On Efficiently Computable Compressed Sensing

A. Nemirovski
nemirovs@isye.gatech.edu, ISyE GaTech

Joint Research with

A. Iouditski
Anatoli.Iouditski@imag.fr, Fourier University, Grenoble, France
F. Kilinc Karzan
fkilinc@gatech.edu, ISyE GaTech

MOPTA, August 18 – 20, 2010

Overview

♣ **Compressed Sensing: What is it?**

♣ **Verifiable sufficient conditions in Compressed Sensing**

- Verifiable sufficient conditions for goodness of a sensing matrix
  - — the relaxation scheme
  - — limits of performance
- Applying the goodness conditions: Error bounds for imperfect ℓ_1 recovery
  - — uncertain-but-bounded observation error
  - — random observation error

♣ **ℓ_1 minimization via First Order algorithms**

- Strategy
- Performance in deterministic case
- Acceleration by randomization

Compressed Sensing: what it is?

♣ **Compressed Sensing** is about recovery of a high-dimensional signal x from its relatively low-dimensional projection

$$y = Ax + \xi$$

• y: observation • ξ: observation noise • A: $m \times n$ sensing matrix, $m \ll n$

♣ It is assumed that x is sparse — possesses at most a known number $s \ll m$ nonzero entries.

♠ Sparsity makes the recovery problem solvable, at least in the noiseless case $\xi = 0$. In this case, for a "general position" sensing matrix A, one has

$$x = \operatorname*{argmin}_{z \in \mathbb{R}^n} \{\mathrm{Card}\{i : z_i \neq 0\} : Az = y\} \qquad (C)$$

However: the arising combinatorial problem is intractable

$\Rightarrow$**The** standard recovery routine in CS is the ℓ_1 recovery:

$$y \mapsto \widehat{x} \in \operatorname*{Argmin}_{z} \{\|z\|_1 : \|Az - y\| \leq \delta\}$$

• δ: properly chosen tolerance, e.g., an a priori upper bound on $\|\xi\|$

s-good sensing matrices

$$\mathbb{R}^n \ni x \mapsto y = Ax + \xi \in \mathbb{R}^m, \ \|\xi\| \le \delta$$
$$\widehat{x} \in \mathrm{Argmin}_z \left\{\|z\|_1 : \|Az - y\| \le \delta\right\}$$

Definition

A is s-good, if in the noiseless case ($\delta = 0$) ℓ_1 recovery is exact ($\widehat{x} = x$) for every x with at most s nonzero entries.

♣ A *necessary and sufficient* condition for A to be s-good is:

$$\gamma_s(A) := \max_x \left\{\|x\|_{s,1} : x \in \mathrm{Ker} A, \|x\|_1 \le 1\right\} < 1/2$$

[$\|x\|_{s,1}$: sum of s largest magnitudes of entries in x]

[Donoho&Huo'01, Zhang'05, Cohen&Dahmen&DeVore'06,...]

♠ $\gamma_s(A)$ is difficult to compute $\Rightarrow$*the condition is unverifiable...*

Verifiable sufficient condition for s-goodness

$$\gamma_s(A) := \max_x \left\{ \|x\|_{s,1} : x \in \mathrm{Ker} A, \|x\|_1 \leq 1 \right\} < 1/2$$
$$\Leftrightarrow \; A \text{ is } s\text{-good}$$

♣ Theorem [Ioud.&Nem.'08]

The efficiently computable quantity

$$\alpha_s(A) = \min_Y \left\{ \max_{1 \leq j \leq n} \|\mathrm{Col}_j[I - Y^T A]\|_{s,1} \right\}$$
$$[\mathrm{Col}_j[B] : \; j\text{-th column of } B]$$

is an upper bound on $\gamma_s(A)$ which is exact for $s = 1$: $\gamma_1(A) = \alpha_1(A)$.

$\Rightarrow$*The verifiable condition $\alpha_s(A) < 1/2$ is sufficient for A to be s-good.*

Remark: $\alpha_s(A) \leq s\alpha_1(A) = s\gamma_1(A)$
$\Rightarrow$*The easily verifiable "rough" condition $\alpha_1(A) < \frac{1}{2s}$ is sufficient for A to be s-good.*

What is inside: a novel (?) relaxation scheme for maximizing convex functions over polytopes

Situation: Consider the problem

$$\mathrm{Opt} = \max_x \{f(x) : x \in \mathrm{Conv}\{g_1, ..., g_N\}, Ax = 0\}$$
$$A : m \times n$$

of maximizing an efficiently computable convex function $f(x)$ over the intersection of a polytope given by its vertices and a linear subspace.

Note: this is a universal form of the problem of maximizing convex function over a polytope.

Relaxation scheme: Let $Y \in \mathbb{R}^{m\times n}$ and $\lambda \in \mathbb{R}^m$, and let

$$U(Y, \lambda) = \max_{1\leq i\leq N} \left\{f([I - Y^T A]g_i) + \lambda^T A g_i\right\}$$

Observation: $U(Y, \lambda)$ *is a convex function of* Y, λ *such that*

$$\mathrm{Opt} \leq U(Y, \lambda)\ \forall (Y, \lambda)$$

$\Rightarrow$*The efficiently computable quantity* $\mathrm{Opt}^+ = \inf_{Y,\lambda} U(Y, \lambda)$ *is an upper bound on* Opt.

