Constructing Test Instances for Sparse Recovery Algorithms

Christian Kruschel and Dirk Lorenz, May 20, 2012
SPEAR - Sparse Exact and Approximate Recovery

TU Braunschweig
Analysis and Algebra

TU Darmstadt
Discrete Optimization

Dirk Lorenz
Marc Pfetsch
Andreas Tillmann
Christian Kruschel

Project period: 2011-2014
Homepage: http://www.math.nat.tu-bs.de/mo/spear
Minimization Problems for Sparse Recovery

We consider $A \in \mathbb{R}^{m \times n}$, $m \leq n$, $b \in \mathbb{R}^m$ and

$$\min_y \|y\|_1 \text{ s.t. } Ay = b,$$

(BP)

further for $\lambda \geq 0$ its denoising variant

$$\min_y \frac{1}{2}\|Ay - b\|_2^2 + \lambda \|y\|_1,$$

(QP$_\lambda$)

with

$$\|y\|_p := \sum_{i=1}^{n} |y_i|^p, p \in \mathbb{N}\setminus\{0\}.$$
So Many Solvers to Compare...

For (BP):
- SPGL1
- NESTA
- YALL1
- ISAL1
- ...

For (QP_\lambda ):
- ISTA
- FISTA
- SparSA
- FPC
- ...

How to compare?

L1TestPack generates test instances for (QP_\lambda ).

- matrix A can be gaussian, bernoulli, partial DCT, heaviside, random orthonormal rows
- solution x∗ can be chosen gaussian, bernoulli, different dynamic ranges.
- number of non-zero entries in x∗ eligible.
- right side b will be calculated (we’ll see later...)

www.tu-braunschweig.de/iaa/personal/lorenz/l1testpack
So Many Solvers to Compare...

For (BP):
- SPGL1
- NESTA
- YALL1
- ISAL1
- ...

For (QP_\lambda):
- ISTA
- FISTA
- SparSA
- FPC
- ...

How to compare (e.g. in Matlab)?
L1TestPack generates test instances for (QP_\lambda).

www.tu-braunschweig.de/iaa/personal/lorenz/l1testpack
So Many Solvers to Compare...

For (BP):
- SPGL1
- NESTA
- YALL1
- ISAL1
- ...

For (QP\(\lambda\)):
- ISTA
- FISTA
- SparSA
- FPC
- ...

How to compare (e.g. in Matlab) ?

L1TestPack generates test instances for (QP\(\lambda\)).

- matrix \(A\) can be gaussian, bernoulli, partial DCT, heaviside, random orthonormal rows
- solution \(x^*\) can be chosen gaussian, bernoulli, different dynamic ranges.
- number of non-zero entries in \(x^*\) eligible.
- right side \(b\) will be calculated (we’ll see later...)

www.tu-braunschweig.de/iaa/personal/lorenz/l1testpack
So Many Solvers to Compare...

For (BP):
- SPGL1
- NESTA
- YALL1
- ISAL1
- ...

For (QP\(\lambda\)):
- ISTA
- FISTA
- SparSA
- FPC
- ...

How to compare (e.g. in Matlab)?

L1TestPack generates test instances for \((QP_\lambda)\).

- matrix \(A\) can be gaussian, bernoulli, partial DCT, heaviside, random orthonormal rows
- solution \(x^*\) can be chosen gaussian, bernoulli, different dynamic ranges.
  - number of non-zero entries in \(x^*\) eligible.
  - right side \(b\) will be calculated (we’ll see later...)

www.tu-braunschweig.de/iaa/personal/lorenz/l1testpack
So Many Solvers to Compare...

For (BP):
- SPGL1
- NESTA
- YALL1
- ISAL1
- ...

For (QP):  
- ISTA
- FISTA
- SparSA
- FPC
- ...

How to compare (e.g. in Matlab)?

L1TestPack generates test instances for \((QP_\lambda)\).

- matrix \(A\) can be gaussian, bernoulli, partial DCT, heaviside, random orthonormal rows
- solution \(x^*\) can be chosen gaussian, bernoulli, different dynamic ranges.
- number of non-zero entries in \(x^*\) eligible.
- right side \(b\) will be calculated (we’ll see later...)

www.tu-braunschweig.de/iaa/personal/lorenz/l1testpack
So Many Solvers to Compare...

For (BP):
- SPGL1
- NESTA
- YALL1
- ISAL1
- ...

For (QP$_\lambda$):
- ISTA
- FISTA
- SparSA
- FPC
- ...

How to compare (e.g. in Matlab)?

L1TestPack generates test instances for (QP$_\lambda$).

- matrix $A$ can be gaussian, bernoulli, partial DCT, heaviside, random orthonormal rows
- solution $x^*$ can be chosen gaussian, bernoulli, different dynamic ranges.
- number of non-zero entries in $x^*$ eligible.
- right side $b$ will be calculated (we’ll see later...)

www.tu-braunschweig.de/iaa/personal/lorenz/l1testpack
So Many Solvers to Compare...

For (BP):
- SPGL1
- NESTA
- YALL1
- ISAL1
- ...

For (QP\(\lambda\)):
- ISTA
- FISTA
- SparSA
- FPC
- ...

