Gradient Descent with Sparsification: An Iterative Algorithm for Sparse Recovery with Restricted Isometry Property

Rahul Garg and Rohit Khandekar
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA

Outline

- Sparse regression
- L1 minimization and compressed sensing
- GraDeS: A near-linear time algorithm for optimal sparse recovery
- Proof outline
- Experimental results
- Conclusions

The Sparse Regression Problem

Given a vector $y \in R^m$, a matrix $\phi \in R^{m \times n}$, $e \geq 0$

Find a vector $x \in R^n$ such that,

$$||y - \phi x||^2 \leq e$$

and x has smallest number of non-zero entries.

Noiseless version: $e = 0$

Noisy version: $e > 0$

Applications of Sparse Regression

- Model selection in machine learning
  - Given a set of observations ϕ (predictors)
  - A response variable y
  - Find the smallest set of observations x (features) that explain the response variable
- Image de-convolution and de-noising
  - $y = x * h + \eta$
  - y: measured signal
  - h known transfer function
  - η unknown noise
  - Objective is to find the “best” x
- Sparse PCA
- Compressed sensing and sub-Nyquist sampling
  - A sparse (or compressible) signal x
  - Measured using $y = \phi x + e$
  - Given y, ϕ, the goal is to reconstruct x

Solving the Sparse Regression Problems

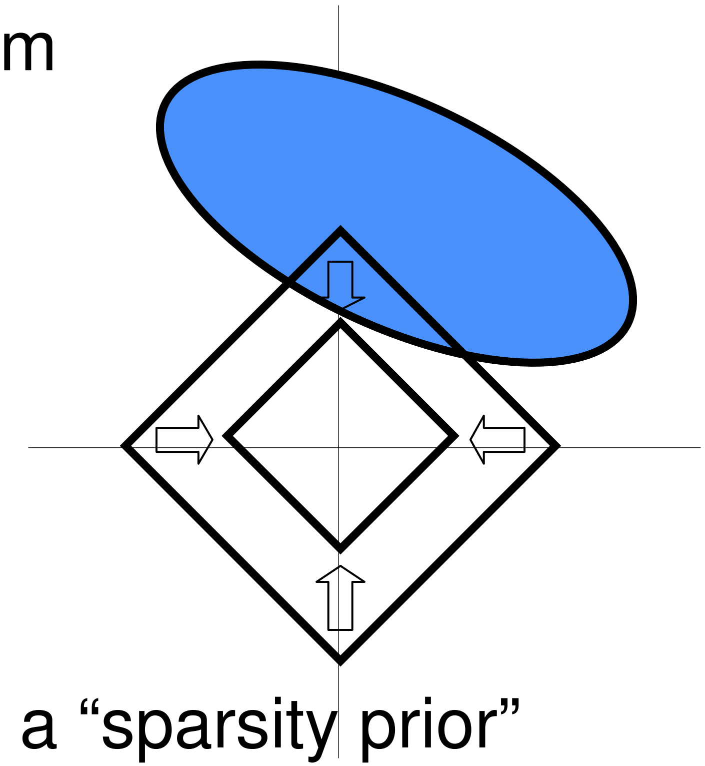
- Combinatorial algorithms
  - Matching pursuit
  - Forward selection
  - Stagewise
  - Orthogonal matching pursuit
- However
  - The problem is NP hard…
  - … and hard to approximate

The L1 Magic

- The “L0” minimization problem “relaxed” to the following L1 minimization problem

$$\text{Minimize} \quad \| x \|_1$$

$$\text{Subject to} \quad \| y - \phi x \|^2 \leq e$$



- Good heuristic
- Finds sparse solution
- “L1-penalty” commonly used as a “sparsity prior”

The Era of Compressed Sensing

- Define *isometry constants* of a matrix as smallest numbers δ_s such that for all s-sparse vectors x:

$$(1-\delta_s)\,||\,x\,||_2^2 \;\le\; ||\,\phi x\,||_2^2 \;\le\; (1+\delta_s)\,||\,x\,||_2^2$$

- If $\delta_{2s} < \sqrt{2} - 1$ (**restricted isometry property**) then the L1 relaxation is guaranteed to
  - Find the exact “L0”-optimal solution when e = 0
  - Find close to the optimal solution otherwise
- Matrices satisfying restricted isometry property (RIP)
  - Random matrices with m > O(s log n)
  - Random Fourier matrices of appropriate dimensions
  - Deterministic constructions
- Universal encoding strategies

