SPARSITY IN SIGNAL AND IMAGE PROCESSING

Compressive Sensing: Theory and applications

Nicolas Keriven, courtesy of Claude Petit and Aline Roumy

November 2023

Outline

About me

Nicolas KERIVEN

Researcher at CNRS, Rennes (Sirocco Team)

email: `nicolas.keriven@cnrs.fr`

Course schedule (tentative)

- Nov. 6, 3 hours
- Nov. 7, 3 hours
- Nov. 13, 3 hours
- Nov. 14, 3 hours

$\rightarrow$ Mix of lecture, exercise, computer lab each time

Course schedule (tentative)

- Nov. 6, 3 hours
- Nov. 7, 3 hours
- Nov. 13, 3 hours
- Nov. 14, 3 hours

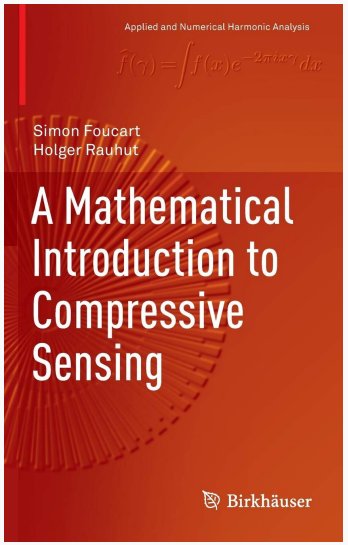
$\rightarrow$ Mix of lecture, exercise, computer lab each time

Tools

- Computer Lab: Python notebooks, either on your machine or on Google Colab
- here: `https://github.com/nkeriven/insaCS`
- We will use a lot G Peyré's Numerical Tours in Python

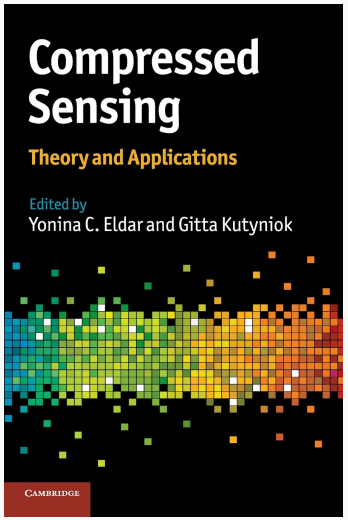
Course material

S. Foucart, H. Rauhut, A mathematical introduction to compressive sensing, Birkhaüser, 2013.



Course material

Compressed Sensing: Theory and Applications, Edited by Y.C. Eldar and G. Kutyniok, Cambridge University Press, 2012.



Introduction

Outline

What is compressive sensing?

Compressive sensing: a way to acquire (or sense or sample) and compress data.

What is compressive sensing?

Compressive sensing: a way to acquire (or sense or sample) and compress data.

Classical = sampling then compression

Compressive sensing = sampling **AND** compression

What is compressive sensing?

Compressive sensing: a way to acquire (or sense or sample) and compress data.

Classical = sampling then compression

Compressive sensing = sampling **AND** compression

Several names exist:

- compressed sensing
- compressed sampling
- compressive sampling
- **compressive sensing**. More accurate. Chosen in this course. The one of the reference book.

What is compressive sensing?

Compressive sensing: a way to acquire (or sense or sample) and compress data.

Classical = sampling then compression

Compressive sensing = sampling **AND** compression

Several names exist:

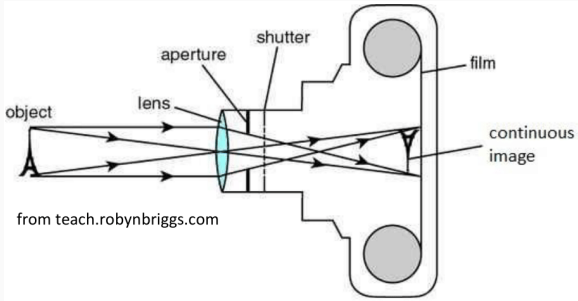
- compressed sensing
- compressed sampling
- compressive sampling
- **compressive sensing**. More accurate. Chosen in this course. The one of the reference book.

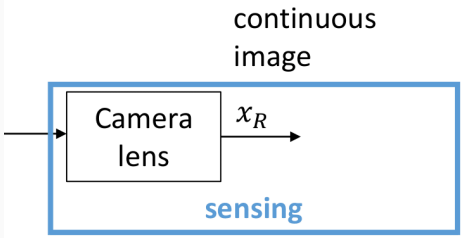
Is CS "old school"? It can seem that way, since the advent of learning-based systems. But the principles and notions are fundamental, and underpin even the most advanced Deep Neural Net.

Film camera

Classical digital acquisition: sampling then compression

Film camera: records images passing through the camera's lens.

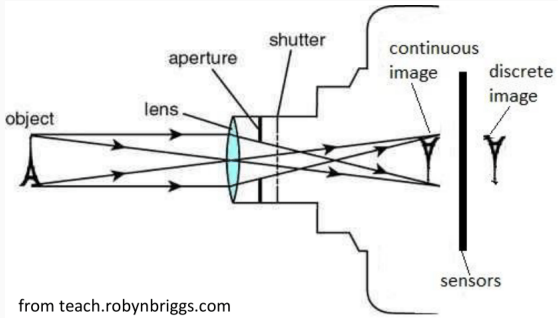




$$x_R : [0,1]^2 \to \mathbb{R}^3$$

Digital camera

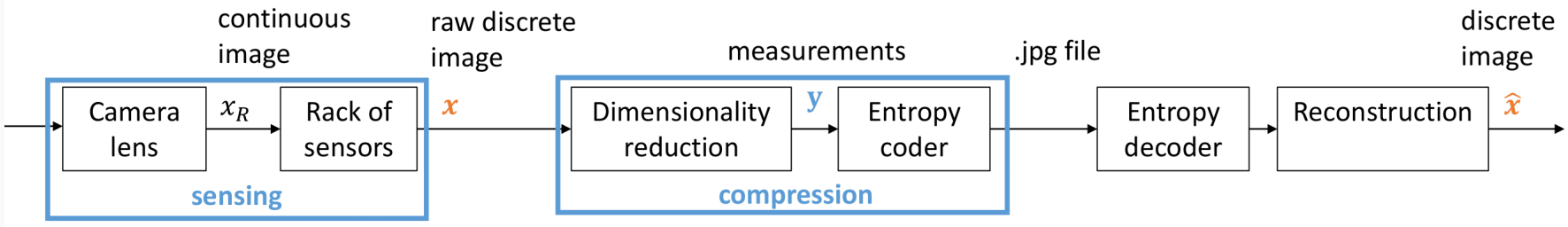
Digital camera: converts an image into **digital** data and **compress** it.



$$x_R : [0,1]^2 \to \mathbb{R}^3$$

$$x : \{1, N_a\} \times \{1, N_b\} \to \{0,255\}^3$$

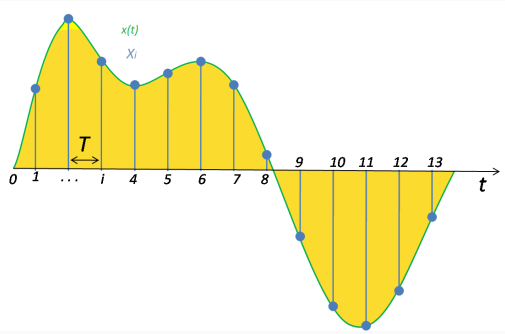
$$y : \{1, M_a\} \times \{1, M_b\} \to \{0,255\}^3$$



Questions related to Digital camera

Question related to sampling:

is it possible to recover a continuous signal from its sampled (discrete) version?



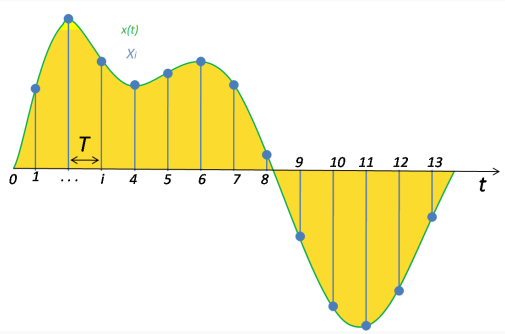
Wikipedia.

cf. course of Clément Elvira: Shannon-Nyquist theorem

Questions related to Digital camera

Question related to sampling:

is it possible to recover a continuous signal from its sampled (discrete) version?



Wikipedia.

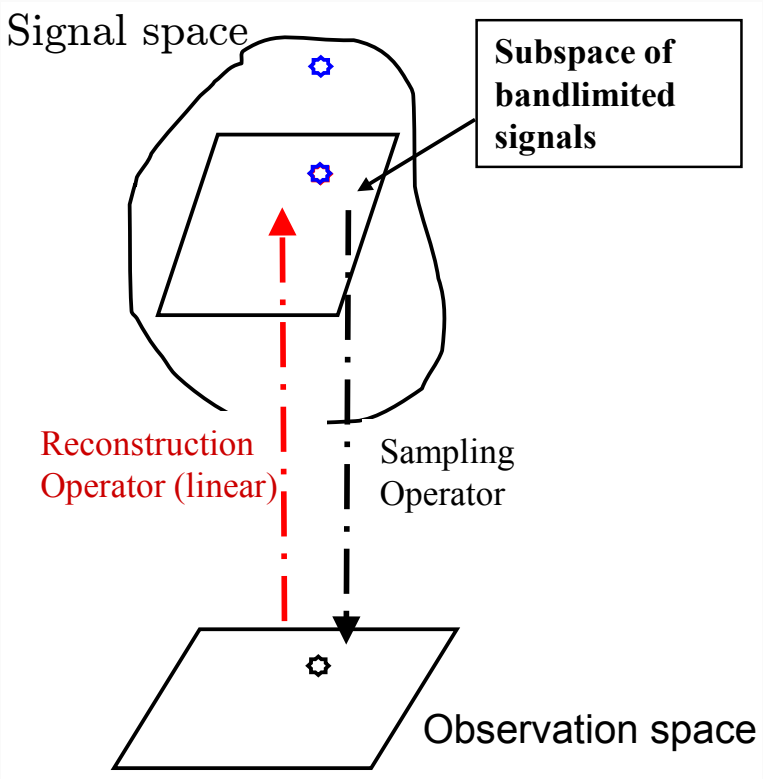
cf. course of Clément Elvira: Shannon-Nyquist theorem

Question related to compression:

is it possible to reduce the size of a *discrete* image? (... and recover it)

Sampling: (1) optimal sampling rate

Nyquist–Shannon sampling theorem: "Exact reconstruction of a continuous-time signal from discrete samples is possible if the signal is bandlimited and the sampling frequency is greater than twice the highest frequency."

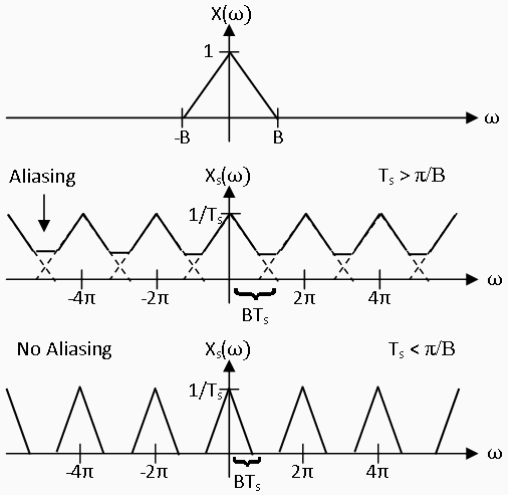


Mike Davies.

Sampling: (2) degradation if “slow” sampling

Sampling below the optimal rate introduces:

(1) aliasing

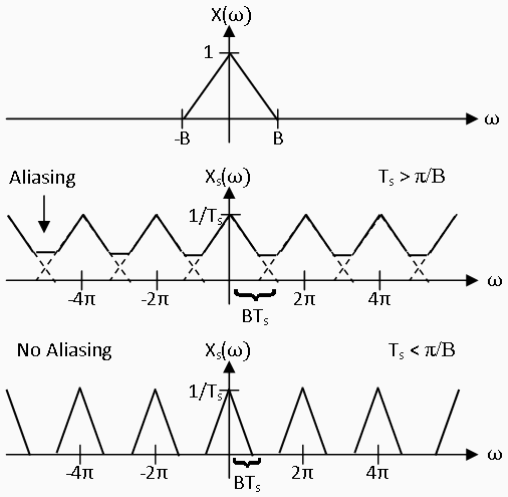


SVI.nl

Sampling: (2) degradation if “slow” sampling

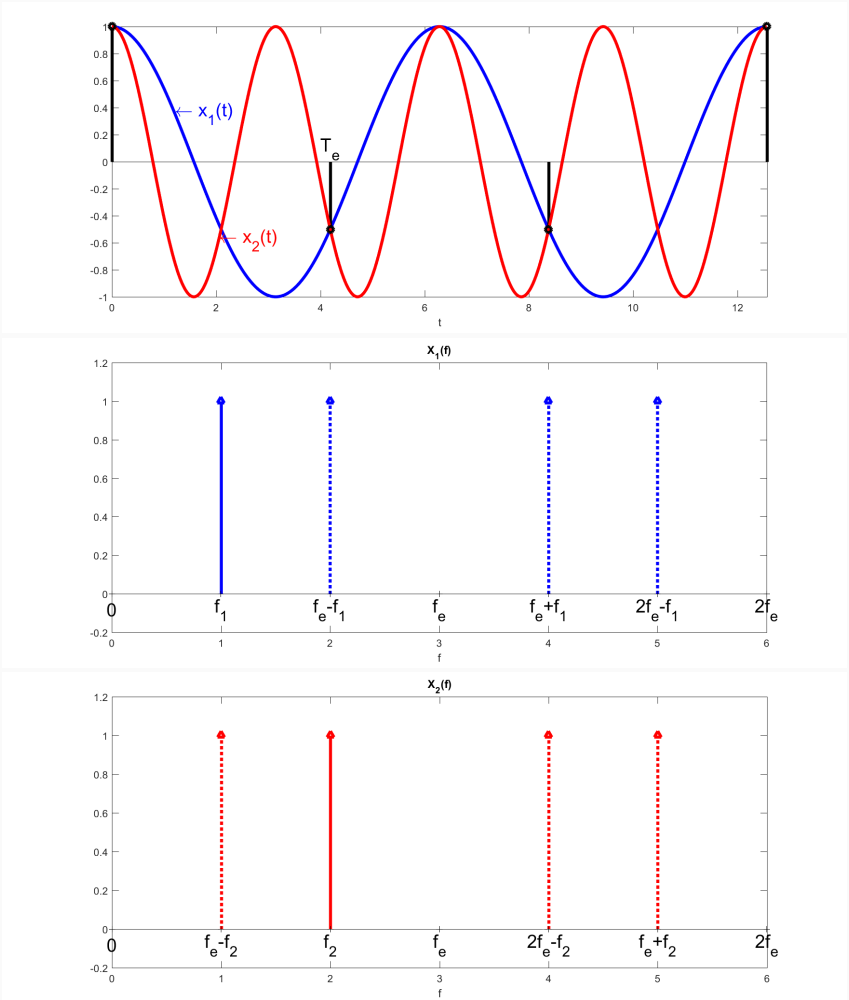
Sampling below the optimal rate introduces:

(1) aliasing



SVI.nl

(2) signal ambiguity



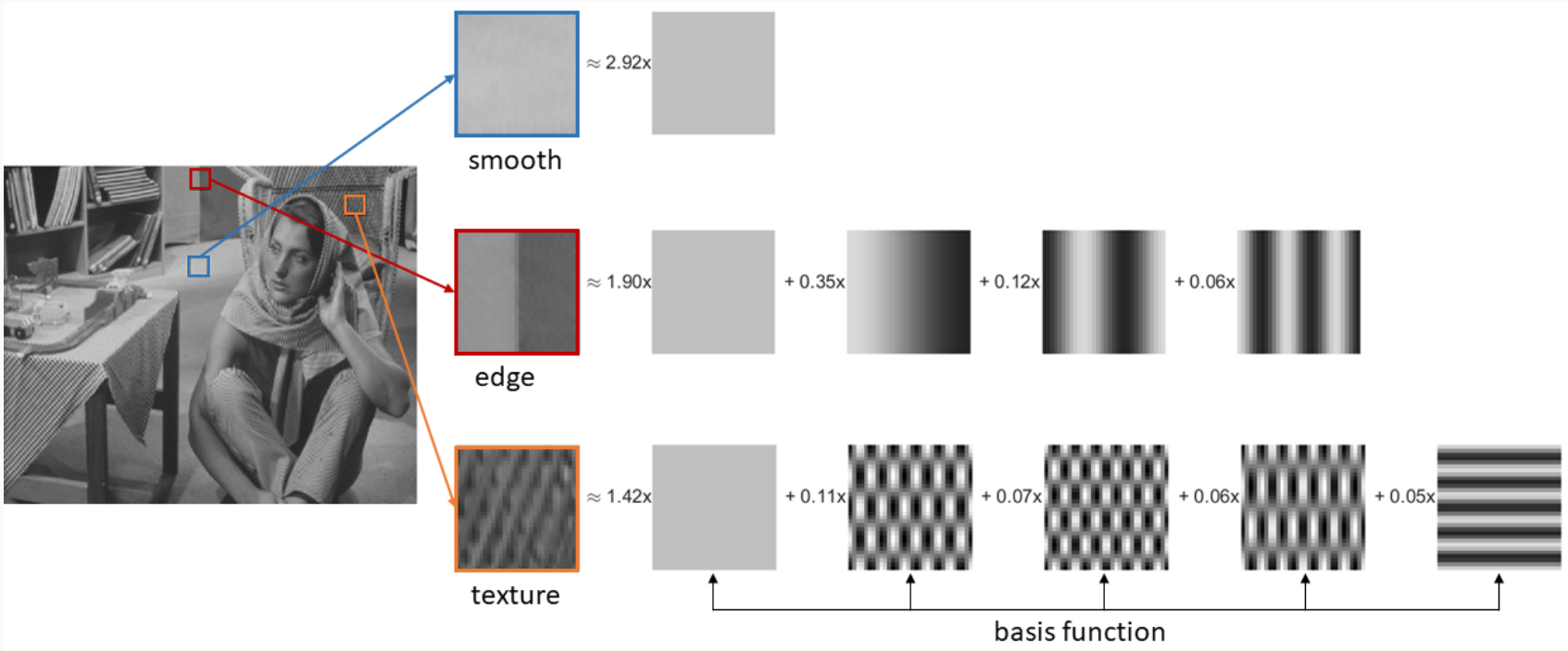
Compression: (1) image decomposition principle

