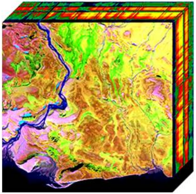
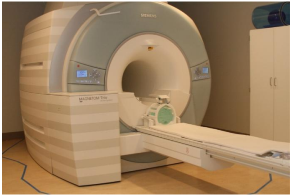
Solving Underdetermined Linear Equations and Overdetermined Quadratic Equations (using Convex Programming)

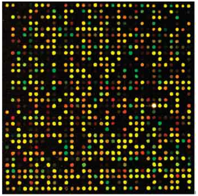
Justin Romberg
Georgia Tech, ECE

Caltech ROM-GR Workshop
June 7, 2013
Pasadena, California

Linear systems in data acquisition







Linear systems of equations are ubiquitous

Model:

$$\begin{bmatrix} \\ \\ y \\ \\ \\ \end{bmatrix} = \begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} \\ x \\ \\ \end{bmatrix}$$

y: data coming off of sensor

A: mathematical (linear) model for sensor

x: signal/image to reconstruct

Classical: When can we stably "invert" a matrix?

- Suppose we have an $M \times N$ observation matrix A with $M \geq N$ (MORE observations than unknowns), through which we observe

$$y = Ax_0 + \text{noise}$$

Classical: When can we stably "invert" a matrix?

- Suppose we have an $M \times N$ observation matrix A with $M \geq N$ (MORE observations than unknowns), through which we observe

$$y = Ax_0 + \text{noise}$$

- Standard way to recover x_0, use the *pseudo-inverse*

$$\text{solve } \min_x \|y - Ax\|_2^2 \quad \Leftrightarrow \quad \hat{x} = (A^T A)^{-1} A^T y$$

Classical: When can we stably "invert" a matrix?

- Suppose we have an $M \times N$ observation matrix A with $M \geq N$ (MORE observations than unknowns), through which we observe

$$y = Ax_0 + \text{noise}$$

- Standard way to recover x_0, use the *pseudo-inverse*

$$\text{solve } \min_x \|y - Ax\|_2^2 \quad \Leftrightarrow \quad \hat{x} = (A^T A)^{-1} A^T y$$

- Q: When is this recovery stable? That is, when is

$$\|\hat{x} - x_0\|_2^2 \sim \|\text{noise}\|_2^2 \quad ?$$

Classical: When can we stably "invert" a matrix?

- Suppose we have an $M \times N$ observation matrix A with $M \geq N$ (MORE observations than unknowns), through which we observe

$$y = Ax_0 + \text{noise}$$

- Standard way to recover x_0, use the *pseudo-inverse*

$$\text{solve } \min_x \|y - Ax\|_2^2 \quad \Leftrightarrow \quad \hat{x} = (A^T A)^{-1} A^T y$$

- Q: When is this recovery stable? That is, when is

$$\|\hat{x} - x_0\|_2^2 \sim \|\text{noise}\|_2^2 \quad ?$$

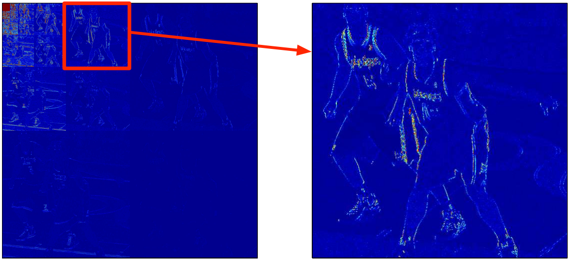
- A: When the matrix A preserves *distances* ...

$$\|A(x_1 - x_2)\|_2^2 \approx \|x_1 - x_2\|_2^2 \quad \text{for all } x_1, x_2 \in \mathbb{R}^N$$

Sparsity

Decompose signal/image $x(t)$ in orthobasis $\{\psi_i(t)\}_i$

$$x(t) = \sum_i \alpha_i \psi_i(t)$$



wavelet transform

zoom

x_0

$\{\alpha_i\}_i$

Wavelet approximation

Take 1% of *largest* coefficients, set the rest to zero (adaptive)

original

approximated

rel. error $= 0.031$

When can we stably recover an S-sparse vector?

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} \quad & \Phi & \quad \end{bmatrix} \begin{bmatrix} x_0 \\ \\ \\ \\ \end{bmatrix}$$

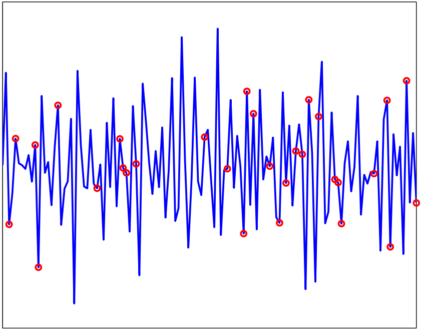
- Now we have an underdetermined $M \times N$ system Φ (FEWER measurements than unknowns), and observe

$$y = \Phi x_0 + \text{noise}$$

Sampling a superposition of sinusoids

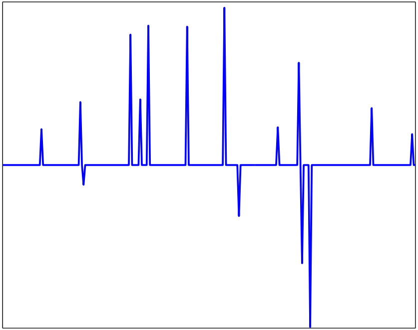
We take M samples of a superposition of S sinusoids:

Time domain $x_0(t)$



Measure M samples
(red circles = samples)

Frequency domain $\hat{x}_0(\omega)$

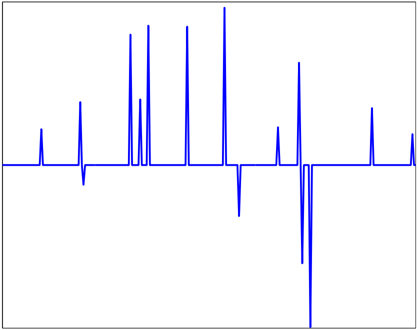


S nonzero components

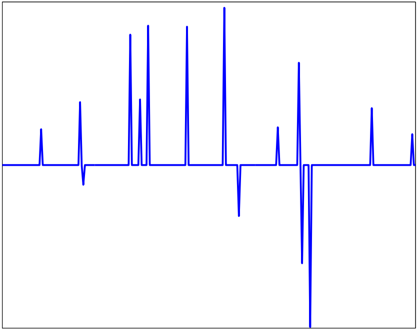
Sampling a superposition of sinusoids

Reconstruct by solving

$$\min_x \ \|\hat{x}\|_{\ell_1} \quad \text{subject to} \quad x(t_m) = x_0(t_m), \quad m = 1, \ldots, M$$



