Introduction to Compressive Sensing and Sparse Optimization

Ming Yan
yanm@math.ucla.edu

Department of Mathematics, UCLA

Compressive Sensing Framework

- Sparse representation
- Encoding and decoding

Sparse Optimization Algorithms

- Primal algorithms
- Dual algorithms
- Greedy algorithms
- Other methods

Outline

Background

- Nyquist/Shannon sampling theory: A band-limited signal of interest with highest frequency $\mathbf{B}$ can be exactly reconstructed from its uniformly spaced samples if the rate of sampling exceeds $2\mathbf{B}$ (**Nyquist rate**). This is independently discovered by Kotelnikov, Nyquist, Shannon, and Whitaker.
- The sampling rate needs to be **very high** if the original signal contains high frequencies (to avoid aliasing). The excessive number of samples make compression a necessary prior to storage or transmission. In addition, increasing the sampling rate is very expensive, time consuming (MRI), or dangerous (CT).
- On the other hand, the chance is that most signals we are interested in are highly compressible, namely, they can be represented by a set of sparse or nearly sparse coefficients. CS exploits this feature of signals and thus allows a sampling rate significantly lower than the Nyquist rate.

Efficient image/signal acquisition

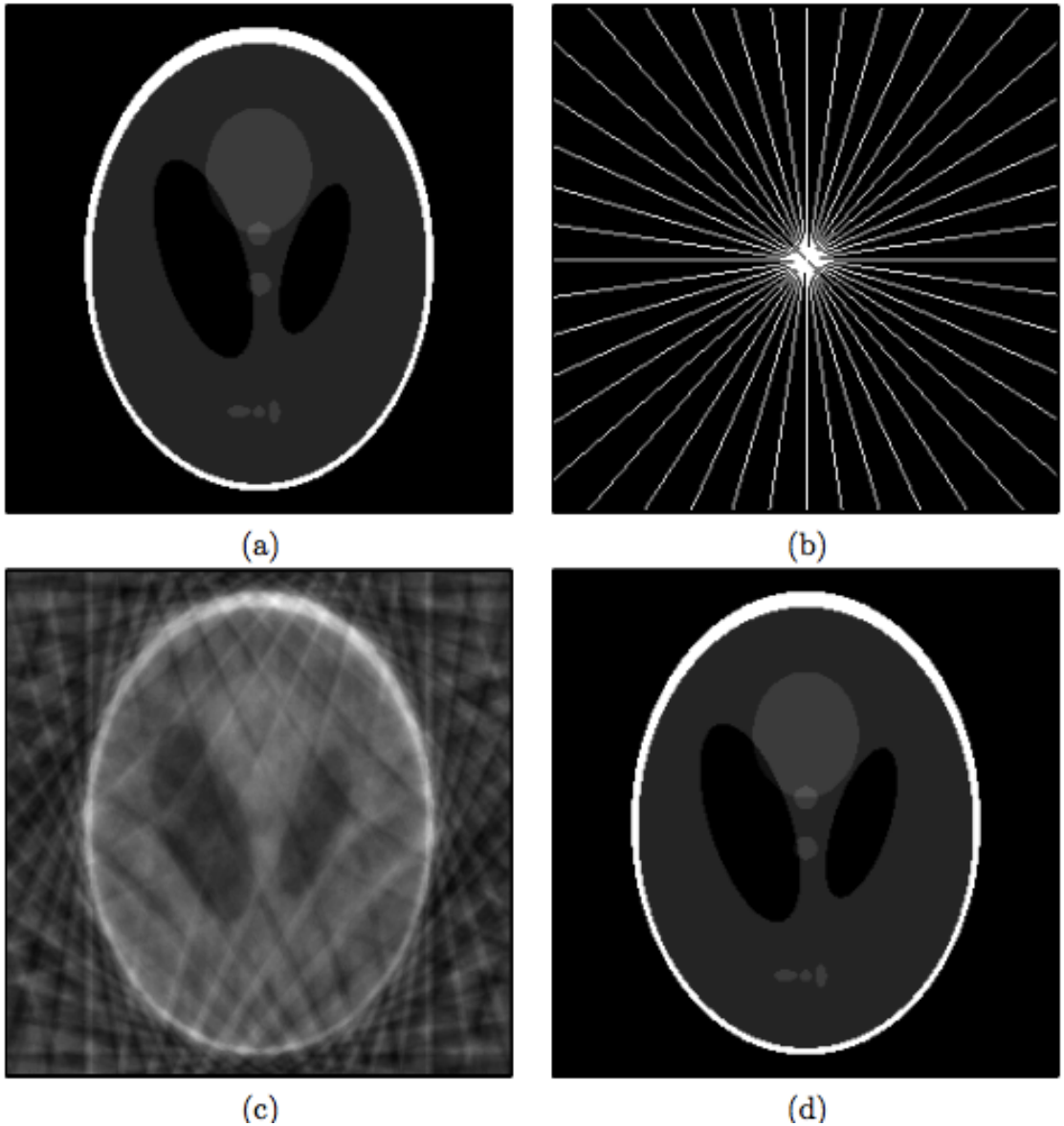
Wish to acquire a digital object $\mathbf{u} \in \mathbf{R}^n$ from m measurements

$$b_k = \langle \mathbf{u}, \mathbf{a}_k \rangle, k = 1, \cdots, m$$

- Few sensors
- Measurements are very expensive
- Measurements process is slow (MRI)
- ...

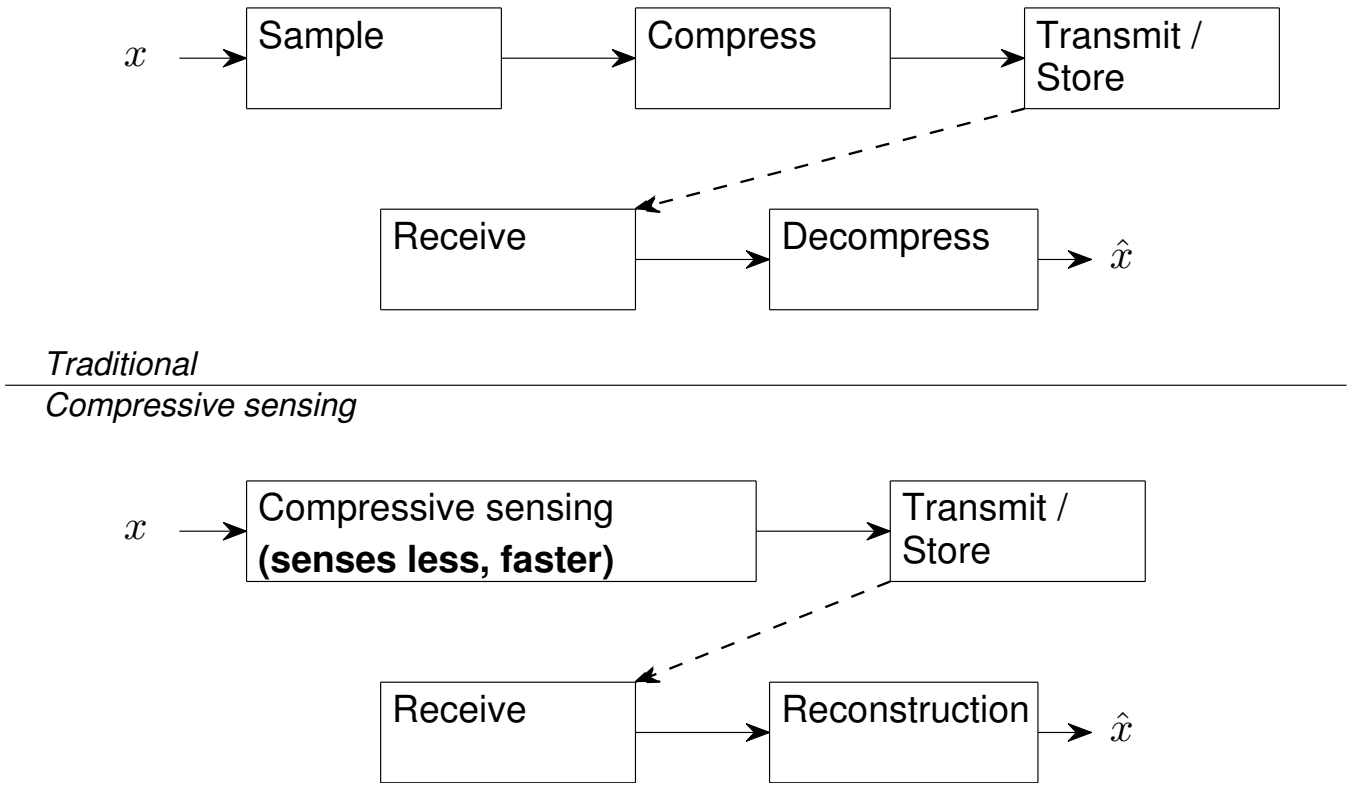
- Is this possible with $m << n$?
- Which measurements should we take?
- How should we reconstruct?

