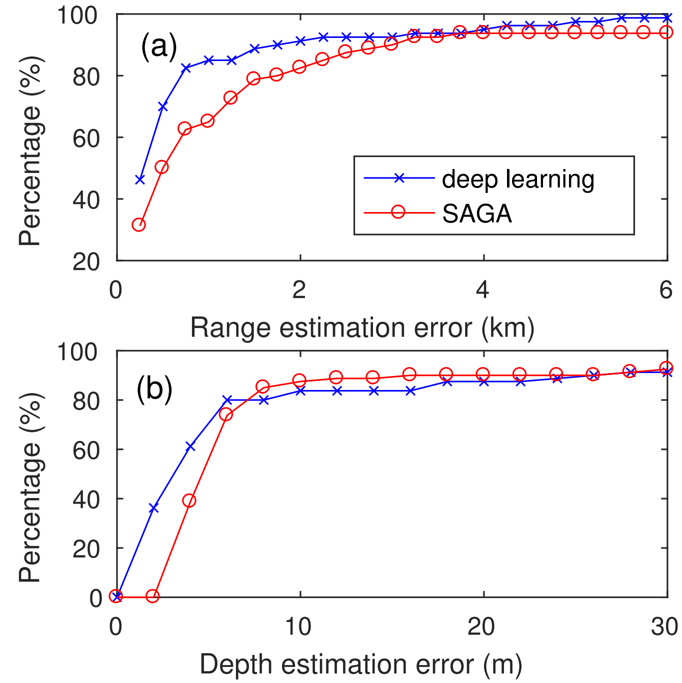
ML and SAGA ranging

Estimate of location:



Niu JASA 2019

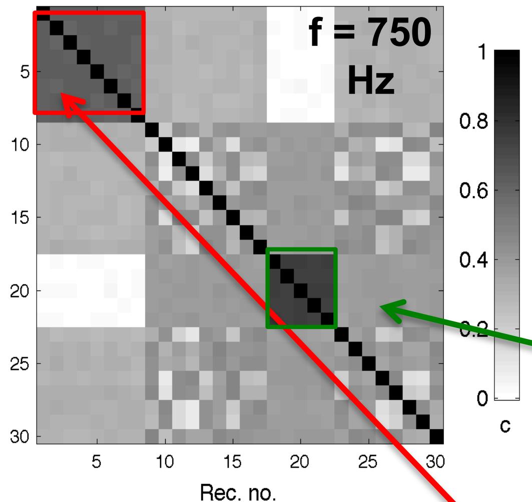
Statistics of location



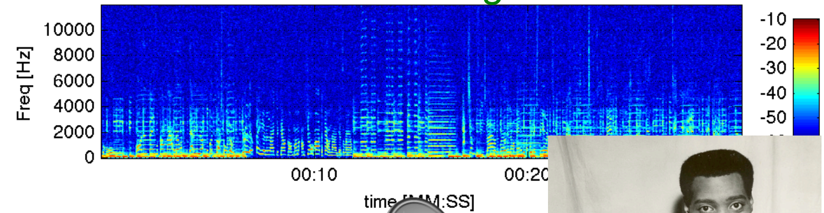
Does ML **beat** SAGA?

Graph Signal Processing for locating a source

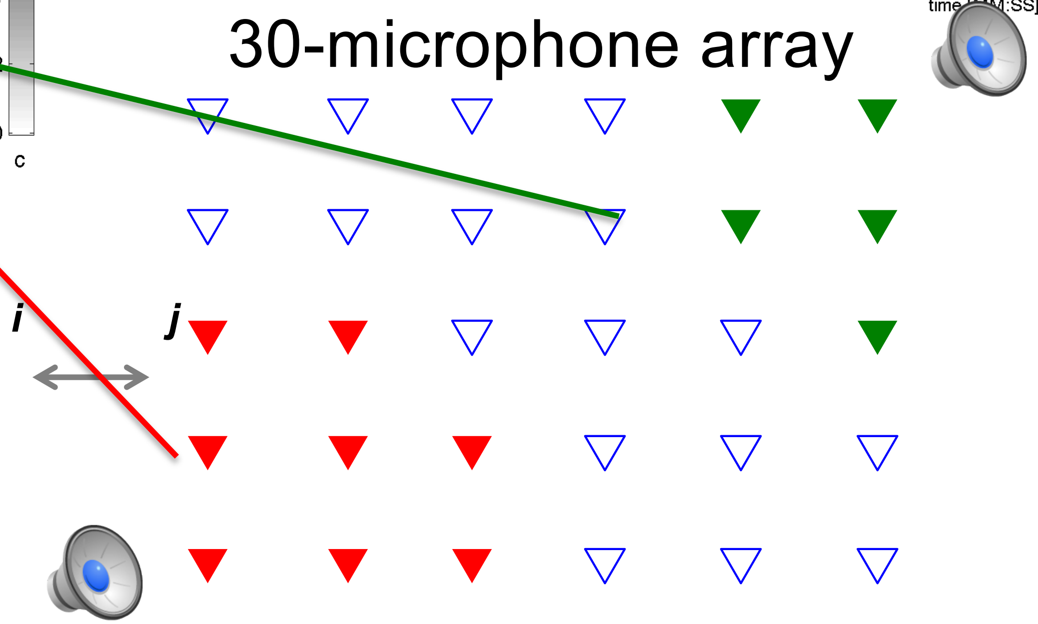
unsupervised



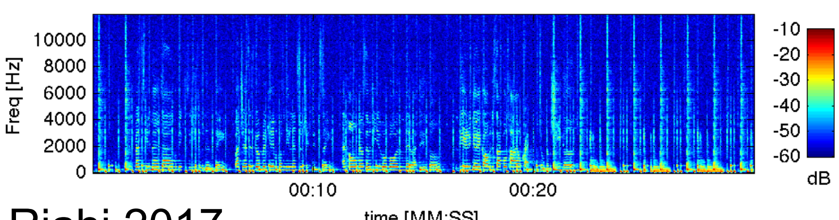
Location 2: Otis Redding - “Hard to handle”



30-microphone array



Location 1: Prince - “Sign o’ the times”



Spectral coherence

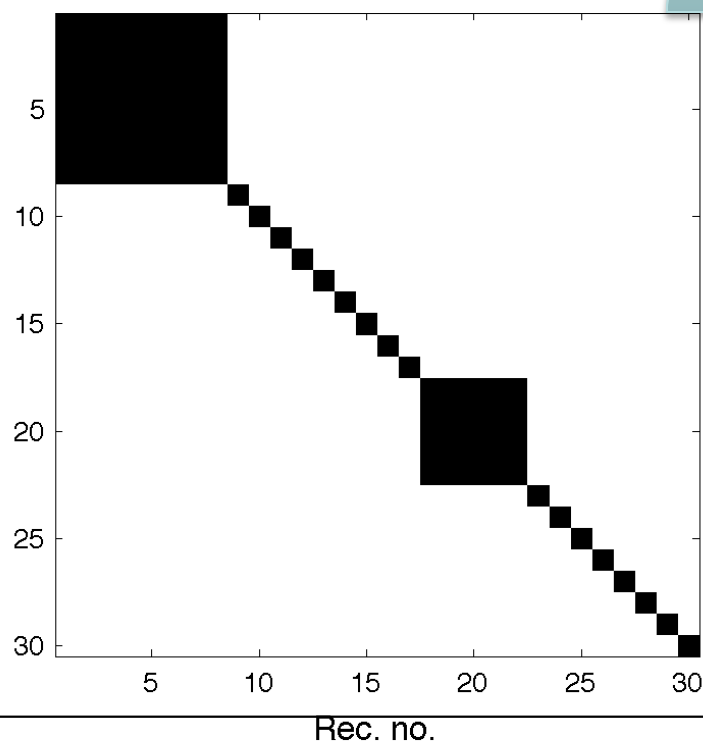
$$\hat{C}_{ij}(f) = \frac{1}{N}\sum_{t=1}^{N} X_i(f,t)\cdot \bar{X}_j(f,t)$$