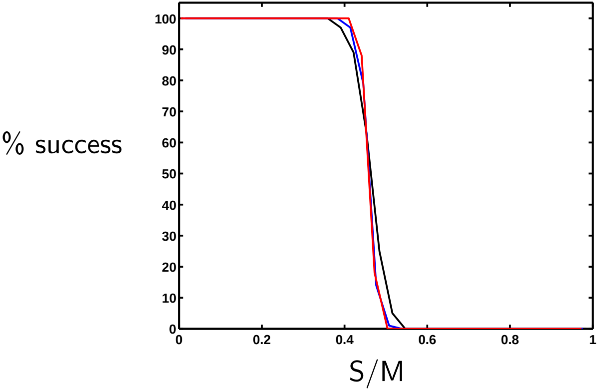
original $\hat{x}_0$, $S = 15$



perfect recovery from 30 samples

Numerical recovery curves

- Resolutions $N = 256, 512, 1024$ (black, blue, red)
- Signal composed of S randomly selected sinusoids
- Sample at M randomly selected locations



- In practice, perfect recovery occurs when $M \approx 2S$ for $N \approx 1000$

A nonlinear sampling theorem

Exact Recovery Theorem (Candès, R, Tao, 2004):

- Unknown $\hat{x}_0$ is supported on set of size S
- Select M sample locations $\{t_m\}$ "at random" with

$$M \;\geq\; \text{Const} \cdot S \log N$$

- Take time-domain samples (measurements) $y_m = x_0(t_m)$
- Solve

$$\min_x \|\hat{x}\|_{\ell_1} \quad \text{subject to} \quad x(t_m) = y_m, \;\; m = 1, \ldots, M$$

- Solution is *exactly* f with extremely high probability

When can we stably recover an S-sparse vector?

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} & \Phi & \end{bmatrix} \begin{bmatrix} x_0 \\ \\ \\ \\ \end{bmatrix}$$

- Now we have an underdetermined $M \times N$ system Φ (FEWER measurements than unknowns), and observe

$$y = \Phi x_0 + \text{noise}$$

When can we stably recover an S-sparse vector?

- Now we have an underdetermined $M \times N$ system Φ (FEWER measurements than unknowns), and observe

$$y = \Phi x_0 + \text{noise}$$

- We can recover x_0 when Φ *keeps sparse signals separated*

$$\|\Phi(x_1 - x_2)\|_2^2 \;\approx\; \|x_1 - x_2\|_2^2$$

for all S-sparse x_1, x_2

When can we stably recover an S-sparse vector?

- Now we have an underdetermined $M \times N$ system Φ (FEWER measurements than unknowns), and observe

$$y = \Phi x_0 + \text{noise}$$

- We can recover x_0 when Φ *keeps sparse signals separated*

$$\|\Phi(x_1 - x_2)\|_2^2 \;\approx\; \|x_1 - x_2\|_2^2$$

  for all S-sparse x_1, x_2

- To recover x_0, we solve

$$\min_x \; \|x\|_0 \quad \text{subject to} \quad \Phi x = y$$

  $\|x\|_0 =$ number of nonzero terms in x

- This program is *computationally intractable*

When can we stably recover an S-sparse vector?

- Now we have an underdetermined $M \times N$ system Φ (FEWER measurements than unknowns), and observe

$$y = \Phi x_0 + \text{noise}$$

- We can recover x_0 when Φ *keeps sparse signals separated*

$$\|\Phi(x_1 - x_2)\|_2^2 \approx \|x_1 - x_2\|_2^2$$

  for all S-sparse x_1, x_2

- A relaxed (convex) program

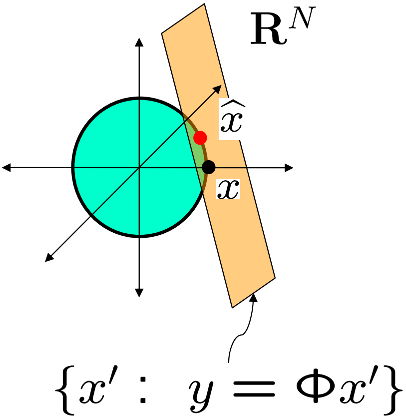
$$\min_x \|x\|_1 \quad \text{subject to} \quad \Phi x = y$$

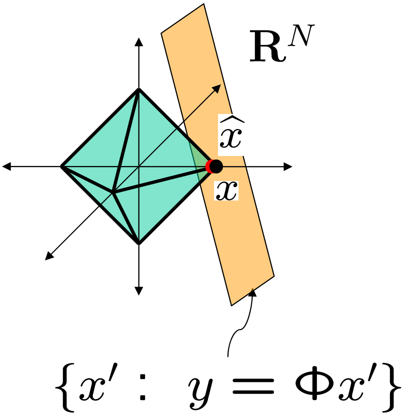
  $\|x\|_1 = \sum_k |x_k|$

- This program is very tractable (linear program)
- The convex program can recover nearly all “identifiable” sparse vectors, and it is *robust*.

Intuition for ℓ_1

$\min_x \|x\|_2$ s.t. $\Phi x = y$ $\quad\quad$ $\min_x \|x\|_1$ s.t. $\Phi x = y$





Sparse recovery algorithms

ℓ_1 can recover sparse vectors "almost anytime" it is possible

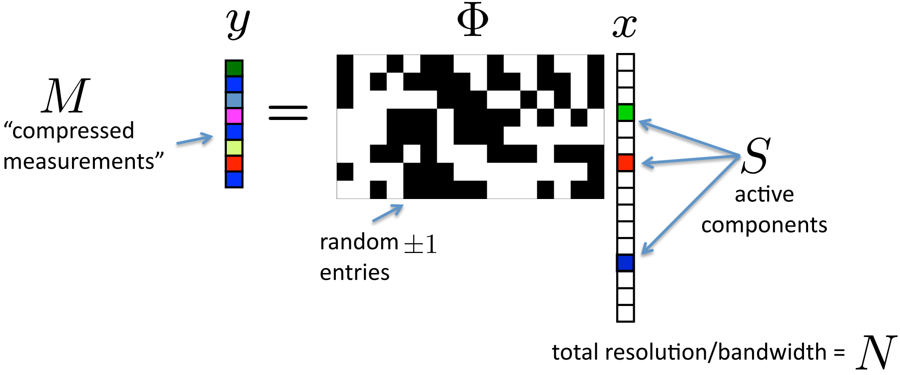
- perfect recovery with no noise
- stable recovery in the presence of noise
- robust recovery when the signal is not exactly sparse

Sparse recovery algorithms

Other recovery techniques have similar theoretical properties
(their practical effectiveness varies with applications)

- greedy algorithms
- iterative thresholding
- belief propagation
- specialized decoding algorithms

What kind of matrices keep sparse signals separated?