Problems with L1 Minimization

- Slow
  - Needs linear programming solvers
  - Efficient algorithms being designed
    - LARS, …
  - Still slow
- MRI Imaging
  - $N = 256^3 = 16M$; $m = 256^3 = 16M$; $s = 100K\text{-}1M$
- Combinatorial algorithms may still be preferred
- Provably optimal combinatorial algorithms

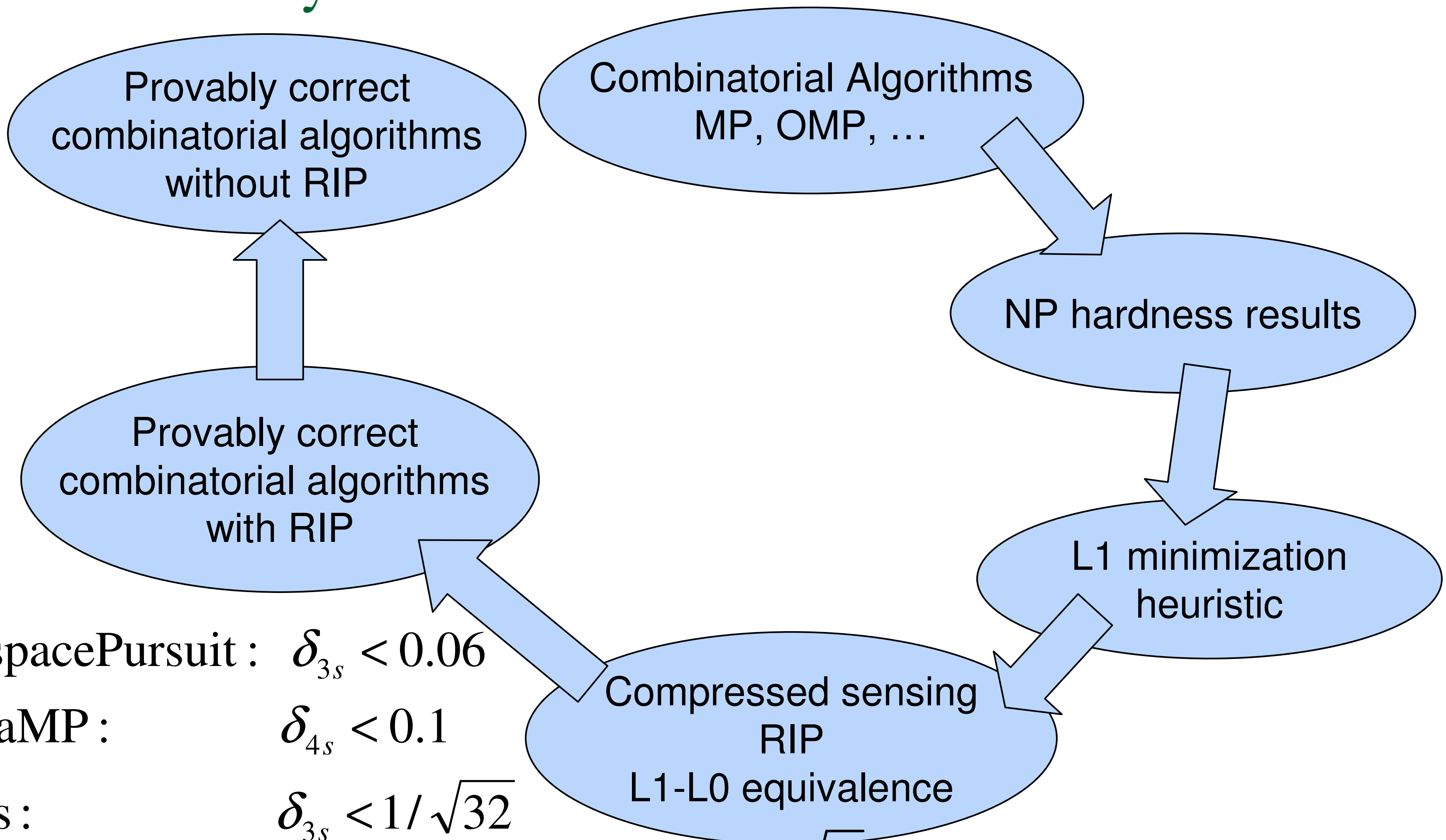
Algorithm	Cost/iteration	Max # iterations	Recovery conditions	#Ops
Basis pursuit (L1)	$O(m+n)^3$	NA	$\delta_{2s} < \sqrt{2} - 1$	10^{21}
LARS (L1)	$K + O(s^2)$	Unbounded	$\delta_{2s} < \sqrt{2} - 1$	10^{14}
Homotopy	K	s	$\mu < 1/(2s-1)$	10^{13}
OMP	$K + O(msL)$	s	$\mu < 1/2s$	10^{19}
StOMP	$K + O(msL)$	O(L)	None	10^{13}
ROMP	$K + O(msL)$	s	$\delta_{2s} < 0.03/\sqrt{(\log(s))}$	10^{19}
SP	$K + O(msL)$	O(L)	$\delta_{3s} < 0.06$	10^{13}
CoSaMP	$K + O(msL)$	L	$\delta_{4s} < 0.1$	10^{13}
IHTs	K	L	$\delta_{3s} < 1/\sqrt{32}$	10^{8}
GraDeS	K	$2L/\log[(1-\delta)/2\delta]$	$\delta_{2s} < 1/3$	10^{8}