How to construct test instances?
Sufficient and Necessary Condition for (BP)

**Theorem**

Let $A = [a_1, \ldots, a_n] \in \mathbb{R}^{m \times n}$, $x^* \in \mathbb{R}^n$ and $l = \text{supp}(x^*)$. $x^*$ solves

$$\min \|y\|_1 \text{ s.t. } Ay = Ax^*$$

uniquely if and only if $A_l$ is injective and there exists $w \in \mathbb{R}^m$

$$a_i^T w = \text{sign}(x_i^*), \quad \text{for } x_i^* \neq 0 \text{ resp. } i \in l,$$

$$|a_j^T w| \leq 1, \quad \text{for } x_j^* = 0 \text{ resp. } j \in l^C.$$

**Notation**

For a given $x^* \in \mathbb{R}^n$ we will denote $l \subset \{1, \ldots, n\}$ as the support of $x^*$, hence $l := \text{supp}(x^*) = \{i : x_i^* \neq 0\}$. $l^C$ is the complement of $l$. 
Lemma (Fuchs, 2004)

The vector $x^*$ is a unique solution of $(QP_\lambda)$ if

$$a_j^T(b - Ax^*) = \lambda \text{sign}(x_j^*), \text{ for } x_j^* \neq 0,$$

$$|a_j^T(b - Ax^*)| \leq \lambda, \text{ for } x_j^* = 0, \text{ and }$$

$A_I$ injective

holds.
Optimality Conditions for \((QP_{\lambda})\)

**Lemma (Fuchs, 2004)**

The vector \(x^*\) is a solution of \((QP_{\lambda})\) if and only if

\[
a_i^T (b - Ax^*) = \lambda \text{sign}(x_i^*), \text{ for } x_i^* \neq 0,
\]

\[
|a_j^T (b - Ax^*)| \leq \lambda, \text{ for } x_j^* = 0
\]

holds.

\[
\text{Sign}(x^*)_i = \text{sign}(x_i^*) \text{ if } x_i^* \neq 0
\]

\[
\text{Sign}(x^*)_j \in [-1, 1] \text{ if } x_j^* = 0
\]
Lemma (Fuchs, 2004)

The vector \( x^\ast \) is a solution of \((QP_\lambda)\) if and only if

\[
A^T (b - Ax^\ast) \in \lambda \text{Sign}(x^\ast)
\]

holds.

\[
\text{Sign}(x^\ast)_i = \text{sign}(x^\ast_i) \text{ if } x^\ast_i \neq 0
\]

\[
\text{Sign}(x^\ast)_j \in [-1, 1] \text{ if } x^\ast_j = 0
\]
Optimality Conditions for $(QP_{\lambda})$

**Lemma**

The vector $x^*$ is a solution of $(QP_{\lambda})$ if and only if

$$A^T(b - Ax^*) \in \lambda \text{Sign}(x^*)$$

holds.

**Idea**

For a given $A$ and $\lambda \geq 0$, choose $x^*$ and construct $b$. 
Constructing instances for \((QP_\lambda)\)

**Theorem (Lorenz, 2011)**

Let \(A \in \mathbb{R}^{m \times n}, \lambda \geq 0, x^* \in \mathbb{R}^n\) and \(z \in \text{rg}(A^T) \cap \text{Sign}(x^*)\). Then for any \(w\) such that \(A^T w = z\) and \(b = \lambda w + Ax^*\), it holds that \(x^*\) solves

\[
\min_y \frac{1}{2} \|Ay - b\|_2^2 + \lambda \|y\|_1.
\]

**Proof.**

\[
A^T(b - Ax^*) = A^T(\lambda w + Ax^* - Ax^*) = \lambda A^T w \in \lambda \text{Sign}(x^*)
\]
Constructing instances for \((QP_\lambda)\)

**Theorem (Lorenz, 2011)**

Let \(A \in \mathbb{R}^{m \times n}, \lambda \geq 0, x^* \in \mathbb{R}^n\) and \(z \in \text{rg}(A^T) \cap \text{Sign}(x^*)\).

Then for any \(w\) such that \(A^Tw = z\) and \(b = \lambda w + Ax^*\), it holds that \(x^*\) solves

\[
\min_y \frac{1}{2} \|Ay - b\|_2^2 + \lambda \|y\|_1.
\]

**Definition (Dual Certificate)**

The element \(w \in \mathbb{R}^m\) satisfying

\[
A^Tw \in \text{Sign}(x^*)
\]

is called **dual certificate**.
What to construct?

Parameters to solve

\[ \min_y \frac{1}{2} \| Ay - b \|_2^2 + \lambda \| y \|_1. \]

uniquely:
- matrix \( A \in \mathbb{R}^{m \times n} \),
- solution \( x^* \in \mathbb{R}^n \),
- index sets
  \[ I^+ = \{ i : x^*_i \geq 0 \}, \]
  \[ I^- = \{ j : x^*_j \leq 0 \}, \]
- dual certificate \( w \in \mathbb{R}^m \)
  satisfying
  \[ A^T w \in \text{Sign}(x^*). \]
What to construct?

Parameters to solve

\[
\min_y \frac{1}{2} \|Ay - b\|_2^2 + \lambda \|y\|_1.
\]

uniquely:

- matrix \( A \in \mathbb{R}^{m \times n} \),
- solution \( x^* \in \mathbb{R}^n \),
- index sets
  \( l^+ = \{i : x_i^* \geq 0\} \),
  \( l^- = \{j : x_j^* \leq 0\} \),
- dual certificate \( w \in \mathbb{R}^m \) satisfying \( A^T w \in \text{Sign}(x^*) \).

Construction 1:

- Given \( A, l^+ \) and \( l^- \)
- Find \( w \).
- Choose \( x^* \) and \( \lambda \) and construct \( b = \lambda w + Ax^* \).
What to construct?

Parameters to solve

\[
\min_y \frac{1}{2} \| Ay - b \|_2^2 + \lambda \| y \|_1.
\]

uniquely:

- matrix \( A \in \mathbb{R}^{m \times n} \),
- solution \( x^* \in \mathbb{R}^n \),
- index sets
  \[
  I^+ = \{ i : x_i^* \geq 0 \},
  I^- = \{ j : x_j^* \leq 0 \},
  \]
- dual certificate \( w \in \mathbb{R}^m \)
  satisfying
  \( A^T w \in \text{Sign}(x^*) \).