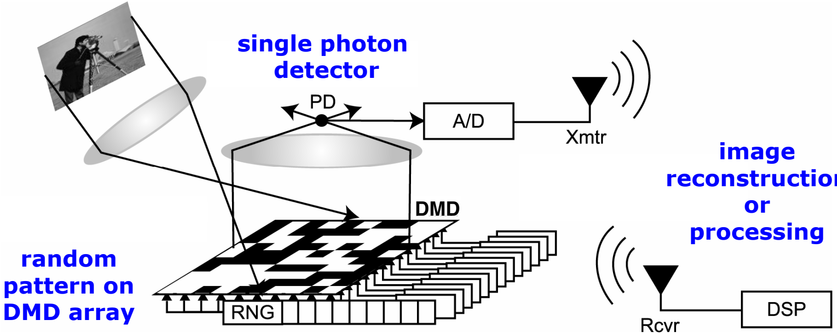


- *Random matrices* are provably efficient
- We can recover S-sparse x from

$$M \gtrsim S \cdot \log(N/S)$$

measurements

Rice single pixel camera

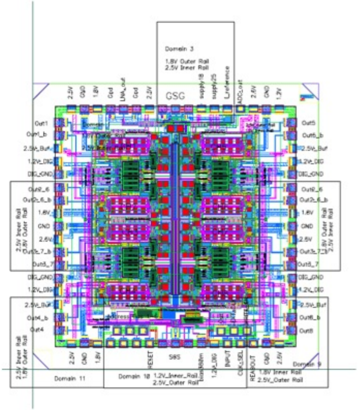


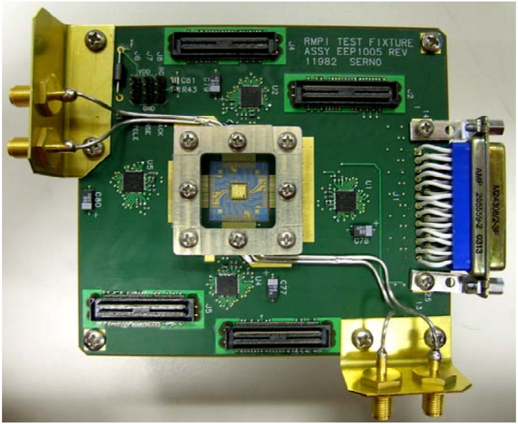
(Duarte, Davenport, Takhar, Laska, Sun, Kelly, Baraniuk '08)

Hyperspectral imaging

256 frequency bands, 10s of megapixels, 30 frames per second ...

DARPA's Analog-to-Information

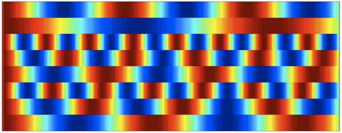




Multichannel ADC/receiver for identifying radar pulses
Covers ~ 3 GHz with ~ 400 MHz sampling rate

Compressive sensing with structured randomness

Subsampled rows of “incoherent” orthogonal matrix



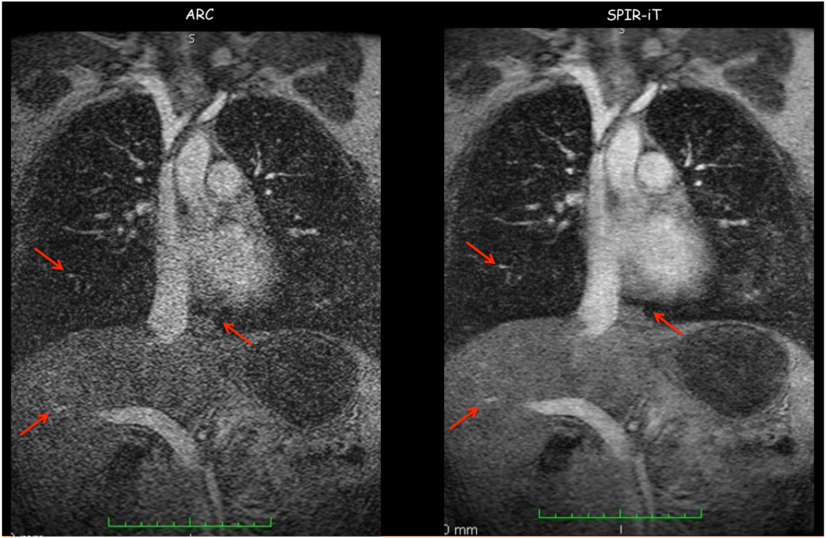
Can recover S-sparse x_0 with

$$M \gtrsim S \log^a N$$

measurements

Candes, R, Tao, Rudelson, Vershynin, Tropp, ...

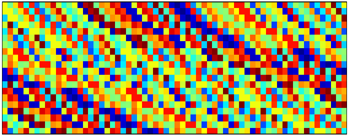
Accelerated MRI



(Lustig et al. '08)

Matrices for sparse recovery with structured randomness

Random convolution + subsampling



Universal; Can recover S-sparse x_0 with

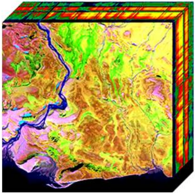
$$M \gtrsim S \log^a N$$

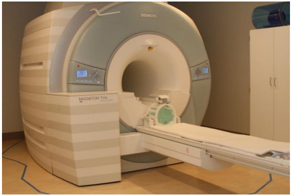
Applications include:

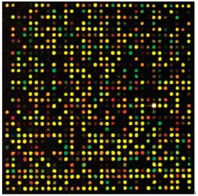
- radar imaging
- sonar imaging
- seismic exploration
- channel estimation for communications
- super-resolved imaging

R, Bajwa, Haupt, Tropp, Rauhut, ...