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

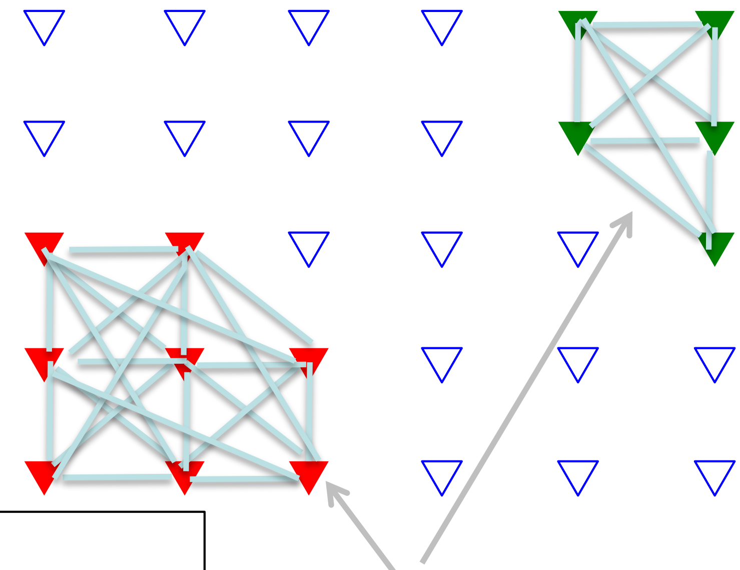
Riahi 2017

Two sources in the network

Statistically significant entries
=> Connectivity matrix



Graph with 30 nodes



- 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).

Connected subgraphs:

5 nodes and 9 edges

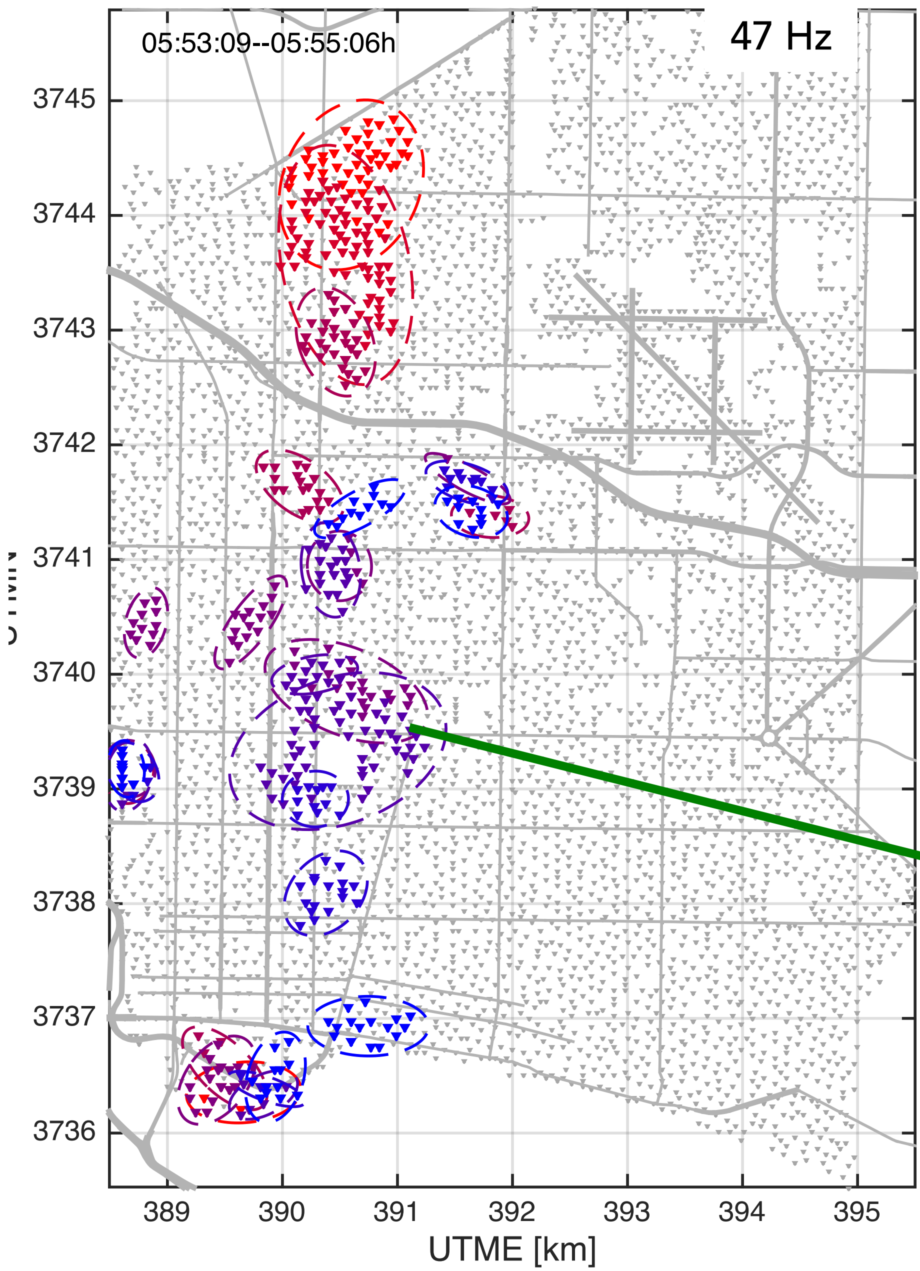
8 nodes and 20 edges

Graph clustering for localization within a sensor array

Peter Gerstoft and Nima Riahi, noiselab.ucsd.edu
Christoph Mecklenbrauker, TU Wien

Based on paper: Riahi and Gerstoft, Signal Processing, 2017

March 5—12, 2011: 3TB, 5200 Stations in Long Beach, California



Helicopter rotor noise (seismo-acoustic coupling)

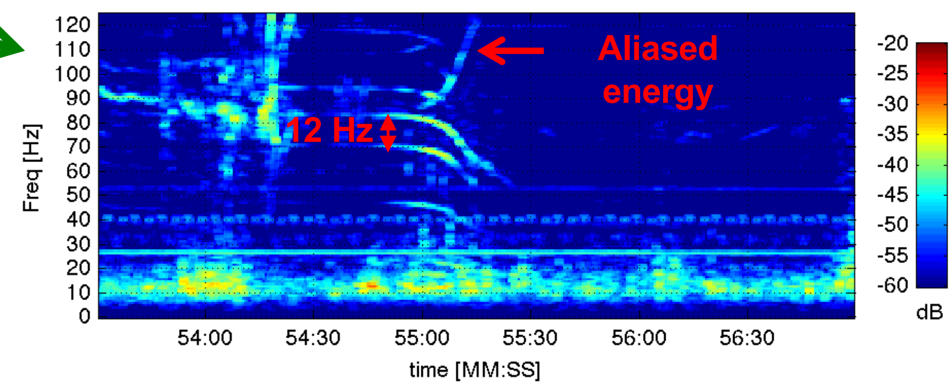
Several peaks consistent with helicopter rotor harmonics (20-100 Hz).

