ParNes: A New Algorithm for Compressed Sensing Problems

Cinna Wu

Department of Mathematics
UC Berkeley

Joint work with Ming Gu and Lek-Heng Lim

$$\boxed{\min \|x\|_1 \ \textbf{s.t.} \ \|Ax - b\|_2 \leq \sigma}$$

Contents

- **Background**
  - Motivation
  - Applications
  - Solvers
- **ParNes**
  - The Algorithm
  - Convergence
  - Numerical Experiments

Sparse Signal Recovery

- **Classical approach: Sample then compress.**

$$\boxed{f = Bx_0}$$

  - $B \in \mathbb{R}^{n\times n}$: **compression matrix**
  - $f \in \mathbb{R}^n$: **sampled signal**
  - $x_0 \in \mathbb{R}^n$: **sparse compressed signal.**

- **Compressed Sensing: Sample and compress in one stage.**

$$\boxed{b = Mf = MBx_0 = Ax_0}$$

  - $M \in \mathbb{R}^{m\times n}$: **measurement matrix with** $m < n$
  - $b \in \mathbb{R}^m$: **measurements**

$$\boxed{\textbf{Can we recover } x_0 \textbf{ given } A \textbf{ and } b\textbf{?}}$$

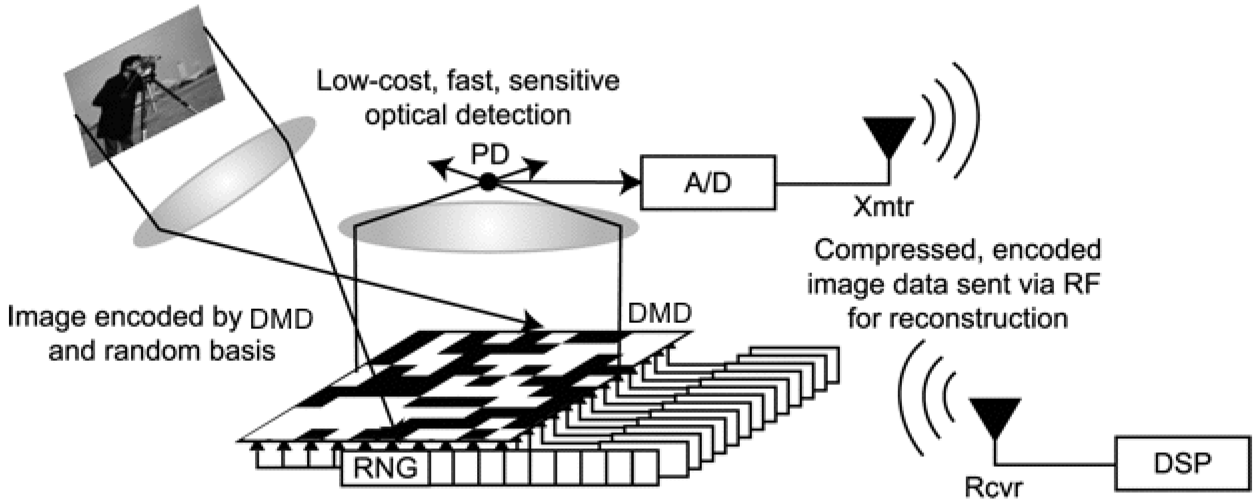
Applications of Compressed Sensing

Compressed sensing may be useful when...

- signals are sparse in a known basis.
- measurements are expensive but computations are cheap.

- **Magnetic Resonance Imaging (MRI):**
  - Lengthy procedure! Needs a large number of measurements of the patient.
  - Compressed sensing can reduce the number of measurements.
  - This could reduce the procedure time or produce better images in the same amount of time.

Rice Single Pixel Camera[1]



$$\boxed{b = Mf = MBx_0 = Ax_0}$$

- $b \in \mathbb{R}^m$: **measurements**
- $M \in \mathbb{R}^{m \times n}$: **measurement matrix with $m < n$, rows determined by the digital micromirror device (DMD)**
- $f \in \mathbb{R}^n$: **the image we wish to recover**
- $x_0 \in \mathbb{R}^n$: **sparse representation of f under the basis given by B.**

[1] Image courtesy of Rice University.