Integrating compression and sensing







Recovering a matrix from limited observations

Suppose we are interested in recovering the values of a **matrix** $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} X_{1,1} & X_{1,2} & X_{1,3} & X_{1,4} & X_{1,5} \\ X_{2,1} & X_{2,2} & X_{2,3} & X_{2,4} & X_{2,5} \\ X_{3,1} & X_{3,2} & X_{3,3} & X_{3,4} & X_{3,5} \\ X_{4,1} & X_{4,2} & X_{4,3} & X_{4,4} & X_{4,5} \\ X_{5,1} & X_{5,2} & X_{5,3} & X_{5,4} & X_{5,5} \end{bmatrix}$$

We are given a series of different *linear combinations* of the entries

$$y = \mathcal{A}(\mathbf{X})$$

Example: matrix completion

Suppose we do not see all the entries in a matrix ...

$$\mathbf{X} = \begin{bmatrix} X_{1,1} & - & X_{1,3} & - & X_{1,5} \\ - & X_{2,2} & - & X_{2,4} & - \\ - & X_{3,2} & X_{3,3} & - & - \\ X_{4,1} & - & - & X_{4,4} & X_{4,5} \\ - & - & - & X_{5,4} & X_{5,5} \end{bmatrix}$$

... can we "fill in the blanks"?

Applications of matrix completion

Recommender Systems

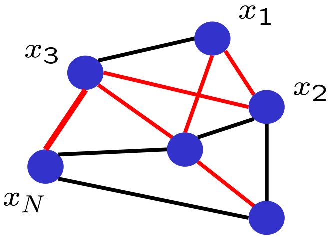
amazon.com

NETFLIX

match.com

chemistry

Euclidean Embedding



Rank of: Data Matrix

Gram Matrix

(slide courtesy of Benjamin Recht)

Low rank structure

$$\underbrace{\begin{bmatrix} \\ \\ & \boldsymbol{X} & \\ \\ \\ \end{bmatrix}}_{K \times N} = \underbrace{\begin{bmatrix} \\ \\ \boldsymbol{L} \\ \\ \\ \end{bmatrix}}_{K \times R} \underbrace{\begin{bmatrix} & \boldsymbol{R}^T & \end{bmatrix}}_{R \times N}$$

When can we stably recover a rank-R matrix?

- We have an underdetermined linear operator $\mathcal{A}$

$$\mathcal{A} : \mathbb{R}^{K\times N} \to L, \qquad L \ll KN$$

and observe

$$\mathbf{y} = \mathcal{A}(\mathbf{X}_0) + \text{noise}$$

where $\mathbf{X}_0$ has rank R

- We can recover $\mathbf{X}_0$ when $\mathcal{A}$ *keeps low-rank matrices separated*

$$\|\mathcal{A}(\mathbf{X}_1 - \mathbf{X}_2)\|_2^2 \;\approx\; \|\mathbf{X}_1 - \mathbf{X}_2\|_F^2$$

for all rank-R $\mathbf{X}_1, \mathbf{X}_2$

When can we stably recover a rank-R matrix?

- We have an underdetermined linear operator $\mathcal{A}$

  $$\mathcal{A} : \mathbb{R}^{K \times N} \to L, \qquad L \ll KN$$

  and observe

  $$\mathbf{y} = \mathcal{A}(\mathbf{X}_0) + \text{noise}$$

  where $\mathbf{X}_0$ has rank R

- To recover $\mathbf{X}_0$, we would like to solve

  $$\min_{\mathbf{X}} \ \operatorname{rank}(\mathbf{X}) \quad \text{subject to} \quad \mathcal{A}(\mathbf{X}) = \mathbf{y}$$

  but this is intractable

When can we stably recover a rank-R matrix?

- We have an underdetermined linear operator $\mathcal{A}$

$$\mathcal{A} : \mathbb{R}^{K \times N} \to L, \qquad L \ll KN$$

  and observe

$$\mathbf{y} = \mathcal{A}(\mathbf{X}_0) + \text{noise}$$

  where $\mathbf{X}_0$ has rank R

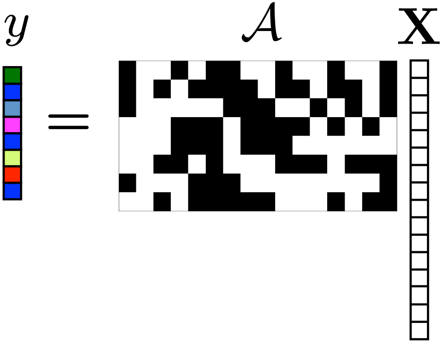
- A relaxed (convex) program

$$\min_{\mathbf{X}} \ \|\mathbf{X}\|_* \quad \text{subject to} \quad \mathcal{A}(\mathbf{X}) = \mathbf{y}$$

  where $\|\mathbf{X}\|_* =$ sum of the singular values of $\mathbf{X}$

Matrix Recovery

Take vectorize $\mathbf{X}$, stack up vectorized $\mathbf{A}_m$ as rows of a matrix



Independent Gaussian entires in the $\mathbf{A}_m$ embeds rank-R matrices when

$$M \ \gtrsim \ R(K+N)$$

(Recht, Fazel, Parillo, Candes, Plan, ...)

Example: matrix completion

Suppose we do not see all the entries in a matrix ...

$$\mathbf{X} = \begin{bmatrix} X_{1,1} & - & X_{1,3} & - & X_{1,5} \\ - & X_{2,2} & - & X_{2,4} & - \\ - & X_{3,2} & X_{3,3} & - & - \\ X_{4,1} & - & - & X_{4,4} & X_{4,5} \\ - & - & - & X_{5,4} & X_{5,5} \end{bmatrix}$$

... we can fill them in from

$$M \ \gtrsim \ R(K+N)\log^2(KN)$$

randomly chosen samples if $\mathbf{X}$ is *diffuse*.

(Recht, Candes, Tao, Montenari, Oh, ...)

Summary: random projections and structured recovery

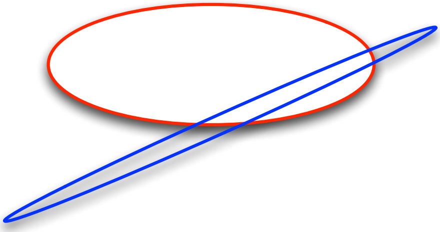
The number of measurements (dimension of the projection) needed for structured recovery depends on the geometrical complexity of the class.

