On Recent Trends in Large-Scale Convex Optimization

Arkadi Nemirovski
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Joint research with Anatoli Juditsky[†], Guanghui Lan[‡], and Alexander Shapiro[§]
[†]: Joseph Fourier University, Grenoble, France; [‡]: ISyE, University of Florida;
[§]: ISyE, Georgia Tech

NIPS 2009 Workshop
Optimization for Machine Learning
December 12, 2009

- Practical scope of Interior Point Methods
- Computationally cheap First Order Methods: Limits of performance
- Utilizing problem's structure: saddle point reformulations of convex minimization problems
- Acceleration of First Order methods by randomization
- How it works

“Practical scope” of IPMs

♣ Theoretically, Convex Programming is within the grasp of Polynomial Time Interior Point Methods allowing to get a high accuracy solution in few tens of iterations.

But: The arithmetic cost of IPM iteration grows *nonlinearly* with the design dimension, and eventually becomes prohibitively large — a single iteration lasts “forever.”

♠ What “eventually” actually means, it depends on problem’s structure:

- LPs of decision-making origin have extremely sparse constraint matrices, which allows to handle *hundreds of thousands* of variables and constraints;
- LPs arising in Machine Learning and Signal Processing may have dense constraint matrices, and here already *few thousands* of variables and constraints could be “too many;”
- Typical SDPs with *few thousands* of variables are beyond the “practical grasp” of IPMs.

First Order Methods

♣ At the present state of our knowledge, the "methods of choice" when solving convex programs which are beyond the grasp of IPMs are *First Order methods* with computationally cheap iterations.

Example: IPM vs. First Order method on two SVM instances:

		Validation error & CPU	
$m \times n$	Density	**IPM**	**FOM**
4,000×2,000	0.1	2.6% & 2963 sec	2.6% & 141 sec
5,000×15,000	0.04	failure	7.9% & 49 sec

m: # of features; n: training sample cardinality

♣ **Good news on FOMs:** *When solving large-scale convex problems* $\min_X f(x)$ *on domains* X *with "favourable geometry," good FOMs exhibit (nearly) dimension-independent rate of convergence*.

♣ **Not so good news on FOMs:** *The rate of convergence of FOMs in the large scale case is only sublinear*.

FOMs: Limits of performance

$$\min_{x \in X} f(x) \qquad (P)$$

- $X \subset \mathbb{R}^n$: convex compact of $\|\cdot\|_2$-diameter R
- f: convex and Lipschitz continuous

♣ FOMs are capable to solve (P) within accuracy ϵ in

- $O(1)\left(LR^2/\epsilon\right)^{\frac{1}{2}}$ steps, if f is smooth:
$$\|f'(x) - f'(y)\|_2 \leq L\|x - y\|_2$$
- $O(1)\left(MR/\epsilon\right)^2$ steps, when f is not so smooth:
$$\|f'(x) - f'(y)\|_2 \leq M$$

with a step reducing to computing f, f' at a point, plus some overhead. For "simple" X, the overhead is *just linear in n*.

♣ When f is given by a black box routine computing f, f', *these rates of convergence are unimprovable in the large scale case.*

♣ Measuring the diameter of X in $\|\cdot\|_1$, and regularity of f' in $\|\cdot\|_\infty$, the step counts remain "nearly intact" – they get extra *logarithmic in n* factor.

FOMs: Limits of performance

$$\min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♠ All known FOMs utilize black box representation of f

$\Rightarrow$When n is large, the iteration counts are *at best* $O(1)\left(LR^2/\epsilon\right)^{\frac{1}{2}}$ in the smooth and $O(1)\left(MR/\epsilon\right)^2$ in the nonsmooth cases.

Consequences:

• Boundedness of the feasible domain becomes of paramount importance — *the complexity rapidly blows up to* ∞ *as* $R \to \infty$*!*

$\Rightarrow$*"Penalized" settings, like in LASSO:*

$$\min_{x \in \mathbb{R}^n} \left[\|Ax - b\|^2 + \lambda \|x\|_1\right]$$

are essentially worse than "bounded domain" settings like

$$\min_{\|x\|_1 \le \rho} \|Ax - b\|$$

• *FOMs are applicable when medium accuracy solutions are sought and are especially attractive in the large-scale "favourable geometry" case where the methods exhibit (nearly) dimension independent convergence rate.*

From nonsmooth minimization to smooth convex-concave saddle points

$$\min_{x \in X} f(x) \qquad (P)$$

♣ Smooth problems are much better suited for FOMs than nonsmooth ones. Unfortunately, problems with simple X and smooth f are rare commodity...

