CoSaMP: Greedy Signal Recovery and Uniform Uncertainty Principles

SIAM Student Research Conference

Deanna Needell

Joint work with Roman Vershynin and Joel Tropp

UC Davis, May 2008

Problem Setup

-
$$\begin{bmatrix} & & \\ & \Phi & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} v \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} x \\ \\ \\ \\ \end{bmatrix}$$

Problem Setup

- $$\begin{bmatrix} & & \\ & \Phi & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} v \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} x \\ \\ \\ \\ \end{bmatrix}$$
- If we know the $d \times d$ matrix Φ and the measurement vector x, can we recover every v?

Problem Setup

- $$\begin{bmatrix} & & \\ & \Phi & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} v \\ \\ \\ \end{bmatrix} = \begin{bmatrix} x \\ \\ \\ \end{bmatrix}$$
- If we know the $d \times d$ matrix Φ and the measurement vector x, can we recover every v?
- What kind of matrix Φ? How would we recover?

Problem Setup

-

$$\begin{bmatrix} & \Phi & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} v \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} x \\ \\ \end{bmatrix}$$

Problem Setup

- $$\begin{bmatrix} & \Phi & \end{bmatrix} \begin{bmatrix} v \end{bmatrix} = \begin{bmatrix} x \end{bmatrix}$$
- If we know the $m \times d$ matrix Φ $(m \ll d)$ and the measurement vector x, can we recover every v?

Problem Setup

- $$\begin{bmatrix} & & \\ & \Phi & \\ & & \end{bmatrix} \begin{bmatrix} \\ v \\ \\ \end{bmatrix} = \begin{bmatrix} x \\ \\ \end{bmatrix}$$
- If we know the $m \times d$ matrix Φ $(m \ll d)$ and the measurement vector x, can we recover every v?
- What kind of matrix Φ? How would we recover?

Problem Setup

- By simple linear algebra, we cannot recover every v in this setting. What if v has few non-zero coordinates?

Problem Setup

- By simple linear algebra, we cannot recover every v in this setting. What if v has few non-zero coordinates?
- We say a vector v is r-sparse if $|\operatorname{supp}(v)| \leq r$.

Problem Setup

- By simple linear algebra, we cannot recover every v in this setting. What if v has few non-zero coordinates?
- We say a vector v is r-sparse if $|\operatorname{supp}(v)| \leq r$.
- It turns out that we can recover any *sparse* vector v, even in the setting where $m \ll d$! This is called Sparse Reconstruction.

Problem Setup

- By simple linear algebra, we cannot recover every v in this setting. What if v has few non-zero coordinates?
- We say a vector v is r-sparse if $|\operatorname{supp}(v)| \leq r$.
- It turns out that we can recover any *sparse* vector v, even in the setting where $m \ll d$! This is called Sparse Reconstruction.
- What kind of matrix Φ? How would we recover? ... *Why do we care?*

