Optimized Projection Directions for Compressed Sensing

Michael Elad
The Computer Science Department
The Technion – Israel Institute of technology
Haifa 32000, Israel

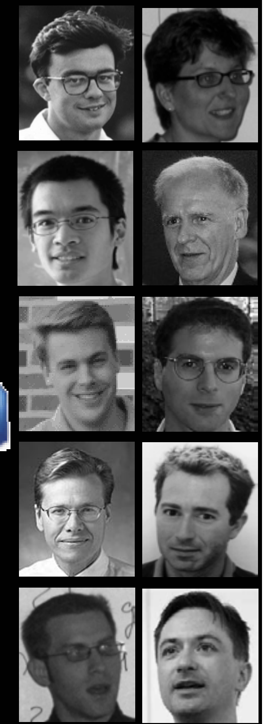
The IV Workshop on SIP & IT
Holon Institute of Technology
June 20th, 2007

מכון טכנולוגי חולון
Holon Institute of Technology

What is This Talk About?

Compressed Sensing (CS):

- ❑ An emerging field of research,
- ❑ Deals with combined of sensing-compression,
- ❑ Offering novel sampling results for signals,
- ❑ Leans on sparse & redundant modeling of signals,
- ❑ Key players: Candes, Tao, Romberg, Donoho, Tropp, Baraniuk, Gilbert, DeVore, Strauss, Cohen, …



A key ingredient in the use of Compressed-Sensing is

Linear Projections

In this talk we focus on this issue, offering a way to design these projections to yield better CS performance.

Agenda

1. What is Compressed-Sensing (CS)?
   Little-bit of Background
2. The Choice of Projections for CS
   Criteria for Optimality
3. The Obtained Results
   Simulations
4. What Next?
   Conclusions & Open problems

This lecture is based on the paper:

M. Elad, "Optimized Projections for Compressed-Sensing", to appear in *IEEE Trans. on Signal Processing*.

Typical Acquisition

The Typical Signal Acquisition Scenario

Sample a signal very densely, and then compress the information for storage or transmission

A typical example

- This 6.1 Mega-Pixels digital camera senses 6.1e+6 samples to construct an image.
- The image is then compressed using JPEG to an average size smaller than 1MB – a compression ratio of ~20.

Compressed–Sensing?

A natural question to ask is

Could the two processes (sensing & compressing) be combined ?

The answer is YES!

This is what **Compressed-Sensing (CS)** is all about.

[Candes, Romberg, & Tao `04, Donoho `06, Candes '06, Tsaig & Donoho `06]

How CS could be Done?

A signal of interest $\underline{x} \in \Re^n$

Instead of the traditional …

Sense all the n samples → Compress the raw data → ••• → Decompress for later processing

Apply the following …

Sense p<<n values $\{f_i(\underline{x})\}_{i=1}^{p}$ → These p values represent the signal in a compressed form ••• → Reconstruct $\underline{x}$ from $\{f_i(\underline{x})\}_{i=1}^{p}$ for later processing

Few Fundamental Questions

A signal of interest $\underline{x} \in \Re^n$ → **Sense** p<<n values $\{f_i(\underline{x})\}_{i=1}^{p}$ → ••• → **Reconstruct** $\underline{x}$ from $\{f_i(\underline{x})\}_{i=1}^{p}$ for later processing

Few Questions Must be Answered:

- ❑ What functions $f_i(\underline{x})$ to use ?

  Linear Projections $f_i(\underline{x})=\underline{x}^T\underline{v}_i$ are appealing due to their simplicity.

- ❑ How many measurements to take (p) ?

  Depends on the complexity (degrees of freedom) of $\underline{x}$.

- ❑ How can we reconstruct $\underline{x}$ from the measured values ?

  Depends on the model we assume on $\underline{x}$.

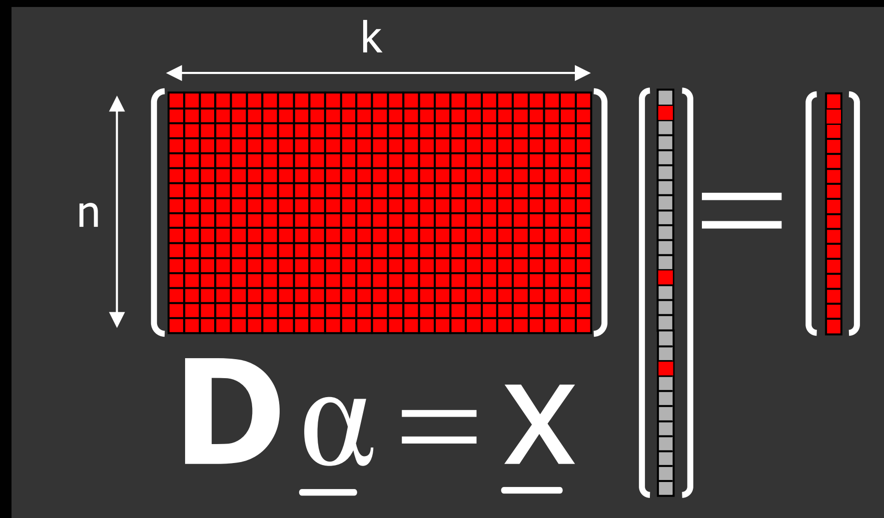
Sparsity and Redundancy

- At the heart of CS, lies a specific choice of a model for our signals.
- The model used in recent work that studies CS is based on

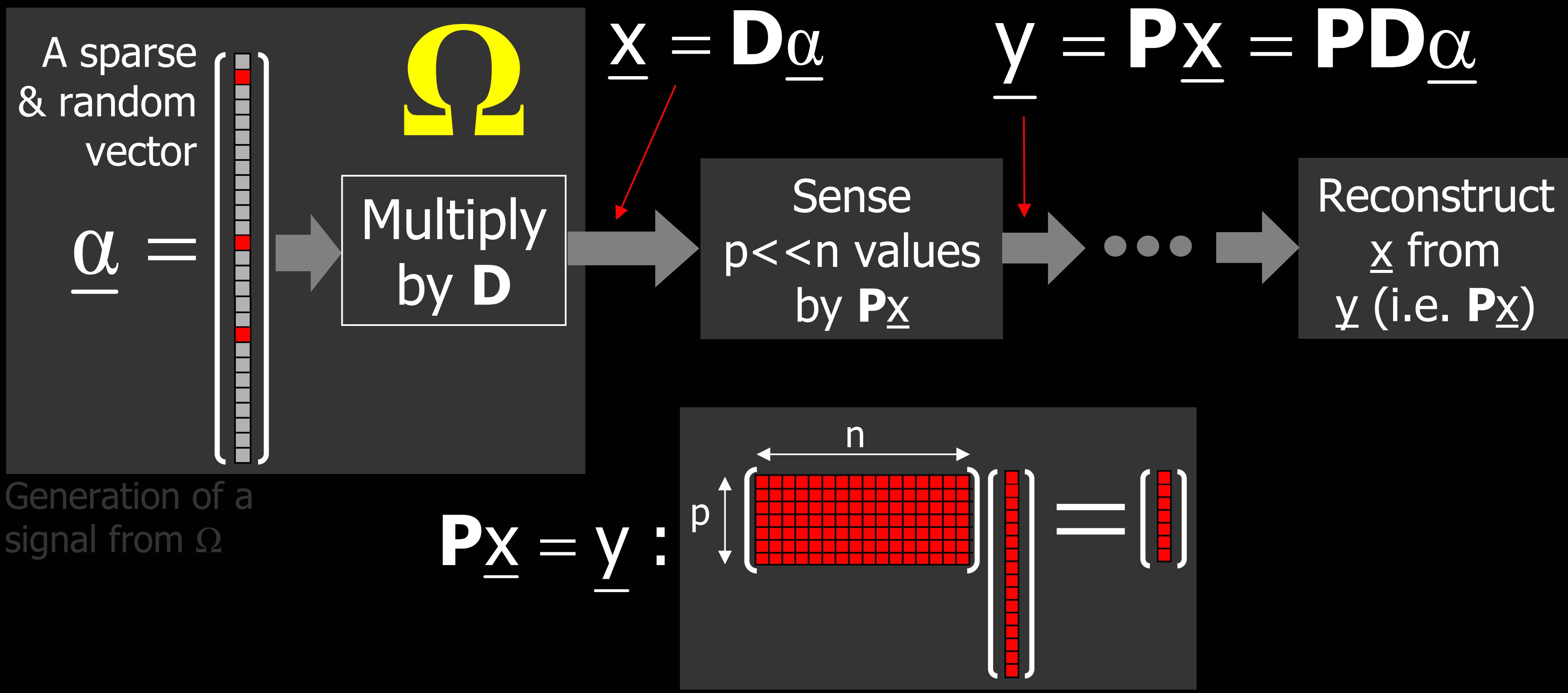
Sparse and Redundant Representations

- We assume that each of our signals could be represented as a linear combination of few columns of a matrix (dictionary) **D**.
- Thus, we define the family of signals, Ω, to be such that

$$\forall \underline{x} \in \Omega,\ \exists \underline{\alpha} \mid \mathbf{D}\underline{\alpha} = \underline{x}\ \&\ \|\underline{\alpha}\|_0 \le T \ll n$$