♣ **Crucial observation [Nesterov '03]:** Nonsmoothness in Convex Optimization arises primarily from taking maxima, so that the objective f can be usually represented as

$$f(x) = \max_{y \in Y} \phi(x, y)$$

with *smooth* (usually, just bilinear) convex-concave $\phi(x, y)$.

$\Rightarrow$*(P) can be reduced to the saddle point problem*

$$\min_{x \in X} \max_{y \in Y} \phi(x, y) \qquad (S)$$

with smooth convex-concave cost function and then solved by properly designed computationally cheap FOMs with iteration count $O(1/\epsilon)$.

Examples of saddle form reformulations

- **Minimizing the maximum of smooth convex functions:**

$$\min_{x\in X} \max_{1\le i\le n} f_i(x)$$
$$\Leftrightarrow \min_{x\in X} \max_{y\in Y} \textstyle\sum_i y_i f_i(x),\ Y = \{y \ge 0, \sum_i y_i = 1\}$$

- **Minimizing maximal eigenvalue:**

$$\min_{x\in X} \lambda_{\max}(\textstyle\sum_i x_i A^i)$$
$$\Leftrightarrow \min_{x\in X} \max_{y\in Y} \mathrm{Tr}(Y[\textstyle\sum_i x_i A^i]),\ Y = \{y \succeq 0, \mathrm{Tr}(y) = 1\}$$

- ℓ_1 **minimization** $\min_x\{\|x\|_1 : \|Ax - b\|_p \le \epsilon\}$ reduces to a small series of parametric problems

$$\min_x\{\|RAx - b\|_p : \|x\|_1 \le 1\}$$
$$\Leftrightarrow \min_{x:\|x\|_1\le 1} \max_{y:\|y\|_q\le 1} y^T(RAx - b),\ \tfrac{1}{p} + \tfrac{1}{q} = 1$$

Examples of saddle form reformulations

• **Nuclear norm minimization** $\min_{x\in\mathbb{R}^{m\times n}}\{\|x\|_* : \|\mathcal{A}(x)-b\|_p \le \epsilon\}$ with linear $\mathcal{A}(\cdot)$ reduces to small series of parametric problems

$$\min_{x\in\mathbb{R}^{m\times n}}\{\|R\mathcal{A}(x)-b\|_p : \|x\|_* \le 1\}$$

$$\Leftrightarrow \min_{x:\|x\|_*\le 1}\max_{y:\|y\|_q\le 1} y^T(R\mathcal{A}(x)-b),\ \frac{1}{p}+\frac{1}{q}=1$$

♣ **The SDP relaxation of the problem of low-dimensional uniform approximation** *"Given N unit vectors $a_i \in \mathbb{R}^n$ and k, find k-dimensional subspace L minimizing* $\max_i \mathrm{dist}_{\|\cdot\|_2}(a_i, L)$*"* reduces to the saddle point problem

$$\min_{\substack{P:0\preceq P\preceq I,\\ \mathrm{Tr}(P)\le k}}\ \max_{\lambda\ge 0:\sum_i\lambda_i=1} \sum_i \lambda_i a_i^T P a_i$$

Examples of saddle form reformulations

- **The parametric SVM problem**

$$\min_{\alpha,b,\xi}\left\{\sum_{i,j}K(x_i,x_j)y_iy_j\alpha_i\alpha_j+R^{-2}\|\xi\|_p:\ \xi_i\geq\max\left[0,1-y_i\left(\sum_j K(x_i,x_j)y_j\alpha_j+b\right)\right]\right\}$$

R being the parameter, can be re-parameterized as

$$\min_{\alpha,b,\xi}\left\{\|\xi\|_p:\ \begin{array}{l}\|\alpha\|:=\sqrt{\sum_{i,j}K(x_i,x_j)y_iy_j\alpha_i\alpha_j}\leq\rho\\ \xi_i\geq\max\left[0,1-y_i\left(\sum_j K(x_i,x_j)y_j\alpha_j+b\right)\right]\end{array}\right\}$$

