Lecture 5 : Sparse Models

- Homework 3 discussion (Nima)

- Sparse Models Lecture
  - Reading : Murphy, Chapter 13.1, 13.3, 13.6.1
  - Reading : Peter Knee, Chapter 2

- Paolo Gabriel (TA) : Neural Brain Control

- After class
  - Project groups (Nima)
  - Installation Tensorflow, Python, Jupyter (TAs)

Homework 3 : Fisher Discriminant

$$P(C_1|x) = \frac{P(x|C_1)\,P(C_1)}{P(x)}$$

$$P(C_1|x) = P(C_2|x)$$

$$P(x|C_1)\,P(C_1) = P(x|C_2)\,P(C_2)$$

$$\Sigma = \Sigma_{C_1} = \Sigma_{C_2} \qquad \hat{\mu}_1, \hat{\mu}_2$$

$$(x-\mu_1)^T \Sigma^{-1} (x-\mu_1) = (x-\mu_2)^T \Sigma^{-1} (x-\mu_2)$$

$$\mathrm{trace}\left(\Sigma^{-1}\left[(x-\mu_1)^T(x-\mu_1) - (x-\mu_2)^T(x-\mu_2)\right]\right) = 0$$

$$\operatorname{tr}\left(\Sigma^{-1}\left[\left(\cancel{x^T x} - x^T\mu_1 - \mu_1^T x + \mu_1^T\mu_1 \cancel{- x^T x} + x^T\mu_2 + \mu_2^T x - \mu_2^T\mu_2\right)\right]\right) = 0$$

$$\operatorname{tr}\left(\Sigma^{-1}\left[\underline{x^T(\mu_2-\mu_1)} + \underline{(\mu_2-\mu_1)^T x} + \mu_1^T\mu_1 - \mu_2^T\mu_2\right]\right) = 0$$

$$\operatorname{tr}\left(\Sigma^{-1}\left(2x^T(\mu_2-\mu_1) + (\mu_2-\mu_1)^T(\mu_2+\mu_1^T)\right)\right) = 0$$

$$2(\mu_2-\mu_1)^T\Sigma^{-1}x + (\mu_2-\mu_1)^T\Sigma^{-1}(\mu_2+\mu_1^T) = 0$$

$$(\mu_2-\mu_1)^T\Sigma^{-1}\left(2x-(\mu_2+\mu_1)\right) = 0$$

$$(\mu_2-\mu_1)^T\Sigma^{-1}(x-\mu_0) = 0$$

$$w^T(x-x_0) = 0$$

Sparse model

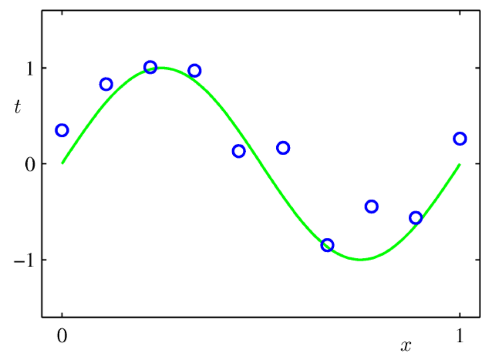
- Linear regression (with sparsity constraints)
- Slide 4 from Lecture 4

Linear regression: Linear Basis Function Models (1)

Generally

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

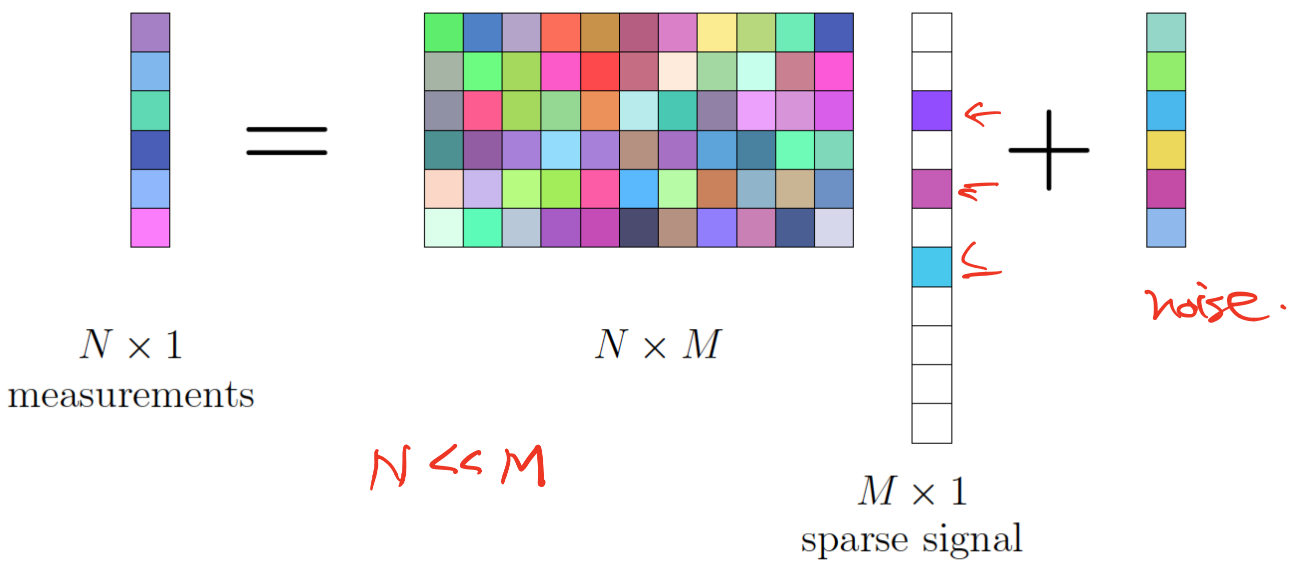
- where $\phi_j(x)$ are known as *basis functions*.
- Typically, $\phi_0(x) = 1$, so that w_0 acts as a bias.
- Simplest case is linear basis functions: $\phi_d(x) = x_d$.



