Overview of the
Mathematics of Compressive Sensing

Simon Foucart

Reading Seminar on
"Compressive Sensing, Extensions, and Applications"
Texas A&M University
1 October 2015

Part 1: Invitation to Compressive Sensing

Keywords

Keywords

- Sparsity

  Essential

Keywords

- Sparsity
  Essential
- Randomness
  Nothing better so far
  (measurement process)

Keywords

- Sparsity

  Essential

- Randomness

  Nothing better so far
  (measurement process)

- Optimization

  Preferred, but competitive alternatives are available
  (reconstruction process)

The Standard Compressive Sensing Problem

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x}$

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x} = \mathrm{card}\big\{j \in \{1, \dots, N\} : x_j \neq 0\big\}$.

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x} = \mathrm{card}\big\{j \in \{1, \ldots, N\} : x_j \neq 0\big\}$.

Find concrete sensing/recovery protocols,

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x} = \mathrm{card}\big\{j \in \{1, \ldots, N\} : x_j \neq 0\big\}$.

Find concrete sensing/recovery protocols, i.e., find

- measurement matrices $A : \mathbf{x} \in \mathbb{K}^N \mapsto \mathbf{y} \in \mathbb{K}^m$

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x} = \mathrm{card}\{j \in \{1, \ldots, N\} : x_j \neq 0\}$.

Find concrete sensing/recovery protocols, i.e., find

- measurement matrices $A : \mathbf{x} \in \mathbb{K}^N \mapsto \mathbf{y} \in \mathbb{K}^m$
- reconstruction maps $\Delta : \mathbf{y} \in \mathbb{K}^m \mapsto \mathbf{x} \in \mathbb{K}^N$

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x} = \mathrm{card}\big\{j \in \{1, \ldots, N\} : x_j \neq 0\big\}$.

Find concrete sensing/recovery protocols, i.e., find

- measurement matrices $A : \mathbf{x} \in \mathbb{K}^N \mapsto \mathbf{y} \in \mathbb{K}^m$
- reconstruction maps $\Delta : \mathbf{y} \in \mathbb{K}^m \mapsto \mathbf{x} \in \mathbb{K}^N$

such that

$$\Delta(A\mathbf{x}) = \mathbf{x} \qquad \text{for any } s\text{-sparse vector } \mathbf{x} \in \mathbb{K}^N.$$

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x} = \mathrm{card}\big\{j \in \{1, \dots, N\} : x_j \neq 0\big\}$.

Find concrete sensing/recovery protocols, i.e., find

- measurement matrices $A : \mathbf{x} \in \mathbb{K}^N \mapsto \mathbf{y} \in \mathbb{K}^m$
- reconstruction maps $\Delta : \mathbf{y} \in \mathbb{K}^m \mapsto \mathbf{x} \in \mathbb{K}^N$

such that

$$\Delta(A\mathbf{x}) = \mathbf{x} \qquad \text{for any } s\text{-sparse vector } \mathbf{x} \in \mathbb{K}^N.$$

In realistic situations, two issues to consider:

The Standard Compressive Sensing Problem

$\mathbf{x}$: unknown signal of interest in $\mathbb{K}^N$

$\mathbf{y}$: measurement vector in $\mathbb{K}^m$ with $m \ll N$,

s : sparsity of $\mathbf{x} = \mathrm{card}\{j \in \{1, \ldots, N\} : x_j \neq 0\}$.

Find concrete sensing/recovery protocols, i.e., find

- measurement matrices $A : \mathbf{x} \in \mathbb{K}^N \mapsto \mathbf{y} \in \mathbb{K}^m$
- reconstruction maps $\Delta : \mathbf{y} \in \mathbb{K}^m \mapsto \mathbf{x} \in \mathbb{K}^N$

such that

$$\Delta(A\mathbf{x}) = \mathbf{x} \qquad \text{for any } s\text{-sparse vector } \mathbf{x} \in \mathbb{K}^N.$$

In realistic situations, two issues to consider:

Stability: $\mathbf{x}$ not sparse but compressible,

Robustness: measurement error in $\mathbf{y} = A\mathbf{x} + \mathbf{e}$.