Relaxation scheme (continued)

Claim: $U(Y,\lambda) := \max_i \left\{ f([I - Y^T A]g_i) + \lambda^T A g_i \right\}$
$$\geq \mathrm{Opt} := \max_x \left\{ f(x) : x \in \mathrm{Conv}\{g_1, ..., g_N\}, Ax = 0 \right\}$$

Indeed, let $x = \sum_i \mu_i g_i$ be a convex combination of g_i such that $Ax = 0$. We have

$$\begin{array}{l} U(Y,\lambda) = \max_i \left\{ f([I - Y^T A]g_i) + \lambda^T A g_i \right\} \\ \geq \sum_i \mu_i \left[f([I - Y^T A]g_i) + \lambda^T A g_i \right] \\ \geq f\left(\sum_i \mu_i [I - Y^T A] g_i \right) + \lambda^T A \left[\sum_i \mu_i g_i \right] \\ = f([I - Y^T A]x) + \lambda^T A x = f(x) \end{array}$$

and Claim follows.

Note: the "λ-component" of the relaxation scheme is the standard Lagrangian relaxation. The "Y–component" seems to be new.

♠ To get verifiable sufficient goodness conditions, one applies the outlined relaxation scheme to

$$\gamma_s(A) = \max_x \left\{ \|x\|_{s,1} : x \in \mathrm{Conv}\{\pm e_1, ..., \pm e_n\}, Ax = 0 \right\}$$

In this case, the Lagrangian component does not help...

Relations with other goodness conditions

Relation to Mutual Incoherence

♣ The only previously known verifiable sufficient condition for $A = [A_1, ..., A_n]$ to be s-good is based on mutual incoherence

$$\mu(A) = \max_{i \neq j} |A_i^T A_j| / A_i^T A_i$$

and states that *A is s-good whenever* $s\mu(A)/(1 + \mu(A)) < 1/2$ [Donoho&Elad&Temlyuakov'06].

Fact [Ioud.&Nem.'08]: *The easily verifiable "rough" sufficient condition for s-goodness* $\alpha_1(A) < \frac{1}{2s}$ *provably is less conservative than the condition based on mutual incoherence.*

Relation to Restricted Isometry Property

♠ The standard in CS unverifiable sufficient goodness condition is based on the Restricted Isometry Property $\mathrm{RIP}(\delta, k)$:

$$(1-\delta)I_k \preceq A_k^T A_k \preceq (1+\delta)I_k \text{ for every } m \times k \text{ submatrix } A_k \text{ of } A$$

• *Every* $\mathrm{RIP}(\frac{2}{5}, 2s)$*-matrix* A *is* s*-good.*

• *For large* m, n*, a randomly generated* $A \in \mathbb{R}^{m\times n}$ *with independent* $\mathcal{N}(0, m^{-1/2})$ *(or* $\pm m^{-1/2}$*) entries with overwhelming probability is* $\mathrm{RIP}(0.1, 2s)$ *with* s *as large as* $O(m/\ln(2n/m))$.

Fact [Ioud.&Nem.'08]**:** *Whenever* A *is* $\mathrm{RIP}(\delta, k)$ *with* $\delta < \sqrt{2}-1$*, one has*

$$s < \frac{(1-\delta)\sqrt{k-1}}{2\sqrt{2}\delta} = O(1)\sqrt{k} \Rightarrow \alpha_1(A) < \frac{1}{2s}.$$

$\Rightarrow$Already rough sufficient condition can certify s-goodness of an $m \times n$ sensing matrix for s as large as $O(1)\sqrt{m/\ln(n/m)}$.

Fact [Ioud.&Nem.'08]**:** *When* A *is not "nearly square:"* $\frac{n}{m} \geq \theta > 1$*, the condition* $\alpha_s(A) < 1/2$ *can be satisfied only if* $s \leq O(1)\frac{\theta}{\sqrt{\theta-1}}\sqrt{m}$.

Note: So far, all explicitly defined families of s-good $m \times n$ sensing matrices A with $n/m \geq \theta > 1$ obey the bound $s \leq O(1)\sqrt{m}$.

Extension to the “signed” case [Ioud.&Kil.-Karz.&Nem.’09]

♣ The above results admit natural extension to the case of “signed” sparse signals

$$x \in \mathbb{R}^n : x_j \geq 0, j \in I_+ \ \&\ x_j \leq 0,\ j \in I_-$$

and associated “signed ℓ_1 recovery”

$$[y = Ax + \xi,\ \|\xi\| \leq \delta] \mapsto \widehat{x} := \operatorname*{argmin}_z \left\{ \|z\|_1 : \begin{array}{c} \|Az - y\| \leq \delta \\ z_j \geq 0,\ j \in I_+ \\ z_j \leq 0,\ j \in I_- \end{array} \right\}$$

Upper bounding of goodness level

♣ In order to certify that A is not s-good, it suffices to show that

$$\begin{array}{rcl}\frac{1}{2} \leq \gamma_s(A) & := & \max\limits_{x} \left\{ \|x\|_{s,1} : \|x\|_1 \leq 1, Ax = 0 \right\} \\ & = & \max\limits_{u,x} \left\{ u^T x : \begin{array}{l} x \in X = \{\|x\|_1 \leq 1, Ax = 0\} \\ u \in U_s = \{\|u\|_\infty \leq 1, \|u\|_1 \leq s\} \end{array} \right\}\end{array}$$

This can be done by bounding $\gamma_s(A)$ from below via several series of randomly initialized alternating maximizations of $u^T x$ over $u \in U_s$ and $x \in X$.