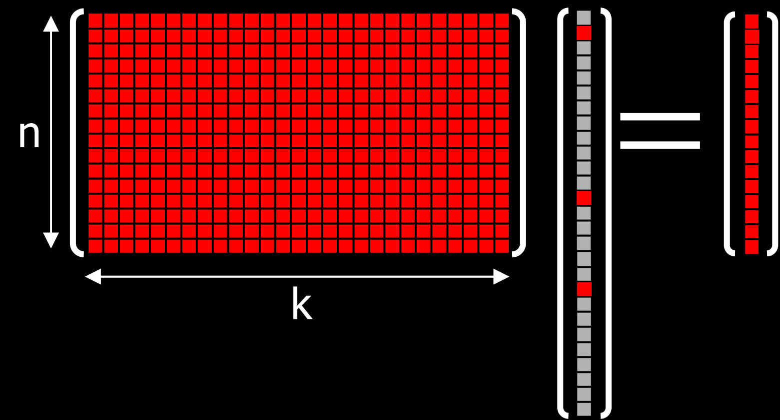
Compressed–Sensing



The Obtained Linear System

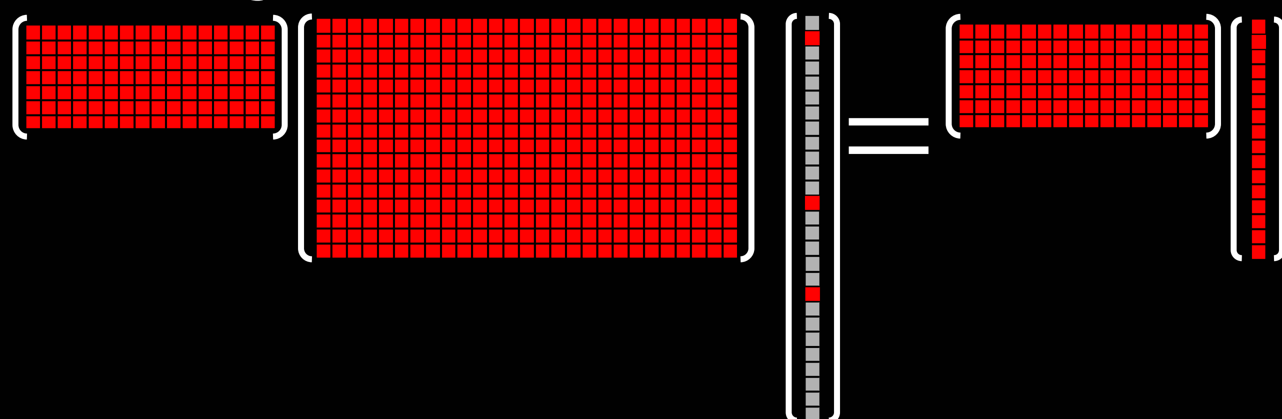
Instead of the original system

$$\mathbf{D}\underline{\alpha} = \underline{x} :$$



we get a new one …

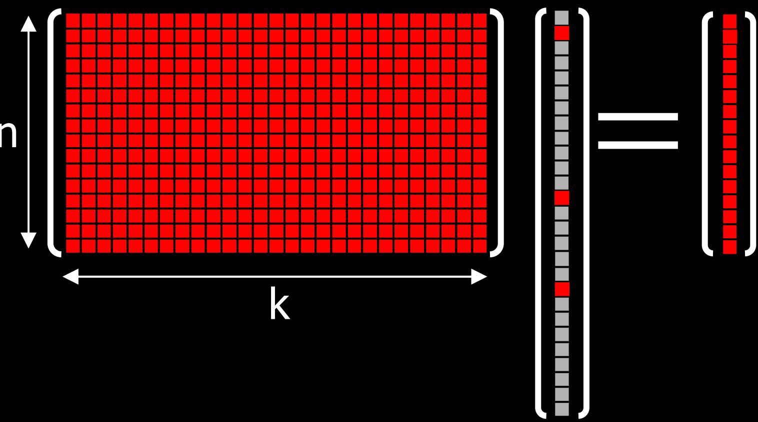
$$\mathbf{P}\mathbf{D}\underline{\alpha} = \mathbf{P}\underline{x} :$$



The Obtained Linear System

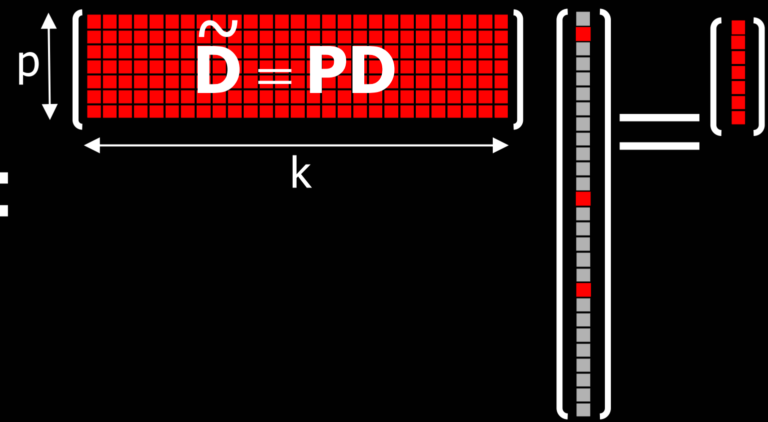
Instead of the original system

$\mathbf{D}\underline{\alpha} = \underline{x}$:



which is really …

$\tilde{\mathbf{D}}\underline{\alpha} = \underline{y}$:



Reconstructing $\underline{x}$

$$\tilde{\mathbf{D}} = \mathbf{PD}$$

$$\tilde{\mathbf{D}}\underline{\alpha} = \underline{y}$$

Known

The following are known:

- **D** – part of the model for Ω;
- **P** – our choice of projections;
- $\underline{y}$ – the set of p measurements; and
- $\underline{\alpha}$ is expected to be sparse !!!

1. Solve $\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{Argmin}} \left\|\underline{\alpha}\right\|_0 \quad \text{s.t.} \quad \tilde{\mathbf{D}}\underline{\alpha} = \underline{y}$

2. Set $\hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$

In Practice?

Is there a practical reconstruction algorithm?

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\mathrm{Argmin}} \left\| \underline{\alpha} \right\|_0 \quad \text{s.t.} \quad \tilde{\mathbf{D}}\underline{\alpha} = \underline{y}$$

1. Replace $\left\| \underline{\alpha} \right\|_0 \rightarrow \left\| \underline{\alpha} \right\|_1$: Basis Pursuit
2. Build $\underline{\alpha}$ greedily: Matching Pursuit

A General Rule* claims:

For signals in Ω, i.e.,
$\forall \underline{x} \in \Omega, \exists \underline{\alpha} \mid \mathbf{D}\underline{\alpha} = \underline{x} \ \& \left\| \underline{\alpha} \right\|_0 \le T << n$
If p>2T up to a constant (and a log factor!?), we get a perfect recovery of the signal with an overwhelming probability.

* See [Candes, Romberg, & Tao `04, Donoho `06, Candes 06`, Tsaig & Donoho `06].