Doppler shift

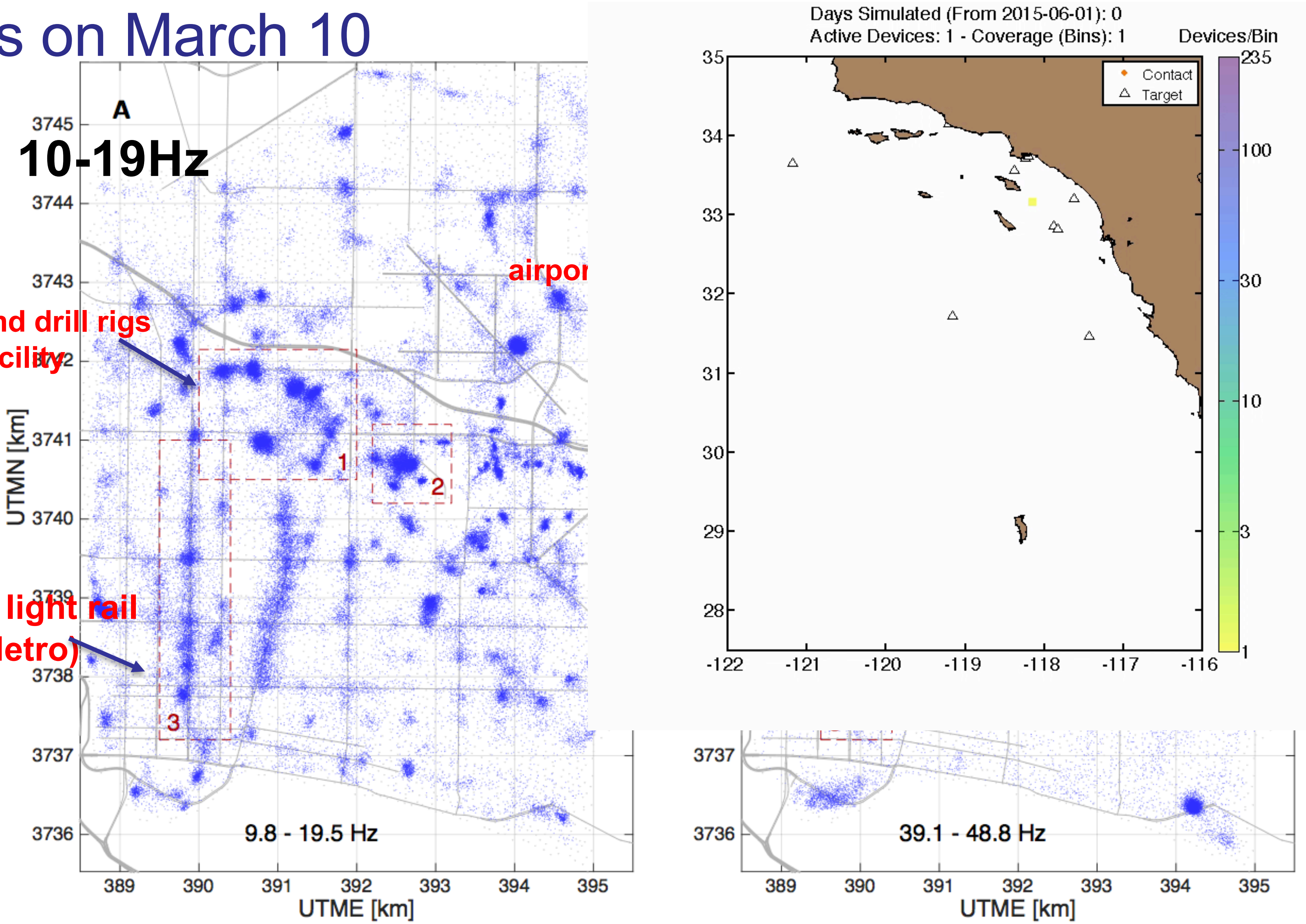
$f_{high}/f_{low}=(v_0+v)/(v_0-v)\approx 1.4$ i.e. $v\approx 250$ km/h