Recovering the Sparse Signal

- **We can try to recover the sparse signal with**

$$\boxed{\min \|x\|_0 \ \textbf{s.t.} \ Ax = b}$$

  – $\|x\|_0$ **: number of nonzero coefficients in** x**.**

  – **Combinatorial and NP-hard!**

- **Relax to the Basis Pursuit (BP) problem:**

$$\boxed{\min \|x\|_1 \ \textbf{s.t.} \ Ax = b}$$

  – **This can recovers the sparse signal!**

    * ***Mutual coherence*** **of** A**: Given** A**, works if the signal is sufficiently sparse. (Donaho, Elad, Huo, etc)**

    * **Given the sparsity of the signal, depends on the *restricted isometry constants* of** A**. (Candès, Romberg, Tao)**

ℓ_1-relaxations for Noisy Measurements

Recover the sparse vector x when $Ax \approx b$.

- **Basis pursuit denoise (BP$_\sigma$)**

$$\boxed{\min \|x\|_1 \ \ \textbf{s.t.} \ \ \|Ax - b\|_2 \leq \sigma}$$

- **Penalized least squares (QP$_\lambda$)**

$$\boxed{\min \|Ax - b\|_2^2 + \lambda \|x\|_1}$$

- **Lasso Problem (LS$_\tau$)**

$$\boxed{\min \|Ax - b\|_2 \ \ \textbf{s.t.} \ \ \|x\|_1 \leq \tau}$$

Solutions coincide for appropriate choices of σ, λ, τ.

- **Many solvers use this relationship.**

Solvers

- **PDCO - Primal-Dual IP method for Convex Objectives (Saunders, Kim):**
  - Solves Basis Pursuit

$$\boxed{\min \|x\|_1 \ \textbf{s.t.} \ Ax = b}$$

  by solving an equivalent Linear Program.

- **HOMOTOPY (Osborne, Presnell, Turlach):**
  - Solves a sequence of QP_λ problems to solve BP_σ.

$$\boxed{\min \|x\|_1 \ \textbf{s.t.} \ \|Ax - b\|_2 \leq \sigma}$$

- **FPC - Fixed Point Continuation Method (Hale, Yin, Zhang):**
  - Uses a version of fixed point iteration to solve QP_λ.

$$\boxed{\min \|Ax - b\|_2^2 + \lambda \|x\|_1}$$

Solvers

- **SPGL1 - spectral gradient-projection method (Berg, Friedlander):**
  - **Solves a sequence of LS_τ problems**

$$\boxed{\min \|Ax - b\|_2 \ \ \textbf{s.t.} \ \ \|x\|_1 \leq \tau}$$

to solve BP_σ

$$\boxed{\min \|x\|_1 \ \ \textbf{s.t.} \ \ \|Ax - b\|_2 \leq \sigma}$$

- **NESTA (Becker, Bobin, Candès):**
  - **Uses a method to minimize non-smooth functions proposed by Yu. Nesterov to solve BP_σ.**

Our algorithm, ParNes, combines the ideas used in NESTA and SPGL1.

Comparison of Solvers

- **Comparison of HOMOTOPY, PDCO, SPGL1 [2].**
- **Two 3GHz CPU's, 4Gb RAM. Problems from the SPARCO toolbox.**
  - $\star$: **solver failed to converge in the allowed CPU time (1 hour)**
  - $nz(x)$: **number of "nonzero" entries of x above some tolerance**
  - r : **residual.**

Problem Data		**PDCO**			**HOMOTOPY**			**SPGL1**		
Problem	size A	$\|r\|_2$	$\|x\|_1$	$nz(x)$	$\|r\|_2$	$\|x\|_1$	$nz(x)$	$\|r\|_2$	$\|x\|_1$	$nz(x)$
blocksig	1024×1024	3.3e-4	4.5e+2	703	1.0e-4	4.5e+2	246	2.0e-14	4.5e+2	21
blurrycam	65536×65536	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	9.9e-5	1.0e+4	8237
blurspike	16384×16384	9.1e-3	3.4e+2	5963	$\star$	$\star$	$\star$	9.9e-5	3.5e+2	5066
cosspike	1024×2048	1.6e-4	2.2e+2	2471	1.0e-4	2.2e+2	500	8.6e-5	2.2e+2	111
sgnspike	600×2560	9.3e-6	2.0e+1	131	1.0e-04	2.0e+1	80	8.0e-5	2.0e+1	56
seismic	41472×480617	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	8.6e-5	3.9e+3	3871

[2] Probing the Pareto Frontier for Basis Pursuit Solutions. E. Berg, M. Friedlander. 2008.