A Selection of Applications

A Selection of Applications

- Magnetic resonance imaging

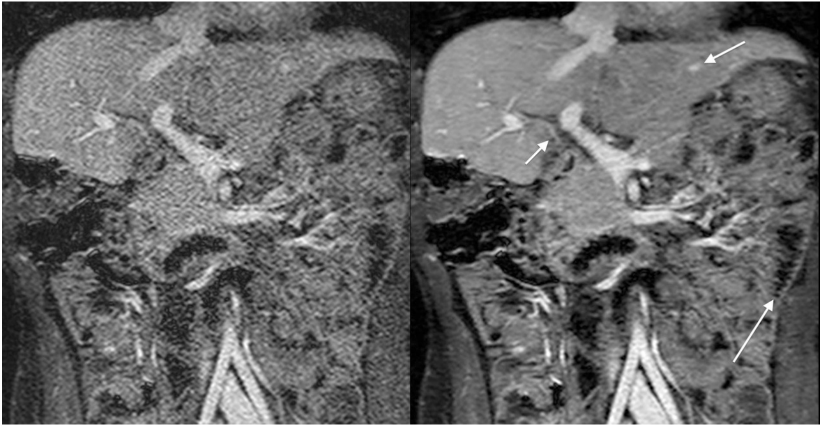

Figure: Left: traditional MRI reconstruction; Right: compressive sensing reconstruction (courtesy of M. Lustig and S. Vasanawala)

A Selection of Applications

- Magnetic resonance imaging
- Sampling theory

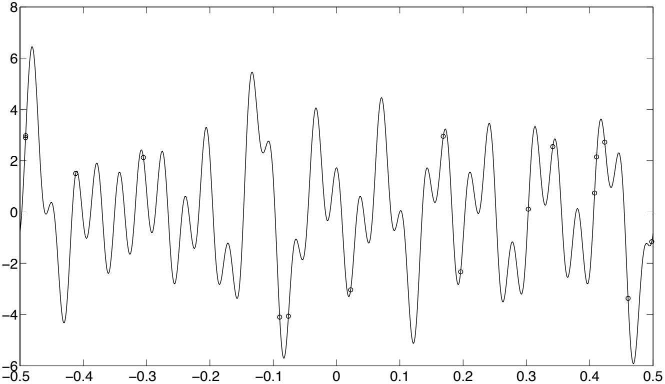

Figure: Time-domain signal with 16 samples.

A Selection of Applications

- Magnetic resonance imaging
- Sampling theory
- Error correction

A Selection of Applications

- Magnetic resonance imaging
- Sampling theory
- Error correction
- and many more...

Application in Metagenomics [Koslicki–F.–Rosen]

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ ($N = 273,727$): concentrations of known bacteria in a given environmental sample.

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ ($N = 273,727$): concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic.

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ ($N = 273,727$): concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic. Note also that $\mathbf{x} \geq \mathbf{0}$ and $\sum_j x_j = 1$.

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ ($N = 273,727$): concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic. Note also that $\mathbf{x} \geq \mathbf{0}$ and $\sum_j x_j = 1$.
- $\mathbf{y} \in \mathbb{R}^m$ ($m = 4^6 = 4,096$): frequencies of length-6 subwords (in 16S rRNA gene reads or in whole-genome shotgun reads)

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ ($N = 273,727$): concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic. Note also that $\mathbf{x} \geq \mathbf{0}$ and $\sum_j x_j = 1$.
- $\mathbf{y} \in \mathbb{R}^m$ ($m = 4^6 = 4,096$): frequencies of length-6 subwords (in 16S rRNA gene reads or in whole-genome shotgun reads)
- $\mathbf{A} \in \mathbb{R}^{m \times N}$: frequencies of length-6 subwords in all known (i.e., sequenced) bacteria.

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ $(N = 273,727)$: concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic. Note also that $\mathbf{x} \geq \mathbf{0}$ and $\sum_j x_j = 1$.
- $\mathbf{y} \in \mathbb{R}^m$ $(m = 4^6 = 4,096)$: frequencies of length-6 subwords (in 16S rRNA gene reads or in whole-genome shotgun reads)
- $\mathbf{A} \in \mathbb{R}^{m \times N}$: frequencies of length-6 subwords in all known (i.e., sequenced) bacteria. It is a frequency matrix, that is,

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ $(N = 273{,}727)$: concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic. Note also that $\mathbf{x} \geq \mathbf{0}$ and $\sum_j x_j = 1$.
- $\mathbf{y} \in \mathbb{R}^m$ $(m = 4^6 = 4{,}096)$: frequencies of length-6 subwords (in 16S rRNA gene reads or in whole-genome shotgun reads)
- $\mathbf{A} \in \mathbb{R}^{m \times N}$: frequencies of length-6 subwords in all known (i.e., sequenced) bacteria. It is a frequency matrix, that is,

$$A_{i,j} \geq 0 \qquad \text{and} \qquad \sum_{i=1}^m A_{i,j} = 1.$$

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ ($N = 273,727$): concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic. Note also that $\mathbf{x} \geq \mathbf{0}$ and $\sum_j x_j = 1$.
- $\mathbf{y} \in \mathbb{R}^m$ ($m = 4^6 = 4,096$): frequencies of length-6 subwords (in 16S rRNA gene reads or in whole-genome shotgun reads)
- $\mathbf{A} \in \mathbb{R}^{m \times N}$: frequencies of length-6 subwords in all known (i.e., sequenced) bacteria. It is a frequency matrix, that is,

$$A_{i,j} \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{m} A_{i,j} = 1.$$

- **Quikr** improves on traditional read-by-read methods, especially in terms of speed.