Applications

- Error Correction

Applications

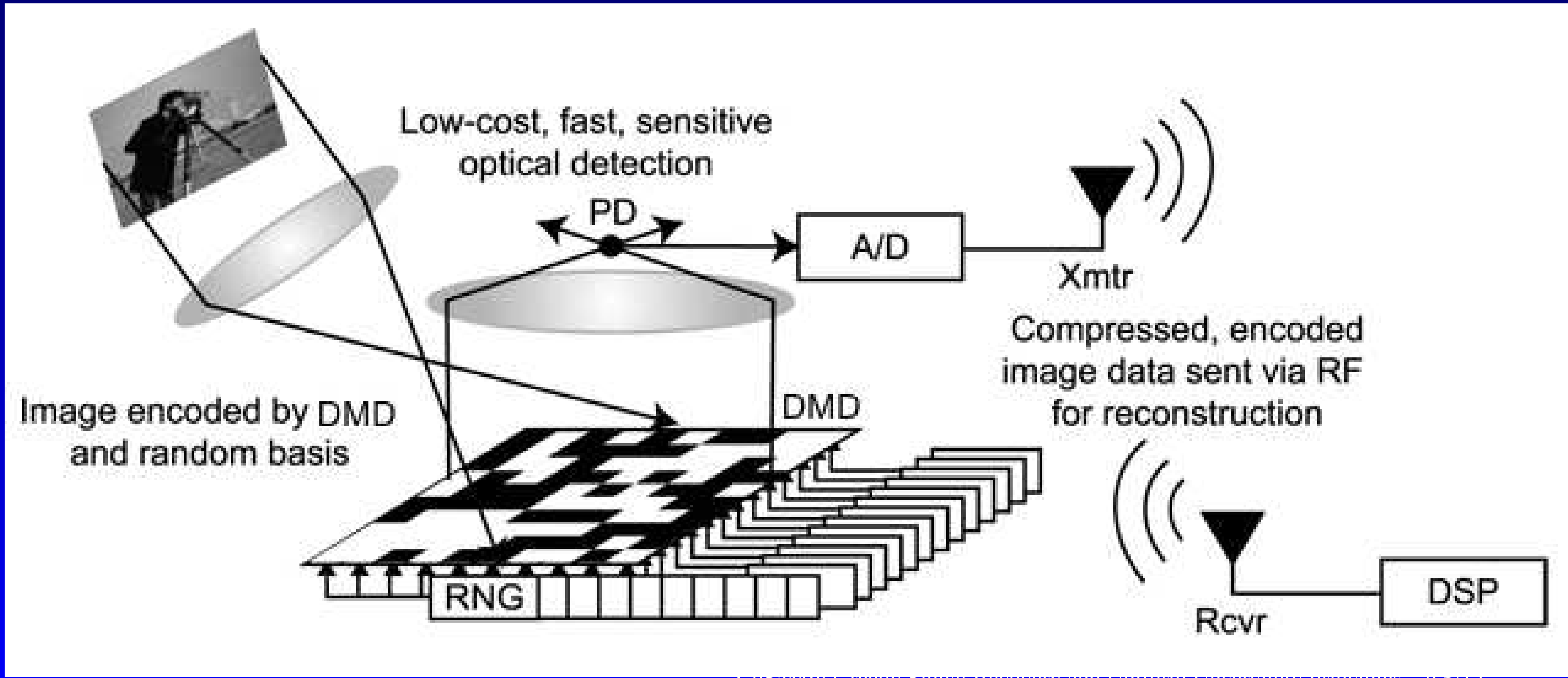
- Error Correction
- Medical Imaging

Applications

- Error Correction
- Medical Imaging
- Data Storage

Applications

- Error Correction
- Medical Imaging
- Data Storage
- Single Pixel Camera (Rice Univ. CS Group)



Important traits

- We thus want a matrix Φ along with a recovery algorithm, so that:

Important traits

- We thus want a matrix Φ along with a recovery algorithm, so that:
  1. Speed: The recovery of the signal v can be done *fast*

Important traits

- We thus want a matrix Φ along with a recovery algorithm, so that:
  1. Speed: The recovery of the signal v can be done *fast*
  2. Stability: If small errors are introduced, the algorithm still produces a good approximation to v.

Important traits

- We thus want a matrix Φ along with a recovery algorithm, so that:
  1. Speed: The recovery of the signal v can be done *fast*
  2. Stability: If small errors are introduced, the algorithm still produces a good approximation to v.
  3. Uniform Guarantees: We would like the matrix Φ and the algorithm to be able to recover *every* signal v.

Important traits

- We thus want a matrix Φ along with a recovery algorithm, so that:
  1. Speed: The recovery of the signal v can be done *fast*
  2. Stability: If small errors are introduced, the algorithm still produces a good approximation to v.
  3. Uniform Guarantees: We would like the matrix Φ and the algorithm to be able to recover *every* signal v.
- So, what kinds of matrices can we use, and how do we recover the signal v?