Three examples:

structure	number of measurements
S-sparse vectors, length N	$S\log(N/S)$
rank-R matrix, size $K \times N$	$R(K+N)$
manifold in $\mathbb{R}^N$, intrins. dim. K	$K \cdot$ (function of vol, curvature, etc)

Systems of quadratic and bilinear equations

Quadratic equations



Quadratic equations contain unknown terms multiplied by one another

$$x_1x_3 + x_2 + x_5^2 = 13$$
$$3x_2x_6 - 7x_3 + 9x_4^2 = -12$$
$$\vdots$$

Their nonlinearity makes them trickier to solve, and the computational framework is nowhere nearly as strong as for linear equations

Recasting quadratic equations

$$vv^T = \begin{bmatrix} v_1^2 & v_1 v_2 & v_1 v_3 & \cdots & v_1 v_N \\ v_2 v_1 & v_2^2 & v_2 v_3 & \cdots & v_2 v_N \\ v_3 v_1 & v_3 v_2 & v_3^3 & \cdots & v_3 v_N \\ \cdots & & & \ddots & \vdots \\ v_N v_1 & v_N v_2 & v_N v_3 & \cdots & v_N^2 \end{bmatrix}$$

$$2v_1^2 + 5v_3 v_1 + 7v_2 v_3 = \cdots$$

$$v_2 v_1 + 9v_2^2 + 4v_3 v_2 = \cdots$$

A quadratic system of equations can be recast as a linear system of equations on a matrix that has rank 1.

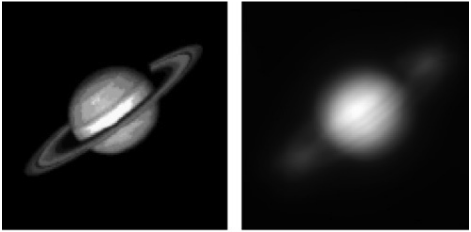
Recasting quadratic equations

$$vv^T = \begin{bmatrix} v_1^2 & v_1 v_2 & v_1 v_3 & \cdots & v_1 v_N \\ v_2 v_1 & v_2^2 & v_2 v_3 & \cdots & v_2 v_N \\ v_3 v_1 & v_3 v_2 & v_3^3 & \cdots & v_3 v_N \\ \cdots & & & \ddots & \vdots \\ v_N v_1 & v_N v_2 & v_N v_3 & \cdots & v_N^2 \end{bmatrix}$$

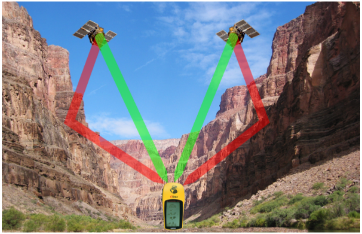
Compressive (low rank) recovery $\Rightarrow$
"Generic" quadratic systems with cN equations and N unknowns can be solved using nuclear norm minimization

Blind deconvolution

image deblurring



multipath in wireless comm



We observe

$$y[n] = \sum_{\ell} w[\ell]\, x[n-\ell]$$

and want to “untangle” w and x.

Recasting as linear equations

While each observation is a *quadratic* combination of the unknowns:

$$y[\ell] = \sum_n w[n]x[\ell - n]$$

it is *linear* is the outer product:

$$wx^{\mathrm{T}} = \begin{bmatrix} w[1]x[1] & w[1]x[2] & \cdots & w[1]x[L] \\ w[2]x[1] & w[2]x[2] & \cdots & w[2]x[L] \\ \vdots & \vdots & \ddots & \\ w[L]x[1] & w[L]x[2] & \cdots & w[L]x[L] \end{bmatrix}$$

So $y = \mathcal{A}(X_0)$, where $X_0 = wx^{\mathrm{T}}$ has rank 1.

Recasting as linear equations

While each observation is a *quadratic* combination of the unknowns:

$$y[\ell] = \sum_n w[n]x[\ell - n]$$

it is *linear* is the outer product:

$$(Bh)(Cm)^{\mathrm{T}} = \begin{bmatrix} \langle b_1, h\rangle\langle m, c_1\rangle & \langle b_1, h\rangle\langle m, c_2\rangle & \cdots & \langle b_1, h\rangle\langle m, c_N\rangle \\ \langle b_2, h\rangle\langle m, c_1\rangle & \langle b_2, h\rangle\langle m, c_2\rangle & \cdots & \langle b_2, h\rangle\langle m, c_N\rangle \\ \vdots & \vdots & \ddots & \\ \langle b_K, h\rangle\langle m, c_1\rangle & \langle b_K, h\rangle\langle m, c_2\rangle & \cdots & \langle b_K, h\rangle\langle m, c_N\rangle \end{bmatrix}$$

where b_k is the kth row of B, and c_n is the nth row of C.

So y is *linear* in hm^{T}.

Blind deconvolution theoretical results

We observe

$$\begin{aligned} y &= w * x, \qquad w = Bh, \;\; x = Cm \\ &= \mathcal{A}(hm^*), \qquad h \in \mathbb{R}^K, \;\; m \in \mathbb{R}^N, \end{aligned}$$

and then solve

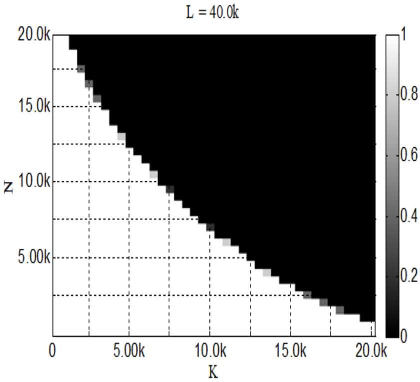
$$\min_X \; \|X\|_* \quad \text{subject to} \quad \mathcal{A}(X) = y.$$

Ahmed, Recht, R, '12:
If B is "incoherent" in the Fourier domain, and C is randomly chosen, then we will recover $X_0 = hm^*$ exactly (with high probability) when

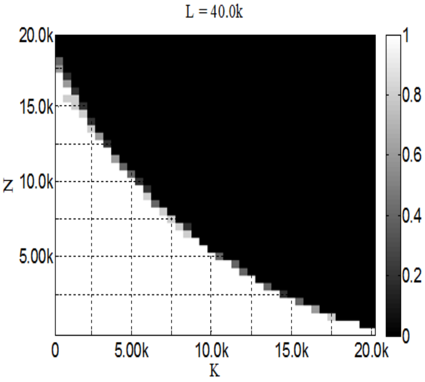
$$\max(K, N) \;\leq\; \frac{L}{\log^3 L}$$

