Least Squares and Linear Systems

Gabriel Peyré

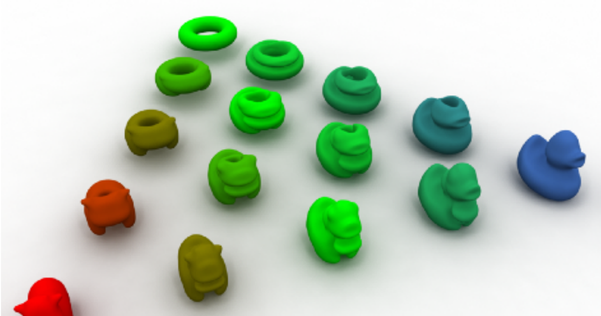
www.numerical-tours.com

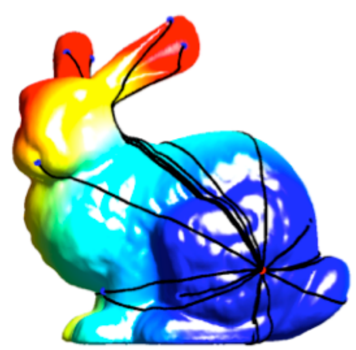
Mathematical Coffees

Huawei-FSMP joint seminars

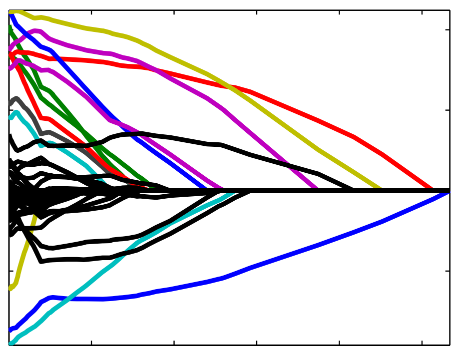
https://mathematical-coffees.github.io

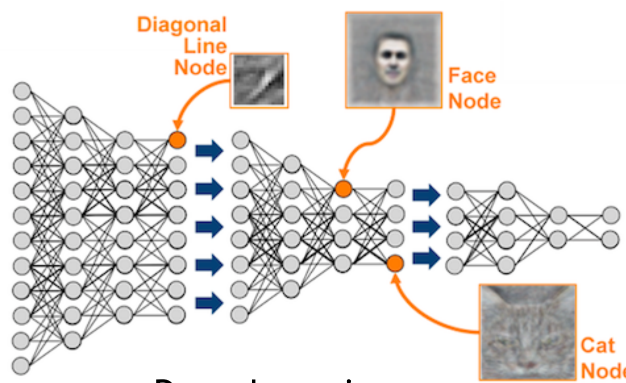
Organized by: Mérouane Debbah & Gabriel Peyré


Optimal Transport

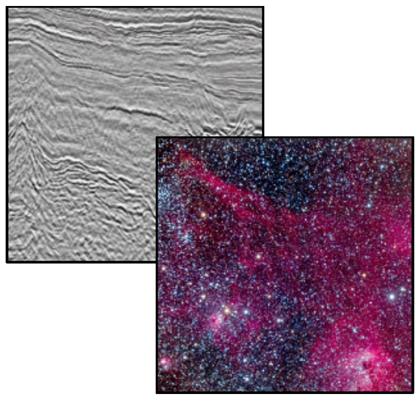

Geodesics

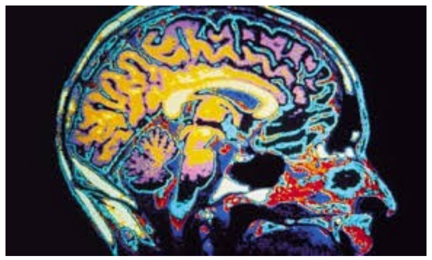

Meshes

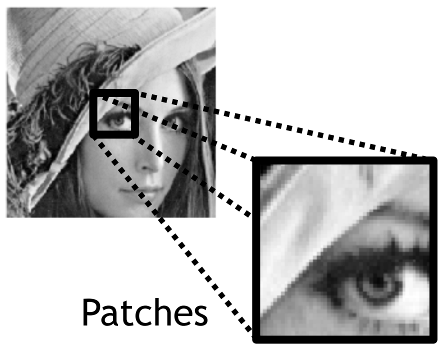

Optimization

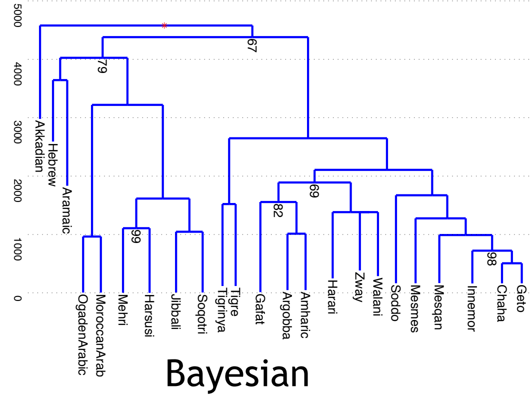


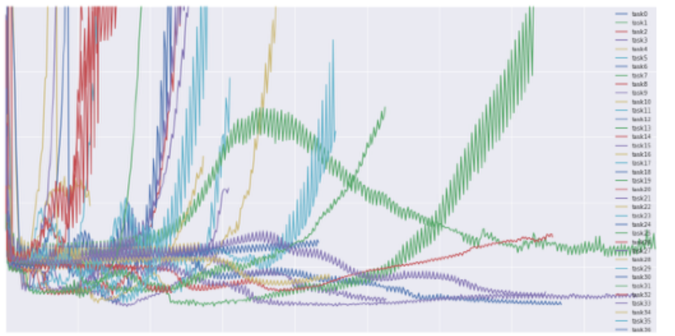
Deep Learning


Sparsity


Neuro-imaging


Patches


Bayesian

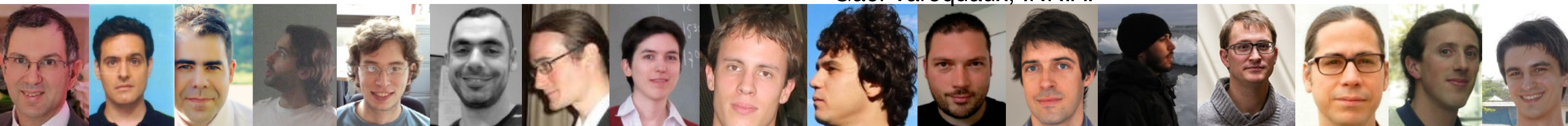

Parallel/Stochastic

Alexandre Allauzen, Paris-Sud.
Pierre Alliez, INRIA.
Guillaume Charpiat, INRIA.
Emilie Chouzenoux, Paris-Est.

Nicolas Courty, IRISA.
Laurent Cohen, CNRS Dauphine.
Marco Cuturi, ENSAE.
Julie Delon, Paris 5.

Fabian Pedregosa, INRIA.
Guillaume Lecué, CNRS ENSAE
Julien Tierny, CNRS and P6.
Robin Ryder, Paris-Dauphine.
Gael Varoquaux, INRIA.

Jalal Fadili, ENSICaen.
Alexandre Gramfort, INRIA.
Matthieu Kowalski, Supelec.
Jean-Marie Mirebeau, CNRS,P-Sud.