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

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



Clusters on March 10

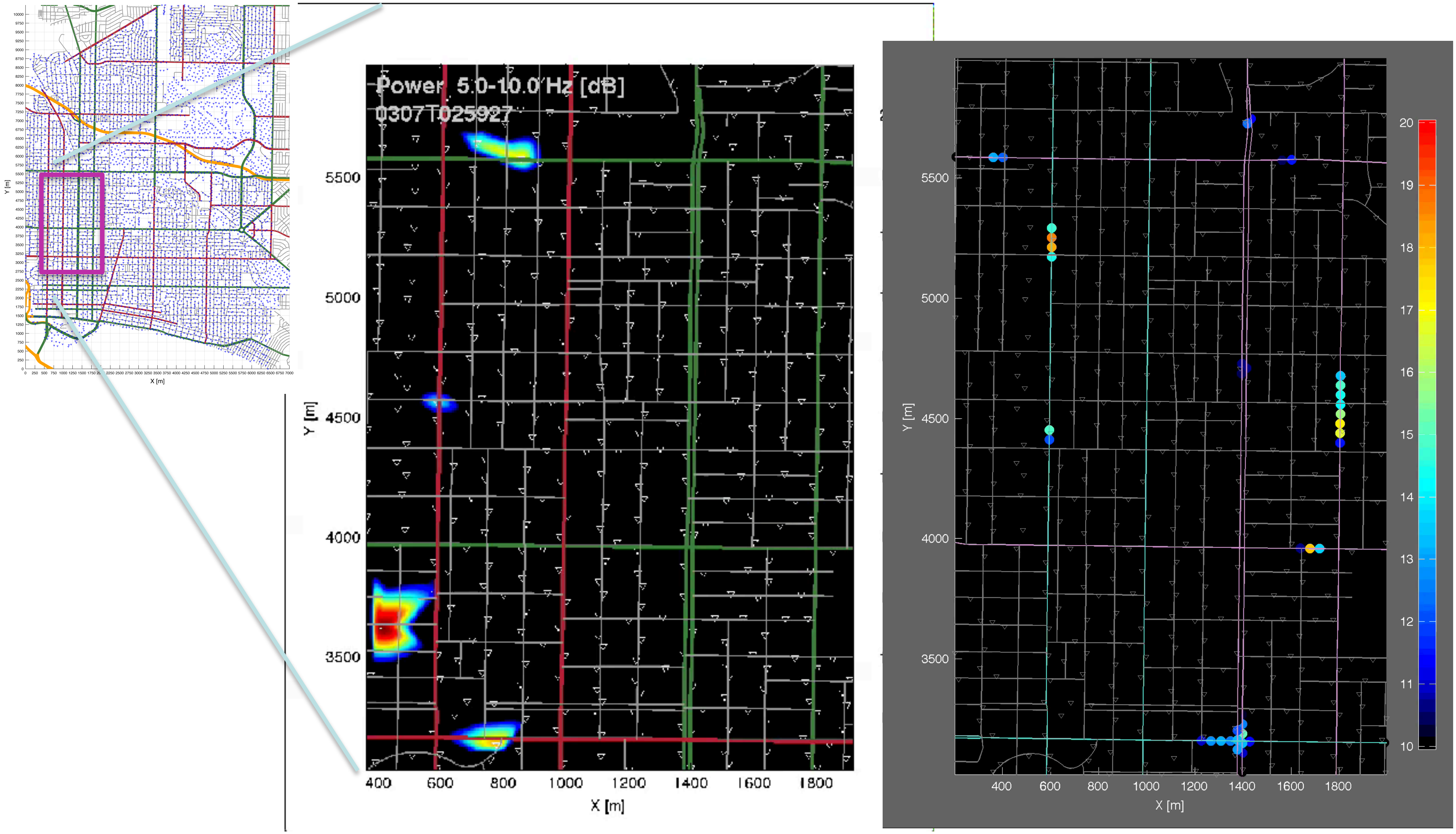


Based on 9400 time windows x 10 frequency bins.
Each dot is the center of a cluster. 90% of the clusters cover <1.5% of the area.
Few false detections

Riahi, Gerstoft, Signal Processing 2017

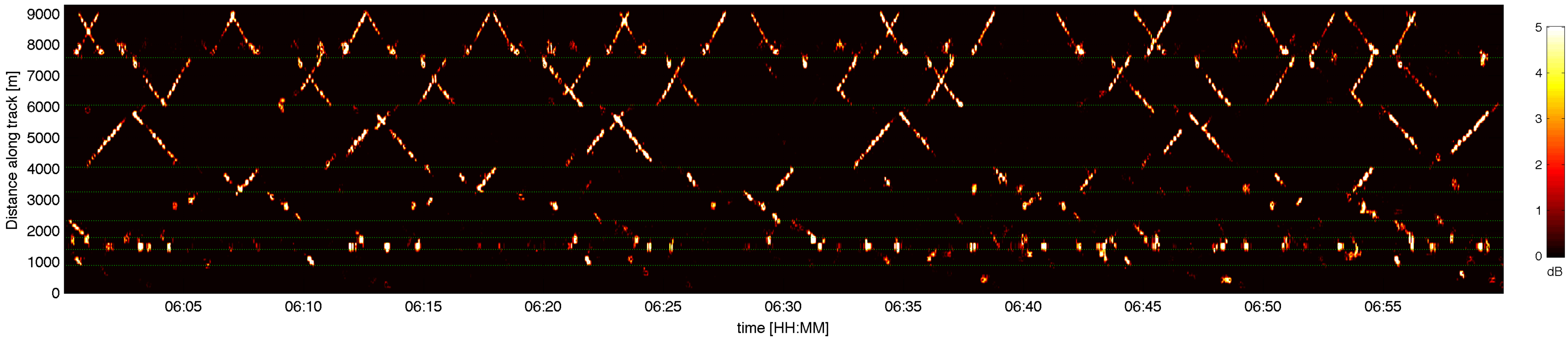
Noise Tracking of Cars/Trains/Airplanes

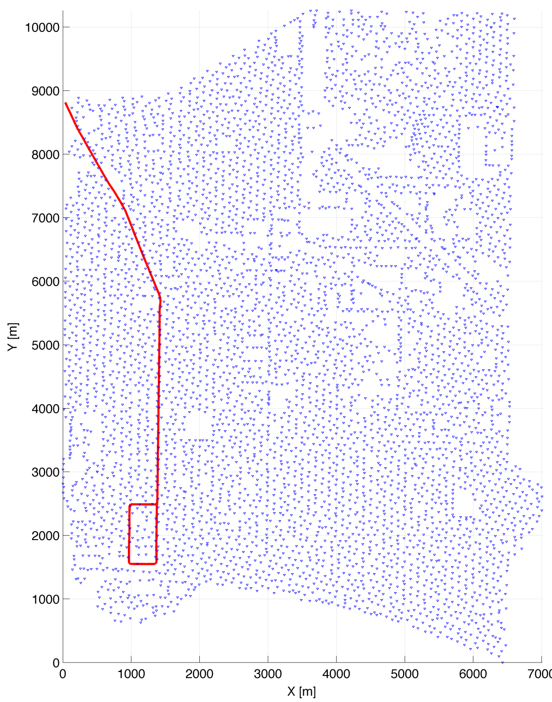
5200 element Long Beach array (Dan Hollis)



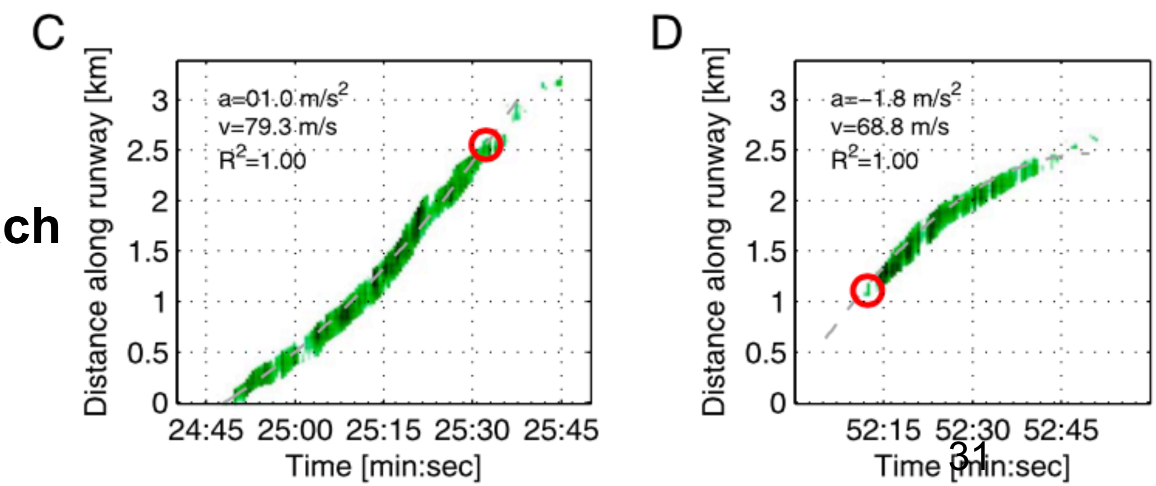
Riahi, Gerstoft, The seismic traffic footprint: Tracking trains, aircraft, and cars seismically, GRL 2015