Compression? Many ways. Example:

Compression: (1) image decomposition principle

Compression? Many ways. Example:

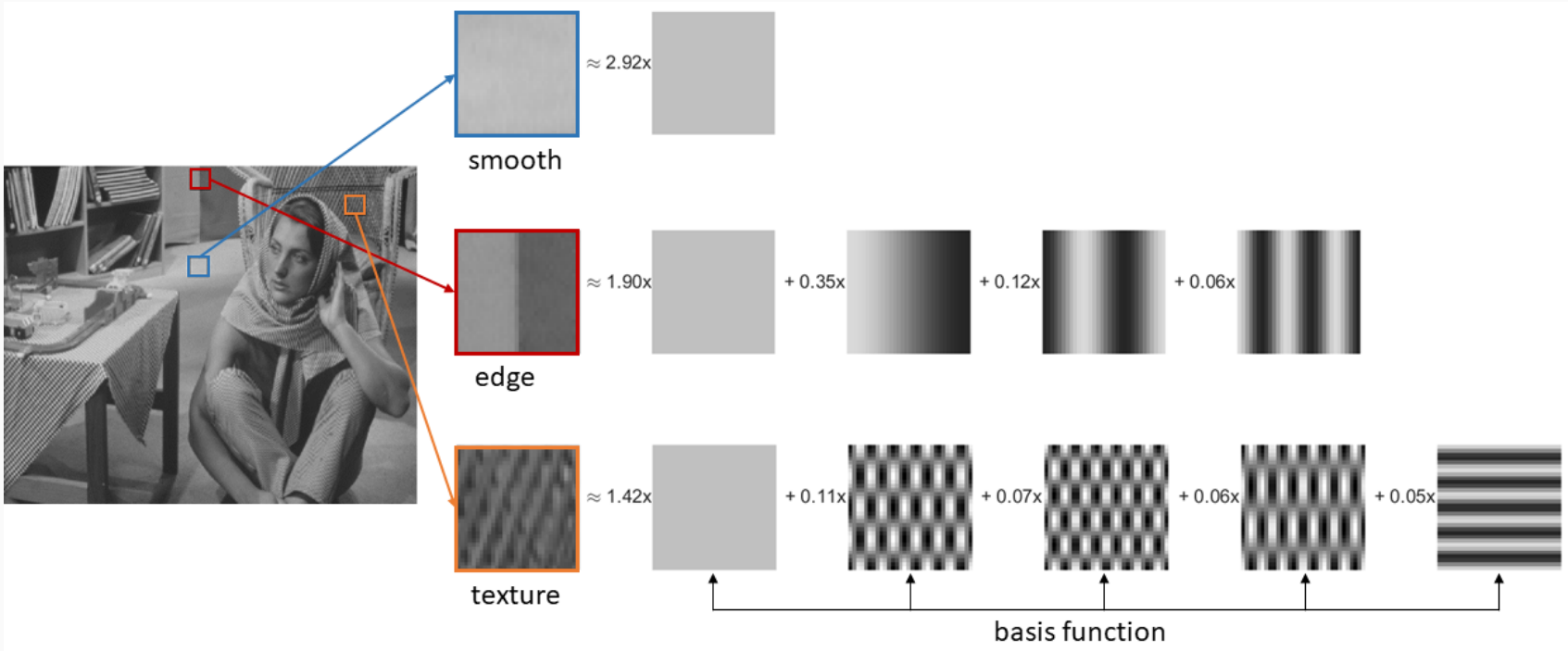
1. Split image into blocks of size $N_1 \times N_2$ each
2. Decompose each $N_1 \times N_2$ image block as basis functions:



Compression: (1) image decomposition principle

Compression? Many ways. Example:

1. Split image into blocks of size $N_1 \times N_2$ each
2. Decompose each $N_1 \times N_2$ image block as basis functions:



How to choose the basis functions? How to compute the coefficients?

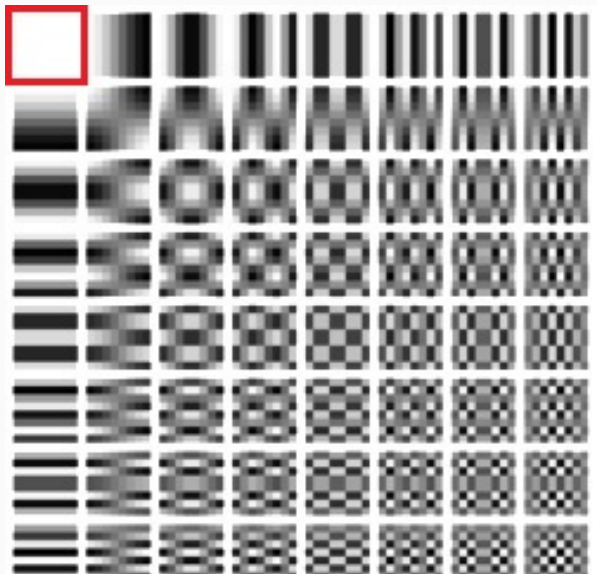
Compression: (2) image decomposition example

Popular method: 2D-discrete cosine transform (DCT) **(orthogonal basis)**

1. Split image into blocks of size $N_1 \times N_2$ each
2. For each $N_1 \times N_2$ image block (x_{n_1,n_2})
   compute the $N_1 \times N_2$ block of transformed image (c_{k_1,k_2}) with:

$$c_{k_1,k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \underbrace{\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]}_{\Phi_{n_1,n_2}(k_1,\,k_2)}$$

Example: 8x8 DCT transform
Top-left matrix is $(\Phi_{n_1,n_2}(k_1 = 0, k_2 = 0))_{n_1,n_2}$
$\rightarrow$ the whole transform is an *invertible* (orthogonal!) linear transform



Compression: (3) image decomposition result

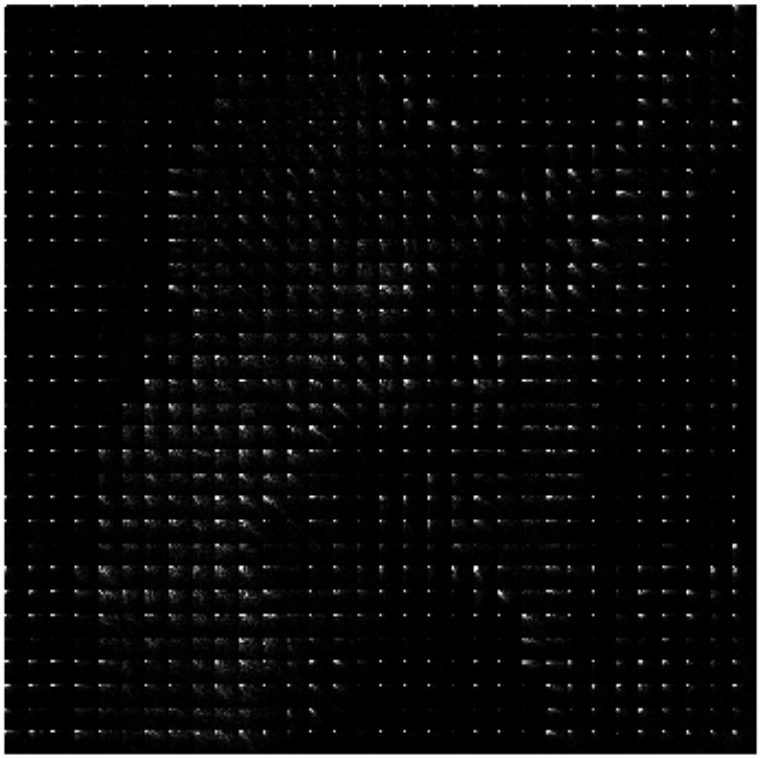
For now, we have transformed an $N_1 \times N_2$ image to an $N_1 \times N_2$ image!
What about compression...?

Compression: (3) image decomposition result

For now, we have transformed an $N_1 \times N_2$ image to an $N_1 \times N_2$ image!
What about compression...?

Left: image

Right: discrete cosine transform of image



Key concept: **few degrees of freedom in the transform domain** $\rightarrow$ **sparsity**

Compression: (4) dimensionality reduction with s-term approximation

1. Dimensionality reduction: keep the s coefficients c_s with largest absolute value
2. Reconstruction: $\hat{x} = \Phi^{-1} c_s$

Compression: (4) dimensionality reduction with s-term approximation

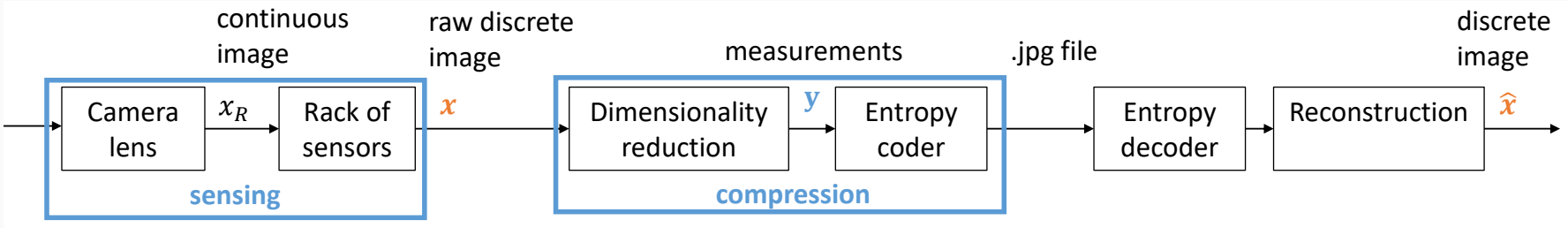
1. Dimensionality reduction: keep the s coefficients c_s with largest absolute value
2. Reconstruction: $\hat{x} = \Phi^{-1} c_s$

Left: 1% kept

Right:5% kept

Summary on classical sensing

Classical



Sampling raw discrete HD video

1920x1080=2.07 M pixels/image

25Hz: images/s,

12(=8+2+2) bits/pixel

$\rightarrow$ 0.6 Gbit/s

Compression

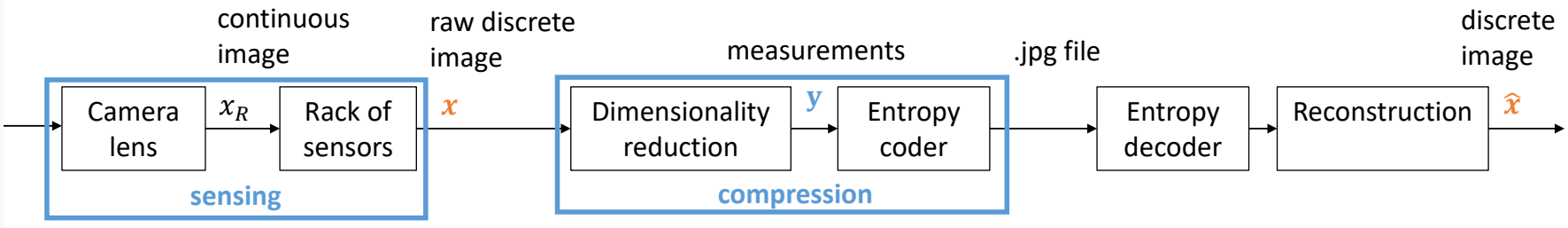
For instance, HEVC (2013)

0.6 Gbit/s $\rightarrow$ 2Mbit/s

compression ratio 300:1!!!

Classical vs compressive sensing

Classical



take lots of samples (Shannon-Nyquist),

throw most of the coefficients away

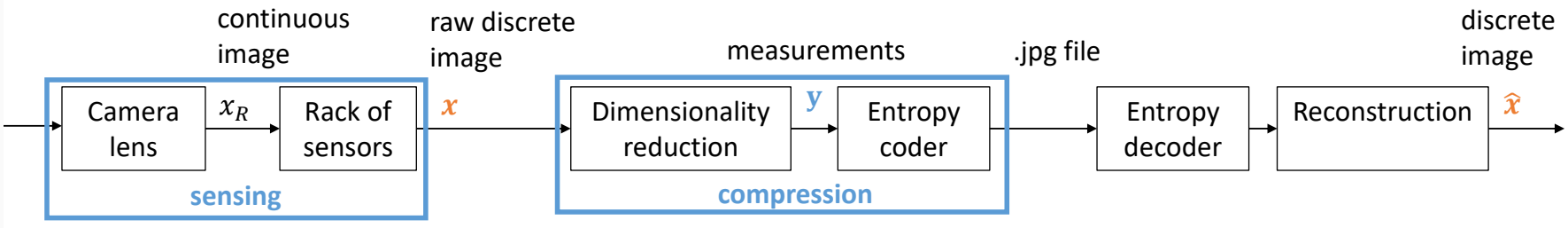
x: $[1, N_a] \times [1, N_b] \to \{0, 255\}^3$

y: $[1, M_a] \times [1, M_b] \to \{0, 255\}^3$

$(M_a M_b \ll N_a N_b)$

Classical vs compressive sensing

Classical



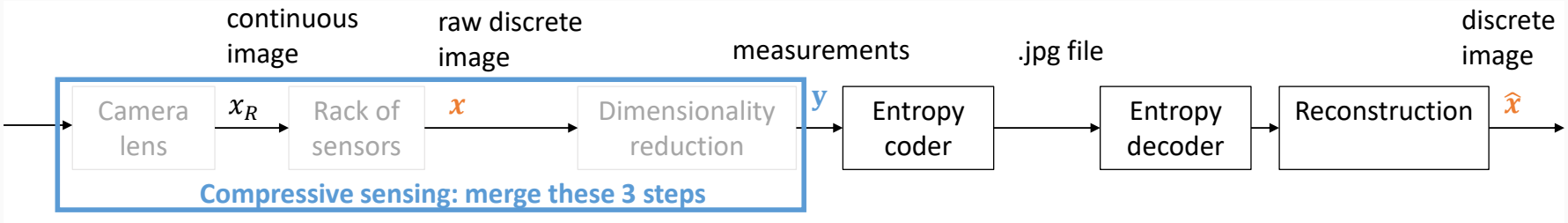
take lots of samples (Shannon-Nyquist),
throw most of the coefficients away

$x: [1, N_a] \times [1, N_b] \to \{0, 255\}^3$

$y: [1, M_a] \times [1, M_b] \to \{0, 255\}^3$

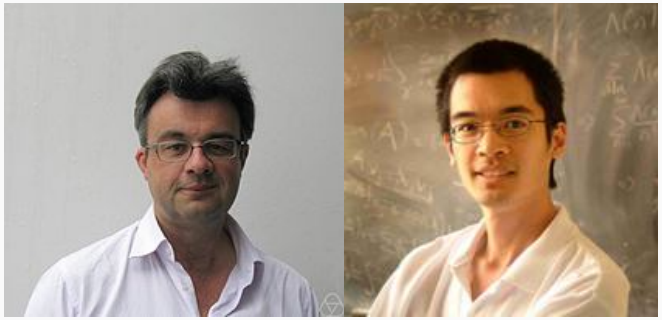
$(M_a M_b \ll N_a N_b)$

Compressive sensing: can we acquire less data in the first place?
Compressive sensing: and still recover $\hat{x}$?



Can we sample signals at the "Information Rate"?

Yes, we can!



Wikipedia.

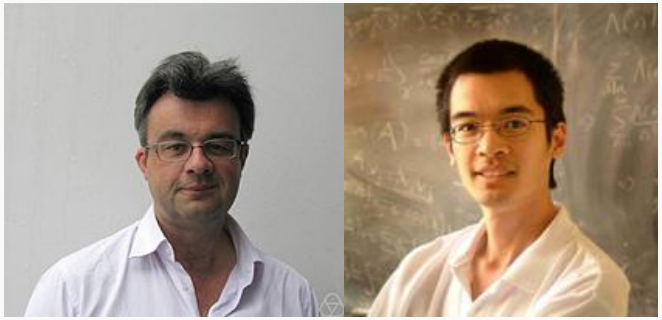
Wikipedia.

E. J. Candes and T. Tao, 2005
"Decoding by linear programming"

D. L. Donoho, 2006
"Compressed sensing"

Can we sample signals at the "Information Rate"?

Yes, we can!



Wikipedia.

Wikipedia.

E. J. Candes and T. Tao, 2005
"Decoding by linear programming"

D. L. Donoho, 2006
"Compressed sensing"

$\rightarrow$ We can sense/sample signals directly proportionally to their **complexity/sparsity**

A bit of history...

1795 Prony's method: estimation of parameters of a small number of complex exponentials

A bit of history...

1795 Prony's method: estimation of parameters of a small number of complex exponentials

1900s Carathéodory show that *any* linear combination of k sinusoids are uniquely determined by *any* $2k$ samples (independent of the frequency!)

A bit of history...

1795 Prony's method: estimation of parameters of a small number of complex exponentials

1900s Carathéodory show that *any* linear combination of k sinusoids are uniquely determined by *any* $2k$ samples (independent of the frequency!)

1936 Beurling: recovering a signal from a *partial* observation of its Fourier transform

A bit of history...

1795 Prony's method: estimation of parameters of a small number of complex exponentials

1900s Carathéodory show that *any* linear combination of k sinusoids are uniquely determined by *any* $2k$ samples (independent of the frequency!)

1936 Beurling: recovering a signal from a *partial* observation of its Fourier transform

2000s Blu, Marzino, Vetterli: generalization to certain signals, recovering k signals from $2k$ samples

A bit of history...

1795 Prony's method: estimation of parameters of a small number of complex exponentials

1900s Carathéodory show that *any* linear combination of k sinusoids are uniquely determined by *any* $2k$ samples (independent of the frequency!)

1936 Beurling: recovering a signal from a *partial* observation of its Fourier transform

2000s Blu, Marzino, Vetterli: generalization to certain signals, recovering k signals from $2k$ samples

2005 - 2008 Candès, Tao, Romberg, Donoho: Compressive Sensing

More difference between Classical sampling and CS

Classical

Compressive Sensing

More difference between Classical sampling and CS

Classical

- sample *then* compress

Compressive Sensing

- directly gather *less* samples

More difference between Classical sampling and CS

Classical

- sample *then* compress
- infinite-length, continuous signals

Compressive Sensing

- directly gather *less* samples
- Finite-dimensional signals

More difference between Classical sampling and CS

Classical

- sample *then* compress
- infinite-length, continuous signals
- sampling pointwise in time

Compressive Sensing

- directly gather *less* samples
- Finite-dimensional signals
- inner-product of the entire signal with a few vectors

More difference between Classical sampling and CS

Classical

- sample *then* compress
- infinite-length, continuous signals
- sampling pointwise in time
- recovery is "simple" interpolation

Compressive Sensing

- directly gather *less* samples
- Finite-dimensional signals
- inner-product of the entire signal with a few vectors
- recovery is complex, non-linear

Compressive Sensing in a nutshell

To recover an s-sparse signal in dimension n, take $m \gtrsim s \log(n/s)$ measurements with a random, dense sensing operator.