Construction 1:
- Given \( A \), \( I^+ \) and \( I^- \)
- Find \( w \).
- Choose \( x^* \) and \( \lambda \) and construct
  \( b = \lambda w + Ax^* \).

Construction 2:
- Given \( A \)
- Find \( w \), \( I^+ \) and \( I^- \).
- Choose \( x^* \) and \( \lambda \) and construct
  \( b = \lambda w + Ax^* \).

Notation

Since \( I^+ \) and \( I^- \) are prescribed before \( x^* \) is chosen,

\( \text{Sign}(x^*) \)

is a synonym for \( I^+ \) and \( I^- \).
What to construct?

Parameters to solve

$$\min_y \frac{1}{2} \| Ay - b \|_2^2 + \lambda \| y \|_1.$$ 

uniquely:

- matrix $A \in \mathbb{R}^{m \times n}$,
- solution $x^* \in \mathbb{R}^n$,
- index sets 
  
  $I^+ = \{ i : x_i^* \geq 0 \}$,
  
  $I^- = \{ j : x_j^* \leq 0 \}$,
- dual certificate $w \in \mathbb{R}^m$ satisfying 
  
  $A^T w \in \text{Sign}(x^*)$.

Construction 1:

- Given $A$, $I^+$ and $I^-$
- Find $w$.
- Choose $x^*$ and $\lambda$ and construct 
  
  $b = \lambda w + A x^*$.

Construction 2:

- Given $A$
- Find $w$, $I^+$ and $I^-$.
- Choose $x^*$ and $\lambda$ and construct 
  
  $b = \lambda w + A x^*$.
Parameters to solve

\[
\min_{y} \frac{1}{2} \|Ay - b\|_2^2 + \lambda \|y\|_1.
\]

uniquely:

- matrix \( A \in \mathbb{R}^{m \times n} \),
- solution \( x^* \in \mathbb{R}^n \),
- index sets
  \[
  l^+ = \{ i : x^*_i \geq 0 \},
  l^- = \{ j : x^*_j \leq 0 \},
  \]
- dual certificate \( w \in \mathbb{R}^m \) satisfying
  \[
  A^Tw \in \text{Sign}(x^*).
  \]

Construction 1:

- Given \( A, l^+ \) and \( l^- \)
- Find \( w \).
- Choose \( x^* \) and \( \lambda \) and construct
  \[
  b = \lambda w + Ax^*.
  \]

Construction 2:

- Given \( A \)
- Find \( w, l^+ \) and \( l^- \).
- Choose \( x^* \) and \( \lambda \) and construct
  \[
  b = \lambda w + Ax^*.
  \]
Constructing instances for \((QP_\lambda)\)

**Idea**

For a given \(A \in \mathbb{R}^{m \times n}\) and \(\lambda \geq 0\) choose \(x^* \in \mathbb{R}^n\) and construct \(b \in \mathbb{R}^m\).

- Specify \(A \in \mathbb{R}^{m \times n}\), \(I\) and restricted sign-vector \(z_I, |z_i| = 1, i \in I\).
- Construct \(z \in \mathbb{R}^n, |z_j| \leq 1, j \in I^C\), and solve \(A^T w = z\).
- Choose \(\lambda \geq 0\) and \(x^*\) according to \(z\) and set

\[
b = \lambda w + A x^*.\]

- Vector \(x^*\) solves \((BP)\) uniquely, \(x^* = \arg \min_y \|y\|_1\) s.t. \(Ay = Ax^*\).
- Vector \(x^*\) solves \((QP_\lambda)\) uniquely,

\[
x^* = \arg \min_y \frac{1}{2} \|Ay - b\|_2^2 + \lambda \|y\|_1.\]
Constructing instances for $(QP_\lambda)$

**Idea**

For a given $A \in \mathbb{R}^{m \times n}$ and $\lambda \geq 0$ choose $x^* \in \mathbb{R}^n$ and construct $b \in \mathbb{R}^m$.

- Specify $A \in \mathbb{R}^{m \times n}$, $I$ and restricted sign-vector $z_i, |z_i| = 1, i \in I$.
- Construct $z \in \mathbb{R}^n, |z_j| \leq 1, j \in I^C$, and solve $A^T w = z$.
- Choose $\lambda \geq 0$ and $x^*$ according to $z$ and set
  \[ b = \lambda w + Ax^*. \]
- Vector $x^*$ solves (BP) uniquely, $x^* = \arg\min_y \|y\|_1$ s.t. $Ay = Ax^*$.
- Vector $x^*$ solves $(QP_\lambda)$ uniquely,
  \[ x^* = \arg\min_y \frac{1}{2}\|Ay - b\|_2^2 + \lambda\|y\|_1. \]
Constructing instances for \((QP_\lambda)\)

**Idea**

For a given \(A \in \mathbb{R}^{m \times n}\) and \(\lambda \geq 0\) choose \(x^* \in \mathbb{R}^n\) and construct \(b \in \mathbb{R}^m\).

- Specify \(A \in \mathbb{R}^{m \times n}\), \(I\) and restricted sign-vector \(z_i, |z_i| = 1, i \in I\).
- Construct \(z \in \mathbb{R}^n, |z_j| \leq 1, j \in I^C\), and solve \(A^T w = z\).
- Choose \(\lambda \geq 0\) and \(x^*\) according to \(z\) and set
  \[
  b = \lambda w + Ax^*.
  \]
- Vector \(x^*\) solves (BP) uniquely, \(x^* = \arg\min_y \|y\|_1\) s.t. \(Ay = Ax^*\).
- Vector \(x^*\) solves \((QP_\lambda)\) uniquely,
  \[
  x^* = \arg\min_y \frac{1}{2}\|Ay - b\|_2^2 + \lambda \|y\|_1.
  \]
Constructing Dual Certificate

- **Method 1:**
  Find \( z \in \text{rg}(A^T) \cap \text{Sign}(x^*) \), solve \( A^T w = z \).
  - by Alternating Projection,
  - \( z_j \approx \pm 1 \), for some \( j \in I^C \).