n = number of columns
m = number of rows
s = sparsity
L = bit precision
K = time to carry out matrix-vector product

GraDeS: Gradient Descent with Sparsification

- Combinatorial algorithm for sparse regressions problems
- Iterative, needs only two matrix-vector products in every iteration
- Number of iterations needed is a constant independent of sparsity
- Moves along gradient and sparsifies
- Very simple to implement (5 lines of Matlab code)
- Needs matrices satisfying RIP
  - e.g. Compressed sensing

Summary



$\delta_{2s} < \sqrt{2} - 1$

SubspacePursuit : $\delta_{3s} < 0.06$

CoSaMP : $\delta_{4s} < 0.1$

IHTs : $\delta_{3s} < 1/\sqrt{32}$

GraDeS : $\delta_{2s} < 1/3$

Algorithm GraDeS(γ)

- Starts with x = 0
- Iteratively moves along the gradient
- Maintains a s-sparse solution

While $|| y - \phi x ||$ is large

  $r = y - \phi x$

  $g = -2 \phi^T r$

  x = Largest s components of $x - g/(2\gamma)$

end

Main Results: Noiseless Recovery

Let

- $y = \phi x^*$ for a s-sparse vector x^*
- $\delta_{2s} < 1/3$
- L = bit precision

After k iterations of algorithm GraDeS with $\gamma = 1 + \delta_{2s}$ the (sparse) solution x satisfies:

$$\| y - \phi x \|^2 < \left(\frac{2\delta_{2s}}{1-\delta_{2s}} \right)^k \| y \|^2$$

The correct solution is found in $O\left(L / \log\left(\frac{2\delta_{2s}}{1-\delta_{2s}} \right) \right)$ iterations

Main Results: Recovery with Noise

Let

- $y = \phi x^* + e$ for a s-sparse vector x^*
- $\delta_{2s} < 1/3$
- D: a small constant depending only on δ_{2s}

The algorithm GraDeS with $\gamma = 1 + \delta_{2s}$ finds a sparse solution x in

$$O\left(1/\log\left((1-\delta_{2s})/4\delta_{2s}\right)\right) \text{iterations}$$

$$\text{such that} \; \| x^* - x \| \leq D \| e \|$$

Comparable to the Candes and Tao results on L1 optimization

Proof Outline

Let

- $y = \phi x^*$ for an s-sparse vector x^*
- Define: $\Psi(x) = || y - \phi x ||^2$

  $= ||\phi(x^*-x)||^2 \approx (1 \pm \delta) \, ||(x^*-x)||^2$

Now

- $\Psi(x^*) = 0$
- $\nabla\Psi(x) \equiv g = -2\phi^T(y - \phi x) = -2\phi^T\phi(x^*-x)$

$$\Psi(x + \Delta x) - \Psi(x) = -g.\Delta x + \Delta x \phi^T \phi \Delta x$$

Proof Outline

Find Δx that minimizes:

Easy
Separable

Difficult
Non separable

$$\Psi(x + \Delta x) - \Psi(x) = -g.\Delta x + \Delta x^T \phi^T \phi \Delta x$$

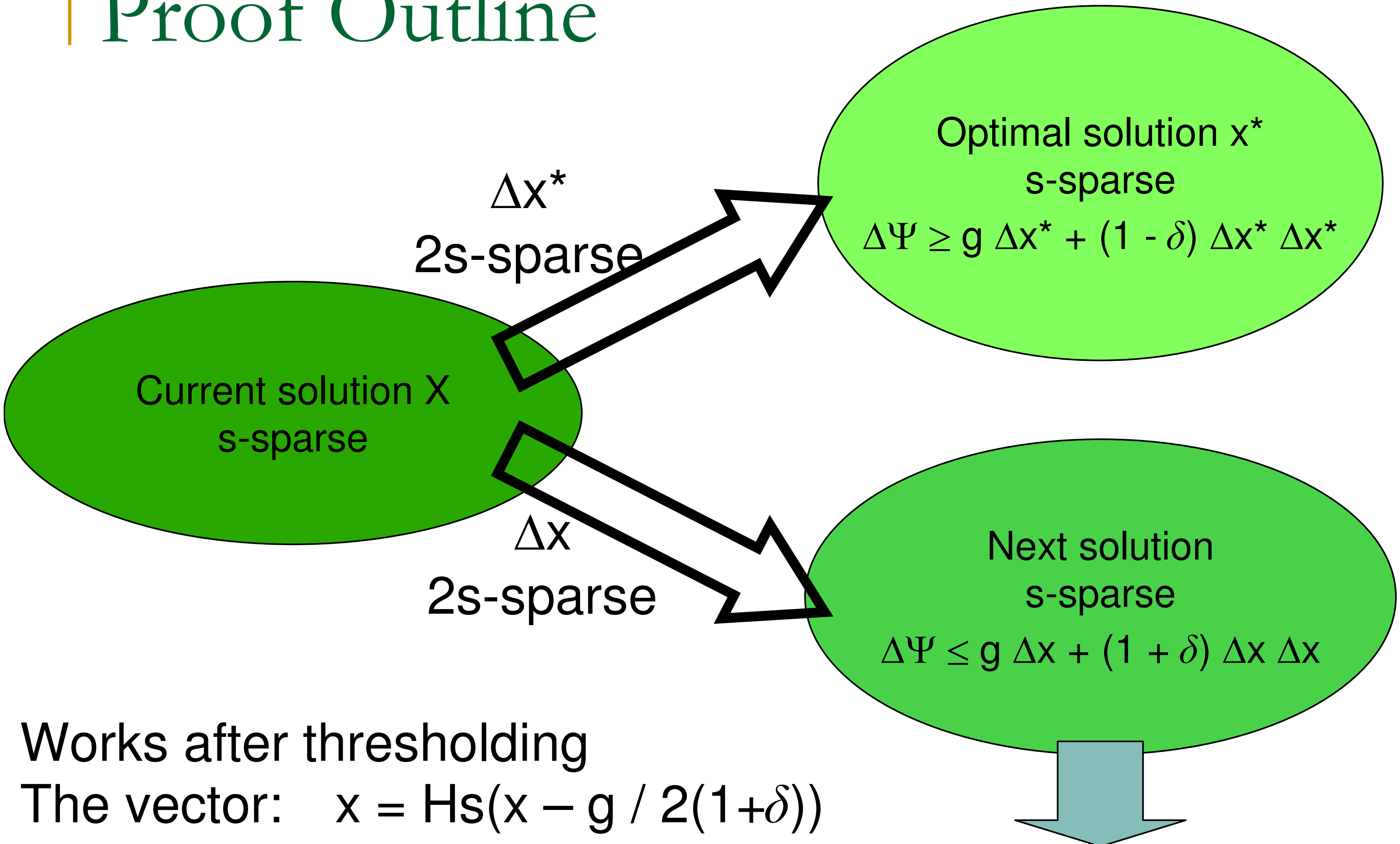
RIP

If Δx is s-sparse $\approx -g.\Delta x + (1 \pm \delta)\ \Delta x.\Delta x$