Noise Tracking of Cars/Trains/Airplanes





March 7th, 6-7am, rush hour, Blue Line

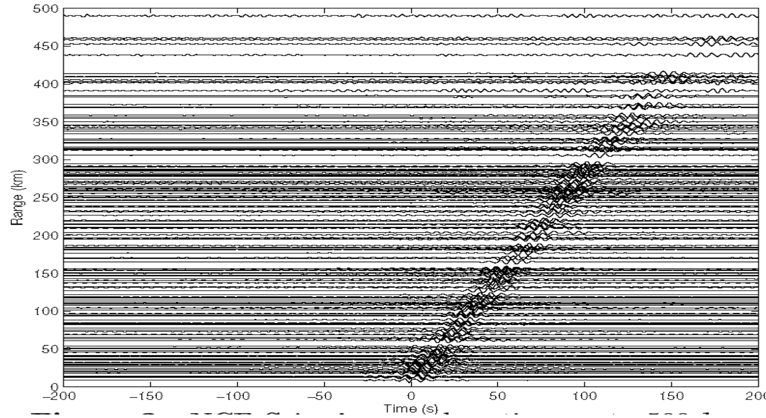


Accelerating airplane on Long Beach airport runway, moving northwest and taking off at about 120 mi/h.

Riahi, Gerstoft, GRL 2015

Travel time tomography

Travel times from noise cross-correlations



distance = speed x time

slowness = 1/speed

- Task: Given travel times, estimate regional phase speed distribution

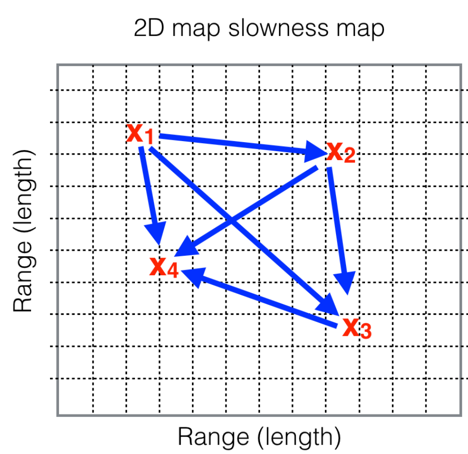
$$\mathbf{d} = \mathbf{A}\mathbf{m} + \mathbf{n},$$

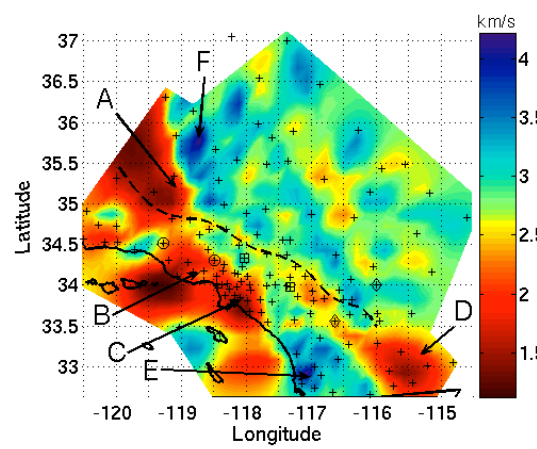
$\mathbf{d}:$ M travel times

$\mathbf{A}:$ "Tomography matrix": ray paths through the discretized map

$\mathbf{m}:$ N-pixel slowness image

Slowness map and measurements
- stations in red
- rays in blue





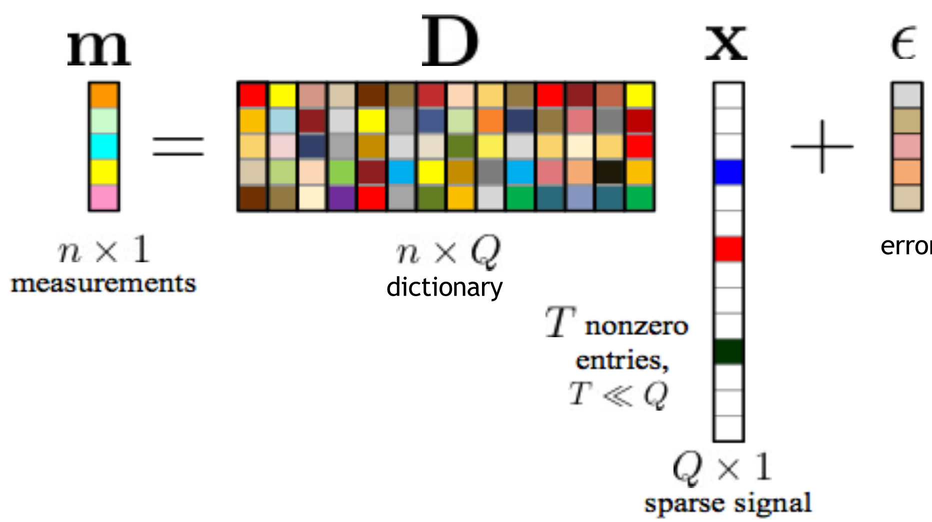
Low Velocity Region~Sedimentary basins A: San Joaquin, B: Ventura, C: L.A., D: Salton Sea, E: Peninsular range, F: Sierra Nevada

Sparse models and dictionaries

- Sparse modeling assumes each signal model can be reconstructed from a few vectors from a large set of vectors, called a dictionary **D**
- Adds auxiliary sparse model to measurement model

$$\mathbf{d} = \mathbf{A}\mathbf{m} + \mathbf{n},\ \ \mathbf{m} \approx \mathbf{D}\mathbf{x} \text{ and } |\mathbf{x}| \ll Q$$

- Optimization changes from estimating **m** to estimating sparse coefficients **x**



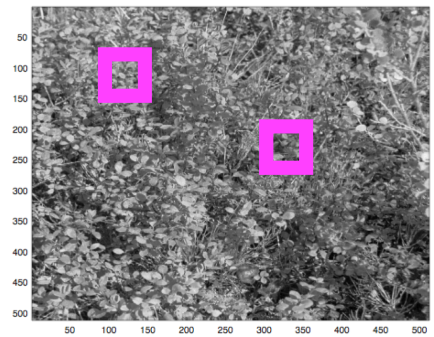
- Sparse objective: $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{D}\mathbf{x} - \mathbf{d}\|_2 \text{ subject to } \|\mathbf{x}\|_0 \leq T$

