Greedy Algorithms for Joint Sparse Recovery

Jeff Blanchard

with Mike Davies, Michael Cermak, David Hanle, Yirong Jing

Grinnell College, Iowa, USA
University of Edinburgh, UK

Funding provided by NSF OISE 0854991 and NSF DMS 1112612.

CIPMA 2013, Mar del Plata, August 15, 2013

The CS Problem: single measurement vector

Measure and recover a k-sparse vector with an $m \times n$ matrix:

- The problem is characterized by three parameters: $k < m < n$
  - n, the signal length;
  - m, number of inner product measurements;
  - k, the sparsity of the signal.
- The measurement matrix A is of size $m \times n$.
- The target vector $x \in \mathbb{R}^n$ is k-sparse, $\|x\|_0 = k$.
- The measurements $y \in \mathbb{R}^m$ where $y = Ax$.

  (Highly Underdetermined)

The CS Problem: single measurement vector

Measure and recover a k-sparse vector with an $m \times n$ matrix:

- The problem is characterized by three parameters: $k < m < n$
  - n, the signal length;
  - m, number of inner product measurements;
  - k, the sparsity of the signal.
- The measurement matrix A is of size $m \times n$.
- The target vector $x \in \mathbb{R}^n$ is k-sparse, $\|x\|_0 = k$.
- The measurements $y \in \mathbb{R}^m$ where $y = Ax$.
  (Highly Underdetermined)

The CS Problem: single measurement vector

Measure and recover a k-sparse vector with an $m \times n$ matrix:

- The problem is characterized by three parameters: $k < m < n$
  - n, the signal length;
  - m, number of inner product measurements;
  - k, the sparsity of the signal.

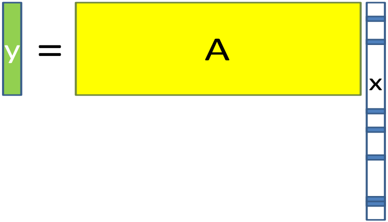
$$y = Ax$$

The CS Problem: single measurement vector

Measure and recover a k-sparse vector with an $m \times n$ matrix:

- The problem is characterized by three parameters: $k \leq m \leq n$
  - n, the signal length;
  - m, number of inner product measurements;
  - k, the sparsity of the signal.

$$y = Ax$$



The CS Problem: multiple measurement vectors

Measure and recover r *jointly* k*-sparse* vectors with a single $m \times n$ measurement matrix.

- A single measurement matrix A of size $m \times n$.
- The set of r target vectors $\{x_1, \ldots, x_r\} \subset \mathbb{R}^n$ which are *jointly* k*-sparse*.
- The measurements $\{y_1, \ldots, y_r\} \subset \mathbb{R}^m$ where $y_i = Ax_i$. (Still Highly Underdetermined)

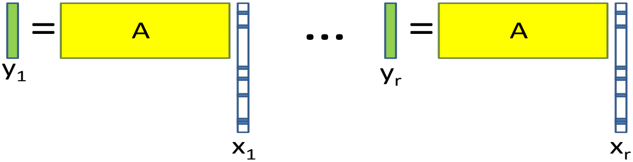
$$y_1 = Ax_1, \;\; \ldots \;\; , y_r = Ax_r$$

The CS Problem: multiple measurement vectors

Measure and recover r *jointly* k*-sparse* vectors with a single $m \times n$ measurement matrix.