Numerical results

white $=$ 100% success, black $=$ 0% success



$\mathbf{w}$ sparse, $\mathbf{x}$ sparse

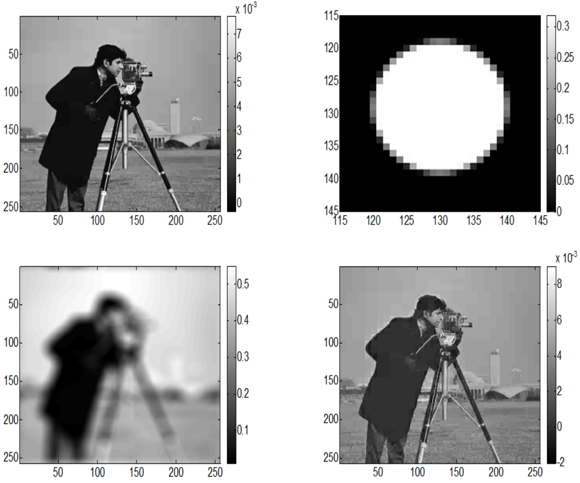


$\mathbf{w}$ sparse, $\mathbf{x}$ short

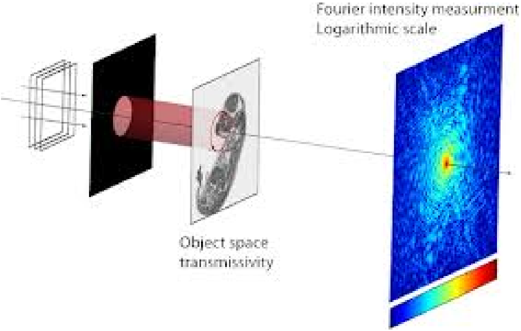
We can take $K + N \approx L/3$

Numerical results

Unknown image with known support in the wavelet domain,
Unknown blurring kernel with known support in spatial domain



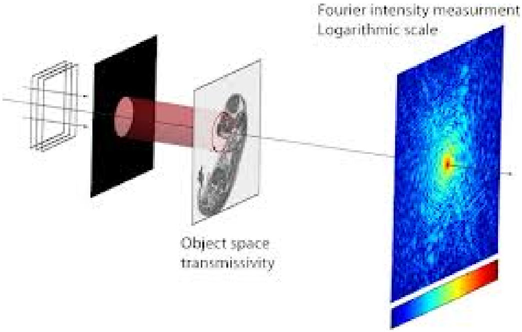
Phase retrieval



(image courtesy of Manuel Guizar-Sicairos)

Observe the *magnitude* of the Fourier transform $|\hat{x}(\omega)|^2$
$\hat{x}(\omega)$ is complex, and its phase carries important information

Phase retrieval



(image courtesy of Manuel Guizar-Sicairos)

Recently, Candès, Strohmer, Voroninski, have looked at stylized version of **phase retrieval**:

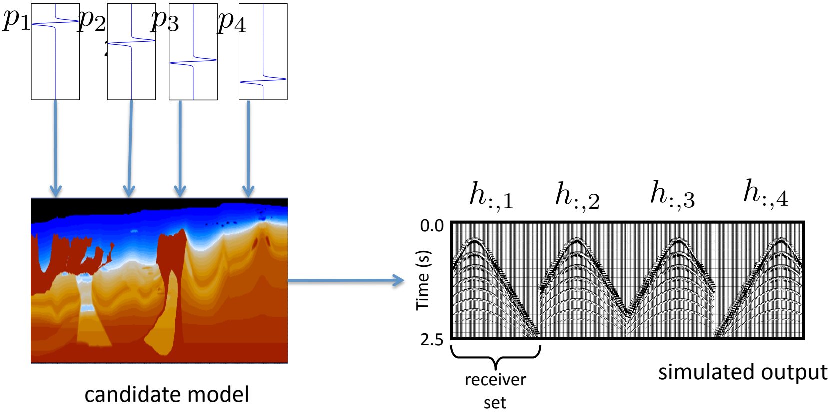
$$\text{observe} \quad y_\ell = |\langle \mathbf{a}_\ell, \mathbf{x} \rangle|^2 \quad \ell = 1, \ldots, L$$

and shown that $\mathbf{x} \in \mathbb{R}^N$ can be recovered when $L \sim \mathrm{Const} \cdot N$ for *random* $\mathbf{a}_\ell$.

Random projections in fast forward modeling

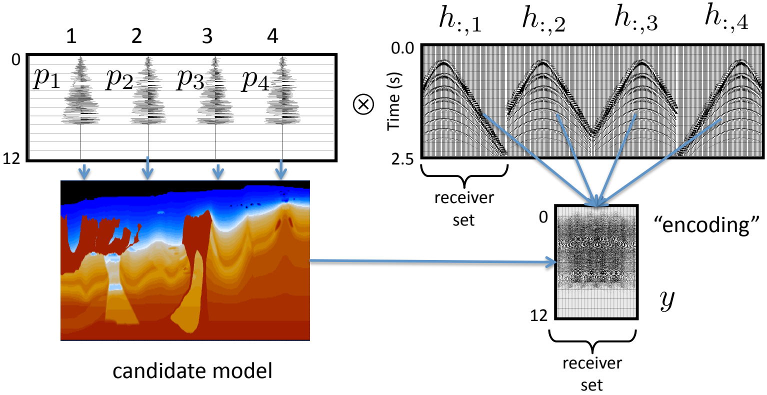
Forward modeling/simulation

- Given a candidate model of the earth, we want to estimate the channel between each source/receiver pair

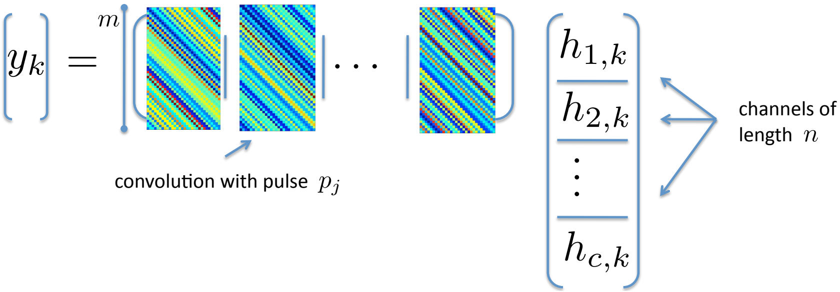


Simultaneous activation

- Run a single simulation with all of the sources activated simultaneously with random waveforms
- The channel responses interfere with one another, but the randomness "codes" them in such a way that they can be separated later