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sparse model

Model : $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, $\quad$ $\mathbf{x}$ is sparse



k-sparsity

- y : measurements, $\quad$ A : dictionary
- n : noise, $\quad$ x : sparse weights
- Dictionary (A) − either from physical models or learned from data (dictionary learning)

Sparse processing

- Linear regression (with sparsity constraints)
  - An underdetermined system of equations has many solutions
  - Utilizing x is sparse it can often be solved
  - This depends on the structure of A (RIP – Restricted Isometry Property)

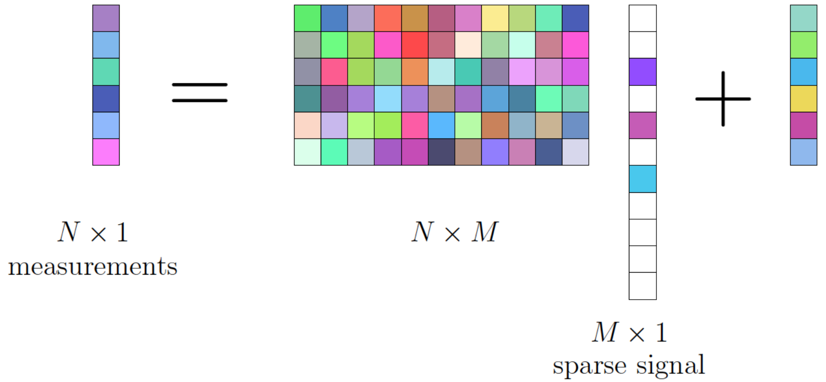
- Various sparse algorithms
  - Convex optimization (Basis pursuit / LASSO / L_1 regularization)
  - Greedy search (Matching pursuit / OMP)
  - Bayesian analysis (Sparse Bayesian learning / SBL)

- Low-dimensional understanding of high-dimensional data sets

- Also referred to as compressive sensing (CS)

Different applications, but the same algorithm

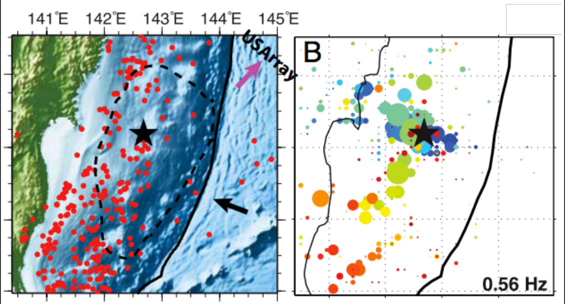
Model : $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, $\mathbf{x}$ is sparse



y	A	x
Frequency signal	DFT matrix	Time-signal
Compressed-Image	Random matrix	Pixel-image
Array signals	Beam weight	Source-location
Reflection sequence	Time delay	Layer-reflector

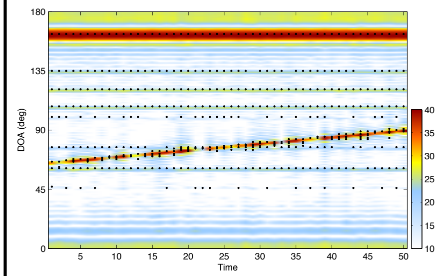
CS approach to geophysical data analysis

CS of Earthquakes



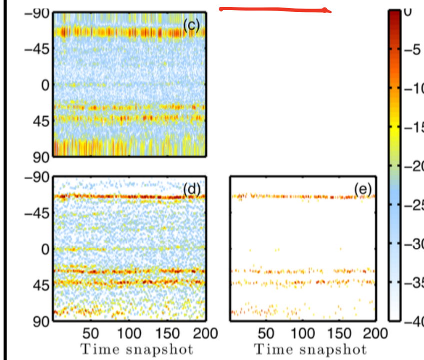
Yao, GRL 2011, PNAS 2013

Sequential CS



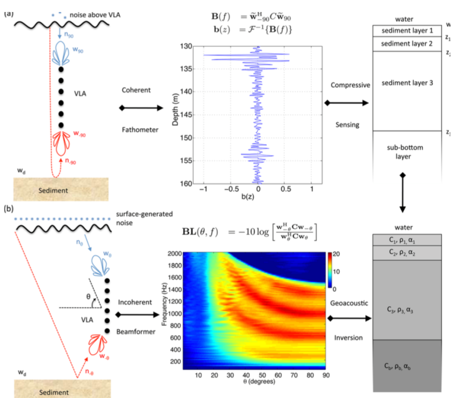
Mecklenbrauker, TSP 2013

CS beamforming



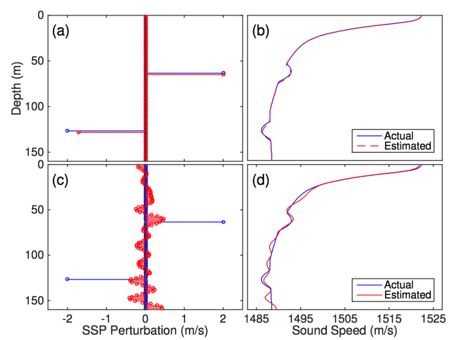
Xenaki, JASA 2014, 2015
Gerstoft JASA 2015

CS fathometer



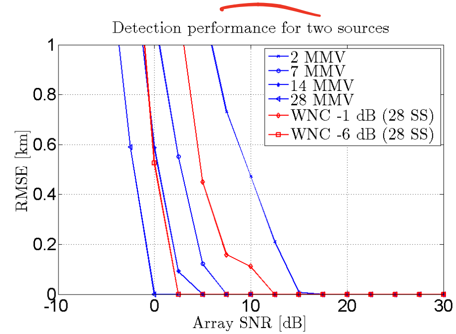
Yardim, JASA 2014

CS Sound speed estimation



Bianco, JASA 2016

CS matched field



Gemba, JASA 2016

Sparse signals /compressive signals are important

- We don’t need to sample at the Nyquist rate

- Many signals are sparse, but are solved under non-sparse assumptions
  - Beamforming
  - Fourier transform
  - Layered structure

- Inverse methods are inherently sparse: We seek the simplest way to describe the data Parameter Est.