Application in Metagenomics [Koslicki–F.–Rosen]

- $\mathbf{x} \in \mathbb{R}^N$ ($N = 273,727$): concentrations of known bacteria in a given environmental sample. Sparsity assumption is realistic. Note also that $\mathbf{x} \geq \mathbf{0}$ and $\sum_j x_j = 1$.
- $\mathbf{y} \in \mathbb{R}^m$ ($m = 4^6 = 4,096$): frequencies of length-6 subwords (in 16S rRNA gene reads or in whole-genome shotgun reads)
- $\mathbf{A} \in \mathbb{R}^{m \times N}$: frequencies of length-6 subwords in all known (i.e., sequenced) bacteria. It is a frequency matrix, that is,

$$A_{i,j} \geq 0 \qquad \text{and} \qquad \sum_{i=1}^m A_{i,j} = 1.$$

- **Quikr** improves on traditional read-by-read methods, especially in terms of speed.
- Codes available at
`sourceforge.net/projects/quikr/`
`sourceforge.net/projects/wgsquikr/`

ℓ_0-Minimization

Since

$$\|\mathbf{x}\|_p^p := \sum_{j=1}^{N} |x_j|^p \underset{p\to 0}{\longrightarrow} \sum_{j=1}^{N} \mathbf{1}_{\{x_j \neq 0\}},$$

ℓ_0-Minimization

Since

$$\|\mathbf{x}\|_p^p := \sum_{j=1}^{N} |x_j|^p \underset{p \to 0}{\longrightarrow} \sum_{j=1}^{N} \mathbf{1}_{\{x_j \neq 0\}},$$

the notation $\|\mathbf{x}\|_0$ *[sic]* has become usual for

$$\|\mathbf{x}\|_0 := \mathrm{card}(\mathrm{supp}(\mathbf{x})), \quad \text{where } \mathrm{supp}(\mathbf{x}) := \{j \in [N] : x_j \neq 0\}.$$

ℓ_0-Minimization

Since

$$\|\mathbf{x}\|_p^p := \sum_{j=1}^{N} |x_j|^p \underset{p \to 0}{\longrightarrow} \sum_{j=1}^{N} \mathbf{1}_{\{x_j \neq 0\}},$$

the notation $\|\mathbf{x}\|_0$ *[sic]* has become usual for

$$\|\mathbf{x}\|_0 := \mathrm{card}(\mathrm{supp}(\mathbf{x})), \quad \text{where } \mathrm{supp}(\mathbf{x}) := \{j \in [N] : x_j \neq 0\}.$$

For an s-sparse $\mathbf{x} \in \mathbb{K}^N$, observe the equivalence of

ℓ_0-Minimization

Since

$$\|\mathbf{x}\|_p^p := \sum_{j=1}^{N} |x_j|^p \underset{p \to 0}{\longrightarrow} \sum_{j=1}^{N} \mathbf{1}_{\{x_j \neq 0\}},$$

the notation $\|\mathbf{x}\|_0$ *[sic]* has become usual for

$$\|\mathbf{x}\|_0 := \mathrm{card}(\mathrm{supp}(\mathbf{x})), \quad \text{where } \mathrm{supp}(\mathbf{x}) := \{j \in [N] : x_j \neq 0\}.$$

For an s-sparse $\mathbf{x} \in \mathbb{K}^N$, observe the equivalence of

- $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = \mathbf{y}$ with $\mathbf{y} = A\mathbf{x}$,

ℓ_0-Minimization

Since

$$\|\mathbf{x}\|_p^p := \sum_{j=1}^{N} |x_j|^p \underset{p \to 0}{\longrightarrow} \sum_{j=1}^{N} \mathbf{1}_{\{x_j \neq 0\}},$$

the notation $\|\mathbf{x}\|_0$ *[sic]* has become usual for

$$\|\mathbf{x}\|_0 := \mathrm{card}(\mathrm{supp}(\mathbf{x})), \quad \text{where } \mathrm{supp}(\mathbf{x}) := \{j \in [N] : x_j \neq 0\}.$$

For an s-sparse $\mathbf{x} \in \mathbb{K}^N$, observe the equivalence of

- $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = \mathbf{y}$ with $\mathbf{y} = A\mathbf{x}$,
- $\mathbf{x}$ can be reconstructed as the unique solution of

$$(\mathrm{P}_0) \qquad \underset{\mathbf{z} \in \mathbb{K}^N}{\text{minimize}} \ \|\mathbf{z}\|_0 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