- **Method 2:**
  \[
  \min_z \| z_{I^C} \|_2^2 \text{ s.t. } z \in \text{rg}(A^T) \cap \text{Sign}(x^*) .
  \]
Constructing Dual Certificate

- **Method 1:**
  Find $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$, solve $A^T w = z$.
  - by Alternating Projection,
  - $z_j \approx \pm 1$, for some $j \in I^C$.

Test: matrix gaussian, size $100 \times 200$, solution gaussian, sparsity 10.
Constructing Dual Certificate

- **Method 1:**
  Find $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$, solve $A^T w = z$.
  - by Alternating Projection,
  - $z_j \approx \pm 1$, for some $j \in I^C$.

- **Method 2:**
  $\min_z \|z|_I\|_2^2$ s.t. $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$.

- **Method 3:**
  $\min_z \|P_{I^C}z\|_\infty$ s.t. $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$. 

May 20, 2012 | Kruschel, Lorenz
Constructing Dual Certificate

- **Method 1:**
  Find $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$, solve $A^T w = z$.
  - by Alternating Projection,
  - $z_j \approx \pm 1$, for some $j \in I^C$.

- **Method 2:**
  \[
  \min_z \|z_{I^C}\|_2^2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*) .
  \]
  - Same $z$ as in Method 1. (see Deutsch (2001))
Constructing Dual Certificate

- **Method 1:**
  Find $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$, solve $A^T w = z$.
  - by Alternating Projection,
  - $z_j \approx \pm 1$, for some $j \in I^C$.

- **Method 2:**
  $\min_z \|Z_{I^C}\|_2^2$ s.t. $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$.
  - Same $z$ as in Method 1. (see Deutsch (2001))

- **Method 2.1:**
  $\min_z \|Z_{I^C}\|_2^2$ s.t. $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$, $\|Z_{I^C}\|_\infty \leq \gamma$.
  - Given $\gamma \in (0, 1)$. 
Constructing Dual Certificate

- **Method 1:**
  Find \( z \in \text{rg}(A^T) \cap \text{Sign}(x^*) \), solve \( A^T w = z \).
  - by Alternating Projection,
  - \( z_j \approx \pm 1 \), for some \( j \in I^C \).

- **Method 2:**
  \[ \min_z \| z_{I^C} \|_2^2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*) \]
  - Similar to Method 1.

- **Method 2.1:**
  \[ \min_z \| z_{I^C} \|_2^2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*), \| z_{I^C} \|_\infty \leq \gamma = 0.9 \]
  - Might be infeasible, but \( \| z_{I^C} \|_\infty \leq 1 \), e.g.

\[
A = \begin{pmatrix}
1 & 0.5 & 0 \\
0 & 0.45 & -1
\end{pmatrix},
I^+ = \{1\}, I^- = \{3\}.
\]

- Decreasing \( \gamma \)
Constructing Dual Certificate

- **Method 1:**
  Find $z \in \text{rg}(A^T) \cap \text{Sign}(x^*)$, solve $A^Tw = z$.
  - by Alternating Projection,
  - $z_j \approx \pm 1$, for some $j \in I^c$.

- **Method 2:**
  \[
  \min_z \|z_{I^C}\|_2^2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*) .
  \]
  - Similar to Method 1.

- **Method 2.1:**
  \[
  \min_z \|z_{I^C}\|_2^2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*), \quad \|z_{I^C}\|_\infty \leq \gamma = 0.9 .
  \]
  - Might be infeasible, but $\|z_{I^C}\|_\infty \leq 1$.

- **Method 3:**
  \[
  \min_z \|z_{I^C}\|_\infty \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*) .
  \]


Constructing Dual Certificate

- **Method 3:**
  \[
  \min_z \| z \|_{\infty} \text{ s.t. } z \in \text{rg}(A^T) \cap \text{Sign}(x^*)
  \]
  - is equivalent to
  \[
  \min_w \| A^T_{/C} w \|_{\infty} \text{ s.t. } A^T_{/C} w = \text{sign}(x^*)_I \text{ and }
  \begin{bmatrix}
  A^T_{/C} \\
  -A^T_{/C}
  \end{bmatrix} w \leq \begin{pmatrix}
  1_{/C} \\
  1_{/C}
  \end{pmatrix},
  \]
  - is realizable as a Linear Program.
Tested Method: \( \min_w \|A_T w\|_\infty \) s.t. \( A_T w \in \text{Sign}(x^*) \)

\( A \) and \( x^* \) chosen by L1TestPack

Used Parameter:
- Gaussian distributed matrix with varying size,
- Bernoulli matrix with varying size, entries \( = \pm 1 \) randomly,
- Partial DCT matrix, generated through uniform sampling of \( m \) rows from the full Discrete Cosine Transform matrix.

Measured Values:
- Error on Support, \( \|A_T w \pm 1\|_2 \),
- Optimized value, \( \|A_T w\|_\infty \),
- Time to solve the problem,
Tested Method: \( \min_w \| A_{\tilde{I}C}^T w \|_{\infty} \) s.t. \( A^T w \in \text{Sign}(x^*) \)

A and \( x^* \) chosen by L1TestPack

Used Parameter:
- Gaussian distributed matrix with varying size,
- Bernoulli matrix with varying size, entries \( = \pm 1 \) randomly,
- Partial DCT matrix, generated through uniform sampling of \( m \) rows from the full Discrete Cosine Transform matrix.