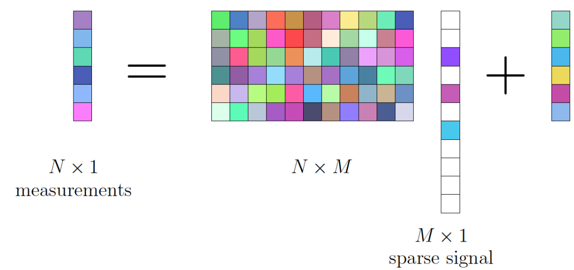
- All this requires **new developments**
  - Mathematical theory
  - New algorithms (interior point solvers, convex optimization)
  - Signal processing
  - New applications/demonstrations

Sparse Recovery

- We try to find the sparsest solution which explains our noisy measurements
- L_0-norm

$$|x|_0 = \sum_m \underbrace{|x_m|^0}_{\substack{1 \text{ if } x_m \neq 0 \\ 0 \text{ if } 0}}$$

Model : $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, $\quad \mathbf{x}$ is sparse



$$|x|_0 = 3$$

- Here, the L_0-norm is a shorthand notation for counting the number of non-zero elements in x.

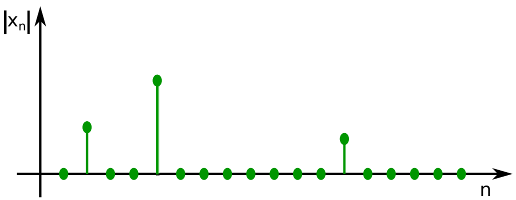
Sparse Recovery using L_0-norm

Underdetermined problem

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

Prior information

$$\mathbf{x}: \text{ K-sparse}, \; K \ll N$$



$$\|\mathbf{x}\|_0 = \sum_{n=1}^{N} 1_{x_n \neq 0} = K$$ L_0 norm

Not really a norm: $\|a\mathbf{x}\|_0 = \|\mathbf{x}\|_0 \neq |a|\,\|\mathbf{x}\|_0$

There are only few sources with unknown locations and amplitudes

- L_0-norm solution involves exhaustive search
- Combinatorial complexity, not computationally feasible

L_p-norm

$$\| \cancel{x} \|_p = \left(\sum_{m=1}^{M} | x_m |^p \right)^{1/p} \quad \text{for } p > 0$$

L_1 : $\|x\|_p = \sum_m |x_m|$

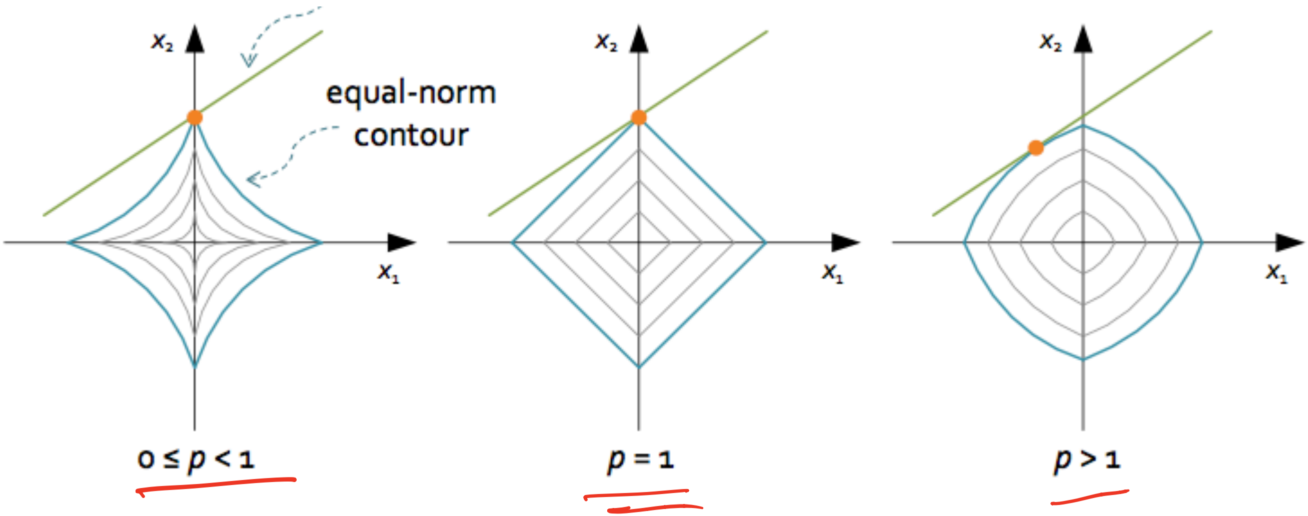
L_2 : $\|x\|_2 = \sum_m |x_m|^2$

L_0 : $p = 0$

- Classic choices for p are 1, 2, and ∞.
- We will misuse notation and allow also p = 0.