Outline

Norms

Definition (l_p-norm)

The **l_p-norm** of $x \in \mathbb{R}^n$, $p > 1$ is defined as

$$||x||_p = \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} & p \in [1, \infty) \\ \max_i |x_i| & p = \infty \end{cases}$$

If $p < 1$, definition still valid, but triangle inequality not satisfied

$\Rightarrow$ **quasi-norm**.

Norms

Definition (l_p-norm)

The **l_p-norm** of $x \in \mathbb{R}^n$, $p > 1$ is defined as

$$||x||_p = \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} & p \in [1, \infty) \\ \max_i |x_i| & p = \infty \end{cases}$$

If $p < 1$, definition still valid, but triangle inequality not satisfied

$\Rightarrow$ **quasi-norm**.

Definition (inner product)

$$\langle x, z \rangle = z^T x = \sum_{i=1}^{n} x_i z_i$$

See textbook F&R for extension to $\mathbb{C}^n$.

Definition (support and l_0-norm)

The **support** of a vector x is the index set of its non-zero entries, i.e.

$$\text{supp}\,(x) = \{j \in [n] : x_j \neq 0\}, \text{ where } [n] = \{1, 2, ..., n\}$$

The **l_0-norm** of x is defined as

$$||x||_0 = \text{ card } (\text{ supp } (x))$$

$||x||_0$ counts the number of non-zero entries of x.

$||.||_0$ is **not even a quasi-norm**: it is not homogeneous $\|\lambda x\|_0 \neq |\lambda|\|x\|_0$

Sparsity definition

Definition (s-sparse)

A signal $x \in \mathbb{R}^n$ is said to be s-sparse if it has at most s non-zero entries, i.e. $||x||_0 \leq s$.

Sparsity definition

Definition (s-sparse)

A signal $x \in \mathbb{R}^n$ is said to be s-sparse if it has at most s non-zero entries, i.e. $||x||_0 \leq s$.

Definition (Σ_s)

We define Σ_s as the set containing all s-sparse signals, i.e. $\Sigma_s = \{x \in \mathbb{R}^n : ||x||_0 \leq s\}$.

Sparsity definition

Definition (s-sparse)

A signal $x \in \mathbb{R}^n$ is said to be s-sparse if it has at most s non-zero entries, i.e. $||x||_0 \leq s$.

Definition (Σ_s)

We define Σ_s as the set containing all s-sparse signals, i.e. $\Sigma_s = \{x \in \mathbb{R}^n : ||x||_0 \leq s\}$.

Note 1: Sparsity is a highly nonlinear model (Σ_s is not a linear subspace: it is a union of subspaces)

Sparsity definition

Definition (s-sparse)

A signal $x \in \mathbb{R}^n$ is said to be s-sparse if it has at most s non-zero entries, i.e. $||x||_0 \leq s$.

Definition (Σ_s)

We define Σ_s as the set containing all s-sparse signals, i.e. $\Sigma_s = \{x \in \mathbb{R}^n : ||x||_0 \leq s\}$.

Note 1: Sparsity is a highly nonlinear model (Σ_s is not a linear subspace: it is a union of subspaces)

Note 2: in many practical cases, x is not sparse itself, but it has a sparse representation in some basis Φ. We still say that x is s-sparse, with the understanding that we can write $x = \Phi u$, and $||u||_0 \leq s$.

Approximate sparsity

- A sparse signal can be represented exactly giving the positions and values of its s nonzero components

Approximate sparsity

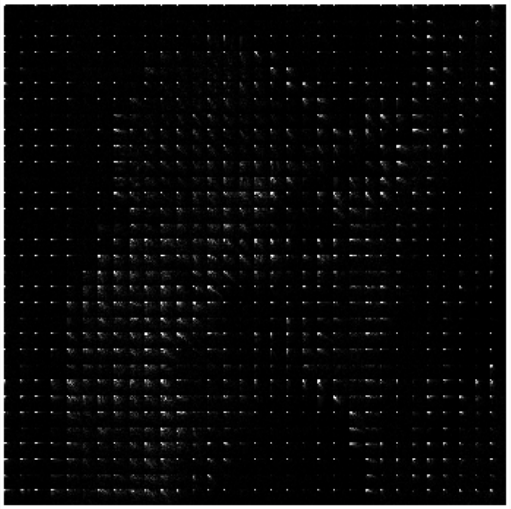
- A sparse signal can be represented exactly giving the positions and values of its s nonzero components
- Real-world signals are rarely exactly sparse.

Approximate sparsity

- A sparse signal can be represented exactly giving the positions and values of its s nonzero components
- Real-world signals are rarely exactly sparse.
  - generalize the definition from "sparse" to "compressible" signals,

Approximate sparsity

- A sparse signal can be represented exactly giving the positions and values of its s nonzero components
- Real-world signals are rarely exactly sparse.
  - generalize the definition from "sparse" to "compressible" signals,
  - describe the representation error, i.e. the error incurred representing just s components of the signal.



Best s-term approximation

The best s-term approximation picks the s components that minimize the representation error

Definition (best s-term approximation)

For $p > 0$, the l_p-error incurred by the best s-term approximation to a vector $x \in \mathbb{R}^n$ is given by

$$\sigma_s(x)_p = \min_{\hat{x} \in \Sigma_s} ||x - \hat{x}||_p$$

Best s-term approximation

The best s-term approximation picks the s components that minimize the representation error

Definition (best s-term approximation)

For $p > 0$, the l_p-error incurred by the best s-term approximation to a vector $x \in \mathbb{R}^n$ is given by

$$\sigma_s(x)_p = \min_{\hat{x} \in \Sigma_s} ||x - \hat{x}||_p$$

- If $x \in \Sigma_s$, then $\sigma_s(x)_p = 0$ for any p.
- $\hat{x}$ is easy to compute: keep the s largest elements (in absolute value), set all others to zero.
- $\hat{x}$ does not depend on p, but $\sigma_s(x)_p$ does!

Compressible signal

Definition (compressible signal)

a signal $x \in \mathbb{R}^n$ is said to be compressible if the error of its best s-term approximation **"decays quickly in s"**.

Compressible signal

Definition (compressible signal)

a signal $x \in \mathbb{R}^n$ is said to be compressible if the error of its best s-term approximation **"decays quickly in s"**.

- Vague notion, several def possible eg

$$\exists C_1, q > 0 \text{ such that } |x_i| \leq C_1 i^{-q}$$

when the coefficients have been ordered $|x_1| \geq ... \geq |x_n|$.

Compressible signal

Definition (compressible signal)

a signal $x \in \mathbb{R}^n$ is said to be compressible if the error of its best s-term approximation **"decays quickly in s"**.

- Vague notion, several def possible eg

$$\exists C_1, q > 0 \text{ such that } |x_i| \leq C_1 i^{-q}$$

  when the coefficients have been ordered $|x_1| \geq ... \geq |x_n|$.

- always true for a *given* x, but one can define the set of "C, q-compressible" vectors.

Left: 1% kept

Right: 5% kept

Sparsity support

Suppose $x \in R^n$. Let $S \subset [n]$ and $S^c = [n] \backslash S$

- S: sparsity support of x, i.e. the locations of the nonzero coefficients of $x \rightarrow$ the **key** in recovering x!
- S^c: set of locations of the 0 coefficients
- S for compressible signal: set of locations of the s largest coefficients

Sparsity support

Suppose $x \in R^n$. Let $S \subset [n]$ and $S^c = [n] \backslash S$

- S: sparsity support of x, i.e. the locations of the nonzero coefficients of $x \rightarrow$ the **key** in recovering x!
- S^c: set of locations of the 0 coefficients
- S for compressible signal: set of locations of the s largest coefficients

Notation

- *x_S vector obtained by setting the entries of x indexed by S^c to 0.*
- *M_S matrix obtained by setting the* **columns** *of M to 0.*
- *Same notation to denote vectors/matrices where the elements/columns have been* **removed**, *instead of being set to 0. Clear from the context.*

$\rightarrow$ *For an s-sparse vector with support S,*

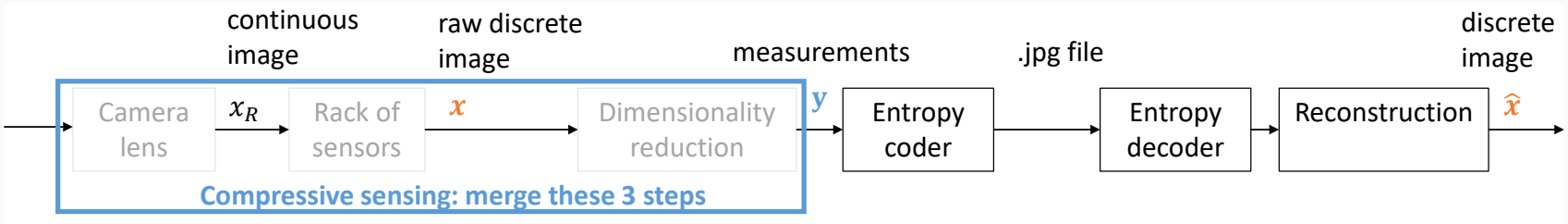
$$Mx = M_S x_S$$

Outline

Compressive sensing

Goal of Compressive sensing (CS):

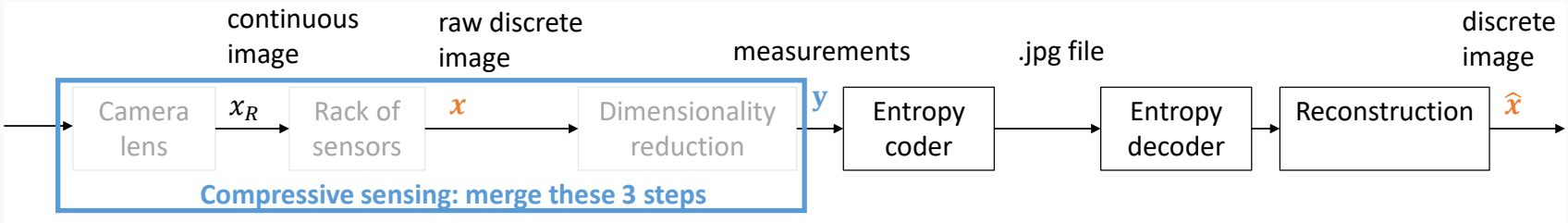
- achieve the same reconstruction quality on $\hat{x}$ as the best s-term approximation
- from the measurement y acquired with a nonadaptive encoder (... it must work for all s-sparse vectors)



Compressive sensing

Goal of Compressive sensing (CS):

- achieve the same reconstruction quality on $\hat{x}$ as the best s-term approximation
- from the measurement y acquired with a nonadaptive encoder (... it must work for all s-sparse vectors)



To achieve this, we need to

1. model the dependency between signal x and measurement y
2. formulate the reconstruction problem
3. Give guarantees for sparsity vs number of measurements

Sensing process model

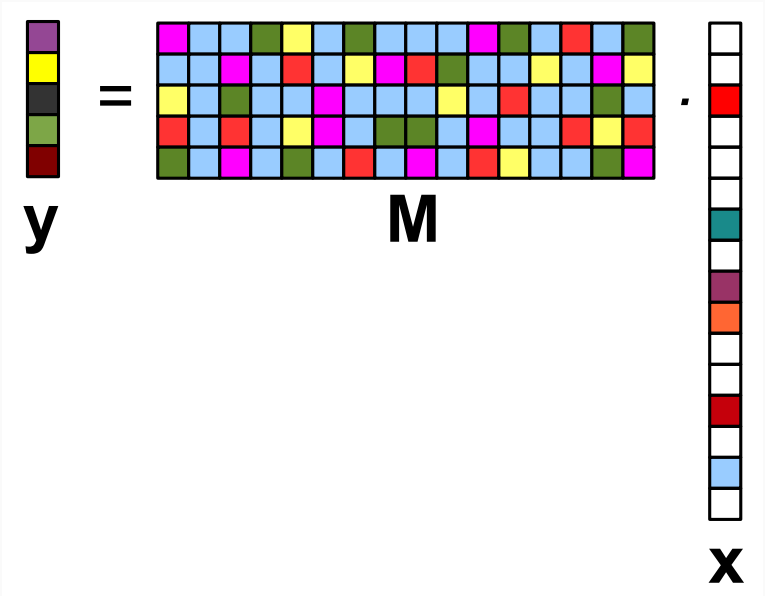
Modeling the dependency between signal and measurement

Let $x \in R^n$ be a s-sparse signal to be recovered.

Let $y \in R^m$, $m < n$, be linear measurements of the signal as

$$y = Mx$$

with $M \in R^{m \times n}$, being the sensing matrix.

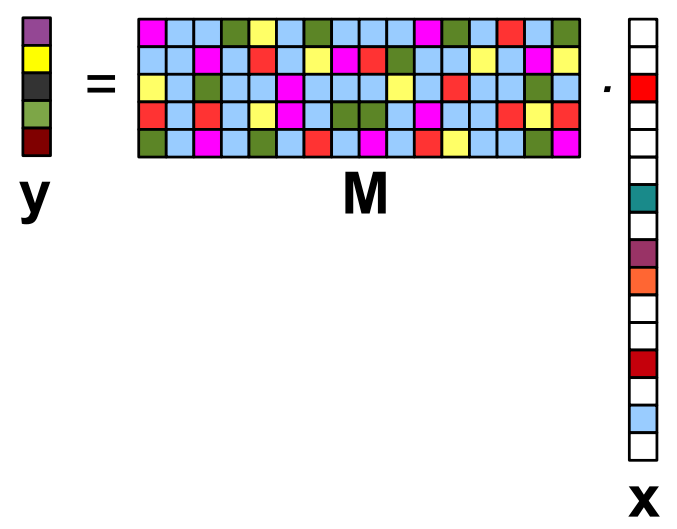


Reconstruction: problem formulation

problem formulation

Given measurement y, sensing matrix M and the model $y = Mx$,

Recover x, s-sparse.

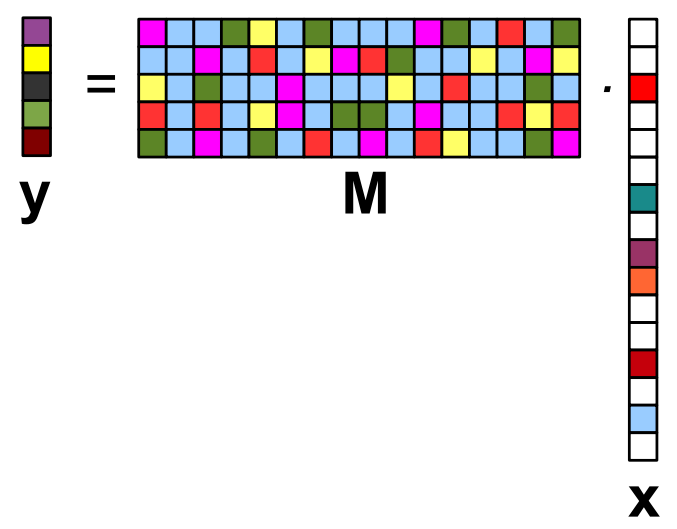


Difficulties?

Reconstruction: problem formulation

problem formulation

Given measurement y, sensing matrix M and the model $y = Mx$,
Recover x, s-sparse.



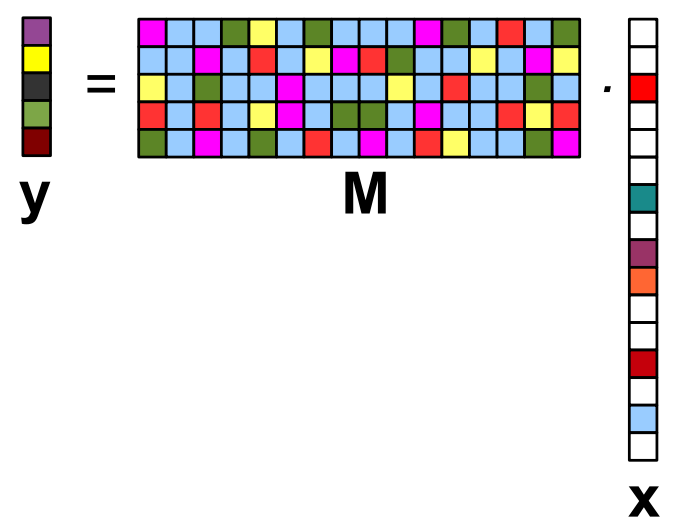
Difficulties?

- Underdetermined system $\Rightarrow$ infinitely many solutions.

Reconstruction: problem formulation

problem formulation

Given measurement y, sensing matrix M and the model $y = Mx$,
Recover x, s-sparse.



Difficulties?

- Underdetermined system $\Rightarrow$ infinitely many solutions.

$\rightarrow$ exploit the sparsity assumption of x.

Minimum l_0-norm solution

$$\hat{x} = \arg\min_{z \in \mathbb{R}^n} ||z||_0 \text{ subject to } Mz = y$$

- Reminder: recover all s-sparse x **iff all sets of $2s$ columns of M are linearly independent** (so-called EsR condition)

Minimum l_0-norm solution

$$\hat{x} = \arg\min_{z \in \mathbb{R}^n} ||z||_0 \text{ subject to } Mz = y$$

- Reminder: recover all s-sparse x **iff all sets of $2s$ columns of M are linearly independent** (so-called EsR condition)
- Complexity?
  - Problem is non-convex
  - Problem is **NP-hard**!