Separable

Separable

Proof Outline



Works after thresholding

The vector: $x = Hs(x - g / 2(1+\delta))$

Minimizes: $g.v + (1+\delta)\ v.v$

Among all s-sparse vectors v

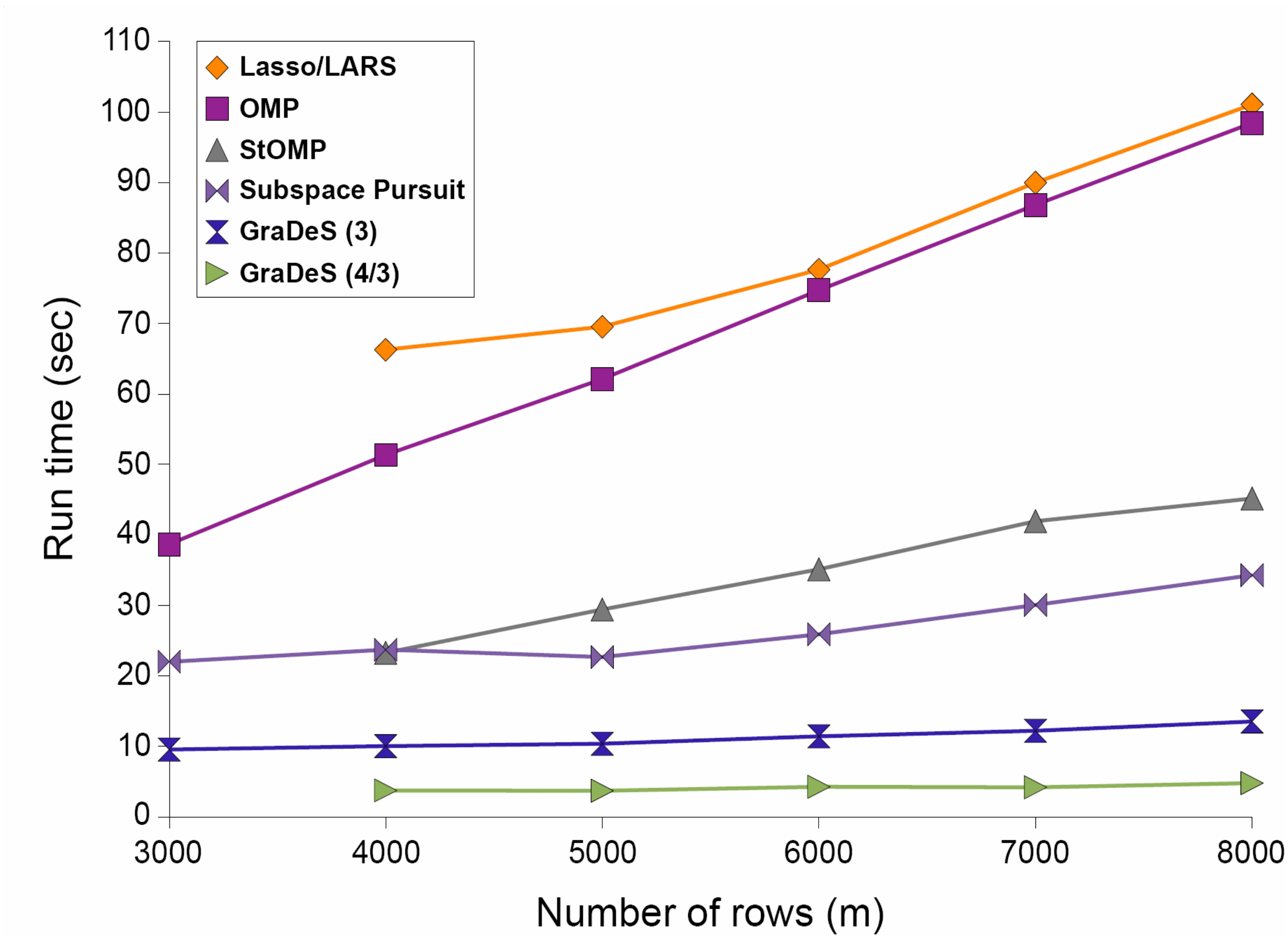
Choose $\Delta x_i = -g_i / 2(1+\delta)$