L_p-norm (graphical representation)

$$\|x\|_p = \left(\sum_{m=1}^{M} |x_m|^p \right)^{1/p}$$



Solutions for sparse recovery

- Exhaustive search
  - L_0 regularization, not computationally feasible
- Convex optimization
  - L_1 regularization / Basis pursuit / LASSO
- Greedy search
  - Matching pursuit / Orthogonal matching pursuit (OMP)
- Bayesian analysis
  - Sparse Bayesian Learning (SBL)
- Regularized least squares
  - L_2 regularization, reference solution, not actually sparse

Regularized least squares

regularizer

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

L_2 norm of weights

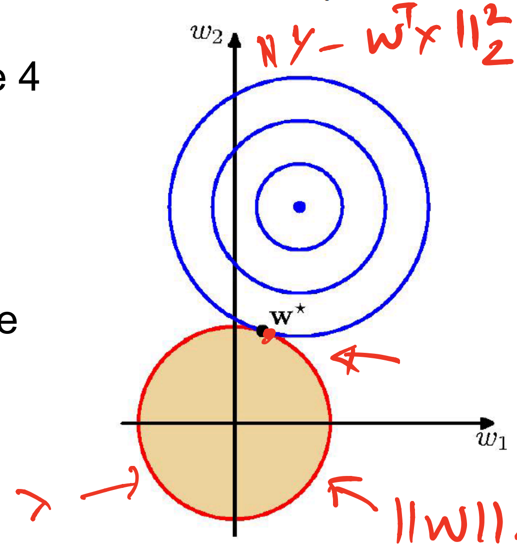
The squared weights penalty is mathematically compatible with the squared error function, giving a closed form for the optimal weights:

Solution is Not Sparse

$$\mathbf{w}^* = (\lambda \mathbf{I} + \mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$$

- Slides 8/9, Lecture 4
- Regularized least squares solution
- Solution not sparse

A picture of the effect of the regularizer



- The overall cost function is the sum of two parabolic bowls.
- The sum is also a parabolic bowl.
- The combined minimum lies on the line between the minimum of the squared error and the origin.
- The L2 regularizer just **shrinks** the weights.

$w_1, w_2 \neq 0$

Basis Pursuit / LASSO / L_1 regularization

- The L_0-norm minimization is not convex and requires combinatorial search making it computationally impractical

- We make the problem convex by substituting the L_1-norm in place of the L_0-norm

$$\min_{x} \|\mathbf{x}\|_1 \qquad \text{subject to } \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 < \varepsilon$$

$\|x\|_0$

- This can also be formulated as

$$\min_{x} \|Ax-y\|_2^2 + \mu\|x\|_1$$

regularizer

convex opt.

CVX - MATLAB

x - sparse

Sparsity is function of μ.

The unconstrained -LASSO- formulation

Constrained formulation of the ℓ_1-norm minimization problem:

$$\widehat{\mathbf{x}}_{\ell_1}(\epsilon) = \arg\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon$$

Unconstrained formulation in the form of least squares optimization with an ℓ_1-norm regularizer:

$$\widehat{\mathbf{x}}_{\text{LASSO}}(\mu) = \arg\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_1$$

For every ϵ exists a μ so that the two formulations are equivalent

Regularization parameter : μ

Basis Pursuit / LASSO / L_1 regularization

- Why is it OK to substitute the L_1-norm for the L_0-norm?

- What are the conditions such that the two problems have the same solution?

Good?

$$\min_x \|x\|_1 \quad \text{subject to } \|Ax - b\|_2 < \varepsilon$$

L_1

$$\min_x \|x\|_0 \quad \text{subject to } \|Ax - b\|_2 < \varepsilon$$

L_0

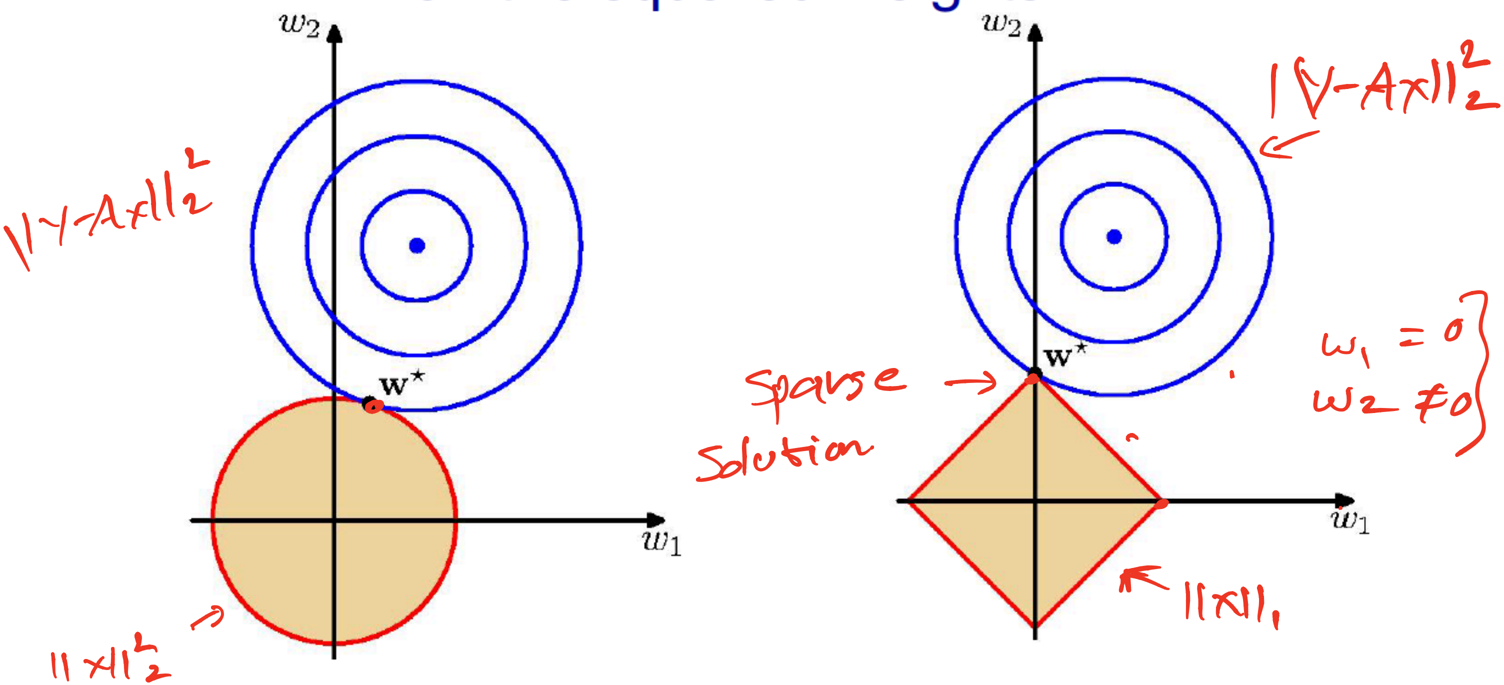
- Restricted Isometry Property (RIP)