Requirements for Φ

- In theory, we need only that Φ be one-to-one on all r-sparse vectors. However, for practical reasons we require that Φ satisifies the Restricted Isometry Condition:

Requirements for Φ

- In theory, we need only that Φ be one-to-one on all r-sparse vectors. However, for practical reasons we require that Φ satisifies the Restricted Isometry Condition:
- $(1-\delta)\|v\|_2 \leq \|\Phi v\|_2 \leq (1+\delta)\|v\|_2$ for all r-sparse vectors v.

Requirements for Φ

- In theory, we need only that Φ be one-to-one on all r-sparse vectors. However, for practical reasons we require that Φ satisifies the Restricted Isometry Condition:
- $(1-\delta)\|v\|_2 \leq \|\Phi v\|_2 \leq (1+\delta)\|v\|_2$ for all r-sparse vectors v.
- Every set of r columns is approximately an orthonormal basis.

Requirements for Φ

- In theory, we need only that Φ be one-to-one on all r-sparse vectors. However, for practical reasons we require that Φ satisifies the Restricted Isometry Condition:
- $(1-\delta)\|v\|_2 \leq \|\Phi v\|_2 \leq (1+\delta)\|v\|_2$ for all r-sparse vectors v.
- Every set of r columns is approximately an orthonormal basis.
- Random matrices (Gaussian, Bernoulli, Random Fourier) satisfy the RIC with high probability for $m \approx r \log d$.

How do we recover?

- Two major approaches: ℓ_1-minimization and Greedy Methods

How do we recover?

- Two major approaches: ℓ_1-minimization and Greedy Methods
- ℓ_1-minimization (Basis Pursuit):

$$v = \operatorname{argmin} \|z\|_1, \ \Phi z = \Phi v = x$$

How do we recover?

- Two major approaches: ℓ_1-minimization and Greedy Methods
- ℓ_1-minimization (Basis Pursuit):

$$v = \operatorname{argmin} \|z\|_1, \; \Phi z = \Phi v = x$$

- Candès-Tao and Rudelson-Vershynin (2005) showed that when Φ satisfies the RIC, Basis Pursuit recovers every r-sparse vector (“uniform guarantees”).

How do we recover?

- Two major approaches: ℓ_1-minimization and Greedy Methods
- ℓ_1-minimization (Basis Pursuit):

$$v = \operatorname{argmin} \|z\|_1, \ \Phi z = \Phi v = x$$

- Candès-Tao and Rudelson-Vershynin (2005) showed that when Φ satisfies the RIC, Basis Pursuit recovers every r-sparse vector (“uniform guarantees”).
- Advantages: Uniform guarantees, Stable

How do we recover?

- Two major approaches: ℓ_1-minimization and Greedy Methods
- ℓ_1-minimization (Basis Pursuit):

$$v = \operatorname{argmin} \|z\|_1, \; \Phi z = \Phi v = x$$

- Candès-Tao and Rudelson-Vershynin (2005) showed that when Φ satisfies the RIC, Basis Pursuit recovers every r-sparse vector ("uniform guarantees").
- Advantages: Uniform guarantees, Stable
- Disadvantage: No known strongly polynomial time algorithm to solve a linear program.

Greedy Approaches

- Greedy methods compute the support of the signal v iteratively.

Greedy Approaches

- Greedy methods compute the support of the signal v iteratively.
- Since Φ forms approximately an ONB, the coordinates of $\Phi^*\Phi v$ are locally good estimaters of v.

Greedy Approaches

- Greedy methods compute the support of the signal v iteratively.
- Since Φ forms approximately an ONB, the coordinates of $\Phi^*\Phi v$ are locally good estimaters of v.
- The greedy method Orthogonal Matching Pursuit [Gilbert-Tropp, 2005] selects the largest coefficient of $\Phi^*\Phi v$ to be in the support, subtracts off its contribution, and iterates.