ℓ_0-Minimization

Since

$$\|\mathbf{x}\|_p^p := \sum_{j=1}^{N} |x_j|^p \underset{p \to 0}{\longrightarrow} \sum_{j=1}^{N} \mathbf{1}_{\{x_j \neq 0\}},$$

the notation $\|\mathbf{x}\|_0$ *[sic]* has become usual for

$$\|\mathbf{x}\|_0 := \mathrm{card}(\mathrm{supp}(\mathbf{x})), \quad \text{where } \mathrm{supp}(\mathbf{x}) := \{j \in [N] : x_j \neq 0\}.$$

For an s-sparse $\mathbf{x} \in \mathbb{K}^N$, observe the equivalence of

- $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = \mathbf{y}$ with $\mathbf{y} = A\mathbf{x}$,
- $\mathbf{x}$ can be reconstructed as the unique solution of

$$(\mathrm{P}_0) \qquad \underset{\mathbf{z} \in \mathbb{K}^N}{\text{minimize}} \ \|\mathbf{z}\|_0 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

This is a combinatorial problem, NP-hard in general.

Minimal Number of Measurements

Minimal Number of Measurements

Given $A \in \mathbb{K}^{m \times N}$, the following are equivalent:

Minimal Number of Measurements

Given $A \in \mathbb{K}^{m \times N}$, the following are equivalent:

1. Every s-sparse $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = A\mathbf{x}$,

Minimal Number of Measurements

Given $A \in \mathbb{K}^{m \times N}$, the following are equivalent:

1. Every s-sparse $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = A\mathbf{x}$,
2. $\ker A \cap \{\mathbf{z} \in \mathbb{K}^N : \|\mathbf{z}\|_0 \leq 2s\} = \{\mathbf{0}\}$,

Minimal Number of Measurements

Given $A \in \mathbb{K}^{m \times N}$, the following are equivalent:

1. Every s-sparse $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = A\mathbf{x}$,
2. $\ker A \cap \{\mathbf{z} \in \mathbb{K}^N : \|\mathbf{z}\|_0 \leq 2s\} = \{\mathbf{0}\}$,
3. For any $S \subset [N]$ with $\mathrm{card}(S) \leq 2s$, the matrix A_S is injective,

Minimal Number of Measurements

Given $A \in \mathbb{K}^{m \times N}$, the following are equivalent:

1. Every s-sparse $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = A\mathbf{x}$,
2. $\ker A \cap \{\mathbf{z} \in \mathbb{K}^N : \|\mathbf{z}\|_0 \leq 2s\} = \{\mathbf{0}\}$,
3. For any $S \subset [N]$ with $\mathrm{card}(S) \leq 2s$, the matrix A_S is injective,
4. Every set of $2s$ columns of A is linearly independent.

Minimal Number of Measurements

Given $A \in \mathbb{K}^{m \times N}$, the following are equivalent:

1. Every s-sparse $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = A\mathbf{x}$,
2. $\ker A \cap \{\mathbf{z} \in \mathbb{K}^N : \|\mathbf{z}\|_0 \leq 2s\} = \{\mathbf{0}\}$,
3. For any $S \subset [N]$ with $\mathrm{card}(S) \leq 2s$, the matrix A_S is injective,
4. Every set of $2s$ columns of A is linearly independent.

As a consequence, exact recovery of every s-sparse vector forces

$$m \geq 2s.$$

Minimal Number of Measurements

Given $A \in \mathbb{K}^{m \times N}$, the following are equivalent:

1. Every s-sparse $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = A\mathbf{x}$,
2. $\ker A \cap \{\mathbf{z} \in \mathbb{K}^N : \|\mathbf{z}\|_0 \leq 2s\} = \{\mathbf{0}\}$,
3. For any $S \subset [N]$ with $\mathrm{card}(S) \leq 2s$, the matrix A_S is injective,
4. Every set of $2s$ columns of A is linearly independent.

As a consequence, exact recovery of every s-sparse vector forces

$$m \geq 2s.$$

This can be achieved using partial Vandermonde matrices.

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Identify an s-sparse $\mathbf{x} \in \mathbb{C}^N$ with a function x on $\{0, 1, \ldots, N-1\}$ with support S, $\mathrm{card}(S) = s$.