Agenda

1. What is Compressed-Sensing (CS)?
   Little-bit of Background

2. The Choice of Projections for CS
   Criteria for Optimality

3. The Obtained Results
   Simulations

4. What Next?
   Conclusions & Open problems

What P to Use?

Reconstruction is performed by approximating the solution of

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\mathrm{Argmin}} \left\| \underline{\alpha} \right\|_0 \quad \text{s.t.} \quad \mathbf{PD}\underline{\alpha} = \underline{y}$$

The success (or failure) of this approach depends strongly on a proper choice of **P**

- The choices recommended in the literature include several random options:
  - o Gaussian IID entries;
  - o Binary (± 1) IID entries; and
  - o Fourier IID sampling.
- An important incentive to choose one of the above choices is the (relative) ease with which theoretical guarantee theorems can be developed.
- Could we propose a better way to design the projection directions **P**?

Better P – Phase I

How about choosing **P** that optimizes the recovery performance on a group of "training" signals?

Here is how such thing could be done:

Step 1: Gather a large set of example signals having sparse representation $\Rightarrow$ $\left\{\underline{\alpha}_i^o\right\}_{i=1}^{L} \Leftrightarrow \left\{\underline{x}_i^o = \mathbf{D}\underline{\alpha}_i^o\right\}_{i=1}^{L}$

Step 2: Solve

$$\underset{\mathbf{P}}{\text{Argmin}} \sum_{i=1}^{L} \left\|\hat{\underline{\alpha}}_i - \underline{\alpha}_i^o\right\|_2^2 \quad \text{s.t.} \quad \hat{\underline{\alpha}}_i = \underset{\underline{\alpha}}{\text{Argmin}} \left\|\underline{\alpha}\right\|_0 \quad \text{s.t.} \quad \mathbf{PD}\underline{\alpha} = \underline{y} = \mathbf{PD}\underline{\alpha}_i^o$$

This is done by a pursuit algorithm

Better P – Phase I

How about choosing **P** that optimizes the recovery performance on a group of "training" signals?

Here is how such thing could be done:

Step 1: Ga… set of ex… mp… e signals … $\}_{i=1} \Leftrightarrow \left\{\underline{x}_i^{o} = \mathbf{D}\underline{\alpha}_i^{o}\right\}_{i=1}^{L}$
having sparse …

TOO
COMPLICATED

Step 2: Solv… $\mathbf{P}$ … $_{i=1}\|\underline{\alpha}_i - \ldots\|_2$ … Arg … $\underline{\alpha}$ … $\mathbf{P}\mathbf{D}\underline{\alpha} = y = \mathbf{P}\mathbf{D}\underline{\alpha}_i^{o}$

This is a bi-level optimization problem, hard to minimize w.r.t. **P**

Better P – Phase II

Design **P** such that pursuit methods **are likely** to perform better – i.e., optimize performance indirectly.

! But … How?

Here is a Theorem we could rely on:

Suppose that $\underline{x}_0=\mathbf{D}\underline{\alpha}_0$, where

$$\|\underline{\alpha}_0\|_0 < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$$

μ is a property of the matrix **D**, describing its columns' dependence

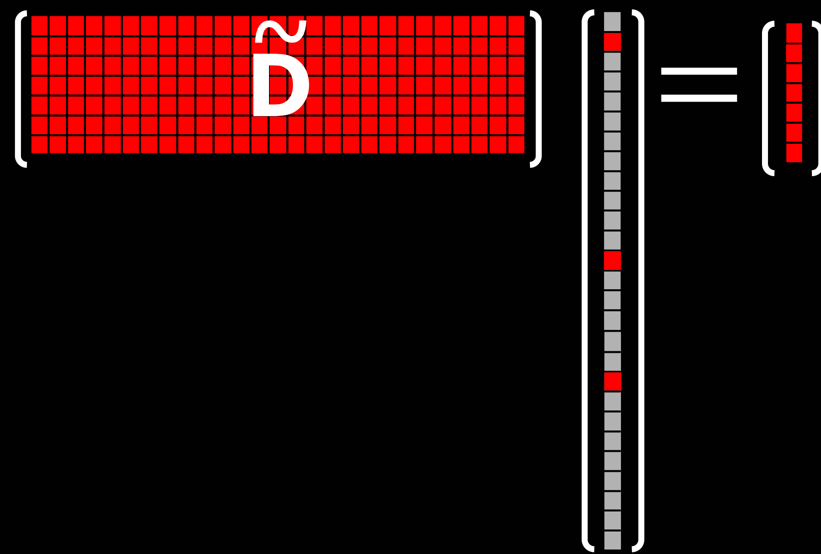
Then,

- The vector $\underline{\alpha}_0$ is the solution of $\min_{\underline{\alpha}} \|\underline{\alpha}\|_0$ s.t. $\underline{x}_0 = \mathbf{D}\underline{\alpha}$,
- Both pursuit algorithms are guaranteed to find it.

[Donoho & Elad `02]
[Gribonval & Nielsen `04]
[Tropp `04]

How Can this Be Used?

Adapting the above Theorem, the implication:



When seeking the sparsest solution to the system

$$\tilde{\mathbf{D}}\underline{\alpha} = \underline{y},$$

if the solution satisfies:

$$\|\underline{\alpha}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\tilde{\mathbf{D}})}\right)$$

then pursuit methods succeed, and find this (sparsest) solution.

The Rationale

Generation of a signal from Ω

A sparse & random vector

$$\underline{\alpha}_0 = \begin{bmatrix} \vdots \end{bmatrix}$$

Multiply by **D**

$$\underline{x} = \mathbf{D}\underline{\alpha}_0$$

Sense p<<n values by $\mathbf{P}\underline{x}$

$$\underline{y} = \mathbf{P}\underline{x} = \tilde{\mathbf{D}}\underline{\alpha}_0$$

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\mathrm{Arg\,min}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \tilde{\mathbf{D}}\underline{\alpha} = \underline{y}$$

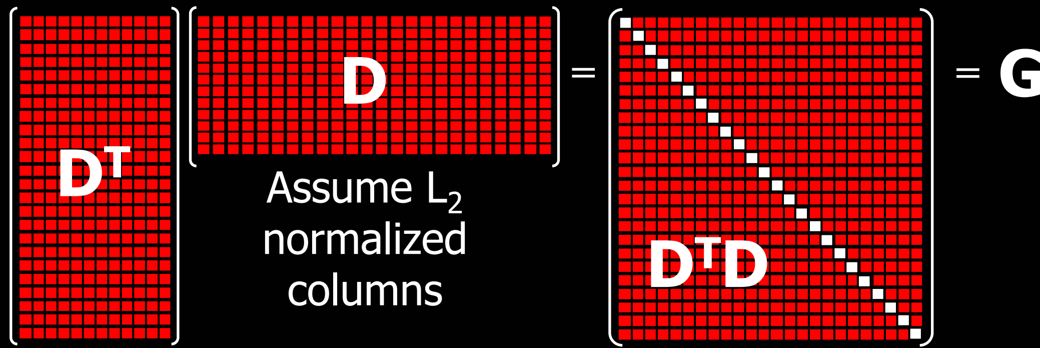
$$\underline{\alpha}_0 = \hat{\underline{\alpha}} ?$$

Yes, provided that $\|\underline{\alpha}_0\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\tilde{\mathbf{D}})}\right)$.

We could design **P** to lead to smallest possible measure μ

Lets Talk about μ

- Compute the Gram matrix

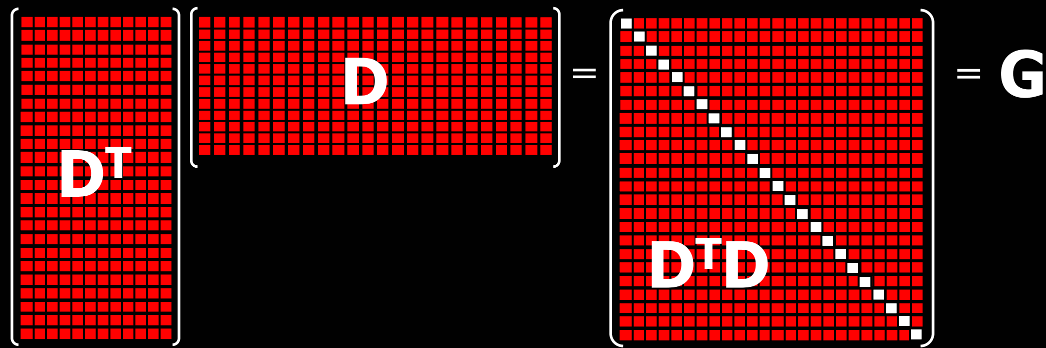


- The Mutual Coherence $\mu(\mathbf{D})$ is the largest off-diagonal entry in absolute value.

