

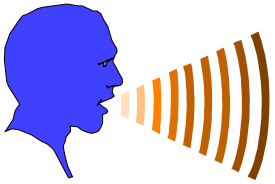



Image and Video Processing

Sparsity-Based Image Recovery

Yao Wang
Tandon School of Engineering, New York University

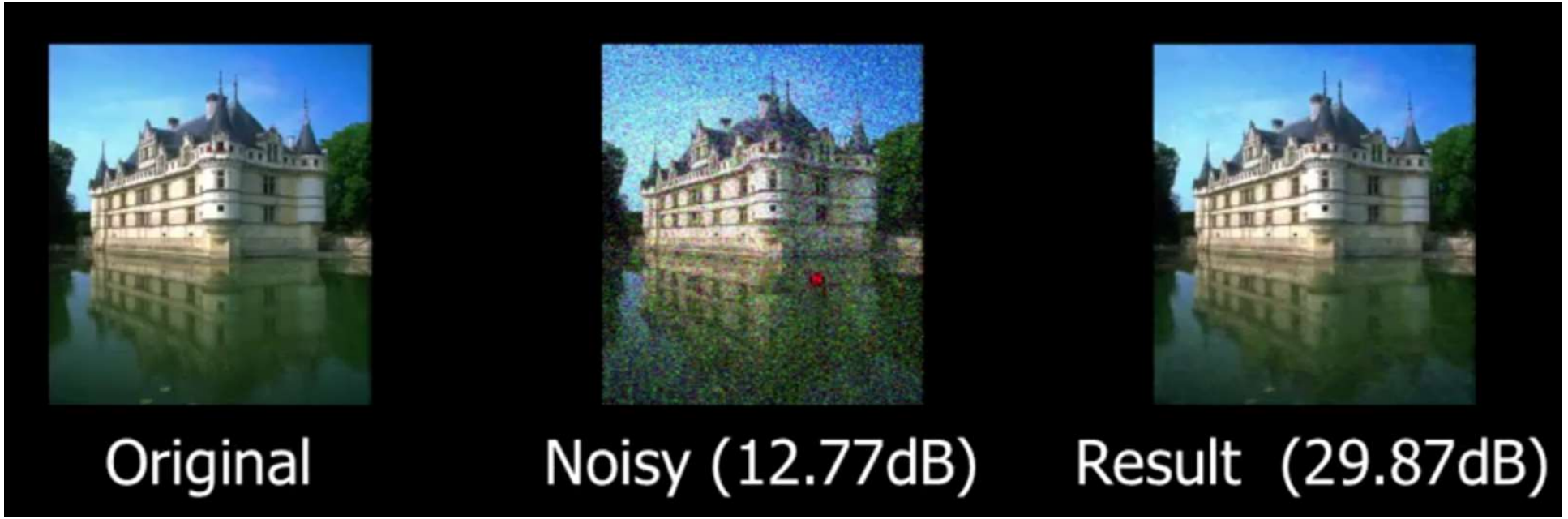
Lecture Outline

- Overview of image recovery problems
- Basics of Optimization (review)
- Least squares formulation and solution (review)
- Lp norm, convex relaxation
- Regularization
  - Sparsity in a transformed domain (Lasso problem)
  - Smoothness (least squares solution)
  - Total variation
- Denoising using soft thresholding in transform domain
- ISTA for solving Lasso
- ADMM: general method
- ADMM for specific problems: Lasso, TV
- Dictionary learning
- Application for images

Image Recovery Problems

- Reorder an image (or image block) into a vector x
- Denoising: remove additive noise on pixel values
  - y= x+n
- Deblurring (bluring kernel h, with matrix representation H)
  - y = h*x+n = H x +n
- Completion/in-painting: filling in missing pixels (covered by letters, logos, etc)
  - y= M x+n (M: mask indicating which pixels are known/missing)
- Compressive sensing
  - y = H x +n, H is the imaging operator (MRI, CT, etc.), y has smaller dimension than x
- General Problem:
  - Recover x from y, assuming y= G x+n
- Note: For large images, we do not use vector/matrix operation, rather "operator" (e.g. convolution or transform) (matrix-free)

Image Denoising Using Sparse Modeling



From Shapiro's Coursera Course on Image Processing, Lecture on Sparse Modeling and Compressed Sensing.

Image Inpainting Using Sparse Modeling

From Shapiro's Coursera Course on Image Processing, Lecture on Sparse Modeling and Compressed Sensing.

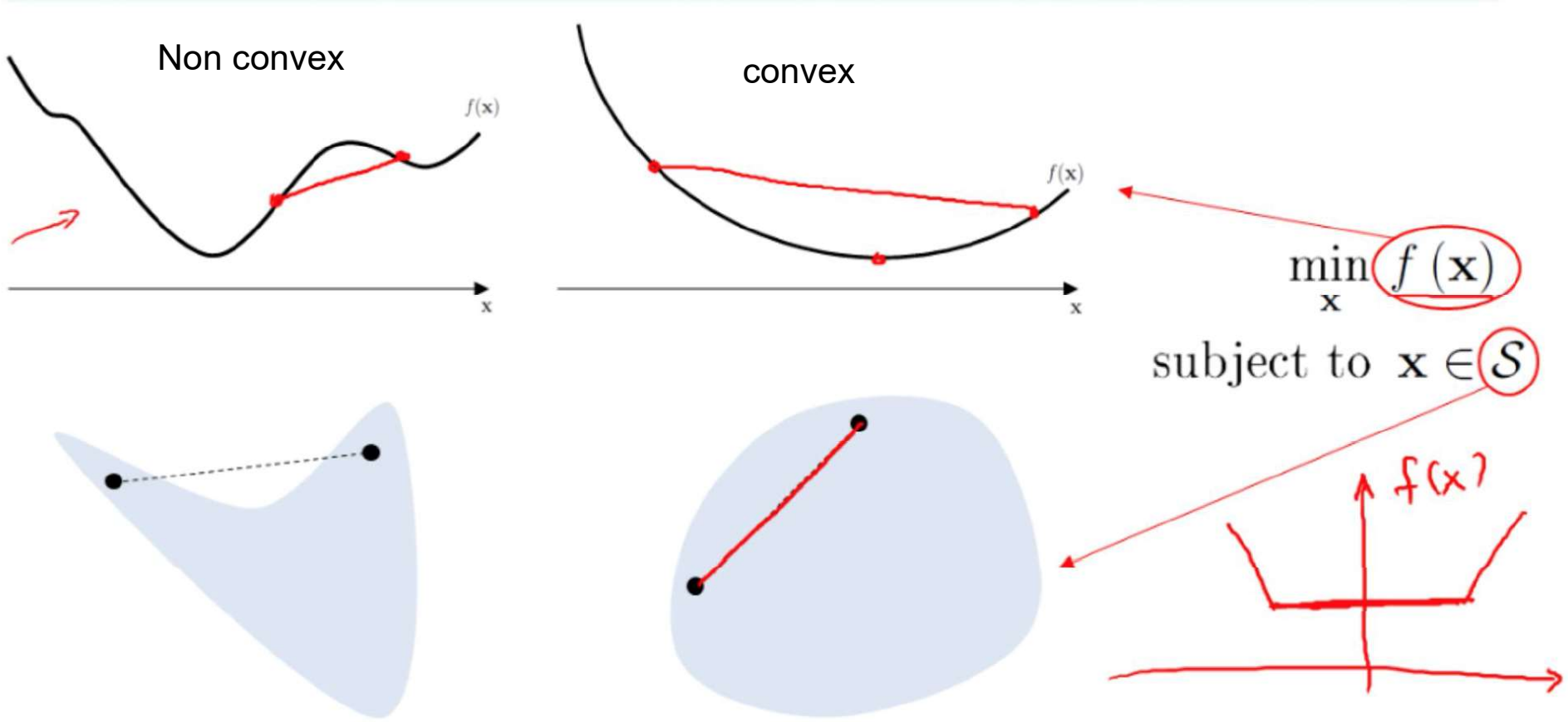
Compressive Sensing for Medical Imaging

- Sensing matrix applied to an entire image
- MRI:
  - Measure 2D DFT of the image (k-space)
  - Each row of the sensing matrix is a DFT Basis
  - Using fewer measurements (fewer DFT coefficients than number of pixels) to reduce imaging time
- CT:
  - Measure line integrals along different directions of an image (Radon transform)
  - Each row of the sensing matrix corresponds to the sum along a particular line connecting x-ray transmitter and receiver
  - Using fewer measurements to reduce radiation exposure
- How to recover an image from incomplete measurements?

Review: Basics of Optimization

- Unconstrained optimization
- Constrained optimization: can change to unconstrained with Lagrangian method
- Local vs. Global minimum
- Convex optimization
  - Loss function is convex, constraint set is convex
- For convex problem, every local minimum is a global minimum.
  - If the objective function J(x) is continuously differentiable, can obtain solution by setting derivative of J(x) to 0
- Convex relaxation:
  - Approximate a non-convex problem by a convex problem so that it is easy to solve.
  - Investigate the conditions under which the two give equal or similar solutions

Convex Function and Convex Sets

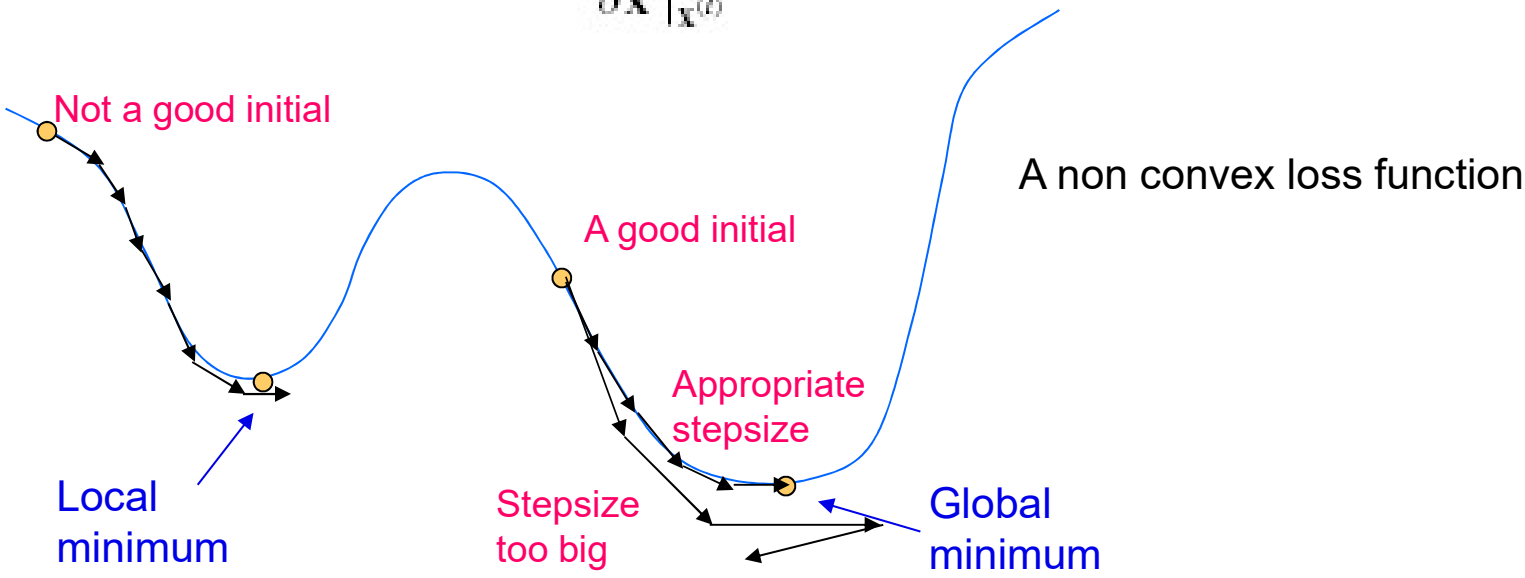


From Katsagelos's Coursera Course on Image Processing, Lecture on Sparsity.

Gradient Descent Method

- Iteratively update the current estimate in the direction opposite the gradient direction.