Efficiently computable goodness bounds

μ-LB $\leq$ α-LB $\leq$ $s_*(A)$ $\leq$ UB

[Goodness $s_*(A)$ of A: the largest s such that A is s-good]

		Unsigned			Nonnegative	
	m	μ-LB	α-LB	UB	LB	UB
$m \times 256$ random submatrix of 256×256 Fourier matrix	128	3	5	11	5	32
	178	3	7	16	7	42
	242	5	11	26	11	89
$m \times 256$ random submatrix of 256×256 Hadamard matrix	128	2	5	7	5	7
	178	4	9	15	9	19
	242	12	26	31	27	31
$m \times 256$ Rademacher matrix	128	1	5	15	5	48
	178	2	8	24	9	78
	242	2	23	47	27	111
$m \times 256$ Gaussian matrix	128	1	5	14	5	44
	178	2	8	24	9	79
	242	2	23	47	27	112

- *α_s-based goodness bounds significantly outperform bounds based on mutual incoherence*
- *Computability has its price: for random matrices, there is a significant gap between upper and lower goodness bounds*

Numerical illustration (continued)

Efficiently computable goodness bounds

μ-LB $\leq$ α-LB $\leq$ $s_*(A)$ $\leq$ UB

	m	μ-LB	α-LB	UB
$m \times 1024$ Gaussian matrix	102	2	2	8
	204	2	4	18
	307	2	6	30
	409	3	7	44
	512	3	10	61
	614	3	12	78
	716	3	15	105
	819	4	21	135
	921	4	32	161
960×1024 convolution matrix	960	0	5	7

• *Matrices with "personal story" seem to have smaller and easier to estimate goodness than random matrices of the same sizes.*

Application: Error bound for imperfect recovery with uncertain-but-bounded noise

$$\mathbb{R}^n \ni x \mapsto y = Ax + \xi \in \mathbb{R}^m, \ \|\xi\| \le \delta$$
$$\mathrm{Opt} := \min_z \{\|z\|_1 : \|Az - y\| \le \delta\}$$
$$\widetilde{x} : \|\widetilde{x}\|_1 \le \mathrm{Opt} + \nu \ \& \ \|A\widetilde{x} - y\| \le \mu$$

♣ Theorem [Ioud.&Nem.'08]:

Let $\alpha < 1/2$, $\beta > 0$ be such that

$$\exists Y : \|\mathrm{Col}_j[I_n - Y^T A]\|_{s,1} \le \alpha \,\forall j \ \& \ \|\mathrm{Col}_j[Y]\|_* \le \frac{\beta}{s} \forall j$$

where $\|\cdot\|_*$ is the conjugate of $\|\cdot\|$. Let also x^s be the best in $\|\cdot\|_1$ s-sparse approximation of $x \in \mathbb{R}^n$. Then

$$\|\widetilde{x} - x\|_1 \le \frac{2\beta(\delta + \mu) + \nu + 2\|x - x^s\|_1}{1 - 2\alpha}$$

Application: Recovery in the case of random observation noise

$$\mathbb{R}^n \ni x \mapsto y = Ax + \xi \in \mathbb{R}^m$$
$$\xi = \sigma\zeta + u,\ \zeta \sim \mathcal{N}(0, I),\ \|u\| \leq \delta$$
x : s-sparse with known s

Goal and Assumptions

♠ **Goal:** Given $\epsilon \in (0,1)$ and s, to ensure with probability $\geq 1-\epsilon$ "good recovery" of nearly s-sparse signals x

♠ **Assumption A:** We have in our disposal matrix Y such that
$$\alpha := s\|I - Y^TA\|_\infty < \frac{1}{2}$$

• We set
$$\|Y\|_{\sigma,\delta} = \max_{1\leq j\leq n}\left[\delta\|\mathrm{Col}_j[Y]\|_* + \sigma\sqrt{2\ln(n/\epsilon)}\|\mathrm{Col}_j[Y]\|_2\right]$$

Regular and Penalized ℓ_1 recoveries

$$\mathbb{R}^n \ni x \mapsto y = Ax + \xi \in \mathbb{R}^m$$
$$\xi = \sigma\zeta + u,\ \zeta \sim \mathcal{N}(0, I),\ \|u\| \leq \delta$$

- Regular ℓ_1 recovery:
$$y \mapsto \widehat{x} = \widehat{x}(y) \in \mathrm{Argmin}_z \left\{ \|z\|_1 : \|Y^T(Az - y)\|_\infty \leq \|Y\|_{\sigma,\delta} \right\}$$
- Penalized ℓ_1 recovery:
$$y \mapsto \widehat{x} = \widehat{x}(y) \in \mathrm{Argmin}_z \left\{ \|z\|_1 + 2s\|Y^T(Az - y)\|_\infty \right\}$$

Note: Penalized recovery does not require knowledge of σ, δ, ϵ!

