

# **Solving basis pursuit: infeasible-point subgradient algorithm, computational comparison, and improvements**

Andreas Tillmann  
(joint work with D. Lorenz and M. Pfetsch)

Technische Universität Braunschweig

SIAM Conference on Applied Linear Algebra 2012



# Outline

- 1 Motivation
- 2 Infeasible-Point Subgradient Algorithm
  - ISAL1
- 3 Comparison of  $\ell_1$ -Solvers
  - Testset Construction and Computational Results
  - Improvements with Heuristic Optimality Check
- 4 Possible Future Research



# Outline

- 1 Motivation
- 2 Infeasible-Point Subgradient Algorithm
  - ISAL1
- 3 Comparison of  $\ell_1$ -Solvers
  - Testset Construction and Computational Results
  - Improvements with Heuristic Optimality Check
- 4 Possible Future Research



# Sparse Recovery via $\ell_1$ -Minimization

- Seek sparsest solution to underdetermined linear system:

$$\min \|\mathbf{x}\|_0 \quad \text{s. t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \quad (\mathbf{A} \in \mathbb{R}^{m \times n}, m < n)$$

- Finding minimum-support solution is  $\mathcal{NP}$ -hard.

Convex “relaxation”:  $\ell_1$ -minimization / Basis Pursuit:

$$\min \|\mathbf{x}\|_1 \quad \text{s. t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \quad (\text{L1})$$

- Several conditions (RIP, Nullspace Property, etc) ensure “ $\ell_0$ - $\ell_1$ -equivalence”



# Solving the Basis Pursuit Problem

- (L1) can be recast as a linear program
- Broad variety of specialized algorithms for (L1)



- A classic algorithm from nonsmooth optimization:  
**(projected) subgradient method – competitive?**
- Which algorithm is “the best”?

# Outline

- 1 Motivation
- 2 Infeasible-Point Subgradient Algorithm
  - ISAL1
- 3 Comparison of  $\ell_1$ -Solvers
  - Testset Construction and Computational Results
  - Improvements with Heuristic Optimality Check
- 4 Possible Future Research

# Projected Subgradient Methods

Problem:  $\min f(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \mathcal{F}$  ( $f, \mathcal{F}$  convex)

standard projected subgradient iteration

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathcal{F}} \left( \mathbf{x}^k - \alpha_k \mathbf{h}^k \right), \quad \alpha_k > 0, \quad \mathbf{h}^k \in \partial f(\mathbf{x}^k)$$

applicability: only reasonable if projection is “easy”

# Projected Subgradient Methods

Problem:  $\min f(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \mathcal{F}$  ( $f, \mathcal{F}$  convex)

standard projected subgradient iteration

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathcal{F}} \left( \mathbf{x}^k - \alpha_k \mathbf{h}^k \right), \quad \alpha_k > 0, \quad \mathbf{h}^k \in \partial f(\mathbf{x}^k)$$

applicability: only reasonable if projection is “easy”

~> idea: replace exact projection by approximation

“infeasible” subgradient iteration

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathcal{F}}^{\varepsilon_k} \left( \mathbf{x}^k - \alpha_k \mathbf{h}^k \right), \quad \|\mathcal{P}_{\mathcal{F}}^{\varepsilon_k} - \mathcal{P}_{\mathcal{F}}\|_2 \leq \varepsilon_k$$

# ISA = Infeasible-Point Subgradient Algorithm

- ... works for arbitrary convex objectives and constraint sets
- ... incorporates adaptive approximate projections  $\mathcal{P}_{\mathcal{F}}^{\varepsilon}$  such that  $\|\mathcal{P}_{\mathcal{F}}^{\varepsilon}(\mathbf{y}) - \mathcal{P}_{\mathcal{F}}(\mathbf{y})\|_2 \leq \varepsilon$  for every  $\varepsilon \geq 0$
- ... converges to optimality (under reasonable assumptions) whenever projection accuracies ( $\varepsilon_k$ ) sufficiently small,
  - for stepsizes  $\alpha_k > 0$  with  $\sum_{k=0}^{\infty} \alpha_k = \infty$ ,  $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$
  - for dynamic stepsizes  $\alpha_k = \lambda_k(f(\mathbf{x}^k) - \varphi)/\|\mathbf{h}^k\|_2^2$  with  $\varphi \leq \varphi^*$
- ... converges to  $\varphi$  with dynamic stepsizes using  $\varphi \geq \varphi^*$

