Computable Performance Analysis of Sparsity Recovery

Arye Nehorai

Preston M. Green Department of Electrical & Systems Engineering
Washington University in St. Louis, USA

July 2014
Acknowledgements

In collaboration with my former PhD student Gongguo Tang.
Figure Acknowledgements

• Figures on slide 8 and slide 16 are adapted from R. Baraniuk, J. Romberg and M. Wakin’s slides “Tutorial on Compressive Sensing.”


• Figure on slide 14 is from D. M. Maliouto, “A sparse signal reconstruction perspective for source localization with sensor arrays,” Master thesis, MIT, 2003.
Outline

• Introduction

• Sparsity recovery

• Future work
Outline

• Introduction

• Sparsity recovery

• Future work
Introduction

Low-dimensional structures are ubiquitous in signals:

- **Sparse vectors**
  - Compressive sensing
  - MRI
  - Image processing and computer vision

- **Block-sparse vectors**
  - Radar
  - Sensor array processing

- **Low-rank matrices**
  - Collaborative filtering
  - Robust principal component analysis

- **Low-dimensional manifolds**
  - Subspace learning
  - Manifold learning

Exploiting low-dimensional structures enables more accurate signal recovery.
Sparsity Example: Compressive Sensing

• Interest in exploiting sparsity grew recently due to developments in compressive sensing.

• Traditional signal sampling acquires a signal using expensive hi-fidelity sensors, then compresses the data with a loss of fidelity.

• Compressive sensing (CS) combines the acquisition with the compression by sampling the signal in a novel way with less data.

• CS replaces samples with general linear projections, and linear reconstruction with non-linear reconstruction, thus shifting the burden from the hi-fidelity sensing to reconstruction.

• Key assumption: Many natural signals $x$ have sparse representations in some transform domains $\Phi$, i.e., $x = \Phi s$ for some sparse vector $s$. 
Sparsity Example: Compressive Sensing (cont.)

**Figure 1:** Paradigm of compressive sensing.

- **Surprising fact**\(^1,2\): Suffices to use \(m = O(k \log n) \ll n\) linear, non-adaptive, random measurements \(y\) to reconstruct a sparse signal, where \(k = \|s\|_0\) is the sparsity level of \(s\).

- **The reconstruction performance** depends on the sensing matrix \(A\).

---


Sparsity Example: MRI

- MRI images can often be well approximated by piece-wise constant functions.
- Instead of observing the image directly, we observe its Fourier transform coefficients sampled along, e.g., a radial trajectory in the 2D spatial frequency domain.

(a) Logan-Shepp phantom  (b) Fourier transform  (c) Sampling trajectory
Sparsity Example: MRI (cont.)

- Reconstruction using minimum energy (or $\ell_2$ norm) results in many artifacts.
- However, the total variation (TV) minimization, which enforces the piece-wise constant property of the image, recovers the original image exactly.

**Figure 2:** Exploiting sparsity improves the MRI recovery.
Sparsity Example: Image Processing and Computer Vision

- Sparse recovery is also useful in face recognition\(^3\) and single-image super-resolution.

- In face recognition, a given facial image is sparsely represented using a dictionary database.

- The significant coefficients in the representation reveals the person’s identity.

---

Sparsity Example: Image Processing and Computer Vision (cont.)

- This approach yields accurate recognition.
- It is also robust to occlusions and noise corruptions.

![Diagram](image.png)

**Figure 4:** Robust face recognition with corruption.
Sparsity Example: Sensor Arrays

• We can also use sparsity to estimate continuous parameters in nonlinear models.

• Consider, for example, the estimation of directions-of-arrival (DOAs) using a sensor array.

• The narrowband observation model is given by \( y = \tilde{A}(\tilde{\theta})\tilde{x} + w \), where \( \tilde{A}(\tilde{\theta}) \) is the array manifold matrix, and \( \tilde{x} \) is the unknown signal vector and \( w \) is the noise.

• The unknown DOA parameter \( \tilde{\theta} \) is continuous.

---

Sparsity Example: Sensor Arrays (cont.)

- We discretize the parameter space into grid points given as $\theta = [\theta_1, \ldots, \theta_N]^T$ to create a sparse recovery problem.

![Image](image.png)

(a) DOAs of two sources.

(b) DOA discretization.

Figure 5: Sparse modeling for DOA estimation.