An Outline of ParNes

Combines the best features of NESTA and SPGL1 to solve BP_σ

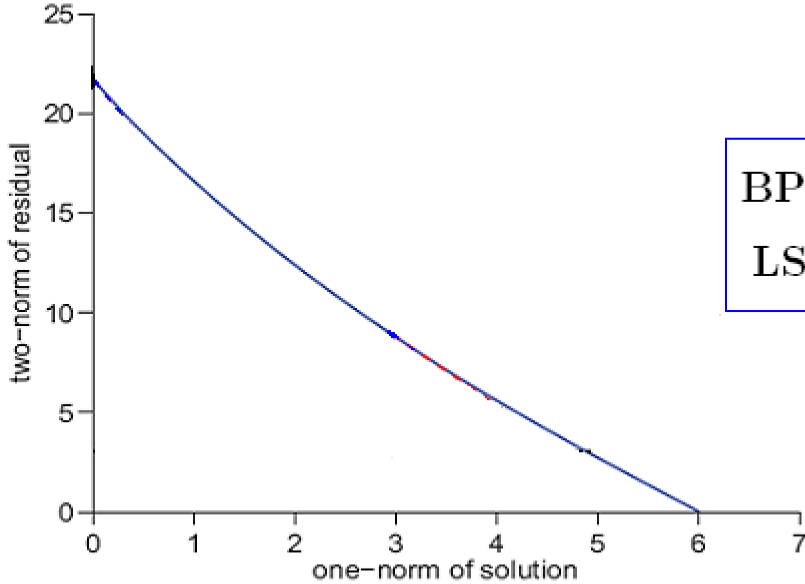
$$\boxed{\min \|x\|_1 \ \ \textbf{s.t.} \ \ \|Ax - b\|_2 \leq \sigma}$$

- **SPGL1**
  - **Like SPGL1, ParNes solves BP_σ by solving a sequence of LS_τ problems.**

$$\boxed{\min \|Ax - b\|_2 \ \ \textbf{s.t.} \ \ \|x\|_1 \leq \tau}$$

  - **LS_τ and BP_σ are related by the Pareto Curve.**
  - **In SPGL1, LS_τ is solved with a spectral projected gradient method.**
- **NESTA**
  - **Uses a method by Y. Nesterov to minimize non-smooth functions.**
  - **ParNes solves the LS_τ problems with a similar method for minimizing smooth functions.**

The Pareto Curve

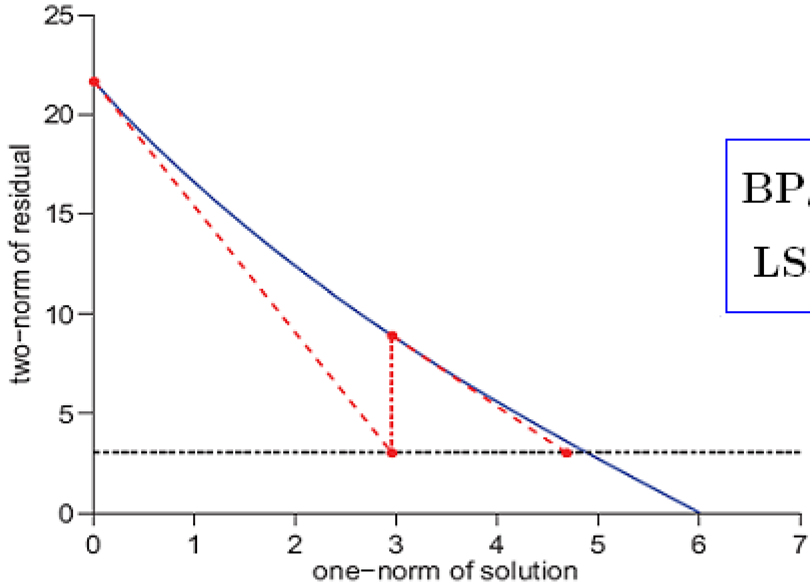


BP$_\sigma$**:** $\min \|x\|_1$ **s.t.** $\|Ax - b\|_2 \leq \sigma$

LS$_\tau$**:** $\min \|Ax - b\|_2$ **s.t.** $\|x\|_1 \leq \tau$

- **Convex, continuously differentiable, and strictly decreasing.**
- **Graph of** $(\|x_\tau\|_1, \|b - Ax_\tau\|_2)$ **where** x_τ **solves LS**$_\tau$**.**
- **Also the graph of** $(\|x_\sigma\|_1, \|b - Ax_\sigma\|_2)$ **where** x_σ **solves BP**$_\sigma$**.**
- **Since** $\|x_\tau\|_1 = \tau$ **and** $\|b - Ax_\sigma\|_2 = \sigma$**, the Pareto curve is the graph of a function** $\phi(\tau) = \sigma$**.**