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \alpha \left. \frac{\partial J}{\partial \mathbf{x}} \right|_{\mathbf{x}^{(l)}}$$



- The solution depends on the initial condition. Reaches the local minimum closest to the initial condition if the stepsize is chosen properly.
- Yield global optimal if J is convex, regardless initial solution

Least Squares Methods

- Model: $y_{MX1} = H_{MxN}x_{Nx1} + n_{Nx1}$
- M=N: H is square and invertible (deblurring)
  - $x = H^{-1} y = x + H^{-1} n$
  - If H is ill conditioned (near singular), noise will blow up
- M>N: H is tall (more measurements than unknowns)
  - Least squares: minimize $J(x) = ||Hx-y||^2$
  - Set derivative of J(x) = 0 -> $x = (H^T H)^{-1} H^T y$
    - Can also amplify noise
- M<N: H is fat (compressive sensing, in-painting)
  - Underdetermined problem, infinite solutions
  - Minimum norm solution: minimize $||x||^2$, subject to y=Hx
    - $X = H^T(H H^T)^{-1}y$, sensitive to noise!

http://eeweb.poly.edu/iselesni/lecture_notes/least_squares/index.html
http://eeweb.poly.edu/iselesni/lecture_notes/least_squares/least_squares_SP.pdf

Regularization: General Ideas

- Given measurements (data): y = Hx+n
  - H represent blurring or imaging operator
  - Typically assuming noise n is Gaussian
- Prior knowledge that can be formulated as minimizing R(x)
  - Necessary for underdetermined problem
- Minimizing
  - $J(x) = ||Hx - y||^2 + \lambda R(x)$

Regularization parameter

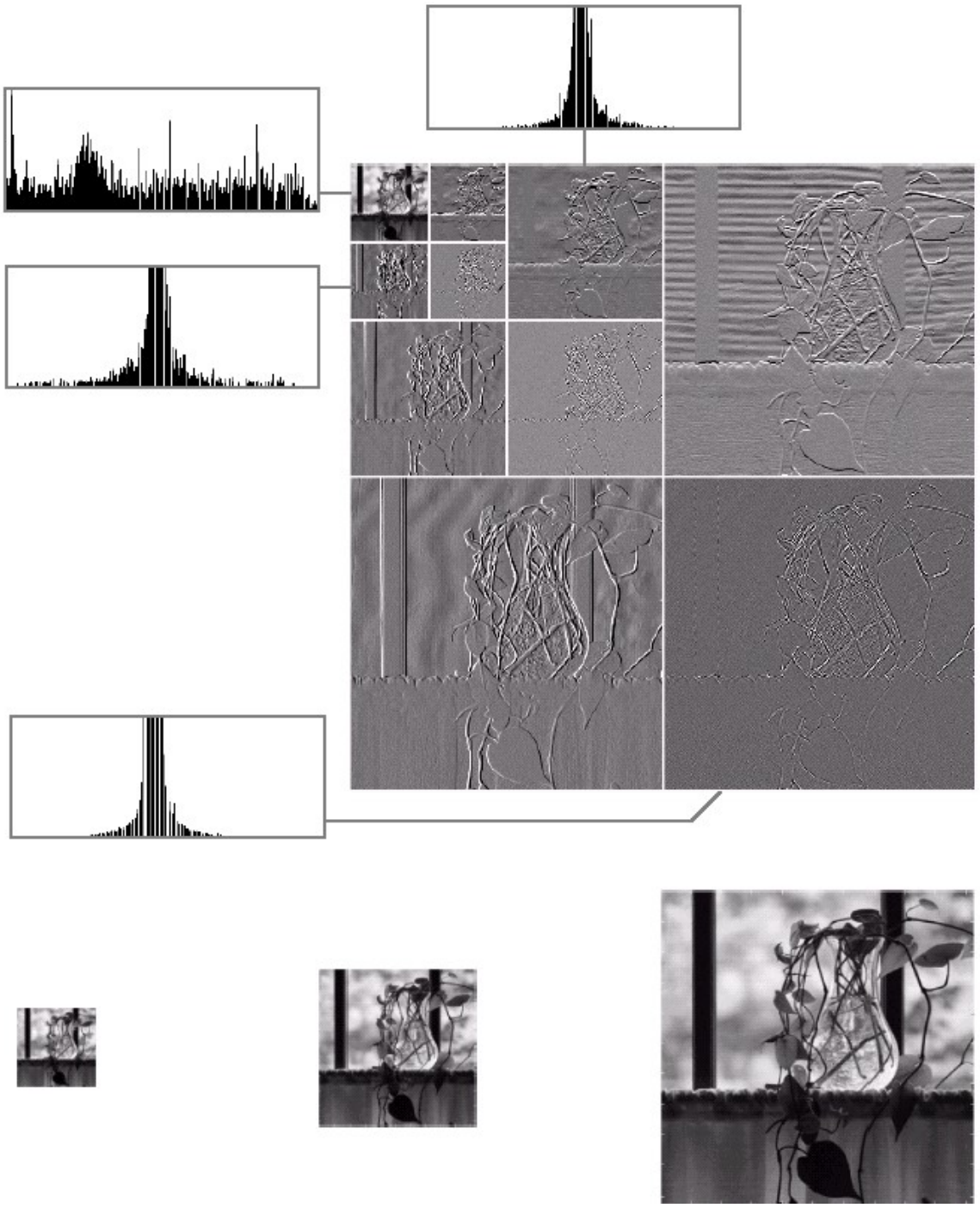
Data consistency term

Regularization term

- Desirable property for R(x):
  - Convex, so that J(x) has a unique minimum

Sparsity-Based Regularization

- Represent the original image in a transform/dictionary
  - Dictionary is a redundant transform with more bases (known as atoms) than the number of signal samples
  - $x = T x_t$ (Columns in T are basis vectors or atoms)
- The transform/dictionary T is chosen such that the coefficients are sparse (many zeros)
  - Ex: DCT transform (block wise), Wavelet transform (whole image)
  - Can also specifically design transform/dictionary to enhance sparsity
- Recover the image by minimizing the number of non-zeros (L0)
  - $R(x) = ||x_t||_0$; Non convex
- Relax L0 by L1 norm (Convex relaxation)
  - $R(x) = ||x_t||_1$
- Instead of recovering x directly, recover x_t instead
  - $y = Hx + n = HT x_t + n = G x_t + n$; $G=HT$
  - $J(x_t) = ||G x_t - y||^2 + \lambda || x_t ||_1$ (L2-L1 problem, for Gaussian noise)
  - $J(x_t) = ||G x_t - y||_1 + \lambda || x_t ||_1$ (L1-L1 problem, for Laplacian noise)



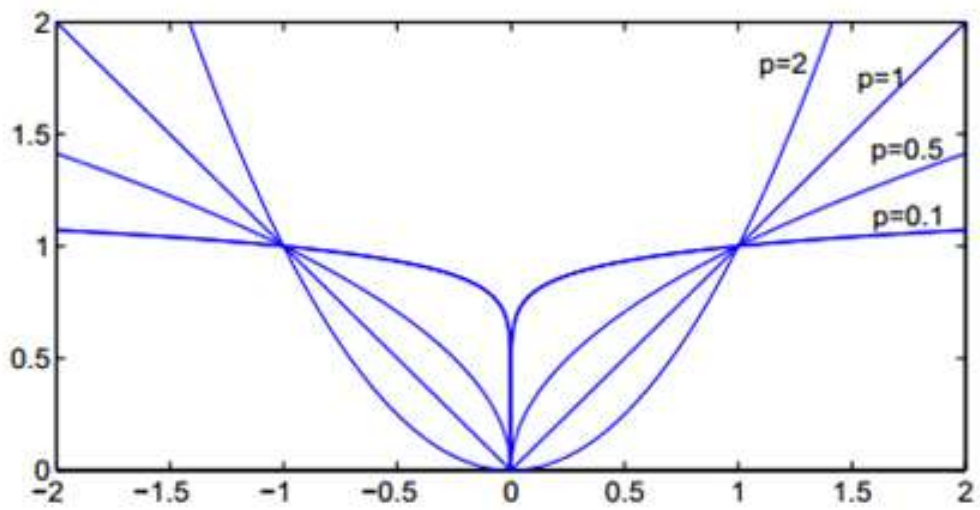
a
b c d

FIGURE 7.8 (a) A discrete wavelet transform using Haar basis functions. Its local histogram variations are also shown; (b)–(d) Several different approximations (64 × 64, 128 × 128, and 256 × 256) that can be obtained from (a).

Wavelet coefficients are sparse!

From [Gonzalez2008]

L0, L1, L2, Lp Norm

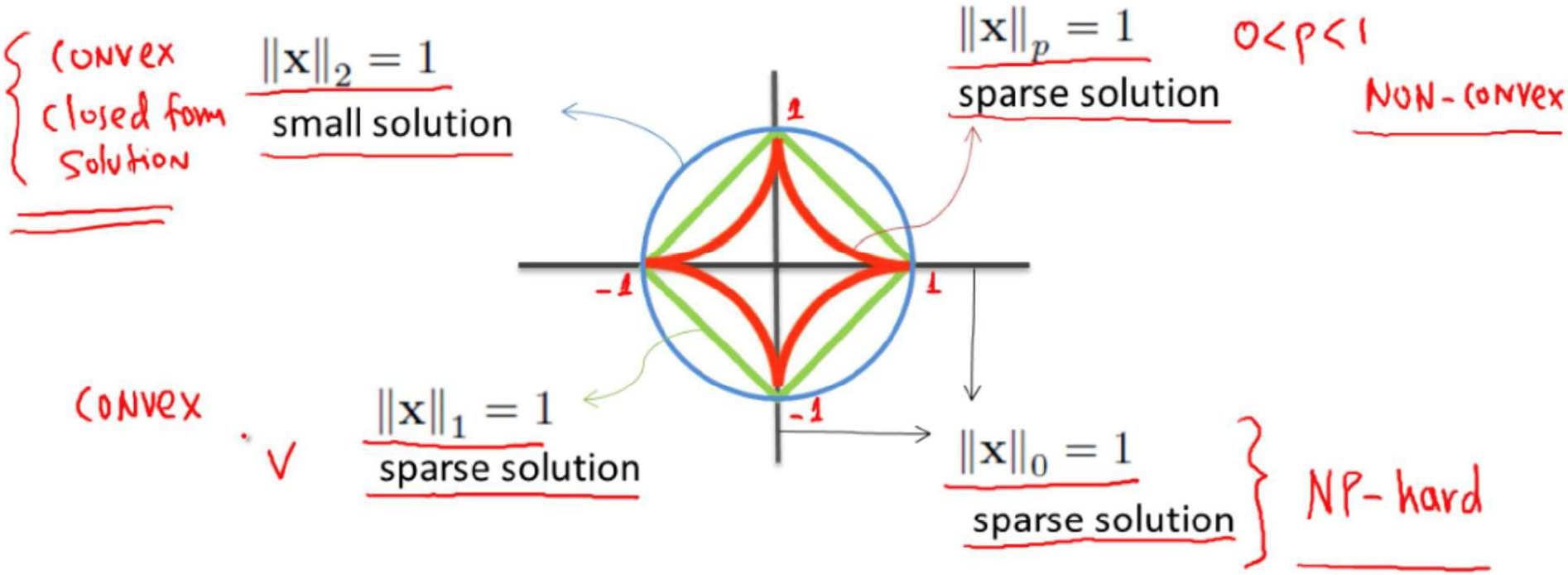


$$\|x\|_p = \left(\sum_{n=1}^{N} |x_n|^p\right)^{1/p}$$

l_2 = Euclidean length of the vector
l_1 = sum of absolute value
l_0 = number of non-zeros
l_∞ = max of x_n