ρ being the parameter, and then reduces to the saddle point form

$$\min_{\beta:\|\beta\|\leq 1}\ \max_{\substack{\lambda:\lambda\geq 0,\\ \|\lambda\|_q\leq 1,\sum_i y_i\lambda_i=0}}\left[\sum_i\lambda_i-\rho\lambda^TQ\beta\right],$$

$$Q=[y_iK(x_i,x_j)y_j]_{i,j},\ 1/q+1/p=1$$

- **"Plain" SVM problem with** $K(x_i,x_j)=x_i^Tx_j$ reduces to

$$\min_{w:\|w\|\leq 1}\ \max_{\substack{\lambda:\lambda\geq 0,\\ \|\lambda\|_q\leq 1,\sum_i y_i\lambda_i=0}}\left[\sum_i\lambda_i-\rho\lambda^TP^Tw\right]$$

$$P=[y_1x_1,...,y_Lx_L],\ 1/q+1/p=1$$

Solving convex-concave saddle point problems by FOMs

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad (S)$$

- X, Y : convex compacts
- ϕ : convex-concave and Lipschitz

- **Accuracy measure for (S):**

$$\epsilon(x,y)=\max_{y'\in Y}\phi(x,y')-\min_{x'\in X}\phi(x',y)$$

Explanation: (S) gives rise to primal-dual pair of optimization problems with equal optimal values:

$$\begin{aligned}\mathrm{Opt}(P) &= \min_{x\in X}\overline{\phi}(x),\ \overline{\phi}(x)=\max_{y\in Y}\phi(x,y)\\ \mathrm{Opt}(D) &= \max_{y\in Y}\underline{\phi}(y),\ \underline{\phi}(y)=\min_{x\in X}\phi(x,y)\end{aligned}$$

$\epsilon(x,y)$ is the sum of non-optimalities, in terms of the respective objectives, of x as a solution to (P) and y as a solution to (D).

Solving saddle point problems by FOMs

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad (S)$$

♣ **Claims:**

- *When ϕ is convex-concave and $\nabla\phi$ is Lipschitz continuous, (S) can be solved by a good FOM within accuracy $\epsilon>0$ at iteration count $O(1/\epsilon)$, with iteration reducing to $O(1)$ computations of $\nabla\phi$ at a point plus some computational overhead.*
  "Good" FOMs include, among others,
  Smoothing method [Nesterov '03]
  Mirror-Prox algorithm [Nem. '04].

- *When X and Y possess "favourable geometry," the iteration count is (nearly) dimension-independent.*

- *There are important situations where FOMs can be further accelerated by randomization — by replacing computationally expensive in the large scale case precise gradients of ϕ with their computationally cheap unbiased stochastic estimates.*

Complexity of Mirror-Prox

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad (S)$$

♣ Let $X\times Y := Z\subset Z^+$, where Z^+ is the direct product of K "basic blocks":

- K_b unit Euclidean balls $B_i\subset E_i=\mathbb{R}^{n_i}$;
- K_s *spectahedrons* $S_j\subset F_j$;

  F_j: spaces of symmetric $k_j\times k_j$ matrices of a given block-diagonal structure with the Frobenius inner product

  S_j: the set of all $\succeq 0$- matrices from F_j with unit trace

♣ **Note:** • The standard simplex $\Delta_n=\{x\in\mathbb{R}^n_+ : \sum_i x_i=1\}$ is a spectahedron;

• The unit ℓ_1 ball $\{x\in\mathbb{R}^n : \|x\|_1\}$ is the image of Δ_{2n} under the mapping $[u;v]\mapsto u-v$

$\Rightarrow$When (S) involves ℓ_1-balls (as in ℓ_1 minimization), these balls can be replaced with the standard simplexes;

• Similarly, when (S) involves $\|\cdot\|_*$-balls (as in nuclear norm minimization), these balls can be replaced with spectahedrons.