$$(1 - \delta_s)\|\boldsymbol{u}\|_2 \leq \|\boldsymbol{A}_S\boldsymbol{u}\|_2 \leq (1 + \delta_s)\|\boldsymbol{u}\|_2$$

Dictionary.

Geometrical view (Figure from Bishop)

Geometrical view of the lasso compared with a penalty on the squared weights



L_2 regularization L_1 regularization

Regularization parameter selection

$\mu \to \infty$

$L \to \|x\|_1$

The objective function of the LASSO problem:

$$\min \quad L(\mathbf{x}, \mu) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \underbrace{\mu}_{\text{Reg.}}\|\mathbf{x}\|_1$$

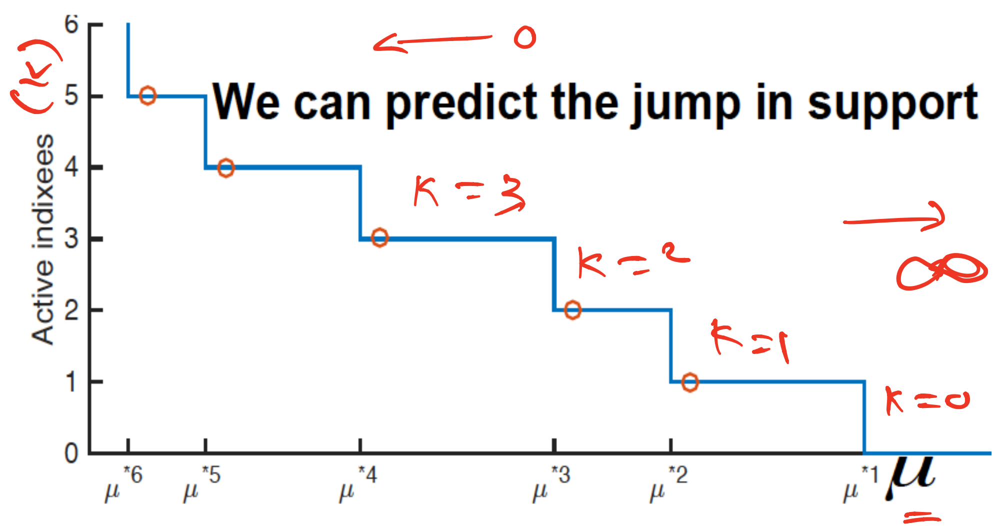
$\mu \to 0$

$L \to \|y - Ax\|_2^2$

- Regularization parameter : μ
- Sparsity depends on μ
- μ large, x = 0
- μ small, non-sparse