CS-MRI[1]



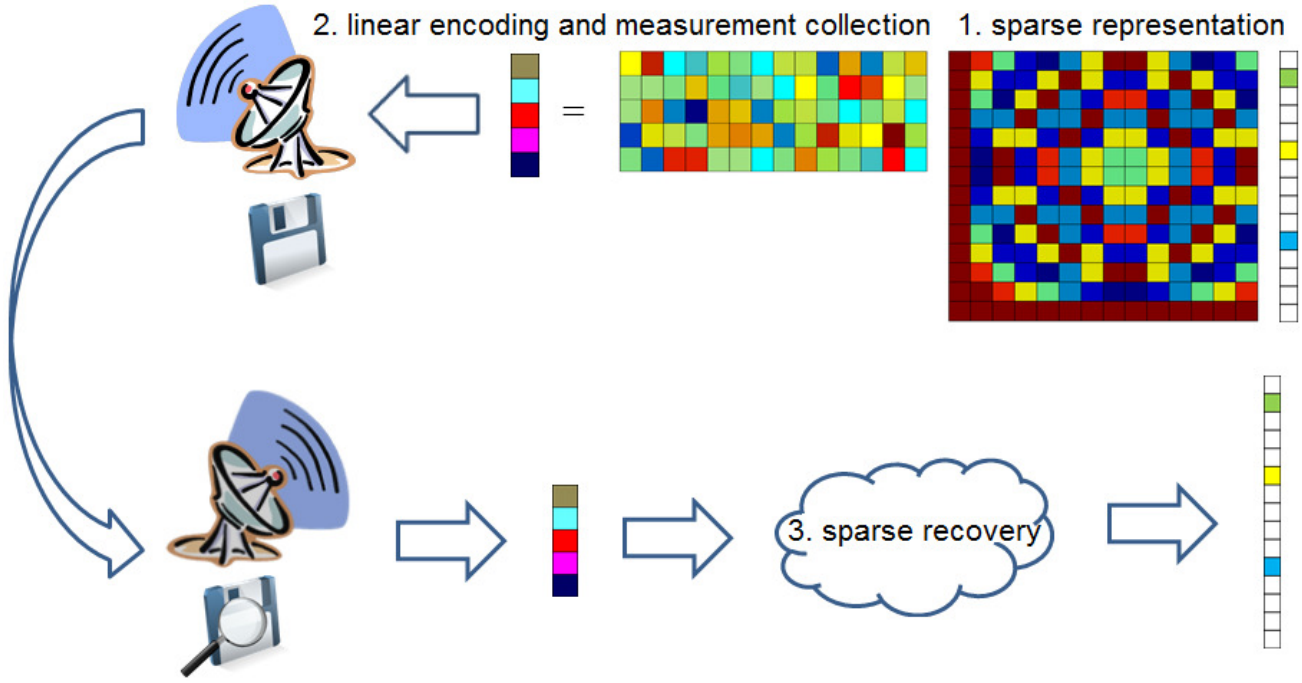
[1] Candes-Romberg-Tao 06'

Traditional sensing versus compressive sensing



The two sensing approaches have their own advantages, address different bottlenecks, fit different needs, and achieve different performance.

Scheme of compressive sensing



- Signal sparse representation
- Linear encoding and measurement collection
- Nonlinear decoding (Sparse recovery)

Outline

Sparse representation I

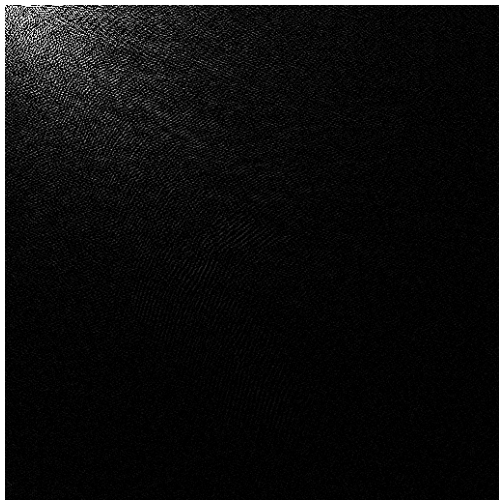
Sparse representation is the basis of CS.

- Express the information of a signal by a small number of real or complex numbers. Mathematically, this is to express a signal $\mathbf{u}^o$ as

$$\mathbf{u}^o = \sum_{i=1}^{p} \psi_i x_i^o,$$

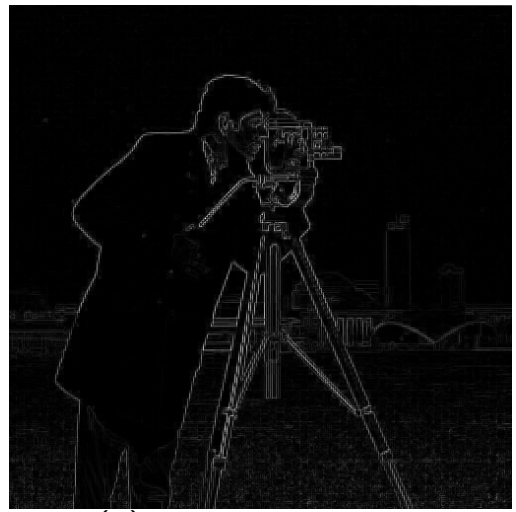
  where all but a small number of entries x_i^o are zero (or small enough to safely neglect). $\Psi = [\psi_1 \ \psi_2 \ \cdots \ \psi_p]$ is called a *dictionary*.
- Besides using a dictionary, a signal can also become sparse under a certain transform Υ, namely, $\Upsilon(\mathbf{u})$ is a sparse vector. Examples include the gradient operator, curvelet transforms, etc.

Sparse representation II

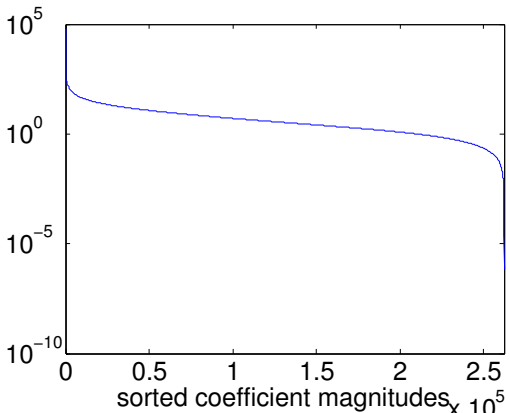


(a) DCT coefficients

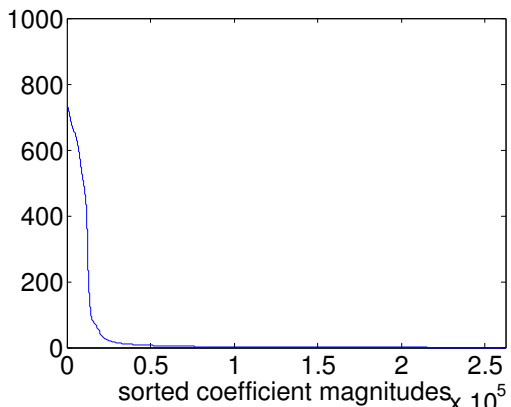
(b) Haar wavelet coefficients



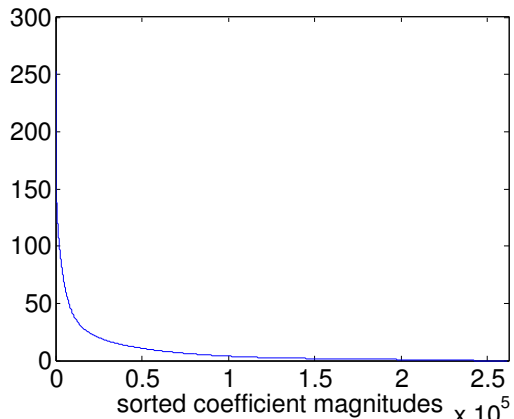
(c) Local variation



(d) DCT coeff's decay



(e) Haar wavelet coeff's decay



(f) Local variation decay

Figure: Sparsity of image Cameraman (the DCT and wavelet coefficients are scaled for better visibility).

Extensions to sparse models I

- **Joint sparse signals.** A set of signals $\mathbf{u}^{(i)}$, $i = 1, \ldots, L$, of the same dimension are jointly sparse if each of them is sparse and their non-zero entries are colocated at (roughly) the same coordinates, or so under dictionaries or transforms. Applications of the recovery of such signals **?** include multikernel machine learning, source localization, neuromagnetic imaging, and many more.

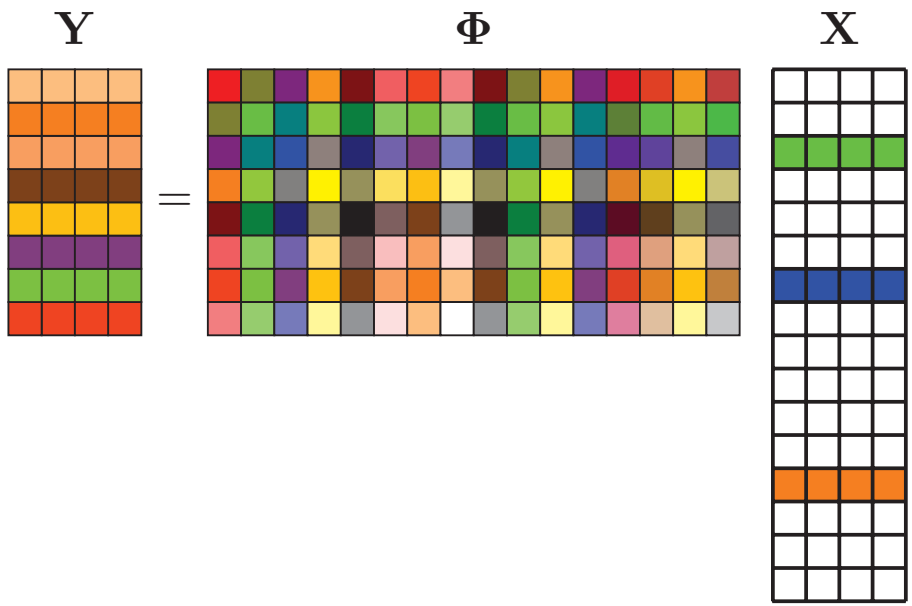


Figure: Joint sparse signal CS.

Extensions to sparse models II

- **Low-rank matrices.** A matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ of rank $r \ll \min\{m, n\}$ has mn entries but only $r(m + n - r)$ degrees of freedom (consider the singular value decomposition $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top$; $\mathbf{U}$, Σ, and $\mathbf{V}$ have $\sum_{i=1}^{r}(m - i)$, r, and $\sum_{i=1}^{r}(n - i)$ degrees of freedom, respectively, which sum to $r(m + n - r)$). Applications of low-rank matrix recovery include model reduction, recovering shape and motion from image streams, the Netflix recommendation problem, and more.