- The observation model then becomes linear:

$$y = [\tilde{A}(\theta_1) \cdots \tilde{A}(\theta_N)]x + w = Ax + w,$$

where the entries of $x$ are nonzero if and only if there is a source direction at the corresponding grid point, implying $x$ is sparse.
**Sparsity Example: Sensor Arrays (cont.)**

- Discretization leads to super resolution when there is no basis mismatch.
- Sensing matrix depends on the sensor configuration and discretization strategy.

**Figure 6:** Super resolution results.
Common Theme

- The sparse signal $x$ is observed by a linear model corrupted by noise:
  $$y = Ax + w.$$  
- The sensing matrix $A \in \mathbb{R}^{m \times n}$ projects the set of sparse vectors in $\mathbb{R}^n$ onto a low dimensional space $\mathbb{R}^m$.  
- There exist convex programs that exploit sparsity to recover the signal.  
- The recovery performance highly depends on the sensing matrix $A$.  
- A good matrix $A$ should preserve the structure of the set of sparse vectors.

Our goal: Find computable bounds on recovery errors for a given $A$.  

Computable Bounds 16
Motivations for Computable Performance Analysis

A computable performance analysis would enable us to:

- Quantify the confidence in the reconstructed signal, especially when there are no other ways to justify the correctness of the reconstructed signal.

- Optimize the system design.

Figure 6: Two MRI sampling trajectories. Left: radial, Right: spiral.
Our Contributions

- Introduce a family of functions that quantify the goodness of sensing matrices in sparsity recovery.

- Derive bounds on reconstruction error in terms of these goodness measures for recovery algorithms.

- Design efficient algorithms to compute these goodness measures and bounds.
Outline

- Introduction
- Sparsity recovery
- Future work
Model

Consider the measurement model

\[ y = Ax + w, \]

where

- the signal \( x \in \mathbb{R}^n \) is sparse with \( \ell_0 \)-sparsity level \( \|x\|_0 = k \ll n \),
- the matrix \( A \in \mathbb{R}^{m \times n} \) has \( m \) rows and \( n \) columns with \( m \ll n \),
- the noise vector \( w \in \mathbb{R}^m \) satisfies
  - \( \|w\|_\diamond \leq \varepsilon \), where \( \diamond = 1, 2, \infty \) or
  - \( \|A^T w\|_\infty \leq \lambda \).

If \( w \) is Gaussian, \( w \sim \mathcal{N}(0, \sigma^2 I) \), then for \( \lambda = \sigma \sqrt{\log(n)} \) it satisfies the above last bound with a high probability.
Sparse Signal Recovery

• In the absence of noise, the signal $x$ can be recovered by solving

$$P_0 : \arg\min_x \|x\|_0 \text{ subject to } y = Ax.$$ 

• $P_0$ is a non-convex optimization problem and it is NP hard to solve.\(^5\)

• Convex relaxation methods replace the $\ell_0$ norm with the $\ell_1$ norm.

Existing Recovery Algorithms

- **Basis Pursuit**: \( \min_{z \in \mathbb{R}^n} \|z\|_1 \) subject to \( \|y - Az\|_\diamond \leq \varepsilon \)

- **Dantzig Selector**: \( \min_{z \in \mathbb{R}^n} \|z\|_1 \) subject to \( \|A^T(y - Az)\|_\infty \leq \lambda \)

- **LASSO Estimator**: \( \min_{z \in \mathbb{R}^n} \frac{1}{2}\|y - Az\|_2^2 + \lambda \|z\|_1 \)

**Figure 6**: Geometry of \( \ell_1 \) minimization in the noise-free case.
Performance Bounds

• Noise-free Case:
  – **Exact recovery**: Upper bound on $\|x\|_0$ for exact recovery.
  – We will derive sufficient conditions, or maximum sparsity level below which exact recovery is guaranteed for existing sparse-recovery algorithms.
  – Higher bound (sparsity level) is better.

• Noisy Case:
  – **Reconstruction**: Upper bounds on the reconstruction error $\|\hat{x} - x\|_2$ for existing sparse-recovery algorithms.
  – We will derive reconstruction bounds on sparse-recovery algorithms.
  – Lower bounds are better.
Previous Approaches for Performance Analysis

Restricted Isometry Constant (RIC)\(^6\):

\[
\delta_k(A) = \max_{z: z \neq 0} \mid \mid A z \mid \mid_2^2 / \mid \mid z \mid \mid_2^2 - 1 \mid \text{ subject to } \mid \mid z \mid \mid_0 \leq k, 
\]

– error bounds: If \( \mid \mid w \mid \mid_2 \leq \varepsilon \), then the recovery error of the Basis Pursuit is bounded as

\[
\mid \mid \hat{x} - x \mid \mid_2 \leq \frac{4\sqrt{1 + \delta_2k(A)}}{1 - (1 + \sqrt{2})\delta_2k(A)}\varepsilon,
\]

– computational difficulty: No practical way to compute \( \delta_k(A) \) exactly.