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Identify an s-sparse $\mathbf{x} \in \mathbb{C}^N$ with a function x on $\{0, 1, \ldots, N-1\}$ with support S, $\mathrm{card}(S) = s$. Consider the $2s$ Fourier coefficients

$$\hat{x}(j) = \sum_{k=0}^{N-1} x(k) e^{-i2\pi jk/N}, \qquad 0 \leq j \leq 2s-1.$$

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Identify an s-sparse $\mathbf{x} \in \mathbb{C}^N$ with a function x on $\{0, 1, \ldots, N-1\}$ with support S, $\mathrm{card}(S) = s$. Consider the $2s$ Fourier coefficients

$$\hat{x}(j) = \sum_{k=0}^{N-1} x(k) e^{-i2\pi jk/N}, \qquad 0 \leq j \leq 2s - 1.$$

Consider a trigonometric polynomial vanishing exactly on S, i.e.,

$$p(t) := \prod_{k \in S} \left(1 - e^{-i2\pi k/N} e^{i2\pi t/N}\right).$$

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Identify an s-sparse $\mathbf{x} \in \mathbb{C}^N$ with a function x on $\{0, 1, \ldots, N-1\}$ with support S, $\mathrm{card}(S) = s$. Consider the $2s$ Fourier coefficients

$$\hat{x}(j) = \sum_{k=0}^{N-1} x(k) e^{-i2\pi jk/N}, \qquad 0 \leq j \leq 2s-1.$$

Consider a trigonometric polynomial vanishing exactly on S, i.e.,

$$p(t) := \prod_{k \in S} \left(1 - e^{-i2\pi k/N} e^{i2\pi t/N}\right).$$

Since $p \cdot x \equiv 0$, discrete convolution gives

$$0 = (\hat{p} * \hat{x})(j) = \sum_{k=0}^{N-1} \hat{p}(k) \hat{x}(j-k), \qquad 0 \leq j \leq N-1.$$

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Identify an s-sparse $\mathbf{x} \in \mathbb{C}^N$ with a function x on $\{0, 1, \ldots, N-1\}$ with support S, $\mathrm{card}(S) = s$. Consider the $2s$ Fourier coefficients

$$\hat{x}(j) = \sum_{k=0}^{N-1} x(k) e^{-i2\pi jk/N}, \qquad 0 \leq j \leq 2s-1.$$

Consider a trigonometric polynomial vanishing exactly on S, i.e.,

$$p(t) := \prod_{k \in S} \left(1 - e^{-i2\pi k/N} e^{i2\pi t/N}\right).$$

Since $p \cdot x \equiv 0$, discrete convolution gives

$$0 = (\hat{p} * \hat{x})(j) = \sum_{k=0}^{N-1} \hat{p}(k) \hat{x}(j-k), \qquad 0 \leq j \leq N-1.$$

Note that $\hat{p}(0) = 1$ and that $\hat{p}(k) = 0$ for $k > s$.

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Identify an s-sparse $\mathbf{x} \in \mathbb{C}^N$ with a function x on $\{0, 1, \ldots, N-1\}$ with support S, $\operatorname{card}(S) = s$. Consider the $2s$ Fourier coefficients

$$\hat{x}(j) = \sum_{k=0}^{N-1} x(k) e^{-i2\pi jk/N}, \qquad 0 \leq j \leq 2s-1.$$

Consider a trigonometric polynomial vanishing exactly on S, i.e.,

$$p(t) := \prod_{k \in S} \left(1 - e^{-i2\pi k/N} e^{i2\pi t/N}\right).$$

Since $p \cdot x \equiv 0$, discrete convolution gives

$$0 = (\hat{p} * \hat{x})(j) = \sum_{k=0}^{N-1} \hat{p}(k) \hat{x}(j-k), \qquad 0 \leq j \leq N-1.$$

Note that $\hat{p}(0) = 1$ and that $\hat{p}(k) = 0$ for $k > s$. The equations $s, \ldots, 2s-1$ translate into a Toeplitz system with unknowns $\hat{p}(1), \ldots, \hat{p}(s)$.

Exact s-Sparse Recovery from $2s$ Fourier Measurements

Identify an s-sparse $\mathbf{x} \in \mathbb{C}^N$ with a function x on $\{0, 1, \ldots, N-1\}$ with support S, $\mathrm{card}(S) = s$. Consider the $2s$ Fourier coefficients

$$\hat{x}(j) = \sum_{k=0}^{N-1} x(k) e^{-i2\pi jk/N}, \qquad 0 \leq j \leq 2s - 1.$$

Consider a trigonometric polynomial vanishing exactly on S, i.e.,

$$p(t) := \prod_{k \in S} \left(1 - e^{-i2\pi k/N} e^{i2\pi t/N}\right).$$

Since $p \cdot x \equiv 0$, discrete convolution gives

$$0 = (\hat{p} * \hat{x})(j) = \sum_{k=0}^{N-1} \hat{p}(k) \hat{x}(j-k), \qquad 0 \leq j \leq N - 1.$$

Note that $\hat{p}(0) = 1$ and that $\hat{p}(k) = 0$ for $k > s$. The equations $s, \ldots, 2s-1$ translate into a Toeplitz system with unknowns $\hat{p}(1), \ldots, \hat{p}(s)$. This determines $\hat{p}$, hence p, then S, and finally $\mathbf{x}$.