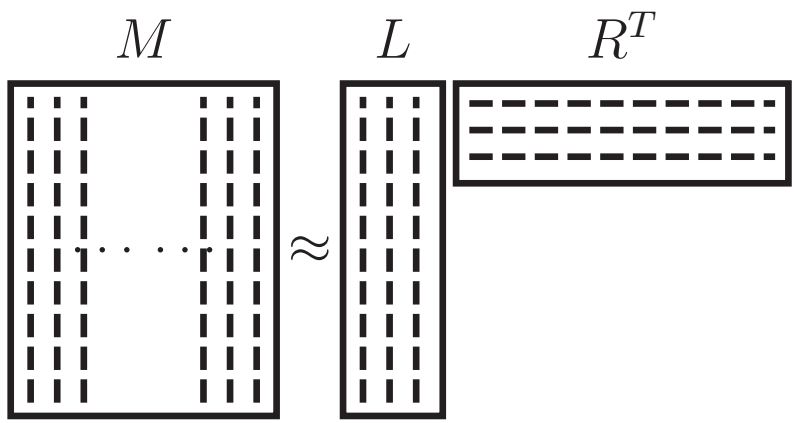


Figure: Approximation of a low-rank matrix.

Extensions to sparse models III

- **Unions of subspaces and model-based CS.** The support of a k-sparse vector is one of the $\binom{n}{k}$ possibilities, but for many signals in practice some or even most of these possibilities are not possible. For example, some transforms' coefficients follow a certain tree structure, some signals' non-zero entries tend to cluster, and some signals must lie in particular linear subspaces. These signals are easier to recover, and their structures can be generalized as the union of certain subspaces.

Outline

CS encoding and decoding

- In CS, the signal $\mathbf{u}^0 = \Psi \mathbf{x}^0$ is encoded to $\mathbf{b} = \mathbf{A}\mathbf{u}^0$. The recovery would be straightforward if $\mathbf{A}$ has full column rank, in which case $\mathbf{u}^0$ would be the unique solution of

$$\underset{\mathbf{u}}{\text{minimize}} \, \|\mathbf{b} - \mathbf{A}\mathbf{u}\|_2^2,$$

  or $\mathbf{u}^0 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$.

- However, CS uses fewer samples and $\mathbf{A}$ has fewer rows than columns. Such a matrix cannot have full column rank, and $\mathbf{b} = \mathbf{A}\mathbf{u}$ has multiple solution.

- What kind of matrix $\mathbf{A}$ allows the recovery of $\mathbf{u}^0$ (or a good approximate of $\mathbf{u}^0$) from $\mathbf{b} = \mathbf{A}\mathbf{u}$ given merely that $\mathbf{x}^0$ is sparse?

Sensing matrix design

Two questions in CS

- How should we design the sensing matrix $\mathbf{A}$ to ensure that it preserves the information in the signal $\mathbf{u}$?
- How can we recover the original signal $\mathbf{u}^0$ from measurements $\mathbf{b}$?

In the case where the data is **sparse** or **compressible**, we can design matrices $\mathbf{A}$ with much fewer rows than columns that ensure that we are able to recover the original signal accurately and efficiently.

Coherence I

- In $\mathbb{R}^n$ or $\mathbb{C}^n$, the coherence between the basis Φ, which has elements $\phi_1, \ldots, \phi_n$, and the basis Ψ, which has columns $\psi_1, \ldots, \psi_n$, is

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{1 \le i,j \le n} \frac{|\langle \phi_i, \psi_j \rangle|}{\|\phi_i\|_2 \|\psi_j\|_2}. \tag{1}$$

- The quantity $\mu(\Phi, \Psi)$ measures how small the closest angle between any two elements of Φ and Ψ can be. $1 \le \mu(\Phi, \Psi) \le \sqrt{n}$.
- Assume $\mathbf{Au}$ gives just m out of the n coefficients of $\Phi^T \mathbf{u}$. Then the question becomes what relation between two orthogonal bases Φ and Ψ allows the recovery of $\mathbf{x}^o$ (and thus $\mathbf{u}^o$) from incomplete coefficients of $\Phi^T \Psi \mathbf{x}^o$.
- With low coherence, every element of Φ is "misaligned" with every element of Ψ, so $\langle \phi_i, \psi_j \rangle$s are roughly equal and stay uniformly away from zero. Hence, each coefficient i of $\Phi^T \Psi \mathbf{x}^o$, which equals $\sum_{1 \le j \le n} \langle \phi_i, \psi_j \rangle x_j^o$, encodes a guaranteed amount of information of each x_j^o.
- In contrast, high coherence leads to uneven $\langle \phi_i, \psi_j \rangle$, and at least for some i, the coefficient i of $\Phi^T \Psi \mathbf{x}^o$ encodes just a few x_j^os while being nearly useless for the others.

Coherence II

Theorem (Candes-Romberg 06')

For a given $\mathbf{u}^o = \Psi \mathbf{x}^o$ where $\mathbf{x}^o$ has at most k non-zero entries, choose m entries of $\Phi^T \mathbf{u}^o$ uniformly at random, denoted as vector $\mathbf{b} = P_\Omega \Phi^T \mathbf{u}^o$, where P_Ω is the selection operator. As long as

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot (k \log n) \tag{2}$$

for some constant $C > 0$ independent of k and n, the solution to $\min\{\|\mathbf{x}\|_1 : \mathbf{b} = P_\Omega \Phi^T \Psi \mathbf{x}\}$ is $\mathbf{x}^o$ with overwhelming probability. (The result is shown for nearly all possible sign sequences of $\mathbf{x}^o$.)

- There are various results telling us when we can trust ℓ_1 minimization. We will consider the simple model with $\Psi = \mathbf{I}$.

Null space conditions I

- Null space of $\mathbf{A}$: $\mathcal{N}(\mathbf{A}) = \{\mathbf{z} : \mathbf{A}\mathbf{z} = \mathbf{0}\}$.
- For any pair of distinct vectors $\mathbf{x}, \mathbf{x}' \in \Sigma_K = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq K\}$, we must have $\mathbf{A}\mathbf{x} \neq \mathbf{A}\mathbf{x}'$. Otherwise, it would be impossible to distinguish $\mathbf{x}$ and $\mathbf{x}'$ based solely on the measurements $\mathbf{b}$.
- If $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}'$, then $\mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0}$ with $\mathbf{x} - \mathbf{x}' \in \Sigma_{2K}$.
- The **spark** of a given matrix $\mathbf{A}$ is the smallest number of columns of $\mathbf{A}$ that are linearly dependent.
- **Theorem:** [2] For any vector $\mathbf{x} \in \mathbf{R}^m$, there exists at most one signal $\mathbf{x} \in \Sigma_K$ such that $\mathbf{b} = \mathbf{A}\mathbf{x}$ if and only if spark$(\mathbf{A}) > 2K$.

[2] Donoho and Elad, 03'

Null space conditions II

- The spark provides a complete characterization of when sparse recovery is possible for dealing with **exactly** sparse vectors. However, in order to deal with **approximately sparse** signals, we have to introduce somewhat more restrictive conditions on the null space of $\mathbf{A}$. We must also ensure that $\mathcal{N}(\mathbf{A})$ does not contain any vectors that are too compressible in addition to vectors that are sparse.
- A matrix $\mathbf{A}$ satisfies the **null space property (NSP)** of order K if there exist a constant $C > 0$ such that

$$\|\mathbf{h}_\Lambda\|_2 \leq C \frac{\|\mathbf{h}_{\Lambda^c}\|_1}{\sqrt{K}}$$

  holds for all $\mathbf{h} \in \mathcal{N}(\mathbf{A})$ and for all Λ such that $|\Lambda| \leq K$.
- The NSP quantifies the notion that vectors in the null space of $\mathbf{A}$ should not be too concentrated on a small subset of indices. If a vector $\mathbf{h}$ is exactly $K-$sparse, then there exist a Λ such that $\|\mathbf{h}_{\Lambda^c}\|_1 = 0$ and hence $\mathbf{h}_\Lambda = \mathbf{0}$ as well. Thus if a matrix $\mathbf{A}$ satisfies the NSP then the only $K-$sparse vector in $\mathcal{N}(\mathbf{A})$ is $\mathbf{h} = \mathbf{0}$.

Null space condition III

- Let $\triangle : \mathbf{R}^m \to \mathbf{R}^n$ represent a specific recovery methods, we will focus on the guarantees of the form

$$\|\triangle(\mathbf{A}\mathbf{x}) - \mathbf{x}\|_2 \leq C\frac{\sigma_K(\mathbf{x})_1}{\sqrt{K}} \tag{3}$$

  for all $\mathbf{x}$, we $\sigma_K(\mathbf{x})_p = \min_{\hat{\mathbf{x}} \in \Sigma_K} \|\mathbf{x} - \hat{\mathbf{x}}\|_p$.

- Let $\mathbf{A} : \mathbf{R}^n \to \mathbf{R}^m$ denote a sensing matrix and $\triangle : \mathbf{R}^m \to \mathbf{R}^n$ denote an arbitrary recovery algorithm. If the pair $(\mathbf{A}, \triangle)$ satisfies (3), then $\mathbf{A}$ satisfies the NSP of order $2K$.

Restricted isometry property I

- The NSP is both necessary and sufficient for establishing recovery guarantees of the form

$$\|\triangle(\mathbf{Ax}) - \mathbf{x}\|_2] \leq C\frac{\sigma_K(\mathbf{x})_1}{\sqrt{K}},$$

  but these guarantees do not account for **noise**. When the measurements are contaminated with noise or have been corrupted by some error such as quantization, it will be useful to consider somewhat stronger conditions.

- A matrix $\mathbf{A}$ satisfies the **restricted isometry property (RIP)** of order K if these exists a $\delta_K \in (0, 1)$ such that

$$(1 - \delta_K)\|\mathbf{u}\|_2^2 \leq \|\mathbf{Au}\|_2^2 \leq (1 + \delta_K)\|\mathbf{u}\|_2^2,$$

  holds for all $\mathbf{u} \in \Sigma_K$.

- If a matrix $\mathbf{A}$ satisfies the RIP of order $2K$, then we can say $\mathbf{A}$ approximately preserves the distance between any pair of $K-$sparse vectors.

