Compressive Sensing of Streams of Pulses

Chinmay Hegde and Richard Baraniuk

Rice University
Concise Signal Model: Sparsity

- **Sparse** signal:

  
  - only $K$ out of $N$ coordinates nonzero
Concise Signal Model: Sparsity

- **Sparse** signal:
  - only $K$ out of $N$ coordinates nonzero

- Geometry: *union* of $K$-dimensional subspaces aligned w/ coordinate axes
Compressive Sensing

• **Sampling** via dimensionality reduction

\[
\begin{align*}
M \times 1 & \quad \text{Measurements} \\
M \times N & \\
N \times 1 & \quad \text{spare signal} \\
\end{align*}
\]
Restricted Isometry Property (RIP)

- Preserve the structure of sparse signals
Compressive Sensing

\[ y = \Phi x \]

- **Random** subgaussian matrix \( \Phi \) has the **RIP** whp if

\[
M = O(K + \log \left( \frac{N}{K} \right))
\]

- \( M \times 1 \) random measurements
- \( \Phi \) is a \( M \times N \) matrix
- \( N \times 1 \) sparse signal
- \( K \) nonzero entries
Stable Recovery

- Efficient, stable algorithms that give back signal
Compressive Sensing

- \( \ell_1 \)-optimization
  - \([C, R, T]; [D]; [F,W,N]; [H,Y,Z]\)

- Greedy algorithms
  - OMP \([G, T]\)
  - iterated thresholding \([N, F]; [D, D, DeM]; [B, D]\)
  - CoSaMP \([N,T]\); Subspace Pursuit \([D,M]\)
Beyond Sparse Models
Beyond Sparse Models

- Sparsity captures *simplistic primary structure*

5% sparse image
Beyond Sparse Models

- Most real-world apps exhibit *additional structure*

5% sparse image
Model-based CS

- **K-sparse structured sparsity model** comprises a reduced set of $K$-dim canonical subspaces

- **Model-RIP:** stable embedding $[B, D]; [B,D,DeV,W]$

  \[ M = O(K + \log(m_K)) \]

- **Recovery:** simple modification of iterative support selection algorithms
Model-based CS

• *K*-sparse structured sparsity model comprises a reduced set of *K*-dim canonical subspaces

• **Model-RIP:** stable embedding
  
  $[B, D]; [B, D, DeV, W]$

  $M = O(K + \log(m_K))$

• **Recovery:** simple modification of iterative support selection algorithms

• **Models are good!!**
When things aren’t exactly sparse..
When things aren’t exactly sparse..

- $S$-sparse signal $\textit{convolved}$ with an $F$-sparse impulse response
When things aren’t exactly sparse.

\[ z = Hx = Xh \]
Streams of pulses

- Neuronal spike trains
- UWB signals
- Astronomical imaging
- Etc.
Streams of pulses

- Overall sparsity: $K = SF$

- Model-based CS: $M = O(K + \log(m_K))$

- BUT #$\{\text{degrees of freedom}\} = O(S + F)$
Streams of pulses

- Overall sparsity: \[ K = SF \]

- Model-based CS: \[ M = O(K + \log(m_K)) \]

- BUT \#\{degrees of freedom\} = \[ O(S + F) \]

- Can we do better?
Geometry of signal set

- *Infinite* union of subspaces
Geometry of signal set

- *Infinite* union of subspaces

\[ \mathbb{R}^N \]

- Very small subset of the set of all SF-sparse signals
Sampling Bound

- **RIP for pulse streams**

\[
M = O(S + F + S \log(N/S))
\]
Recovery

- Similar in spirit to **blind deconvolution**

- Slightly different goal: recover the *pulse stream*

- Iterate between estimating spikes and filter coefficients
 Recovery algorithm

Algorithm 1

Inputs: Projection matrix Φ, measurements y, model parameters Δ, S, F.
Output: $\mathcal{M}_{S,F}$-sparse approximation $\hat{z}$ to true signal z

$\hat{x} = 0$, $\hat{h} = (1^T_F, 0, \ldots, 0)$; $i = 0$ {initialize}

while halting criterion false do

1. $i \leftarrow i + 1$
2. $\hat{z} \leftarrow \hat{x} \ast \hat{h}$ {current signal estimate}
3. $\hat{H} = \mathcal{C}(\hat{h}), \Phi_h = \Phi \hat{H}$ {form dictionary for spike domain}
4. $e \leftarrow \Phi^T_h (y - \Phi_h \hat{z})$ {residual}
5. $\Omega \leftarrow \text{supp}(\mathbb{D}_2(e))$ {prune residual according to (2S, 2Δ) model}
6. $T \leftarrow \Omega \cup \text{supp}(\hat{x}_{i-1})$ {merge supports}
7. $b|_T \leftarrow (\Phi_h)^T_T y$, $b|_{T^c} = 0$ {update spike estimate}
8. $\hat{x} \leftarrow \mathbb{D}(b)$ {prune spike estimate according to (S, Δ) model}
9. $\hat{X} = \mathcal{C}(\hat{x})$, $\Phi_x = \Phi \hat{X}$ {form dictionary for filter domain}
10. $\hat{h} \leftarrow \Phi^T_{xy} y$ {update filter estimate}

end while

return $\hat{z} \leftarrow \hat{x} \ast \hat{h}$
Numerical example

Stream of pulses: $N = 1024, S = 8, F = 11$
Numerical example

CS recovery using $M = 90$ measurements
Numerical example

CS recovery using $M = 90$ measurements
Monte Carlo Simulation
Real-data experiment
Real-data experiment

CS recovery using the pulse-stream model
Real data experiment
Extension to 2D

- Original image
  \( (N = 4096, K = 175) \)

- CoSaMP (\( M = 290 \))
  \( \text{MSE} = 16.95 \)

- Pulse-field (\( M = 290 \))
  \( \text{MSE} = 0.07 \)
Summary

• Why CS works: stable embedding for signals with concise geometric structure

• Contribution: a CS framework for pulse streams

Advantages: provably fewer measurements
simple, flexible algorithm

www.dsp.rice.edu/cs
• Blank slide
• Another blank slide
Yet another blank slide