ℓ_1-Minimization (Basis Pursuit)

ℓ_1-Minimization (Basis Pursuit)

Replace (P_0) by

$$(P_1) \qquad \underset{\mathbf{z}\in\mathbb{K}^N}{\text{minimize}}\ \|\mathbf{z}\|_1 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

ℓ_1-Minimization (Basis Pursuit)

Replace (P_0) by

$$(P_1) \qquad \underset{\mathbf{z}\in\mathbb{K}^N}{\text{minimize}} \ \|\mathbf{z}\|_1 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

- Geometric intuition

ℓ_1-Minimization (Basis Pursuit)

Replace (P_0) by

$$(P_1) \qquad \underset{\mathbf{z}\in\mathbb{K}^N}{\text{minimize}} \; \|\mathbf{z}\|_1 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

- Geometric intuition
- Unique ℓ_1-minimizers are at most m-sparse (when $\mathbb{K} = \mathbb{R}$)

ℓ_1-Minimization (Basis Pursuit)

Replace (P_0) by

$$(P_1) \qquad \underset{\mathbf{z}\in\mathbb{K}^N}{\text{minimize}}\ \|\mathbf{z}\|_1 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

- Geometric intuition
- Unique ℓ_1-minimizers are at most m-sparse (when $\mathbb{K} = \mathbb{R}$)
- Convex optimization program, hence solvable in practice

ℓ_1-Minimization (Basis Pursuit)

Replace (P_0) by

$$(\mathrm{P}_1) \qquad \underset{\mathbf{z}\in\mathbb{K}^N}{\text{minimize}}\ \|\mathbf{z}\|_1 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

- Geometric intuition
- Unique ℓ_1-minimizers are at most m-sparse (when $\mathbb{K} = \mathbb{R}$)
- Convex optimization program, hence solvable in practice
- In the real setting, recast as the linear optimization program

$$\underset{\mathbf{c},\mathbf{z}\in\mathbb{R}^N}{\text{minimize}}\ \sum_{j=1}^{N} c_j \quad \text{subject to } A\mathbf{z} = \mathbf{y} \text{ and } -c_j \le z_j \le c_j.$$

ℓ_1-Minimization (Basis Pursuit)

Replace (P_0) by

$$(\mathrm{P}_1) \qquad \underset{\mathbf{z}\in\mathbb{K}^N}{\text{minimize}}\ \|\mathbf{z}\|_1 \quad \text{subject to } A\mathbf{z} = \mathbf{y}.$$

- Geometric intuition
- Unique ℓ_1-minimizers are at most m-sparse (when $\mathbb{K} = \mathbb{R}$)
- Convex optimization program, hence solvable in practice
- In the real setting, recast as the linear optimization program

$$\underset{\mathbf{c},\mathbf{z}\in\mathbb{R}^N}{\text{minimize}}\ \sum_{j=1}^{N} c_j \quad \text{subject to } A\mathbf{z} = \mathbf{y} \text{ and } -c_j \leq z_j \leq c_j.$$

- In the complex setting, recast as a second order cone program

Basis Pursuit — Null Space Property

Basis Pursuit — Null Space Property

$\Delta_1(A\mathbf{x}) = \mathbf{x}$ for every vector $\mathbf{x}$ supported on S if and only if

Basis Pursuit — Null Space Property

$\Delta_1(A\mathbf{x}) = \mathbf{x}$ for every vector $\mathbf{x}$ supported on S if and only if

(NSP) $\quad \|\mathbf{u}_S\|_1 < \|\mathbf{u}_{\overline{S}}\|_1, \quad$ all $\mathbf{u} \in \ker A \setminus \{\mathbf{0}\}$.

Basis Pursuit — Null Space Property

$\Delta_1(A\mathbf{x}) = \mathbf{x}$ for every vector $\mathbf{x}$ supported on S if and only if

$$\text{(NSP)} \qquad \|\mathbf{u}_S\|_1 < \|\mathbf{u}_{\overline{S}}\|_1, \qquad \text{all } \mathbf{u} \in \ker A \setminus \{\mathbf{0}\}.$$

For real measurement matrices,

Basis Pursuit — Null Space Property

$\Delta_1(A\mathbf{x}) = \mathbf{x}$ for every vector $\mathbf{x}$ supported on S if and only if

(NSP) $\|\mathbf{u}_S\|_1 < \|\mathbf{u}_{\overline{S}}\|_1,$ all $\mathbf{u} \in \ker A \setminus \{\mathbf{0}\}$.