Restricted isometry property II

- Let $\mathbf{A} : \mathbf{R}^n \to \mathbf{R}^m$ denote a sensing matrix and $\triangle : \mathbf{R}^m \to \mathbf{R}^n$ be a recovery algorithm. We say that the pair $(\mathbf{A}, \triangle)$ is **C-stable** if for any $\mathbf{u} \in \Sigma_K$ and any $\mathbf{e} \in \mathbf{R}^m$ we have that

$$\|\triangle(\mathbf{Au} + \mathbf{e}) - \mathbf{u}\|_2 \leq C\|\mathbf{e}\|_2.$$

- If the pair $(\mathbf{A}, \triangle)$ is C-stable, then

$$\frac{1}{C}\|\mathbf{x}\|_2 \leq \|\mathbf{Ax}\|_2 \tag{4}$$

  for all $\mathbf{x} \in \Sigma_{2K}$.

- If $C \to 1$, we have that $\mathbf{A}$ must satisfy the lower bound of RIP condition with $\delta_{2K} = 1 - \frac{1}{C^2} \to 0$.

RIP and NSP

- **The RIP is strictly stronger than the NSP:** If $\mathbf{A}$ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$, then $\mathbf{A}$ satisfies the NSP of order $2K$ with constant

$$C = \frac{\sqrt{2}\delta_{2K}}{1 - (1 + \sqrt{2})\delta_{2K}}.$$

Matrix that satisfy the RIP

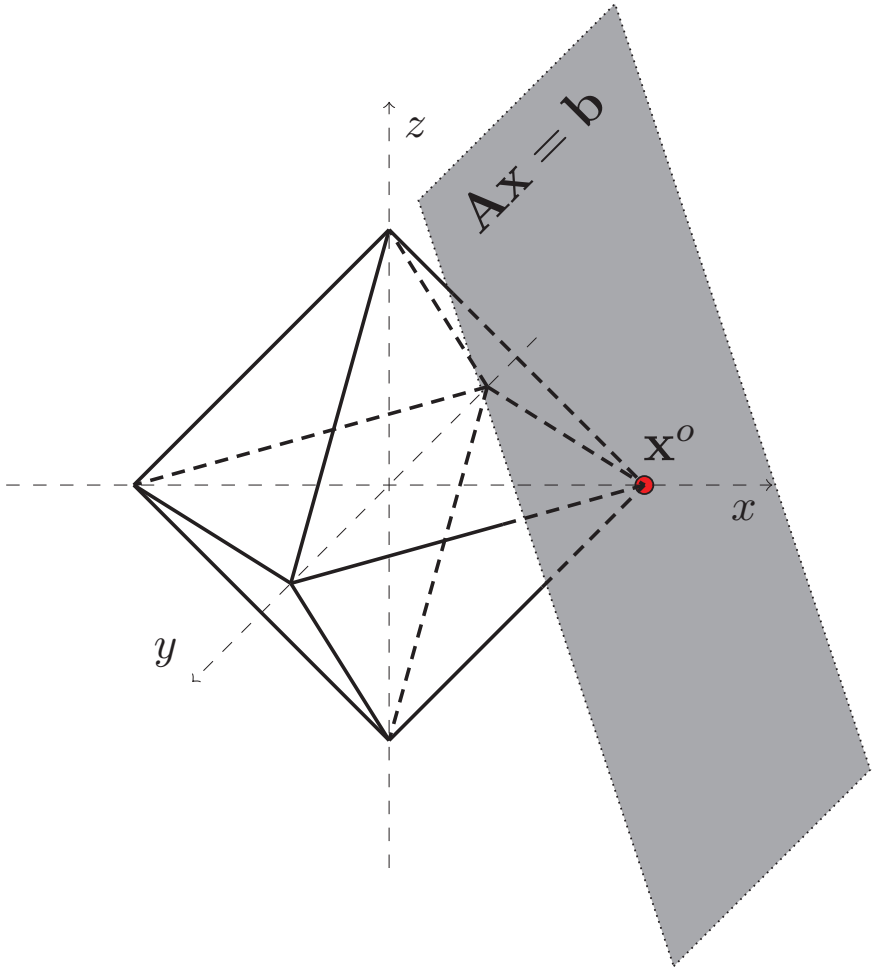
- It is possible to deterministically construct matrices of size $m \times n$ that satisfy the RIP of order K, but such constructions also require m to be relativey large. [3]
- Fortunately these limitations can be overcome by randomizing the matrix construction.
- It is difficult to verify the conditions.
- We can still use CS even if the conditions are not satisfied: The conditions are uniform condition, and sufficient condition.

[3] DeVore 07', Ubdyk 08'

Outline

Basis pursuit

$$\underset{\mathbf{x}}{\text{minimize}}\{\|\mathbf{x}\|_1 : \mathbf{A}\mathbf{x} = \mathbf{b}\}$$



Basis pursuit denoising and LASSO

$$\underset{\mathbf{x}}{\text{minimize}}\{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 : \|\mathbf{x}\|_1 \leq \tau\}, \tag{5a}$$

$$\underset{\mathbf{x}}{\text{minimize}}\, \|\mathbf{x}\|_1 + \frac{\mu}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2, \tag{5b}$$

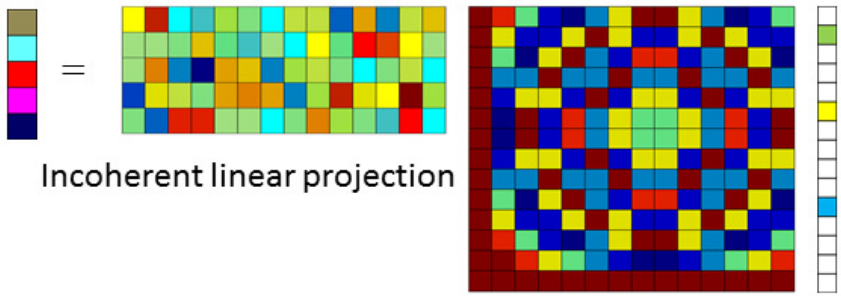
$$\underset{\mathbf{x}}{\text{minimize}}\{\|\mathbf{x}\|_1 : \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 \leq \sigma\}. \tag{5c}$$

Questions:

1. Are they equivalent? in what sense?
   - Solution can be non-unique. Why?
   - A solution to one of them is also the solution to the other two with appropriate parameters?
   - Solution sets $\mathcal{X}_\tau = \mathcal{X}_\mu = \mathcal{X}_\sigma$?
2. How to choose parameters?
   - τ, μ, and σ have different meanings.
   - Applications determine which one is easier to set.
   - Use a test data set, then scale parameters for other data.
   - Cross validation

Sparse under basis Ψ

$$\underset{\mathbf{s}}{\text{minimize}}\{\|\mathbf{s}\|_1 : \mathbf{A}\Psi\mathbf{s} = \mathbf{b}\} \tag{6}$$



If Ψ is orthogonal, problem (6) is equivalent to

$$\underset{\mathbf{x}}{\text{minimize}}\{\|\Psi^*\mathbf{x}\|_1 : \mathbf{A}\mathbf{x} = \mathbf{b}\}. \tag{7}$$

Also,

$$\underset{\mathbf{x}}{\text{minimize}}\{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 : \|\Psi^*\mathbf{x}\|_1 \leq \tau\},$$

$$\underset{\mathbf{x}}{\text{minimize}}\, \|\Psi^*\mathbf{x}\|_1 + \frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2,$$

$$\underset{\mathbf{x}}{\text{minimize}}\{\|\Psi^*\mathbf{x}\|_1 : \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \leq \sigma\}.$$

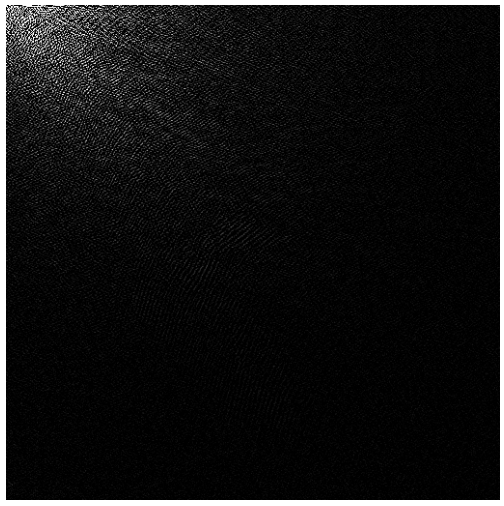
Sparse after transform $\mathcal{L}$

$$\underset{\mathbf{x}}{\text{minimize}}\{\|\mathcal{L}\mathbf{x}\|_1 : \mathbf{A}\mathbf{x} = \mathbf{b}\} \tag{8}$$

Examples of $\mathcal{L}$:

- DCT, wavelets, curvelets, ridgelets, ...
- tight frames, Gabor, ...
- (weighted) total variation

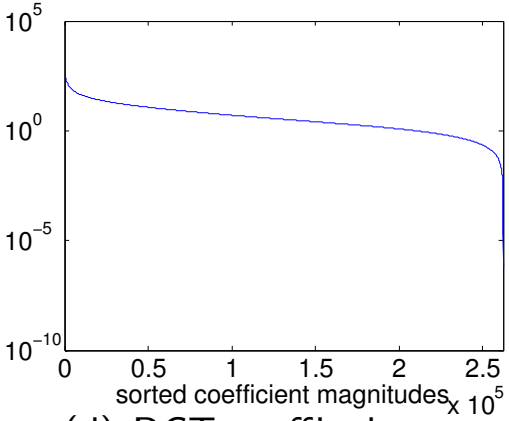
See: E. J. Candès, Y. Eldar, D. Needell and P. Randall. Compressed sensing with coherent and redundant dictionaries. Applied and Computational Harmonic Analysis 31(1), 59–73. ($\mathcal{L}$-RIP $\Rightarrow$ stable recovery of $\mathcal{L}\mathbf{x}$)



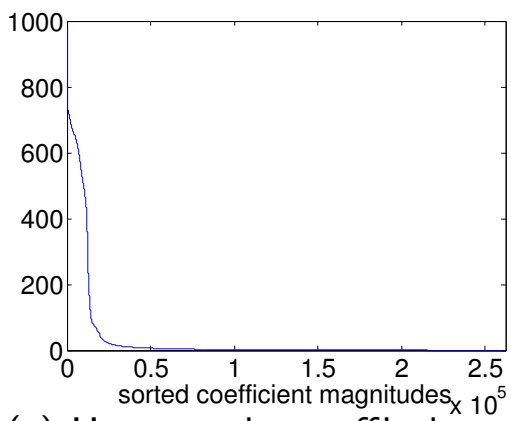
(a) DCT coefficients

(b) Haar wavelet coefficients

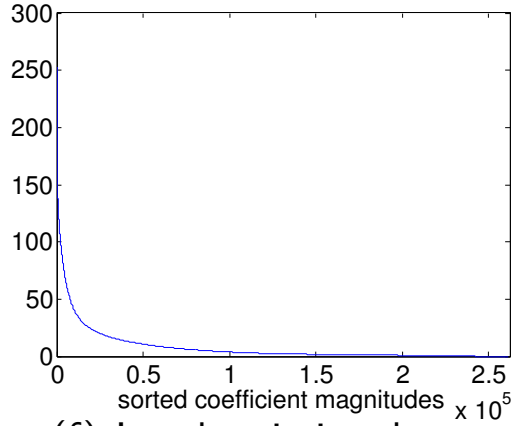
(c) Local variation



(d) DCT coeff's decay



(e) Haar wavelet coeff's decay



(f) Local variation decay

Figure: Sparsity of image Cameraman (the DCT and wavelet coefficients are scaled for better visibility).