Measured Values:
- Error on Support, \( \| A_{\tilde{I}}^T w \pm 1 \|_2 \approx 10^{-12} \),
- Optimized value, \( \| A_{\tilde{I}C}^T w \|_{\infty} \),
- Time to solve the problem,
Constructing Dual Certificate - Test

Uniform norm of $A_{cw}^T$ with $A$ as $m \times 1000$ matrix, Sparsity: 17 resp.

$||A_{cw}^T||_\infty$

$m$: Number of Rows in $A$

- Green: Gaussian
- Red: Bernoulli
- Blue: Partial DCT
Uniform norm of $A_{cw}^T$ with $A$ as $100 \times n$ matrix, Sparsity: 17 resp. 1
Time, $A$ as $m \times 1000$ matrix, Sparsity: 17 resp. 1

- Gaussian
- Bernoulli
- Partial DCT

$m$: Number of Rows in $A$
Parameters to solve

\[
\min_y \frac{1}{2} \|Ay-b\|_2^2 + \lambda \|y\|_1.
\]

uniquely:

- matrix \( A \in \mathbb{R}^{m \times n} \),
- solution \( x^* \in \mathbb{R}^n \),
- index sets
  \( I^+ = \{ i : x_i^* \geq 0 \} \),
  \( I^- = \{ j : x_j^* \leq 0 \} \),
- dual certificate \( w \in \mathbb{R}^m \) satisfying
  \( A^T w \in \text{Sign}(x^*) \).

Construction 1:
- Given \( A, I^+ \) and \( I^- \)
- Find \( w \).
- Choose \( x^* \) and \( \lambda \) and construct
  \( b = \lambda w + Ax^* \).

Construction 2:
- Given \( A \)
- Find \( w, I^+ \) and \( I^- \).
- Choose \( x^* \) and \( \lambda \) and construct
  \( b = \lambda w + Ax^* \).

May 20, 2012 | Kruschel, Lorenz

Constructing Test Instances for Sparse Recovery Algorithms | Page 16
What to construct?

Construction 2:

- Given $A$
- Find $w$, $I^+$ and $I^-$.
- Choose $x^*$ and $\lambda$ and construct $b = \lambda w + Ax^*$.

Additional Goal:

Find index set $I^+ \cup I^-$ with largest possible cardinality.
What to construct?

Construction 2:
- Given $A$
- Find $w$, $I^+$ and $I^-$. 
- Choose $x^*$ and $\lambda$ and construct $b = \lambda w + A x^*$.

Additional Goal:

Find index set $I^+ \cup I^-$ with largest possible cardinality.

$\rightarrow$ any other cardinality $\leq \text{rank}(A)$ chooseable
Recoverable Indices

Definition

An index set $I \subset \{1, ..., n\}$ is called recoverable for $A \in \mathbb{R}^{m \times n}$, if there exists $x^* \in \mathbb{R}^n$, $I = \text{supp}(x^*)$, solving uniquely

$$\min \|y\|_1 \text{ s.t. } Ay = Ax^*.$$ 

A recoverable index set $I$ is called maximal, if there exists no recoverable index set $J$ satisfying $|I| \not\leq |J|$.

The index set $I$ is recoverable if there exists $w \in \mathbb{R}^m$ satisfying

$$|a_i^T w| = 1, \text{ for } i \in I,$$

$$|a_j^T w| \leq 1, \text{ for } j \in I^C,$$

$A_I$ is injective.
Recoverable Indices

**Definition**

An index set \( I \subset \{1, \ldots, n\} \) is called **recoverable** for \( A \in \mathbb{R}^{m \times n} \), if there exists \( x^* \in \mathbb{R}^n, I = \text{supp}(x^*) \), solving uniquely

\[
\min \|y\|_1 \text{ s.t. } Ay = Ax^*.
\]

A recoverable index set \( I \) is called **maximal**, if there exists no recoverable index set \( J \) satisfying \( |I| \not\leq |J| \).

The index set \( I \) is recoverable if there exists \( w \in \mathbb{R}^m \) satisfying

\[
|a_i^T w| = 1, \text{ for } i \in I,
\]

\[
|a_j^T w| \leq 1, \text{ for } j \in I^C,
\]

\( A_I \) is injective.
Recoverable Indices

Definition

An index set $I \subset \{1, \ldots, n\}$ is called recoverable for $A \in \mathbb{R}^{m \times n}$, if there exists $x^* \in \mathbb{R}^n$, $I = \text{supp}(x^*)$, solving uniquely

$$\min \|y\|_1 \text{ s.t. } Ay = Ax^*. $$

A recoverable index set $I$ is called maximal, if there exists no recoverable index set $J$ satisfying $|I| \prec |J|$. 

The index set $I$ is recoverable if there exists $w \in \mathbb{R}^m$ satisfying

$$|a_i^T w| = 1, \text{ for } i \in I,$$

$$|a_j^T w| \preceq 1, \text{ for } j \in I^C, \quad \text{Recov. index set } I \text{ depends on } A \text{ and } w.$$

$A_I$ is injective.
Recoverable Indices

Definition

An index set $I \subset \{1, \ldots, n\}$ is called recoverable for $A \in \mathbb{R}^{m \times n}$, if there exists $x^* \in \mathbb{R}^n$, $I = \text{supp}(x^*)$, solving uniquely

$$\min \|y\|_1 \text{ s.t. } Ay = Ax^*.$$ 

A recoverable index set $I$ is called maximal, if there exists no recoverable index set $J$ satisfying $|I| \lesssim |J|$.