Root Finding



$$\text{BP}_\sigma\text{: } \min \|x\|_1 \ \text{ s.t. } \ \|Ax-b\|_2 \le \sigma$$

$$\text{LS}_\tau\text{: } \min \|Ax-b\|_2 \ \text{ s.t. } \ \|x\|_1 \le \tau$$

- **BP_σ can be solved by finding a root τ_σ to $\phi(\tau) = \sigma$.**
- **Newton's method can be applied to $\phi(\tau) = \sigma$ to get $\tau_k \to \tau_\sigma$:**

$$\boxed{\tau_{k+1} = \tau_k + (\sigma - \phi(\tau_k))/\phi'(\tau_k)}$$

- **Since each iteration involves solving a potentially large LS_{τ_k} problem, an inexact Newton method is used.**

Solving $\mathbf{LS}_{\tau_k}$: $\min \|Ax - b\|_2$ s.t. $\|x\|_1 \leq \tau$

- **Each iteration of SPGL1 involves computing:**

$$\boxed{\tau_{k+1} = \tau_k + (\sigma - \phi(\tau_k))/\phi'(\tau_k)}$$

- **Let x_{τ_k} approximately solve $\mathbf{LS}_{\tau_k}$ and $r_{\tau_k} = Ax_{\tau_k} - b$, then**

$$\boxed{\phi(\tau_k) = \left\|r_{\tau_k}\right\|_2 \text{ and } \phi'(\tau_k) = \left\|A^\top r_{\tau_k}\right\|_\infty / \left\|r_{\tau_k}\right\|_2}$$

- **Note: $\phi(\tau_k)$ and $\phi'(\tau_k)$ are the approximate solution and dual solution to $\mathbf{LS}_{\tau_k}$, respectively.**

- **SPGL1 uses a SPG (Spectral Projected Gradient) method to solve $\mathbf{LS}_{\tau_k}$.**

- **ParNes uses the same framework except the SPG method is replaced with a proximal gradient method.**

Nesterov's Proximal Gradient Algorithm for Smooth Minimization

- **Solves:**

$$\boxed{\min f(x) \textbf{ s.t. } x \in Q}$$

where $Q \subseteq \mathbb{R}^n$ is closed and convex and $f : Q \to \mathbb{R}$ is smooth, convex, and Lipschitz differentiable with Lipschitz constant L.

- **Computes the sequences:**

$$y_k = \textbf{argmin}_{y \in Q} \nabla f(x_k)^\top (y - x_k) + \frac{L}{2} \|y - x_k\|_2^2,$$
$$z_k = \textbf{argmin}_{z \in Q} \sum_{i=0}^{k} \frac{i+1}{2} \nabla f(x_i)^\top (z - x_i) + \frac{L}{2} \|z - c\|_2^2,$$
$$x_k = \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k. \quad \textbf{(}f(x_k)\textbf{ converges to the solution)}$$

- **c is a constant called the prox-center.**

Nesterov's Algorithm for Non-Smooth Minimization

- **Solves:**

$$\boxed{\min f(x) \textbf{ s.t. } x \in Q}$$

where $Q \subseteq \mathbb{R}^n$ is closed and convex and $f : Q \to \mathbb{R}$ is convex but not necessarily differentiable.

- **Assume there is a convex set $Q_d \subset \mathbb{R}^p$ and $W \in \mathbb{R}^{p \times n}$ where f can be written as $f(x) = \max_{u \in Q_d} \langle u, Wx \rangle$.**

- **Replace $f(x)$ with the smooth approximation**

$$\boxed{f_\mu(x) = \max_{u \in Q_d} \langle u, Wx \rangle - \frac{\mu}{2} \|u\|_2^2}$$

- **Apply Nesterov's algorithm for smooth minimization to $f_\mu(x)$.**

Convergence of Nesterov's Algorithms

- **Convergence of Smooth Version:**
  - **Let x^* be the optimal solution to:**

$$\boxed{\min f(x) \textbf{ s.t. } x \in Q}$$

  - **The iterates y_k satisfy:**

$$\boxed{f(y_k) - f(x^*) \leq \frac{2L}{(k+1)(k+2)} \|x^* - c\|_2^2 = O\left(\frac{L}{k^2}\right)}$$

implying $O\left(\sqrt{\frac{L}{\epsilon}}\right)$ iterations bring $f(y_k)$ within ϵ of the optimal value.