# ISAL1 = Spezialization of ISA to $\ell_1$ -Minimization

- $f(\mathbf{x}) = \|\mathbf{x}\|_1, \quad \mathcal{F} = \{ \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b} \}, \quad \text{sign}(\mathbf{x}) \in \partial\|\mathbf{x}\|_1$
- exact projected subgradient step for (L1):

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathcal{P}_{\mathcal{F}} \left( \mathbf{x}^k - \alpha_k \mathbf{h}^k \right) \\ &= (\mathbf{x}^k - \alpha_k \mathbf{h}^k) - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \left( \mathbf{A}(\mathbf{x}^k - \alpha_k \mathbf{h}^k) - \mathbf{b} \right)\end{aligned}$$

# ISAL1 = Spezialization of ISA to $\ell_1$ -Minimization

- $f(\mathbf{x}) = \|\mathbf{x}\|_1, \quad \mathcal{F} = \{ \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b} \}, \quad \text{sign}(\mathbf{x}) \in \partial\|\mathbf{x}\|_1$
- exact projected subgradient step for (L1):

$$\mathbf{y}^k \leftarrow \mathbf{x}^k - \alpha_k \mathbf{h}^k$$

$$\mathbf{z}^k \leftarrow \text{Solution of } \mathbf{A}\mathbf{A}^\top \mathbf{z} = \mathbf{A}\mathbf{y}^k - \mathbf{b}$$

$$\mathbf{x}^{k+1} \leftarrow \mathbf{y}^k - \mathbf{A}^\top \mathbf{z}^k = \mathcal{P}_{\mathcal{F}}(\mathbf{y}^k)$$

$\mathbf{A}\mathbf{A}^\top$  is s.p.d.  $\Rightarrow$  may employ CG to solve equation system

# ISAL1 = Spezialization of ISA to $\ell_1$ -Minimization

- $f(\mathbf{x}) = \|\mathbf{x}\|_1, \quad \mathcal{F} = \{ \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b} \}, \quad \text{sign}(\mathbf{x}) \in \partial\|\mathbf{x}\|_1$
- **inexact** projected subgradient step for (L1):

$$\mathbf{y}^k \leftarrow \mathbf{x}^k - \alpha_k \mathbf{h}^k$$

$$\mathbf{z}^k \leftarrow \text{Solution of } \mathbf{A}\mathbf{A}^\top \mathbf{z} \approx \mathbf{A}\mathbf{y}^k - \mathbf{b}$$

$$\mathbf{x}^{k+1} \leftarrow \mathbf{y}^k - \mathbf{A}^\top \mathbf{z}^k = \mathcal{P}_{\mathcal{F}}^{\varepsilon_k}(\mathbf{y}^k)$$

$\mathbf{A}\mathbf{A}^\top$  is s.p.d.  $\Rightarrow$  may employ CG to solve equation system

- **Approximation:** Stop after a few CG iterations  
(CG residual norm  $\leq \sigma_{\min}(\mathbf{A}) \cdot \varepsilon_k \Rightarrow \mathcal{P}_{\mathcal{F}}^{\varepsilon_k}$  fits ISA framework)

# Outline

- 1 Motivation
- 2 Infeasible-Point Subgradient Algorithm
  - ISAL1
- 3 Comparison of  $\ell_1$ -Solvers
  - Testset Construction and Computational Results
  - Improvements with Heuristic Optimality Check
- 4 Possible Future Research