Theorem [Ioud.,Kil.-Karz.,Nem.'10]

Under Assumption A, there exists a set $\mathcal{Z}$ of "good" ζ such that

- $\mathrm{Prob}\{\zeta \in \mathcal{Z}\} \geq 1 - \epsilon$
- *When $\zeta \in \mathcal{Z}$, for both Regular and Penalized ℓ_1 recovery one has* $\forall (x \in \mathbb{R}^n, u, \|u\| \leq \delta, y = Ax + \sigma\zeta + u)$:
$$\left\{ \begin{array}{lcl} \|x - \widehat{x}(y)\|_\infty & \leq & \omega := 2\frac{s^{-1}\|x - x^s\|_1 + 2\|Y\|_{\sigma,\delta}}{1 - 2\alpha} \\ \|x - \widehat{x}(y)\|_1 & \leq & s\omega \end{array} \right.$$

where x^s is the best in $\|\cdot\|_1$ s-sparse approximation of x.

• When $\zeta \in \mathcal{Z}$, for both Regular and Penalized ℓ_1 recovery one has $\forall (x \in \mathbb{R}^n, u, \|u\| \leq \delta, y = Ax + \sigma\zeta + u)$:

$$\begin{cases} \|x - \widehat{x}(y)\|_\infty & \leq \quad \omega := 2\frac{s^{-1}\|x - x^s\|_1 + \|Y\|_{\sigma,\delta}}{1-2\alpha} \\ \|x - \widehat{x}(y)\|_1 & \leq \quad s\omega \end{cases}$$

Remarks:

• $\omega \leq O(\sigma + \delta + s^{-1}\|x - x^s\|_1)$ *is small when when σ, δ are small and x is nearly s-sparse.*

• The set $\mathcal{Y} = \{(Y, t, \tau) : s\|I - Y^T A\|_\infty \leq t, \ \|Y\|_{\sigma,\delta} \leq \tau\}$ is convex

$\Rightarrow$*Given s, σ, δ and an upper bound on $\|x - x^s\|_1$, we can efficiently optimize the quality of the recovery, as given by Theorem, in Y.*

How it works

♣ Gaussian Setup

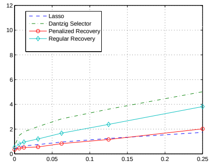
- A: Gaussian 161×256 with normalized columns
- $\|\cdot\|$: $\{u : \|u\| \leq 1\} = \left\{ Av : \begin{array}{l} |v_1| \leq 1, \ |v_2 - v_1| \leq 1 \\ |v_{j+1} - 2v_j + v_{j-1}| \leq 1 \ \forall j \end{array} \right\}$
- $\epsilon = 0.01$

♣ Convolution Setup

- A: 2D signal $[x_{ij}]_{0 \leq i,j \leq 15}$ is convolved with kernel $[K_{ij}]_{-7 \leq i,j \leq 7}$. The output is observed on the “deficient” grid $\{1 \leq i \leq 15, 0 \leq j \leq 15\}$, which results in a linear mapping $x \mapsto Ax : \mathbb{R}^{256} \to \mathbb{R}^{240}$.
- $\|\cdot\|$: $\{u : \|u\| \leq 1\} = \{Av : v \in V\}$
- V : all functions $v \in \mathbb{R}(\mathbf{Z}_{16} \times \mathbf{Z}_{16})$ with zero mean satisfying $\|\Delta^2 v\|_\infty \leq 1$
- Δ: discrete Laplacian on $\mathbb{R}(\mathbf{Z}_{16} \times \mathbf{Z}_{16})$
- $\epsilon = 0.01$

How it works (continued)

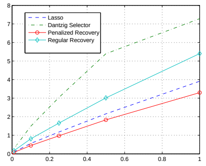
Empirical Averages of Recovery Errors, Gaussian A



ℓ_∞ error vs. δ

$\sigma = 0.1, s = 2,$

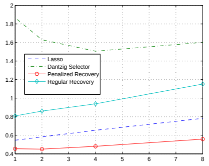
$\alpha = 0.2, \|x\|_1 = 10$



ℓ_∞ error vs. σ

$\delta = 0.01, s = 2,$

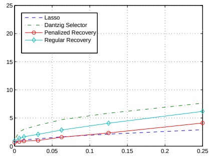
$\alpha = 0.2, \|x\|_1 = 10$



ℓ_∞ error vs. s

$\delta = 0.01, \sigma = 0.1,$

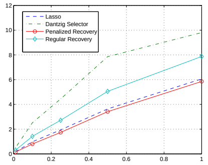
$\alpha = 0.1s, \|x\|_1 = 5s$



ℓ_1 error vs. δ

$\sigma = 0.1, s = 2,$

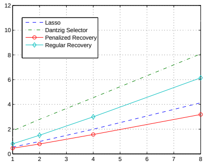
$\alpha = 0.2, \|x\|_1 = 10$



ℓ_1 error vs. σ

$\delta = 0.01, s = 2,$

$\alpha = 0.2, \|x\|_1 = 10$



ℓ_1 error vs. s

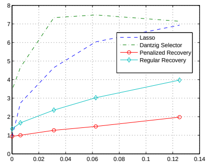
$\delta = 0.01, \sigma = 0.1,$

$\alpha = 0.1s, \|x\|_1 = 5s$

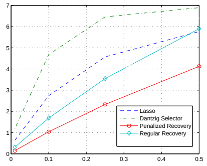
- Winners: Lasso and Penalized ℓ_1 Recovery