Regression Problems

(Noisy) observations $(t_i, y_i)_{i=1}^m$, try to infer $y \approx f(t)$.

$$f : \mathbb{R} \to \mathbb{R} \text{ (extend to any dimension)}$$

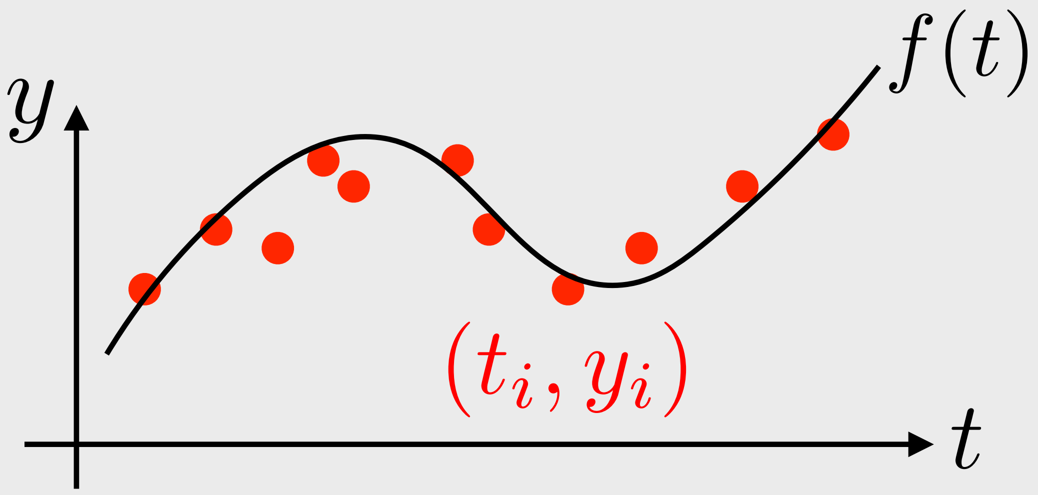
Expansion in a dictionary $(\varphi_j)_{j=1}^n$:

$$f(t) \stackrel{\text{def.}}{=} \sum_{j=1}^n x_j \varphi_j(t)$$

Polynomials: $\varphi_j(t) = t^{k_j}$.

Fourier: $\varphi_j(t) = e^{i\omega_j t}$.

Wavelets: $\varphi_j(t) = \psi\left(\frac{t - 2^{s_j} n_j}{2^{s_j}}\right)$.



Regression Problems

(Noisy) observations $(t_i, y_i)_{i=1}^m$, try to infer $y \approx f(t)$.

$$f : \mathbb{R} \to \mathbb{R} \text{ (extend to any dimension)}$$

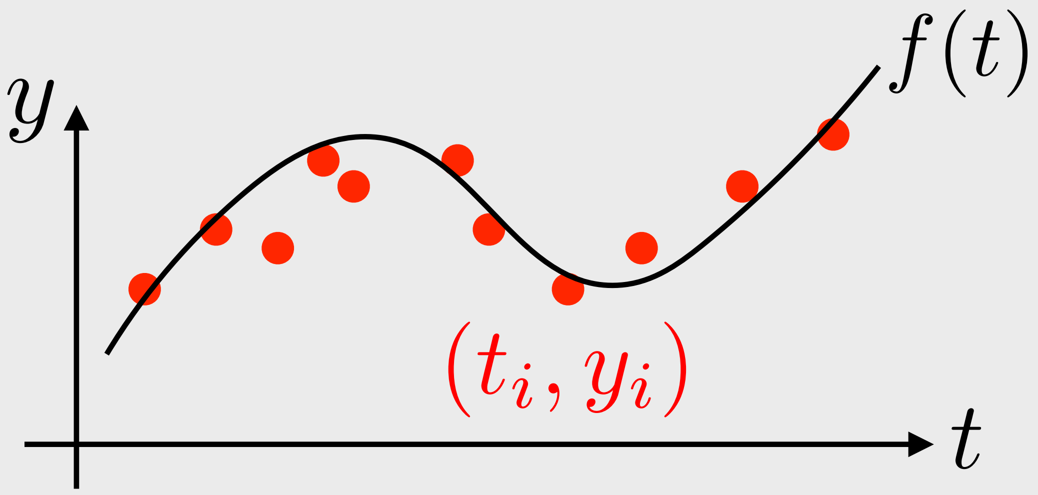
Expansion in a dictionary $(\varphi_j)_{j=1}^n$*:*

$$f(t) \overset{\text{def.}}{=} \sum_{j=1}^n x_j \varphi_j(t)$$

Polynomials: $\varphi_j(t) = t^{k_j}$.

Fourier: $\varphi_j(t) = e^{i\omega_j t}$.

Wavelets: $\varphi_j(t) = \psi\left(\frac{t - 2^{s_j} n_j}{2^{s_j}}\right)$.



Forward model:

$$y_i = f(t_i) + w_i$$

$$y = A\, x + w \in \mathbb{R}^m$$

Observations (y), Dictionary (A), Coefficients (x), Residual (w)

$$A \in \mathbb{R}^{m \times n} : \mathbb{R}^n \to \mathbb{R}^m$$

$$A_{i,j} \overset{\text{def.}}{=} \varphi_j(t_i)$$

Inverse Problems

Forward model: $y = A\,x + w \in \mathbb{R}^m$

Observations | Operator | (Unknown) Input | Noise

$A \in \mathbb{R}^{m\times n} : \mathbb{R}^n \to \mathbb{R}^m$

Denoising: $A = \mathrm{Id}_n$, $m = n$

Inverse Problems

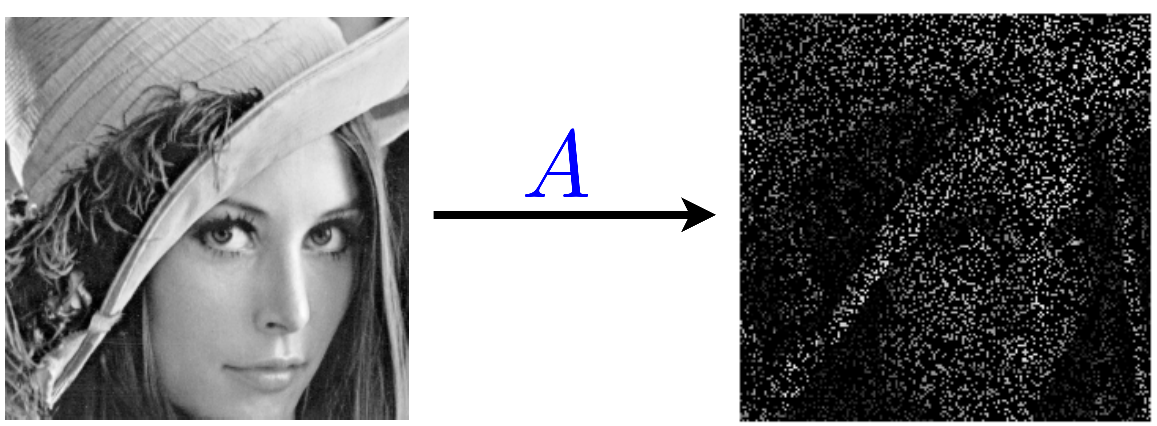
Forward model: $y = A\,x + w \in \mathbb{R}^m$

Observations (y), Operator (A), (Unknown) Input (x), Noise (w)

$A \in \mathbb{R}^{m \times n} : \mathbb{R}^n \to \mathbb{R}^m$

Denoising: $A = \mathrm{Id}_n$, $m = n$

Inpainting: set Ω of available pixels, $m = |\Omega|$, $Ax = (x_i)_{i \in \Omega}$