  $\rightarrow$ for a given s, try all possible $\binom{n}{s}$ supports, estimate the s nonzero values of x, check if constraint is satisfied

  $\rightarrow$ **infeasible** for practical problem sizes

Practical philosophies

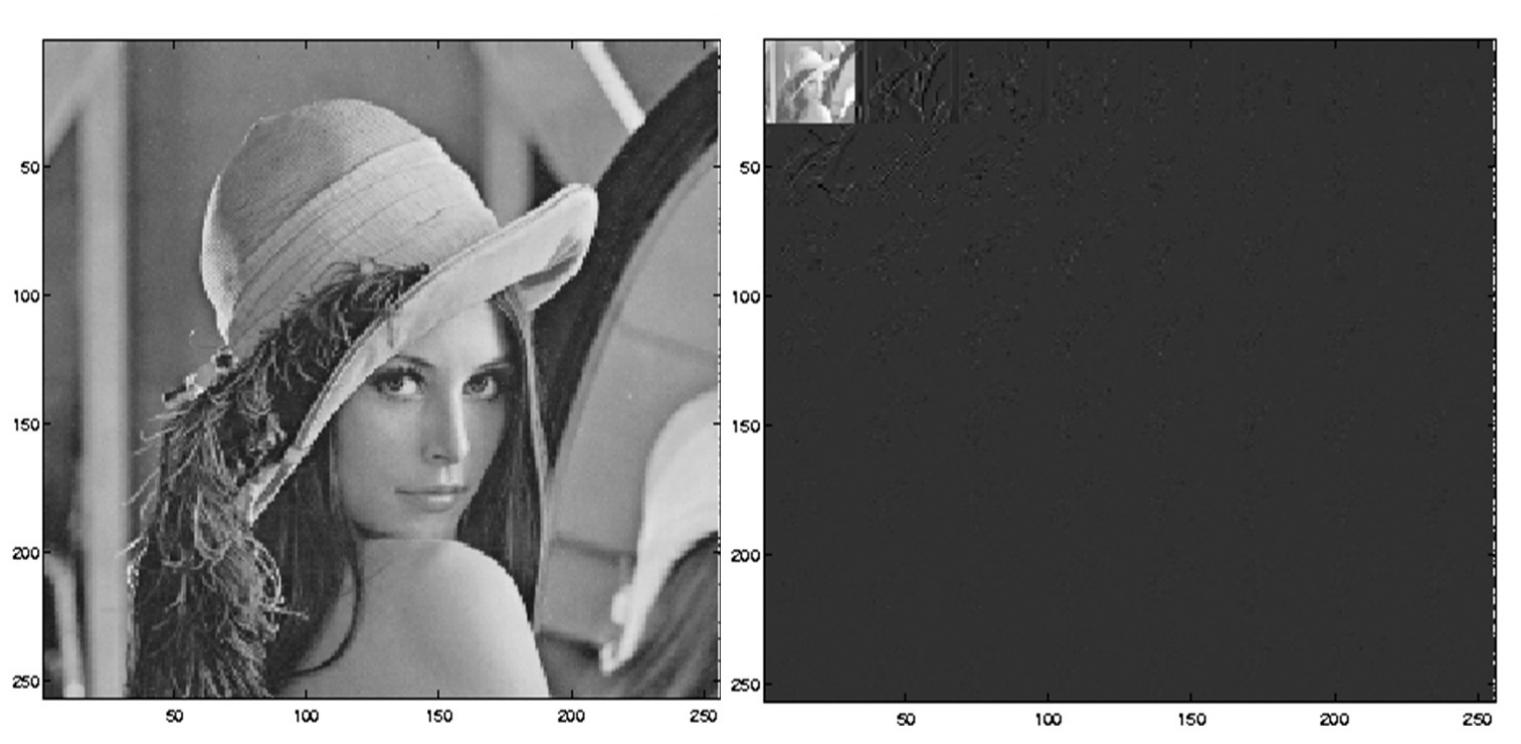
$$\hat{x} = \arg\min_{z \in \mathbb{R}^n} ||z||_0 \text{ subject to } Mz = y$$

Greedy algorithms	Thresholding algorithms	Convex relaxation algorithms
Focus on $\|\|x\|\|_0$	Focus on $y \sim Mx$	Solve a nicer problem
MP, OMP, OLS...	IHT...	BP, LASSO...
see course T. Guyard	see course T. Guyard	see course L. Le Magoarou

Signal sparse in transform domain

Real signals are rarely directly sparse...

but rather sparse in a transform domain



original image

DCT coefficients of the image in the transform domain

original image

Signal sparse vs signal sparse in transform domain

x sparse

SENSING

$y = Mx$

RECONSTRUCTION

$\hat{x} = \arg\min_{z \in \mathbb{R}^n} ||z||_1$

subject to $Mz = y$

.

$x = \Phi u$, u sparse

SENSING

$y = Mx$

RECONSTRUCTION

$\hat{u} = \arg\min_{z \in \mathbb{R}^n} ||z||_1$

subject to $M\Phi z = y$

$\hat{x} = \Phi \hat{u}$

In conclusion: sparse vs sparse in the transform domain

- same sensing
- similar reconstruction problem
- Make sure that $M\Phi$ (and not M) is a **"good" sensing matrix** (see later...)

Outline

CS vs SA

Compressive sensing and **sparse approximation** refer to similar, but subtly different, ideas.

CS vs SA

Compressive sensing and **sparse approximation** refer to similar, but subtly different, ideas.

- Sparse approximation is generally used in classical sensing, as a **compression scheme**.

CS vs SA

Compressive sensing and **sparse approximation** refer to similar, but subtly different, ideas.

- Sparse approximation is generally used in classical sensing, as a **compression scheme**. For instance:
  - take sparse transform (DCT) $c = \Phi x$
  - **keep s largest coefficient c_s** (non-linear!)
  - To recover (approximately) x, invert $\hat{x} = \Phi^{-1} c_s$ (linear!)

CS vs SA

Compressive sensing and **sparse approximation** refer to similar, but subtly different, ideas.

- Sparse approximation is generally used in classical sensing, as a **compression scheme**. For instance:
  - take sparse transform (DCT) $c = \Phi x$
  - **keep s largest coefficient c_s** (non-linear!)
  - To recover (approximately) x, invert $\hat{x} = \Phi^{-1} c_s$ (linear!)

- Compressive Sensing directly takes less samples, but recovery is highly non-linear
  - $y = Mx$ (linear!)
  - Recovery is done by sparse decoding (non-linear!)

CS vs SA (con't)

More precisely:

CS Given y and M, find $\hat{x}$
sparse such that $M\hat{x} \approx y$.

Return $\hat{x}$ with guarantee that
$||\hat{x} - x||$ small

SA Given x and **dictionary** D,
find $\hat{c}$ sparse such that
$\hat{x} = D\hat{c} \approx x$.
(generally, but not always, $\hat{c}$ is the
best-s-term approx. of Φx, and
$D = \Phi^{-1}$)

Return $\hat{x}$ with guarantee that
$||\hat{x} - x|| = ||D(\hat{c} - c)||$ small

CS vs SA (con't)

More precisely:

CS Given y and M, find $\hat{x}$ sparse such that $M\hat{x} \approx y$.

Return $\hat{x}$ with guarantee that

$$||\hat{x} - x|| \quad \text{small}$$

Same decomposition algorithms

SA Given x and **dictionary** D, find $\hat{c}$ sparse such that $\hat{x} = D\hat{c} \approx x$.

(generally, but not always, $\hat{c}$ is the best-s-term approx. of Φx, and $D = \Phi^{-1}$)

Return $\hat{x}$ with guarantee that

$$||\hat{x} - x|| = ||D(\hat{c} - c)|| \quad \text{small}$$

Different recovery criteria

Example: root finding

Root-finding algorithm: Given $y = 0$ and a function f, find $\hat{x}$ such that $y = 0 \approx f(\hat{x})$

- CS: Return $\hat{x}$ with guarantees that

$$\|\hat{x} - x\| \text{ small}$$

SA: Return $\hat{y} = f(\hat{x})$ with guarantees that

$$\|f(\hat{x}) - 0\| \text{ small}$$

Example: root finding

Root-finding algorithm: Given $y = 0$ and a function f, find $\hat{x}$ such that $y = 0 \approx f(\hat{x})$

- CS: Return $\hat{x}$ with guarantees that

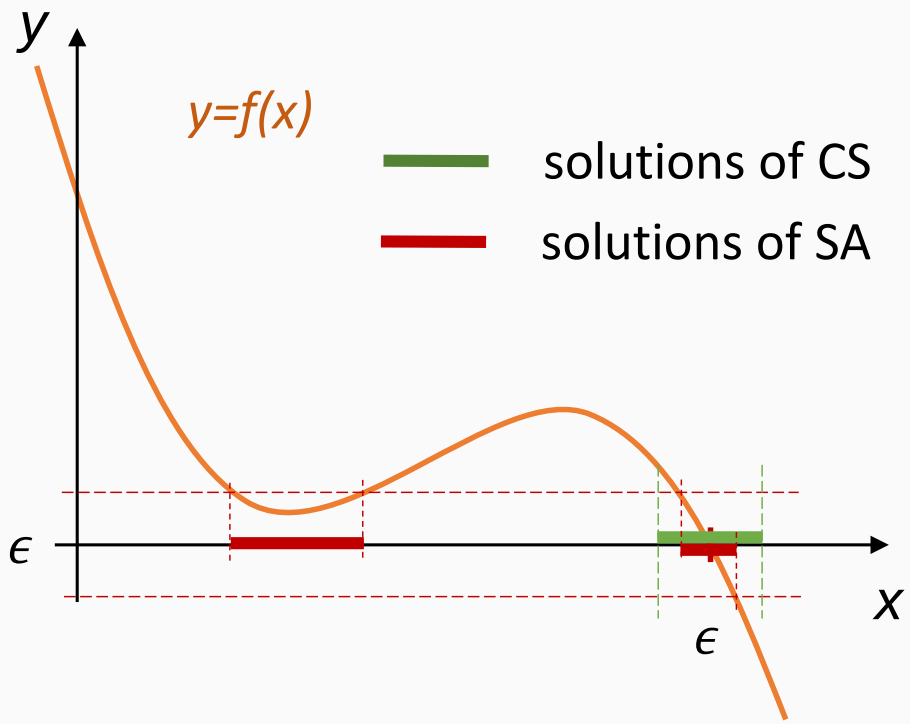
$$\|\hat{x} - x\| \text{ small}$$

SA: Return $\hat{y} = f(\hat{x})$ with guarantees that

$$\|f(\hat{x}) - 0\| \text{ small}$$

CS: proximity to the true root
SA: proximity to zero in the range of the function

CS vs SA (con't)



Root-finding algorithm:

CS Given $y = 0$ and f, find $\hat{x}$ such that $y = 0 \approx f(\hat{x})$.
Return $\hat{x}$ with guarantee that

$$||\hat{x} - x|| \quad \text{small}$$

SA Given $y = 0$ and f, find $\hat{x}$ such that $y = 0 \approx \hat{y} = f(\hat{x})$.
Return $\hat{y}$ with guarantee that

$$||f(\hat{x}) - 0|| \quad \text{small}$$

CS: proximity to the true root
SA: proximity to zero in the range of the function

Other terminology

Given a measured signal x,

- **Analysis:** Φx is sparse
  - Φ is an analysis operator
  - Ex: DCT, Wavelet *transform*

Other terminology

Given a measured signal x,

- **Analysis:** Φx is sparse
  - Φ is an analysis operator
  - Ex: DCT, Wavelet *transform*

- **Synthesis:** $x = Du$, u is sparse
  - D is a dictionary
  - Ex: Wavelet *dictionary*, etc.

$\rightarrow$ When invertible transform, $D = \Phi^{-1}$, but that is not always the case, and the interpretation is different

Compressive sensing: summary

Compressive sensing overview

Observe $x \in \mathbb{R}^n$ via m measurements, with $m \ll n$

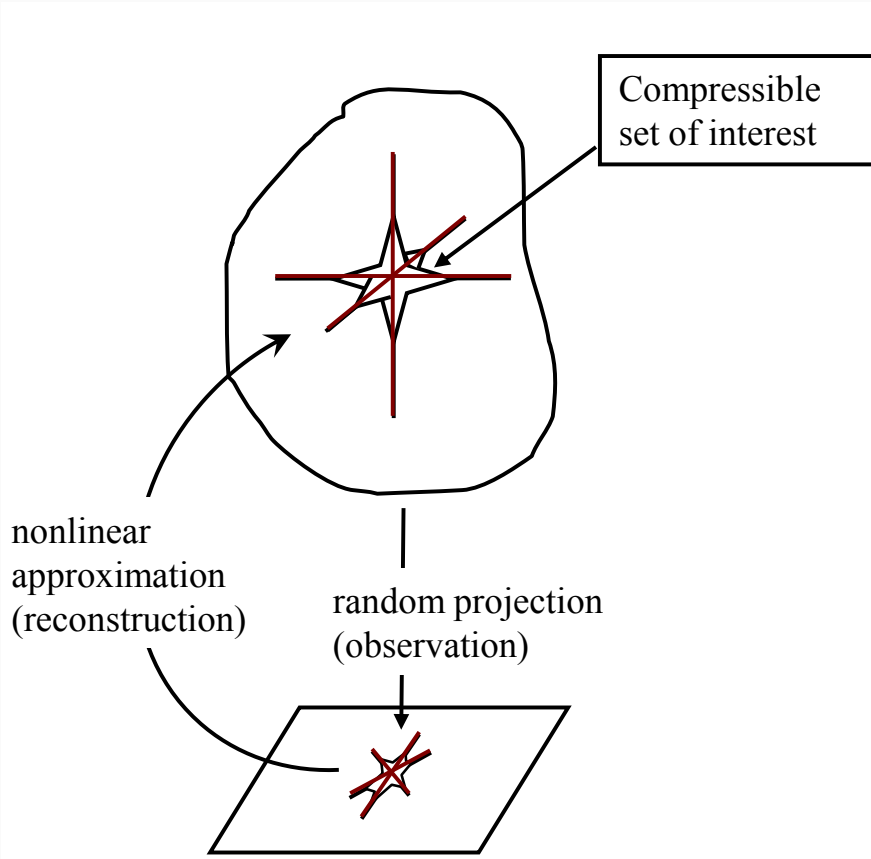
More precisely, $y = Mx$ where $y \in \mathbb{R}^m$

Assumptions:

- signal approximately s-sparse
- use $m \geq c\ s \log \frac{n}{s}$, c=constant,

random linear measurements

- reconstruct by a non linear mapping

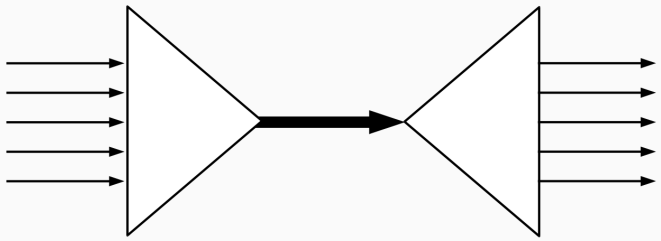


Mike Davies.

How to spot a compressive sensing system?

Case 1

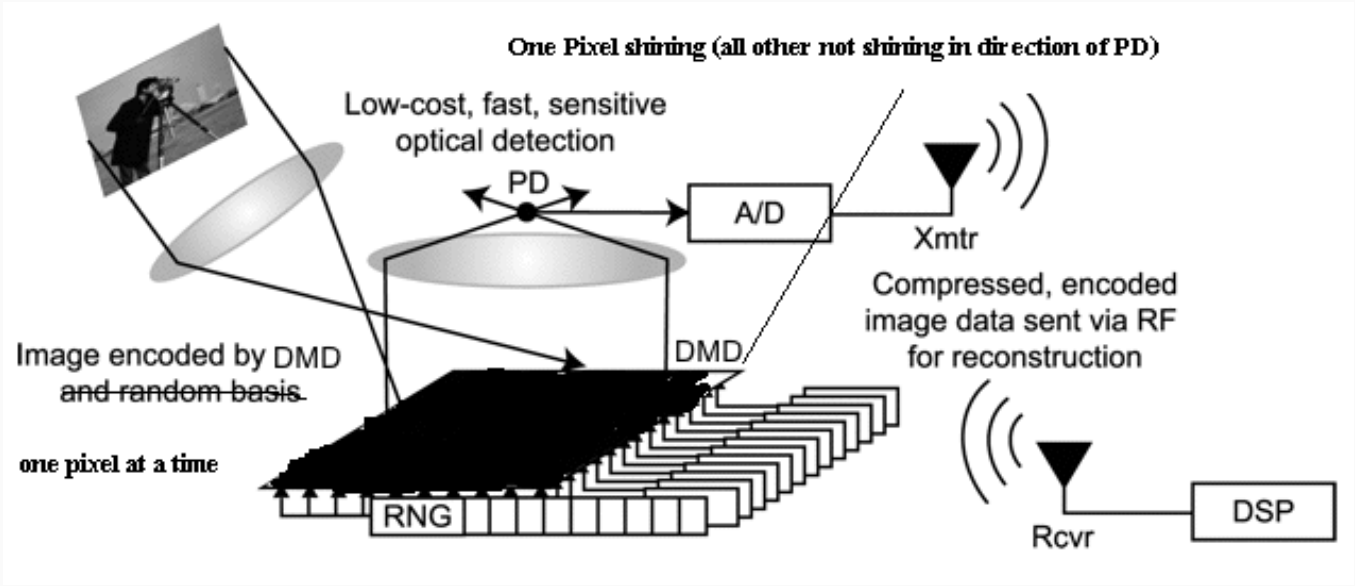
- Think about systems that use a raster mode for sampling
  then think of physical ways to perform multiplexing instead



- Once you perform the multiplexing,
  use compressive sensing solvers to reconstruct signal
- Does it work better or as well with fewer measurements ?