Complexity of Mirror-Prox

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad (S)$$

$$X\times Y := Z\subset Z^+ = B_1\times\ldots\times B_{K_b}\times S_1\times\ldots\times S_{K_s}$$

- $B_i\subset E_i=\mathbb{R}^{n_i}$: Euclidean balls
- $S_j\subset F_j$: spectahedrons

♣ We equip the embedding space $E = E_1\times\ldots\times E_{K_b}\times F_1\times\ldots\times F_{K_s}$ of $Z = X\times Y$ with the norm

$$\|z\| = \sqrt{\sum_{i=1}^{K_b}\|z^i\|_2^2+\sum_{j=1}^{K_s}\|z_j\|_{\mathrm{tr}}^2}$$

- z^i: ball components of z • z_j: spectahedron components of z
- $\|\cdot\|_{\mathrm{tr}}$: trace norm of a symmetric matrix.

♠ The conjugate norm is

$$\|\zeta\|_* = \sqrt{\sum_{i=1}^{K_b}\|\zeta^i\|_2^2+\sum_{j=1}^{K_s}\|\zeta_j\|_{2,2}^2},$$

where $\|A\|_{2,2}$ is the spectral norm of a matrix.

Complexity of Mirror-Prox (continued)

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad (S)$$
$$X\times Y := Z\subset Z^+ = B_1\times\ldots\times B_{K_b}\times S_1\times\ldots\times S_{K_s}$$

Theorem [Jud. & Nem., '08]

Assume that

- *ϕ is convex-concave function with the gradient satisfying*
$$\|\nabla\phi(z)-\nabla\phi(z')\|_*\le L\|z-z'\|+M$$
- *we have access to Stochastic Oracle reporting unbiased estimates of $\nabla\phi$. Specifically, at ℓ-th call to SO, z being the query point, the oracle returns $G(z,\xi_\ell)$, where $\xi_1,\xi_2,\ldots$ are iid "oracle noises" such that*

$$\mathbf{E}_\xi\{G(z,\xi)\}\equiv\nabla\phi(z),\ \mathbf{E}_\xi\{\|G(z,\xi)-\nabla\phi(z)\|_*^2\}\le M^2$$

Then for every $N=1,2,\ldots$, the N-step Mirror Prox algorithm generates a random solution $z^N\in Z$ to (S) such that

$$\mathbf{E}\{\epsilon(z^N)\}\le O(1)\Theta\left[\frac{\Theta L}{N}+\frac{M}{\sqrt{N}}\right]$$

$$\Theta=\left[K_b+\sum_{j=1}^{K_s}\ln k_j\right]^{\frac{1}{2}},\ k_j:\ \text{sizes of spectahedron blocks}$$

Complexity of Mirror-Prox (continued)

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad (S)$$
$$X\times Y := Z\subset Z^+ = B_1\times\ldots\times B_{K_b}\times S_1\times\ldots\times S_{K_s}$$

Theorem (continued)

$$\mathbf{E}\{\epsilon(z^N)\}\le O(1)\Theta\left[\frac{\Theta L}{N}+\frac{M}{\sqrt{N}}\right]$$
$$\Theta=\left[K_b+\sum_{j=1}^{K_s}\ln k_j\right]^{\frac{1}{2}},\ k_j:\ \text{sizes of spectahedron blocks}$$

The effort per step reduces to two calls to SO plus $\mathcal{C}$-a.o. When Z is cut off Z^+ by $O(1)$ linear inequalities, one has

$$\mathcal{C}=O(1)\big[\dim Z^+ + \text{ effort for eigenvalue decomposition of a matrix from } S_1\times\ldots\times S_{K_s}\ \big]$$

♠ When K_b and K_s are fixed, the efficiency estimate is nearly dimension independent. If Z is product of $O(1)$ balls, this efficiency is *unimprovable in the large-scale case*.

♠ When Z is cut off Z^+ by $O(1)$ linear inequalities and all S_j are simplexes, $\mathcal{C}$ is *just linear* in $\dim Z$.