- A single measurement matrix A of size $m \times n$.
- The set of r target vectors $\{x_1, \ldots, x_r\} \subset \mathbb{R}^n$ which are *jointly* k*-sparse*.
- The measurements $\{y_1, \ldots, y_r\} \subset \mathbb{R}^m$ where $y_i = Ax_i$. (Still Highly Underdetermined)

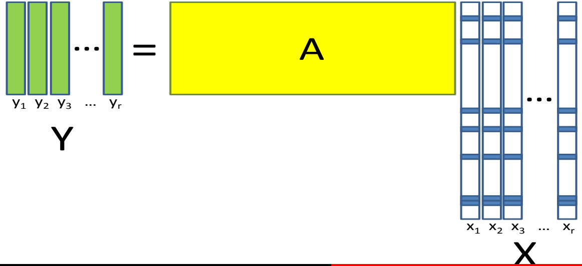
$$y_1 = Ax_1, \ \ldots \ , y_r = Ax_r$$



The CS Problem: multiple measurement vectors

Measure and recover a $n \times r$ *k-row-sparse* matrix with a $m \times n$ measurement matrix.

- The measurement matrix A is of size $m \times n$.
- The matrix of r target vectors $X = [x_1|\cdots|x_r] \in \mathbb{R}^{n \times r}$ is k*-row-sparse*.
- The measurements $Y = [y_1|\cdots|y_r] \in \mathbb{R}^{m \times r}$ where $Y = AX$. (Still Highly Underdetermined)



The MMV Problem: incomplete history

A highly unfair, incomplete (compressive) sampling of results:

- Tropp, Gilbert, Strauss: Simultaneous Orthogonal Matching Pursuit and ℓ_1-minimization, 2006.
- Foucart: Hard Thresholding Pursuit for MMV problems, 2011.
- Davies, Eldar: Rank Aware Algorithms, 2012.
- Many others: primarily focused on relaxations, rank-blind variants of OMP, mixed matrix norm techniques.

The MMV Problem: this presentation

- A rank-aware recovery guarantee.
- Extension of SMV greedy algorithms to the MMV problem.
- Empirical performance comparison.
- Totally unrelated plug for something else.

Simultaneous OMP

SOMP [Tropp, Gilbert, Strauss]

Initialization: $X^0 = 0$, $T^0 = \emptyset$, $R^0 = Y$,

for $j = 1$**;** $j = j + 1$**; do**

1. *Max Correlation*: $i^j = \arg\max_i \|A_i^* R^{j-1}\|_2$
2. *New Support*: $T^j = T^{j-1} \cup i^j$
3. *Update Approximation*: $X^j = A_{T^j}^{\dagger} Y$
4. *Update Residual*: $R^j = Y - AX^j$

Output: $\hat{X} = X^{j^\star}$ where $j^\star$ is the final completed iteration.

Rank Aware Simultaneous OMP

RA-SOMP [Davies, Eldar]

Initialization: $X^0 = 0$, $T^0 = \emptyset$, $R^0 = Y$,

for $j = 1$; $j = j + 1$; **do**

1. *Rank Awareness*: compute $U^{j-1} = ortho(R^{j-1})$
2. *Max Correlation*: $i^j = \arg\max_i \|A_i^* U^{j-1}\|_2$
3. *New Support*: $T^j = T^{j-1} \cup i^j$
4. *Update Approximation*: $X^j = A_{T^j}^{\dagger} Y$
5. *Update Residual*: $R^j = Y - AX^j$

Output: $\hat{X} = X^{j^\star}$ where $j^\star$ is the final completed iteration.

Preserving Rank Awareness

RA-SOMP suffers from *rank degeneration* of the residual.

- Two solutions:
  - RA-Order Recursive MP [Davies/Eldar]

    Max Correlation: $i^j = \arg\max_i \|A_i^* U^{j-1}\|_2 / \|P_{T^{j-1}}^{\perp} A_i\|_2.$

  - RA-SOMP + MUSIC [B./Davies & Lee/Bresler/Junge]

    Apply RA-SOMP for $k - r$ iterations, then apply MUSIC.

MMV Recovery Guarantees

Typical worst case MMV recovery guarantees reduce to the SMV case.