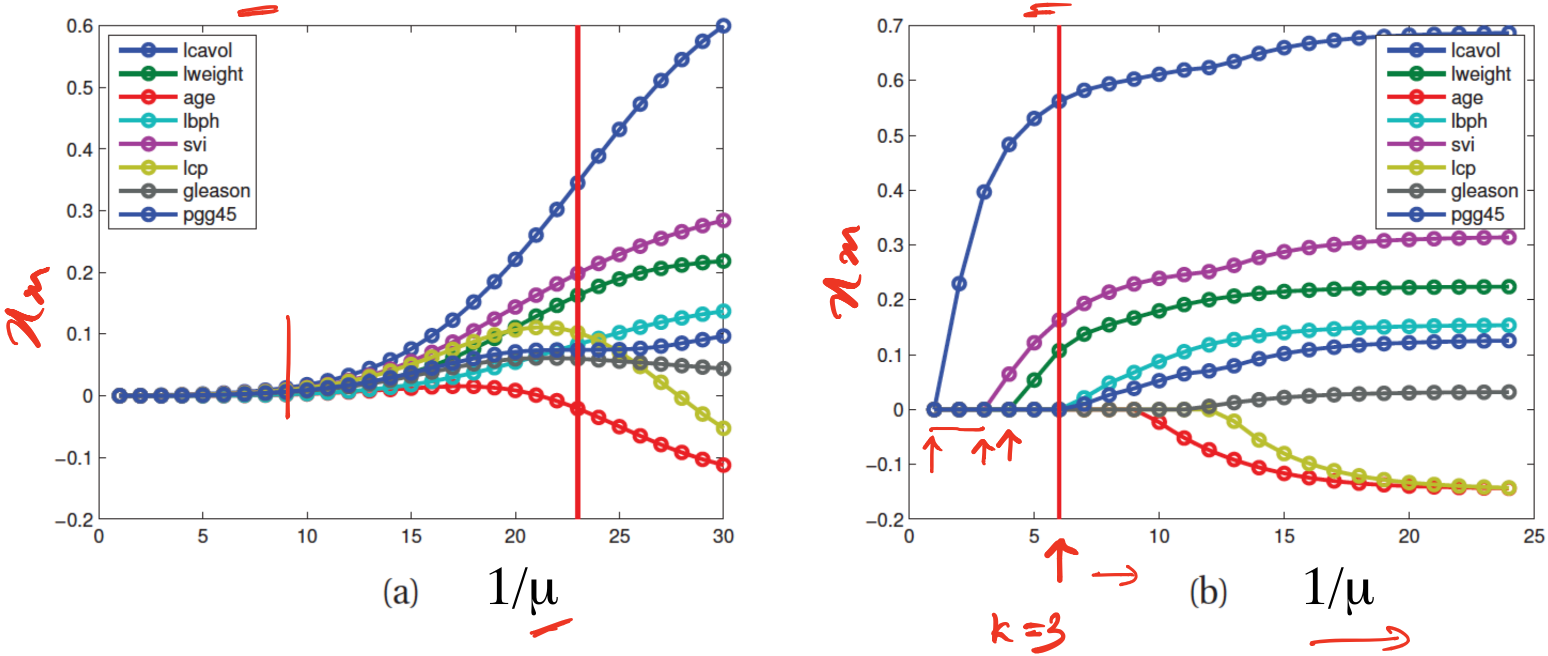
$\|x\|_0 = k$

L_0



Regularization Path (Figure from Murphy)

L$_2$ regularization   L$_1$ regularization

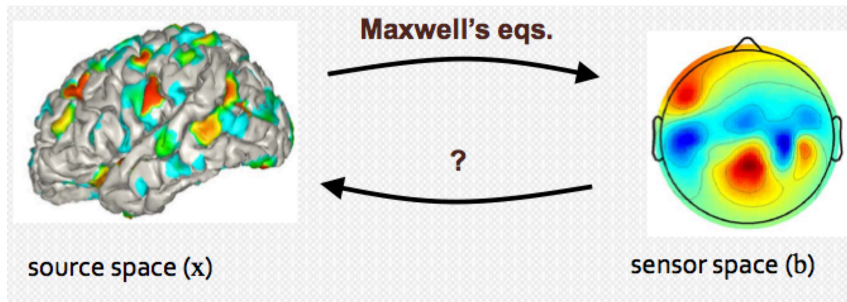


- As regularization parameter μ is decreased, more and more weights become active
- Thus μ controls sparsity of solutions

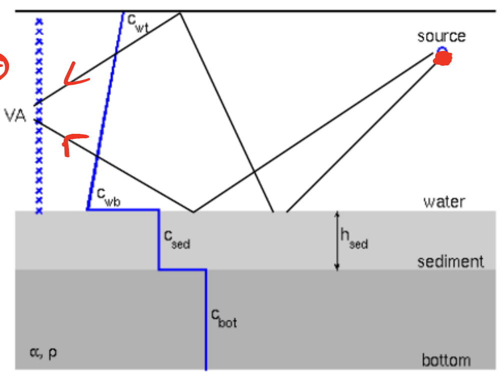
Applications

- MEG/EEG/MRI source location (earthquake location)
- Channel equalization
- Compressive sampling (beyond Nyquist sampling)
- Compressive camera!

- Beamforming
- Fathometer
- Geoacoustic inversion
- Sequential estimation



Sensor Array



Ocean

Beamforming / DOA estimation

DOA estimation with sensor arrays

sensor Array measurements

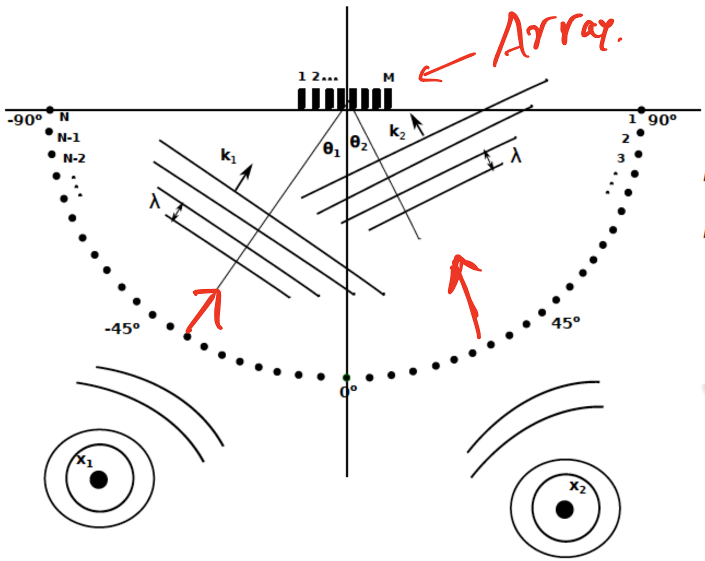
$$y_m = \sum_n x_n e^{j\frac{2\pi}{\lambda} r_m \sin\theta_n}$$

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

phase of arrivals

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

$\theta : -90 : 1 : 90 \}$



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

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

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

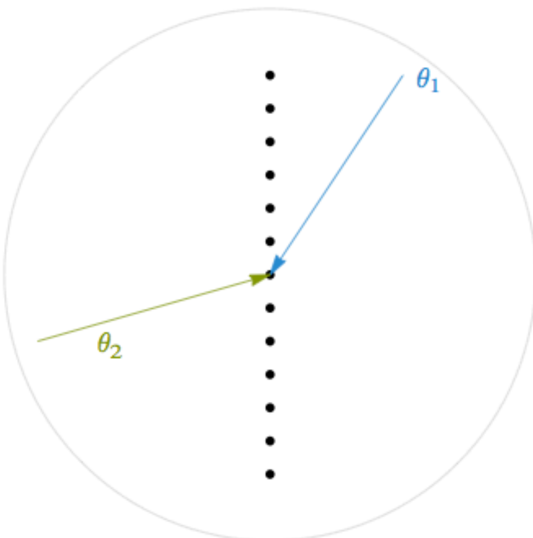
$$\mathbf{a}_n = \frac{1}{\sqrt{M}}[e^{j\frac{2\pi}{\lambda} r_1 \sin\theta_n}, \cdots, e^{j\frac{2\pi}{\lambda} r_M \sin\theta_n}]^T$$