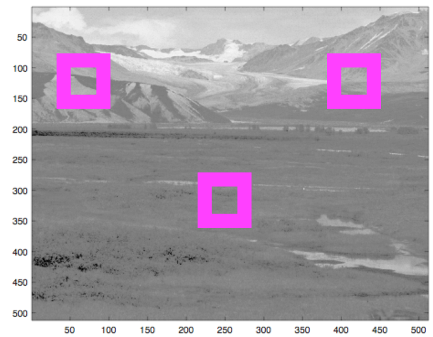
Dictionary learning and sparsity

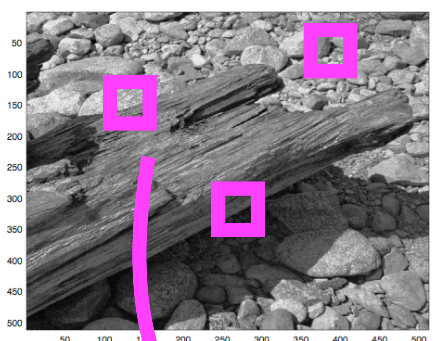
unsupervised

- Dictionary learning obtains "optimal" sparse modeling dictionaries directly from data
- Dictionary learning was developed in neuroscience (a.k.a. sparse coding) to help understand mammalian visual cortex structure
- Assumes (1) Redundancy in data: image patches are repetitions of a few elemental shapes; and (2) Sparsity: each patch is represented with few atoms from dictionary

"Natural" images, patches shown in **magenta**

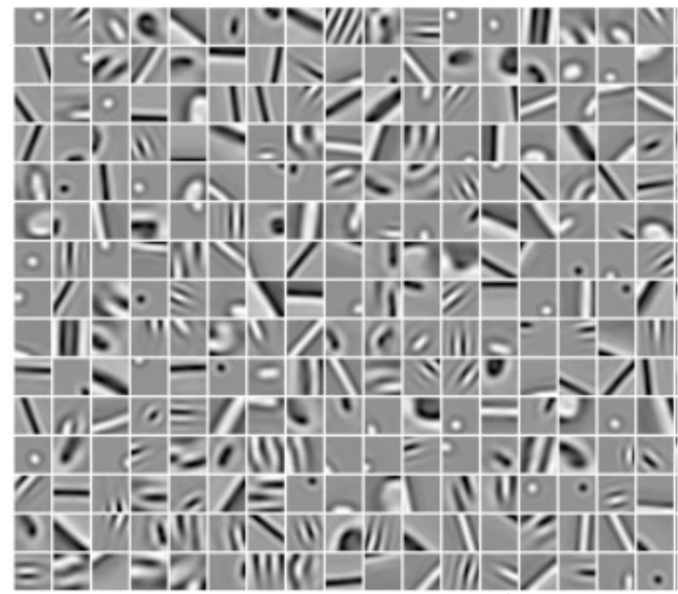






- Each patch is signal $\mathbf{y}_i$
- Set of all patches $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_I]$

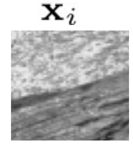
Learn dictionary **D** describing $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_I]$



Olshausen 2009

Sparse model for patch $\mathbf{y}_i$ composed of few atoms from **D**

$$\hat{\mathbf{x}}_i = \underset{\mathbf{x}_i}{\arg\min} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \le T$$

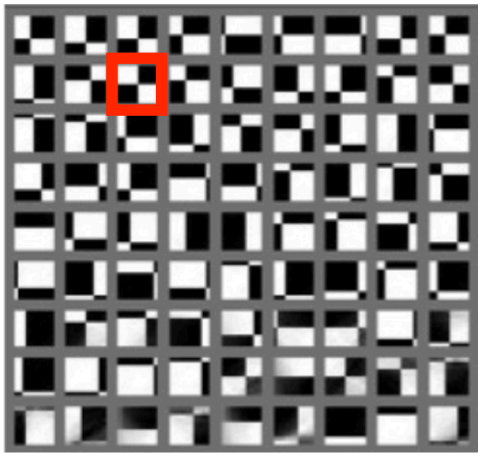
$\mathbf{y} =$  $=$  $x_1 +$  $x_2 + \ldots$

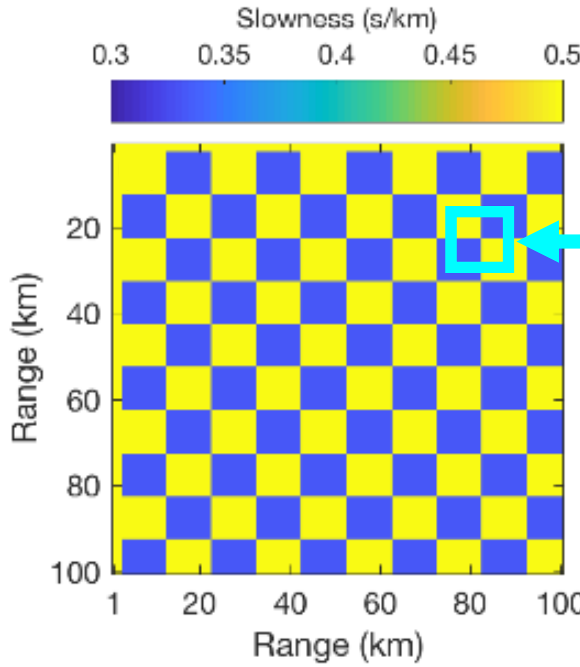
Bianco 2018, 2019

Checkerboard dictionary example

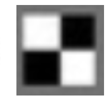
Dictionary $\mathbf{D}$

Slowness





$$\mathbf{y} = \mathbf{R}_i\mathbf{s} = \mathbf{D}\mathbf{x}_i$$

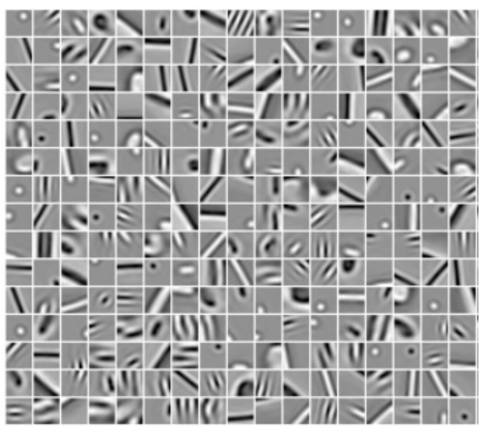
$\mathbf{R}_i\mathbf{s} =$  x_i

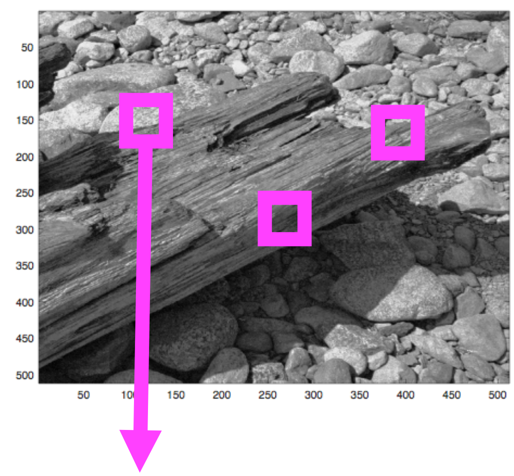
10x10 pixel patches

$$\mathbf{D} \in \mathbb{R}^{n\times Q}$$

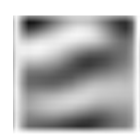
$$\mathbf{R}_i \in \{0,1\}^{n\times N}$$