# Our Testset

- 100 matrices  $\mathbf{A}$  (74 dense, 26 sparse)
  - dense: 512  $\times$   $\{1024, 1536, 2048, 4096\}$
  - 1024  $\times$   $\{2048, 3072, 4096, 8192\}$
  - sparse: 2048  $\times$   $\{4096, 6144, 8192, 12288\}$
  - 8192  $\times$   $\{16384, 24576, 32768, 49152\}$
- random (e.g., partial Hadamard, random signs, ...)
- concatenations of dictionaries (e.g., [Haar, ID, RST], ...)
- columns normalized
- 2 or 3 vectors  $\mathbf{x}^*$  per matrix such that each resulting (L1) instance (with  $\mathbf{b} := \mathbf{A}\mathbf{x}^*$ ) has unique optimum  $\mathbf{x}^*$

# Constructing Unique Solutions

274 instances with known, unique solution vectors  $\mathbf{x}^*$ :

- For each matrix  $\mathbf{A}$ , choose support  $S$  which obeys

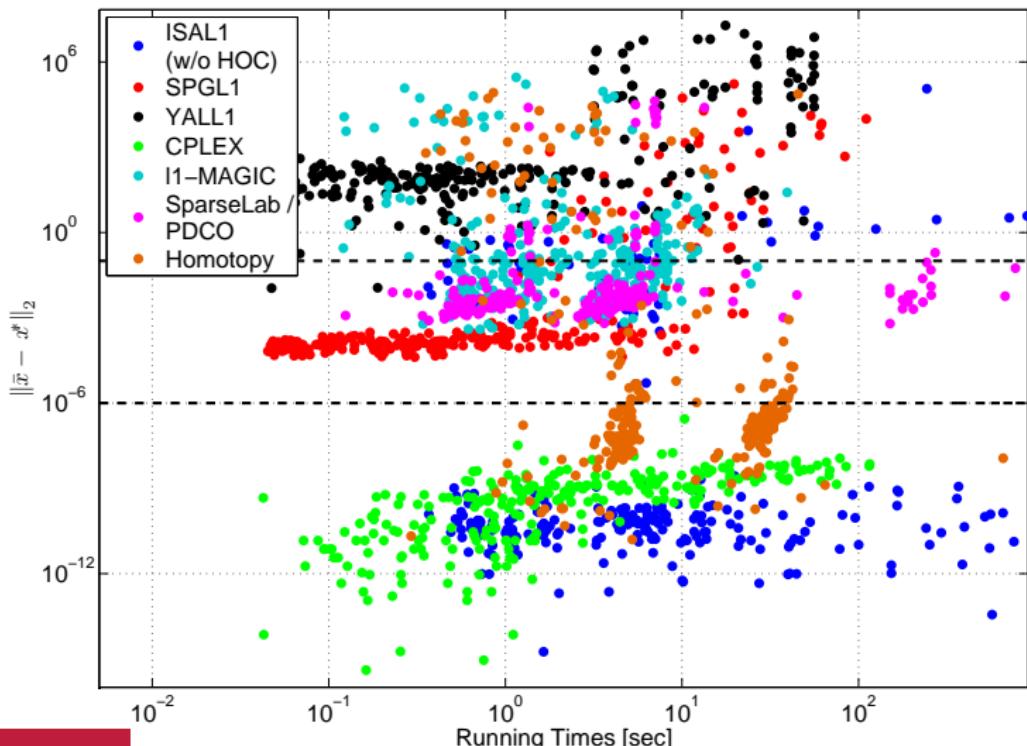
$$ERC(\mathbf{A}, S) := \max_{j \notin S} \|\mathbf{A}_S^\dagger \mathbf{a}_j\|_1 < 1.$$

- pick  $S$  at random, and
- try increasing some  $S$  by repeatedly adding the resp. arg max
- For dense  $\mathbf{A}$ 's, use L1TestPack to construct another unique solution support (via optimality condition for (L1))
- Entries of  $\mathbf{x}_S^*$  random with high dynamic range