- Worst case MMV problem: rank$(X) = 1$

$$x = x_1 = x_2 = \cdots = x_r \quad \text{so that} \quad X = [x|x|\cdots|x]$$

- For A from the Gaussian ensemble (entries drawn iid from $\mathcal{N}(0, m^{-1})$), SOMP recovers X from Y with high probability provided

$$m \gtrsim Ck\left(\log(n) + 1\right).$$

Rank Aware Recovery Guarantees

Rank aware algorithms incorporate rank in the analysis:

- For rank aware algorithms, the rank reduces the logarithmic penalty:

Theorem (B.,Davies 2012)

Suppose $X \in \mathbb{R}^{n \times r}$, $T = rowsupp(X)$ with $|T| = k$, $rank(X) = r < k$, and $X_{(T)}$ is in general position. If A is drawn from the Gaussian ensemble (independently from X), then both RA-SOMP+MUSIC and RA-ORMP recover X from Y with high probability provided

$$m \gtrsim Ck\left(\frac{log(n)}{r} + 1\right).$$

- When $r \sim \log(n)$, the number of required measurements is linearly proportional to the row-sparsity of X.

Rank Aware Recovery Guarantees

Rank aware algorithms incorporate rank in the analysis:

- For rank aware algorithms, the rank reduces the logarithmic penalty:

Theorem (B.,Davies 2012)

Suppose $X \in \mathbb{R}^{n\times r}$, $T = \mathit{rowsupp}(X)$ with $|T| = k$, $\mathit{rank}(X) = r < k$, and $X_{(T)}$ is in general position. If A is drawn from the Gaussian ensemble (independently from X), then both RA-SOMP+MUSIC and RA-ORMP recover X from Y with high probability provided

$$m \gtrsim Ck\left(\frac{\log(n)}{r} + 1\right).$$

- When $r \sim \log(n)$, the number of required measurements is linearly proportional to the row-sparsity of X.

Rank Aware Recovery Guarantees

Rank aware algorithms incorporate rank in the analysis:

- For rank aware algorithms, the rank reduces the logarithmic penalty:

Theorem (B.,Davies 2012)

Suppose $X \in \mathbb{R}^{n\times r}$, $T =$ *rowsupp*(X) *with* $|T| = k$, *rank*$(X) = r < k$, *and* $X_{(T)}$ *is in general position. If* A *is drawn from the Gaussian ensemble (independently from* X*), then both RA-SOMP+MUSIC and RA-ORMP recover* X *from* Y *with high probability provided*

$$m \gtrsim Ck\left(\frac{log(n)}{r} + 1\right).$$

- When $r \sim \log(n)$, the number of required measurements is linearly proportional to the row-sparsity of X.

Greedy SMV Algorithms for MMV Problems

Following Tropp et al. & Foucart, we extend SMV algorithms to the MMV setting.

- Tropp et al. described the extension to the MMV setting as "capitalization".
- Foucart extended Hard Thresholding Pursuit (HTP) to MMV problems.
- We extended and analyzed five greedy SMV algorithms to the MMV setting (with Cermak, Hanle, Jing).
  - Iterative Hard Thresholding (IHT) [Blumensath & Davies]
  - Normalized IHT (NIHT) [Blumensath & Davies]
  - HTP and Normalized HTP (NHTP) [Foucart]
  - Compressive Sampling Matching Pursuit (CoSaMP) [Needell & Tropp]

Simultaneous Normalized Iterative Hard Thresholding

SNIHT [Blumensath/Davies & B./Cermak/Hanle/Jing]

Initialization: $X^0 = 0$, $R^0 = Y$,

$$T^0 = \{k \text{ indices for largest row } \ell_2 \text{ norms of } A^* R^0\}$$

for $j = 1$; $j = j + 1$; **do**

1. *Step Size*: compute the steepest descent step on T^{j-1}
$$w^j = \frac{\left\| \left(A^* R^{j-1}\right)_{(T^{j-1})} \right\|_F}{\left\| A_{T^{j-1}} \left(A^* R^{j-1}\right)_{(T^{j-1})} \right\|_F}$$
2. *Update Approximation*: $X^j = X^{j-1} + w^j \left(A^* R^{j-1}\right)$
3. *Support Identification*:
$$T^j = \{k \text{ indices for largest row } \ell_2 \text{ norms of } X^j\}$$
4. *Threshold*: $X^j = X^j_{(T^j)}$
5. *Update Residual*: $R^j = Y - AX^j$

Output: $\hat{X} = X^{j^\star}$ where $j^\star$ is the final completed iteration.