For real measurement matrices, real and complex NSPs read

Basis Pursuit — Null Space Property

$\Delta_1(A\mathbf{x}) = \mathbf{x}$ for every vector $\mathbf{x}$ supported on S if and only if

(NSP) $\qquad \|\mathbf{u}_S\|_1 < \|\mathbf{u}_{\overline{S}}\|_1, \qquad$ all $\mathbf{u} \in \ker A \setminus \{\mathbf{0}\}$.

For real measurement matrices, real and complex NSPs read

$$\sum_{j \in S} |u_j| < \sum_{\ell \in \overline{S}} |u_\ell|, \qquad \text{all } \mathbf{u} \in \ker_{\mathbb{R}} A \setminus \{0\},$$

Basis Pursuit — Null Space Property

$\Delta_1(A\mathbf{x}) = \mathbf{x}$ for every vector $\mathbf{x}$ supported on S if and only if

$$\text{(NSP)} \qquad \|\mathbf{u}_S\|_1 < \|\mathbf{u}_{\overline{S}}\|_1, \qquad \text{all } \mathbf{u} \in \ker A \setminus \{\mathbf{0}\}.$$

For real measurement matrices, real and complex NSPs read

$$\sum_{j \in S} |u_j| < \sum_{\ell \in \overline{S}} |u_\ell|, \qquad \text{all } \mathbf{u} \in \ker_{\mathbb{R}} A \setminus \{0\},$$

$$\sum_{j \in S} \sqrt{v_j^2 + w_j^2} < \sum_{\ell \in \overline{S}} \sqrt{v_\ell^2 + w_\ell^2}, \qquad \text{all } (\mathbf{v}, \mathbf{w}) \in (\ker_{\mathbb{R}} A)^2 \setminus \{0\}.$$

Basis Pursuit — Null Space Property

$\Delta_1(A\mathbf{x}) = \mathbf{x}$ for every vector $\mathbf{x}$ supported on S if and only if

(NSP) $\quad \|\mathbf{u}_S\|_1 < \|\mathbf{u}_{\overline{S}}\|_1, \quad$ all $\mathbf{u} \in \ker A \setminus \{\mathbf{0}\}$.

For real measurement matrices, real and complex NSPs read

$$\sum_{j \in S} |u_j| < \sum_{\ell \in \overline{S}} |u_\ell|, \qquad \text{all } \mathbf{u} \in \ker_{\mathbb{R}} A \setminus \{0\},$$

$$\sum_{j \in S} \sqrt{v_j^2 + w_j^2} < \sum_{\ell \in \overline{S}} \sqrt{v_\ell^2 + w_\ell^2}, \qquad \text{all } (\mathbf{v}, \mathbf{w}) \in (\ker_{\mathbb{R}} A)^2 \setminus \{0\}.$$

Real and complex NSPs are in fact equivalent.

Orthogonal Matching Pursuit

Orthogonal Matching Pursuit

Starting with $S^0 = \emptyset$ and $\mathbf{x}^0 = \mathbf{0}$, iterate

$$(\mathrm{OMP}_1) \qquad S^{n+1} = S^n \cup \left\{ j^{n+1} := \underset{j \in [N]}{\operatorname{argmax}} \left\{ |(A^*(\mathbf{y} - A\mathbf{x}^n))_j| \right\} \right\},$$

$$(\mathrm{OMP}_2) \qquad \mathbf{x}^{n+1} = \underset{\mathbf{z} \in \mathbb{C}^N}{\operatorname{argmin}} \left\{ \|\mathbf{y} - A\mathbf{z}\|_2, \operatorname{supp}(\mathbf{z}) \subseteq S^{n+1} \right\}.$$

Orthogonal Matching Pursuit

Starting with $S^0 = \emptyset$ and $\mathbf{x}^0 = \mathbf{0}$, iterate

$$(\mathrm{OMP}_1) \qquad S^{n+1} = S^n \cup \left\{ j^{n+1} := \underset{j \in [N]}{\mathrm{argmax}} \left\{ |(A^*(\mathbf{y} - A\mathbf{x}^n))_j| \right\} \right\},$$

$$(\mathrm{OMP}_2) \qquad \mathbf{x}^{n+1} = \underset{\mathbf{z} \in \mathbb{C}^N}{\mathrm{argmin}} \left\{ \|\mathbf{y} - A\mathbf{z}\|_2, \mathrm{supp}(\mathbf{z}) \subseteq S^{n+1} \right\}.$$

- The norm of the residual decreases according to

$$\|\mathbf{y} - A\mathbf{x}^{n+1}\|_2^2 \leq \|\mathbf{y} - A\mathbf{x}^n\|_2^2 - \left| (A^*(\mathbf{y} - A\mathbf{x}^n))_{j^{n+1}} \right|^2.$$