# Comparison Setup

- Only **exact** solvers for (L1):  $\min \|\mathbf{x}\|_1$  s.t.  $\mathbf{A}\mathbf{x} = \mathbf{b}$
- Tested algorithms:  
ISAL1, SPGL1, YALL1,  $\ell_1$ -Magic, SolveBP (SparseLab),  
 $\ell_1$ -Homotopy, CPLEX (Dual Simplex)
- Use default settings (black box usage)
- Solution  $\bar{\mathbf{x}}$  “optimal”, if  $\|\bar{\mathbf{x}} - \mathbf{x}^*\|_2 \leq 10^{-6}$
- Solution  $\bar{\mathbf{x}}$  “acceptable”, if  $\|\bar{\mathbf{x}} - \mathbf{x}^*\|_2 \leq 10^{-1}$

# Running Time vs. Distance from Unique Optimum



# Results: Average Performances

- CPLEX (Dual Simplex): most reliable solver
- SPGL1: apparently very fast with usually acceptable solutions
- ISAL1: mostly very accurate, but rather slow
- SolveBP: often produces acceptable solutions, but slow
- $\ell_1$ -Magic: fast for sparse  $\mathbf{A}$ , but results often unacceptable
- YALL1: very fast, but results largely unacceptable
- $\ell_1$ -Homotopy: usually pretty accurate (not always acceptable), but on the slow side

Can we achieve better performances without changing default (tolerance) settings?

# New Optimality Check for $\ell_1$ -Minimization

- Optimality criterion for (L1):

$$\mathbf{x}^* \in \arg \min_{\mathbf{x}: \mathbf{A}\mathbf{x}=\mathbf{b}} \|\mathbf{x}\|_1 \quad \Leftrightarrow \quad \partial \|\mathbf{x}^*\|_1 \cap \text{Im}(\mathbf{A}^\top) \neq \emptyset$$

- Exact evalution too expensive

# New Optimality Check for $\ell_1$ -Minimization

- Optimality criterion for (L1):

$$\mathbf{x}^* \in \arg \min_{\mathbf{x}: \mathbf{A}\mathbf{x}=\mathbf{b}} \|\mathbf{x}\|_1 \quad \Leftrightarrow \quad \partial \|\mathbf{x}^*\|_1 \cap \text{Im}(\mathbf{A}^\top) \neq \emptyset$$

- Exact evalution too expensive
- **Heuristic Optimality Check (HOC):**  
Estimate support  $S$  of given  $\mathbf{x}$  and (approximately) solve

$$\mathbf{A}_S^\top \mathbf{w} = \text{sign}(\mathbf{x}_S).$$

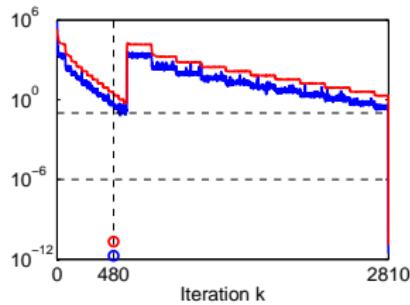
If  $\mathbf{w}$  is dual-feasible ( $\|\mathbf{A}^\top \mathbf{w}\|_\infty \leq 1$ ), compute  $\bar{\mathbf{x}}$  from  $\mathbf{A}_S \bar{\mathbf{x}}_S = \mathbf{b}$ .

If  $\bar{\mathbf{x}}$  is primal-feasible, it is optimal if  $\mathbf{b}^\top \mathbf{w} = \|\bar{\mathbf{x}}\|_1$ .