Greedy Approaches

- Greedy methods compute the support of the signal v iteratively.
- Since Φ forms approximately an ONB, the coordinates of $\Phi^*\Phi v$ are locally good estimaters of v.
- The greedy method Orthogonal Matching Pursuit [Gilbert-Tropp, 2005] selects the largest coefficient of $\Phi^*\Phi v$ to be in the support, subtracts off its contribution, and iterates.
- Advantage: Fast

Greedy Approaches

- Greedy methods compute the support of the signal v iteratively.
- Since Φ forms approximately an ONB, the coordinates of $\Phi^*\Phi v$ are locally good estimaters of v.
- The greedy method Orthogonal Matching Pursuit [Gilbert-Tropp, 2005] selects the largest coefficient of $\Phi^*\Phi v$ to be in the support, subtracts off its contribution, and iterates.
- Advantage: Fast
- Disadvantages: Not known to be stable, provides non-uniform guarantees (works for each *fixed* signal with high probability)

Bridging the Gap

- There existed this gap between the approaches. We bridged the gap with Regularized Orthogonal Matching Pursuit (ROMP) [N-Vershynin 2007].

Bridging the Gap

- There existed this gap between the approaches. We bridged the gap with Regularized Orthogonal Matching Pursuit (ROMP) [N-Vershynin 2007].
- Computes the support iteratively: Selects multiple coordinates of $\Phi^*\Phi v$ that have comparable magnitudes (are "regularized") at each iteration.

Bridging the Gap

- There existed this gap between the approaches. We bridged the gap with Regularized Orthogonal Matching Pursuit (ROMP) [N-Vershynin 2007].
- Computes the support iteratively: Selects multiple coordinates of $\Phi^*\Phi v$ that have comparable magnitudes (are “regularized”) at each iteration.
- Advantages: Fast, Stable, Uniform Guarantees

Bridging the Gap

- There existed this gap between the approaches. We bridged the gap with Regularized Orthogonal Matching Pursuit (ROMP) [N-Vershynin 2007].
- Computes the support iteratively: Selects multiple coordinates of $\Phi^*\Phi v$ that have comparable magnitudes (are “regularized”) at each iteration.
- Advantages: Fast, Stable, Uniform Guarantees
- Is it perfect?

Bridging the Gap

- There existed this gap between the approaches. We bridged the gap with Regularized Orthogonal Matching Pursuit (ROMP) [N-Vershynin 2007].
- Computes the support iteratively: Selects multiple coordinates of $\Phi^*\Phi v$ that have comparable magnitudes (are “regularized”) at each iteration.
- Advantages: Fast, Stable, Uniform Guarantees
- Is it perfect?
- Not quite: Requires a slightly stronger condition on the RIC than Basis Pursuit.

Finding Perfection

- We overcome this disadvantage with Compressive Sampling Matching Pursuit (CoSaMP) (N-Tropp, 2008).

Finding Perfection

- We overcome this disadvantage with Compressive Sampling Matching Pursuit (CoSaMP) (N-Tropp, 2008).
- Computes the signal iteratively: Selects the largest $O(r)$ coordinates of $\Phi^*\Phi v$, estimates the signal using this support, and prunes the signal to be r-sparse. Then repeats.

Finding Perfection

- We overcome this disadvantage with Compressive Sampling Matching Pursuit (CoSaMP) (N-Tropp, 2008).
- Computes the signal iteratively: Selects the largest $O(r)$ coordinates of $\Phi^*\Phi v$, estimates the signal using this support, and prunes the signal to be r-sparse. Then repeats.
- Advantages: Fast, Stable, Uniform Guarantees, no stronger condition needed on RIC

For more information

- dneedell@math.ucdavis.edu
- www.math.ucdavis.edu/~dneedell
- N and Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” submitted
- N and Vershynin, “Stable signal recovery from incomplete and inaccurate samples,” submitted
- N and Vershynin, “Uniform Uncertainty Principle and signal recovery via Regularized Orthogonal Matching Pursuit,” Found. Comput. Math., to appear.