- Minimizing $\mu(\mathbf{PD})$: finding **P** such that the worst-possible pair of columns in **PD** are as distant as possible – a well-defined (but not too easy) problem.

- The above is possible but … worthless for our needs!
  μ is a "worst-case" measure – it does not reveal the average performance of pursuit methods.

Better P – Phase III (and Last)



We propose to use an average measure, taking into account all entries in **G** that are above a pre-specified threshold, t:

$$\mu_t(\mathbf{D}) = \frac{\sum_{i,j=1,i\neq j}^{k} \left(\left|G[i,j]\right| \ge t\right)\cdot \left|G[i,j]\right|}{\sum_{i,j=1,i\neq j}^{k} \left(\left|G[i,j]\right| \ge t\right)}$$

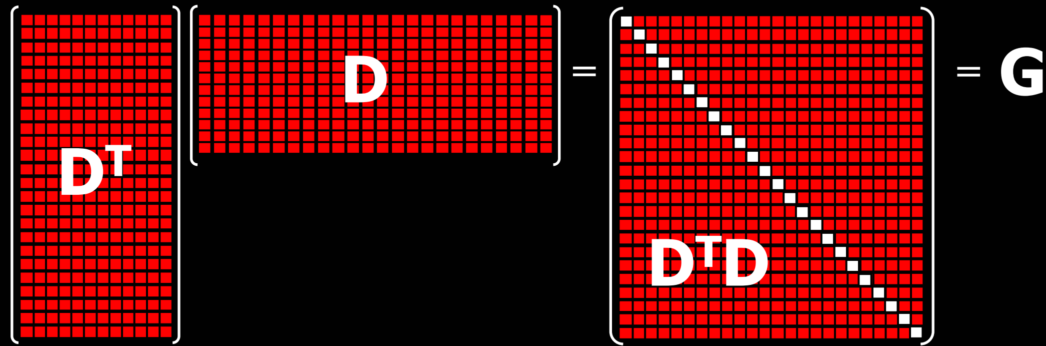
$t=0$

$t=\mu(\mathbf{D})$

$\mu_t(\mathbf{D})$ is an average of the off-diagonal entries of $|\mathbf{G}|$

$\mu_t(\mathbf{D})=\mu(\mathbf{D})$, i.e., it becomes the largest entry in $|\mathbf{G}|$

Better P – Phase III (and Last)



We propose to use an average measure, taking into account all entries in **G** that are above a pre-specified threshold, t:

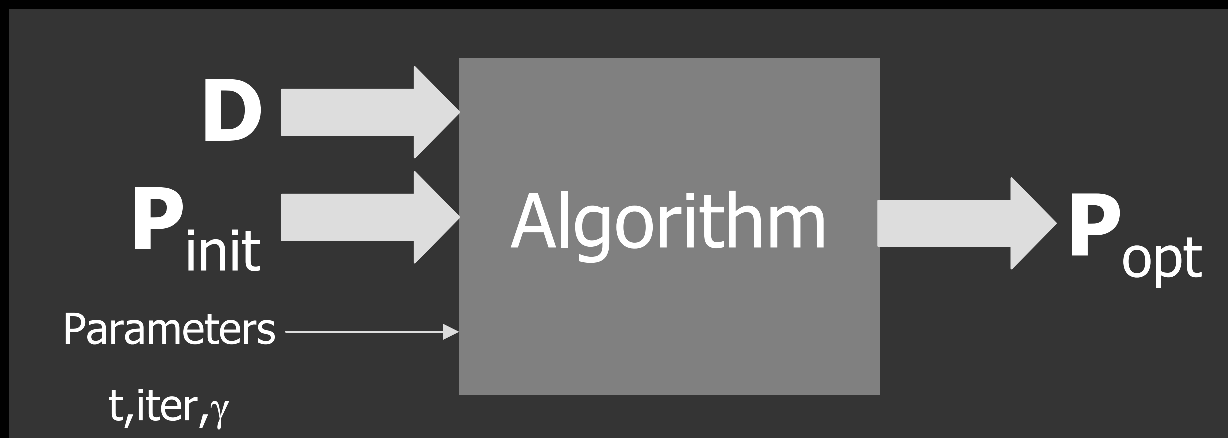
Instead of working with a fixed threshold t,
we can also work with t% representing an average
of the top t% entries in |**G**|

So, Our Goal Now is to …

Minimize the average coherence μ_t measure w.r.t. **P**

$$\mathbf{P}_{opt} = \underset{\mathbf{P}}{\mathrm{Argmin}}\ \mu_t(\mathbf{PD})$$

We need a numerical algorithm to do it



The Involved Forces

Our Goal: $$\mathbf{P}_{opt} = \underset{\mathbf{P}}{\mathrm{Arg\,min}}\ \mu_t(\mathbf{PD})$$

Defining $\mathbf{G} = (\mathbf{PD})^T(\mathbf{PD})$, we know that the following properties must hold true:

1. The rank of **G** must be p,
2. The square-root of **G** should be factorized to **PD**,
3. Some entries in **G** should be as small as possible.

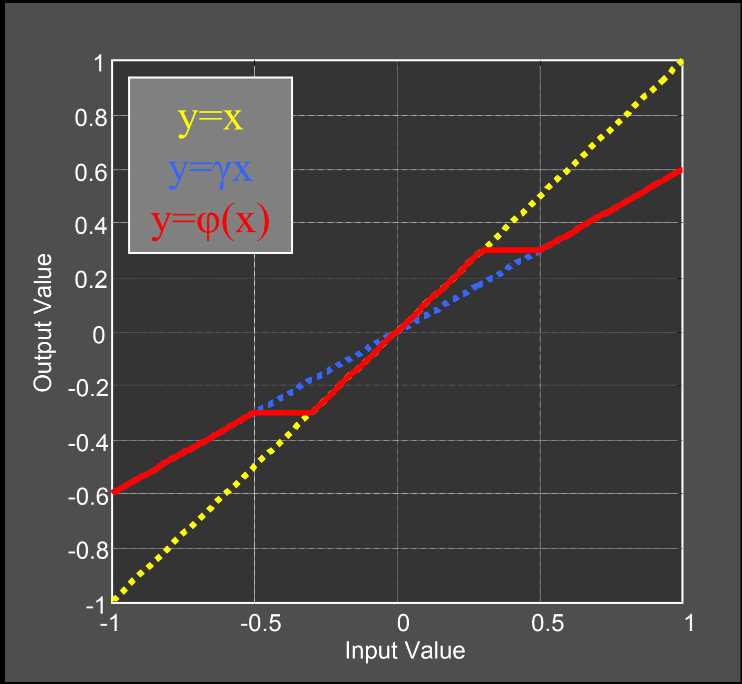
- We propose an algorithm that projects iteratively onto each of the above constraints, getting gradually a better **P**.
- Closely related to the work in [Tropp, Dhillon, Heath, Strohmer, `05].