Inverse Problems

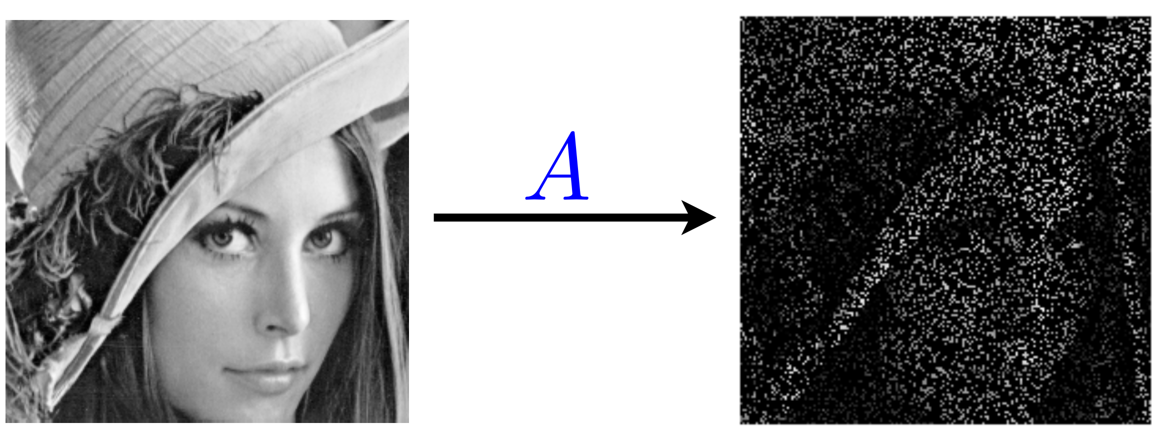
Forward model: $y = A\,x + w \in \mathbb{R}^m$

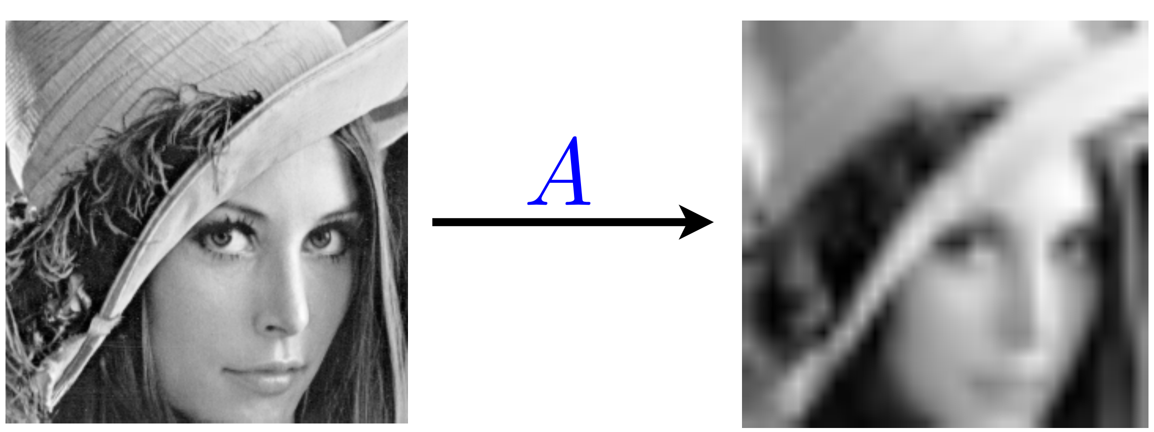
- y: Observations
- A: Operator, $A \in \mathbb{R}^{m \times n} : \mathbb{R}^n \to \mathbb{R}^m$
- x: (Unknown) Input
- w: Noise

Denoising: $A = \mathrm{Id}_n$, $m = n$

Inpainting: set Ω of available pixels, $m = |\Omega|$, $Ax = (x_i)_{i \in \Omega}$

Super-resolution: $Ax = (x \star k) \downarrow_\tau$, $m = n/\tau$.





Inverse Problems

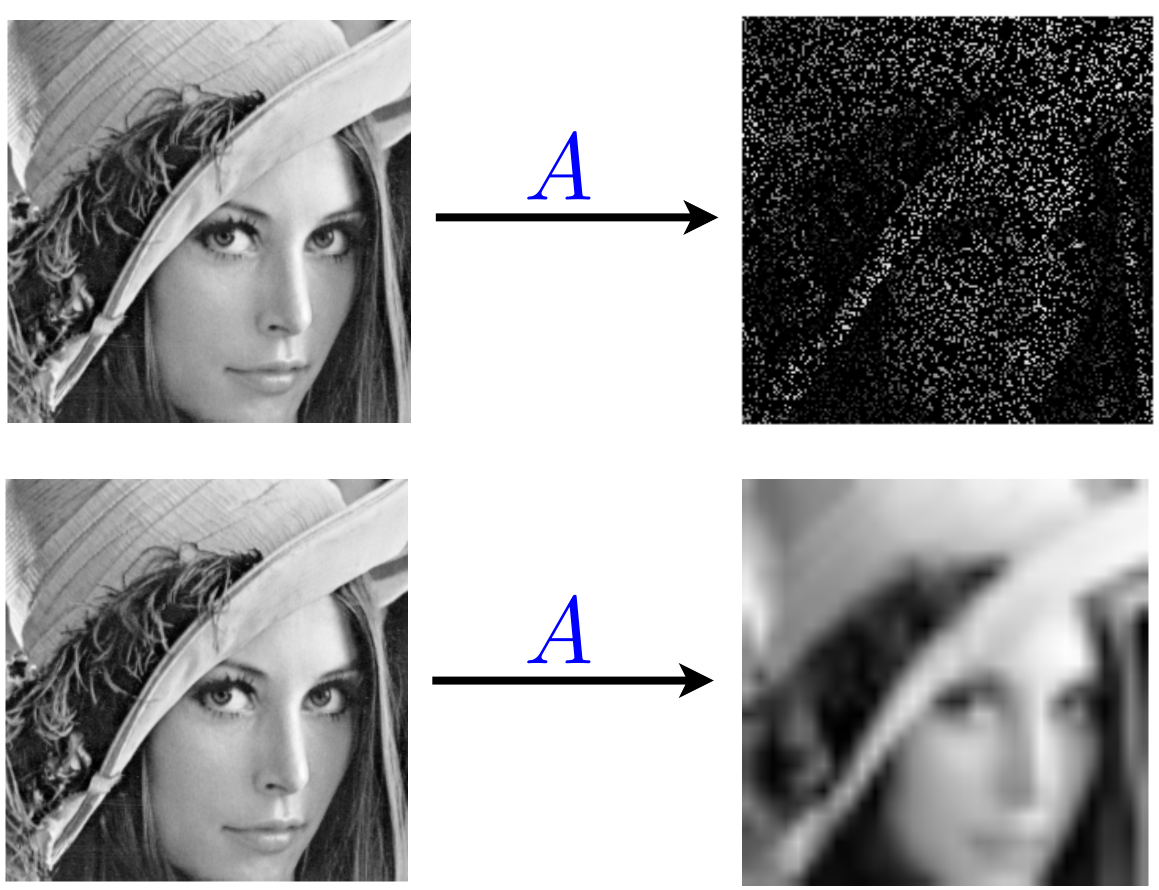
Forward model: $y = A\,x + w \in \mathbb{R}^m$

- y: Observations
- A: Operator, $A \in \mathbb{R}^{m\times n} : \mathbb{R}^n \to \mathbb{R}^m$
- x: (Unknown) Input
- w: Noise