Natural image



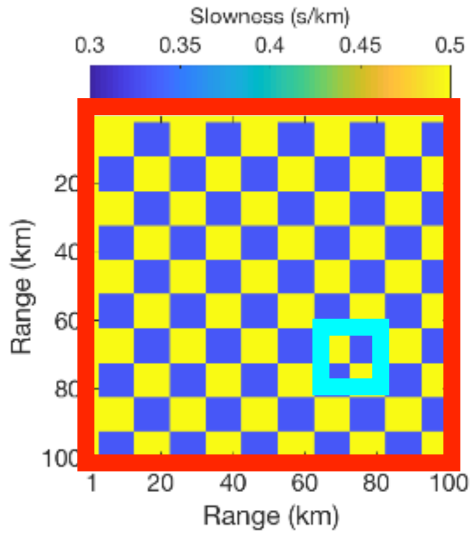


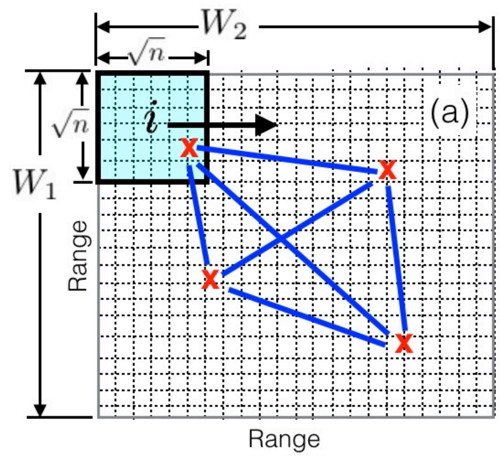
$$\widehat{\mathbf{x}}_i = \arg\min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \le T$$

$\mathbf{y} =$  $=$  $x_1 +$  $x_2 + \ldots$

Bianco 2018, 2019

LST slowness image and sampling





Pixels and “patches”

Slowness map and measurements
- stations in red
- rays in blue

Slowness map and sampling:

- Discrete slowness map *N=W₁ x W₂* pixels
- I overlapping $\sqrt{n} \times \sqrt{n}$ pixel patches
- *M* straight-ray paths

Tomography matrix (straight ray) $\mathbf{A} \in \mathbb{R}^{M \times N}$

Slowness dictionary $\mathbf{D} \in \mathbb{R}^{n \times Q}$

$Q \ll I$

“Local” model $\widehat{\mathbf{x}}_i = \underset{\mathbf{x}_i}{\arg\min} \|\mathbf{R}_i \mathbf{s}_\mathrm{s} - \mathbf{D}\mathbf{x}_i\|_2^2 \text{ subject to } \|\mathbf{x}_i\|_0 = T$

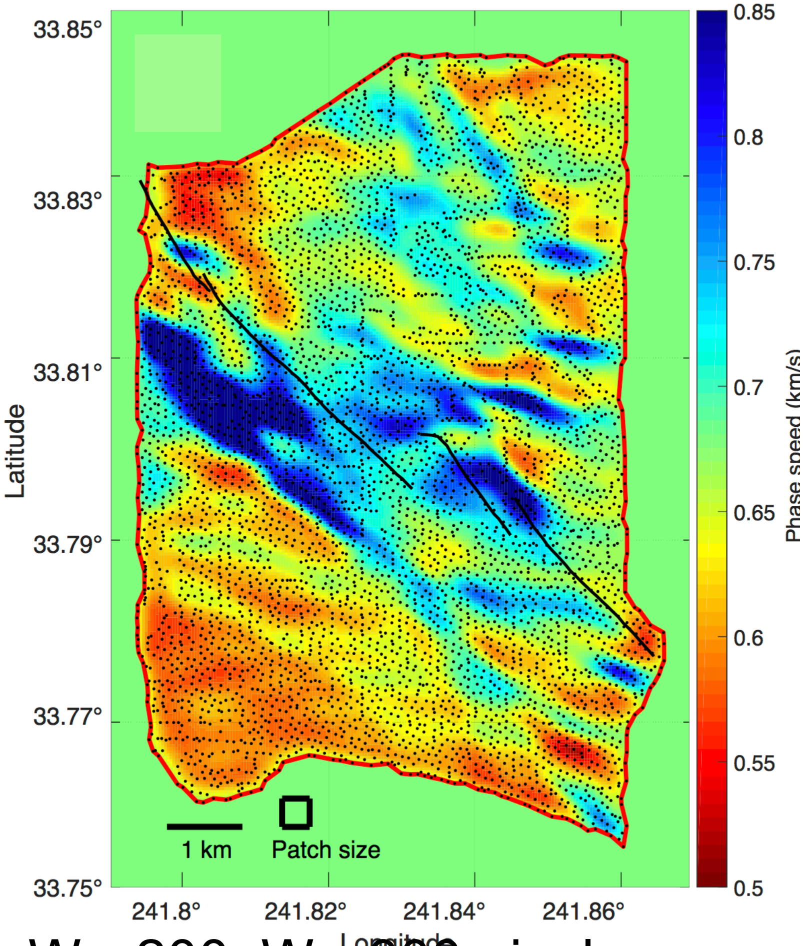
“Global” model $\mathbf{t} = \mathbf{A}\mathbf{s}_\mathrm{g} + \epsilon$ $\widehat{\mathbf{s}}_\mathrm{g} = \underset{\mathbf{s}_\mathrm{g}}{\arg\min} \|\mathbf{t} - \mathbf{A}\mathbf{s}_\mathrm{g}\|_2^2 + \lambda_1 \|\mathbf{s}_\mathrm{g} - \mathbf{s}_\mathrm{s}\|_2^2,$

Bayesian formulation

LST versus conventional tomography

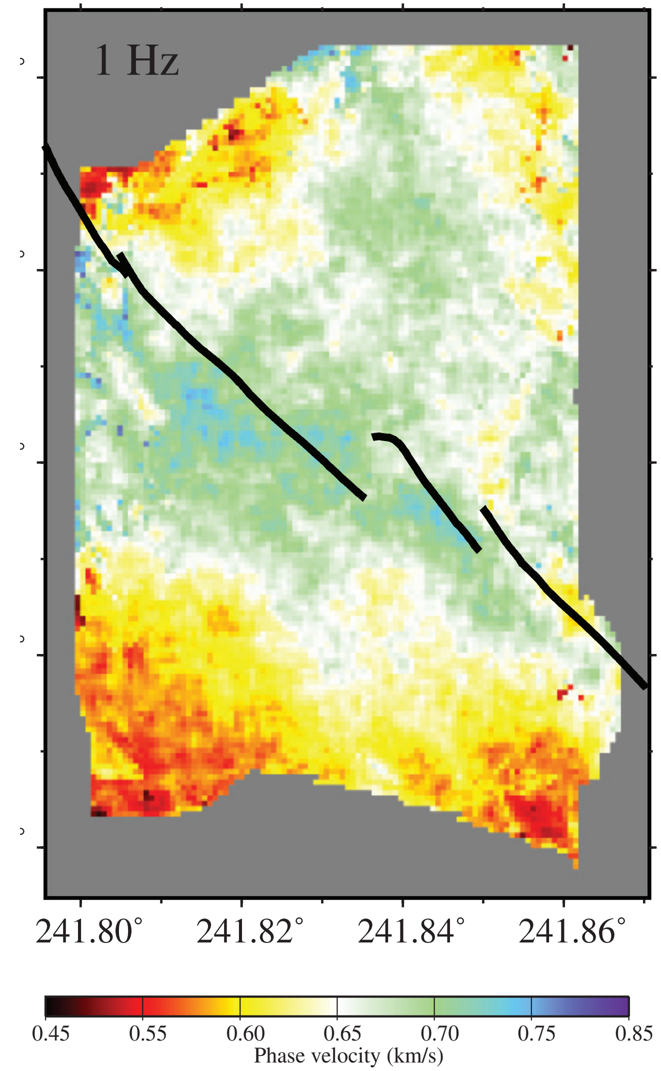
Both use same travel times (from Fan-Chi Lin), unsupervised