The Numerical Algorithm

Set k=0 & $\mathbf{P}_0=\mathbf{P}_{init}$

↓

Normalize columns of $\tilde{\mathbf{D}}_k = \mathbf{P}_k\mathbf{D}$ → Compute the Gram $\mathbf{G}_k = \tilde{\mathbf{D}}_k^T\tilde{\mathbf{D}}_k$ → Shrink entries (γ) $\hat{\mathbf{G}}_k = \varphi(\mathbf{G}_k)$ →



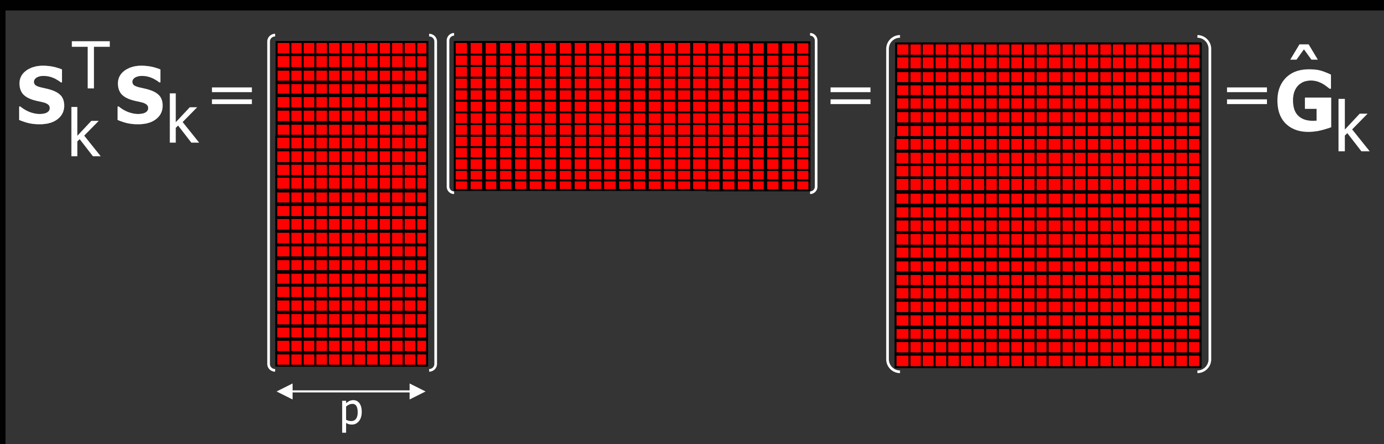
The Numerical Algorithm

Set k=0 & $\mathbf{P}_0=\mathbf{P}_{init}$

↓

Normalize columns of $\tilde{\mathbf{D}}_k = \mathbf{P}_k\mathbf{D}$ → Compute the Gram $\mathbf{G}_k = \tilde{\mathbf{D}}_k^T\tilde{\mathbf{D}}_k$ → Shrink entries (γ) $\hat{\mathbf{G}}_k = \varphi(\mathbf{G}_k)$ → Reduce rank to p: SVD($\hat{\mathbf{G}}_k$) → Compute SQRT $\hat{\mathbf{G}}_k = \mathbf{S}_k^T\mathbf{S}_k$

↓



The Numerical Algorithm

Set k=0 & $\mathbf{P}_0=\mathbf{P}_{init}$

Normalize columns of $\tilde{\mathbf{D}}_k = \mathbf{P}_k\mathbf{D}$

Compute the Gram $\mathbf{G}_k = \tilde{\mathbf{D}}_k^T\tilde{\mathbf{D}}_k$

Shrink entries (γ) $\hat{\mathbf{G}}_k = \varphi(\mathbf{G}_k)$

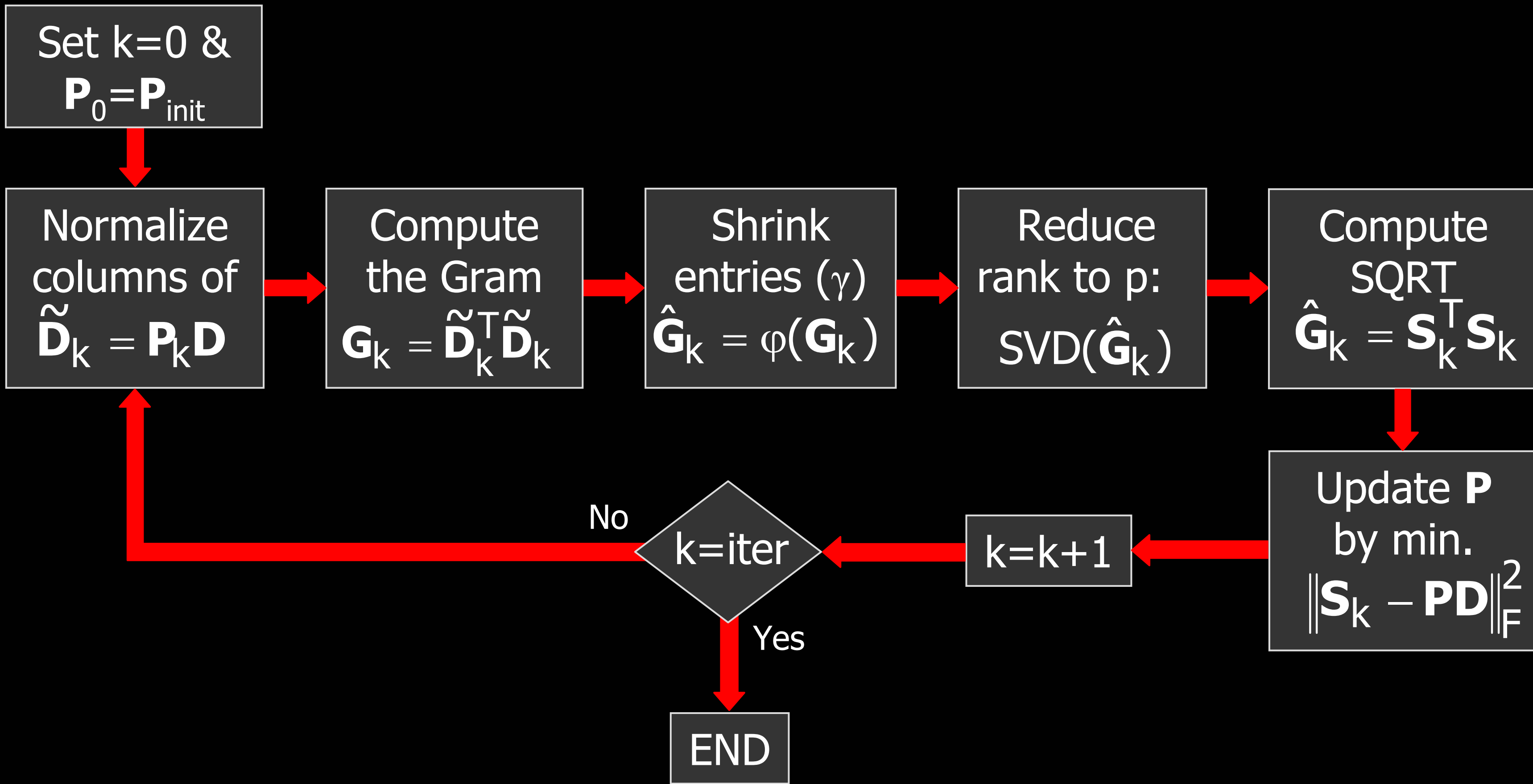
Reduce rank to p: SVD($\hat{\mathbf{G}}_k$)

Compute SQRT $\hat{\mathbf{G}}_k = \mathbf{S}_k^T\mathbf{S}_k$

Update $\mathbf{P}$ by min. $\left\|\mathbf{S}_k - \mathbf{P}\mathbf{D}\right\|_F^2$

$$\left\| \left[\mathbf{P}\right]\left[\mathbf{D}\right] - \left[\mathbf{S}_k\right] \right\|_F^2$$

The Numerical Algorithm



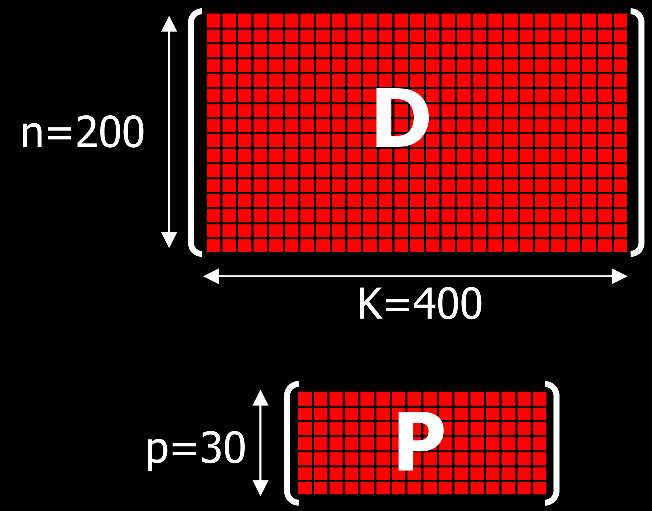
Agenda

1. What is Compressed-Sensing (CS)?
   Little-bit of Background
2. The Choice of Projections for CS
   Criteria for Optimality
3. The Obtained Results
   Simulations
4. What Next?
   Conclusions & Open problems