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

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

The DOA estimation is formulated as a linear problem

Direction of arrival estimation



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

True DOA is sparse in the angle domain

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

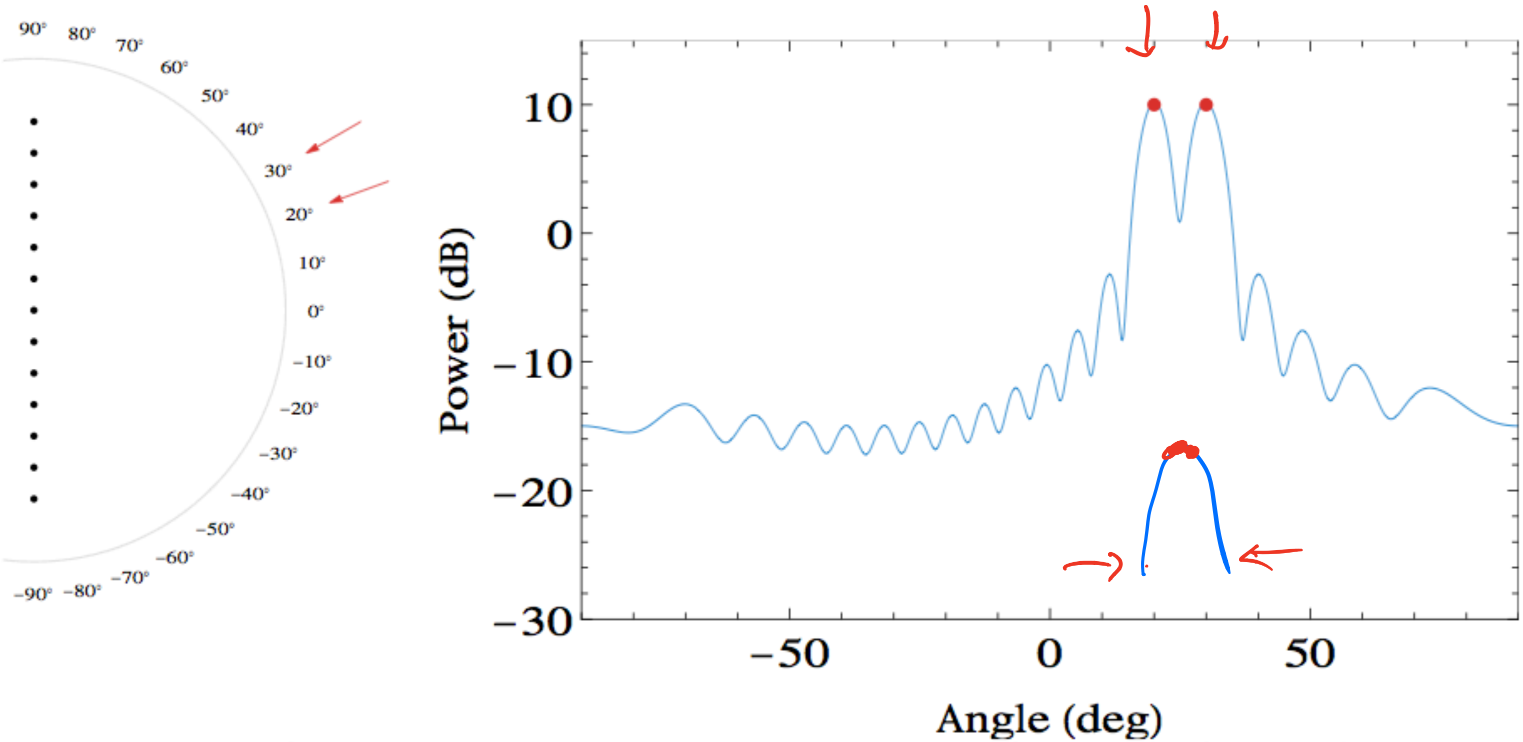
x

Amplitudes

Conventional beamforming

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

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

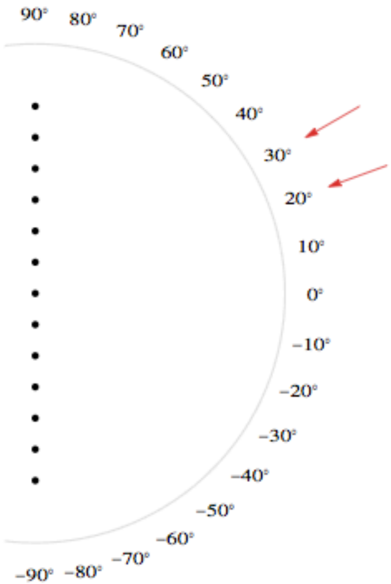


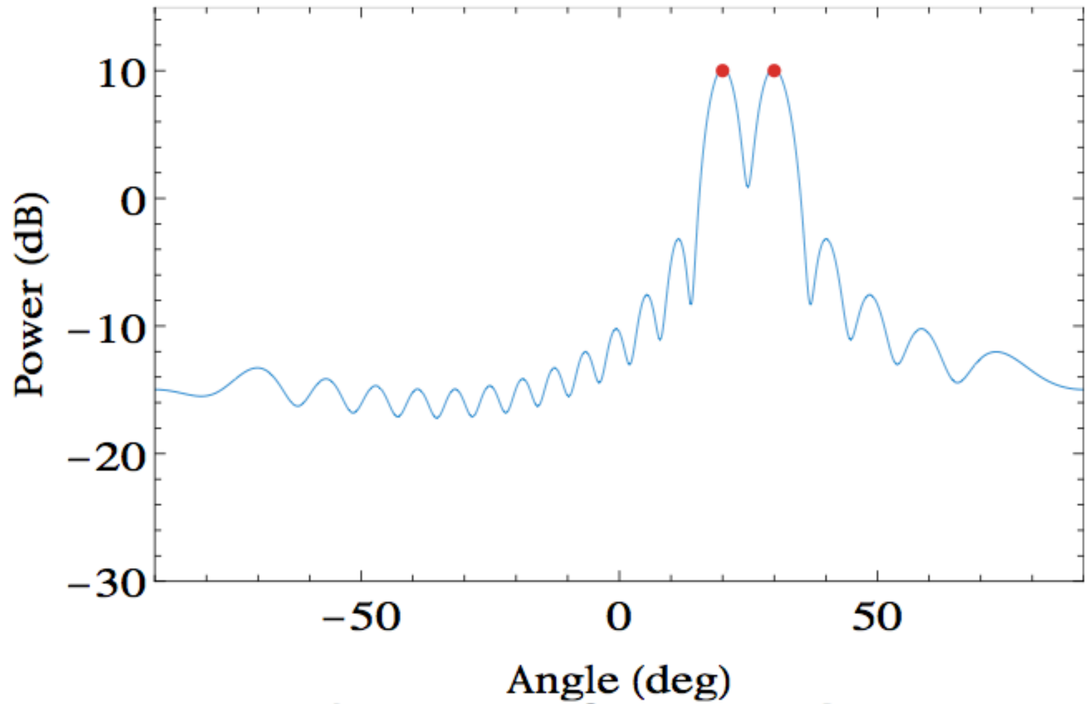
ULA, half-wavelength spacing, $N = 20$ sensors, $\theta_1 = 20°$, $\theta_2 = 30°$,