How it works (continued)

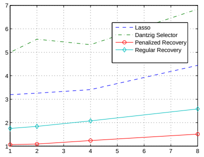
Empirical Averages of Recovery Errors, Convolution A



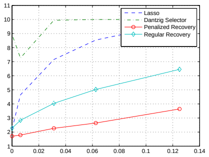
ℓ_∞ error vs. δ
$\sigma = 0.1, s = 2,$
$\alpha = 0.4, \|x\|_1 = 10$



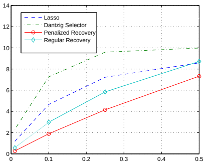
ℓ_∞ error vs. σ
$\delta = 0.01, s = 2,$
$\alpha = 0.4, \|x\|_1 = 10$



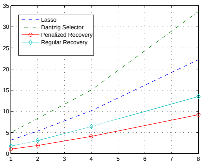
ℓ_∞ error vs. s
$\delta = 0.01, \sigma = 0.1,$
$\alpha = 0.2s, \|x\|_1 = 5s$



ℓ_1 error vs. δ
$\sigma = 0.1, s = 2,$
$\alpha = 0.4, \|x\|_1 = 10$



ℓ_1 error vs. σ
$\delta = 0.01, s = 2,$
$\alpha = 0.4, \|x\|_1 = 10$



ℓ_1 error vs. s
$\delta = 0.01, \sigma = 0.1,$
$\alpha = 0.2s, \|x\|_1 = 5s$

- Winner: Penalized ℓ_1 Recovery

ℓ_1 minimization via deterministic and randomized first order algorithms

♣ Problems of ℓ_1 minimization arising in Signal Processing

$$\mathrm{Opt} = \min_z \{\|z\|_1 : \|Az - b\|_p \le \delta\} \quad [p = \infty \text{ or } p = 2]$$

may have dense sensing matrices A with sizes in the range of 10^4 - 10^5 and more. Whenever this is the case, an iteration of every known polynomial time algorithm becomes too time consuming.

♠ At present, *the most attractive alternative to IP methods in the extremely large-scale ℓ_1 minimization is offered by computationally cheap First Order methods*.

♠ In FOMs, *the effort per iteration is dominated by computing $O(1)$ matrix-vector products involving A and A^T*, which is much easier than solving systems of linear equations of sizes comparable with those of A, as required in IPMs.

♠ *One can further simplify an iteration by replacing precise matrix-vector multiplications by their randomized versions*.

Limits of performance of FOMs

♠ FOMs are provably badly suited for solving large-scale problems to high accuracy.
However: FOMs can be theoretically and practically efficient when medium accuracy solutions are sought. In this case, *FOMs under favorable circumstances (e.g., in ℓ_1 minimization) exhibit nearly dimension-independent rate of convergence*, which is crucial in large-scale applications.

The approach

$$\mathrm{Opt} = \min_z \{\|z\|_1 : \|Az - b\|_p \le \delta\} \quad [p = \infty \text{ or } p = 2] \qquad (\ell_1)$$

The strategy

♣ The state-of-the-art complexity results on the first order methods suggest the following strategy:

♠ The problem of interest (ℓ_1) is reformulated as

$$\frac{1}{\mathrm{Opt}} = \max \left\{ \rho : \quad \Phi(\rho) := \min_{x, \|x\|_1 \le 1} \|Ax - \rho b\|_p - \delta\rho \right.$$
$$\left. = \min_{x:\|x\|_1 \le 1} \max_{y:\|y\|_{\frac{p}{p-1}} \le 1} y^T(Ax - \rho b) - \delta\rho \le 0 \right\}$$

• The solution is found by a Newton-type root finding routine as applied to the *master problem* $\max\{\rho : \Phi(\rho) \le 0\}$

• (Approximate) information on $\Phi(\cdot)$ used by root finding is given by the *Mirror Prox FOM* [Nem.'04,Ioud.&Kil.-Karz.&Nem.'10] as applied to the bilinear saddle point problem

$$\Phi(\rho) = \min_{x:\|x\|_1 \le 1} \max_{y:\|y\|_{p_*} \le 1} \left[y^T(Ax - \rho b) - \delta\rho \right], \quad p_* = \frac{p}{p-1}$$

The approach (continued)

Acceleration by randomization

♣ With our approach, ℓ_1-minimization reduces to a "small series" of bilinear saddle point problems

$$\min_{x\in X}\max_{y\in Y}\left[\langle a,x\rangle+\langle b,y\rangle+\langle y,Ax\rangle\right] \qquad (S)$$

♠ When solving (S) by a FOM, the main effort is to compute matrix-vector products involving A and A^T. These computations are easy to randomize: to estimate Bu, we

- treat the vector $\frac{\mathrm{abs}[u]}{\|u\|_1}$ as a probability distribution on the set of columns of B,
- draw at random a column $B_\jmath$ of B. The vector $\|u\|_1\mathrm{sign}(u_\jmath)B_\jmath$ is the desired unbiased estimate of Bu.