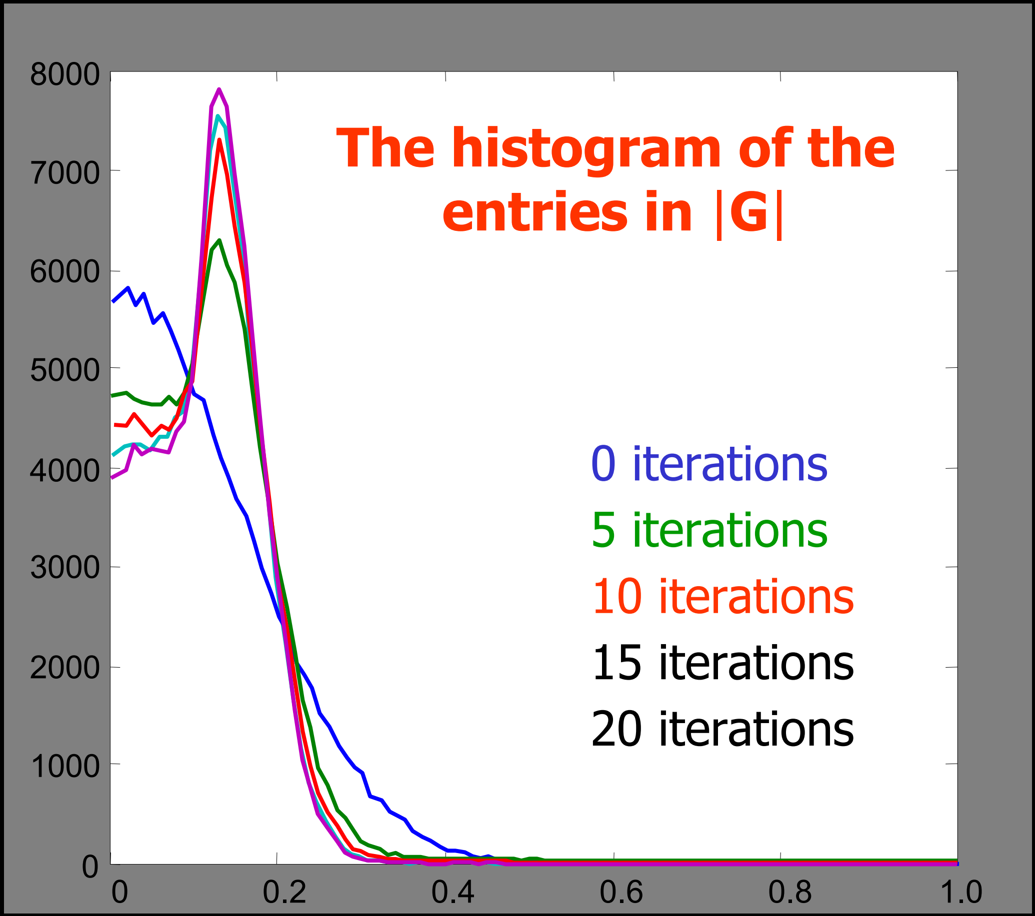
A Core Experiment

Details:

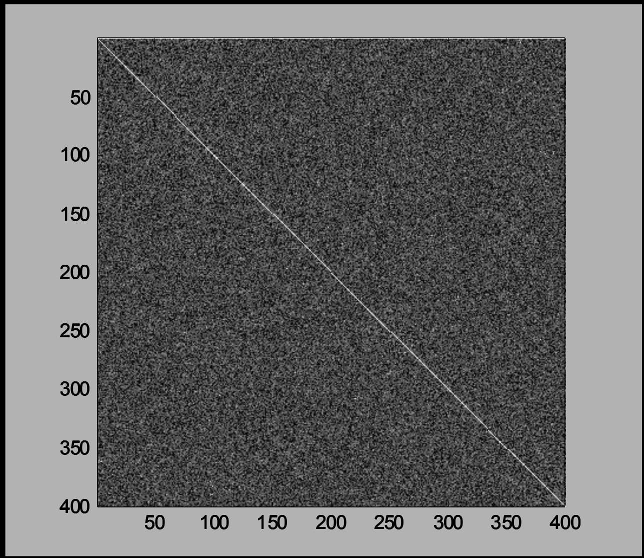
- We build a random matrix **D** of size 200×400 with Gaussian zero mean IID entries.
- We find a 30×200 matrix **P** according to the above-described algorithm.
- We present the obtained results, and the convergence behavior.
- Parameters: t=20%, γ=0.5.