- **Convergence of Non-Smooth Version:**
  - **∇f_μ has Lipschitz constant $L_\mu = 1/\mu$.**
  - **Assuming μ is chosen to be proportional to ϵ, $O\left(\frac{1}{\epsilon}\right)$ iterations bring $f(y_k)$ within ϵ of the optimal value.**

Nesterov-LASSO

- **NESTA uses Nesterov's algorithm for non-smooth minimization to solve BP$_\sigma$.**

$$\boxed{\min \|x\|_1 \;\; \textbf{s.t.} \;\; \|Ax-b\|_2 \le \sigma}$$

- **ParNes uses the smooth version to solve LS$_{\tau_k}$ in each iteration**

$$\boxed{\min \|Ax-b\|_2 \;\; \textbf{s.t.} \;\; \|x\|_1 \le \tau}$$

- **The sequences in Nesterov's smooth algorithm simplify to:**

$$\boxed{\begin{array}{l} y_k = \mathbf{proj}_1(x_k - \nabla f(x_k)/L, \tau), \\ z_k = \mathbf{proj}_1\left(c - \frac{1}{L}\sum_{i=0}^{k} \frac{i+1}{2}\nabla f(x_i), \tau\right), \\ x_k = \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k. \; (f(x_k) \textbf{ converges to the solution}) \end{array}}$$

where $\mathbf{proj}_1(s,\tau) := \textbf{argmin}\, \|s-x\|_2$ **s.t.** $\|x\|_1 \le \tau$.

One-norm Projector

- **Each iteration of Nesterov-LASSO involves two solves of**

$$\boxed{\mathbf{proj}_1(s,\tau) := \mathbf{argmin}\,\|s-x\|_2 \;\; \mathbf{s.t.} \;\; \|x\|_1 \le \tau}$$

- **Assume the coefficients of s are positive and ordered from largest to smallest.**

- **The solution x^* is given by**

$$\boxed{x_i^* = \max\{0, s_i - \eta\} \quad \mathbf{with} \quad \eta = \frac{\tau-(s_1+\ldots+s_k)}{k}}$$

where k is the largest index such that $\eta \le s_k$. (Duchi, Shalev-Schwartz, Berg, etc.)

- **Each solve costs $O(n \log n)$ in the worst case and much less in practice.**

Convergence of Nesterov-LASSO

- **Recall minimizing f with Nesterov's method gives ($x^* = \mathrm{argmin}_{x\in Q} f(x)$)**

$$\boxed{f(y_k) - f(x^*) \le \frac{2L}{(k+1)(k+2)} \left\|x^* - c\right\|_2^2 = O\left(\frac{L}{k^2}\right)}$$

- **Assume x^* is unique. Since $x_k \to y_k$, updating c with x_k should speed up the convergence.**

- **In ParNes, Nesterov-LASSO is restarted every q iterations with $c = x_{k_{current}}$.**

- **q can be chosen in an optimal manner if**

  1. the solution x^* is s-sparse,

  2. the iterates x_k are s-sparse,

  3. A satisfies the *restricted isometry property* of order $2s$: $\exists \delta_{2s} \in (0,1)$ s.t

$$\boxed{(1-\delta_{2s})\left\|x\right\|_2^2 \le \left\|Ax\right\|_2^2 \le (1+\delta_{2s})\left\|x\right\|_2^2}$$

  whenever x is $2s$-sparse.

Convergence Results of Nesterov-LASSO

- **Let $x_{p,q}$ represent the q-th iterate after the p-th prox-center change.**

- **With the assumptions on the previous slide, we have the following results:**

  – **Let e be the base of the natural logarithm and**

$$\boxed{q_{\text{opt}} = e\sqrt{\frac{L}{\delta_{2s}}} \textbf{ and } p_{\text{tot}} = -\log \varepsilon}$$

  Then the total number of iterations, $p_{\text{tot}} \times q_{\text{opt}}$, to get $\|x_{p,q} - x^*\|_2 \leq \varepsilon$ is minimized with these choices of q_{opt} and p_{tot}.