Evaluation of GraDeS in Practice

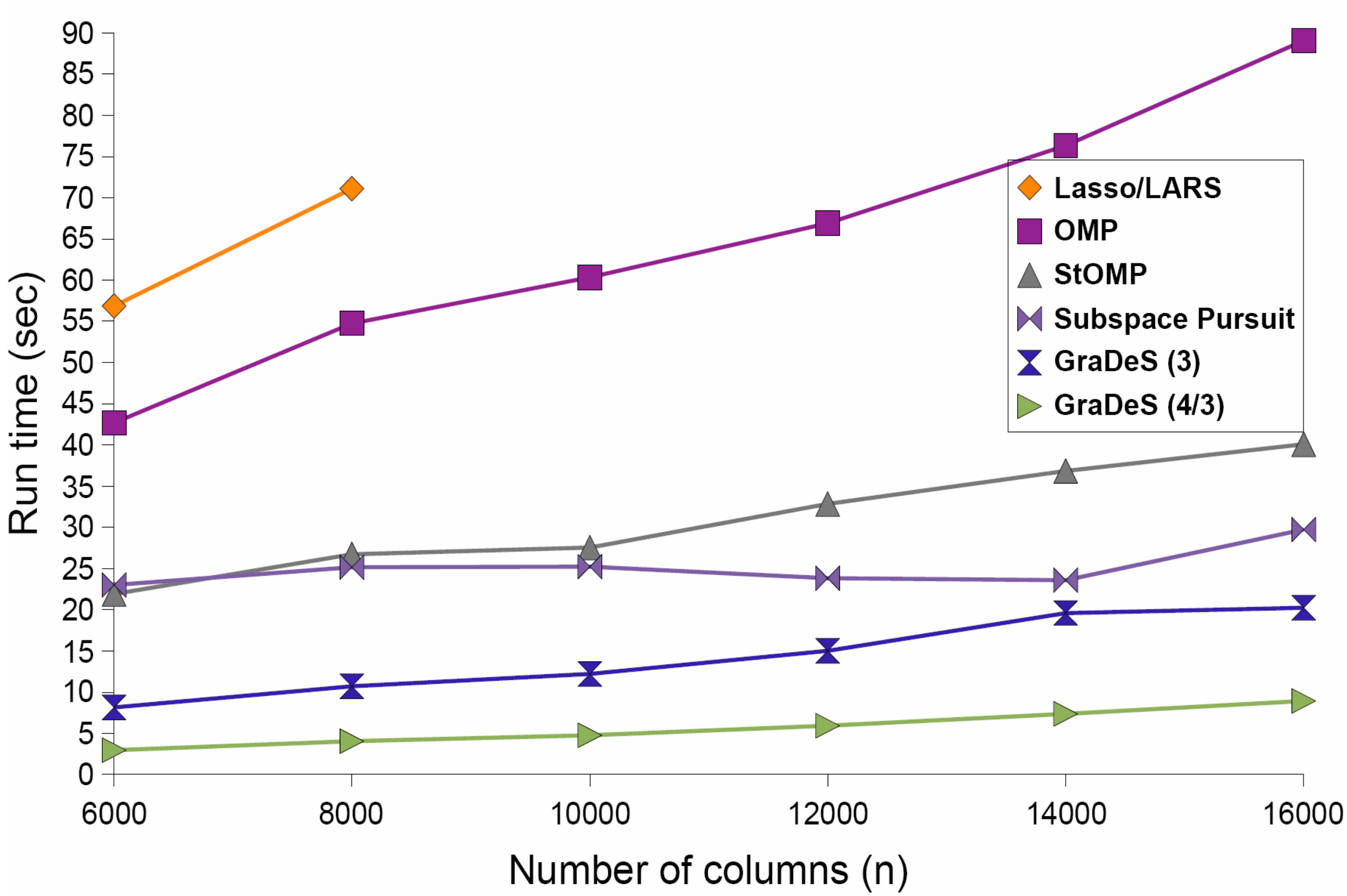
Experimental Methodology

- GraDeS was implemented using Matlab
  - Evaluated for $\gamma = 3$ and $\gamma = 4/3$
- SparseLab implementation for
  - OMP, StOMP, Lasso
- Optimized implementation for Subspace Pursuit
- Random and ϕ random sparse x
  - Normally distributed entries
  - Columns of ϕ normalized to unit variance and zero mean
- Computed $y = \phi x$
- Solved for x using (ϕ, y)