♣ Randomization simplifies dramatically an iteration, while increasing the number of iterations required to get an ϵ-solution. *In a meaningful range of problem sizes and desired accuracies, the tradeoff between iteration complexity and iteration count is in favor of randomization.*

Results, Deterministic case

Theorem [Ioud.&Nem.'09]

Consider a feasible and nontrivial ($\|b\|_p \geq 2\delta$) ℓ_1 minimization problem

$$\mathrm{Opt}_p = \min_z \{\|z\|_1 : \|Az - b\|_p \leq \delta\} \qquad (\ell_1)$$

with $A \in \mathbb{R}^{m\times n}$ and $p \in \{2, \infty\}$, and let

$$\|A\|_{1,p} = \max_j \|\mathrm{Col}_j[A]\|_p.$$

Given ϵ, $0 < \epsilon < \|A\|_{1,p}\mathrm{Opt}_p$, one can find an ϵ-solution x_ϵ to (ℓ_1):

$$\|x_\epsilon\|_1 \leq \mathrm{Opt}_p \;\&\; \|Ax_\epsilon - b\|_p \leq \delta + \epsilon$$

in no more than $\left(\frac{\Omega_p\|A\|_{1,p}\mathrm{Opt}_p}{\epsilon}\right) \ln\left(\frac{\Omega_p\|A\|_{1,p}\mathrm{Opt}_p}{\epsilon}\right)$ *steps, where*

$$\Omega_p = O(1) \cdot \begin{cases} \sqrt{\ln(m)\ln(n)}, & p = \infty \\ \sqrt{\ln(n)}, & p = 2 \end{cases}.$$

Computational effort per step is dominated by the necessity to multiply $O(1)$ vectors by A and A^T.

How it works: ℓ_1 minimization by Deterministic MP

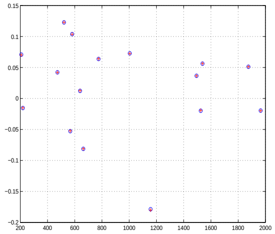
$$\widehat{x} \approx \underset{x}{\operatorname{argmin}} \{\|Ax - b\|_\infty : \|x\|_1 \le 1\}$$
$$\Leftrightarrow \min_{\|x\|_1 \le 1} \max_{\|y\|_1 \le 1} y^T(Ax - b)$$

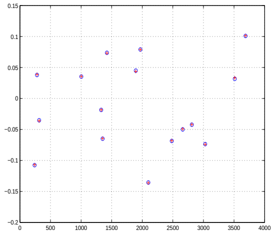
$$\left[\begin{array}{l} A\text{: random } m \times n \text{ submatrix of } n \times n \text{ D.F.T. matrix} \\ b\text{: } \|Ax_* - b\|_\infty \le \delta = \text{5.e-3 with 16-sparse } x_*,\ \|x_*\|_1 = 1 \end{array}\right]$$

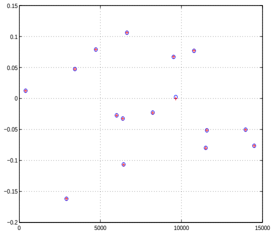
		Errors			CPU
$m \times n$	Method	$\|x_* - \widehat{x}\|_1$	$\|x_* - \widehat{x}\|_2$	$\|x_* - \widehat{x}\|_\infty$	sec
512×2048	DMP	0.005	0.002	0.001	3.3
	IP	0.039	0.006	0.002	321.6
1024×4096	DMP	0.010	0.003	0.002	3.5
	IP	Out of space (2GB RAM)			
4096×16384	DMP	0.006	0.003	0.002	46.4
	IP	not tested			

- DMP: Deterministic Mirror Prox utilizing FFT
- IP: Commercial Interior Point LP solver `mosekopt`

How it works (continued)







512×2048 1024×4096 4096×16384

ℓ_1-recovery: o: true signal, $+$: DMP recovery

How it works (continued)

Situation and Goal

We observe randomly selected pixels in a 256×256 image X at sampling rate 33% and want to recover the image.

Approach

• The underlying signal is the vector x of coefficients of X in a 2D wavelet basis: $X = Ux$, with $n \times n$ orthogonal U, $n = 65,536$.

• Observed part of the image is $y = Ax$ with the $m = 21,789 \approx n/3$ rows of A selected at random from rows of U.

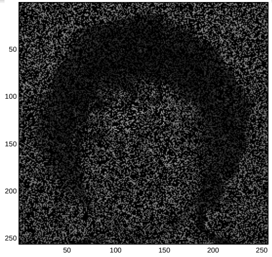
Note: A is rather dense ($3.4\% \approx 5.3 \cdot 10^7$ nonzeros).