Restricted Isometry Property

Definition (Asymmetric RIP Constants)

For the matrix $Z \in \mathbb{R}^{m \times n}$, the *asymmetric restricted isometry constants* L_k and U_k are the smallest values such that

$$(1 - L_k)\|x\|_2 \leq \|Ax\|_2 \leq (1 + U_k)\|x\|_2$$

for all k-sparse vectors x.

Let $\mu^{alg}(k; A)$ be a function of the asymmetric restricted isometry constants of A. We find sufficient restricted isometry conditions in the form of $\mu^{alg}(k; A) < 1$ that guarantee the algorithm alg will recover X from Y.

These results are not rank aware. The algorithms are not explicitly rank aware.

Restricted Isometry Property

Definition (Asymmetric RIP Constants)

For the matrix $Z \in \mathbb{R}^{m \times n}$, the *asymmetric restricted isometry constants* L_k and U_k are the smallest values such that

$$(1 - L_k)\|x\|_2 \leq \|Ax\|_2 \leq (1 + U_k)\|x\|_2$$

for all k-sparse vectors x.

Let $\mu^{alg}(k; A)$ be a function of the asymmetric restricted isometry constants of A. We find sufficient restricted isometry conditions in the form of $\mu^{alg}(k; A) < 1$ that guarantee the algorithm alg will recover X from Y.

These results are not rank aware. The algorithms are not explicitly rank aware.

RIP Recovery Guarantees

Theorem (B.,Cermak,Hanle,Jing)

Let $A \in \mathbb{R}^{m\times n}$, $X \in \mathbb{R}^{n\times r}$ with T the index set of the rows of X with the k largest row-ℓ_2-norms. Let $Y = AX + E$ for some error matrix $E \in \mathbb{R}^{m\times r}$. For each algorithm alg from SIHT, SNIHT, SHTP, SNHTP, and SCoSaMP, there exists asymmetric restricted isometry functions $\mu^{alg} \equiv \mu^{alg}(k;A)$ and $\xi^{alg} \equiv \xi^{alg}(k;A)$ guaranteeing that after iteration j,

$$\|X^j - X_{(T)}\|_F \leq (\mu^{alg})^j \|X\|_F + \frac{\xi^{alg}}{1-\mu^{alg}} \|AX_{(T^c)} + E\|_F.$$

Therefore, when $\mu^{alg} < 1$, the error is proportional to the measurements on the non-optimal support plus noise.

If $T = \text{rowsupp}(X)$ and $E = 0$, the algorithm converges to the k-row-sparse matrix X provided $\mu^{alg}(k;A) < 1$.

Phase Transitions

We present empirical performance comparisons in the form of weak recovery phase transitions.

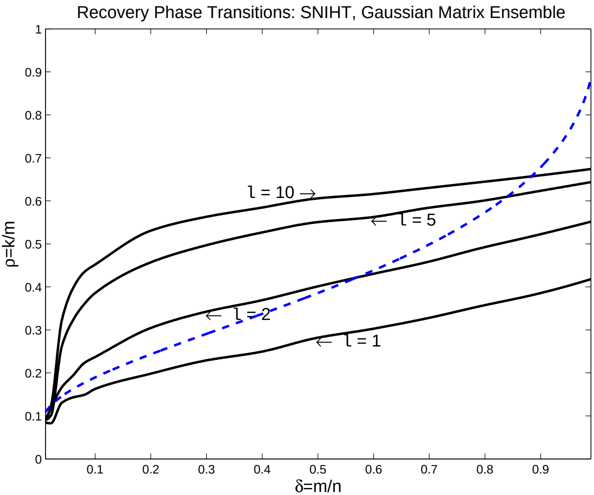
- The phase space is the unit square $[0,1]^2$ defined by two parameters:

$$\delta = \frac{m}{n} \qquad \text{(undersampling ratio)}$$

$$\rho = \frac{k}{m} \qquad \text{(oversampling ratio)}$$

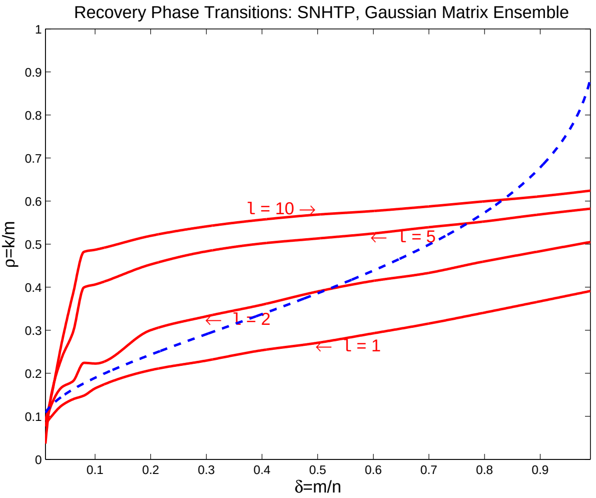
- The tests are conducted in Matlab with $n = 1024$.
- The matrix A is drawn randomly from the Gaussian ensemble.
- The row support is chosen uniformly.
- The entries of the rows are drawn from $\{-1, 1\}$ with equal probability.
- The empirical weak recovery phase transition is the location 50% successful recovery.

SNIHT: $l = 1, 2, 5, 10$ and $n = 1024$



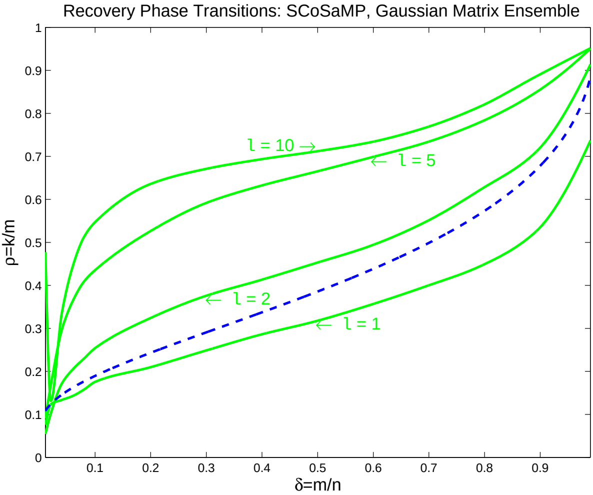
Blue dashed overlay is the theoretical weak phase transition for ℓ_1-minimization.

SNHTP: $l = 1, 2, 5, 10$ and $n = 1024$



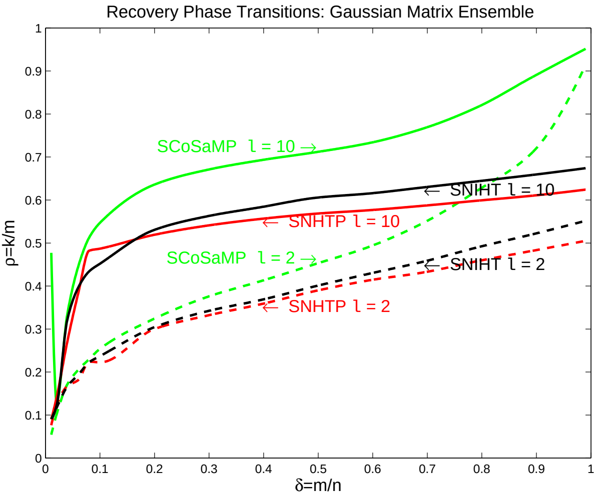
Blue dashed overlay is the theoretical weak phase transition for ℓ_1-minimization.