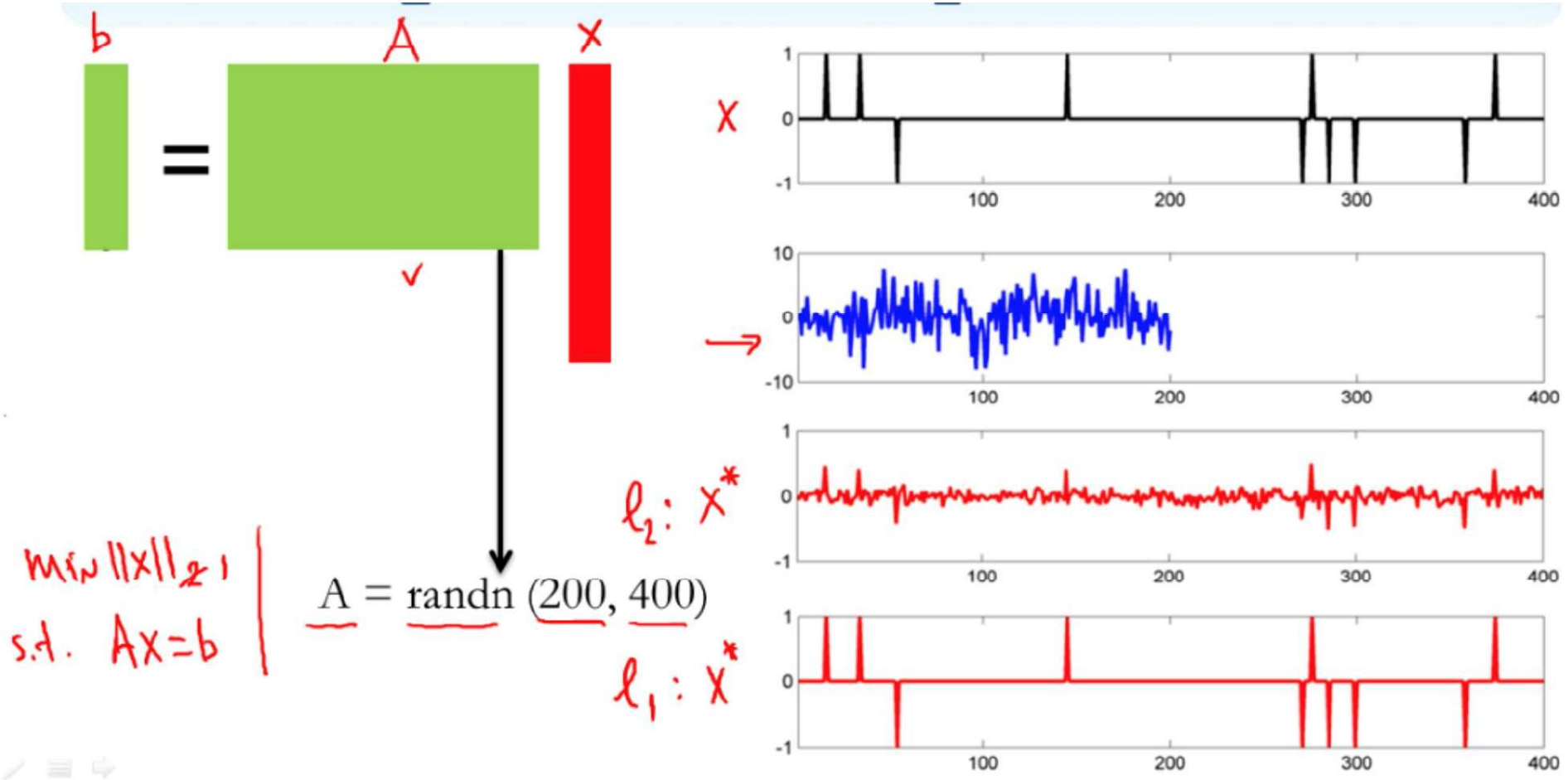
- Which one is convex?
- Minimizing L2 puts more penalty on large values -> many small but non-zero values
- Minimizing L1 puts proportional penalty on small and large values -> many zeros and several large values (sparse)
- L1 is a convex surrogate of L0

Sparsity of Solution by Minimizing Lp Norm



From Katsagelos's Coursera Course on Image Processing, Lecture on Sparsity.

L2 vs. L1 Solution



From Katsagelos's Coursera Course on Image Processing, Lecture on Sparsity.

Denoising Using Orthonormal Transform

- Denoising: H=I (no blurring)
- T is orthonormal: $T^H T = T T^H = I$ (^{H} represents transpose and conjugate)
- Assume y=x+n, $x=T x_t$
- Apply forward transform T^T to y: $T^T y = T^T x + T^T n$
  - $y_t = x_t + n_t$
  - If n is i.i.d (Covariance matrix $C_n = \sigma_n^2 I$) and T is orthonormal, n_t is also i.i.d. with same variance.
- Consider the problem of y= x+n (x,y,n are all transform coefficients)
  - $J(x) = ||y-x||^2 + \lambda ||x||_1$
  - Have closed-form solution!

Soft Thresholding

- What to minimize: $J(\mathbf{x}) = \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1.$
- The variables are uncoupled!

$$J(\mathbf{x}) = (y_1 - x_1)^2 + \lambda|x_1| + (y_2 - x_2)^2 + \lambda|x_2| + \cdots + (y_N - x_N)^2 + \lambda|x_N|$$

- Just need to know how to minimize $f(x) = (y - x)^2 + \lambda|x|.$
  (x and y are scalars here)
- Setting gradient to zero yields:

$$x = \text{soft}(y,\ \lambda/2)$$

$$\text{soft}(x, T) := \begin{cases} x + T & x \le -T \\ 0 & |x| \le T \\ x - T & x \ge T \end{cases}$$

- Derive on the board

http://eeweb.poly.edu/iselesni/lecture_notes/SoftThresholding.pdf

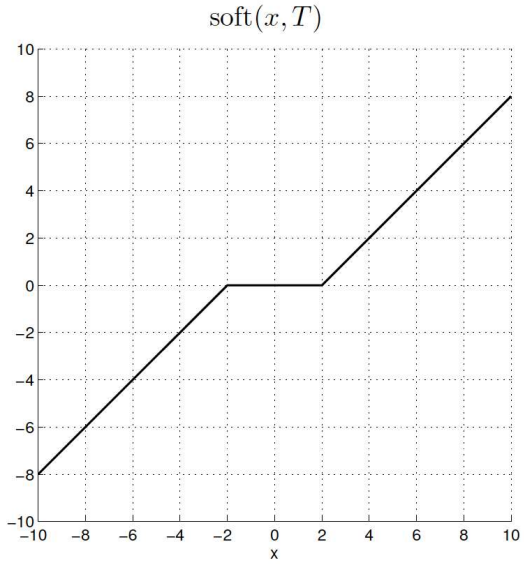


FIGURE 2. The soft-threshold rule (16) with threshold $T = 2$.

How to determine regularization parameter λ?

- Assuming coefficients x=w follows Laplacian distribution with STD σ
- Assuming n follows Gaussian distribution with STD σ_n
- Maximum a posteriori (MAP) estimator
- Notation: y=w+n

$$\hat{w}(y) = \arg\max_w p_{w|y}(w|y) \qquad p_{w|y}(w|y) = \frac{p_{y|w}(y|w)\, p_w(w)}{p_y(y)}$$

$$\hat{w}(y) = \arg\max_w \left[p_{y|w}(y|w) \cdot p_w(w)\right] \qquad p_{y|w}(y|w) = p_n(y-w)$$

$$\hat{w}(y) = \arg\max_w \left[p_n(y-w) \cdot p_w(w)\right]$$

$$\hat{w}(y) = \arg\max_w \left[\log(p_n(y-w)\, p_w(w))\right]$$

$$\hat{w}(y) = \arg\max_w \left[\log(p_n(y-w)) + \log(p_w(w))\right]$$

[ref]: http://eeweb.poly.edu/iselesni/lecture_notes/SoftThresholding.pdf

$$p_n(n) = \frac{1}{\sigma_n\sqrt{2\pi}} \cdot \exp\left(-\frac{n^2}{2\,\sigma_n^2}\right) \qquad p_w(w) = \frac{1}{\sqrt{2}\,\sigma} \exp\left(-\frac{\sqrt{2}}{\sigma}\,|w|\right)$$

$$\log\big(p_n(y-w)\big) = -\frac{(y-w)^2}{2\sigma_n^2} + const;\ \log\big(p_w(w)\big) = -\tfrac{\sqrt{2}}{\sigma}|w| + \text{const.}$$

$$\hat{w}(y) = \arg\max_w \left[\log(p_n(y-w)) + \log(p_w(w))\right]$$

$$\widehat{w}(y) = argmin\left[\tfrac{(y-w)^2}{2\sigma_n^2} + \tfrac{\sqrt{2}}{\sigma}|w|\right] = argmin\left[(y-w)^2 + \tfrac{2\sqrt{2}\sigma_n^2}{\sigma}|w|\right]$$

λ

$$\lambda = \tfrac{2\sqrt{2}\sigma_n^2}{\sigma} \qquad T = \frac{\lambda}{2} = \tfrac{\sqrt{2}\sigma_n^2}{\sigma}$$

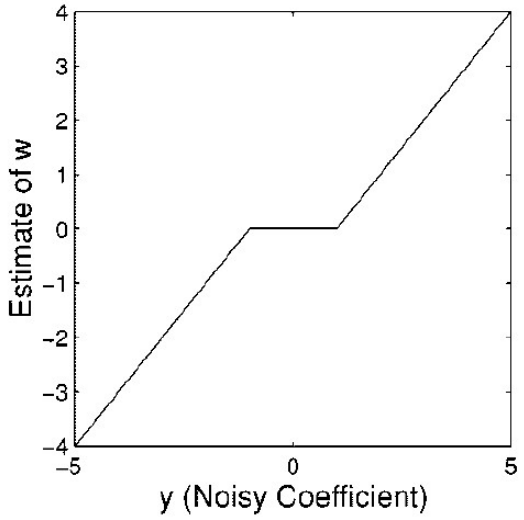
$$\hat{w}(y) = \text{soft}\left(y, \frac{\sqrt{2}\,\sigma_n^2}{\sigma}\right)$$

[ref]:http://eeweb.poly.edu/iselesni/lecture_notes/SoftThresholding.pdf

Wavelet Domain Image Denoising Using Soft Thresholding

- Wavelet transform is an orthonormal transform
- Apply wavelet transform to a noisy image to obtain y
- Modify the coefficients y based on signal and noise statistics
  - If noise is Gaussian $N(0,\sigma_n)$, true signal coeff is Laplacian with STD σ
  - Soft-thresholding (shrinkage function)

$$\hat{w}(y) = \text{soft}\left(y, \frac{\sqrt{2}\sigma_n^2}{\sigma}\right)$$



- Inverse wavelet transform
- Remove noise yet not blurring the edges!
- How to estimate signal and noise statistics?

[ref]:http://eeweb.poly.edu/iselesni/lecture_notes/SoftThresholding.pdf

Subband Adaptive Thresholds

- Wavelet signal variances differ among subbands
- Should estimate the variance for each subband
- But the observed subbands are noisy!
- For each subband:

$$\mathrm{VAR}[y] = \mathrm{VAR}[w] + \mathrm{VAR}[n]$$

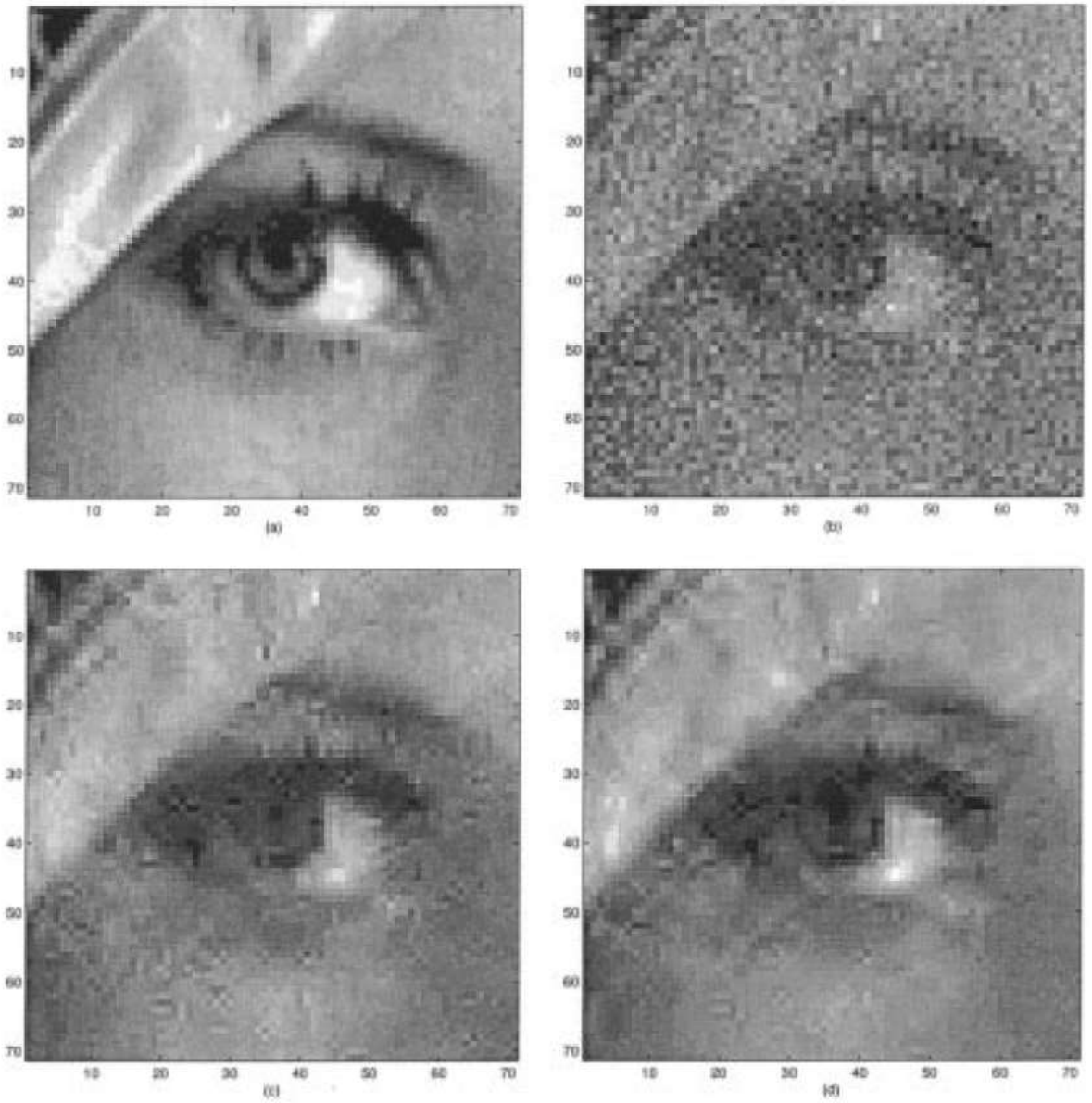
$$\mathrm{VAR}[w] = \mathrm{VAR}[y] - \mathrm{VAR}[n]$$

$$\sigma^2 = \mathrm{VAR}[y] - \sigma_n^2$$

$$\mathrm{VAR}[y] = \mathrm{MEAN}[y^2].$$

$$\hat{\sigma} = \sqrt{\max\left(\mathrm{MEAN}[y^2] - \sigma_n^2, 0\right)}$$

[ref]:http://eeweb.poly.edu/iselesni/lecture_notes/SoftThresholding.pdf



g. 13. (a) Original image. (b) Noisy image with PSNR = 20.02 dB. (c) Denoised image using soft thresholding; PSNR = 27.73 dB. (d) Denoised

From: Sendur, Levent, and Ivan W. Selesnick. "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency." IEEE Transactions on signal processing 50.11 (2002): 2744-2756. http://eeweb.poly.edu/iselesni/bishrink/BiShrinkTSP.pdf

Sparse Recovery in Underdetermined Setting

- y = Hz + n = G x + n, G=HT is fat (MxN, M<N)
  - y: M measurements; H: MxK
  - z: K unknown elements
  - x: N coefficients, transform matrix T: KxN; z=Tx
- Application scenarios
  - Denoising: y is direct noisy measurement of signal, H=I (M=K), G=T is a dictionary (KxN, K<N)
  - Deblurring: A=H T: H is complete measurement (M=K), T is a dictionary
  - Compressed sensing: H (MxK) is a compressed imaging operator (M<K, less measurements than unknowns), T is either orthonormal (N=K) or overcomplete (K<N)
  - In-painting: A = M T: M is the mask of known pixels (M<K), T is either orthonormal (K=N) or overcomplete (K<N)
- G^T G \= I, cannot change to a problem of $y_t = x_t + n_t$
  - Cannot directly use soft thresholding!

Multiple Forms of L1 Problem

- Noise free measurements y=Gx
  - $\min\|x\|_1, s.t.\ Gx = y$ (Constrained optimization)
- Noisy measurements
  - P1: $\min\|x\|_1, s.t.\ \|Gx - y\|_2 \leq \epsilon$
    - Need to know noise variance
  - P2: $\min\|Gx - y\|_2^2,\ \ s.t.\ \|x\|_1 \leq s$
    - Need to know sparsity of x (number of non-zeros)
  - P3: Bring constraint to the objective function:

    $\min\|Gx - y\|_2^2 + \lambda\|x\|_1$ (LASSO problem)
    - Need to set regularization parameter λ properly

Solving LASSO Problem

- ISTA /FISTA
- ADMM

Iterative Soft Thresholding Algorithm (ISTA)

- Solving LASSO Problem: min $J(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1.$
- Through majorization-minimization (MM), each step solve a single variable problem (using Soft thresholding)
- Convergence very slow, but easy to implement

$$\mathbf{x}_{k+1} = \text{soft}\left(\mathbf{x}_k + \frac{1}{\alpha}\mathbf{H}^\mathsf{T}(\mathbf{y} - \mathbf{H}\mathbf{x}_k),\ \frac{\lambda}{2\alpha}\right)$$

$$\alpha \geq \text{maxeig}(\mathbf{H}^\mathsf{T}\mathbf{H})$$

- For faster convergence, choose $\alpha = \max \text{eig}(\mathrm{H}^T H)$

Original ISTA paper:

For derivation, see Selesnick's note on Sparse Signal Recovery at http://eeweb.poly.edu/iselesni/lecture_notes/sparse_signal_restoration.pdf

Special Case of ISTA: H is an Orthonormal Operator

- If H=B represents an orthonormal transform: Hx is inverse transform, $H^T y$ is forward transform
- When H is orthonormal

$$H^T H = I, \alpha = 1,\ x_{k+1} = soft(H^T y, \frac{\lambda}{2})$$

- This means that only one iteration is sufficient. Solution is simply soft-thresholding of the original coefficients
  - Reduces to the previous solution with orthonormal transform!

Faster Algorithms

- Faster ISTA (FISTA): Converge much faster!
  - A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci., 2(1):183{202, 2009.
- SALSA (split variable augmented Lagrangian shrinkage algorithm)
  - M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. IEEE Trans. Image Process., 19(9):2345-2356, September 2010.
  - M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. IEEE Trans. Image Process., 20(3):681-695, March 2011.
- Explanation using Proximal operators
  - http://stanford.edu/~boyd/papers/pdf/prox_slides.pdf
  - http://stanford.edu/~boyd/papers/pdf/prox_algs.pdf

ADMM Algorithm

- A flexible optimization algorithm that can handle many convex optimization problems
  - Represent different terms of the objective function using additional variables and introducing constraints
- Built on the Lagrangian multiplier method
- Solve the dual problem
- Add quadratic penalty (method of multipliers) to ease update of the dual variable and be more robust
- Note: Notations are different from before (y is not measurement!)

[Ref] Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers (Boyd, Parikh, Chu, Peleato, Eckstein)
https://web.stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf

Lagrangian Method for Constrained Problem

- convex equality constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

- Lagrangian: $L(x, y) = f(x) + y^T(Ax - b)$

Lagrangian multiplier

Need to solve for both the multiplier y and unknown x!

Cropped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

Dual Problem (Optional)

- convex equality constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

- Lagrangian: $L(x, y) = f(x) + y^T(Ax - b)$
- dual function: $g(y) = \inf_x L(x, y)$
- dual problem: maximize $g(y)$
- recover $x^\star = \operatorname{argmin}_x L(x, y^\star)$

Lagrangian multiplier

Cropped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

Dual Ascent (Optional)

- gradient method for dual problem: $y^{k+1} = y^k + \alpha^k \nabla g(y^k)$
- $\nabla g(y^k) = A\tilde{x} - b$, where $\tilde{x} = \text{argmin}_x \, L(x, y^k)$
- dual ascent method is

$$
\begin{aligned}
x^{k+1} &:= \text{argmin}_x \, L(x, y^k) && \text{// } x\text{-minimization} \\
y^{k+1} &:= y^k + \alpha^k (Ax^{k+1} - b) && \text{// dual update}
\end{aligned}
$$

- works, with lots of strong assumptions

Need to select α_k properly!

Cropped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

Dual Decomposition (optional)

- suppose f is separable:

$$f(x) = f_1(x_1) + \cdots + f_N(x_N), \quad x = (x_1, \ldots, x_N)$$

- then L is separable in x: $L(x, y) = L_1(x_1, y) + \cdots + L_N(x_N, y) - y^T b$,

$$L_i(x_i, y) = f_i(x_i) + y^T A_i x_i$$

- x-minimization in dual ascent splits into N separate minimizations

$$x_i^{k+1} := \underset{x_i}{\operatorname{argmin}}\, L_i(x_i, y^k)$$

  which can be carried out in parallel

Cropped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

Dual Decomposition (optional)

- dual decomposition (Everett, Dantzig, Wolfe, Benders 1960–65)

$$x_i^{k+1} := \operatorname{argmin}_{x_i} L_i(x_i, y^k), \quad i = 1, \ldots, N$$
$$y^{k+1} := y^k + \alpha^k (\textstyle\sum_{i=1}^N A_i x_i^{k+1} - b)$$

- scatter y^k; update x_i in parallel; gather $A_i x_i^{k+1}$
- solve a large problem
  - by iteratively solving subproblems (in parallel)
  - dual variable update provides coordination
- works, with lots of assumptions; often slow

Croped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

Method of Multipliers (optional)

- a method to robustify dual ascent
- use **augmented Lagrangian** (Hestenes, Powell 1969), $\rho > 0$

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + (\rho/2)\|Ax - b\|_2^2$$

- method of multipliers (Hestenes, Powell; analysis in Bertsekas 1982)

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x L_\rho(x, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} - b) \end{aligned}$$

(note specific dual update step length ρ)

Cropped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

Alternating direction method of multipliers (ADMM)

- ADMM problem form (with f, g convex)

$$\begin{array}{ll} \text{minimize} & f(x)+g(z) \\ \text{subject to} & Ax+Bz=c \end{array}$$

Typical use case:
f(x) is quadratic in x
g(z) contains L1 norm
B= diagonal

  - two sets of variables, with separable objective

Can be grouped as \rho/2 ||y/rho+ (Ax+Bz-c)||^2 by completing square

- $L_\rho(x,z,y)=f(x)+g(z)+y^T(Ax+Bz-c)+(\rho/2)\|Ax+Bz-c\|_2^2$

- ADMM:

Minimizing a quadratic problem, with closed-form solution

$$\begin{aligned} x^{k+1} &:= \operatorname{argmin}_x L_\rho(x,z^k,y^k) && \text{// } x\text{-minimization} \\ z^{k+1} &:= \operatorname{argmin}_z L_\rho(x^{k+1},z,y^k) && \text{// } z\text{-minimization} \\ y^{k+1} &:= y^k+\rho(Ax^{k+1}+Bz^{k+1}-c) && \text{// dual update} \end{aligned}$$

Soft thresholding if B=diagonal

Croped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

ADMM Advantages

- assume (very little!)
  - f, g convex, closed, proper
  - L_0 has a saddle point
- then ADMM converges:
  - iterates approach feasibility: $Ax^k + Bz^k - c \to 0$
  - objective approaches optimal value: $f(x^k) + g(z^k) \to p^\star$
- a method
  - with good robustness of method of multipliers
  - which can support decomposition
- "robust dual decomposition" or "decomposable method of multipliers"
- proposed by Gabay, Mercier, Glowinski, Marrocco in 1976

Croped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

ADMM for LASSO

- LASSO problem: minimize $(1/2)\|Ax-b\|_2^2+\lambda\|x\|_1$

- Derive in class
  - ADMM formulation
  - Augmented objective function
  - x minimization
  - z minimization
  - y update

Croped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

ADMM for LASSO

- LASSO problem: minimize $(1/2)\|Ax-b\|_2^2+\lambda\|x\|_1$

  Introducing a new variable on any term with L1 norm and add corresponding constraints

  Note: matrix A here is not the same as in previous slide

- ADMM form:

$$\begin{array}{ll}\text{minimize} & (1/2)\|Ax-b\|_2^2+\lambda\|z\|_1\\ \text{subject to} & x-z=0\end{array}$$