• The recovery is $X_\epsilon = Ux_\epsilon$,

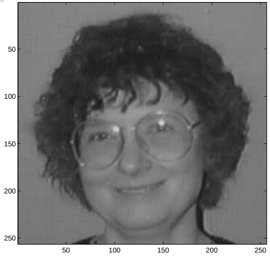
$$x_\epsilon : \left\{ \begin{array}{l} \|x_\epsilon\|_1 \leq \min\{\|z\|_1 : Az = y\} \\ \|Ax_\epsilon - y\|_2 \leq \epsilon \|y\|_2 \end{array} \right.$$

• Multiplication by A and A^T takes time linear in n

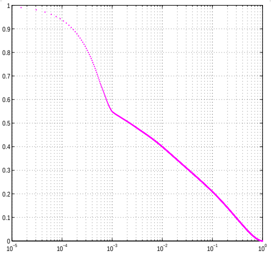
$\Rightarrow$*we are in an ideal position to apply deterministic first order methods*



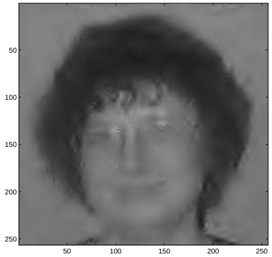
Observations



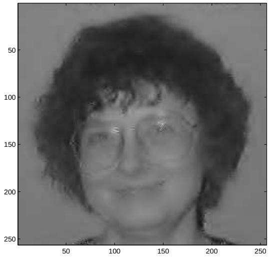
True image



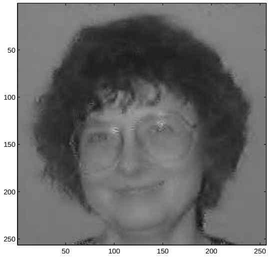
$\|x - x^s\|_1$ vs. s/n



$X(0.0647)$
Steps: 328 Calls[†]:918
CPU$=$ 99 sec



$X(0.0271)$
Steps: 947 Calls: 2,682
CPU$=$ 290 sec



$X(0.0075)$
Steps: 4,746 Calls: 13,469
CPU$=$ 1460 sec

[†] Call: a pair of matrix-vector multiplications $(x, y) \mapsto (Ax, A^T y)$.

Acceleration by Randomization

Theorem [Ioud.&Kil.-Karz.&Nem.'10]

Consider a feasible problem

$$\mathrm{Opt}_p = \min_z \{\|z\|_1 : \|Az - b\|_p \leq \delta\} \qquad (\ell_1)$$

with $A \in \mathbb{R}^{m\times n}$ *and* $p \in \{2, \infty\}$, *and let* $\epsilon \in (0, \|A\|_{1,p}\mathrm{Opt}_p]$ *be given. Then, for every* $\chi \in (0, 1/2]$,

(i) *In the case of* $p = \infty$, *assuming* δ *small enough (namely,* $2\delta \leq \|b\|_\infty$*), an* ϵ*-solution to* (ℓ_1) *can be found, with confidence* $\geq 1 - \chi$, *in at most*

$$O(1)\left[\frac{\sqrt{\ln(m)\ln(n)}\|A\|_{1,\infty}\mathrm{Opt}_\infty}{\epsilon}\ln\left(\frac{\sqrt{\ln(m)\ln(n)}\|A\|_{1,\infty}\mathrm{Opt}_\infty}{\epsilon\chi}\right)\right]^2$$

steps of a randomized algorithm, with effort per step dominated by the necessity to extract from A $O(1)$ *columns and rows.*

Note: Setting $\omega = \epsilon/(\|A\|_{1,\infty}\mathrm{Opt}_\infty)$ and modulo logarithmic factors, randomization rises the iteration count from $O(\omega^{-1})$ to $O(\omega^{-2})$, while reducing the effort per iteration from $O(mn)$ to $O(m+n)$ a.o.

Acceleration by Randomization (continued)

$$\mathrm{Opt}_p = \min_z \{\|z\|_1 : \|Az - b\|_p \leq \delta\} \qquad (\ell_1)$$

Theorem (continued)

(ii) *In the case of $p = 2$, assuming δ small enough (namely, $2\sqrt{m}\delta \leq \|b\|_2$), an ϵ-solution to (ℓ_1) can be found with confidence $\geq 1 - \chi$ in at most*

$$O(1)\left[\frac{\sqrt{\ln(n)}\Gamma(A)\|A\|_{1,2}\mathrm{Opt}_2}{\epsilon}\ln\left(\frac{\sqrt{\ln(n)}\Gamma(A)\|A\|_{1,2}\mathrm{Opt}_2}{\epsilon\chi}\right)\right]^2,$$

$$\Gamma(A) = \sqrt{m}\|A\|_{1,\infty}/\|A\|_{1,2}$$

steps of a randomized algorithm with the same as in (i) *effort per step.*

• *With randomized preprocessing*

$$[A, b] \leftarrow [U\mathrm{Diag}\{\xi\}A, U\mathrm{Diag}\{\xi\}b]$$

(U is an appropriate orthogonal matrix, ξ is a random ± 1 vector), with confidence $\geq 1 - \chi$ one has $\Gamma(A) \leq O(1)\sqrt{\ln(mn/\chi)}$. The cost of this preprocessing does not exceed $O(1)mn\ln(m)$ a.o.