SCoSaMP: $l = 1, 2, 5, 10$ and $n = 1024$



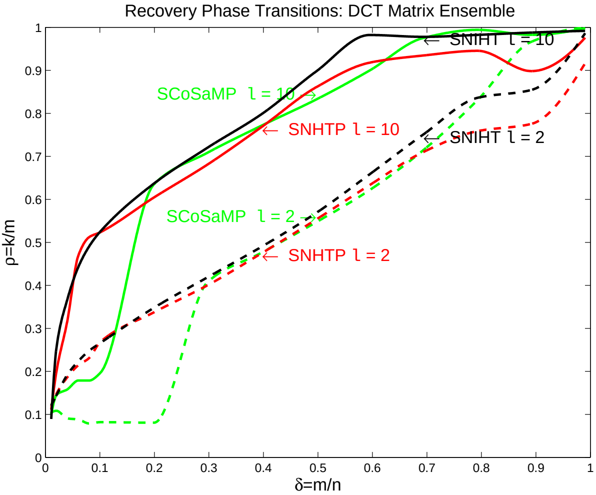
Blue dashed overlay is the theoretical weak phase transition for ℓ_1-minimization.

All: $l = 2, 10$, $n = 1024$ and A Gaussian



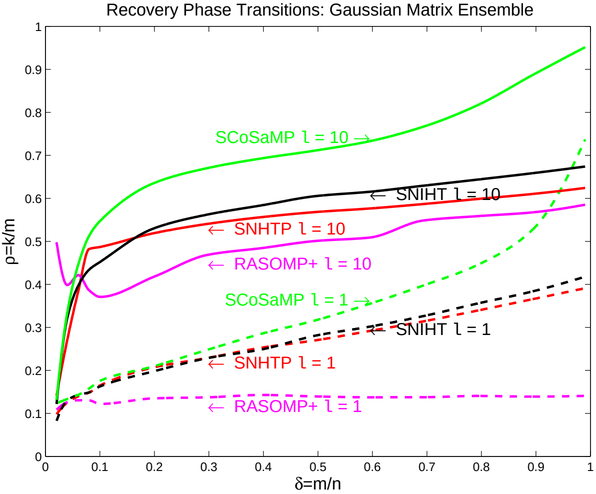
A is drawn from the Gaussian ensemble.

All: $l = 2, 10$, $n = 1024$ and A subsampled DCT



A is a randomly subsampled DCT matrix.

Rank Aware?: $l = 1, 10$, $n = 1024$ and A Gaussian



The *rank-blind* greedy algorithms outperform the *rank aware* algorithm.

Summary

- Rank aware recovery: $m \gtrsim Ck\left(\frac{\log(n)}{r}+1\right)$.
- Sufficient RIP guarantees for extending well-known SMV algorithms to the MMV setting.
- Low complexity, but sophisticated simultaneous greedy algorithms appear to be rank aware.

[1.] *Recovery Guarantees for Rank Aware Pursuits*, with M. Davies, IEEE Signal Processing Letters 19(7):427–430, 2012.

[2.] *Greedy Algorithms for Joint Sparse Recovery*, with M. Cermak, D. Hanle, Y. Jing, submitted, 2013.

[3.] Preprints available: `www.math.grinnell.edu/~blanchaj/Research.html`

Questions?

GAGA: GPU Accelerated Greedy Algorithms
for Compressed Sensing
with Jared Tanner (Oxford)
www.gaga4cs.org

- Fast GPU implementations of greedy algorithms executed from Matlab.
- Solve problems up to 2^{20} in fractions of a second.
- Robust testing suite.
- Freely available for research.
- Extension to matrix completion in progress.
- Requires CUDA capable NVIDIA GPU.
- Does NOT require parallel processing toolbox.