- ADMM:

$$\begin{aligned}x^{k+1} &:= (A^TA+\rho I)^{-1}(A^Tb+\rho z^k-y^k)\\ z^{k+1} &:= S_{\lambda/\rho}(x^{k+1}+y^k/\rho)\\ y^{k+1} &:= y^k+\rho(x^{k+1}-z^{k+1})\end{aligned}$$

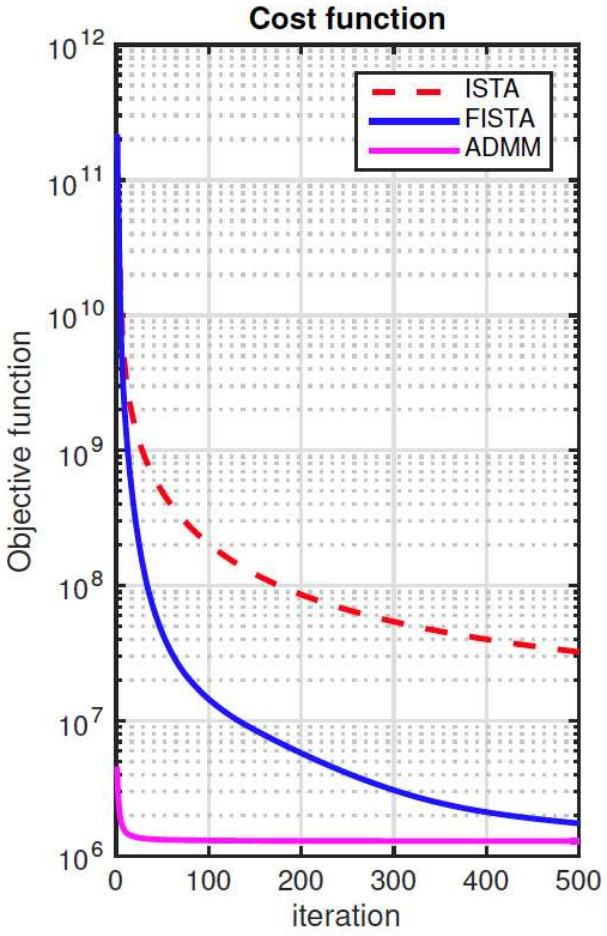
Croped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

Comparison of ISTA and ADMM

- ADMM is usually much faster than ISTA to reach a reasonable (practically acceptable) solution, but then slows down to reach the optimal more accurate solution.
- Plot from Amirhossein Khalilian-Gourtani

blurred noisy image σ = 0.05

ADMM restored



FISTA restored

ISTA restored

"Smooth" Regularization

- Assume the signal is smooth (difference between pixels is small)
  - Difference between pixels z(n)=x(n)-x(n-1) -> z=Dx

$$D = \begin{bmatrix} 1 & 0 & 0 & & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

  - D can also be second-order difference operator. D=?
- 2D image:
  - Can perform 1D difference along rows and columns respectively, yielding $D_1 x$ (row-wise), $D_2 x$ (column-wise),
  - If we order an image into a vector x row-by-row, how do D_1, D_2 look?
- Maximize smoothness = Minimize difference magnitude

http://eeweb.poly.edu/iselesni/lecture_notes/least_squares/least_squares_SP.pdf

Smooth Regularization Using Least Squares

- Using squared magnitude of difference Dx
- Optimization problem (1D)
  - Minimize: $J(x) = \|Hx - y\|^2 + \lambda\|Dx\|^2$
  - $\frac{\partial}{\partial x} J(x) = H^T(Hx - y) + \lambda\, D^T Dx = 0$
  - $x = (H^T H + \lambda\, D^T D)^{-1}\, H^T y$
  - Directly inverting the matrix is not computationally efficient
  - Can find corresponding filters corresponding to $(H^T H + \lambda\, D^T D)^{-1}\, H^T$
- Optimization problem (2D)
  - Minimize: $J(x) = \|Hx - y\|^2 + \lambda_1\|D_1 x\|^2 + \lambda_2\|D_2 x\|^2$
  - $\frac{\partial}{\partial x} J(x) = H^T(Hx - y) + \lambda\, D^T Dx = 0$
  - $x = (H^T H + \lambda_1 D_1^T D_1 + \lambda_2\, D_2^T D_2)^{-1}\, H^T y$
  - Alternatively, perform 1D operation horizontally, and then vertically
- Tend to blur edges (many small differences)

Total Variation (TV) Regularization

- Instead of using squared magnitude of the difference, using L_1
  - Minimizing L2 tend to yield small solution: all elements are small
  - Minimizing L1 tend to yield sparser solution: each element is either 0 or large
  - With L1 on Dx, the difference between two nearby pixels are either 0, or have large difference
  - TV tend to yield piecewise smooth images, keeping sharp edges
- Anisotropic TV (sum of TV in each direction, anisotropic TV):

$$TV(x) = \sum_{m,n} (|x(m,n) - x(m-1,n)| \quad + |x(m,n) - x(m,n-1)|$$

$$= \|D_1 x\|_1 \; + \|D_2 x\|_1 \; = \|Dx\|_1, \qquad D = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix}$$

- Isotropic TV (magnitude of TV in multiple direction): TV:

$$TV(x) = \left(\sum_{m,n} (x(m,n) - x(m-1,n))^2 + (x(m,n) - x(m,n-1))^2 \right)^{1/2}$$

$$= (\|D_1 x\|_2^2 + \|D_2 x\|_2^2)^{1/2}$$

ADMM for TV-Based Deblurring of 1D Signal

- Minimize J(x) = ½ $|Ax-b|^2 + \lambda|Fx|_1$
  - F is the first-order or second order difference operator
    - note notation change
  - Have F in front of x, not possible to do soft thresholding directly.

ADMM Trick: Introduce z=Fx

$$\begin{aligned} &\text{minimize} && (1/2)\|Ax-b\|_2^2 + \lambda\|z\|_1 \\ &\text{subject to} && Fx - z = 0, \end{aligned}$$

$$\begin{aligned} x^{k+1} &:= (A^TA + \rho F^TF)^{-1}(A^Tb + \rho F^T(z^k - u^k)) \\ z^{k+1} &:= S_{\lambda/\rho}(Fx^{k+1} + u^k) \\ u^{k+1} &:= u^k + Fx^{k+1} - z^{k+1}. \end{aligned}$$

Homework: derive the above ADMM solution!

For denoising: A=I, and if F corresponds to a single 1D difference operator for 1D signal, the matrix to be inverted is tridiagonal, and can be inverted in O(N) flops.

Croped from: https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

ADMM for TV Denoising/Debluring for Images/Video (Optional)

- Previous algorithm still works for anisotropic TV using vector representation
- However, $F^T F$ is not a simple tridiagonal matrix any more
- Matrix free operations through Fourier Transform!
- Following use different notations!
- L2+TV: (good for Gaussian noise)

$$\underset{\mathbf{f}}{\text{minimize}} \quad \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{Df}\|_1$$

$$\begin{aligned}\underset{\mathbf{f},\mathbf{u}}{\text{minimize}} \quad & \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{u}\|_1 \\ \text{subject to} \quad & \mathbf{u}=\mathbf{Df}.\end{aligned}$$

- L1+TV (Good for Laplacian noise)

$$\underset{\mathbf{f}}{\text{minimize}} \quad \mu\|\mathbf{Hf}-\mathbf{g}\|_1 + \|\mathbf{f}\|_{TV}$$

$$\begin{aligned}\underset{\mathbf{f},\mathbf{r},\mathbf{u}}{\text{minimize}} \quad & \mu\|\mathbf{r}\|_1 + \|\mathbf{u}\|_1 \\ \text{subject to} \quad & \mathbf{r}=\mathbf{Hf}-\mathbf{g} \\ & \mathbf{u}=\mathbf{Df}.\end{aligned}$$

S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill and T. Q. Nguyen, "An Augmented Lagrangian Method for Total Variation Video Restoration," in *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3097-3111, Nov. 2011.

https://ieeexplore.ieee.org/document/5779734

TV Based Deblurring

Input $\mu = 10^2$ $\mu = 10^3$ $\mu^* = 10352$ $\mu = 10^5$

1. TV/L2 Image recovery using different choices μ. The optimal (in terms of PSNR compared to the reference) is $\mu = 10352$. The image is b
Gaussian blur kernel of size 9×9, $\sigma = 5$. Addition Gaussian noise is added to the image so that the blurred signal to noise ratio (BSNR) is 40

Input $\mu = 0.01$ $\mu = 1$ $\mu^* = 7$ $\mu = 50$

2. TV/L1 Image recovery using different choices μ. The optimal (in terms of PSNR compared to the reference) is $\mu = 7$. The image is blurred
ssian blur kernel of size 9×9, $\sigma = 1$. 10% of the pixels are corrupted by salt and pepper noise. Image source: [39].

S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill and T. Q. Nguyen, "An Augmented Lagrangian Method for Total Variation Video Restoration," in *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3097-3111, Nov. 2011.
https://ieeexplore.ieee.org/document/5779734

Other Ways for TV denoising

- Can also be solved using MM
  - http://eeweb.poly.edu/iselesni/lecture_notes/TVDmm/index.html

ADMM for Least Absolute Deviations

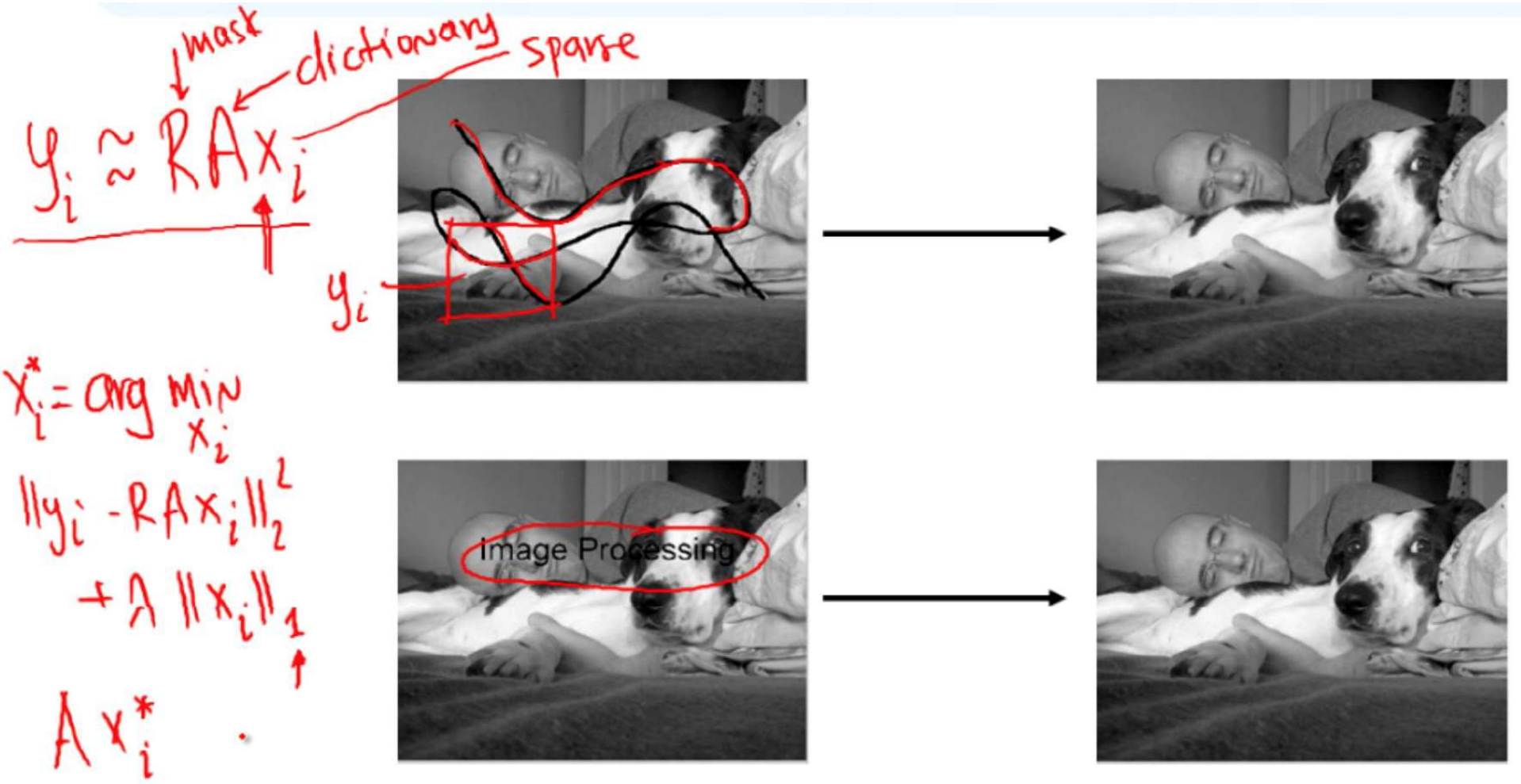
- Regression to minimize sum of absolute error instead of squared error (more robust to outliers)
- Original problem (assuming b=Ax+n)
  - Minimize $||Ax-b||_1$
- ADMM formulation:

$$\begin{array}{ll} \text{minimize} & \|z\|_1 \\ \text{subject to} & Ax - z = b, \end{array} \qquad f = 0 \text{ and } g = \|\cdot\|_1$$