Acceleration by Randomization: how it works

$$\mathrm{Opt}_p = \min_z \{\|z\|_1 : \|Az - b\|_p \leq \delta\} \qquad (\ell_1)$$

• A: randomly drawn $m \times n$ matrix with i.i.d. entries taking values $\pm m^{-1/p}$ with probabilities $1/2$
• $b = Ax_* + \xi$ with randomly selected sparse ($\lfloor\sqrt{n}\rfloor$ nonzeros) vector x_*, $\|x_*\|_1 = 1$, and randomly generated ξ, $\|\xi\|_p = \delta = 0.005$.

Acceleration by Randomization: how it works

♠ Uniform fit $p = \infty$, $\epsilon = 0.0025$

		DMP		SMP		
Size of A		Calls	CPU	FCalls	CPU	$\frac{\text{CPU,DMP}}{\text{CPU,SMP}}$
1000 x 2000	min	811	32.5	88.8	22.5	1.238
	mean	1500	61.0	130.0	31.6	1.975
	max	2339	98.1	188.2	44.7	3.325
2000 x 4000	min	963	142.3	84.8	77.1	1.846
	mean	2340	346.2	121.0	105.2	3.243
	max	4217	622.4	158.8	135.9	5.747
4000 x 8000	min	1697	992.3	69.2	271.3	2.565
	mean	2570	1470.7	90.2	348.0	4.368
	max	4380	2516.6	104.4	394.5	7.324

Deterministic algorithm DMP vs. randomized algorithm SMP

5 experiments per each size.

- Calls: # of matrix-vector multiplications in DMP run
- FCalls: equivalent # of full matrix-vector multiplications in SMP run

Acceleration via Randomization: how it works

♠ ℓ_2 fit $p = 2$, $\epsilon = 0.0025$

		DMP		SMP		
Size of A		Calls	CPU	FCalls	CPU	$\frac{\text{CPU,DMP}}{\text{CPU,SMP}}$
1000 x 2000	min	321	12.5	102.2	29.9	0.374
	mean	719	28.3	139.3	43.2	0.703
	max	916	35.5	194.9	60.0	1.187
2000 x 4000	min	515	74.2	54.1	68.4	0.763
	mean	616	89.0	63.6	80.9	1.136
	max	720	104.5	71.0	97.3	1.528
4000 x 8000	min	526	293.3	42.6	195.6	1.257
	mean	756	424.6	45.2	210.7	2.045
	max	935	526.8	48.6	233.3	2.625

Deterministic algorithm DMP vs. randomized algorithm SMP

5 experiments per each size.

- Calls: # of matrix-vector multiplications in DMP run
- FCalls: equivalent # of full matrix-vector multiplications in SMP run

Acceleration by Randomization: how it works

♠ Uniform and ℓ_2 fits

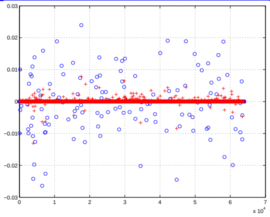
							$\|\widehat{x}-x_*\|_r$		
	p	Steps	Calls	FCalls	CPU	$\|A\widehat{x}-b\|_p$	$r=1$	$r=2$	$r=\infty$
DMP	2	21	45	45	7229	0.0350≈ $0.6\|b\|_p$	1.21 121%	0.095 86%	0.020 76%
SMP	2	9104	13,648	29.4	7252	0.0080≈ $0.1\|b\|_p$	0.167 17%	0.015 13%	0.003 10%
DMP	∞	19	40	40	7364	0.1638≈ $0.6\|b\|_p$	1.25 125%	0.113 97%	0.033 98%
SMP	∞	12006	17816	19.3	6050	0.0075≈ $0.03\|b\|_p$	0.090 9%	0.007 6%	0.002 6%

Experiments with $32,000\times 64,000$ matrix

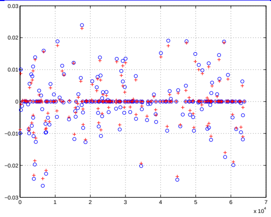
$\approx 7,200$ sec CPU limit

Percents: $\|\widehat{x}-x_*\|/\|x_*\|_r$

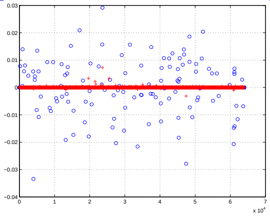
Acceleration by Randomization: how it works



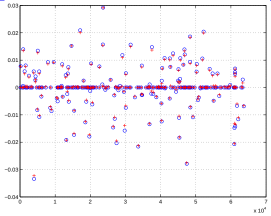
DMP, $p = 2$



SMP, $p = 2$



DMP, $p = \infty$



SMP, $p = \infty$

DMP-based (left) and SMP-based (right) recovery of sparse signals in the $32,000 \times 64,000$ experiments.

Circles: true signal Crosses: recovery