Joint/group sparsity

Joint sparse recovery model:

$$\underset{\mathbf{X}}{\text{minimize}}\{\|\mathbf{X}\|_{2,1} : \mathbf{AX} = \mathbf{b}\} \tag{9}$$

where

$$\|\mathbf{X}\|_{2,1} := \sum_{i=1}^{m} \left\|\begin{bmatrix} x_{i1} & x_{i,2} \cdots x_{in} \end{bmatrix}\right\|_2 .$$

also $\|\mathbf{X}\|_{p,1}$ for $p > 1$. Complex-valued signals are a special case.

Joint/group sparsity

Decompose $\{1, \ldots, n\} = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \cdots \cup \mathcal{G}_S$.

- Non-overlapping groups: $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset, \ \forall i \neq j$.
- Otherwise, groups may overlap (what kind of structure can be modeled?).

Group-sparse recovery model:

$$\underset{\mathbf{x}}{\text{minimize}} \{\|\mathbf{x}\|_{\mathcal{G},2,1} : \mathbf{A}\mathbf{x} = \mathbf{b}\} \tag{10}$$

where

$$\|\mathbf{x}\|_{\mathcal{G},2,1} = \sum_{s=1}^{S} w_s \|\mathbf{x}_{\mathcal{G}_s}\|_2.$$

Side constraints

- Nonnegativity: $\mathbf{x} \geq \mathbf{0}$
- Bound (box) constraints: $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$
- General inequalities: $\mathbf{Q}\mathbf{x} \leq \mathbf{q}$

They can be very effective in practice. They also generate "corners".

Outline

Prox-linear algorithm

Consider the general form

$$\underset{\mathbf{x}}{\text{minimize}} \;\; r(\mathbf{x}) + f(\mathbf{x}),$$

where r is the regularization function and f is the data fidelity function. The prox-linear algorithm is:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \;\; r(\mathbf{x}) + f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{1}{2\delta_k} \|\mathbf{x} - \mathbf{x}^k\|_2^2.$$

The last term keeps $\mathbf{x}^{k+1}$ close to $\mathbf{x}^k$, and the parameter δ_k determines the step size. It is equivalent to

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \;\; r(\mathbf{x}) + \frac{1}{2\delta_k} \left\| \mathbf{x} - \left(\mathbf{x}^k - \delta_k \nabla f(\mathbf{x}^k) \right) \right\|_2^2 .$$

Shrinkage operation I

- The problem

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{1}{2\tau}\|\mathbf{x}-\mathbf{z}\|_2^2, \tag{11}$$

where $\tau > 0$, is equivalent to solving $\min_{x_i} |x_i| + \frac{1}{2\tau}|x_i - z_i|^2$ over each i. And one can obtain the closed-form solution

$$(x_{\text{opt}})_i = \begin{cases} z_i - \tau, & z_i > \tau \\ 0, & -\tau \le z_i \le \tau, \\ z_i + \tau, & z_i < -\tau. \end{cases} \tag{12}$$

Shrinkage operation II

- The solution is illustrated in the figure below.

$$\mathbf{x}_{\text{opt}} = \text{shrink}(\mathbf{z}, \tau). \tag{13}$$

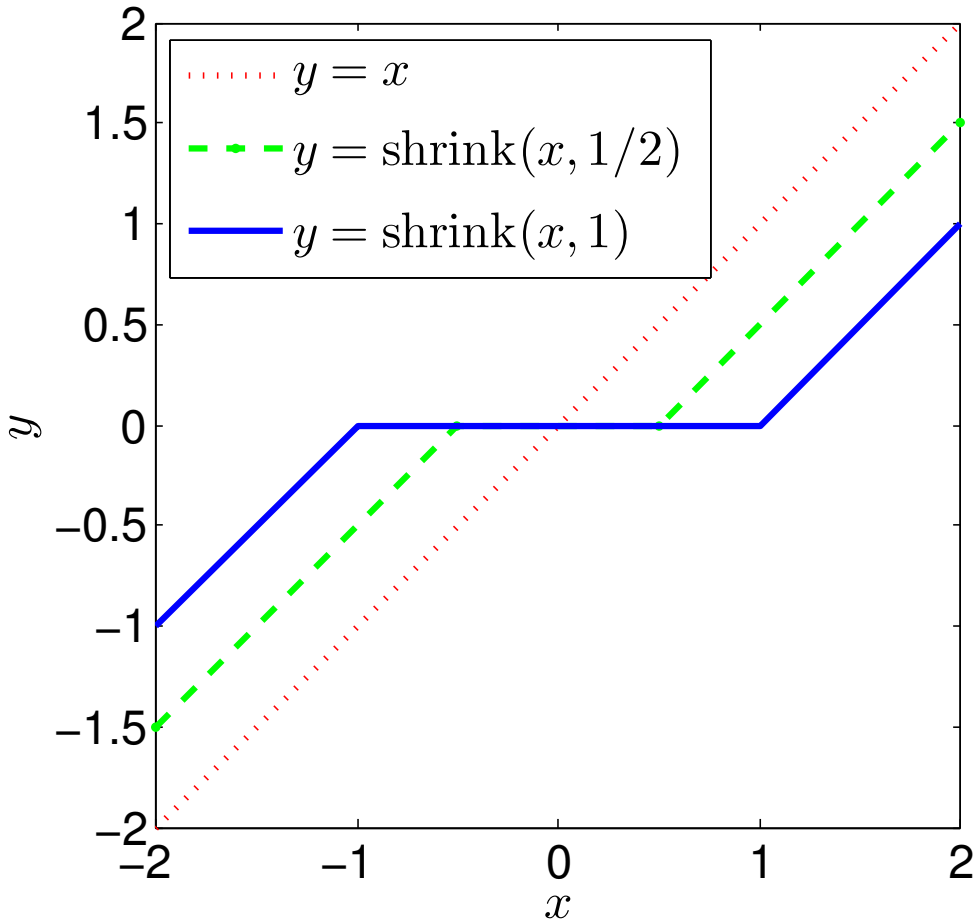


Figure: Illustration of $\text{shrink}(x, \tau)$.

Basis pursuit denoising

Let $r(\mathbf{x}) = \|\mathbf{x}\|_1$ and $f(\mathbf{x})$ be a differentiable function (e.g., $\frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$). The backward step is a shrinkage. Hence, it becomes

$$\mathbf{x}^{k+1} = \text{shrink}(\mathbf{x}^k - \delta_k \nabla f(\mathbf{x}^k), \delta_k^{-1}).$$

The main computation at each iteration k is $\nabla f(\mathbf{x}^k)$ (e.g., $\mu \mathbf{A}^T \mathbf{A}\mathbf{x}^k$). If we generalize to $r(\mathbf{x}) = \|\Psi\mathbf{x}\|_1$ for an orthogonal linear transform, then it is given by