Mirror Prox: large deviations

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad (S)$$
$$\|\nabla\phi(z)-\nabla\phi(z')\|_*\le L\|z-z'\|+M$$
$$X\times Y:=Z\subset Z^+=B_1\times...\times B_{K_b}\times S_1\times...\times S_{K_s}$$
$$\mathbf{E}_\xi\{G(z,\xi)\}=\nabla\phi(z)$$

♠ Assuming light tails in the distribution of stochastic gradients:

$$\mathbf{E}_\xi\left\{\exp\{\|G(z,\xi)-\nabla\phi(z)\|_*^2/M^2\}\right\}\le\exp\{1\}$$

"large deviations" in the quality of the approximate solution z^N generated by N-step MP admit exponential bounds: *for all* $\Omega>0$ *it holds*

$$\mathrm{Prob}\left\{\epsilon(z^N)>O(1)\left[\Theta\left[\frac{\Theta L}{N}+\frac{M}{\sqrt{N}}\right]+\frac{\Omega\Theta M}{\sqrt{N}}\right]\right\}$$
$$\le\exp\{-\Omega^2/3\}+\exp\{-\Omega N\}$$

Randomization

♣ Consider saddle point problem with *bilinear* cost function

$$\min_{x\in X}\max_{y\in Y}\phi(x,y) := x^T a + y^T b + y^T A x \qquad (S)$$

When solving (S) by deterministic FOMs, computing $\nabla\phi(\cdot)$ requires two matrix-vector multiplications. When A is dense and large-scale, these multiplications become time consuming.

♠ Matrix-vector multiplication $(B,x)\mapsto Bx$ is easy to randomize:

♡ Treat the vector $\mathrm{abs}(x)/\|x\|_1$ as the probability distribution on the set of indices $1,\ldots,\dim x$ and draw from it a random index $\jmath$;

♡ Return $\xi = \|x\|_1 \mathrm{sign}(x_\jmath) B_\jmath$, where B_j is j-th column of B.

- We have $\mathbf{E}\{\xi\} = Bx$ and $\|\xi\| \le \|x\|_1 \max_j \|B_j\|$.
- The arithmetic cost of a call to the resulting SO is just linear in the sizes of B.
- We can call the SO several times and average the answers, thus reducing the variance of the unbiased estimate of Bx.

Randomization: ℓ_1 minimization with uniform fit

$$\min_{x:\|x\|_1\le 1} \|Ax-b\|_\infty \Leftrightarrow \min_{x:\|x\|_1\le 1} \max_{y:\|y\|_1\le 1} y^T[Ax-b] \qquad (S)$$
$$A: m\times n, \quad L := \max_{i,j}|A_{ij}|$$

♣ For m, n large, the best known complexity of finding ϵ-solution by *deterministic* methods is $O(1)\ln(m+n)L/\epsilon$ steps, with computing two involving A matrix-vector products per step.
$\Rightarrow$Total effort $O(1)\ln(mn)mnL/\epsilon$ a.o.

♠ Randomizing matrix-vector multiplications and applying *Stochastic* version of *Mirror Prox*, ϵ-solution can be found with confidence $\ge 1-\beta$ in $O(1)\ln(mn/\beta)(L/\epsilon)^2$ steps, with extracting from A randomly chosen row and column per step.
$\Rightarrow$Total effort $O(1)\ln(mn/\beta)(m+n)(L/\epsilon)^2$ a.o.
♡ *With ϵ, L fixed and $m = O(n)$ large, SMP*
- *outperforms by orders of magnitude deterministic methods*
- *exhibits sublinear time behaviour: ϵ-solution is found by inspecting negligible part of the data,* cf. Matrix Game algorithm of Grigoriadis & Khachiyan '95.

Randomization: ℓ_1 minimization with Least Squares fit

$$\min_{x:\|x\|_1\le 1} \|Ax-b\|_2 \Leftrightarrow \min_{x:\|x\|_1\le 1}\max_{y:\|y\|_2\le 1} y^T[Ax-b] \qquad (S)$$
$$A: m\times n, \quad L := \max_j \|A_j\|_2$$

♣ For m, n large, the best known complexity of finding ϵ-solution by *deterministic* methods is $O(1)\ln(mn)mnL/\epsilon$ a.o.