- ADMM solution?
  - Homework (optional)

[Ref] Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers (Boyd, Parikh, Chu, Peleato, Eckstein)
https://web.stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf

Image Inpainting



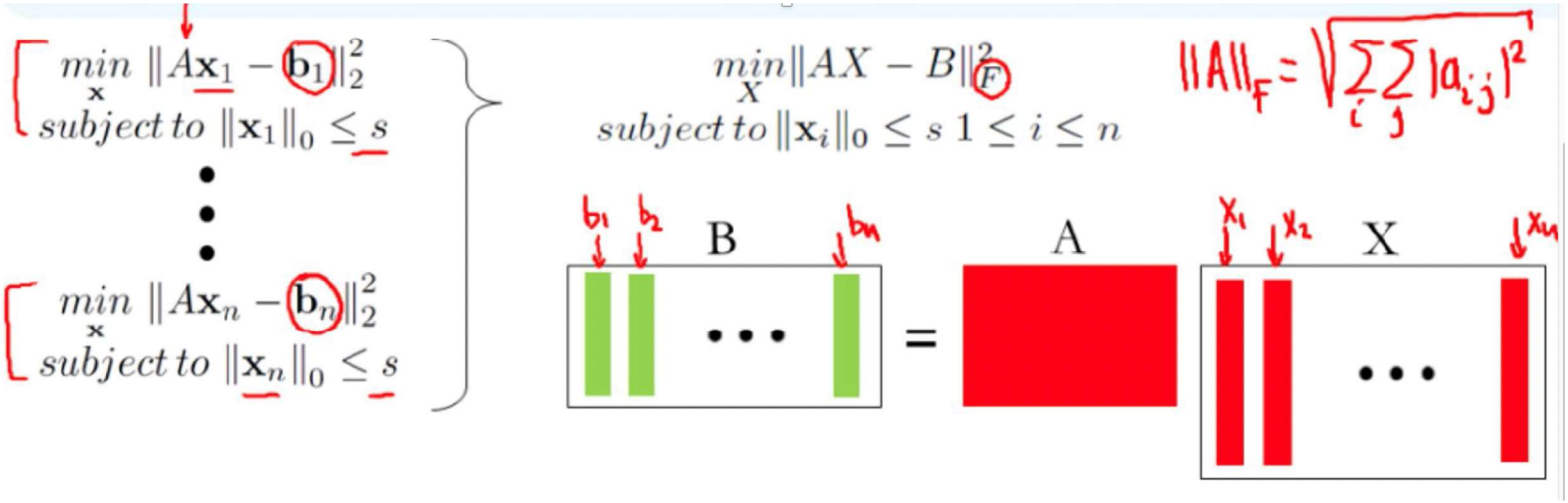
From Katsagelos's Coursera Course on Image Processing, Lecture on Sparsity.

ADMM for Image Completion

- Known samples are related to all samples through a mask: b = Mx
  - Properties of M: Diagonal, $M M^T=I$, $M^TM=diag(m)$
  - see Selesnick_sparse_sp_intro.pdf
- All samples are represented through a transform x=Tz
- Constraint: b= MTz = Gz
- Two formulations
  - Requiring the known values to be retained: using Gz=b as a constraint, minimizing $|z|_1$
  - Alternative (assuming known values are noisy): Minimize |Gz-b|^2 +λ $|z|_1$
  - How to solve using ADMM?
  - homework!

Dictionary Learning

- Want to use a transform/dictionary that has the sparsest representation
- Given many data samples b_i (e.g. image blocks), how to determine the dictionary (atoms a_k, k=1,2,…)?

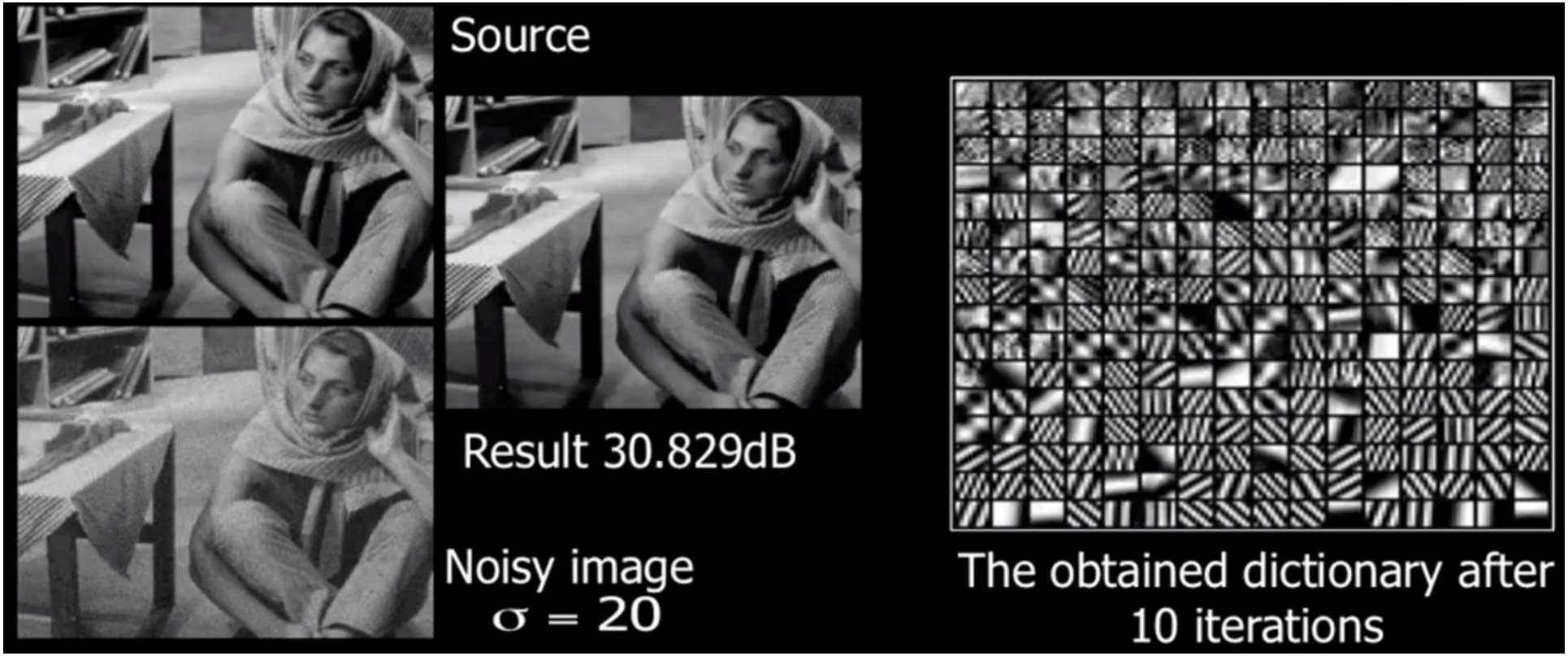


From Katsagelos's Coursera Course on Image Processing, Lecture on Sparsity.

Transform / Dictionary Learning (Optional)

- Non-redundant orthonormal transform
  - KLT maximizes energy compaction, not sparsity
  - Sparse orthogonal transform (SOT)
    - O. G. Sezer, O. G. Guleryuz and Y. Altunbasak, "Approximation and Compression With Sparse Orthonormal Transforms," in *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2328-2343, Aug. 2015.
- Redundant transform (dictionary)
  - KSVD (iterative algorithm)
    - Given a dictionary, solve sparse coding problem (given A and bi, solve x_i to minimize |x_i|_0 (matching pursuit), or minimize |x_i|_1 (LASSO)
    - Then Update the dictionary (through SVD)
    - M. Aharon, M. Elad and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," in *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, Nov.
  - Online learning
    - Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09). ACM, New York, NY, USA, 689-696. https://www.di.ens.fr/willow/pdfs/icml09.pdf

Image Denoising With Dictionary Learning



From Shapiro's Coursera Course on Image Processing, Lecture on Sparse Modeling and Compressed Sensing.

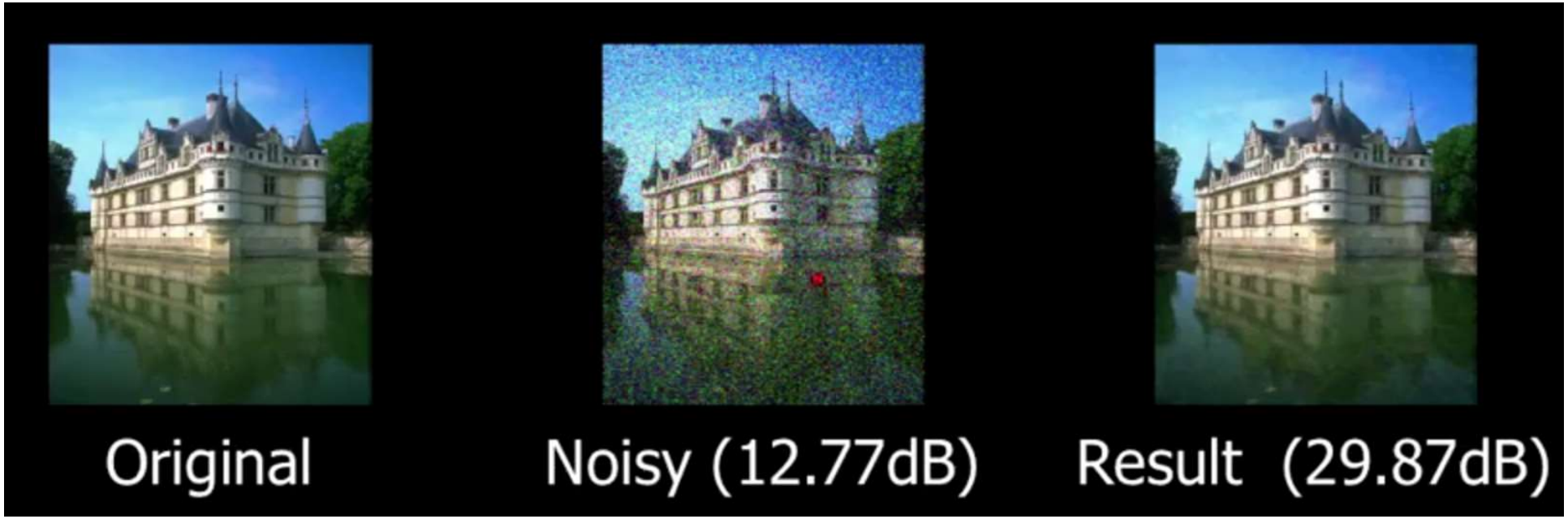
Application to Images

- Processing each image block separately
  - Using DCT or other fixed dictionary
  - Using learnt dictionary that maximizes sparsity
    - Non-redundant (SOT) vs. redundant dictionary (KSVD)
  - Can use overlapping blocks to remove boundary effect
- Process the entire image
  - Using wavelet transforms (orthonormal)
  - Using wavelet frames (redundant transform, satisfying $T\,T^T=I$)