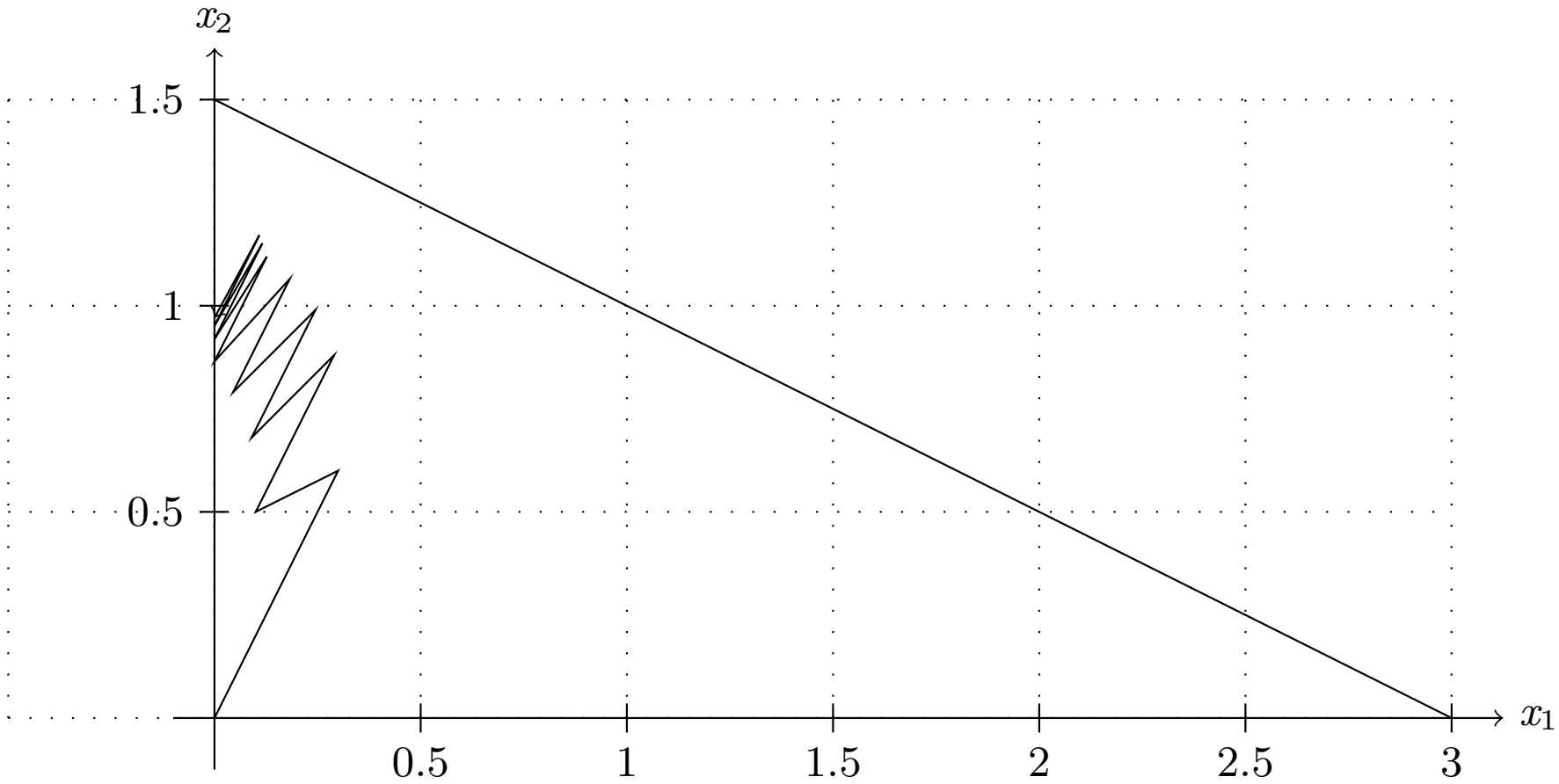
$$\mathbf{x}^{k+1} = \Psi^T \text{shrink}(\Psi(\mathbf{x}^k - \delta_k \nabla f(\mathbf{x}^k)), \delta_k^{-1}).$$

Example

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{1}{4}(x_1 + 2x_2 - 3)^2$$

Optimal solution: $\mathbf{x}^* = [0, 1]$. Let $\delta_k = 1/5$, $\mathbf{x}^0 = [0, 0]$

Forward step	Backward step
[0.3,0.6]	[0.1,0.5]
[0.29,0.88]	[0.09, 0.68]
[0.245,0.99]	[0.045,0.79]
[0.1825,1.065]	[0,0.865]
[0.127,1.119]	[0,0.919]
[0.1162,1.1514]	[0,0.9514]
[0.10972,1.17084]	[0,0.97084]
⋮	⋮


x_2
1.5
1
0.5
x_1
0.5
1
1.5
2
2.5
3

Other variants

- Accelerated prox-linear algorithms: FISTA etc.
- High-order pros-linear algorithms.

Outline

Dual (sub)gradient ascent

Primal problem

$$\underset{\mathbf{x}}{\text{minimize}}\, f(\mathbf{x}), \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.$$

The Lagrangian dual is a maximization problem

$$\underset{\mathbf{y}}{\text{maximize}} \;\; g(\mathbf{y})$$

If g is *differentiable*, you can apply

$$\mathbf{y}^{k+1} \leftarrow \mathbf{y}^k + \alpha^k \nabla g(\mathbf{y}^k).$$

Derive ∇g by hand, or from the definition of Lagrangian $\mathcal{L}(\mathbf{x}; \mathbf{y})$:

If we let $\bar{\mathbf{x}} \leftarrow \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \mathbf{y}^k)$, then

$$\nabla g(\mathbf{y}^k) = \mathbf{A}\bar{\mathbf{x}} - \mathbf{b}.$$

Dual (sub)gradient ascent

Iteration:

$$\mathbf{x}^{k+1} \leftarrow \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \mathbf{y}^k),$$
$$\mathbf{y}^{k+1} \leftarrow \mathbf{y}^k + \alpha^k(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}).$$

Works but need properties of ∇g and in turn properties of f (e.g., strict convexity). *Example*: linearized Bregman.

Also, there are dual subgradient ascent methods.

Augmented Lagrangian (*a.k.a.* Method of Multipliers)

Augment $\mathcal{L}(\mathbf{x}; \mathbf{y}^k) = r(\mathbf{x}) - (\mathbf{y}^k)^T(\mathbf{A}\mathbf{x} - \mathbf{b})$ by adding $\frac{\delta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$.

Iteration:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} r(\mathbf{x}) - (\mathbf{y}^k)^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \frac{\delta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2,$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \delta(\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}).$$

from $k = 0$ and $\mathbf{y}^0 = \mathbf{0}$. $\delta > 0$ can change in k.

The objective of the first step is convex in $\mathbf{x}$ (if $r(\cdot)$ is convex) and linear in $\mathbf{y}$.

Augmented Lagrangian (*a.k.a.* Method of Multipliers)

Recall KKT conditions

$$\begin{aligned} \text{(primal feasibility)} \quad & \mathbf{0} = \mathbf{A}\mathbf{x}_{\text{opt}} - \mathbf{b}, \\ \text{(dual feasibility)} \quad & \mathbf{0} \in \partial r(\mathbf{x}_{\text{opt}}) - \mathbf{A}^T \mathbf{y}_{\text{opt}}. \end{aligned}$$

Compare with

$$\mathbf{0} \in \partial r(\mathbf{x}^{k+1}) - \mathbf{A}^T(\mathbf{y}^k + \delta\left(\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}\right)) = \partial r(\mathbf{x}^{k+1}) - \mathbf{A}^T\mathbf{y}^{k+1}.$$

Dual feasibility is maintained for $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})$!

Also, it works toward primal feasibility:

$$-(\mathbf{y}^k)^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \frac{\delta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \frac{\delta}{2}\langle \mathbf{A}\mathbf{x} - \mathbf{b}, \sum_{i=1}^{k}(\mathbf{A}\mathbf{x}^i - \mathbf{b}) + (\mathbf{A}\mathbf{x} - \mathbf{b})\rangle.$$

Keep adding penalty to the violation of $\mathbf{A}\mathbf{x} = \mathbf{b}$, achieving it in the limit (for polyhedral $r(\cdot)$ in finitely many steps).

Augmented Lagrangian (*a.k.a.* Method of Multipliers)

BTW, the iteration is equivalent to *proximal dual ascent*

$$\mathbf{y}^{k+1} \leftarrow \arg\max_{\mathbf{y}} g(\mathbf{y}) - \frac{1}{2\delta}\|\mathbf{y} - \mathbf{y}^k\|_2^2.$$

Compared to dual gradient ascent

- Pros: converges for nonsmooth and extended-value f (thanks to the proximal term)
- Cons:
  - If f is nice and dual ascent works, it may be slower than dual ascent (especially one with line search, 2nd-order ascent, e.g.)
  - The term $\frac{1}{2\delta}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ in the $\mathbf{x}$-subproblem prevents splitting (unless $\mathbf{A}$ has a block-diagonal structure)

Alternating direction method of multipliers (ADMM)

Start with separable formulation

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & f(\mathbf{x}) + g(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{b}. \end{aligned}$$

f and g are **convex**, maybe **nonsmooth**, can **include constraints**

Basic ADMM iteration

1. $\mathbf{x}^{k+1} \leftarrow \min f(\mathbf{x}) + \cancel{g(\mathbf{y}^k)} + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k\|_2^2$,
2. $\mathbf{y}^{k+1} \leftarrow \min \cancel{f(\mathbf{x}^{k+1})} + g(\mathbf{y}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y} - \mathbf{b} - \mathbf{z}^k\|_2^2$,
3. $\mathbf{z}^{k+1} \leftarrow \mathbf{z}^k - (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{b})$.

Dates back to Douglas, Peaceman, and Rachford (50s–70s, operator splitting for PDEs); Glowinsky et al.'80s, Gabay'83; Spingarn'85; Eckstein and Bertsekes'92, He et al.'02 in variational inequality.

Lost favor for nonlinear programming around 1990–2004.

Alternating direction method of multipliers (ADMM)

- Now, $\mathbf{z}$ is the *scaled* dual variable (i.e., $\mathbf{z} = \beta\lambda$, where λ is the Lagrange multipliers)
- At each iteration, apply Gauss-Seidel to update $\mathbf{x}$ and then $\mathbf{y}$
- If $\mathbf{x}$ and $\mathbf{y}$ are minimized jointly, it reduces to the augmented Lagrangian method
- Can be extended to multiple blocks (some questions remain open)
- Can be extended to Jacobian (parallel) updates of $\mathbf{x}$ and $\mathbf{y}$ (dampen the update of $\mathbf{z}$)
- Can be extended to *inexact* updates of $\mathbf{x}$ and $\mathbf{y}$ (dampen the update of $\mathbf{z}$)
- If f and $\mathbf{x}$ are separable and $\mathbf{A} = I$, $\mathbf{x}$-update is decomposable

Bregman Methods

Three different versions:

- (original) Bregman = Generalized proximal point = Residual addback = augmented Lagrangian method
- linearized Bregman = smoothing and dual ascent
- split Bregman $\approx$ alternating direction of multipliers

Bregman iterations update (sub)gradients, instead of Lagrange multipliers

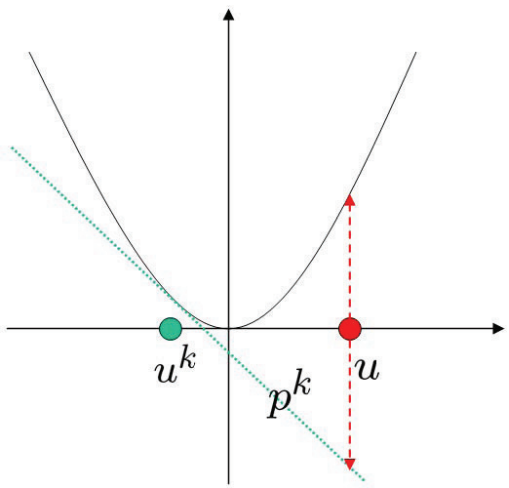
Bregman distance

Definition: let r be a convex function

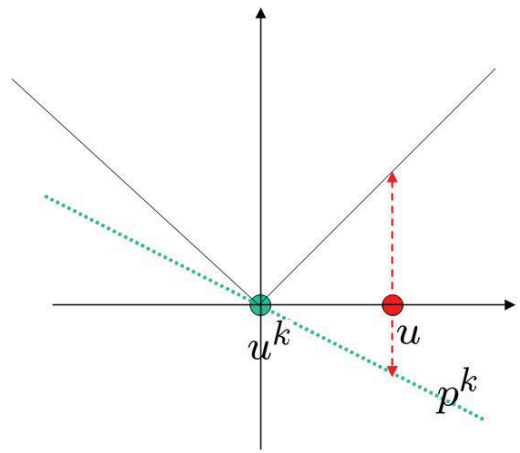
$$D_r(\mathbf{x}, \mathbf{y}; \mathbf{p}) = r(\mathbf{x}) - r(\mathbf{y}) - \langle \mathbf{p}, \mathbf{x} - \mathbf{y} \rangle, \text{ where } \mathbf{p} \in \partial r(\mathbf{y}).$$

Not a distance but has its flavor.