Orthogonal Matching Pursuit

Starting with $S^0 = \emptyset$ and $\mathbf{x}^0 = \mathbf{0}$, iterate

$$(\mathrm{OMP}_1) \qquad S^{n+1} = S^n \cup \left\{ j^{n+1} := \underset{j \in [N]}{\operatorname{argmax}} \left\{ \left| (A^*(\mathbf{y} - A\mathbf{x}^n))_j \right| \right\} \right\},$$

$$(\mathrm{OMP}_2) \qquad \mathbf{x}^{n+1} = \underset{\mathbf{z} \in \mathbb{C}^N}{\operatorname{argmin}} \left\{ \|\mathbf{y} - A\mathbf{z}\|_2, \operatorname{supp}(\mathbf{z}) \subseteq S^{n+1} \right\}.$$

- The norm of the residual decreases according to

$$\|\mathbf{y} - A\mathbf{x}^{n+1}\|_2^2 \leq \|\mathbf{y} - A\mathbf{x}^n\|_2^2 - \left| (A^*(\mathbf{y} - A\mathbf{x}^n))_{j^{n+1}} \right|^2.$$

- Every vector $\mathbf{x}_{\neq \mathbf{0}}$ supported on S, $\operatorname{card}(S) = s$, is recovered from $\mathbf{y} = A\mathbf{x}$ after at most s iterations of OMP if and only if A_S is injective and

$$(\mathrm{ERC}) \qquad \max_{j \in S} |(A^*\mathbf{r})_j| > \max_{\ell \in \overline{S}} |(A^*\mathbf{r})_\ell|$$

for all $\mathbf{r}_{\neq \mathbf{0}} \in \left\{ A\mathbf{z}, \operatorname{supp}(\mathbf{z}) \subseteq S \right\}$.

Iterative Hard Thresholding and Hard Thresholding Pursuit

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^*A\mathbf{x} = A^*\mathbf{y}$,

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^*A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^* A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,
- at each iteration, keep s largest absolute entries and set the other ones to zero.

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^*A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,
- at each iteration, keep s largest absolute entries and set the other ones to zero.

IHT: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$\text{(IHT)} \qquad \mathbf{x}^{n+1} = H_s(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))$$

until a stopping criterion is met.

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^* A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,
- at each iteration, keep s largest absolute entries and set the other ones to zero.

IHT: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$\text{(IHT)} \qquad \mathbf{x}^{n+1} = H_s(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))$$

until a stopping criterion is met.

HTP: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^* A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,
- at each iteration, keep s largest absolute entries and set the other ones to zero.

IHT: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

(IHT) $$\mathbf{x}^{n+1} = H_s(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))$$

until a stopping criterion is met.

HTP: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

(HTP_1)

(HTP_2)

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^*A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,
- at each iteration, keep s largest absolute entries and set the other ones to zero.

IHT: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$\text{(IHT)} \qquad \mathbf{x}^{n+1} = H_s(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))$$

until a stopping criterion is met.

HTP: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$(\mathrm{HTP}_1) \qquad S^{n+1} = \big\{ s \text{ largest abs. entries of } \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n) \big\},$$

(HTP_2)

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^* A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,
- at each iteration, keep s largest absolute entries and set the other ones to zero.

IHT: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$\text{(IHT)} \qquad \mathbf{x}^{n+1} = H_s(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))$$

until a stopping criterion is met.

HTP: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$\text{(HTP}_1\text{)} \qquad S^{n+1} = \{s \text{ largest abs. entries of } \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)\},$$

$$\text{(HTP}_2\text{)} \qquad \mathbf{x}^{n+1} = \operatorname{argmin}\{\|\mathbf{y} - A\mathbf{z}\|_2, \operatorname{supp}(\mathbf{z}) \subseteq S^{n+1}\},$$

Iterative Hard Thresholding and Hard Thresholding Pursuit

- solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^*A\mathbf{x} = A^*\mathbf{y}$,
- classical iterative methods suggest the iteration $\mathbf{x}^{n+1} = \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)$,
- at each iteration, keep s largest absolute entries and set the other ones to zero.

IHT: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$(\mathrm{IHT}) \qquad \mathbf{x}^{n+1} = H_s(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))$$

until a stopping criterion is met.

HTP: Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$ and iterate:

$$(\mathrm{HTP}_1) \quad S^{n+1} = \big\{ s \text{ largest abs. entries of } \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n) \big\},$$

$$(\mathrm{HTP}_2) \quad \mathbf{x}^{n+1} = \mathrm{argmin}\big\{ \|\mathbf{y} - A\mathbf{z}\|_2, \mathrm{supp}(\mathbf{z}) \subseteq S^{n+1} \big\},$$

until a stopping criterion is met ($S^{n+1} = S^n$ is natural here).