♠ Randomizing matrix-vector multiplications and applying SMP, ϵ-solution can be found with confidence $\ge 1-\beta$ at the cost of

$$O(1)\left[mn\ln(m)+\ln^2(mn/\beta)(m+n)(L/\epsilon)^2\right] \text{ a.o.}$$

• $O(1)mn\ln(m)$: cost of preprocessing $[A,b]\hookleftarrow[HDA,HDb]$ with Hadamard matrix H and $D=\mathrm{Diag}\{\xi_1,...,\xi_m\}$ with random iid $\xi_i=\pm 1$ aimed at making the magnitudes of entries of A "small" – $O(1)\sqrt{\ln(mn)/m}$.

Randomization: minimizing maximum of convex polynomials over a simplex

♣ ℓ_1-minimization problem with uniform fit is nothing but the problem of minimizing the maximum of m affine forms over the standard simplex $\Delta_n = \{x \in \mathbb{R}^n_+ : \sum_i x_i = 1\}$. The approach we have used can be extended to the problem

$$\min_{x\in\Delta_n} \max_{1\le i\le m} p_i(x) \qquad (P)$$

of minimizing over the simplex the maximum of *convex* polynomials $p_i(x) = \sum_{\ell=0}^{d} P^{i\ell}[x, ..., x]$.

♠ Let the moduli of coefficients of the symmetric ℓ-linear forms $P^{i\ell}$ be $\le L$.

A properly designed randomized SMP-based algorithm finds with confidence $\ge 1-\beta$ an ϵ-solution to (P) at the cost of $O(1)\ln(mn/\beta)d^2(L/\epsilon)^2$ steps, with a step reducing to extracting $O(1)d(m+n)$ coefficients of the forms $P^{i\ell}$, given the randomly selected "addresses" $i, \ell, j_1, ..., j_\ell$ of the coefficients.

How it works: ℓ_1-minimization by Deterministic MP

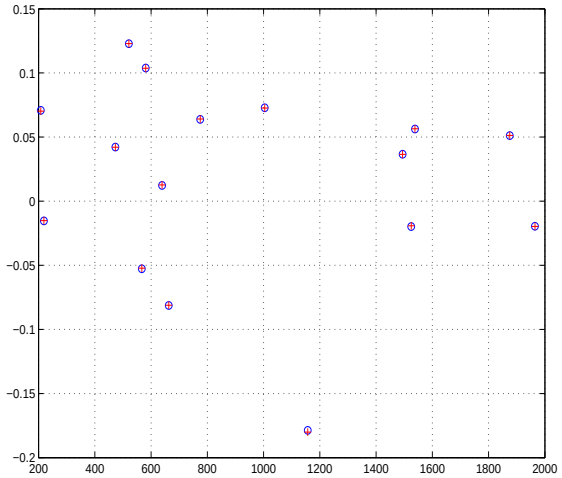
$$\widehat{x} \approx \operatorname*{argmin}_x \{\|Ax - b\|_\infty : \|x\|_1 \le 1\}$$

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

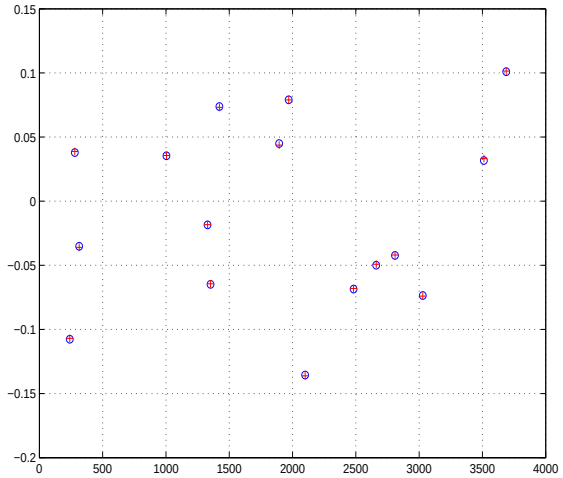
		Errors			CPU
$m \times n$	Method	$\|x_* - \widehat{x}\|_1$	$\|x_* - \widehat{x}\|_2$	$\|x_* - \widehat{x}\|_\infty$	sec
512×2048	DMP	0.0052	0.0018	0.0013	3.3
	IP	0.039	0.0061	0.0021	321.6
1024×4096	DMP	0.0096	0.0028	0.0015	3.5
	IP	Out of space (2GB RAM)			
4096×16384	DMP	0.0057	0.0026	0.0024	46.4
	IP	not tested			