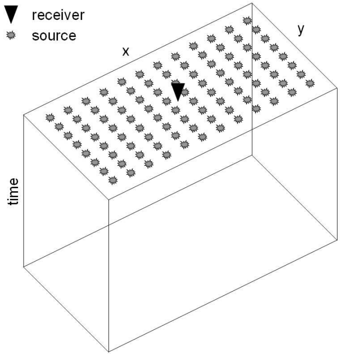


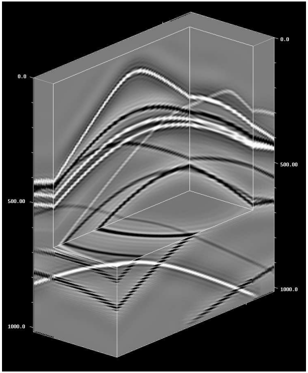
Multiple channel linear algebra



- How long does each pulse need to be to recover all of the channels? (the system is $m \times nc$, $m =$ pulse length, $c = \#$ channels)
- Of course we can do it for $m \geq nc$
- But if the channels have a combined sparsity of S, then we can take $m \sim s + n$

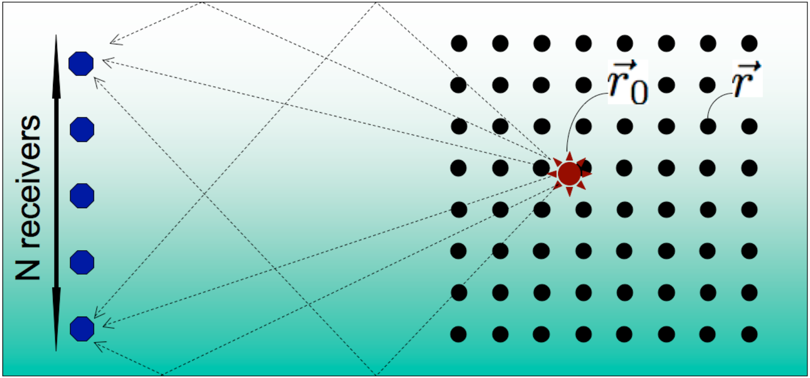
Seismic imaging simulation





- Array of 128×64 (8192) sources activated simultaneously (1 receiver)
- Sparsity enforced in the *curvelet domain*
- Can "compress" computations $\sim 20\times$

Source localization

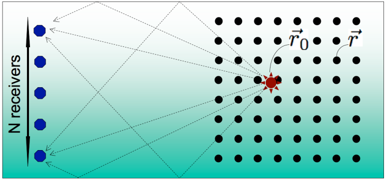


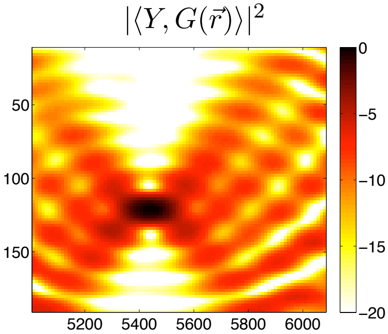
We observe a narrowband source emitting from (unknown) location $\vec{r_0}$:

$$Y = \alpha G(\vec{r_0}) + \text{noise}, \quad Y \in \mathbb{C}^N$$

Goal: estimate $\vec{r_0}$ using only *implicit* knowledge of the channel G

Matched field processing



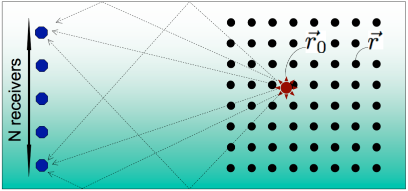


Given observations Y, estimate $\vec{r_0}$ by "matching against the field":

$$\hat{r} = \arg\min_{\vec{r}} \min_{\beta\in\mathbb{C}} \|Y - \beta G(\vec{r})\|^2 = \max_{\vec{r}} \frac{|\langle Y, G(\vec{r})\rangle|^2}{\|G(\vec{r})\|^2} \approx |\langle Y, G(\vec{r})\rangle|^2$$

We do not have direct access to G, but can calculate $\langle Y, G(\vec{r})\rangle$ for all $\vec{r}$ using *time-reversal*

Multiple realizations



- We receive a series of measurements $Y_1, \ldots, Y_K$ from the same environment (but possibly different source locations)
- Calculating $G^H Y_k$ for each instance can be expensive (requires a PDE solve)
- A naïve precomputing approach:
  - set off a source at each receiver location $\vec{s}_i$
  - time reverse $G^H 1_{\vec{s}_i}$ to "pick off" a row of G
- We can use ideas from compressive sensing to significantly reduce the amount of precomputation

Coded simulations

- Pre-compute the responses to a series of *randomly and simultaneously activated* sources along the receiver array

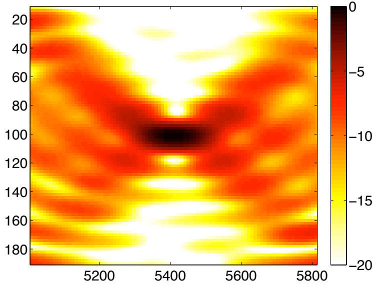
$$b_1 = G^H \phi_1, \quad b_2 = G^H \phi_2, \quad \dots \quad b_M = G^H \phi_M,$$

  where the ϕ_m are random vectors
- Stack up the b_m^H to form the matrix ΦG
- Given the observations Y, code them to form ΦY, and solve

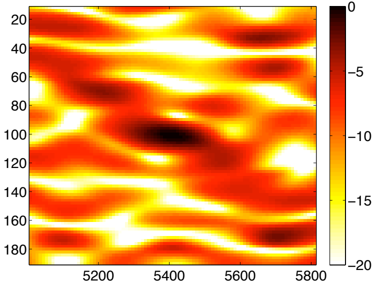
$$\hat{r}_{cs} = \arg\min_{\vec{r}} \min_{\beta \in \mathbb{C}} \|\Phi Y - \beta \Phi G(\vec{r})\|_2^2 = \arg\max_{\vec{r}} \frac{|\langle \Phi Y, \Phi G(\vec{r}) \rangle|^2}{\|\Phi G(\vec{r})\|^2}$$

Compressive ambiguity functions

ambiguity function $(G^H Y)(\vec{r})$



compressed amb func $(G^H \Phi^H \Phi Y)(\vec{r})$

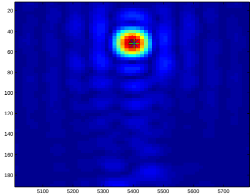


$M = 10$ (compare to 37 receivers)

- The compressed ambiguity function is a *random process* whose mean is the true ambiguity function
- For very modest M, these two functions peak in the same place