---

Previous Approaches for Performance Analysis (cont.)

Mutual Coherence (MC)\(^7\):

\[ \mu(A) = \max_{i \neq j} \frac{|A_i^T A_j|}{\|A_i\|_2 \|A_j\|_2}, \]

- sufficient condition: if \( \|x\|_0 \leq \frac{1}{2} \left( 1 + \frac{1}{\mu(A)} \right) \), then we get exact recovery (noise-free case) via Basis Pursuit. However, this condition is weak.
- If \( \|w\|_2 \leq \varepsilon \), then the recovery error of the Basis Pursuit is bounded as\(^8\)

\[ \|\hat{x} - x\|_2 \leq \frac{2}{1 - \mu(A)(4k - 1)} \varepsilon, \]

which is also weak and limited (see later).


Related Work on Computable Performance Analysis

The following papers included sufficient and necessary conditions on the maximal sparsity level for exact recovery of sparse vector in the noise-free case:


Our Innovations

- **Verification**: More efficient algorithms to verify exact recovery without noise.
- **Computation**: Computable performance bounds on recovery errors in noise.
Quality Measure $\omega_\diamond(Q, s)$

- For $s \in [1, n]$ and $A \in \mathbb{R}^{m \times n}$, we define
  
  $$\omega_\diamond(Q, s) = \min_{z: z \neq 0} \frac{\|Qz\|_\diamond}{\|z\|_\infty} \quad \text{subject to} \quad \frac{\|z\|_1}{\|z\|_\infty} \leq s,$$

  where $Q = A$ or $A^T A$, and $\diamond = 1, 2$, or $\infty$.

- $s$ is a measure of the sparsity of $z$: Smaller $s$ implies more sparse $z$; also larger $\omega_\diamond(Q, s)$ and better reconstruction performance (see later).

- Without the sparsity constraint, with $\ell_\infty$ norm replaced by $\ell_2$ and $\diamond$ replaced with $\ell_2$, $\omega_\diamond(Q, s)$ is the minimal singular value of $Q$.

- $\omega_\diamond(Q, s)$ is a measure of the incoherence (quality) of $A$, and it will determine the performance bounds.

Figure 7: Constraint set in $\mathbb{R}^3$ for $s = 1.4$. 
Theorem 1. Suppose the noise $w$ satisfies $\|w\|_\diamond \leq \varepsilon$, $\|A^T w\|_\infty \leq \lambda$, and $\|A^T w\|_\infty \leq \kappa \lambda$, $\kappa \in (0, 1)$, for the Basis Pursuit, the Dantzig Selector, and the LASSO estimator, respectively, then we have

$$\|\hat{x} - x\|_\infty \leq \frac{2\varepsilon}{\omega_\diamond(A, 2k)}$$

for the Basis Pursuit,

$$\|\hat{x} - x\|_\infty \leq \frac{2\lambda}{\omega_\infty(A^T A, 2k)}$$

for the Dantzig Selector, and

$$\|\hat{x} - x\|_\infty \leq \frac{(1 + \kappa)\lambda}{\omega_\infty(A^T A, 2k/(1 - \kappa))}$$

for the LASSO estimator.

- These error bounds are inversely proportional to $\omega_\diamond(Q, s)$.
- $\omega_\diamond(Q, s) > 0$ implies exact recovery in the noise-free case (where $\varepsilon = 0, \lambda = 0$).
- When the sparsity level $k$ of the signal decreases, $\omega_\diamond(Q, s)$ becomes larger, implying smaller reconstruction error.
Reconstruction Error Bounds (cont.)

The error bounds on the $\ell_1$ and $\ell_2$ norms can be expressed via

$$\|\hat{x} - x\|_1 \leq ck\|\hat{x} - x\|_\infty \quad \text{and} \quad \|\hat{x} - x\|_2 \leq \sqrt{ck}\|\hat{x} - x\|_\infty.$$

## Topics of Next Slides

### Recovery Error Bounds

<table>
<thead>
<tr>
<th>Noise Free</th>
<th>Noisy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification of $\omega^\diamond &gt; 0$</td>
<td>Computation of $\omega^\diamond$</td>
</tr>
<tr>
<td>General $A$</td>
<td>General $A$</td>
</tr>
<tr>
<td>Fourier $A$</td>
<td>Fourier $A$</td>
</tr>
<tr>
<td>Numerical Examples</td>
<td>Numerical Examples</td>
</tr>
<tr>
<td>Numerical Examples</td>
<td>Numerical Examples</td>
</tr>
</tbody>
</table>
Verification of $\omega_\diamond(Q, s) > 0$: General Case

We provide a computable way to verify sufficient conditions for exact sparse recovery in the noise-free case (see also Shtok et. al.\textsuperscript{12}).