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

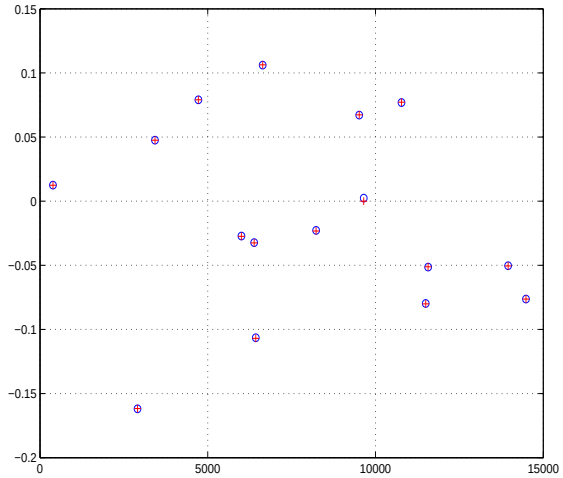
How it works: ℓ_1-minimization by Deterministic MP



512×2048



1024×4096



4096×16384

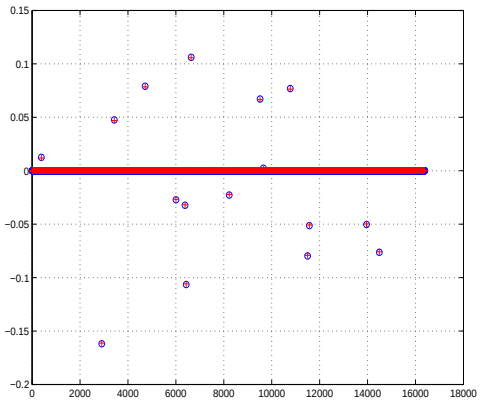
ℓ_1-recovery: o: true signal, $+$: DMP recovery

How it works: ℓ_1-minimization by Deterministic MP

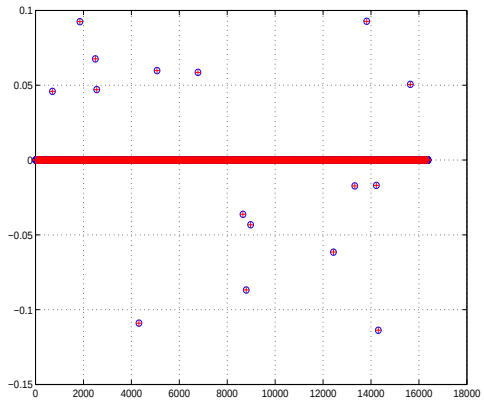
$$\widehat{x} \approx \operatorname*{argmin}_x \left\{ \|x\|_1 : \|Ax - b\|_p \leq \tfrac{3}{2}\delta \right\}$$

A: random 4096×16384 submatrix of D.F.T. matrix

b: $\|Ax_* - b\|_p \leq \delta = 5.e\text{-}3$ with 16-sparse x_*, $\|x_*\|_1 = 1$



Uniform fit



Least Squares fit

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

Fit	Errors			CPU, sec
	$\|x_* - \widehat{x}\|_1$	$\|x_* - \widehat{x}\|_2$	$\|x_* - \widehat{x}\|_\infty$	
Uniform	0.011	0.0021	0.00062	82.5
Least squares	0.0023	0.00015	0.000044	55.0

How it works: Verifiable Sufficient Conditions in Compressed Sensing via Deterministic MP

- A verifiable sufficient condition for ℓ_1-recovery

$$x \mapsto y = Ax \mapsto \widehat{x} = \mathrm{argmin}_u \{\|u\|_1 : Ax = y\} \qquad [A : m \times n]$$

to be exact for all signals x with at most s nonzero entries reads:

$$\mathrm{Opt} := \min_Y \left\{\max_j \|[I - Y^T A]_j\|_{s,1}\right\} < \tfrac{1}{2}$$

$\|u\|_{s,1}$: the sum of s largest magnitudes of entries in u

♠ Computing Opt reduces to solving an LP with *quadratic* in n number of variables and constraints. With $m = 240$ and $n = 256$, this LP has 131,328 constraints, 127,232 variables and 16,908,800 nonzeros in the constraint matrix

⇒With 2GB RAM, IP solver `mosekopt` runs out of memory .