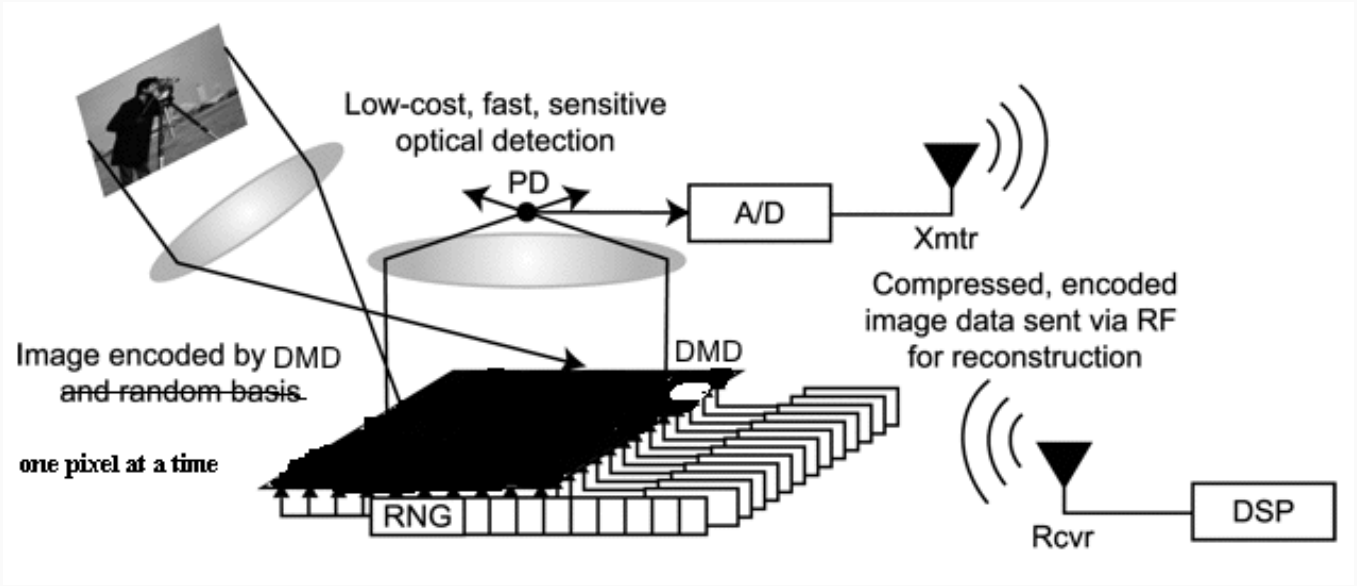
Example: single pixel camera

Classical i=1



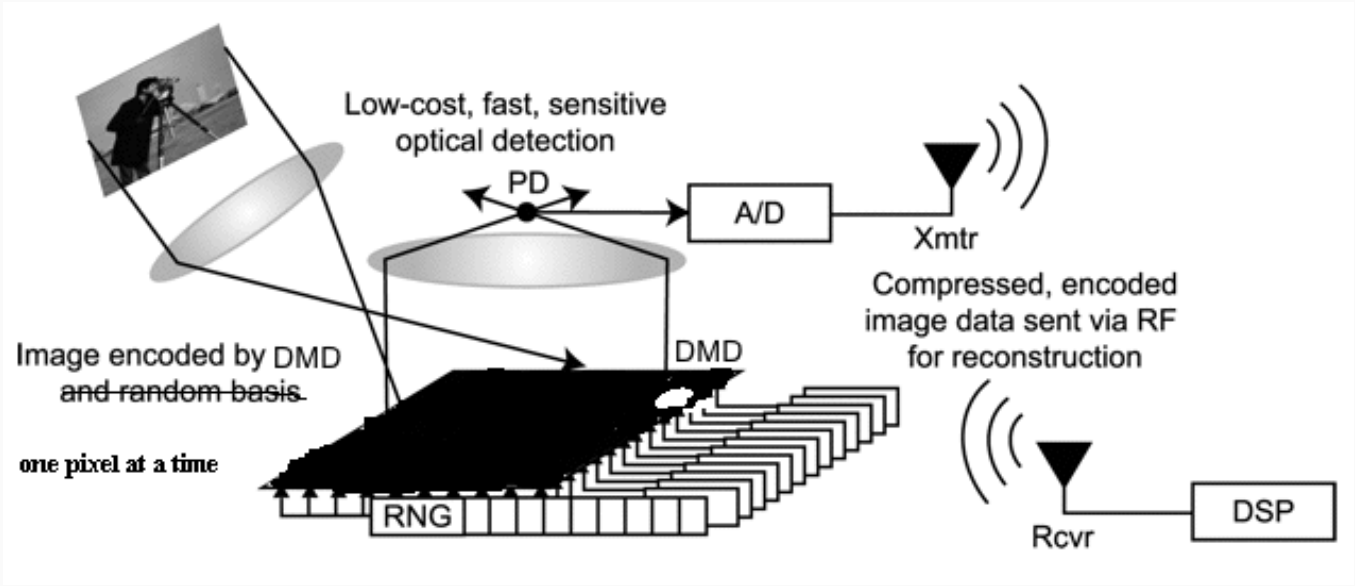
Example: single pixel camera

Classical i=2



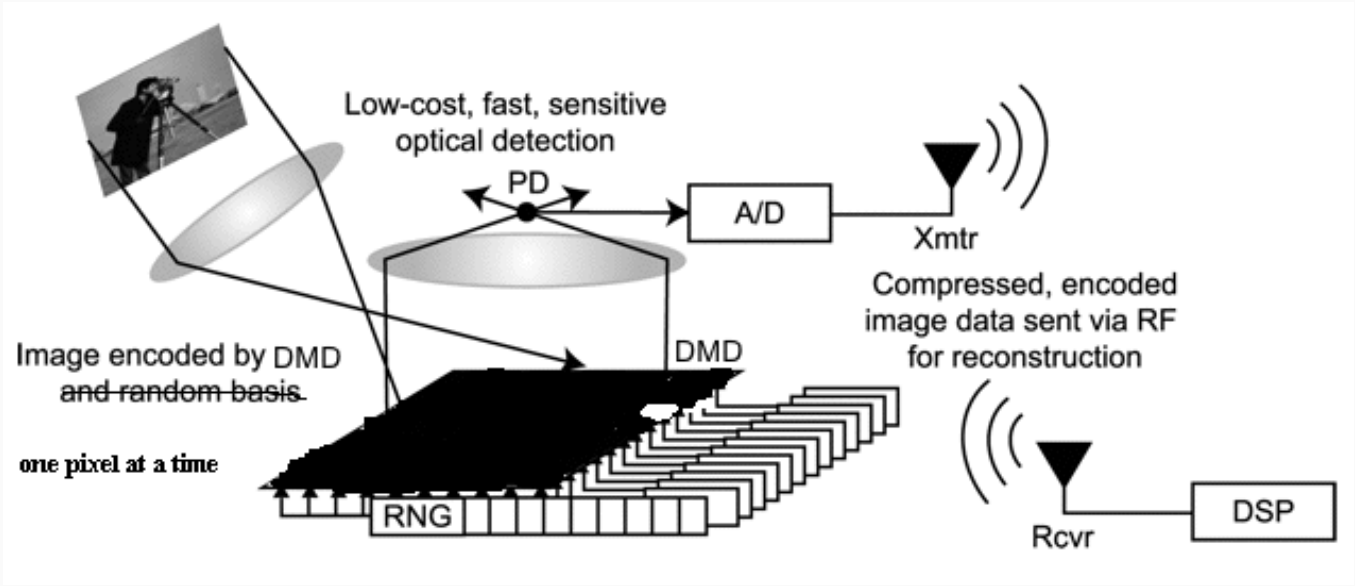
Example: single pixel camera

Classical i=3



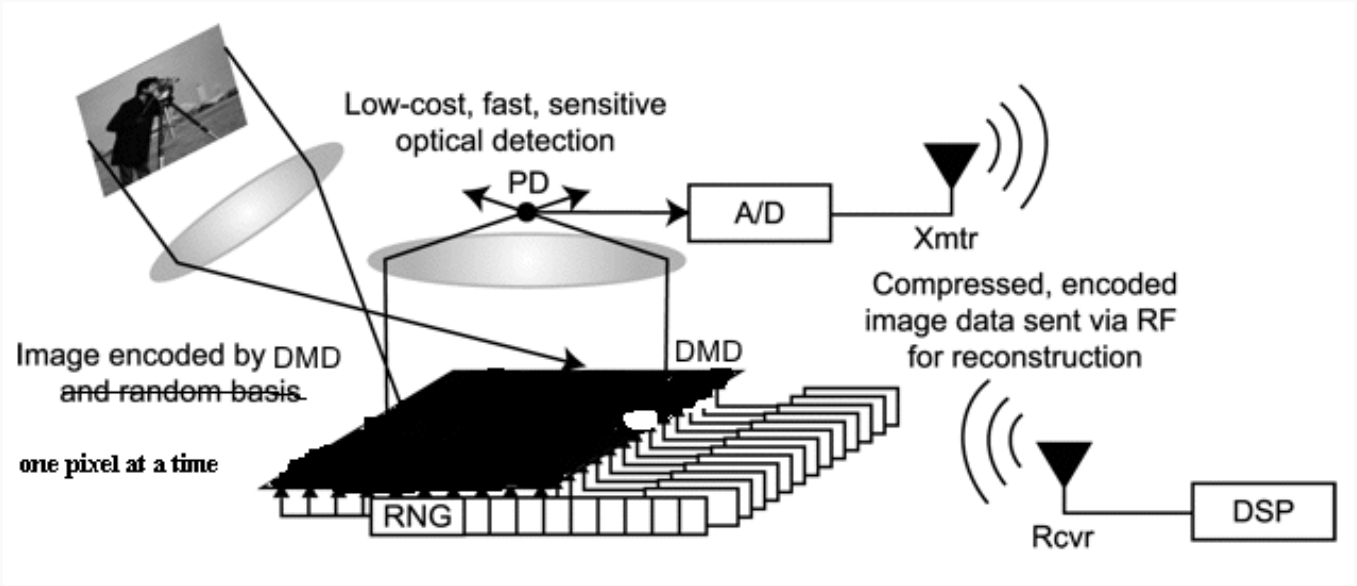
Example: single pixel camera

Classical i=4



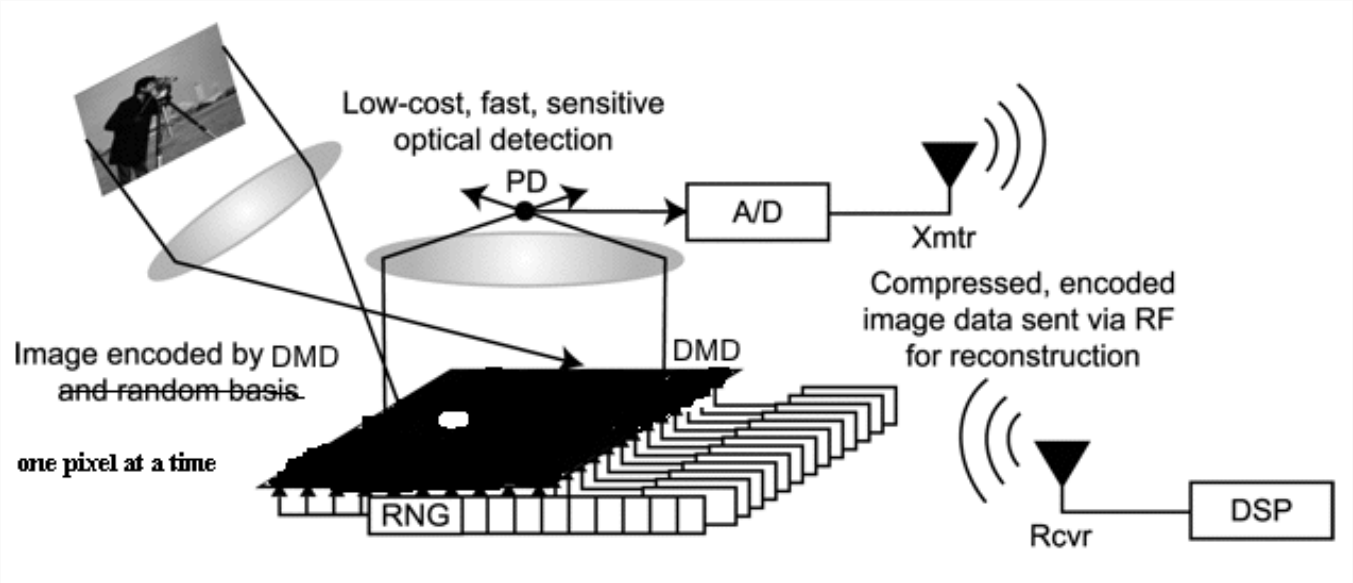
Example: single pixel camera

Classical i=5



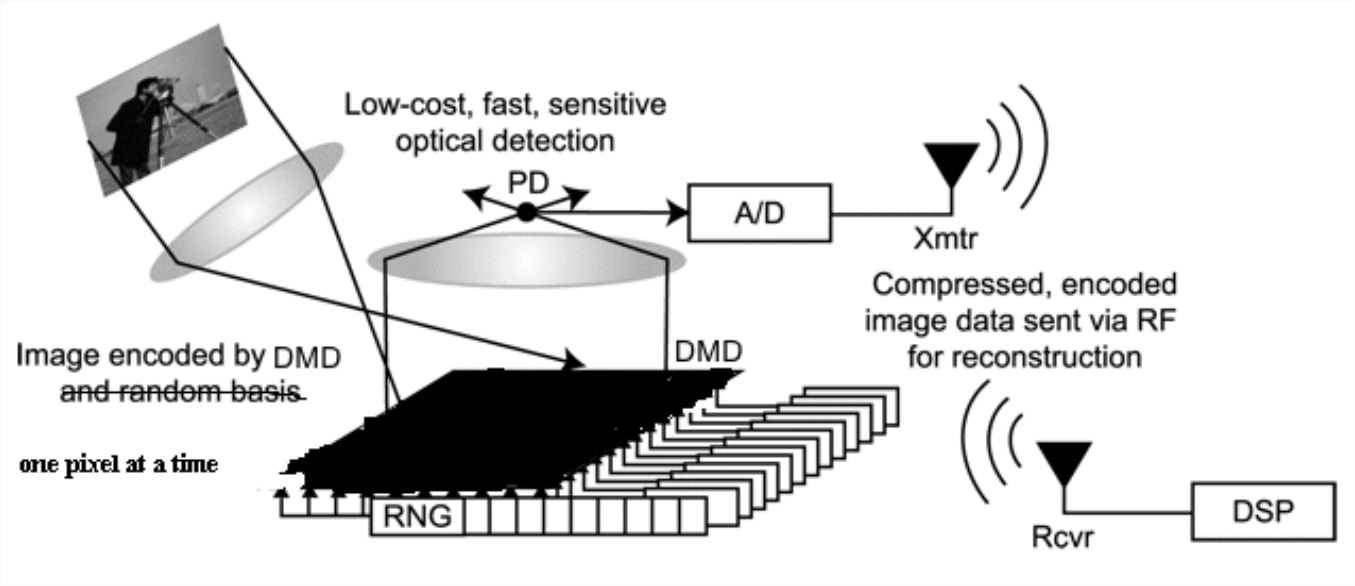
Example: single pixel camera

Classical i=1000



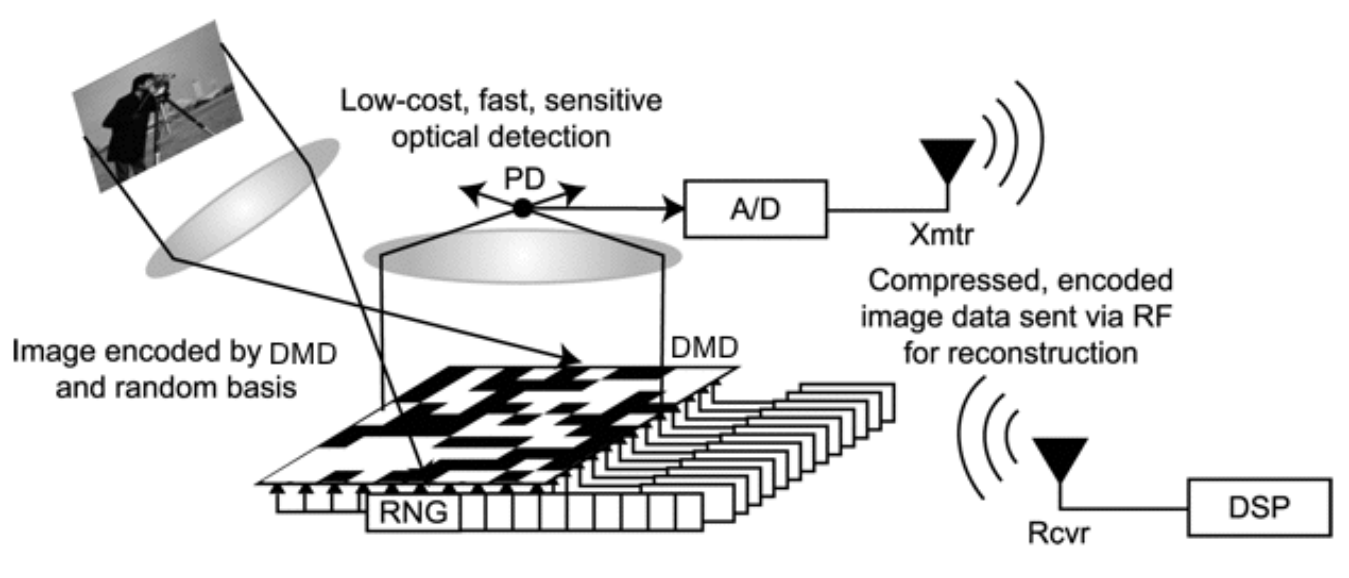
Example: single pixel camera

Classical i=10000000



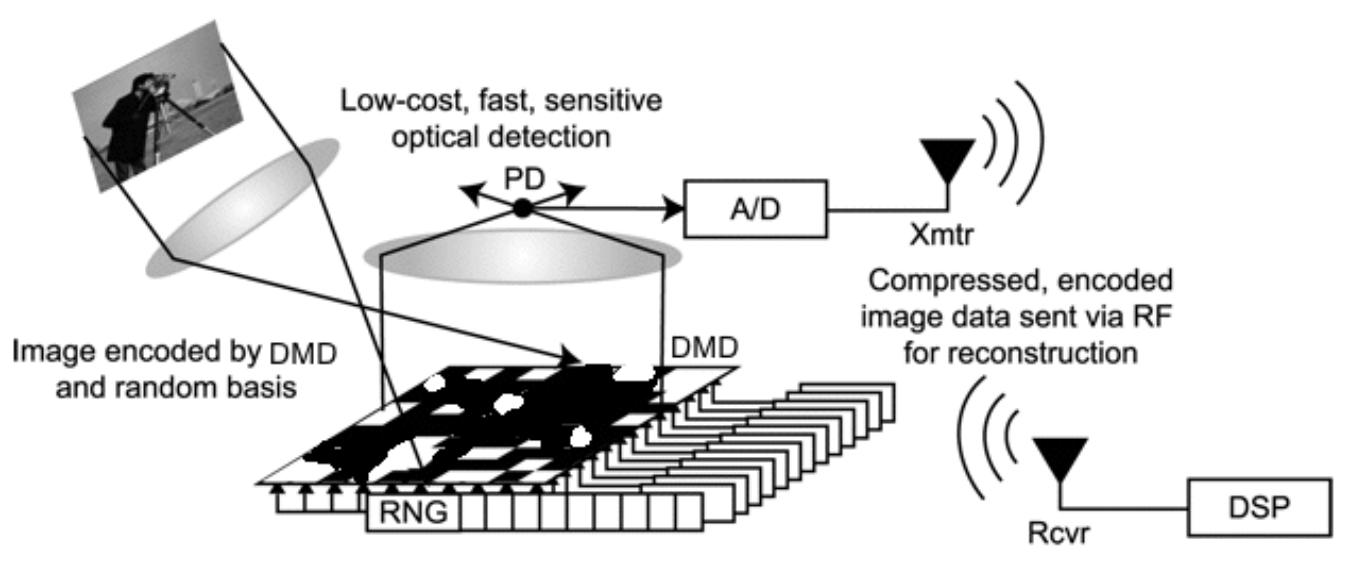
Example: single pixel camera

Compressive sensing i=1



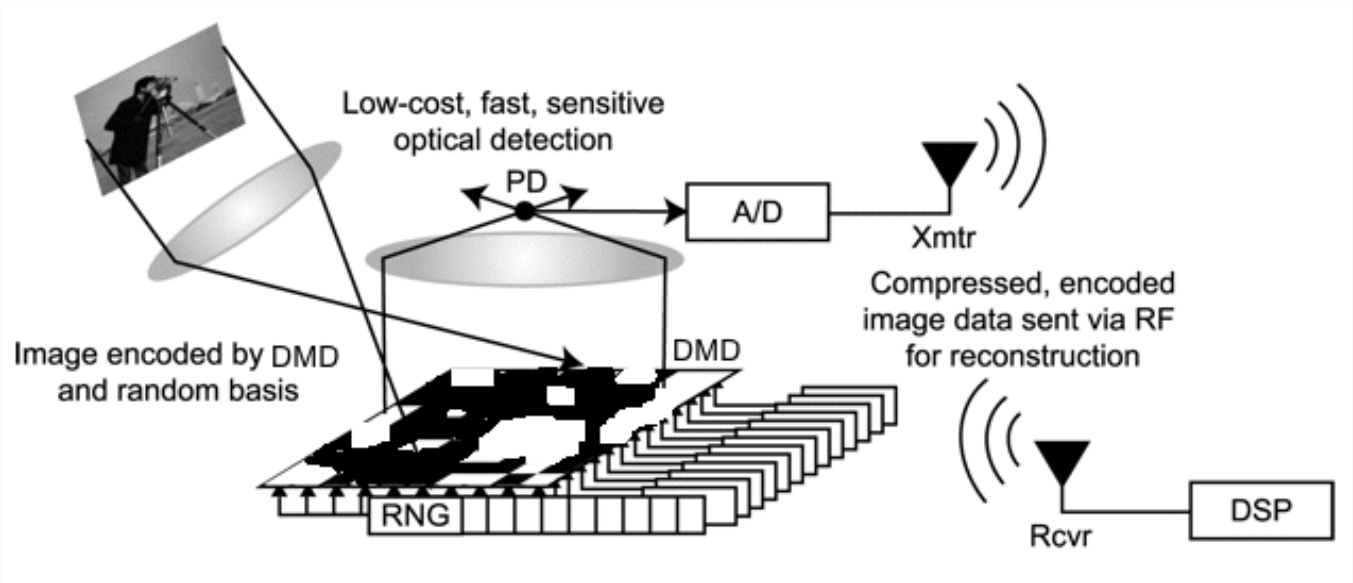
Example: single pixel camera

Compressive sensing i=2



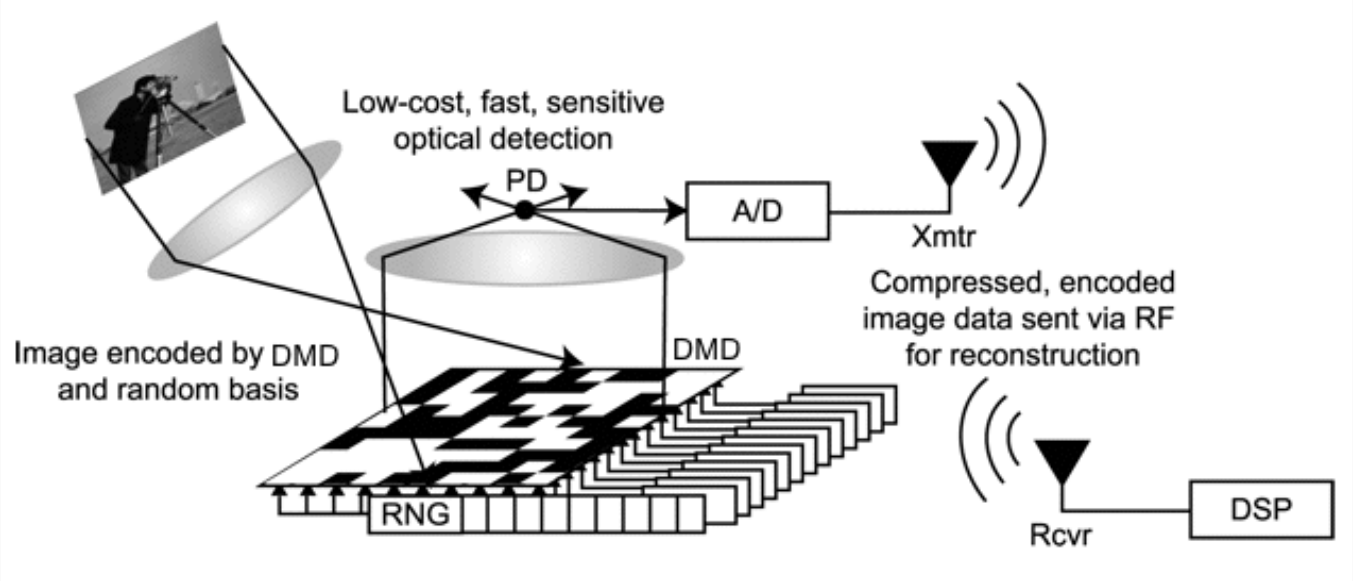
Example: single pixel camera

Compressive sensing i=3



Example: single pixel camera

Compressive sensing



if image is 3-sparse, the sufficient number of measurements scales with 3 and not the size of the image!!!!

How to spot a compressive sensing system?

Case 2

- Look for acquisition schemes that multiplexes a signal already
- Is the signal produced by this system sparse in some basis?
- If yes, subsample the acquisition,
  use compressive sensing solvers to reconstruct signal
- Does it work better or as well with fewer measurements ?

Recovery guarantees

Outline

Reminder: ℓ_0 and EsR

Decoder for sparse vectors:

$$\min \|z\|_0 \text{ s.t. } y = Mz \qquad (P_0)$$

Reminder: ℓ_0 and EsR

Decoder for sparse vectors:

$$\min \|z\|_0 \text{ s.t. } y = Mz \qquad (P_0)$$

(P_0) achieves exact recovery of all s-sparse vectors iff

All subsets of $2s$ columns of M are linearly independent (EsR)

Reminder: ℓ_0 and EsR

Decoder for sparse vectors:

$$\min \|z\|_0 \text{ s.t. } y = Mz \qquad (P_0)$$

(P_0) achieves exact recovery of all s-sparse vectors iff

$$\textit{All subsets of } 2s \textit{ columns of } M \textit{ are linearly independent} \qquad (\text{E}s\text{R})$$

But (P_0) is NP-hard, so *one* solution is to solve instead

$$\min \|z\|_p^p \text{ s.t. } y = Mz \qquad (P_p)$$

$\rightarrow$ You have seen $p = 1$ (aka Basis Pursuit) as a *convex* relaxation of ℓ_0, but the general case is also interesting. Other possibilities: greedy approaches, etc.

Reminder: ℓ_0 and EsR

Decoder for sparse vectors:

$$\min \|z\|_0 \text{ s.t. } y = Mz \qquad (P_0)$$

(P_0) achieves exact recovery of all s-sparse vectors iff

All subsets of $2s$ columns of M are linearly independent (EsR)

But (P_0) is NP-hard, so *one* solution is to solve instead

$$\min \|z\|_p^p \text{ s.t. } y = Mz \qquad (P_p)$$

$\rightarrow$ You have seen $p = 1$ (aka Basis Pursuit) as a *convex* relaxation of ℓ_0, but the general case is also interesting. Other possibilities: greedy approaches, etc.