Analysis

- Suppose we can approximate $G^T G(\vec{r})$ with a 2D Gaussian with widths λ_1, λ_2



set $W = \frac{\text{area}}{\lambda_1 \lambda_2}$

then we can reliably estimate $\vec{r}_0$ when

$$M \gtrsim \log W$$

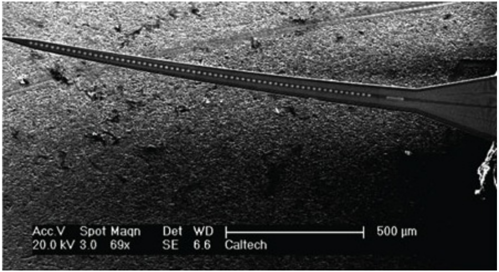
and withstand noise levels

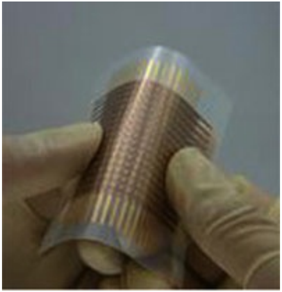
$$\sigma^2 \lesssim A^2 \frac{M}{W \log W}$$

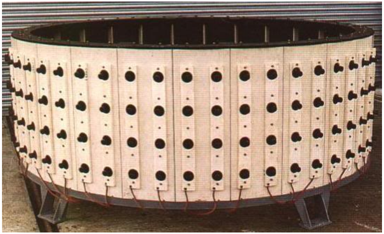
$A =$ source amplitude

Sampling ensembles of correlated signals

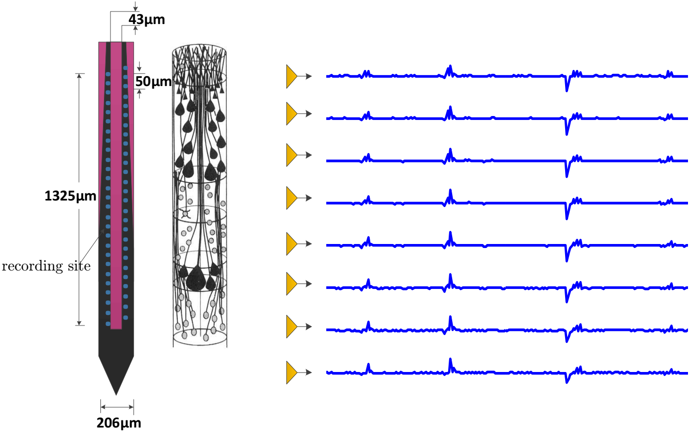
Sensor arrays







Neural probes

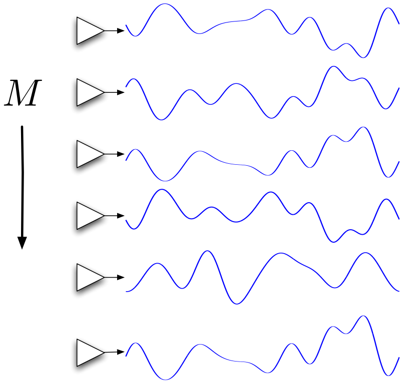


Up to 100s of channels sampled at $\sim$ 100 kHz

10s of millions of samples/second

Near Future: 1 million channels, terabits per second

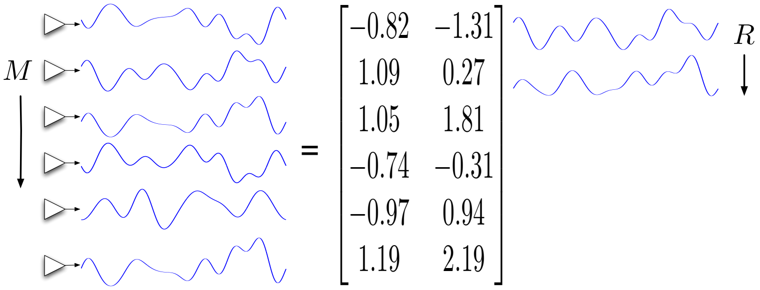
Correlated signals



Nyquist acquisition:

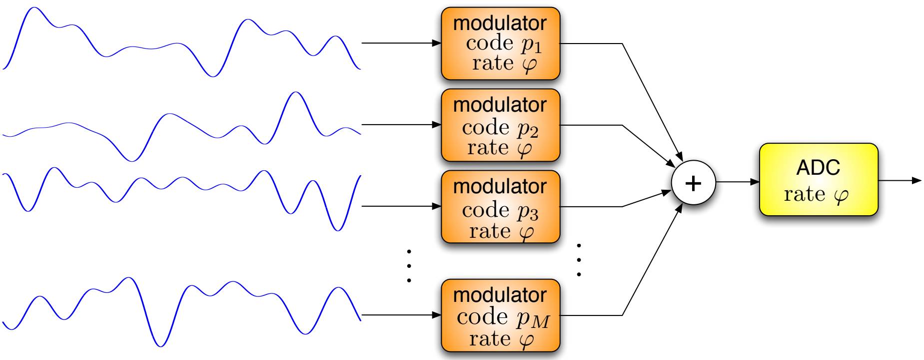
$$\text{sampling rate} \approx (\text{number of signals}) \times (\text{bandwith}) \\ = M \cdot W$$

Correlated signals



Can we exploit the *latent* correlation structure to reduce the sampling rate?

Coded multiplexing



Architecture that achieves

$$\text{sampling rate} \approx (\text{independent signals}) \times (\text{bandwidth})$$
$$\approx RW \log^{3/2} W$$

References

- E. Candes, J. Romberg, T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *CPAM*, 2006.
- E. Candes and J. Romberg, "Sparsity and Incoherence in Compressive Sampling," *Inverse Problems*, 2007.
- J. Romberg, "Compressive Sensing by Random Convolution," *SIAM J. Imaging Science*, 2009.
- A. Ahmed, B. Recht, and J. Romberg, "Blind Deconvolution using Convex Programming," submitted to *IEEE Transactions on Information Theory*, November 2012.

- A. Ahmed and J. Romberg, "Compressive Sampling of Correlated Signals," Manuscript in preparation. Preliminary version at *IEEE Asilomar Conference on Signals, Systems, and Computers*, October 2011.

- A. Ahmed and J. Romberg, "Compressive Multiplexers for Correlated Signals," Manuscript in preparation. Preliminary version at *IEEE Asilomar Conference on Signals, Systems, and Computers*, November 2012.

http://users.ece.gatech.edu/~justin/Publications.html