♠ Computing Opt reduces to solving bilinear saddle point problem with $\dim X = 127,232$, $\dim Y = 131,072$.

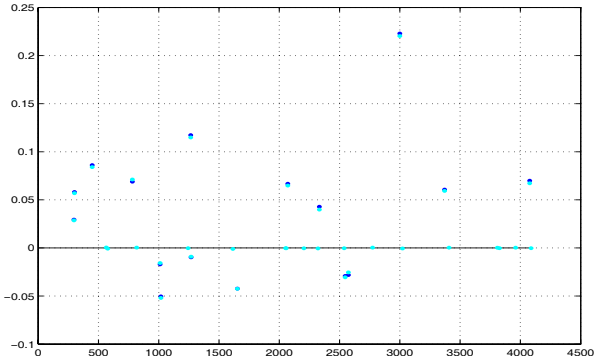
DMP solves the problem within accuracy 0.0017 in 5,088 calls to the deterministic First Order oracle, CPU= $1^h 13' 19''$.

How it works: SMP vs. DMP in ℓ_1-minimization

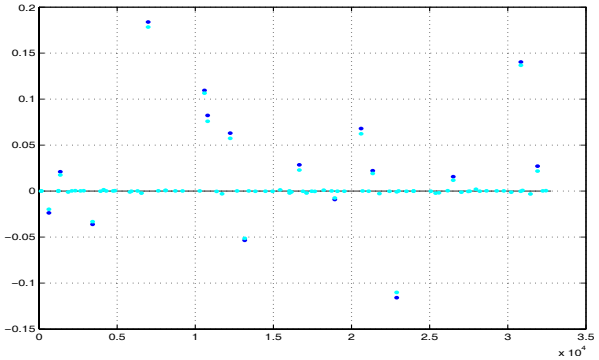
$$\widehat{x} \approx \mathrm{argmin}_x \{\|Ax - b\|_\infty : \|x\|_1 \le 1\}$$

A: dense analytically given $m \times n$ matrix

b: $\|Ax_* - b\|_\infty \le \delta = 1.e\text{-}3$ with 16-sparse x_*, $\|x_*\|_1 = 1$



2048×4096



8192×32768

ℓ_1-recovery: Blue: true signal, Cyan: SMP

		Errors			CPU	
$m \times n$	Method	$\|x_* - \widehat{x}\|_1$	$\|x_* - \widehat{x}\|_2$	$\|x_* - \widehat{x}\|_\infty$	sec	Mult
2048×4096	DMP	0.0014	0.00052	0.00036	122.8	1770
	SMP	0.039	0.0079	0.0030	325.4	29.3
8192×32768	DMP	1.006	0.319	0.184	3141.9	5
	SMP	0.120	0.0196	0.00634	3000.5	4.7

Mult: equivalent # of matrix-vector multiplications

How it works: DMP vs. SMP on Low-dimensional approximation

♣ **Problem:** *Given* $100,000$ *unit vectors* $a_i \in \mathbb{R}^{100}$ *known to be at a distance* ≤ 0.2 *from a* 10*-dimensional subspace of* $\mathbb{R}^{100}$*, find such a subspace.*

♠ **Solution** via SDP relaxation:

$$\max_P \left\{ \min_i a_i^T P a_i : 0 \leq P \leq I, \mathrm{Tr}(P) = 10 \right\} \Leftrightarrow \max_{\substack{Q: 0 \leq Q \leq 0.1I, \\ \mathrm{Tr}(Q)=1}} \min_{\substack{\lambda \geq 0, \\ \sum_i \lambda_i = 1}} \sum_i \lambda_i a_i^T Q a_i$$

- Approximating plane: span of 10 leading eigenvectors of Q.
- Termination when the maximal deviation of a_i from the current approximating plane becomes ≤ 0.21.

Method	Oracle calls	Mult	MaxDev	CPU, sec
DMP	17	17	0.200	507.5
SMP	742	0.31	0.204	37.2

Oracle calls: # of calls to deterministic (DMP) or stochastic (SMP) oracle
Mult: equivalent # of matrix-vector multiplications