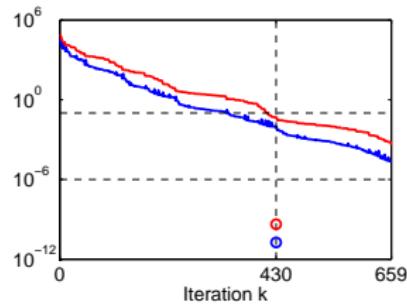
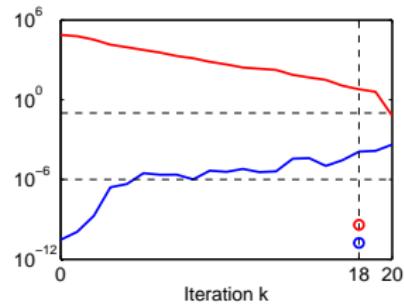
Allows safe “jumping” to optimum (from infeasible points)

# Impact of Heuristic Optimality Check (Example)

HOC in ISAL1



HOC in SPGL1

HOC in  $\ell_1$ -Magic

(red curves: distance to known optimum, blue curves: feasibility violation)

# Improvements with HOC (numbers)

(200+74 instances)	ISAL1	SPGL1	YALL1	$\ell_1$ -Mag.	$\ell_1$ -Hom.
solved faster w/ HOC	184/212	-/0	-/0	-/0	161/181
solved only w/ HOC	42	191	13	170	56
improved (ERC-based)	99.5%	88.0%	6.5%	78.0%	91.0%
improved (other part)	36.5%	20.3%	0.0%	18.9%	74.3%

# Improvements with HOC (numbers)

(200+74 instances)	ISAL1	SPGL1	YALL1	$\ell_1$ -Mag.	$\ell_1$ -Hom.
solved faster w/ HOC	184/212	-/0	-/0	-/0	161/181
solved only w/ HOC	42	191	13	170	56
improved (ERC-based)	99.5%	88.0%	6.5%	78.0%	91.0%
improved (other part)	36.5%	20.3%	0.0%	18.9%	74.3%

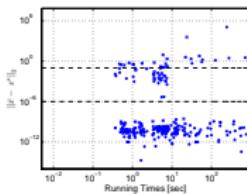
Explanation for higher HOC success rate on ERC-based testset:

$$\text{ERC} \implies \mathbf{w}^* = (\mathbf{A}_{S^*}^\top)^\dagger \text{sign}(\mathbf{x}_{S^*}^*) \text{ satisfies } \mathbf{A}^\top \mathbf{w}^* \in \partial \|\mathbf{x}^*\|_1.$$

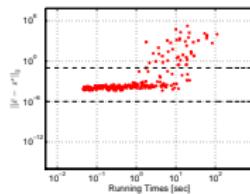
# Improvements with HOC (pictures)

without HOC:

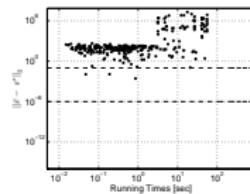
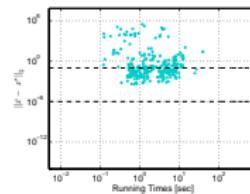
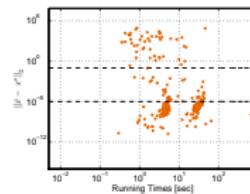
ISAL1



SPGL1

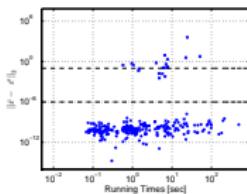


YALL1

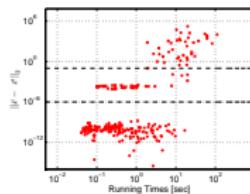
 $\ell_1$ -Magic $\ell_1$ -Hom.

with HOC:

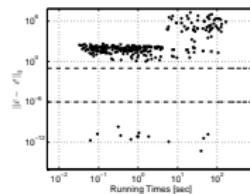
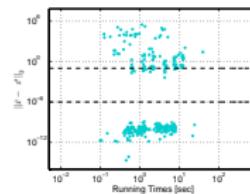
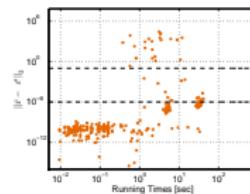
ISAL1



SPGL1



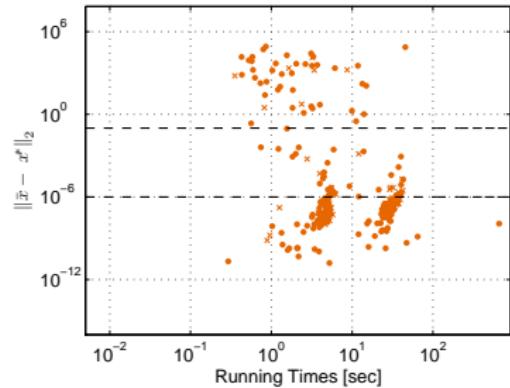
YALL1

 $\ell_1$ -Magic $\ell_1$ -Hom.