The index set $I$ is recoverable if there exists $w \in \mathbb{R}^m$ satisfying

$$|a_i^T w| = 1, \text{ for } i \in I,$$

$$|a_j^T w| \lesssim 1, \text{ for } j \in I^C,$$

Recov. index set $I$ depends on $A$ and $w$. $A_I$ is injective. How to compute a maximal index set?
Find Maximal Recoverable Index Set

- Unit Cube in $\mathbb{R}^3$
- Cube $= \{(x, y, z) : |x| \leq 1, |y| \leq 1, |z| \leq 1\}$
Find Maximal Recoverable Index Set

- Unit Cube in $\mathbb{R}^3$
- Cube $= \{(x, y, z) : |x| \leq 1, |y| \leq 1, |z| \leq 1\}$
Find Maximal Recoverable Index Set

\[ \mathbb{R}^3 \]

- Range of \( A^T \)
- \( \text{rg}(A^T) = \{z : \exists w : A^T w = z\} \)
Find Maximal Recoverable Index Set

- Intersection of Cube and $\text{rg}(A^T)$
- $\exists z \in \text{Cube} \cap \text{rg}A^T$
Find Maximal Recoverable Index Set

- Intersection of Cube and $\text{rg}(A^T)$
- $\exists z \in \text{Cube} \cap \text{rg}A^T$
- Recoverable Index Sets: $I = I^+ \cup I^-$
  - $I^+ = \{2\}$
  - $I^+ = \{2\}, I^- = \{1\}$
  - $I^+ = \{1\}$
  - $I^+ = \{1, 2\}$
  - etcetera
Find Maximal Recoverable Index Set

- Intersection of Cube and \( \text{rg}(A^T) \)
- \( \exists z \in \text{Cube} \cap \text{rg} A^T \)
- Recoverable Index Sets: \( I = I^+ \cup I^- \)
  - \( I^+ = \{2\} \)
  - \( I^+ = \{2\}, I^- = \{1\} \)
  - \( I^+ = \{1\} \)
  - \( I^+ = \{1,2\} \)
  - etcetera
Find Maximal Recoverable Index Set

- Intersection of Cube and $\text{rg}(A^T)$
- $\exists z \in \text{Cube} \cap \text{rg}A^T$
- Recoverable Index Sets: $I = I^+ \cup I^-$
  - $I^+ = \{2\}$
  - $I^+ = \{2\}, I^- = \{1\}$
  - $I^+ = \{1\}$
  - $I^+ = \{1,2\}$
  - etcetera
Find Maximal Recoverable Index Set

- Intersection of Cube and \( \text{rg}(A^T) \)
- \( \exists z \in \text{Cube} \cap \text{rg}A^T \)
- Recoverable Index Sets: \( I = I^+ \cup I^- \)
  - \( I^+ = \{2\} \)
  - \( I^+ = \{2\}, I^- = \{1\} \)
  - \( I^+ = \{1\} \)
  - \( I^+ = \{1, 2\} \)
  - etcetera
Find Maximal Recoverable Index Set

- Intersection of Cube and \( \text{rg}(A^T) \)
- \( \exists z \in \text{Cube} \cap \text{rg}A^T \)
- Recoverable Index Sets: \( I = I^+ \cup I^- \)
  - \( I^+ = \{2\} \)
  - \( I^+ = \{2\}, I^- = \{1\} \)
  - \( I^+ = \{1\} \)
  - \( I^+ = \{1, 2\} \)
  - etcetera
Find Maximal Recoverable Index Set:

- Start at $w^{(0)} = (0, 0)$. 

$\mathbb{R}^3$
Find Maximal Recoverable Index Set:

- Start at $w^{(0)} = (0, 0)$. 
- Find $j \in \text{argmax}_i \|a_i\|_2$. 

$A^T w^{(0)}$
Find Maximal Recoverable Index Set:

- Start at $w^{(0)} = (0, 0)$.
- Find $j \in \arg \max_i \|a_i\|_2$.
- Set $w^{(1)} := \frac{1}{\|a_j\|^2} a_j$ and $I_1 := \{j\}$. 
Find Maximal Recoverable Index Set:

- Set \( w^{(1)} := \frac{1}{\|a_j\|_2^2} a_j \)
  and \( l_1 := \{j\} \).
Find Maximal Recoverable Index Set:

- Set \( w^{(1)} := \frac{1}{\|a_j\|_2^2} a_j \) and \( l_1 := \{j\} \).
- Choose \( h_1 \in \ker A^T_{l_1} \backslash \{0\} \).
Find Maximal Recoverable Index Set:

- Set \( w^{(1)} := \frac{1}{\|a_j\|_2^2} a_j \)
  and \( I_1 := \{j\} \).
- Choose \( h_1 \in \ker A^T_{I_1} \setminus \{0\} \).
- Find \( \alpha \in \mathbb{R} \) such that
  \( \|A^T_{I_1} (w^{(1)} + \alpha h_1)\|_{\infty} = 1 \).
Find Maximal Recoverable Index Set:

- Set \( w^{(1)} := \frac{1}{\|a_j\|_2^2} a_j \) and \( l_1 := \{j\} \).
- Choose \( h_1 \in \ker A^T_{l_1} \{0\} \).
- Find \( \alpha \in \mathbb{R} \) such that \( \|A^T_{l_1^c} (w^{(1)} + \alpha h_1)\|_\infty = 1 \).
- Set \( w^{(2)} := w^{(1)} + \alpha h_1 \) and \( l_2 := \{i : |w_i^{(2)}| = 1\} \).
Find Maximal Recoverable Index Set:

- Set $\mathbf{w}^{(2)} := \mathbf{w}^{(1)} + \alpha \mathbf{h}_{1}$
  and $I_2 := \{ i : |w_i^{(2)}| = 1 \}$. 
Find Maximal Recoverable Index Set:

- Set \( w^{(2)} := w^{(1)} + \alpha h_1 \)
  and \( l_2 := \{i : |w_i^{(2)}| = 1\} \).
- \( \text{rank}(A_{l_2}) = \text{rank}(A) \)
  \( \rightarrow \) End Algorithm
Find Maximal Recoverable Index Set:

- Set \( w^{(2)} := w^{(1)} + \alpha h_1 \)
  and \( l_2 := \{ i : |w_i^{(2)}| = 1 \} \).
- \( \text{rank}(A_{l_2}) = \text{rank}(A) \) → End Algorithm
- Set \( w := w^{(2)} \) and \( l := l_2 \).
Find Maximal Recoverable Index Set:

- Set \( w^{(2)} := w^{(1)} + \alpha h_1 \) and \( l_2 := \{ i : |w_i^{(2)}| = 1 \} \).
- \( \text{rank}(A_{l_2}) = \text{rank}(A) \)
- → End Algorithm
- Set \( w := w^{(2)} \) and \( l := l_2 \).

The index set \( l = l^+ = \{1, 2\} \) is a maximal recoverable index set.
Find Maximal Recoverable Index Set

Special Cases:

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]
Find Maximal Recoverable Index Set

$\mathbb{R}^3$

Special Cases:

$$A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix}$$
Find Maximal Recoverable Index Set

$\mathbb{R}^3$

$A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix}$

Special Cases:

- Start at $w^{(0)} = (0,0)$. 

Start at $w^{(0)} = (0,0)$. 

$\mathbb{R}^3$
Find Maximal Recoverable Index Set

$\mathbb{R}^3$

Special Cases:

$A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix}$

- Start at $w^{(0)} = (0, 0)$.
- Find $j \in \arg\max_i \|a_i\|_2$. It is $j = 1$. 

$A^T w^{(0)}$
Find Maximal Recoverable Index Set

Special Cases:

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

- Start at \( w^{(0)} = (0,0) \).
- Find \( j \in \arg\max_i \|a_i\|_2 \). It is \( j = 1 \).
- Set \( w^{(1)} = \frac{1}{\|a_1\|_2^2} A^T a_1 \), \( I_1 = \{1\} \).
Find Maximal Recoverable Index Set

Special Cases:

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

- Set \( w^{(1)} = \frac{1}{\|a_1\|_2^2} A^T a_1 \).
- Choose \( h \in \ker A^T_{l_1} \backslash \{0\} \).
Find Maximal Recoverable Index Set

Special Cases:
\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

- Set \( w^{(1)} = \frac{1}{\|a_1\|_2^2} A^T a_1 \).
- Choose \( h \in \ker A_{l_1}^T \setminus \{0\} \).
- Find \( \alpha \in \mathbb{R} \) s.t.
  \[ \|A_{l_1}^T(w^{(1)} + \alpha h)\|_\infty = 1 \]
  and set \( l_2 = \{ i : |a_i^T(w^{(1)} + \alpha h)| = 1 \} \).
Find Maximal Recoverable Index Set

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

**Special Cases:**
- Set \( w^{(1)} = \frac{1}{\|a_1\|^2} A^T a_1 \).
- Choose \( h \in \ker A^T_{I_1} \setminus \{0\} \).
- Find \( \alpha \in \mathbb{R} \) s.t.
  \[ \| A^T_{I_{1}^C} (w^{(1)} + \alpha h) \|_\infty = 1 \]
  and set \( I_2 = \{ i : |a_i^T (w^{(1)} + \alpha h)| = 1 \} \).
- \( |I_2| \geq \text{rank}(A_{I_2}) \)
Find Maximal Recoverable Index Set

Special Cases:

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

- Set \( w^{(1)} = \frac{1}{\|a_1\|^2_2} A^T a_1 \).
- Choose \( h \in \ker A^T_{l_1}\setminus\{0\} \).
- Find \( \beta \neq \alpha \) s.t.
  \[ \|A^T_{l_1^c}(w^{(1)} + \beta h)\|_\infty = 1 \]
  and set \( l_3 = \{i: |a_i^T(w^{(1)} + \beta h)| = 1\} \).
Find Maximal Recoverable Index Set

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

Special Cases:
- Set \( w^{(1)} = \frac{1}{\|a_1\|_2^2} A^T a_1 \).
- Choose \( h \in \ker A^T_{l_1} \backslash \{0\} \).
- Find \( \beta \neq \alpha \) s.t.
  \[ \| A^T_{l_1} (w^{(1)} + \beta h) \|_\infty = 1 \]
  and set \( l_3 = \{i : |a_i^T (w^{(1)} + \beta h)| = 1\} \).
- \( \text{rank}(A_{l_3}) = \text{rank}(A) \) → Stop Algorithm
Find Maximal Recoverable Index Set

Special Cases:

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

- Set \( w := w^{(1)} + \alpha h \) and \( l := l_3 \).
Find Maximal Recoverable Index Set

\[ A = \begin{pmatrix} -3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \]

- Set \( w := w^{(1)} + \alpha h \) and \( l := l_3 \).

The index set \( l = l^+ = \{1, 2\} \) is the maximal recoverable index set.

Special Case 2
Find Maximal Recoverable Index Set - Test

Used Parameter:

- Gaussian distributed matrix with varying size,
- Bernoulli matrix with varying size, entries $= \pm 1$ randomly,
- Partial DCT matrix, generated through uniform sampling of $m$ rows from the full Discrete Cosine Transform matrix.