**Theorem 2.** Define $s^* = \max\{s : \omega_\diamond(Q, s) > 0\}$. Then,

$$k \leq \lfloor s^*/2 \rfloor \implies \text{exact sparse recovery.}$$

In addition, $s^*$ is the inverse of the maximum of the $n$ optimal values of the following linear programs:

$$\max_z z_i \text{ subject to } Qz = 0, \|z\|_1 \leq 1, \ i = 1, \ldots, n.$$ 

Thus, $\lfloor s^*/2 \rfloor$ is the maximal sparsity level below which exact recovery is guaranteed in the noise-free case.

Verification of $\omega_{\diamond}(Q, s) > 0$: Fourier Case

For the special yet important class of Fourier sensing matrices, the computational cost can be greatly reduced.

**Theorem 3.** If $H$ is the Fourier transform matrix on a finite abelian group, and the rows of $A$ are sampled from the rows of $H$, then the optimal values of

$$\max_{z} z_i \quad \text{subject to } Qz = 0, \quad \|z\|_1 \leq 1$$

are equal for $i = 1, 2, \ldots, n$.

- For these sensing matrices, we compute $s^*$ by solving a single linear program.

- Examples include the **Fourier matrix** and the **Hadamard matrix**, which are widely used in compressive sensing.
### Numerical Examples: Maximal Sparsity Levels

**Table 1:** Comparison of our algorithm for sufficient conditions for exact recovery with Juditsky and Nemirovski’s in bounding the maximal sparsity levels \([s^*/2]\) for Gaussian sensing matrices \((n = 256)\).

<table>
<thead>
<tr>
<th>m</th>
<th>Max Sparsity Level</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>([s^*/2])</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Our Algorithm</td>
<td>JN’s Algorithm</td>
</tr>
<tr>
<td></td>
<td>Our Algorithm</td>
<td>JN’s Algorithm</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>51</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>76</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>102</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>128</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>153</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>179</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>204</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>230</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>

**Observation:** The two algorithms give similar maximal sparsity levels, but ours is much faster.
**Numerical Examples: Maximal Sparsity Levels (cont.)**

**Table 2:** Comparison of sufficient conditions for exact recovery based on $\omega$ with the Mutual Coherence for Hadamard matrices.

<table>
<thead>
<tr>
<th>$m/n$</th>
<th>$n = 2048$</th>
<th>$n = 4096$</th>
<th>$n = 8192$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lfloor s^*/2 \rfloor$</td>
<td>$\lfloor 1/2(1 + 1/\mu) \rfloor$</td>
<td>$\lfloor s^*/2 \rfloor$</td>
</tr>
<tr>
<td>.2</td>
<td>6</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>.3</td>
<td>9</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>.4</td>
<td>12</td>
<td>5</td>
<td>17</td>
</tr>
<tr>
<td>.5</td>
<td>15</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td>.6</td>
<td>19</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>.7</td>
<td>25</td>
<td>9</td>
<td>35</td>
</tr>
<tr>
<td>.8</td>
<td>34</td>
<td>13</td>
<td>48</td>
</tr>
<tr>
<td>.9</td>
<td>53</td>
<td>20</td>
<td>74</td>
</tr>
</tbody>
</table>

**Observation:** Our sufficient condition for exact recovery is stronger than given by the Mutual Coherence for Hadamard matrices.
**Computation of \( \omega_\diamond (Q, s) \): General Case**

We provide a way to compute \( \omega_\diamond (Q, s) \) for any given \( Q \) and \( s \), which readily translates to upper bounds on recovery errors, in the noisy case.