Conventional beamforming

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

$$\min \|\mathbf{x}\|_2^2 \text{ subject to } \|\mathbf{A}\mathbf{x} = \mathbf{b}\|_2^2 < \varepsilon$$





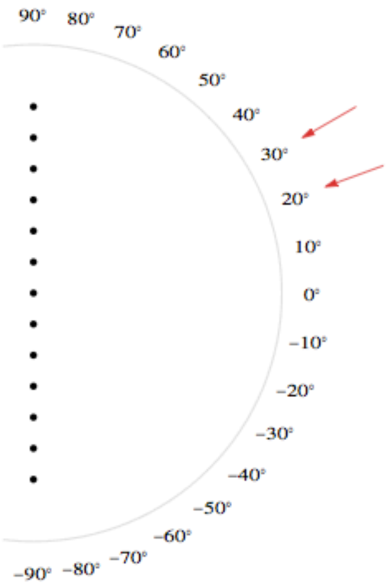
A is an overcomplete dictionary of candidate DOA vectors. Columns span $-90°$ to $90°$ in steps of $1°$ ($M = 181$).

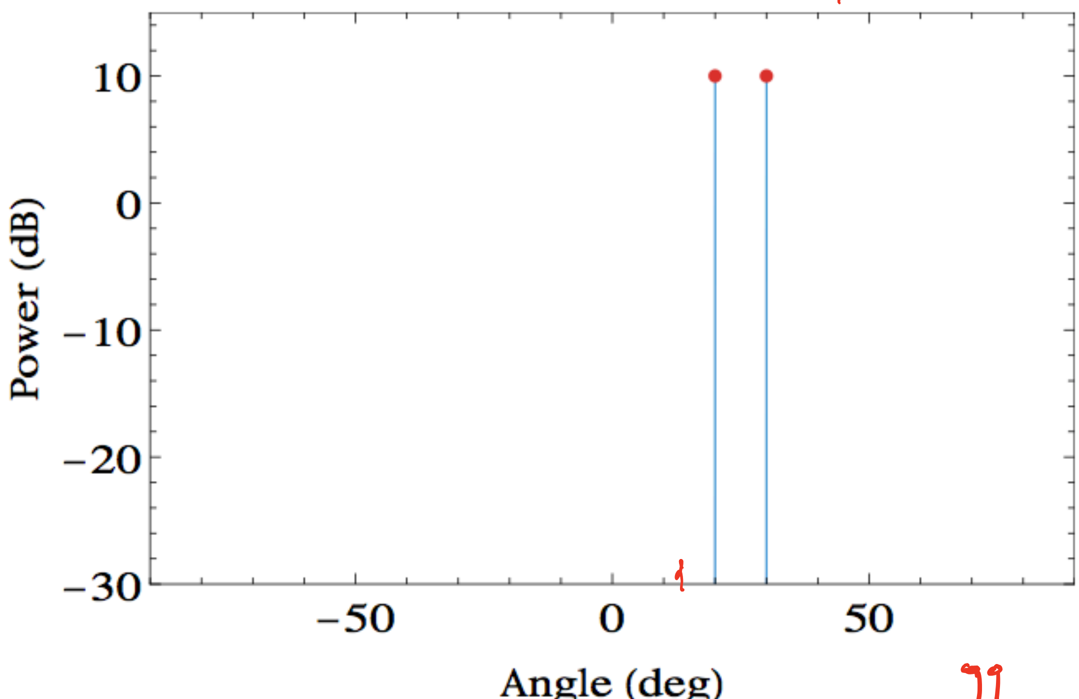
ℓ_1 minimization

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

$$\min \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{A}\mathbf{x} = \mathbf{b}\|_2^2 \le \varepsilon$$

Sparse





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

Additional Resources