More Advanced Techniques

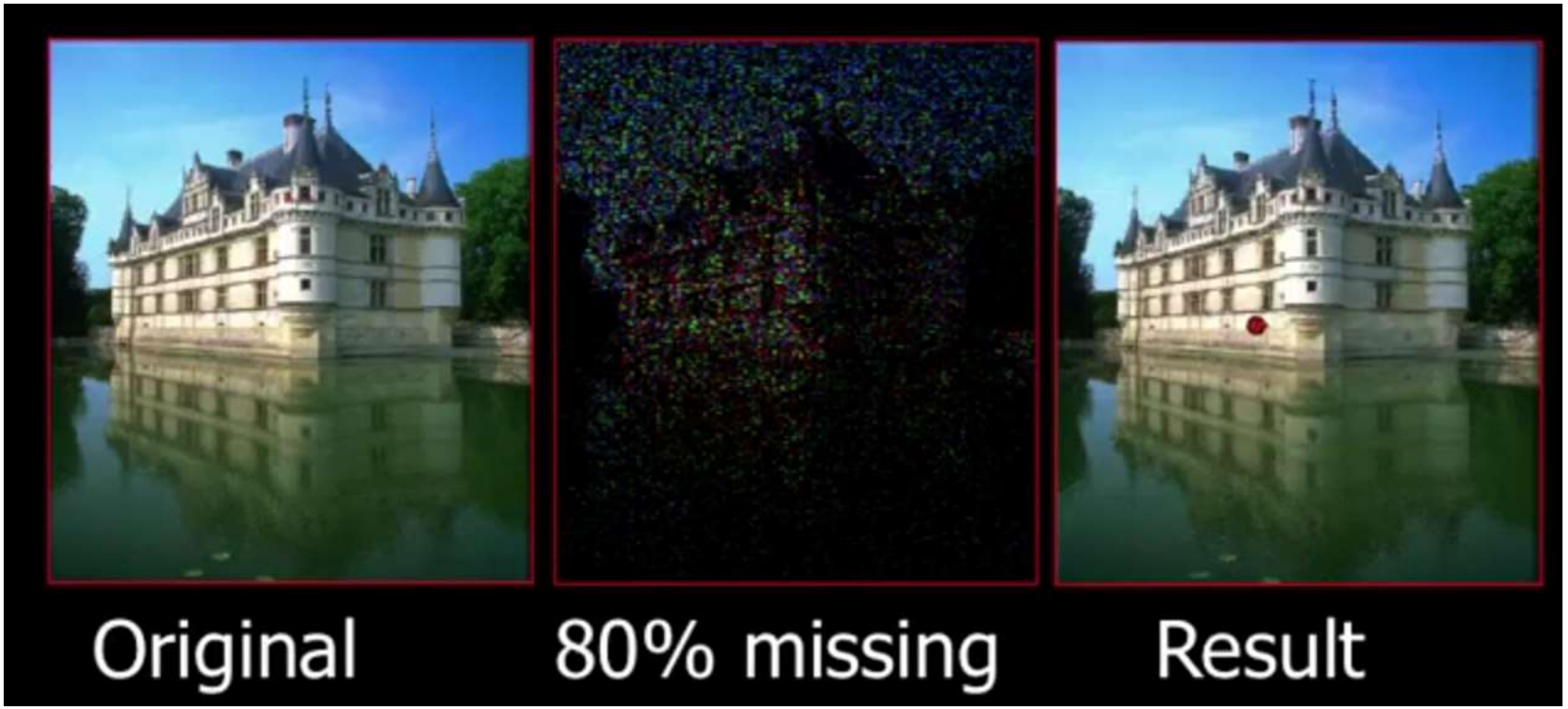
- Using overlapping image blocks
  - Average results from overlapping reconstructed blocks
  - Overlapping provides significant gain!
- Dictionary learning
  - Same dictionary applied to all (overlapping) blocks
  - Pretrained dictionary: Using blocks from training images
  - Image adaptive dictionary: Using image blocks in the given (noisy/incomplete) image to learn the dictionary
  - Online adaptation: Dictionary is updated with each new sample

Image Denoising (with Overlapping Blocks and Image Adaptive Dictionary)



From Shapiro's Coursera Course on Image Processing, Lecture on Sparse Modeling and Compressed Sensing.

Image Inpainting (with Overlapping Blocks and Image Adaptive Dictionary)



From Shapiro's Coursera Course on Image Processing, Lecture on Sparse Modeling and Compressed Sensing.

Image Inpainting (with Overlapping Blocks and Image Adaptive Dictionary)

From Shapiro's Coursera Course on Image Processing, Lecture on Sparse Modeling and Compressed Sensing.

Summary

- Regularization using prior knowledge:
  - Image is sparse in a properly chosen transform/dictionary (transform based methods)
  - Image is smooth everywhere except near edges (TV-based methods)
  - Both need to minimize L0 norm
- Convex relaxation:
  - Relax L0 to L1
- Optimization approach for solving L2-L1 problem
  - Soft thresholding
  - ISTA
  - ADMM
- How to set regularization parameter \lambda?
  - Given signal and noise distribution, can set optimally based on MAP formulation
  - Generally has to use “trial-and-error”

Reading Assignment

- Notes by Prof. Ivan Selesnick
  - http://eeweb.poly.edu/iselesni/lecture_notes/
  - Least_squares_SP.pdf, sparse_SP_intrp.pdf, sparse_signal_restoration.pdf, SoftThresholding.pdf
- ADMM
  - Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers (Boyd, Parikh, Chu, Peleato, Eckstein. Sec. 2,3,6. https://web.stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf

Other References

- Some excellent review papers:
  - M. Zibulevsky and M. Elad, "L1-L2 Optimization in Signal and Image Processing," in *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 76-88, May 2010. https://ieeexplore.ieee.org/document/5447114
  - Mairal, Julien, Francis Bach, and Jean Ponce. "Sparse modeling for image and vision processing." *Foundations and Trends® in Computer Graphics and Vision* 8.2-3 (2014): 85-283. https://arxiv.org/abs/1411.3230 (including dictionary learning)
  - Chambolle, Antonin, and Thomas Pock. "An introduction to continuous optimization for imaging." *Acta Numerica* 25 (2016): 161-319. https://hal.archives-ouvertes.fr/hal-01346507/document
- Other links
  - Coursera Course by Prof. Katsaggelos, https://www.coursera.org/learn/digital/home/welcome
  - Coursera Course by Prof. Shapiro, https://www.coursera.org/learn/image-processing/home/welcome
  - Homepage of Prof. John Wright's course on sparsity: http://www.columbia.edu/~jw2966/6886_Fa2015.html
  - M. Elad, Sparse and redundant representations: From Theory to Applications in Signal and Image Processing, Springer, 2010. http://www.springer.com/us/book/9781441970107
  - Patrick L. Combettes† and Jean-Christophe Pesquet, "Proximal Splitting Methods in Signal Processing," https://arxiv.org/pdf/0912.3522.pdf

Softwares

- Sparse Modeling Software
  - http://spams-devel.gforge.inria.fr/
- Optimization Software
  - Matlab: linprog, quadprog
  - http://cvxr.com/cvx/

Written Homework

1. In the MAP formulation of the wavelet denoising problem in slides 19-21, we assumed that the wavelet coefficients follow a Laplacian distribution. Derive a corresponding solution if the coefficients follow a Gaussian distribution.
2. As described in the class, image deblurring using TV regularization can generally be formulated as the following optimization problem:

   $x= \text{argmin}_x \{ J(x) = ½ |Ax-b|^2 + \lambda|Fx|_1 \}$

   Where F is the first-order or second order difference operator

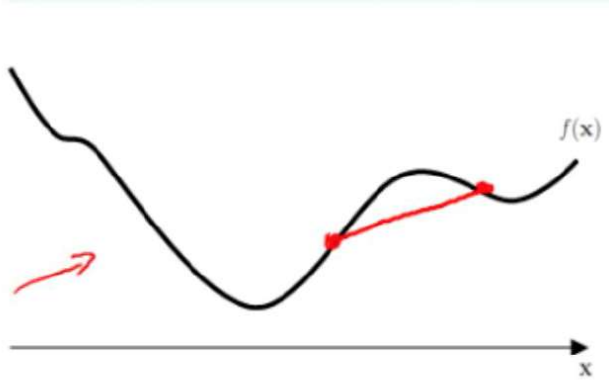
   Reformulate the problem so that it can be solved using ADMM and derive the corresponding iterative algorithm for solving the problem.

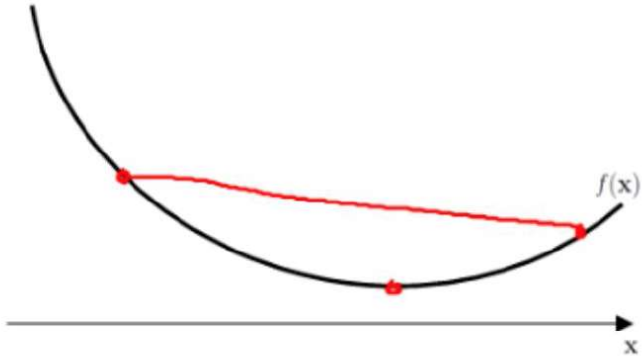
3. For the image completion problem (Slide 50-51), propose one ADMM formulation and the corresponding iterative algorithm.

Review: Basics of Optimization

- Unconstrained optimization:
- Constrained optimization
- Local vs. Global minimum
- Convex optimization
  - Loss function is convex, constraint set is convex
- For convex problem, every local minimum is a global minimum.
- Convex relaxation: Approximate a non-convex problem by a convex problem so that it is easy to solve. But need to study the conditions under which the two give equal or similar solutions

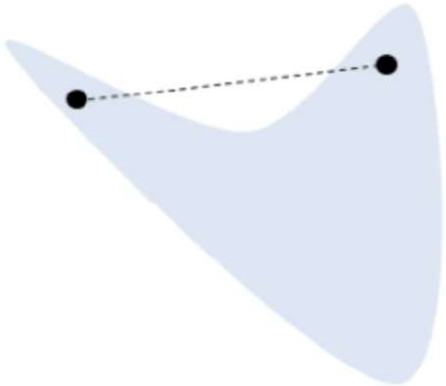
Convex Function and Convex Sets

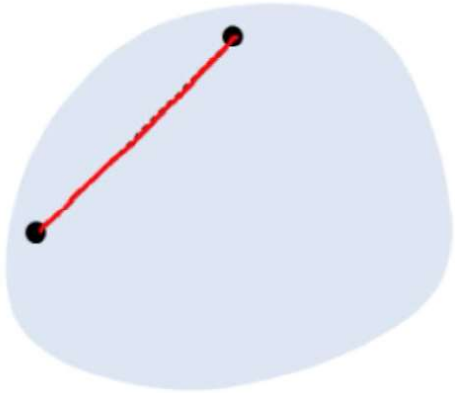


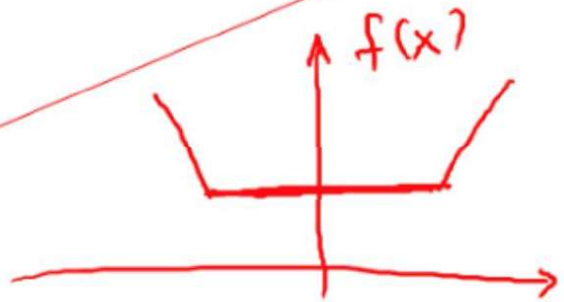


$$\min_{\mathbf{x}} f(\mathbf{x})$$

subject to $\mathbf{x} \in \mathcal{S}$







Unconstrained Optimization and Gradient Descent

- Unconstrained: x*= argmin J(x)
- Necessary condition: Gradient of J(x)=0: $\frac{\partial}{\partial x}J(x)=0$
- If there are close form solution for the above equation, we are done!
- If not, use gradient descent

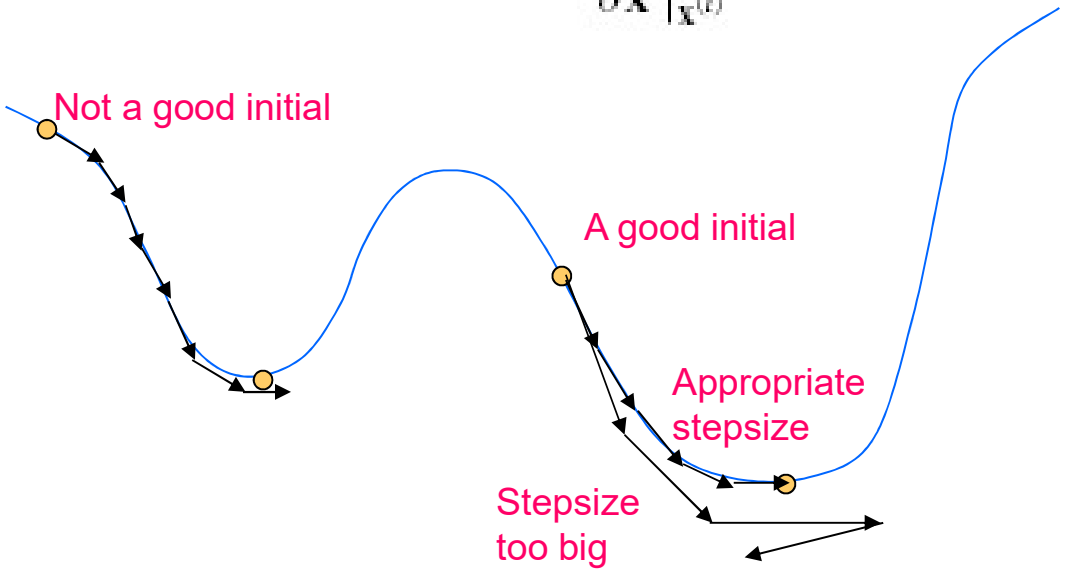
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \frac{\partial}{\partial x} J(x)\Big|_{x^{(t)}}$$

- If J(x) is convex, there is only one local minimum
  - Does not matter what is the initial condition $x^{(0)}$

Gradient Descent Method

- Iteratively update the current estimate in the direction opposite the gradient direction.