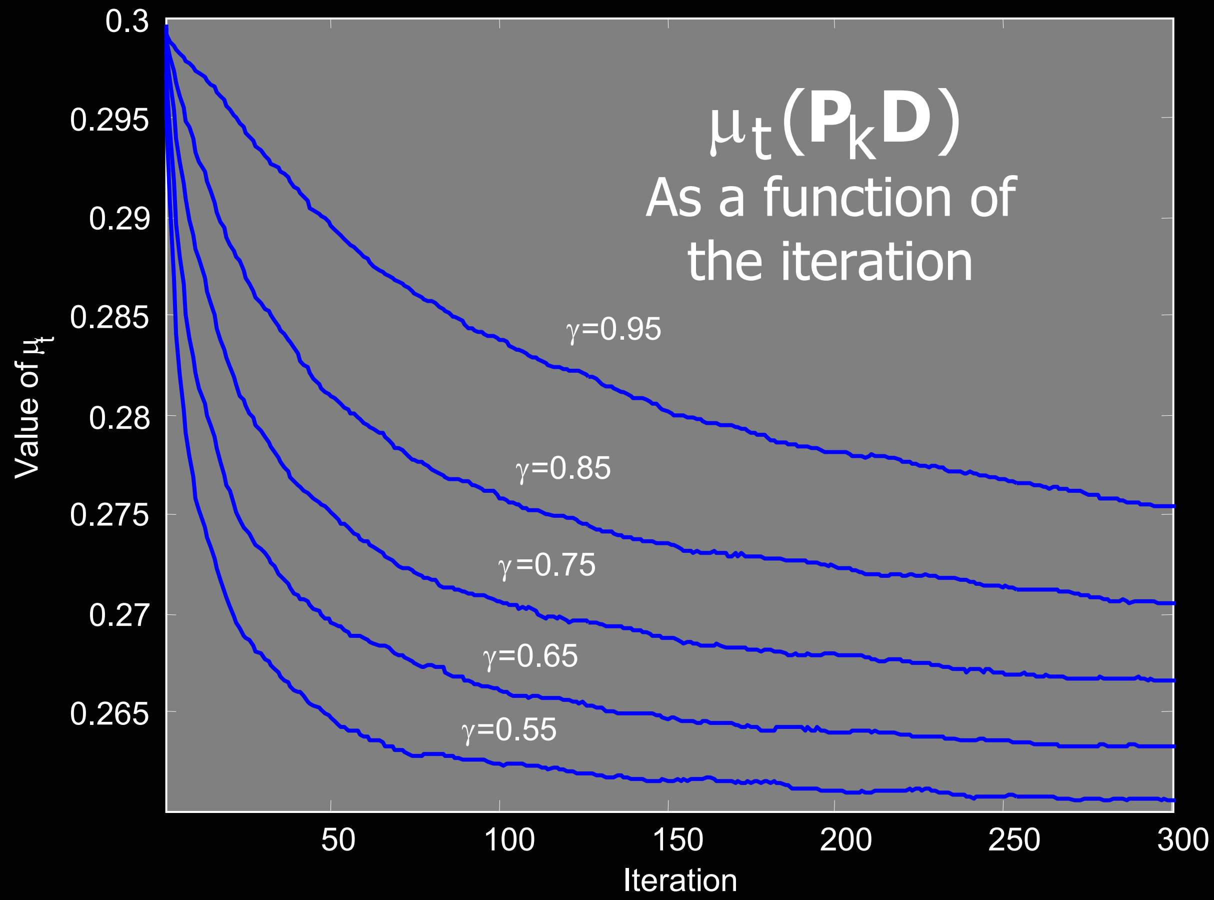
Entries in |G|



The matrices |G| as a function of the iteration



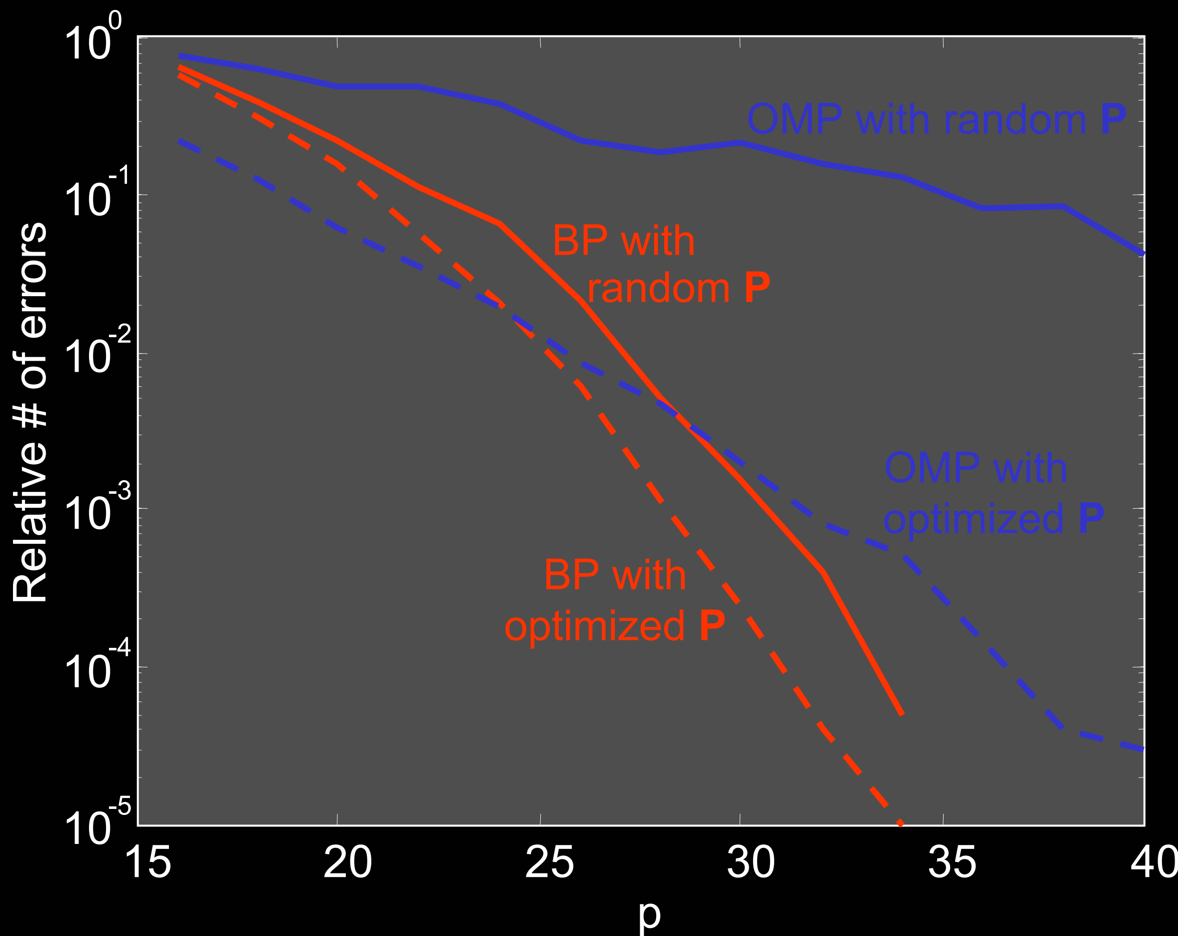
Convergence



CS Performance (1): Effect of P

Details:

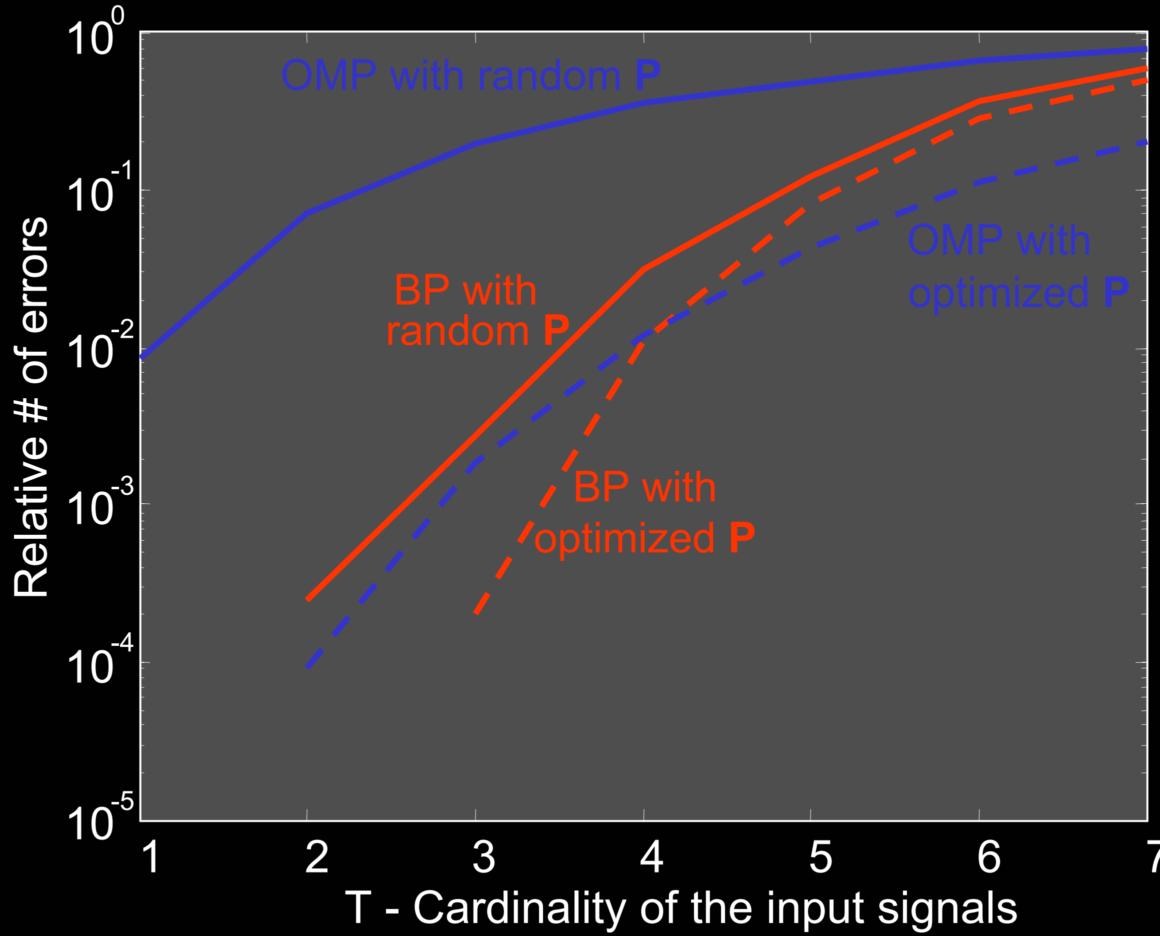
- **D**: 80×120 random.
- 100,000 signal examples to test on. Each have T=4 non-zeros in their (random) $\underline{\alpha}$.
- p varied in the range [16,40].
- CS performance: before and after optimized **P**, for both BP and OMP.
- Optimization: t=20%, γ=0.95, 1000 iterations.
- Show average results over 10 experiments.