LST 3 mill rays



W_1=200, W_2=300 pixels
n=100, Q=200, T=1

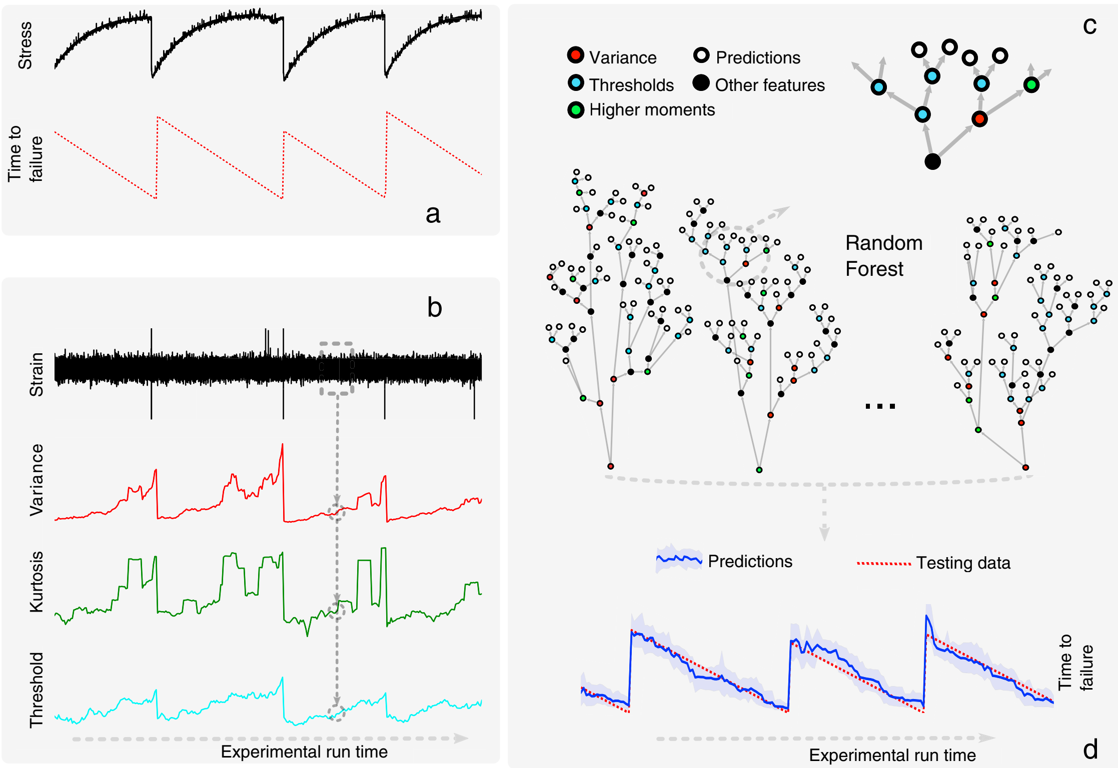
Fan-Chi Lin, Geophysics, 8mill Rays



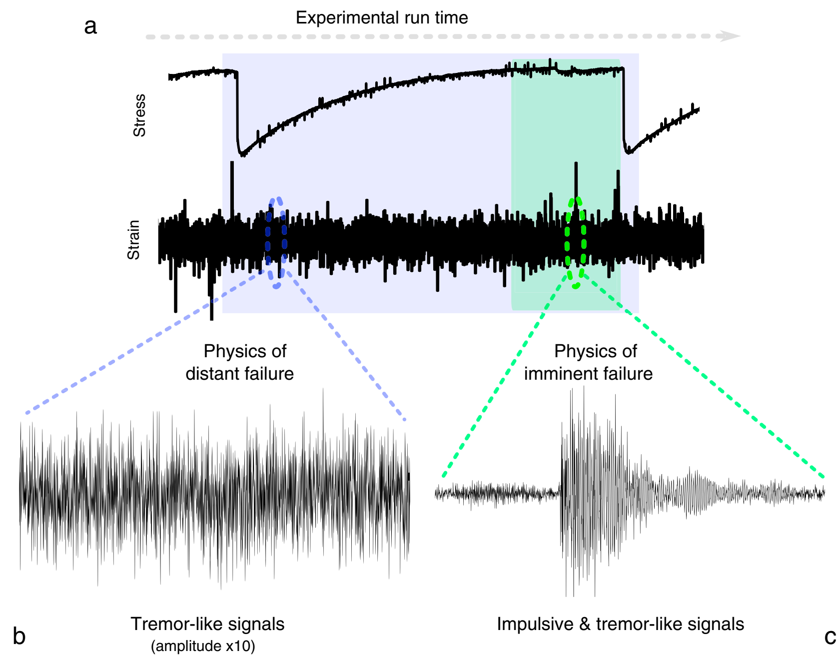
Bianco 2018, 2019

Predicting Earthquakes in Laboratory

- Kaggle competition.



Once they found a ML that could predict lab-EQ, they also could see the feature.



ML gives little or no insight into the model. We want the ML algorithm to provide a line of reasoning together with the calculated result. Not just the outcome of Bayes formalism.

=> That will come

First principles vs Data driven

	First principles	Data driven
Data	Small data	Big data to train
Domain expertise	High reliance on domain expertise	Results with little domain knowledge
Fidelity/ Robustness	Universal link can handle non-linear complex relations	Limited by the range of values spanned by training data
Adaptability	Complex and time consuming derivation to use new relations	Rapidly adapt to new problems
Interpretability	Parameters are physical!	Physically agnostic, limited by the rigidity of the functional form

Perceived Importance. **Geophys** **SignalProc** **Peter** **Googl**

Summary

- Machine learning, big data, data science, artificial intelligence are similar.
- **Data science** has lots of opportunities in **geophysics**.
- Neural networks is one method. Similar methods are Support Vector Machines (SVM) and Random Forrest (RF). Use the latter for a first test.
- **Unsupervised learning** is more challenging than supervised learning
- We need explainable artificial intelligence. We want the ML algorithm to provide a line of reasoning together with the calculated result / fit / decision.

Actions: Download ML JASA review

- TRY http://playground.tensorflow.org

Can ML

- Replace CTBTO processing chain?
- Discover PDE (Partial differential equation) in video?
- Find sea mines?
- Design metamaterials?
- Predict earthquakes?
- Replace 50 years of array processing
- Source location in the ocean waveguide w/o training.

FINITO