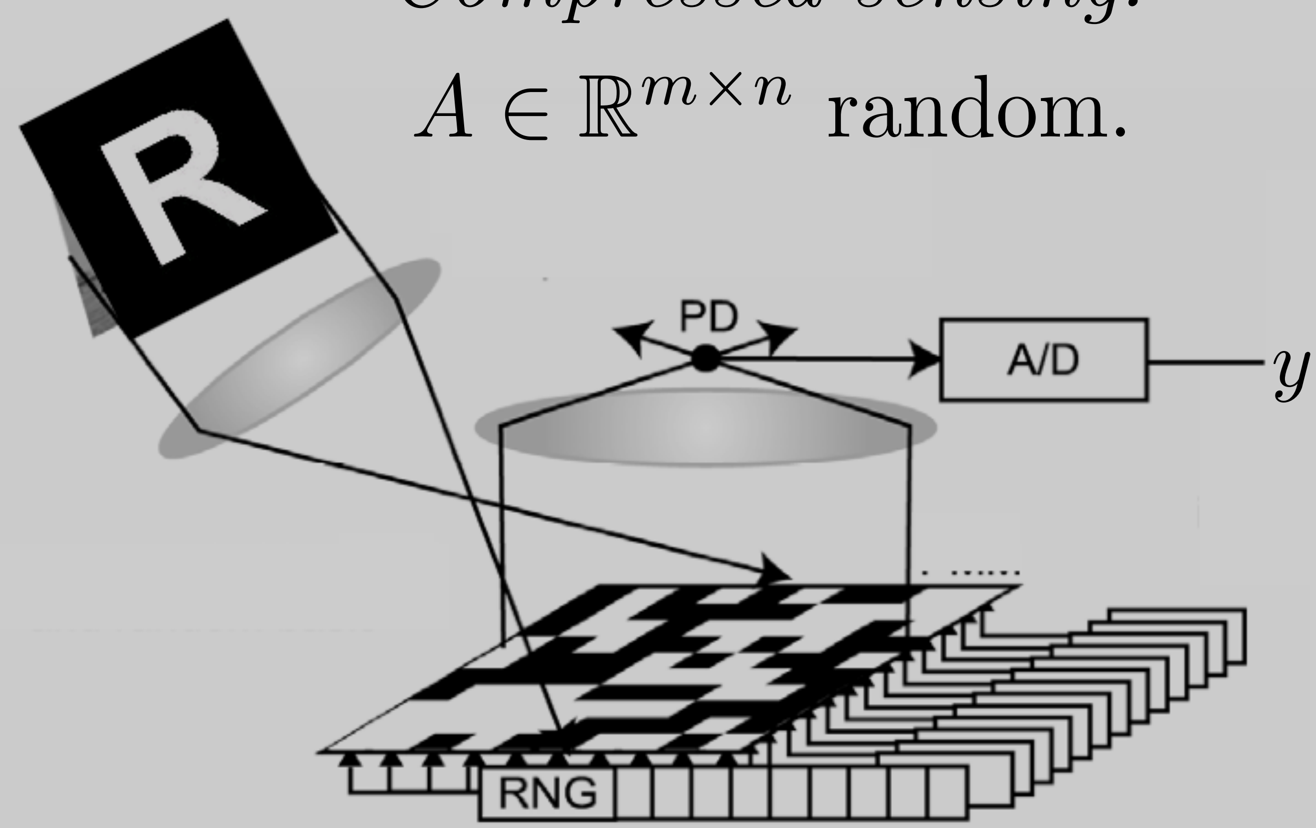
Denoising: $A = \mathrm{Id}_n$, $m = n$

Inpainting: set Ω of available pixels, $m = |\Omega|$, $Ax = (x_i)_{i\in\Omega}$

Super-resolution: $Ax = (x \star k)\downarrow_\tau$, $m = n/\tau$.



Compressed sensing:
$A \in \mathbb{R}^{m\times n}$ random.



Inverse Problem in Medical Imaging

Tomography projection: $Ax = (p_{\theta_k})_{k=1}^K$

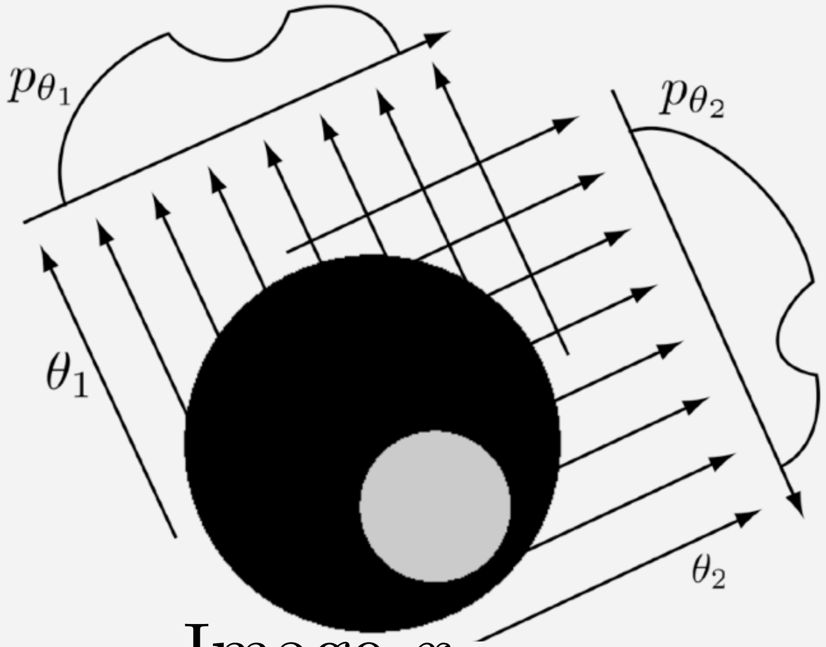
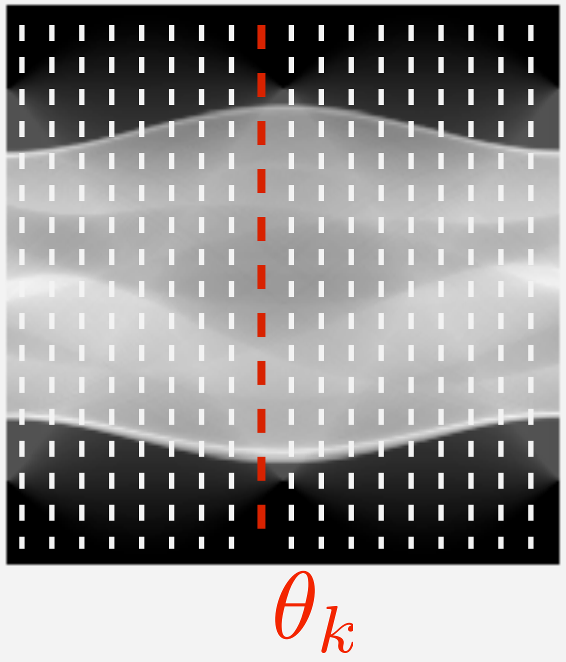
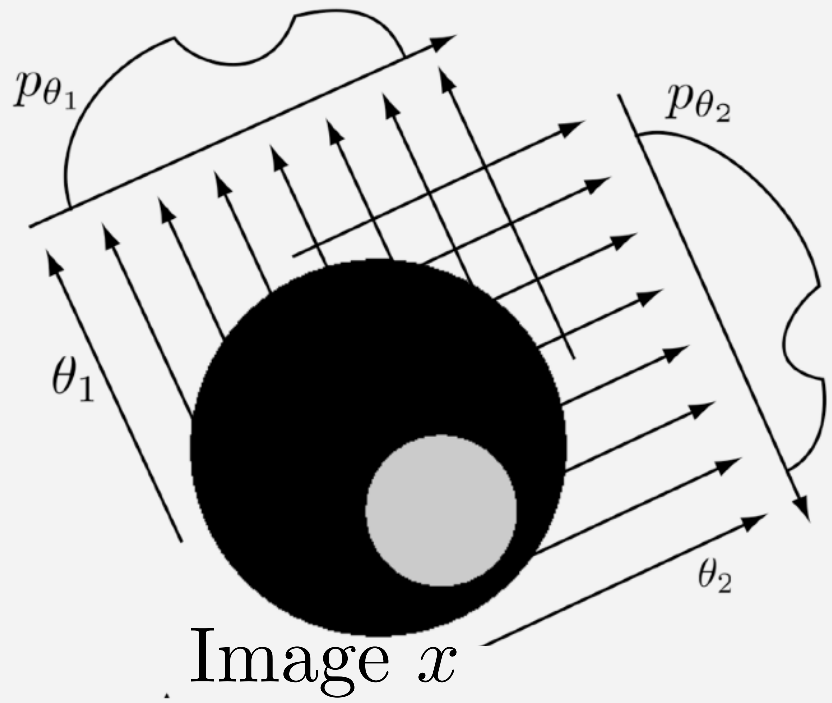