CS Performance (2): Effect of T

Details:

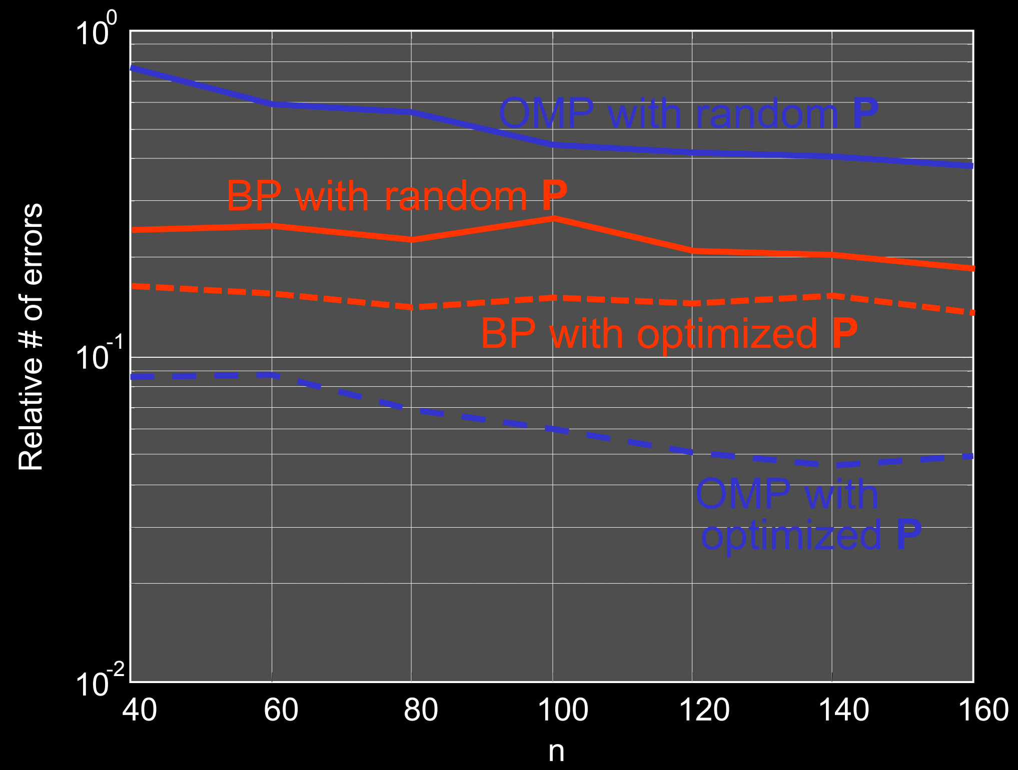
- **D**: 80×120 random.
- 100,000 signal examples to test on. Each have T (varies) non-zeros in their $\underline{\alpha}$.
- T varied in the range [1,7].
- Fixed p=25.
- CS performance: before and after optimized **P** for both BP and OMP.
- Optimization: t=20%, γ=0.95, 1000 iterations.
- Show average results over 10 experiments.



CS Performance (3): Effect of n

Details:

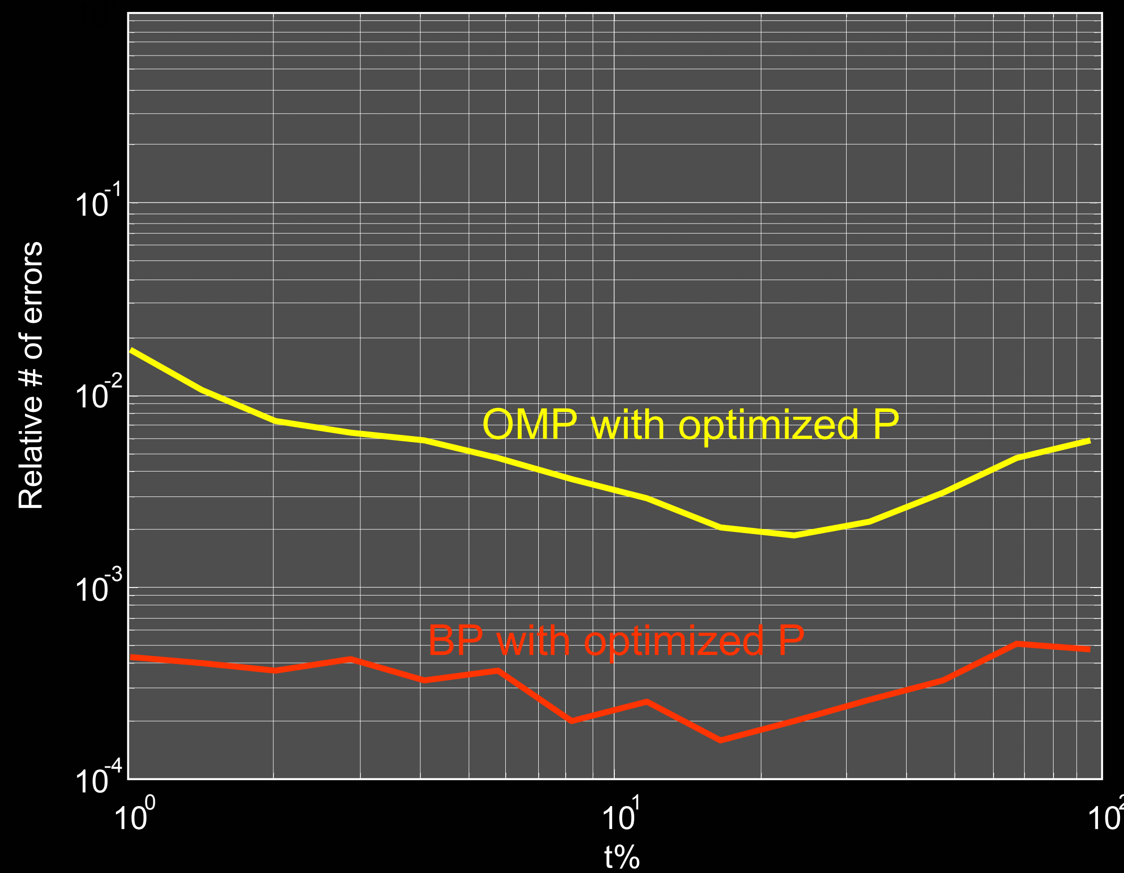
- **D**: n×1.5n with n varied in the range [40,160] (random).
- 100,000 signal examples to test on. Each have T (varies) non-zeros in their $\underline{\alpha}$.
- T=n/20.
- p=n/4.
- CS performance: before and after optimized **P** – OMP and BP.
- Optimization: t=20%, γ=0.95, 1000 iterations.
- …



CS Performance (4):Effect of t%

Details:

- **D**: 80×120 random.
- 100,000 signal examples to test on. Each have T=4 non-zeros in their $\underline{\alpha}$.
- P=30, T=4: fixed.
- CS performance: before and after optimized **P**.
- Optimization: t=varies in the range [1,100]%, γ=0.95, 1000 iterations.
- …



Agenda

1. What is Compressed-Sensing (CS)?
   Little-bit of Background
2. The Choice of Projections for CS
   Criteria for Optimality
3. The Obtained Results
   Simulations
4. What Next?
   Conclusions & Open problems

Conclusions

Compressed-Sensing (CS)
An emerging & exciting research field; combining sensing and compression of signals

Sparse Representation over a dictionary **D** is the model used for deriving "sampling theorems" in CS work.

Linear projection P is used for generating the compressed & sensed measurements in CS

How should **P** be chosen ?

Ideally, we want to choose **P** to **Optimize CS Performance** for the class of signals in mind

Since it is too difficult, we propose to **Optimize w.r.t. a Different Measure**, indirectly affecting CS performance

The measure we propose is an Average Mutual Coherence. We present an algorithm for its minimization

Various Experiments that we provide show that the damn thing works rather well.

More Work is required in order to further improve this method in various ways – see next slide

Future Work

- Could we optimize the true performance of BP/OMP using the Bi-Level optimization we have shown?
- How about optimizing **P** w.r.t. a simplified pursuit algorithm like simple thresholding?
- What to do when the dimensions involved are huge? For example, when using the curvelets, contourlets, or steerable-wavelet transforms. In these cases the proposed algorithm is impractical!
- Could we find a clear theoretical way to tie the proposed measure (average coherence) with pursuit performance?
- Maybe there is a better (yet simple) alternative measure. What is it?