Recovery guarantees? Many... In this section: Null Space Property

Reminder: solution set

Recall: The set of potential solutions is

$$\{x' \mid y = Mx'\} = \{x + v \mid v \in \ker(M)\}$$

Reminder: solution set

Recall: The set of potential solutions is

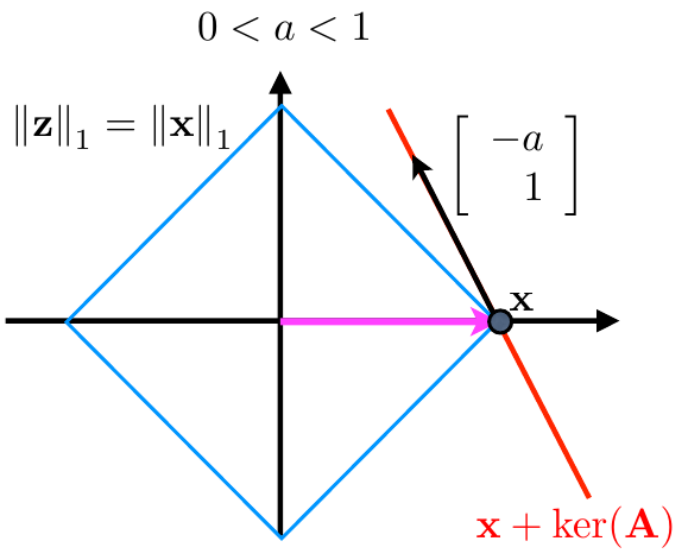
$$\{x' \mid y = Mx'\} = \{x + v \mid v \in \ker(M)\}$$

$\rightarrow$ The key lies in studying the *kernel*, or *null space* of M

$\rightarrow$ Easy: x is the unique solution of (P_p) iff
$\|x\|_p^p < \|x + v\|_p^p \ \forall v \in \ker(M) \backslash \{0\}$

Illustration

$p = 1$: Success! and convex problem :)



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^2$$

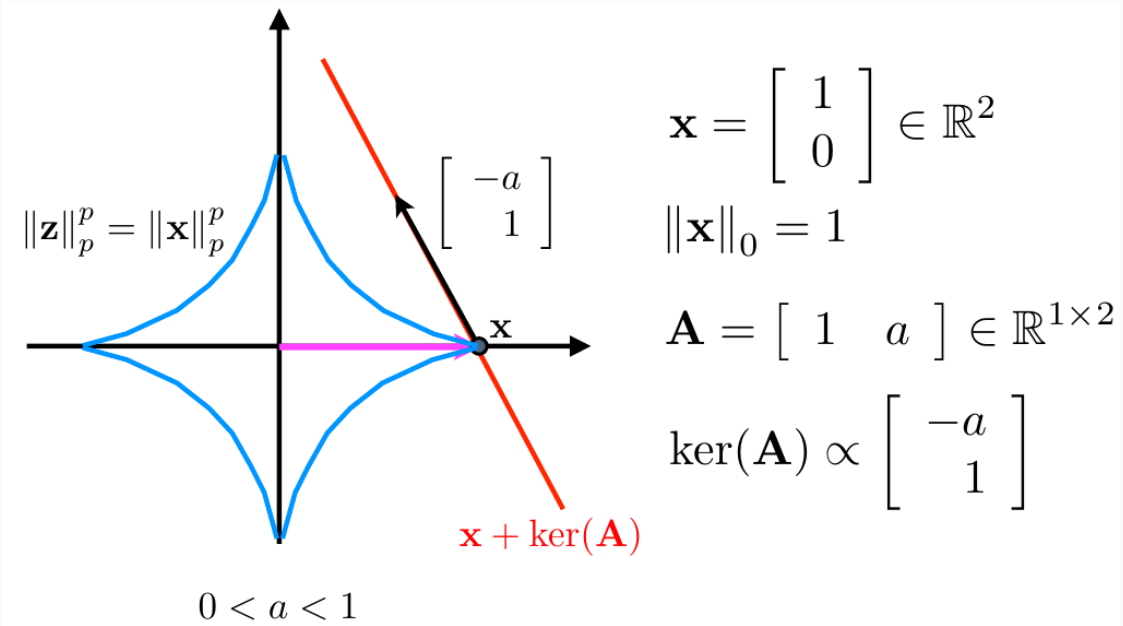
$$\|\mathbf{x}\|_0 = 1$$

$$\mathbf{A} = \begin{bmatrix} 1 & a \end{bmatrix} \in \mathbb{R}^{1\times 2}$$

$$\ker(\mathbf{A}) \propto \begin{bmatrix} -a \\ 1 \end{bmatrix}$$

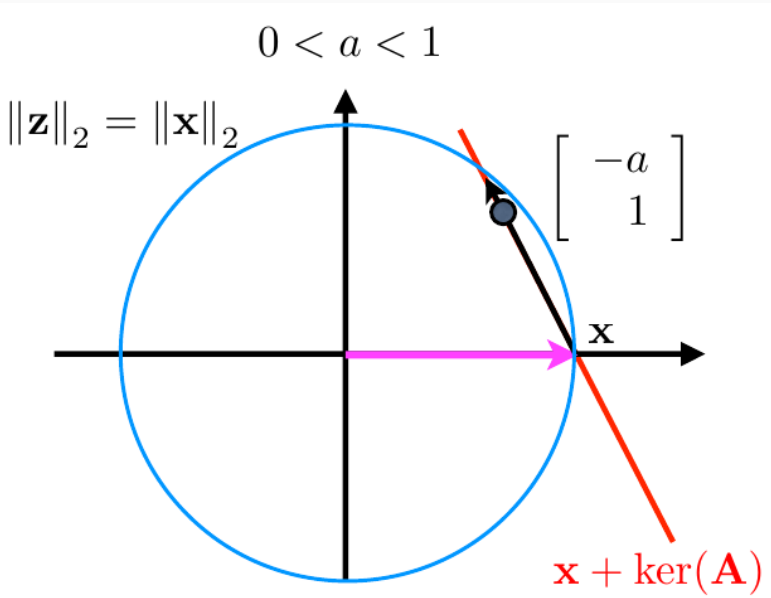
Illustration

$0 < p < 1$: Success! But non-convex...



Illustration

$p = 2$: Fail... In fact, $p > 1$ will "always" fail to recover sparse vectors



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^2$$

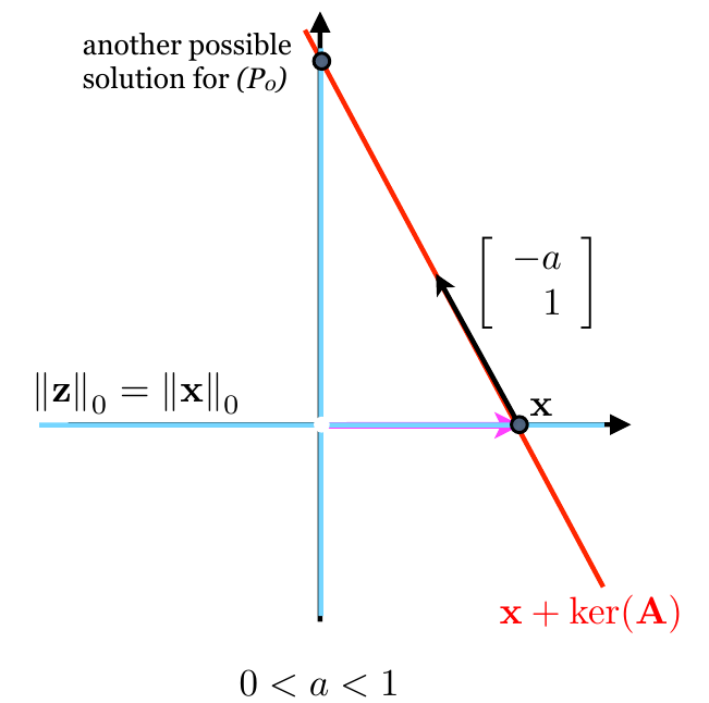
$$\|\mathbf{x}\|_0 = 1$$

$$\mathbf{A} = \begin{bmatrix} 1 & a \end{bmatrix} \in \mathbb{R}^{1\times 2}$$

$$\ker(\mathbf{A}) \propto \begin{bmatrix} -a \\ 1 \end{bmatrix}$$

Illustration

$p = 0$: Fail!! ...for this example. *"But I thought $p = 0$ was the best?"* $\rightarrow$ Uniform vs non-uniform recovery: more to come.



Null Space Property

Null Space Property (F&R, Thm 4.4)

Take $p \leq 1$.

"Every s-sparse vectors x **supported on** S is the solution of (P_p) with $y = Mx$"

iff

$$\|v_S\|_p^p < \|v_{S^c}\|_p^p \quad \forall v \in \ker(M) \backslash \{0\} \qquad (NSP_p(S))$$

Null Space Property

Null Space Property (F&R, Thm 4.4)

Take $p \leq 1$.

"Every s-sparse vectors x **supported on** S is the solution of (P_p) with $y = Mx$"

iff

$$\|v_S\|_p^p < \|v_{S^c}\|_p^p \quad \forall v \in \ker(M) \backslash \{0\} \qquad (NSP_p(S))$$

Proof sketch: *(exo for $p = 1$)*

Null Space Property

Null Space Property (F&R, Thm 4.4)

Take $p \leq 1$.

"Every s-sparse vectors x **supported on** S is the solution of (P_p) with $y = Mx$"

iff

$$\|v_S\|_p^p < \|v_{S^c}\|_p^p \quad \forall v \in \ker(M)\backslash\{0\} \qquad (NSP_p(S))$$

Proof sketch: *(exo for $p = 1$)*

$\Leftarrow$ Take x supported on S and $z \neq x$ with $Mz = Mx$, and $v = x - z \in \ker(M)\backslash\{0\}$. Then, by the *quasi-triangle inequality*,

$$\begin{aligned}\|x\|_p^p &\leq \|x - z_S\|_p^p + \|z_S\|_p^p = \|v_S\|_p^p + \|z_S\|_p^p \\ &< \|v_{S^c}\|_p^p + \|z_S\|_p^p = \|z_{S^c}\|_p^p + \|z_S\|_p^p = \|z\|_p^p\end{aligned}$$

Hence $\|x\|_p^p$ is minimal

Null Space Property

Null Space Property (F&R, Thm 4.4)

Take $p \leq 1$.