Examples: $D_{\ell_2^2}(u, u^k; p^k)$ versus $D_{\ell_1}(u, u^k; p^k)$



differentiable case



non-differentiable case

Bregman algorithm

Iteration

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \; D_r(\mathbf{x}, \mathbf{x}^k; \mathbf{p}^k) + f(\mathbf{x}), \tag{14a}$$

$$\mathbf{p}^{k+1} = \mathbf{p}^k - \nabla f(\mathbf{x}^{k+1}), \tag{14b}$$

starting at $k = 0$ and $(\mathbf{x}^0, \mathbf{p}^0) = (\mathbf{0}, \mathbf{0})$. The update of $\mathbf{p}$ follows from

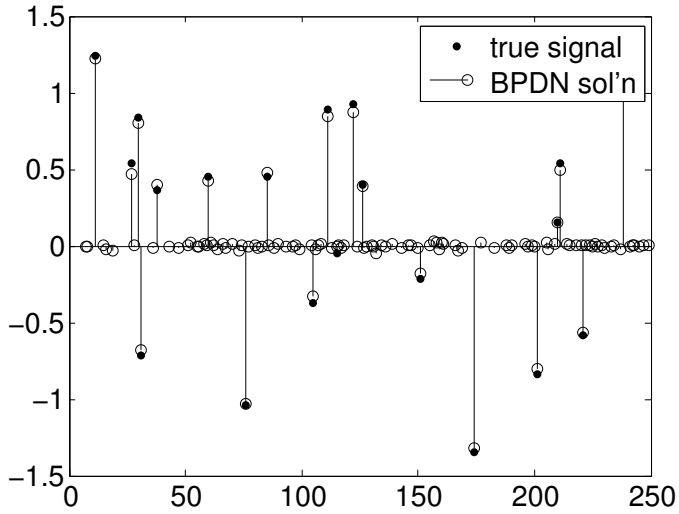
$$\mathbf{0} \in \partial r(\mathbf{x}^{k+1}) - \mathbf{p}^k + \nabla f(\mathbf{x}^{k+1}), \tag{15}$$

so the Bregman distance $D_r(\mathbf{x}, \mathbf{x}^{k+1}; \mathbf{p}^{k+1})$ is well defined.

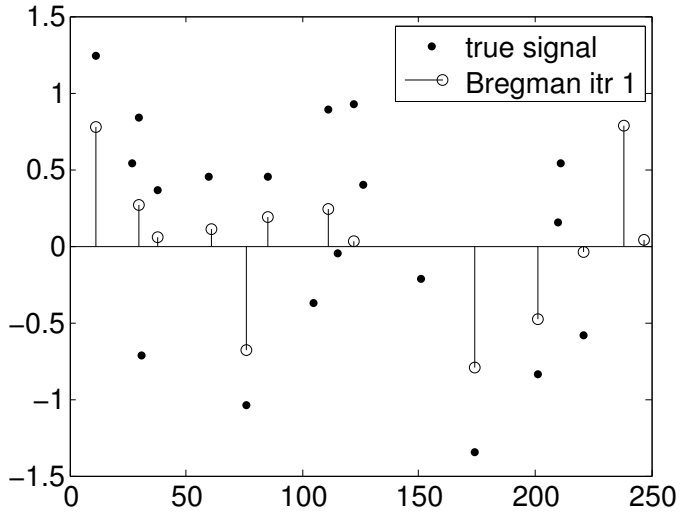
Interestingly, Bregman iteration has three another interpretations

1. Proximal point iteration
2. Residual addback iteration
3. Augmented Lagrangian iteration

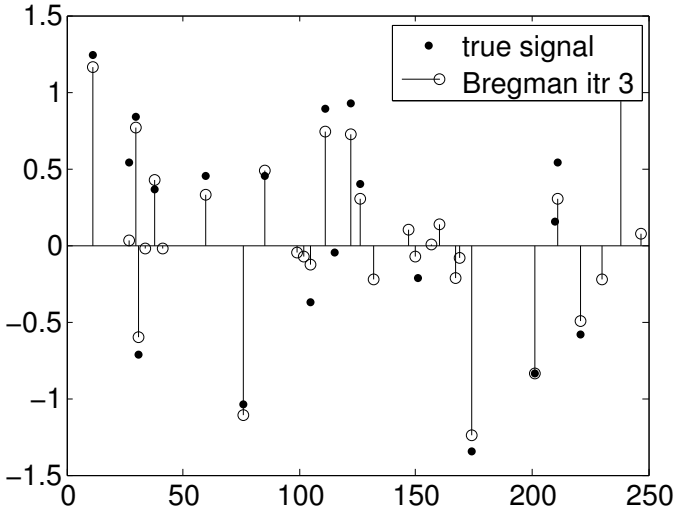
Bregman iterations and denoising



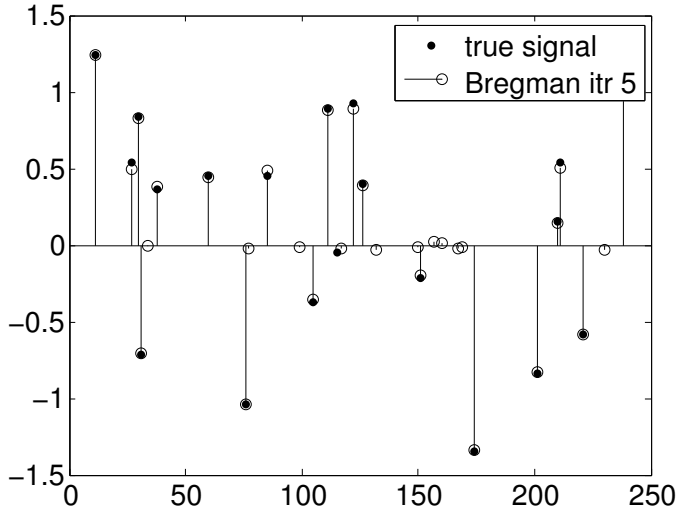
(a) BPDN with a true $\sigma = \|\mathbf{n}\|_2$



(b) Bregman, Iteration 1



(c) Bregman, Iteration 3



(d) Bregman, Iteration 5

Figure: BPDN versus Bregman iteration.

Linearized Bregman

We can consolidate $D_r(\mathbf{x}, \mathbf{x}^k; \mathbf{p}^k)$ and $\frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}^k\|_2^2$ as follows:

Introduce

$$\bar{r}(\mathbf{x}) := r(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x}\|_2^2$$

and

$$D_{\bar{r}}(\mathbf{x}, \mathbf{x}^k; \bar{\mathbf{p}}^k) = D_r(\mathbf{x}, \mathbf{x}^k; \mathbf{p}^k) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}^k\|_2^2.$$

We can rewrite the iteration as

$$\begin{aligned}
\mathbf{x}^{k+1} &\leftarrow \arg\min_{\mathbf{x}} \; D_{\bar{r}}(\mathbf{x}, \mathbf{x}^k; \bar{\mathbf{p}}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} \rangle, \\
\bar{\mathbf{p}}^{k+1} &\leftarrow \bar{\mathbf{p}}^k - \nabla f(\mathbf{x}^k).
\end{aligned}$$

Since $\bar{r}$ is strongly convex, the iterations work without any proximal terms.

Linearized Bregman and Dual Gradient Ascent

Consider $f(\mathbf{x}) = \frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$.

Linearized Bregman iteration:

$$\begin{aligned}
\mathbf{x}^{k+1} &\leftarrow \arg\min_{\mathbf{x}} \ \bar{r}(\mathbf{x}) - (\bar{\mathbf{p}}^k)^T\mathbf{x} + \mu\langle \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b}), \mathbf{x}\rangle, \\
\bar{\mathbf{p}}^{k+1} &\leftarrow \bar{\mathbf{p}}^k - \mu\mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b}).
\end{aligned}$$

Dual gradient ascent iteration:

$$\begin{aligned}
\mathbf{x}^{k+1} &\leftarrow \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \bar{\mathbf{y}}^k) = \bar{r}(\mathbf{x}) - (\bar{\mathbf{y}}^k)^T(\mathbf{A}\mathbf{x} - \mathbf{b}), \\
\bar{\mathbf{y}}^{k+1} &\leftarrow \bar{\mathbf{y}}^k + \tau(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}).
\end{aligned}$$

The two iterations are equivalent under $\mu = \tau$ and $\bar{\mathbf{p}}^k = -\mathbf{A}^T\bar{\mathbf{y}}^{k-1}$.