# Rehabilitation of $\ell_1$ -Homotopy

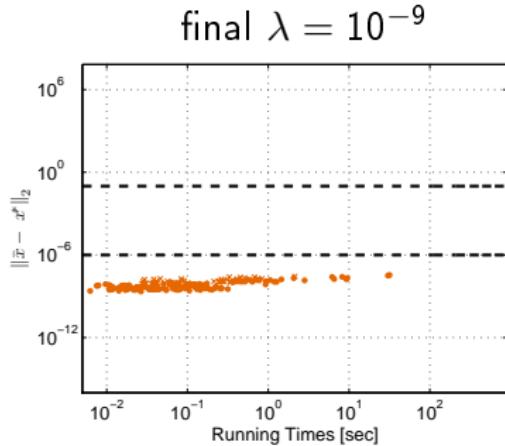
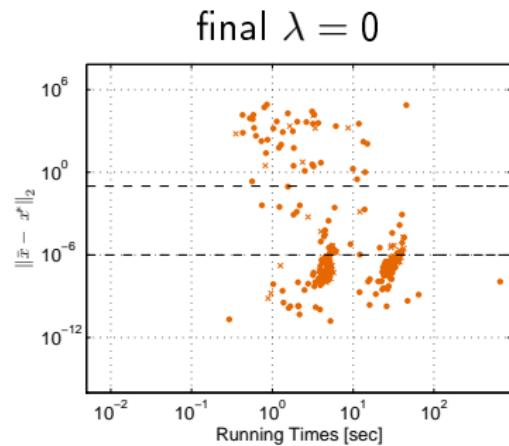
The Homotopy method **provably** solves (L1) via a sequence of problems  $\min \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 + \lambda \|\mathbf{x}\|_1$  for parameters  $\lambda \geq 0$  decreasing to **zero**. Also: Provably **fast** for suff. sparse solutions.

not fast, inaccurate ?!



# Rehabilitation of $\ell_1$ -Homotopy

The Homotopy method **provably** solves (L1) via a sequence of problems  $\min \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 + \lambda \|\mathbf{x}\|_1$  for parameters  $\lambda \geq 0$  decreasing to **zero**. Also: Provably **fast** for suff. sparse solutions.



Numerical issues?

Winner!

# Possible Future Research

- ISAL1 with variable target values?
- HOC helpful in Approximate Homotopy Path algorithm?
- Extensions to Denoising problems?
  - $\mathcal{P}_{\mathcal{F}}^{\varepsilon}$  for, e.g.,  $\mathcal{F} = \{ \mathbf{x} \mid \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \leq \delta \}$ ?
  - HOC schemes?
  - Testsets, solver comparisons (also for implicit matrices), ... ?
- “Really hard” instances?
  - e.g., construction of Mairal & Yu for which the (exact) Homotopy path has  $\mathcal{O}(3^n)$  kinks?
  - ...

# SPEAR Project

ISA theory:

*Lorenz, Pfetsch & T.: "An Infeasible-Point Subgradient Method Using Adaptive Approximate Projections", 2011*

ISAL1 theory, HOC & numerical results:

*Lorenz, Pfetsch & T.: "Solving Basis Pursuit: Subgradient Algorithm, Heuristic Optimality Check, and Solver Comparison", 2011/2012*

Papers, MATLAB Codes (ISAL1, HOC, L1TestPack), Testset, Slides, Posters etc. — available at:

[www.math.tu-bs.de/mo/spear](http://www.math.tu-bs.de/mo/spear)