"Every s-sparse vectors x **supported on** S is the solution of (P_p) with $y = Mx$"

iff

$$\|v_S\|_p^p < \|v_{S^c}\|_p^p \quad \forall v \in \ker(M)\backslash\{0\} \qquad (NSP_p(S))$$

Proof sketch: *(exo for $p = 1$)*

$\Leftarrow$ Take x supported on S and $z \neq x$ with $Mz = Mx$, and $v = x - z \in \ker(M)\backslash\{0\}$. Then, by the *quasi-triangle inequality*,

$$\begin{aligned}\|x\|_p^p &\leq \|x - z_S\|_p^p + \|z_S\|_p^p = \|v_S\|_p^p + \|z_S\|_p^p \\ &< \|v_{S^c}\|_p^p + \|z_S\|_p^p = \|z_{S^c}\|_p^p + \|z_S\|_p^p = \|z\|_p^p\end{aligned}$$

Hence $\|x\|_p^p$ is minimal

$\Rightarrow$ Take $v \in \ker(M)\backslash\{0\}$. By hypothesis, $x = v_S$ is the unique minimizer of (P_p) with $y = Mv_S$. Since we have also: $v \neq 0 \Rightarrow v_S \neq -v_{S^c}$, and $Av = 0 \Rightarrow A(-v_{S^c}) = Av_S = y$, then $\|v_S\|_p^p < \|v_{S^c}\|_p^p$

Null Space Property

Immediately, we have:

Null Space Property (F&R, Thm 4.5)

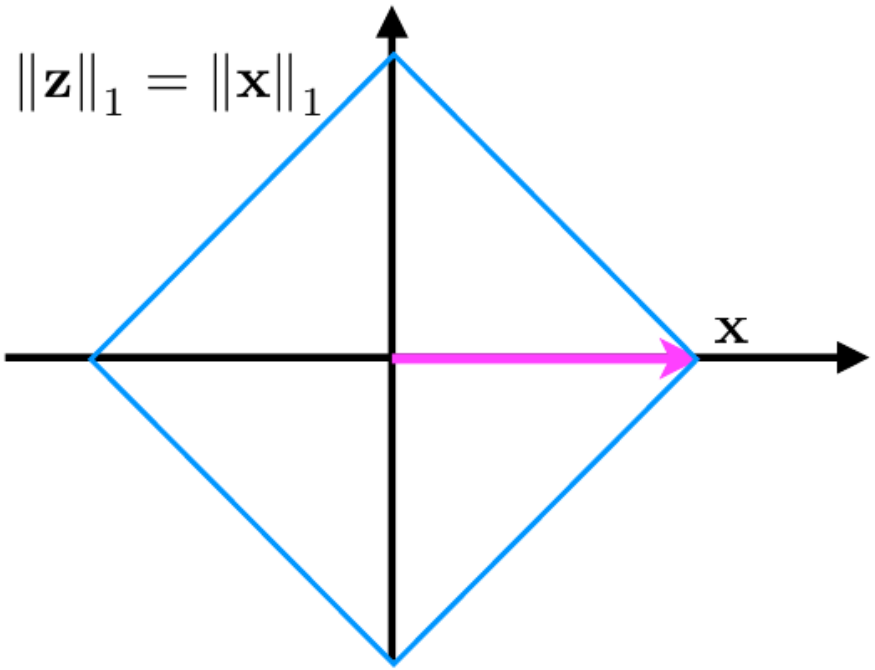
Take $p \leq 1$. Every s-sparse vector x is the solution of (P_p) with $y = Mx$

iff

$$\forall S \text{ with } card(S) = s,\ NSP_p(S) \text{ is verified} \qquad (NSP_p(s))$$

- In particular $NSP_0(s)$ is equivalent to EsR ! (Homework)

NSP example



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^2$$

$$\|\mathbf{x}\|_0 = 1, S = \{1\}$$

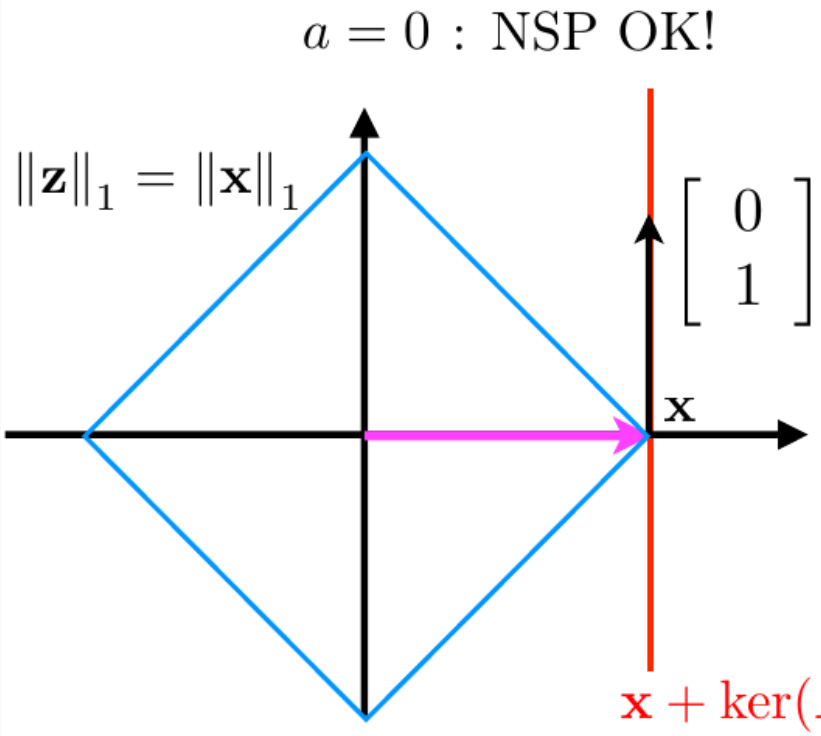
$$\mathbf{A} = \begin{bmatrix} 1 & a \end{bmatrix} \in \mathbb{R}^{1\times 2}$$

$$\ker(\mathbf{A}) \propto \begin{bmatrix} -a \\ 1 \end{bmatrix}$$

$$(\mathrm{NSP}_p(S)) : \|\mathbf{v}_S\|_1^1 < \|\mathbf{v}_{\bar{S}}\|_1^1 \equiv |a| < 1$$

Can we have $NSP_p(1)$?

NSP example



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^2$$

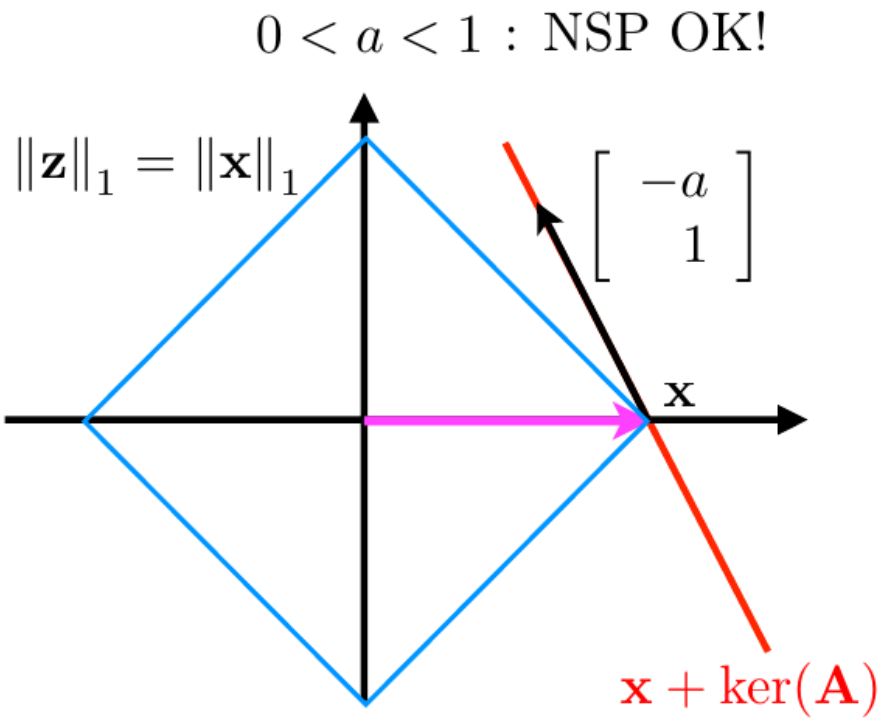
$$\|\mathbf{x}\|_0 = 1, S = \{1\}$$

$$\mathbf{A} = \begin{bmatrix} 1 & a \end{bmatrix} \in \mathbb{R}^{1\times 2}$$

$$\ker(\mathbf{A}) \propto \begin{bmatrix} -a \\ 1 \end{bmatrix}$$

$$(\mathrm{NSP}_p(S)) : \|\mathbf{v}_S\|_1^1 < \|\mathbf{v}_{\bar{S}}\|_1^1 \equiv |a| < 1$$

NSP example



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^2$$

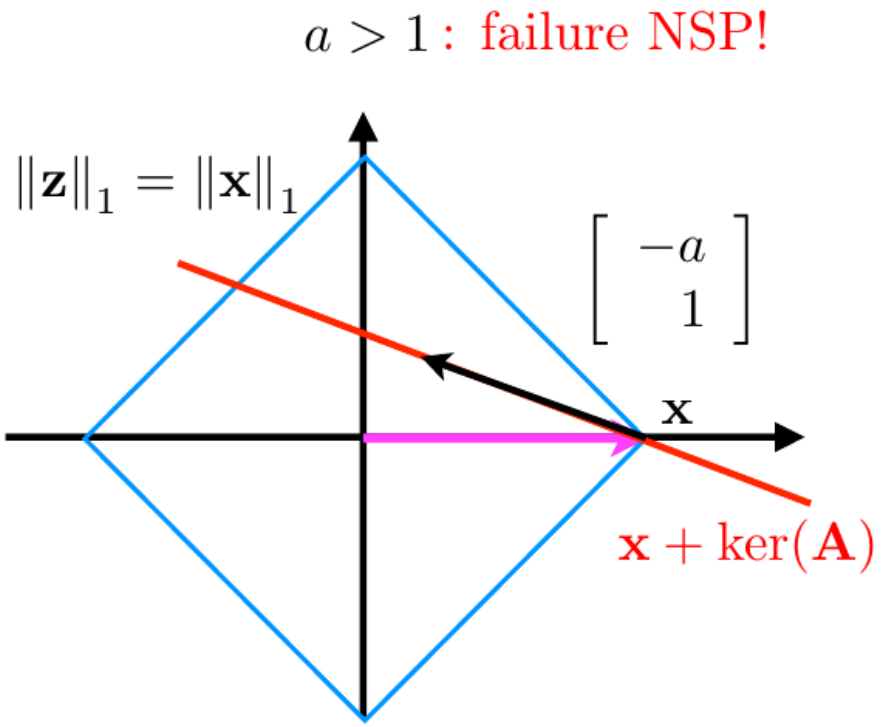
$$\|\mathbf{x}\|_0 = 1, S = \{1\}$$

$$\mathbf{A} = \begin{bmatrix} 1 & a \end{bmatrix} \in \mathbb{R}^{1\times 2}$$

$$\ker(\mathbf{A}) \propto \begin{bmatrix} -a \\ 1 \end{bmatrix}$$

$$(\mathrm{NSP}_p(S)) : \|\mathbf{v}_S\|_1^1 < \|\mathbf{v}_{\bar{S}}\|_1^1 \equiv |a| < 1$$

NSP example



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^2$$

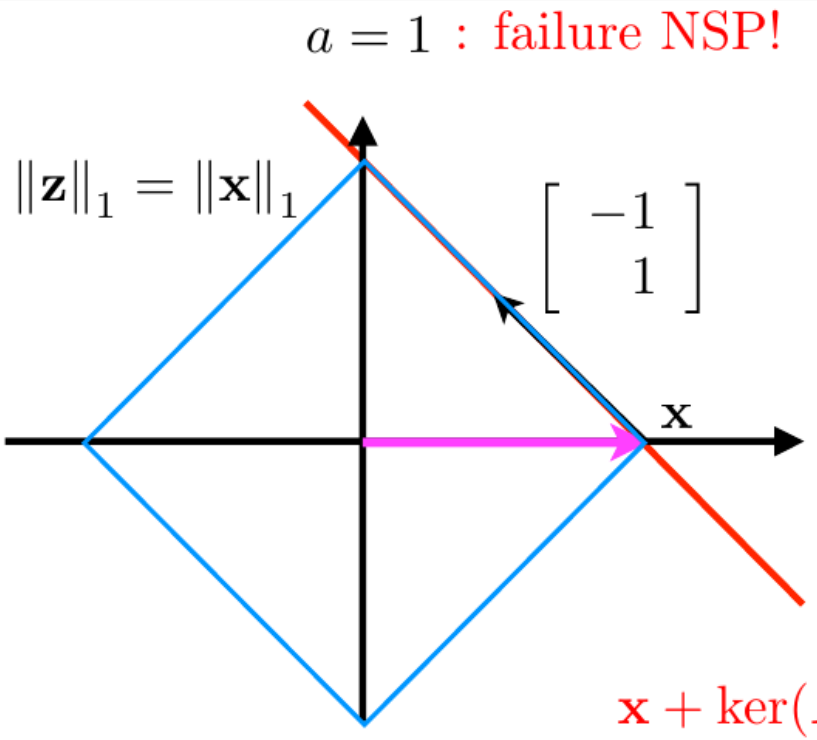
$$\|\mathbf{x}\|_0 = 1, S = \{1\}$$

$$\mathbf{A} = \begin{bmatrix} 1 & a \end{bmatrix} \in \mathbb{R}^{1\times 2}$$

$$\ker(\mathbf{A}) \propto \begin{bmatrix} -a \\ 1 \end{bmatrix}$$

$$(\mathrm{NSP}_p(S)) : \|\mathbf{v}_S\|_1^1 < \|\mathbf{v}_{\bar{S}}\|_1^1 \equiv |a| < 1$$

NSP example



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^2$$

$$\|\mathbf{x}\|_0 = 1, S = \{1\}$$

$$\mathbf{A} = \begin{bmatrix} 1 & a \end{bmatrix} \in \mathbb{R}^{1\times 2}$$

$$\ker(\mathbf{A}) \propto \begin{bmatrix} -a \\ 1 \end{bmatrix}$$

$$(\mathrm{NSP}_p(S)) : \|\mathbf{v}_S\|_1^1 < \|\mathbf{v}_{\bar{S}}\|_1^1 \equiv |a| < 1$$

Here x is **a** solution. Modified NSP: $\|v_S\|_p^p \leq \|v_{S^c}\|_p^p$

Instance vs Uniform recovery

x is the unique solution...	is equivalent to
...for a given x	$\|x\|_p^p < \|x+v\|_p^p$ for all $v \in \ker(M)\backslash\{0\}$
...for all x supported on S	$NSP_p(S)$
...for all s-sparse x	$NSP_p(x)$

- Rk: NSP is a "worst-case" condition. Failure means that *some* x cannot be recovered.

Some properties of the NSP

Sparser is better!

- If $S' \subset S$, then $NSP_p(S) \Rightarrow NSP_p(S')$
- If $s' \leq s$, then $NSP_p(s) \Rightarrow NSP_p(s')$

Proof: easy.

Some properties of the NSP

Sparser is better!

- If $S' \subset S$, then $NSP_p(S) \Rightarrow NSP_p(S')$
- If $s' \leq s$, then $NSP_p(s) \Rightarrow NSP_p(s')$

Proof: easy.

Preservation under shuffling (of y!), rescaling, new observations

If $NSP_p(S)$ (idem for $NSP_p(s)$) is satisfied for M, it is also satisfied for

- $M' = GM$ with $G \in \mathbb{R}^{m \times m}$ invertible (shuffling, rescaling)
  - Proof: the kernel does not change!
- $M' \in \mathbb{R}^{m' \times n}$ is obtained by adding rows to M (new observations)
  - Proof: $\ker(M') \subset \ker(M)$

Some properties of the NSP

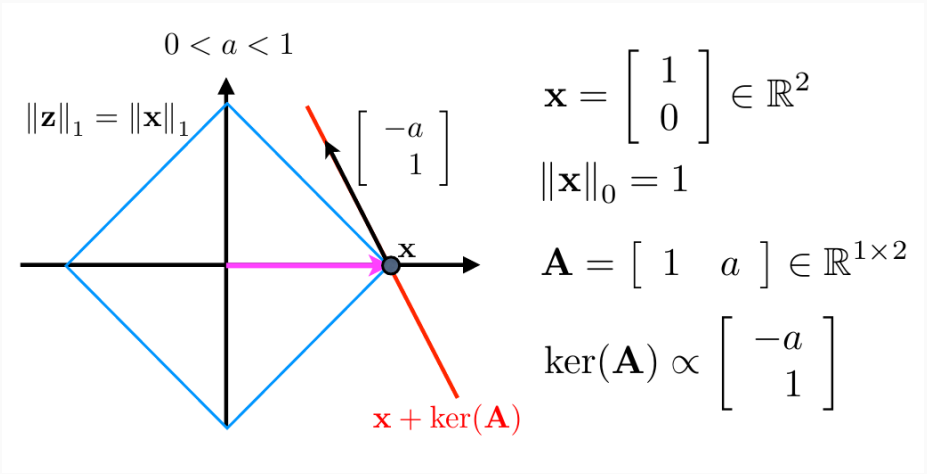
Is ℓ_0 really the best?

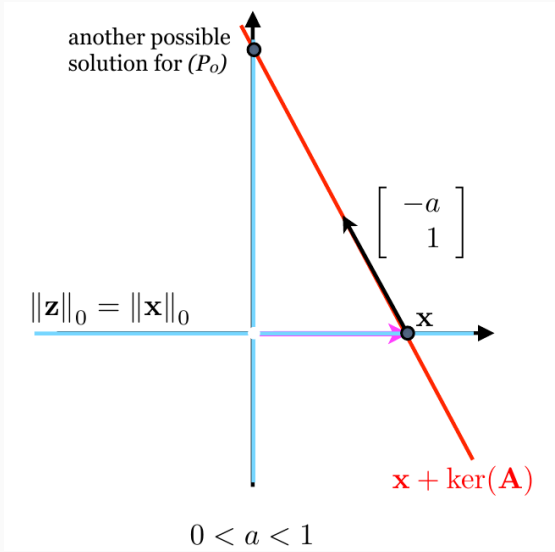
- Let $q < p$. Do we have $NSP_p(S) \Rightarrow NSP_q(S)$?

Some properties of the NSP

Is ℓ_0 really the best?