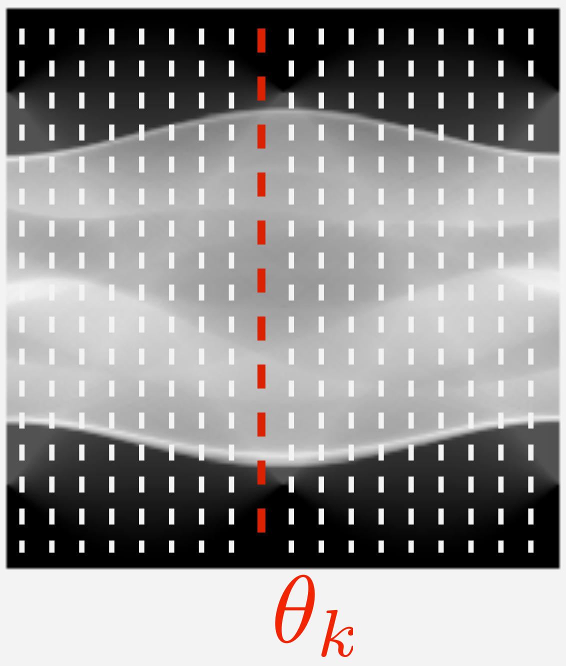
Image x



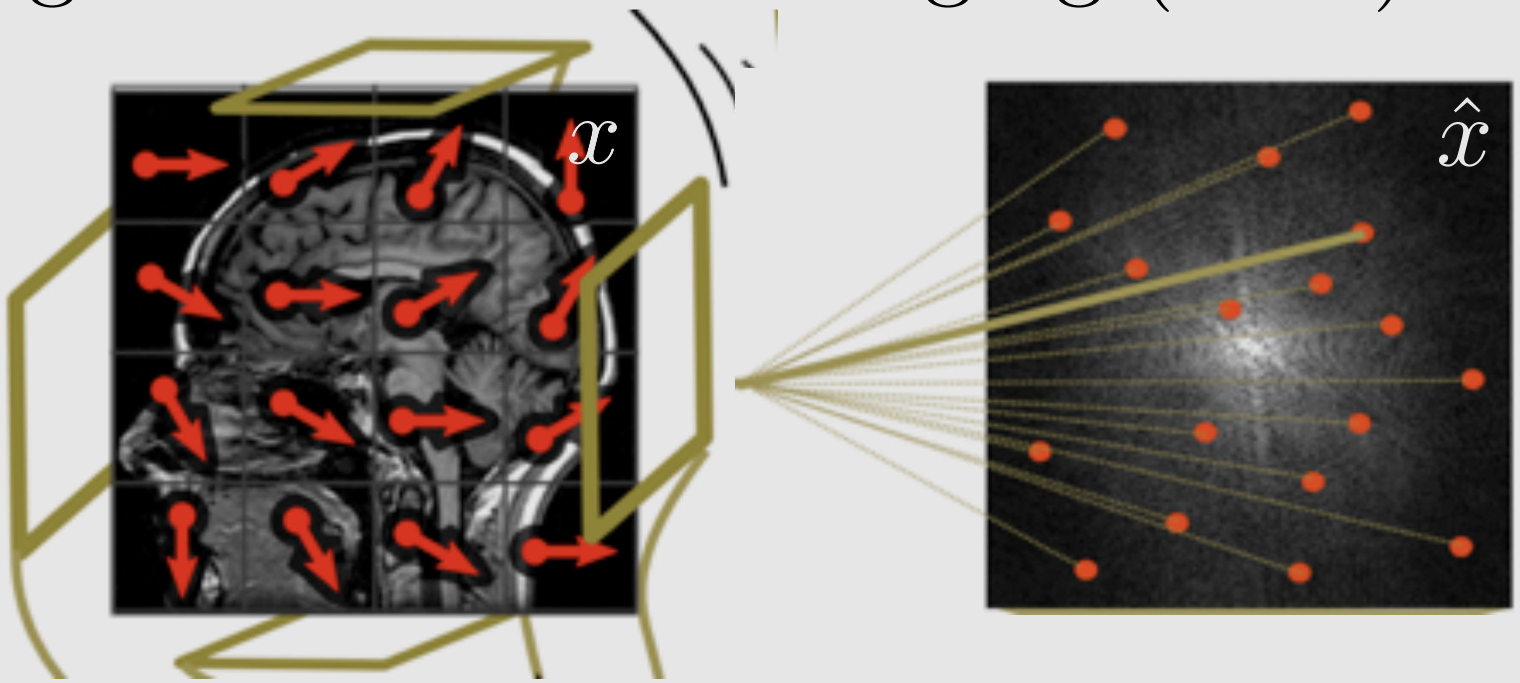
Inverse Problem in Medical Imaging

Tomography projection: $Ax = (p_{\theta_k})_{k=1}^K$



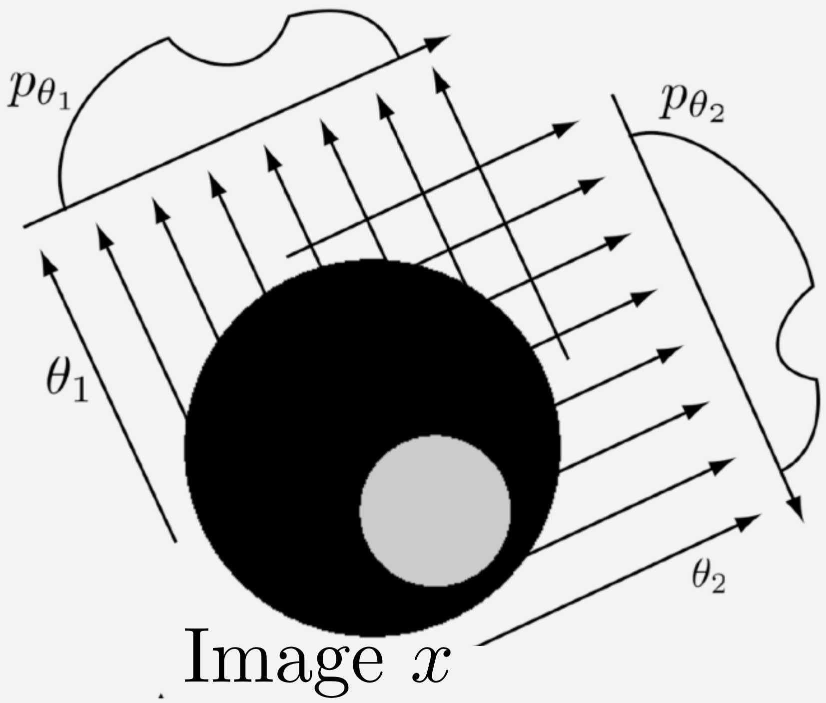


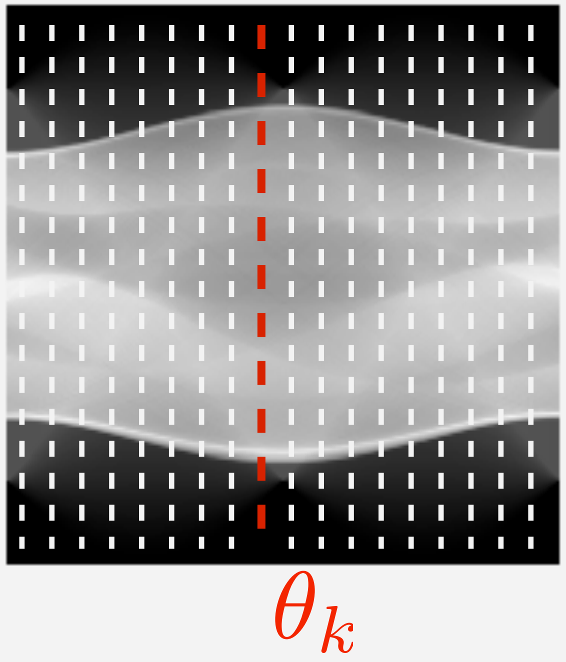
Magnetic resonance imaging (MRI): $Ax = (\hat{x}(\omega))_{\omega \in \Omega}$