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \alpha \left.\frac{\partial J}{\partial \mathbf{x}}\right|_{\mathbf{x}^{(l)}}$$



- The solution depends on the initial condition. Reaches the local minimum closest to the initial condition
- Yield optimal solution if J is convex regardless initial solution

Method of Lagrange Multipliers

- Convert a constrained problem with Equality constraint to a non-constrained problem
- Original Problem:
  - Min J(x)
  - Subject to g(x)=c
- Augmented problem
  - Min L(x)=J(x)+λ (g(x)-c))
  - Necessary condition: Gradient$_{x,\lambda}$ L(x, λ)=0
  - Equivalent to solve:
    - Gradient $_x$ J(x)= -λ gradient$_x$ g(x)
    - g(x)=c

A “sloppy” Way

- Just minimize L(x, λ) for some chosen λ
- λ controls the weighting between the original cost and constraint
- Trial and error to see which λ yields best solution

Other More Advanced Method

- When some terms of J(x) are not differentiable but convex
- MM (majorization minimization)
  - Selesnick's note on Sparse Signal Recovery at http://eeweb.poly.edu/iselesni/lecture_notes/sparse_signal_restoration.pdf
- ADMM (Alternating direction method of multipliers)
  - Boyd's et al, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers."
  - http://web.stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf
- Proximal Splitting
  - Patrick L. Combettes† and Jean-Christophe Pesquet, "Proximal Splitting Methods in Signal Processing," https://arxiv.org/pdf/0912.3522.pdf

Review: Vector Derivatives

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \qquad \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_N} \end{bmatrix}$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{b}^T \mathbf{x} = \mathbf{b}, \qquad \frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{y} - \mathbf{x})^T \mathbf{A} (\mathbf{y} - \mathbf{x}) = 2\mathbf{A}(\mathbf{x} - \mathbf{y})$$

- Note: Chain Rule Apply
- Trick: Think of all variables as scalars, and order items so that the dimension matches for matrix multiplications

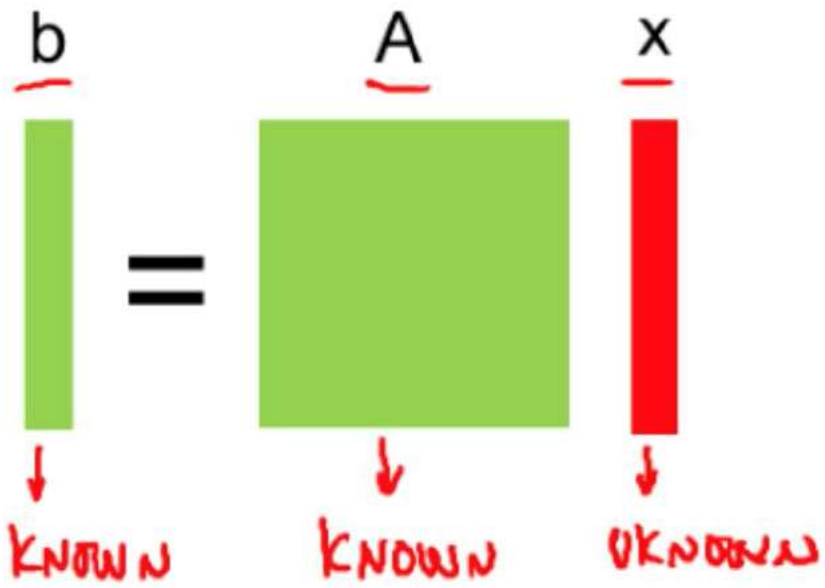
From: http://eeweb.poly.edu/iselesni/lecture_notes/sparse_signal_restoration.pdf

Review: Solution of Linear Equations

- Solving $A_{MxN}\, x_{Nx1} = b_{Mx1}$

- M: # equations, N: # unknowns (dimension of vector)
- M=N, A is Full Rank: Uniquely determined
- M>N: Over-determined
  - Least squares solution
- M<N: Under-determined
  - Need extra constraint!

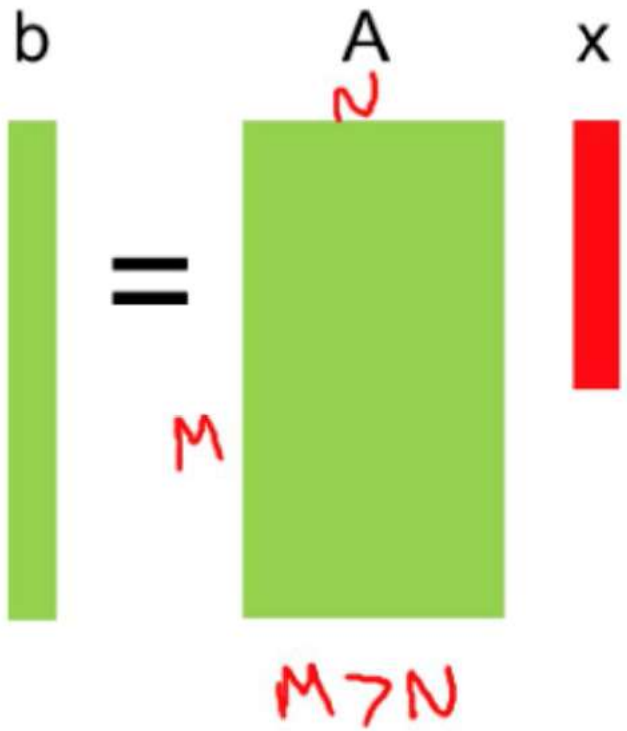
Uniquely Determined (M=N)



If A is full rank (all columns are non-linearly correlated)

$$\mathbf{x}^* = A^{-1}\mathbf{b}$$

Over-Determined Problem (M>N)



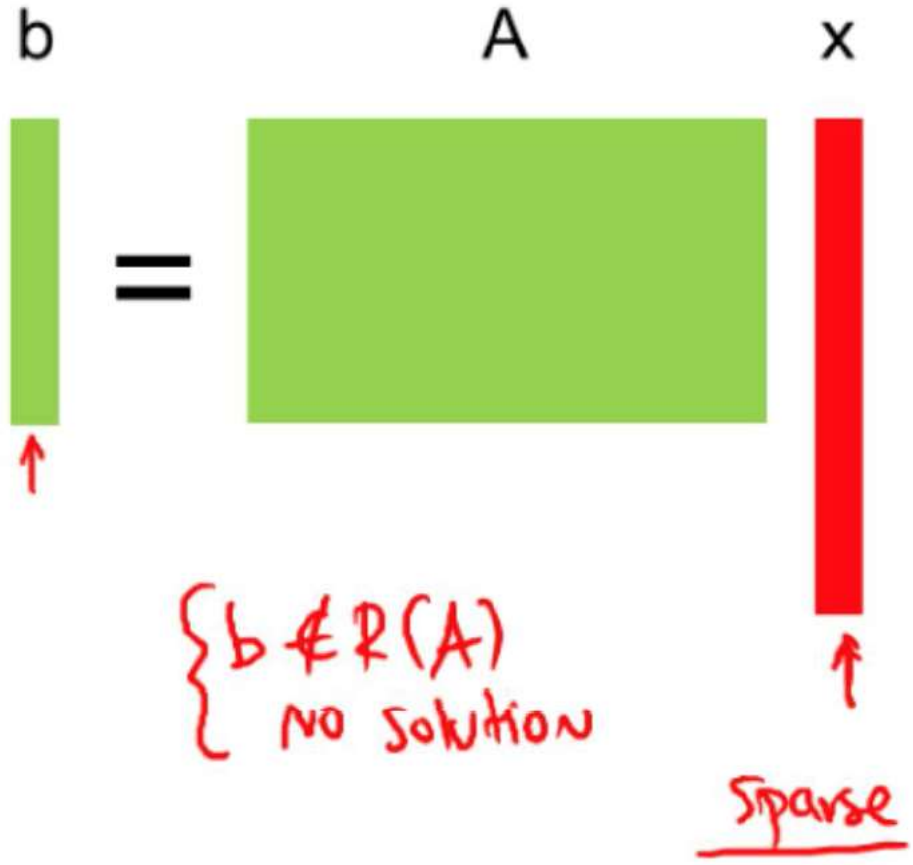
Least squares solution =
Unconstrained Optimization

$J(x)= |Ax-b|_2^2=(Ax-b)^T (Ax-b)$

$$\mathbf{x}^* = \left(A^T A\right)^{-1} A^T \mathbf{b}$$

NXN .

Under-Determined Problem (M < N)



- Infinitely many solutions
- Needs to make use of prior knowledge about x (called priors) (known as regularization)
- The prior knowledge can usually be represented as a cost function J(x)
- Change to constrained optimization problem with equality constraint:

$$\min_{\mathbf{x}} J(\mathbf{x}) \text{ subject to } \mathbf{b} = \mathbf{A}\mathbf{x}$$

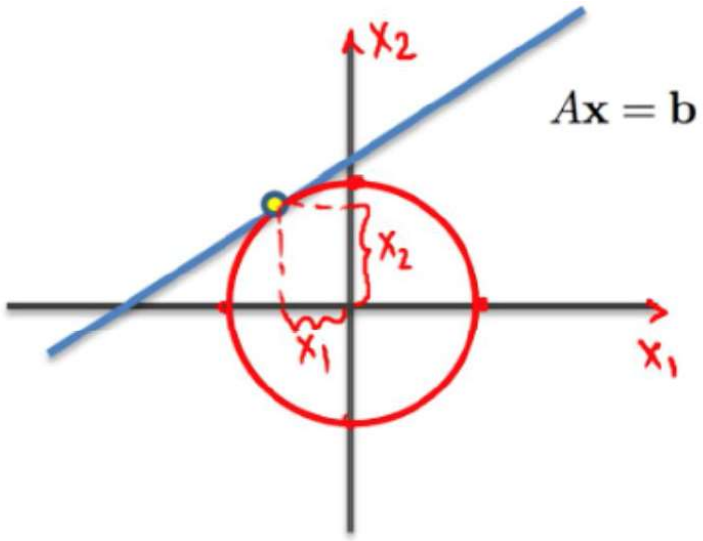
Minimum L2-Norm Solution (Easy!)

- Among all solutions, find one with minimum energy (=L2 norm squa
- L2 Norm: $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$
- Optimization problem:

$$\min_{\mathbf{x}} ||\mathbf{x}||_2$$

$$subject\ to\ A\mathbf{x} = \mathbf{b}$$

- Solution (Using Lagrange Method)



$$\mathbf{x}^* = A^T \left(AA^T\right)^{-1} \mathbf{b}$$