**Theorem 4.** The quantity \( \omega_\diamond (Q, s) \) is the minimum of the optimal values of the following \( n \) linear programs or quadratic programs:

\[
\min_{u \in \mathbb{R}^{n-1}} \| q_i - Q(:, -i)u \|_\diamond \quad \text{s.t.} \quad \| u \|_1 \leq s - 1, \quad i = 1, \ldots, n.
\]

- Here \( q_i \) is the \( i \)th column of \( Q \) and \( Q(:, -i) \) are columns except the \( i \)th one.
- The \( i \)th optimization finds the best approximation of the \( i \)th column using a sparse (measured by \( \ell_1 \) norm) linear combination of the rest columns.
- Thus, if the columns of \( Q \) can well approximate each other using sparse linear combinations, \( \omega_\diamond \) is small and the reconstruction error is large.
- Note, the Mutual Coherence considers the approximability between two columns. Our \( \omega_\diamond \) is more accurate because it considers all columns.
Computation of $\omega_\diamond(Q, s)$: Fourier Case

The computation cost can be greatly reduced for the special yet important class of Fourier sensing matrices, similar to the verification case.

**Theorem 5.** If $H$ is the Fourier transform matrix on a finite abelian group, and the rows of $A$ are sampled from the rows of $H$, then the optimal values of

$$\min_{u \in \mathbb{R}^{n-1}} \|q_i - Q(:, -i)u\|_\diamond \text{ subject to } \|u\|_1 \leq s - 1$$

are equal for all $i = 1, \ldots, n$.

- For these sensing matrices, we compute a single $\omega_\diamond(Q, s)$ by solving a single linear program or quadratic program.

- Examples include the **Fourier matrix** and the **Hadamard matrix**.
**Numerical Examples: Performance Bounds Comparison I**

**Table 3:** $\omega_2(A, s)$ based bounds $\|\hat{x} - x\|_2 \leq \frac{2\sqrt{2k}}{\omega_2(A, 2k)}\varepsilon$ vs. RIC based bounds $\|\hat{x} - x\|_2 \leq \frac{4\sqrt{1+\delta_{2k}}}{1-(1+\sqrt{2})\delta_{2k}}\varepsilon$ for the Basis Pursuit with Bernoulli sensing matrices and $n = 256$ with $\varepsilon = 1$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$m$</th>
<th>Bound on Estimation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>51</td>
</tr>
<tr>
<td>1</td>
<td>$\omega$ bound</td>
<td>4.2</td>
</tr>
<tr>
<td></td>
<td>RIC bound</td>
<td>23.7</td>
</tr>
<tr>
<td>2</td>
<td>$\omega$ bound</td>
<td>31.4</td>
</tr>
<tr>
<td></td>
<td>RIC bound</td>
<td>302.0</td>
</tr>
<tr>
<td>3</td>
<td>$\omega$ bound</td>
<td>52.3</td>
</tr>
<tr>
<td>4</td>
<td>$\omega$ bound</td>
<td>57.0</td>
</tr>
<tr>
<td>5</td>
<td>$\omega$ bound</td>
<td>1256.6</td>
</tr>
<tr>
<td>6</td>
<td>$\omega$ bound</td>
<td>161.6</td>
</tr>
<tr>
<td>7</td>
<td>$\omega$ bound</td>
<td>93.1</td>
</tr>
<tr>
<td>8</td>
<td>$\omega$ bound</td>
<td>258.7</td>
</tr>
</tbody>
</table>
Numerical Examples: Performance Bounds Comparison II

Figure 8: $\|\hat{x} - x\|_2 \leq \frac{2\sqrt{2k}}{\omega_2(A,2k)} \varepsilon$ vs. MC based bound $\|\hat{x} - x\|_2 \leq \frac{2}{1 - \mu(A)(4k-1)} \varepsilon$ for the Basis Pursuit for Hadamard sensing matrices with $n = 2048$. 
Numerical Examples: Observations

- The bounds using $\omega_\diamond$ are tighter than the bounds based on Mutual Coherence or RIC.

- Bounds based on $\omega_\diamond$ still apply even when the bounds based on Mutual Coherence or RIC do not apply, e.g., for small $m$ and large $k$. 
Outline

- Introduction
- Sparsity recovery
- Future work
Open Problems

- Develop computable tight bounds.

- Bounds for other signal structures, such as block-sparse vectors and low-rank matrices.

- Procedures for optimization of system design.

- Computationally efficient algorithms for sensing matrices other than Fourier or Hadamard.

- Minimization of modeling errors due to discrete grid mismatch.
Possible Approaches

• Develop computable tighter bounds that control the average or typical system performance.

• Encode more signal structures into the model rather than using bounds, for example using the continuous-parameter domain.
References


• G. Tang and A. Nehorai, “Computable performance bounds on sparsity recovery,” in revision for *IEEE Trans. on Signal Processing*

Questions ?
Thank You!