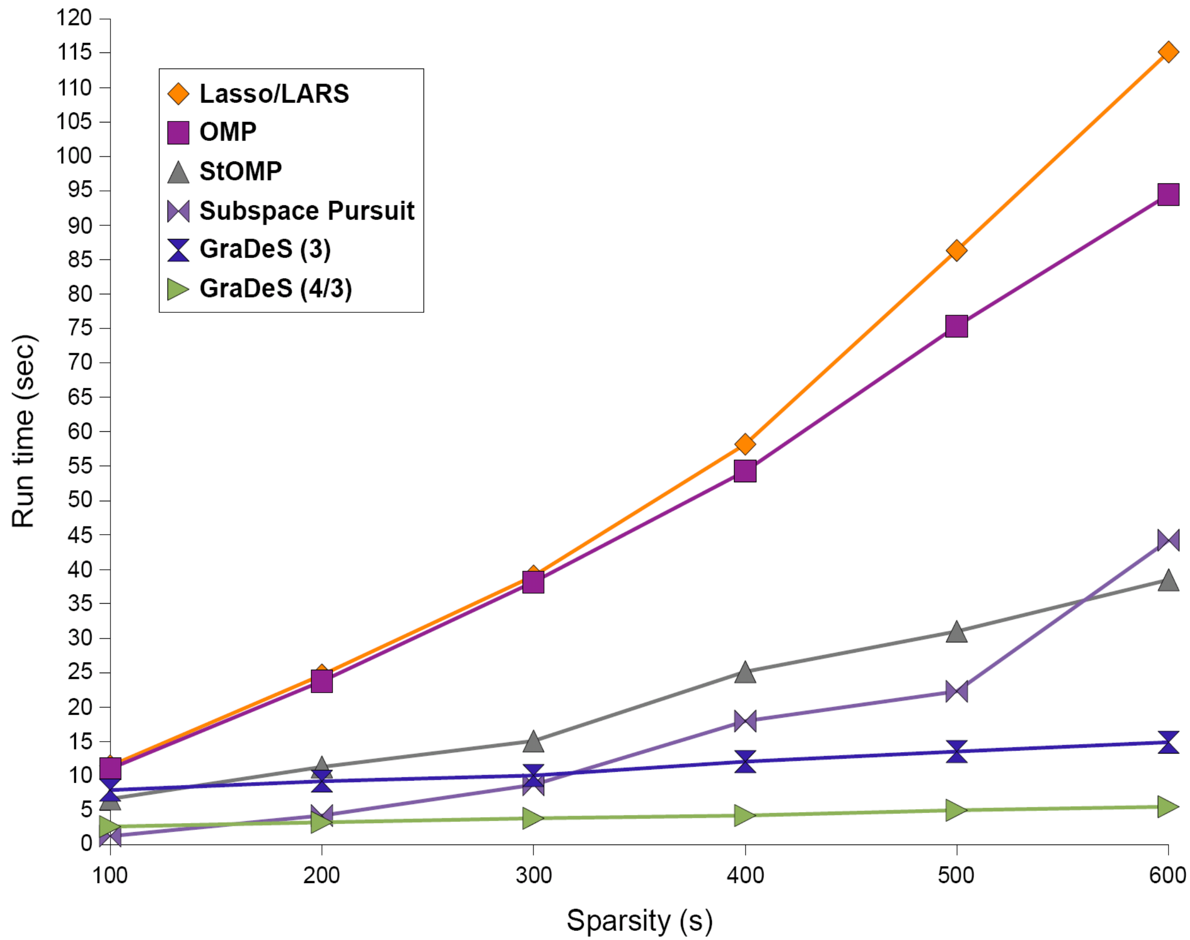
Runtimes with Different Number of Rows



Runtimes with Different Number of Columns



Runtimes with Different Sparsity Level



Recovery Properties

m	3000	3000	3000	3000	6000	3000	4000	8000
n	10000	10000	8000	10000	8000	10000	10000	8000
s	2100	1050	500	600	500	300	500	500
Lasso/LARS								Y
GraDeS(4/3)					Y	Y	Y	Y
StOMP				Y		Y	Y	Y
GraDeS(3)			Y	Y	Y	Y	Y	Y
OMP		Y	Y	Y	Y	Y	Y	Y
Subspace Pursuit		Y	Y	Y	Y	Y	Y	Y

Conclusions

- Sparse regression: A very important problem
- Compressed sensing
  - Strong theoretical foundations
  - Practical applications in medical imaging and other areas
  - Sparse recovery with RIP usually based on L1 optimization
- GraDeS: A fast algorithm for sparse recovery
  - Works with RIP assumption
  - Circumvents the L1 reduction and therefore does not need linear programming
  - Iterative and uses only two matrix-vector products / iteration
  - Needs only a constant number of iteration
  - Independent of sparsity s
  - In practice, order of magnitude faster than other algorithms
  - Better recovery properties than L1 regression (**caveat**)