Measured Values:

- Time to get an index set
- Value $\|A^T w\|_\infty$
Find Maximal Recoverable Index Set - Test
Find Maximal Recoverable Index Set - Test

A as $m \times 1000$ matrix

<table>
<thead>
<tr>
<th>$m$</th>
<th>$|A_{IC}^T w|_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.994</td>
</tr>
<tr>
<td>200</td>
<td>0.997</td>
</tr>
<tr>
<td>300</td>
<td>0.987</td>
</tr>
<tr>
<td>400</td>
<td>0.993</td>
</tr>
<tr>
<td>500</td>
<td>0.992</td>
</tr>
<tr>
<td>600</td>
<td>0.984</td>
</tr>
<tr>
<td>700</td>
<td>0.99993</td>
</tr>
<tr>
<td>800</td>
<td>0.935</td>
</tr>
<tr>
<td>900</td>
<td>0.945</td>
</tr>
</tbody>
</table>

Bernoulli matrix
Conclusion

- One can find test instances for Basis Pursuit / Basis Pursuit Denoising with prescribed sign
- Dual certificate detectable by linear programming or alternating projections
- Found maximal recoverable index set
Future Work

Construction 3:

- Given $w$, $l^+$, and $l^-$
- Find $A$.
- Choose $x^*$ and $\lambda$ and construct $b = \lambda w + Ax^*$.

Dual certificate $w \sim \mathcal{N}(0, 1)$ could be noise to the right side $Ax^*$. 
Thank You For Your Attention

References:

- D. Lorenz: Constructing test imstances for Basis Pursuit Denoising, arXiv.org, 2011 (paper)

Contact Christian Kruschel

- c.kruschel@tu-bs.de
- Facebook, Google+
Constructing Dual Certificate

- **Method 2.1:**
  \[
  \min_z \| P_I c z \|_2^2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*), \quad \| P_I c z \|_\infty \leq \gamma.
  \]
  - Given \( \gamma \in [0, 1] \).

\[ \gamma = 1.0 \]

Test: matrix gaussian, size 100 \( \times \) 200, sparsity 10.
Method 2.1:
\[
\min_z \| P_{\mathcal{C}} z \|_2^2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*), \quad \| P_{\mathcal{C}} z \|_\infty \leq \gamma.
\]
- Given \( \gamma \in [0, 1] \).

\[\gamma = 0.9\]

Test: matrix gaussian, size 100 \( \times \) 200, sparsity 10.
Constructing Dual Certificate

- Method 2.1:
  \[ \min_z \| P_I c z \|_2^2 \text{ s.t. } z \in \text{rg}(A^T) \cap \text{Sign}(x^*), \| P_I c z \|_\infty \leq \gamma. \]
  - Given \( \gamma \in [0, 1] \).

\[ \gamma = 0.8 \]

Test: matrix gaussian, size 100 \times 200, sparsity 10.
Method 2.1:

\[
\min_z \|P_I c z\|^2_2 \quad \text{s.t.} \quad z \in \text{rg}(A^T) \cap \text{Sign}(x^*), \quad \|P_I c z\|_\infty \leq \gamma.
\]

- Given \( \gamma \in [0, 1] \).

\[\gamma = 0.7\]

Test: matrix gaussian, size 100 \( \times \) 200, sparsity 10.
Constructing Dual Certificate

Method 2.1:
\[
\min_z \|P_Ic z\|_2^2 \quad \text{s.t. } z \in \text{rg}(A^T) \cap \text{Sign}(x^*), \|P_Ic z\|_\infty \leq \gamma.
\]
- Given \( \gamma \in [0, 1] \).

\( \gamma = 0.6 \)

Test: matrix gaussian, size 100 \( \times \) 200, sparsity 10.

Infeasible
Find Maximal Recoverable Index Set

\[ \mathbb{R}^3 \]

Special Cases:

\[ A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & -3 & -2 \end{pmatrix} \]
Find Maximal Recoverable Index Set

\[ \mathbb{R}^3 \]

\[ A^T w^{(1)} \]

\[ A^T w^{(0)} \]

Special Cases:

\[ A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & -3 & -2 \end{pmatrix} \]

- \[ w^{(1)} = (0, -\frac{1}{3})^T, I_1 = \{2\} \]
Find Maximal Recoverable Index Set

$\mathbb{R}^3$

$A^T w^{(1)}$

Special Cases:

$A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & -3 & -2 \end{pmatrix}$

- Next possible support $I_2 = [-1, 1, 1]$ leads to $A_{I_2}$ not injective
Find Maximal Recoverable Index Set

Special Cases:
\[
A = \begin{pmatrix}
1 & 0 & -1 \\
2 & -3 & -2
\end{pmatrix}
\]

- Next possible support \( I_3 = [1, 1, -1] \) leads to \( A_{I_3} \) not injective
Find Maximal Recoverable Index Set

\[ \mathbb{R}^3 \]

\[ A^T w^{(1)} \]

Special Cases:

\[ A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & -3 & -2 \end{pmatrix} \]

- Choose one of these supports
Find Maximal Recoverable Index Set

Special Cases:

\[ A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & -3 & -2 \end{pmatrix} \]

- Support \( l_4 = [-1, 0, 1] \) is a maximal rec. support
Find Maximal Recoverable Index Set

$A^T w$

**Special Cases:**

\[
A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & -3 & -2 \end{pmatrix}
\]

- Support $I_4 = [-1, 0, 1]$ is a maximal rec. support
Consider $x = (x_1, ..., x_n)^T \in \mathbb{R}^n, p \in \mathbb{N}\{0\}$

$$\|x\|_p^p := \sum_{i=1}^{n} |x_i|^p.$$  

Example:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|,$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}.$$