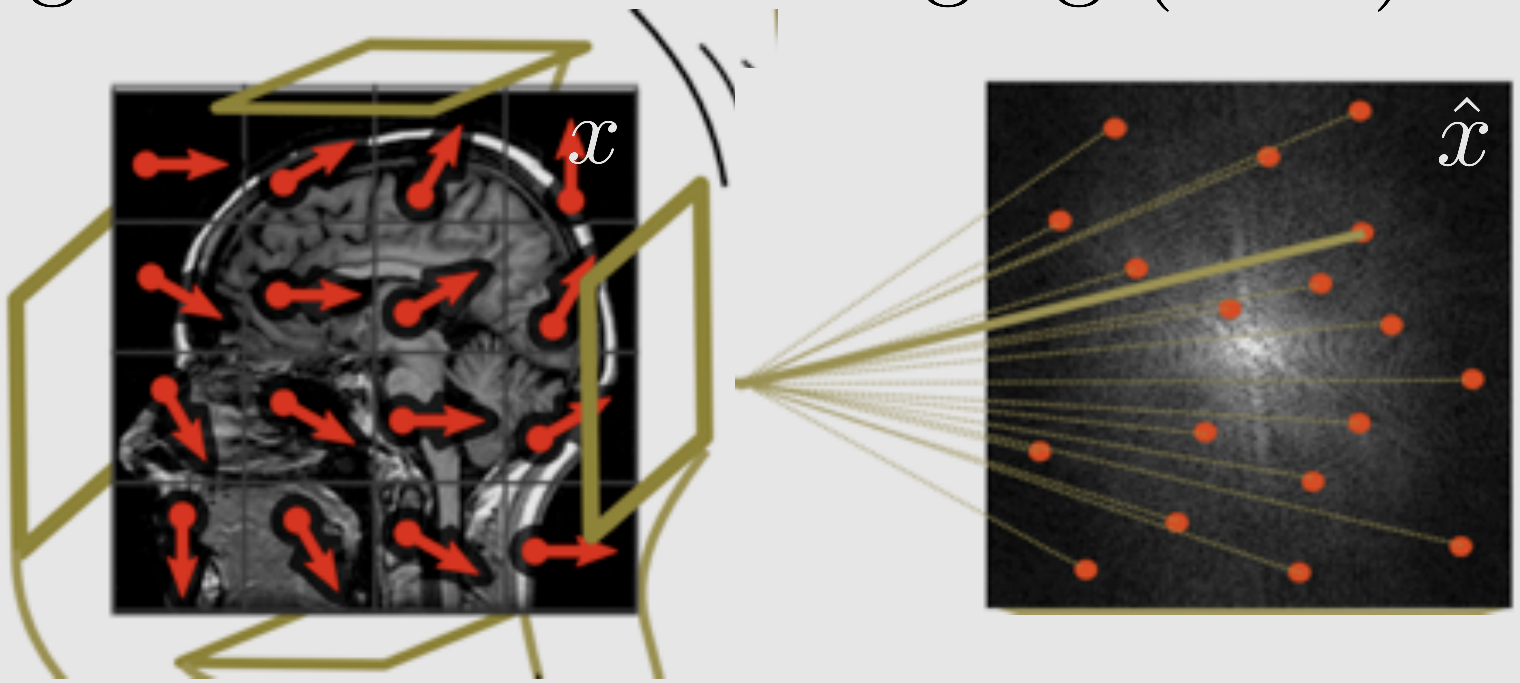
Inverse Problem in Medical Imaging

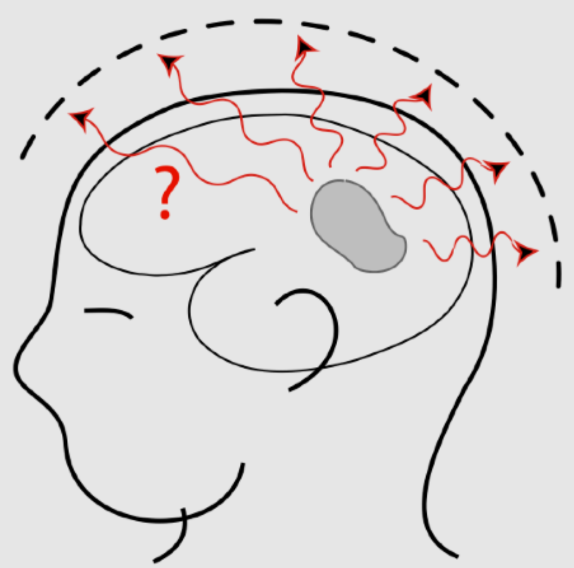
Tomography projection: $Ax = (p_{\theta_k})_{k=1}^K$





Magnetic resonance imaging (MRI): $Ax = (\hat{x}(\omega))_{\omega \in \Omega}$



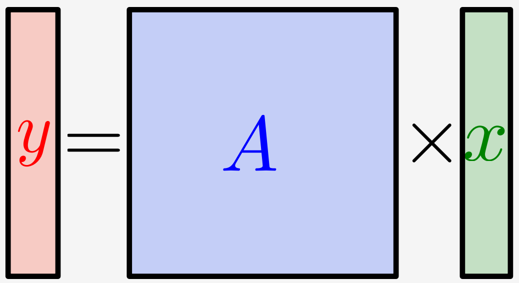


Other examples: MEG, EEG, ...