  – **For each p,**

$$\boxed{\left\|x_{p,q_{\text{opt}}} - x^*\right\|_2 \leq 1/e \left\|x_{p,1} - x^*\right\|_2}$$

  – **Nesterov-LASSO is linearly convergent under the previous assumptions!**

ParNes: Experiment Details

- **To gain a good comparison, we repeat some of the experiments done in the NESTA paper (Becker, Bobin, Candès) using their code.**

- **Tests some of the most competitive algorithms using hard, realistic problems.**

- **The next two experiments recover an s-sparse signal with $n = 262144$, $m = n/8$, $s = m/5$, and noise level $\sigma = 0.1$.**

  - **Tests dynamic range values (ratio of the largest and smallest non-zero coefficients of the unknown signal) of $d = 20, 40, 60, 80, 100$ dB.**

  - **A is a randomly subsampled discrete cosine transform.**

  - **Let x_{NES} be NESTA's solution when the relative variation of the objective function is less than 10^{-7}. The stopping rule is**

$$\|x_k\|_1 \leq \|x_{\text{NES}}\|_1 \quad \textbf{and} \quad \|b - Ax_k\|_2 \leq 1.05\,\|b - Ax_{\text{NES}}\|_2.$$

Numerical Experiments: Speed

- **Table gives the number of function calls.**
- **DNC if calls to A or $A^\top$ exceeds 20,000.**

Method	20 dB	40 dB	60 dB	80 dB	100 dB
PARNES	122	172	214	470	632
NESTA	383	809	1639	4341	15227
NESTA + CT	483	513	583	685	787
GPSR	64	622	5030	DNC	DNC
GPSR + CT	271	219	357	1219	11737
SPARSA	323	387	465	541	693
SPGL1	58	102	191	374	504
FISTA	69	267	1020	3465	12462
FPC-AS	209	231	299	371	287
FPC-AS (CG)	253	289	375	481	361
FPC	474	386	478	1068	9614
FPC-BB	164	168	206	278	1082
BREGMAN-BB	211	223	309	455	1408

Numerical Experiments: Accuracy

- **DNC if calls to A or $A^\top$ (N_A) exceeds 20,000.**
- **Dynamic range is $d = 100$ dB.**

Methods	N_A	$\|x\|_1$	$\frac{\|x-x^*\|_1}{\|x^*\|_1}$
PARNES	632	942197.606	0.000693
NESTA	15227	942402.960	0.004124
NESTA + CT	787	942211.581	0.000812
GPSR	DNC	DNC	DNC
GPSR + CT	11737	942211.377	0.001420
SPARSA	693	942197.785	0.000783
SPGL1	504	942211.520	0.001326
FISTA	12462	942211.540	0.000363
FPC-AS	287	942210.925	0.000672
FPC-AS (CG)	361	942210.512	0.000671
FPC	9614	942211.540	0.001422
FPC-BB	1082	942209.854	0.001378
BREGMAN-BB	1408	942286.656	0.000891

Numerical Experiments: Speed

An approximately sparse signal (obtained from permuting the Haar wavelet coefficients of a 512×512 image) is recovered with the same stopping rule as before.

- **The measurement vector b consists of $m = n/8 = 512^2/8 = 32,768$ random discrete cosine measurements, and the noise level is set to 0.1.**

Method	Run 1	Run 2	Run 3	Run 4	Run 5
PARNES	838	810	1038	1098	654
NESTA	8817	10867	9887	9093	11211
NESTA + CT	3807	3045	3047	3225	2735
GPSR	DNC	DNC	DNC	DNC	DNC
GPSR + CT	DNC	DNC	DNC	DNC	DNC
SPARSA	2143	2353	1977	1613	DNC
SPGL1	916	892	1115	1437	938
FISTA	3375	2940	2748	2538	3855
FPC-AS	DNC	DNC	DNC	DNC	DNC
FPC-AS (CG)	DNC	DNC	DNC	DNC	DNC
FPC	DNC	DNC	DNC	DNC	DNC
FPC-BB	5614	7906	5986	4652	6906
BREGMAN-BB	3288	1281	1507	2892	3104

Software Download

- Resources used in paper and talk
  - NESTA - http://www.acm.caltech.edu/ nesta/
  - SPGL1 - http://www.cs.ubc.ca/labs/scl/index.php/Main/Spgl1
  - BREGMAN - http://www.caam.rice.edu/ optimization/L1/2006/10/bregman-iterative-algorithms-for.html
  - SparseLab - http://sparselab.stanford.edu/
  - FPC-AS - http://www.caam.rice.edu/ optimization/L1/FPC_AS/
  - FPC - http://www.caam.rice.edu/ optimization/L1/fpc/
  - SPARCO - http://www.cs.ubc.ca/labs/scl/sparco/
  - GSPR - http://www.lx.it.pt/ mtf/GPSR/
  - SpaRSA - http://www.lx.it.pt/ mtf/SpaRSA/
- Many other resources available at - http://www-dsp.rice.edu/cs