So, linearized Bregman is dual ascent applied to

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{minimize}} \ \ \bar{r}(\mathbf{x}) = r(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x}\|_2^2 \\
&\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.
\end{aligned}$$

Exact Regularization / Smoothing Effect

- In general, the *smoothing* term $\frac{1}{2\alpha}\|\mathbf{x}\|_2$ changes the solution.
- However, if $r(\mathbf{x}) = \|\mathbf{x}\|_1$ (or any piecewise linear function) and if α is sufficiently large, minimizing $r(\mathbf{x})$ and $\bar{r}(\mathbf{x})$ are equivalent!
- Consequences of adding $\frac{1}{2\alpha}\|\mathbf{x}\|_2^2$ to $r(\mathbf{x})$
  - $\bar{r}(\mathbf{x})$ is strongly convex (so, faster convergence)
  - the dual function is continuously differentiable (smoothed to C^1)
  - dual gradient ascent is applicable
  - classical techniques for gradient decent such as Barzilai-Borwein steps, line search, Nesterov's method can speed up convergence
- Caution: dual is not C^2, so 2nd-order methods such as Newton and quasi-Newton are not safe!

Split Bregman

Split Bregman = Bregman iteration + Splitting + Sequentially solving subproblems for multiple iterations

Since

- Bregman iteration = augmented Lagrangian iteration
- ADMM = augmented Lagrangian iteration + Splitting + sequentially solving subproblems for just one pass

So, Split Bregman $\approx$ ADMM.

Outline

Orthogonal matching pursuit

- Orthogonal matching pursuit (OMP)[4]: With an initial point $\mathbf{x}^0$ and empty initial support $\mathcal{S}^0$, starting at $k = 1$, it iterates

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^{k-1}, \tag{16a}$$

$$\mathcal{S}^k = \mathcal{S}^{k-1} \cup \arg\min_i \{\|\phi_i \alpha - \mathbf{r}^k\|_2 : i \notin \mathcal{S}^{k-1}, \alpha \in \mathbb{R}\}, \tag{16b}$$

$$\mathbf{x}^k = \arg\min_{\mathbf{x}} \{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 : \mathrm{supp}(\mathbf{x}) \subseteq \mathcal{S}^k\}, \tag{16c}$$

  until $\|\mathbf{r}^k\|_2 \leq \epsilon$ is satisfied.

[4] Tropp 07'

CoSaMP

- CoSaMP[5]: With an initial point $\mathbf{x}^0$ and an estimate sparsity level s, starting at $k = 1$, CoSaMP iterates

$$\mathbf{r}^k \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}^{k-1} \qquad \text{(residual)} \qquad (17a)$$

$$\mathbf{a} \leftarrow \mathbf{A}^*\mathbf{r}^k \qquad \text{(correlation)} \qquad (17b)$$

$$\mathcal{T} \leftarrow \mathrm{supp}(\mathbf{x}^{k-1}) \cup \mathrm{supp}(\mathbf{a}_{2s}) \qquad \text{(merge supports)} \qquad (17c)$$

$$\mathbf{c} \leftarrow \arg\min_{\mathbf{x}}\{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 : \mathrm{supp}(\mathbf{x}) \subseteq \mathcal{T}\} \qquad \text{(least-squares)} \qquad (17d)$$

$$\mathbf{x}^k \leftarrow \mathbf{c}_s, \qquad \text{(pruning)} \qquad (17e)$$

until $\|\mathbf{r}^k\|_2 \leq \epsilon$ is satisfied, where $\mathbf{a}_{2s} = \arg\min_{\mathbf{a}}\{\|\mathbf{x} - \mathbf{a}\|_2 : \|\mathbf{x}\|_0 \leq 2s\}$ is the best $2s$-approximate of $\mathbf{a}$ and similarly, $\mathbf{c}_s$ is the best s-approximate of $\mathbf{c}$. Over the iterations, $\mathrm{supp}(\mathbf{x}^k)$ are updated but kept to contain no more than s components. Hence, $\mathcal{T}$ has no more than $3s$ components.

[5] Needell-Tropp 08'

Hard thresholding

- Iterative hard thresholding

$$\mathbf{c} \leftarrow \left(\mathbf{x}^k + \mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^k)\right), \tag{18a}$$

$$\mathbf{x}^{k+1} \leftarrow \mathbf{c}_s, \tag{18b}$$

- Hard thresholding pursuit[6]

$$\mathbf{c} \leftarrow \left(\mathbf{x}^k + \mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^k)\right), \tag{19a}$$

$$\mathcal{T} \leftarrow \mathrm{supp}(\mathbf{c}_s) \tag{19b}$$

$$\mathbf{c} \leftarrow \arg\min_{\mathbf{x}} \{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 : \mathrm{supp}(\mathbf{x}) \subseteq \mathcal{T}\} \tag{19c}$$

[6] Foucart 11'

Outline

Homotopy algorithms

- For model,

$$\min_{\mathbf{x}} \mu\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2. \tag{20}$$

  there is a method to compute its solutions corresponding to all values of $\mu > 0$ since, assuming the uniqueness of solution $\mathbf{x}^*$ for each μ, the solution path $\mathbf{x}^*(\mu)$ is continuous and piece-wise linear in μ.

- Optimality condition:

$$\mu\mathbf{p}(\mu) + \mathbf{A}^T(\mathbf{A}\mathbf{x}(\mu) - \mathbf{b}) = \mathbf{0}. \tag{21}$$

$$\frac{d}{d\mu}(\mu\mathbf{p}(\mu)) + \mathbf{A}^T\mathbf{A}\frac{d}{d\mu}\mathbf{x}(\mu) = \mathbf{0}. \tag{22}$$

  For active set Λ (non-zero subset of $\mathbf{x}$), we have

$$\mathbf{p}_\Lambda(\mu) + \mathbf{A}_\Lambda^T\mathbf{A}_\Lambda\frac{d}{d\mu}\mathbf{x}_\Lambda(\mu) = \mathbf{0}. \tag{23}$$

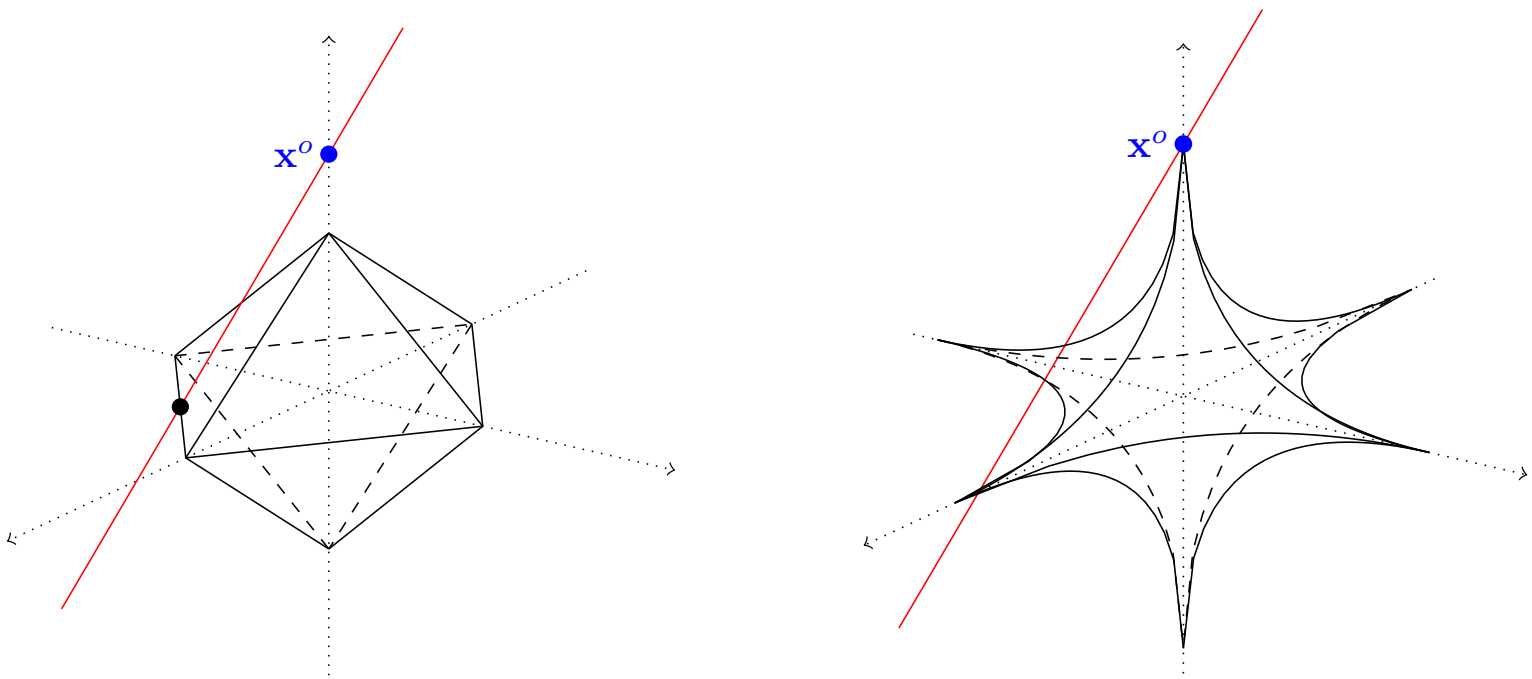
$$\frac{d}{d\mu}\mathbf{x}_\Lambda(\mu) = (\mathbf{A}_\Lambda^T\mathbf{A}_\Lambda)^{-1}\mathbf{p}_\Lambda(\mu). \tag{24}$$

Non-convex approaches

- Non-convex optimization, includes ones based on minimizing the non-convex ℓ_q quasi-norm

$$\|\mathbf{x}\|_q = (\sum_i |x_i|^q)^{1/q}, \quad 0 < q < 1,$$

and its variants.



(a) ℓ_1 Minimization (b) $\ell_{1/2}$ Minimization

Figure: ℓ_1 vs. $\ell_{1/2}$ minimization.

Thank You!

References:

- "Compressive Sensing for Wireless Networks" by Z. Han, H. Li, and W. Yin, Cambridge University Press, 2013
- `http://www.caam.rice.edu/~optimization/sparse/index.html` (Course at Rice University: Email me for the password)
- `http://nuit-blanche.blogspot.com/`
- `http://dsp.rice.edu/cs`