Least Squares

Solving $y \underset{\approx}{=} A\,x \in \mathbb{R}^m$ $\quad A \in \mathbb{R}^{m \times n}$

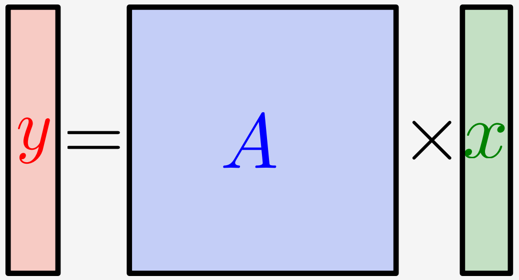
Determined $(m = n)$: $\quad x = A^{-1} y$



Least Squares

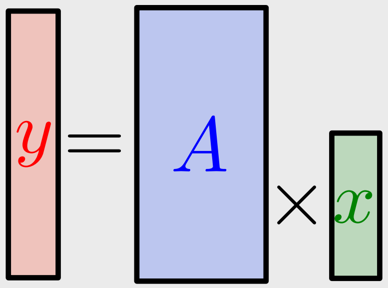
Solving $y \underset{\approx}{=} A\,x \in \mathbb{R}^m \qquad A \in \mathbb{R}^{m\times n}$

Determined $(m = n)$: $\quad x = A^{-1}y$



Over-determined $(m > n)$: $\quad \min_x \|Ax - y\|^2$

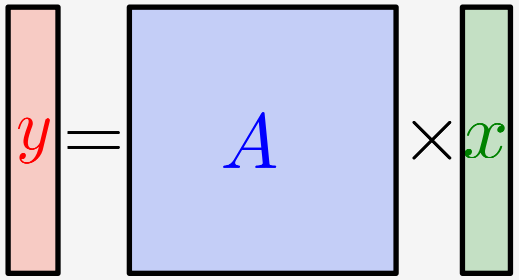
$$x = (A^\top A)^{-1} A^\top y \overset{\text{def.}}{=} A^+ y$$



Least Squares

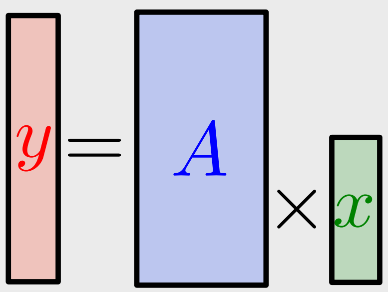
Solving $y \underset{\approx}{=} A\,x \in \mathbb{R}^m$ $\quad A \in \mathbb{R}^{m \times n}$

Determined $(m = n)$: $\quad x = A^{-1} y$



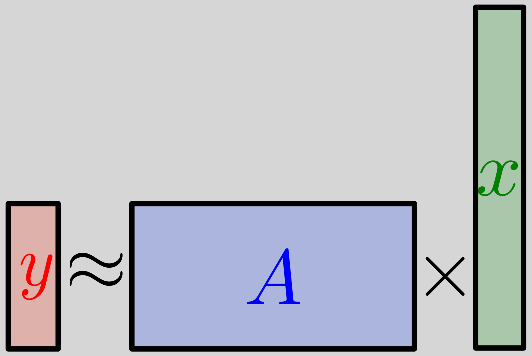
Over-determined $(m > n)$: $\quad \min_x \|Ax - y\|^2$

$$x = (A^\top A)^{-1} A^\top y \overset{\text{def.}}{=} A^+ y$$



Under-determined $(m < n)$: $\quad \min_x \{\|x\| \; ; \; Ax = y\}$

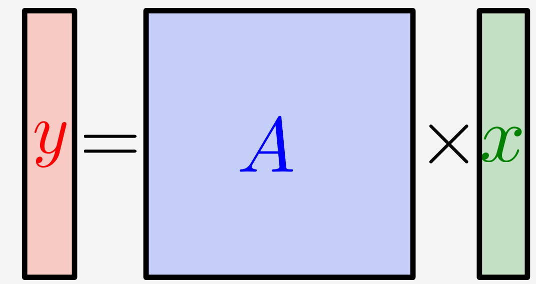
$$x = A^\top (A A^\top)^{-1} y \overset{\text{def.}}{=} A^+ y$$



Least Squares

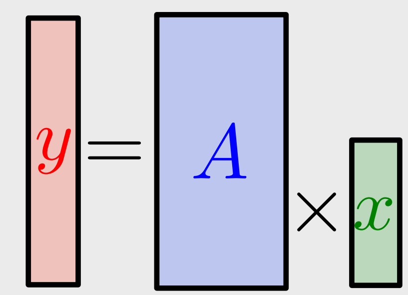
Solving $y \underset{\approx}{=} A\,x \in \mathbb{R}^m \qquad A \in \mathbb{R}^{m\times n}$

Determined $(m = n)$: $\qquad x = A^{-1} y$



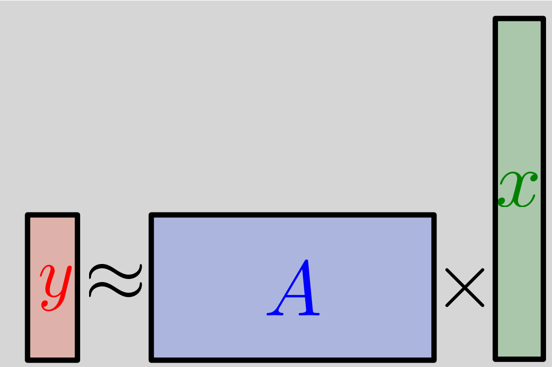
Over-determined $(m > n)$: $\qquad \min_x \|Ax - y\|^2$

$$x = (A^\top A)^{-1} A^\top y \overset{\text{def.}}{=} A^+ y$$



Under-determined $(m < n)$: $\qquad \min_x \; \{\|x\| \;;\; Ax = y\}$

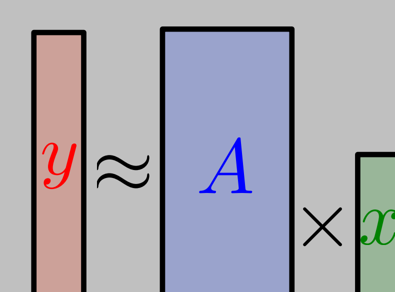
$$x = A^\top (A A^\top)^{-1} y \overset{\text{def.}}{=} A^+ y$$



A ill-posed and/or noise: $\qquad \min_x \|Ax - y\|^2 + \lambda \|x\|^2$

$$x = (A^\top A + \lambda \mathrm{Id}_n)^{-1} A^\top y \xrightarrow{\lambda \to 0} A^+ y$$

$$= A^\top (A A^\top + \lambda \mathrm{Id}_m)^{-1} y \quad \text{(Woodbury identity)}$$



Linear Solver

$$x = \underbrace{(A^\top A + \lambda \mathrm{Id}_n)^{-1} A^\top y}_{\substack{\text{If } n < m \\ \text{(over-determined)}}} = \underbrace{A^\top (A A^\top + \lambda \mathrm{Id}_m)^{-1} y}_{\substack{\text{If } m < n \\ \text{(under-determined)}}}$$

Need to solve: $\underbrace{(A^\top A + \lambda \mathrm{Id}_n)}_{S} x = A^\top y$

S symmetric, positive

Linear Solver

$$x = \underbrace{(A^\top A + \lambda \mathrm{Id}_n)^{-1} A^\top y}_{\substack{\text{If } n < m \\ \text{(over-determined)}}} = \underbrace{A^\top (A A^\top + \lambda \mathrm{Id}_m)^{-1} y}_{\substack{\text{If } m < n \\ \text{(under-determined)}}}$$

Need to solve: $\underbrace{(A^\top A + \lambda \mathrm{Id}_n)}_{S} x = A^\top y$

S symmetric, positive

Direct methods:

– Cholesky factorization $S = LL^\top$

– QR factorization $A = QR \rightarrow S = R^\top R + \lambda \mathrm{Id}_n$

– Singular Value Decomposition $A = U \operatorname{diag}(\Lambda) V^\top$

slower (2x)
more stable

slower
more general

Linear Solver

$$x = \underbrace{(A^\top A + \lambda \mathrm{Id}_n)^{-1} A^\top y}_{\text{If } n < m \text{ (over-determined)}} = \underbrace{A^\top (A A^\top + \lambda \mathrm{Id}_m)^{-1} y}_{\text{If } m < n \text{ (under-determined)}}$$

Need to solve: $\underbrace{(A^\top A + \lambda \mathrm{Id}_n)}_{S} x = A^\top y$

S symmetric, positive

Direct methods:

- Cholesky factorization $S = LL^\top$
- QR factorization $A = QR \rightarrow S = R^\top R + \lambda \mathrm{Id}_n$
- Singular Value Decomposition $A = U \,\mathrm{diag}(\Lambda) V^\top$

Cholesky → QR: slower (2x), more stable

QR → SVD: slower, more general

Iterative methods:

- Conjugate gradient:
  - as an exact method: even slower ...
  - use it for sparse A

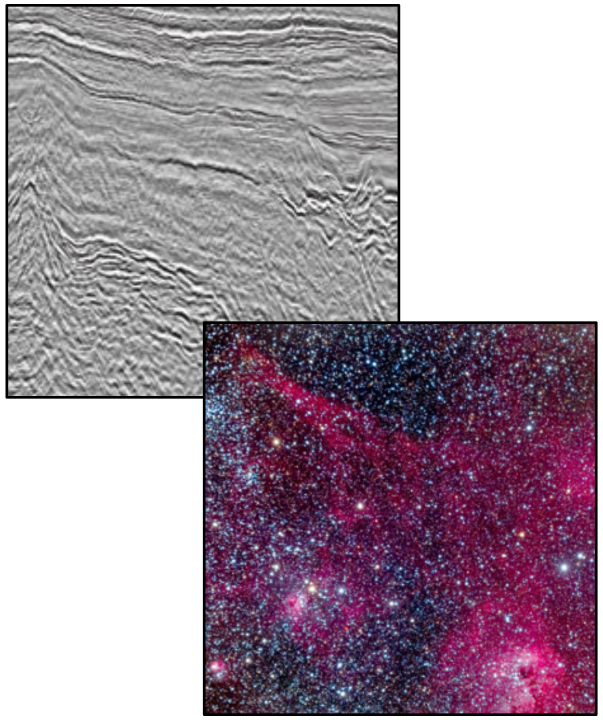
Convergence rate:

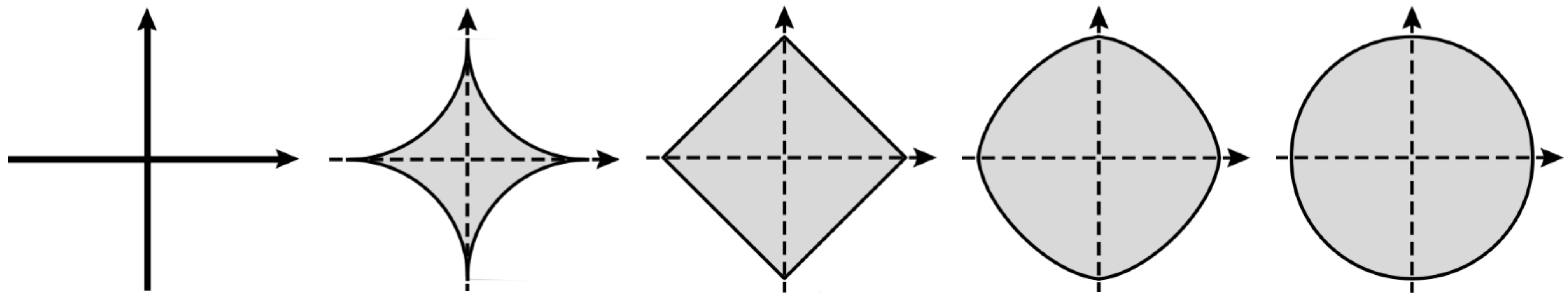
$$\|x_k - x^\star\|_S \leqslant 2 \left(\frac{\sqrt{\kappa(S)} - 1}{\sqrt{\kappa(S)} + 1} \right)^k \|x_0 - x^\star\|_S$$

$\rightarrow$ preconditioning is important (for small λ).

What's Next

Jalal Fadili: sparse regularization, ℓ^0, ℓ^1.

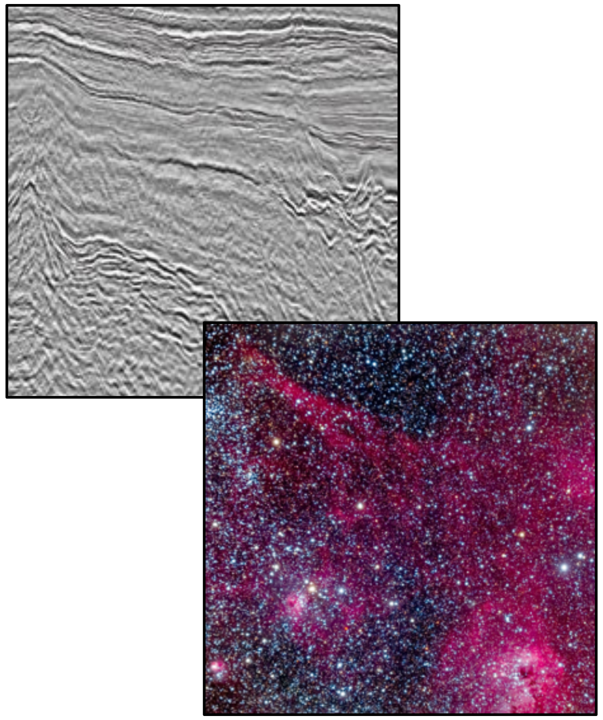


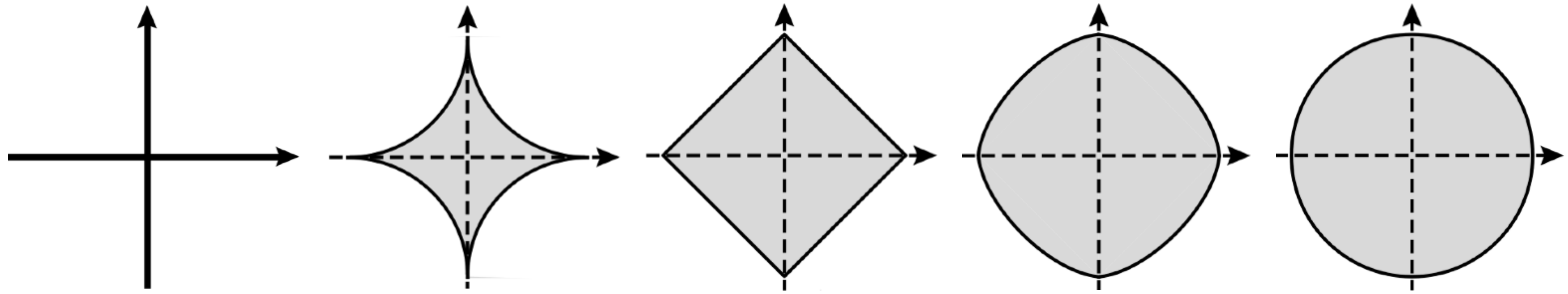


Guillaume Lecué: compressed sensing, random matrices.

What’s Next

Jalal Fadili: sparse regularization, ℓ^0, ℓ^1.





Guillaume Lecué: compressed sensing, random matrices.