- Let $q < p$. Do we have $NSP_p(S) \Rightarrow NSP_q(S)$?
  - **NO**! Think about the previous example $|a|^p < 1^p$: we have $|a| < 1$ but not $|a|^0 < 1^0$

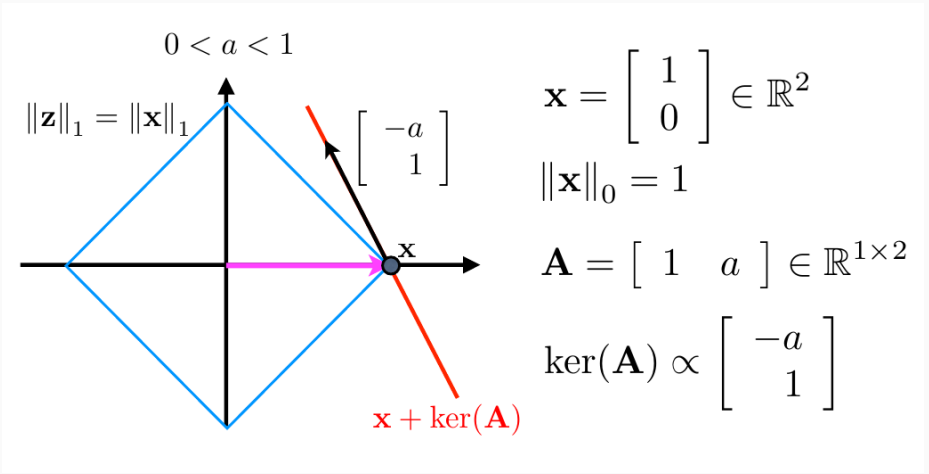


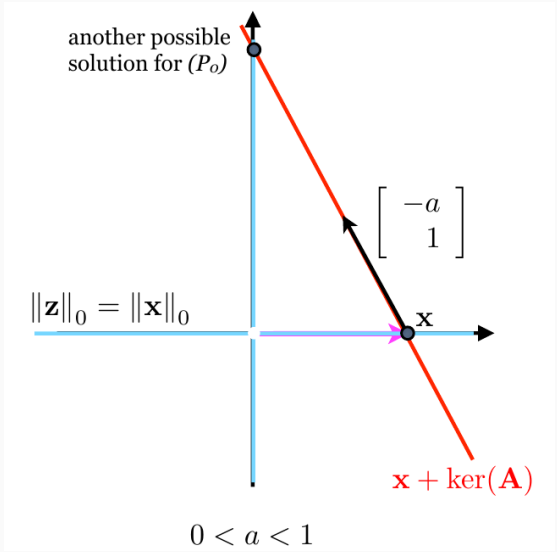


Some properties of the NSP

Is ℓ_0 really the best?

- Let $q < p$. Do we have $NSP_p(S) \Rightarrow NSP_q(S)$?
  - **NO**! Think about the previous example $|a|^p < 1^p$: we have $|a| < 1$ but not $|a|^0 < 1^0$



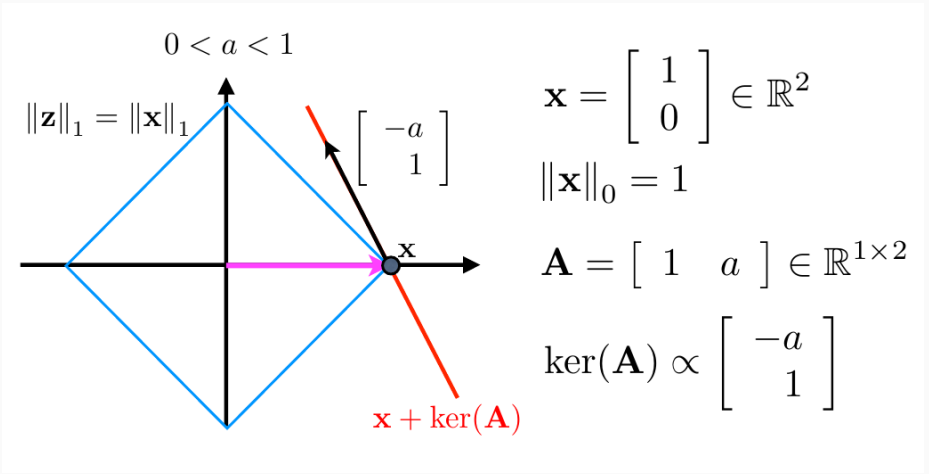


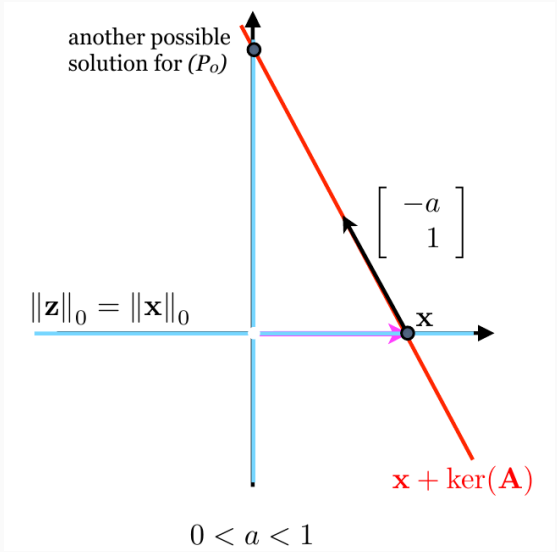
- Let $q < p$. Do we have $NSP_p(s) \Rightarrow NSP_q(s)$?

Some properties of the NSP

Is ℓ_0 really the best?

- Let $q < p$. Do we have $NSP_p(S) \Rightarrow NSP_q(S)$?
  - **NO**! Think about the previous example $|a|^p < 1^p$: we have $|a| < 1$ but not $|a|^0 < 1^0$



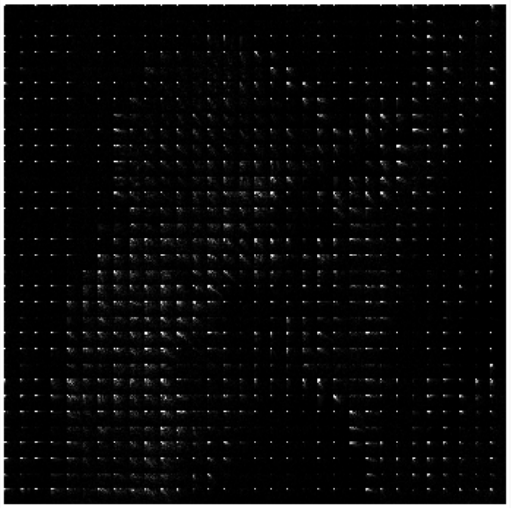


- Let $q < p$. Do we have $NSP_p(s) \Rightarrow NSP_q(s)$?
  - **YES** :) [F&R Thm 4.10]

$\rightarrow$ p small is indeed better for **uniform** recovery, for instance recovery we cannot conclude.

Stability

In practice, we have seen that x is not exactly sparse, but **compressible**: fast decaying $\sigma_s(x)_1 = \min_{z \in \Sigma_s} \|x - z\|_1$. Can we handle that? This is called **stability**.



Stability (for ℓ_1)

Stable NSP

M satisfies the stable NSP with constant $0 < \rho < 1$ relative to S if

$$\|v_S\|_1 \leq \rho \|v_{S^c}\|_1 \quad \forall v \in \ker(M) \qquad (\rho\text{-NSP}_1(S))$$

$\rightarrow$ stronger than $\text{NSP}_1(S)$ since $\rho < 1$

$\rightarrow$ using a "$\leq$" this time does not matter

$\rightarrow$ ρ-$\text{NSP}_1(s)$ when ρ-$\text{NSP}_1(S)$ for all s-support S

$\rightarrow$ NB: here we do ℓ_1, there are extensions for ℓ_p

Stability (for ℓ_1)

Stable NSP

M satisfies the stable NSP with constant $0 < \rho < 1$ relative to S if

$$\|v_S\|_1 \leq \rho \|v_{S^c}\|_1 \quad \forall v \in \ker(M) \qquad (\rho\text{-NSP}_1(S))$$

$\rightarrow$ stronger than $\text{NSP}_1(S)$ since $\rho < 1$

$\rightarrow$ using a "$\leq$" this time does not matter

$\rightarrow$ ρ-$\text{NSP}_1(s)$ when ρ-$\text{NSP}_1(S)$ for all s-support S

$\rightarrow$ NB: here we do ℓ_1, there are extensions for ℓ_p

Thm [F&R 4.12]

Suppose that ρ-$\text{NSP}_1(s)$ is satisfied. Then for **any** x, **a** solution z of (P_1) with $y = Mx$ satisfies

$$\|x - z\|_1 \leq \frac{2(1+\rho)}{1-\rho} \sigma_s(x)_1$$

$\rightarrow$ s appears in the bound and the NSP, not in any assumption on x!

$\rightarrow$ (P_1) still uses the "exact" constraint $y = Mz$

Robustness

In practice, there is (always!) measurement **noise**! Bounded for some norm $\|\cdot\|$.

$$y = Mx + e \text{ with } \|e\| \leq \eta$$

Recovery guarantees in the presence of noise if called **robustness**.

Robustness

In practice, there is (always!) measurement **noise**! Bounded for some norm $\|\cdot\|$.

$$y = Mx + e \text{ with } \|e\| \leq \eta$$

Recovery guarantees in the presence of noise if called **robustness**.

(P_p) must be replaced. For instance, by Constrained BP (CBP)

$$\min_z \|z\|_p^p \text{ s.t. } \|Mz - y\| \leq \eta \qquad (P_{p,\eta})$$

$\rightarrow$ still convex for $p = 1$

$\rightarrow$ recall that this is *one* formulation, other (more or less "equivalent") include BPD and Lasso. See course of T. Guyard

Robustness (for ℓ_1)

Robust NSP

A matrix M satisfies the robust NSP with constant $0 < \rho < 1$ and $\tau > 0$ relative to S and $\|\cdot\|$ if

$$\|v_S\|_1 \leq \rho\|v_{S^c}\|_1 + \tau\|Mv\| \quad \forall v \in \mathbb{R}^n \qquad ((\rho,\tau)\text{-NSP}_1(S))$$

$\rightarrow$ Now for all v and not only v in kernel!

$\rightarrow$ stronger that ρ-NSP$_1(S)$ (take v in kernel)

$\rightarrow$ again, (ρ,τ)-NSP$_1(s)$ when (ρ,τ)-NSP$_1(S)$ for all s-support S

Robustness (for ℓ_1)

Robust NSP

A matrix M satisfies the robust NSP with constant $0 < \rho < 1$ and $\tau > 0$ relative to S and $\|\cdot\|$ if

$$\|v_S\|_1 \leq \rho\|v_{S^c}\|_1 + \tau\|Mv\| \quad \forall v \in \mathbb{R}^n \qquad ((\rho,\tau)\text{-NSP}_1(S))$$

$\rightarrow$ Now for all v and not only v in kernel!

$\rightarrow$ stronger that ρ-NSP$_1(S)$ (take v in kernel)

$\rightarrow$ again, (ρ,τ)-NSP$_1(s)$ when (ρ,τ)-NSP$_1(S)$ for all s-support S

Thm [F&R 4.19]

Suppose that (ρ,τ)-NSP$_1(s)$ is satisfied. Then for **any** x, **a** solution z of $(P_{1,\eta})$ with $y = Mx + e$ where $\|e\| \leq \eta$ satisfies

$$\|x - z\|_1 \leq \frac{2(1+\rho)}{1-\rho}\sigma_s(x)_1 + \frac{4\tau}{1-\rho}\eta$$

Summary on Null Space Property

- Yields guarantees for exact ℓ_p minimization with $p \leq 1$
  - not any particular algorithm
  - convex iff $p = 1$

Summary on Null Space Property

- Yields guarantees for exact ℓ_p minimization with $p \leq 1$
  - not any particular algorithm
  - convex iff $p = 1$
- "low-level" property that directly focuses on the *kernel* of M
  - As a consequence, recovery proofs are often "easy"
  - exact recovery guarantees are **if and only if** conditions (rare)

Summary on Null Space Property

- Yields guarantees for exact ℓ_p minimization with $p \leq 1$
  - not any particular algorithm
  - convex iff $p = 1$
- "low-level" property that directly focuses on the *kernel* of M
  - As a consequence, recovery proofs are often "easy"
  - exact recovery guarantees are **if and only if** conditions (rare)
- can be modified to be stable and robust
- But...
  - hard to "check" in practice!
  - hard to come up with "good" matrix M conveniently!
  - We will see more convenient notions: coherence, RIP

Outline

Reminder: OMP

Let us (briefly) study greedy approaches. Recall: they progressively build a support:

$$\hat{S}^{(l+1)} = \hat{S}^{(l)} \cup \{j\}$$

where j is "well-chosen".

Reminder: OMP

Let us (briefly) study greedy approaches. Recall: they progressively build a support:

$$\hat{S}^{(l+1)} = \hat{S}^{(l)} \cup \{j\}$$

where j is "well-chosen".

For OMP:

$$j \in \arg\max\{|m_i^\top (y - M_{\hat{S}^{(l)}} \hat{x}_{\hat{S}^{(l)}})|\} \tag{1}$$

where

$$\hat{x}_{\hat{S}^{(l)}} = \arg\min_z \|y - M_{\hat{S}^{(l)}} z\|_2 \tag{2}$$

Exact Recovery Condition (ERC)

ERC (F&R Prop 3.5, Rem 3.6)

OMP recovers any x supported on S after s iterations

if and only if

M_S is injective, and

$$\max_{i \in S} |m_i^\top r| > \max_{j \in S^c} |m_j^\top r| \ \forall r \in \text{span}(M_S) \backslash \{0\} \tag{3}$$

Exact Recovery Condition (ERC)

ERC (F&R Prop 3.5, Rem 3.6)

OMP recovers any x supported on S after s iterations

if and only if

M_S is injective, and

$$\max_{i \in S} |m_i^\top r| > \max_{j \in S^c} |m_j^\top r| \ \forall r \in \text{span}(M_S) \backslash \{0\} \tag{3}$$

The last equation is equivalent to the *exact recovery condition*

$$\max_{i \in S^c} \|M_S^\dagger m_i\|_1 < 1 \tag{ERC(S)}$$

where $M_S^\dagger$ is the pseudo inverse.

Exact Recovery Condition (ERC)

ERC (F&R Prop 3.5, Rem 3.6)

OMP recovers any x supported on S after s iterations

if and only if

M_S is injective, and

$$\max_{i \in S} |m_i^\top r| > \max_{j \in S^c} |m_j^\top r| \ \forall r \in \text{span}(M_S) \backslash \{0\} \tag{3}$$

The last equation is equivalent to the *exact recovery condition*

$$\max_{i \in S^c} \|M_S^\dagger m_i\|_1 < 1 \tag{ERC(S)}$$

where $M_S^\dagger$ is the pseudo inverse.

Again, we call ERC(s) when ERC(S) is satisfied for all supports of size s, and it is equivalent to exact recovery of all s-sparse vectors by OMP.

ERC properties

- Let $S' \subset S$. ERC(S) does **not** imply ERC(S') !
  - unlike $\text{NSP}_p(S) \Rightarrow \text{NSP}_p(S')$!
  - Intuitively, this is because we require OMP to succeed in exactly s steps. It could succeed after more steps! (but not guaranteed either)

ERC properties

- Let $S' \subset S$. ERC(S) does **not** imply ERC(S') !
  - unlike $\text{NSP}_p(S) \Rightarrow \text{NSP}_p(S')$!
  - Intuitively, this is because we require OMP to succeed in exactly s steps. It could succeed after more steps! (but not guaranteed either)

- Let $s' \leq s$. This time, ERC(s) $\Rightarrow$ ERC(s') [Mailhé et al. ICASSP13]

ERC properties

- Let $S' \subset S$. ERC(S) does **not** imply ERC(S') !
  - unlike $\text{NSP}_p(S) \Rightarrow \text{NSP}_p(S')$!
  - Intuitively, this is because we require OMP to succeed in exactly s steps. It could succeed after more steps! (but not guaranteed either)

- Let $s' \leq s$. This time, $\text{ERC}(s) \Rightarrow \text{ERC}(s')$ [Mailhé et al. ICASSP13]

- We have $\text{ERC}(S) \Rightarrow \text{NSP}_1(S)$
  - BP succeeds in more cases than OMP! (...after s steps. For exact recovery)
  - At the end of the day:

$$\text{ERC}(s) \Rightarrow \text{NSP}_1(s) \Rightarrow \text{NSP}_p(s) \text{ with } p \leq 1$$

Outline

Why coherence?

- ERC(s) (resp. NSP$_p$(s)) is true if ERC(S) (resp NSP$_p$(S)) is true **for all** S

$\rightarrow$ Checking them is combinatorial!

- Is there a more convenient criterion?

$\rightarrow$ Yes, coherence (which you have already seen in T. Guyard and L. Le Magoarou's course).

Why coherence?

- ERC(s) (resp. NSP$_p$(s)) is true if ERC(S) (resp NSP$_p$(S)) is true **for all** S

$\rightarrow$ Checking them is combinatorial!

- Is there a more convenient criterion?

$\rightarrow$ Yes, coherence (which you have already seen in T. Guyard and L. Le Magoarou's course).

Coherence

Take M **with** ℓ_2 **normalized columns**. The coherence of M is

$$\mu = \max_{i \neq j} |m_i^\top m_j| \tag{4}$$

- Intuitively, although M is "fat", we want its columns to be *as less correlated as possible*, to be "distinguishible" within the mix $y = \sum_{i \in S} m_i x_i$
- $\mu \leq 1$ by Cauchy-Schwartz

Babel function

The coherence itself does not take into account sparsity size. We can look at the more precise notion:

Babel function

$$\mu_1(s) = \max_i \max_{S \text{ of size } s, i \notin S} \left\{ \sum_{j \in S} |m_i^\top m_j| \right\} \tag{5}$$

- $\mu_1(1) = \mu$
- $\mu \leq \mu_1(s) \leq s\mu$

Computational complexity

We have seen that NSP/ERC were costly to compute. What about coherence?

Computational complexity

We have seen that NSP/ERC were costly to compute. What about coherence?

- Computing μ
  - Computing the Gram matrix $G = M^\top M$: $O(mn^2)$
  - Finding the maximum off-diagonal term : $O(n^2)$

Computational complexity

We have seen that NSP/ERC were costly to compute. What about coherence?

- Computing μ
  - Computing the Gram matrix $G = M^\top M$: $O(mn^2)$
  - Finding the maximum off-diagonal term : $O(n^2)$

- Computing $\mu_1(s)$
  - Computing the Gram matrix $G = M^\top M$: $O(mn^2)$
  - For each row, find the s largest term $O(n \log s)$
  - Finding the maximum among their sum : $O(sn)$

Some properties of the coherence

(recall we take M with ℓ_2-normalized columns)

- (F&R, Thm 5.7):

$$\mu \geq \sqrt{\frac{n-m}{m(n-1)}} \sim \frac{1}{\sqrt{m}}$$

  when $m \ll n$. Equality holds iff:

  - M is **equiangular**: $|m_i^\top m_j| = c$ constant $\forall i \neq j$
  - M is a **tight frame**: $MM^\top \propto Id_m$

Some properties of the coherence

(recall we take M with ℓ_2-normalized columns)

- (F&R, Thm 5.7):

$$\mu \geq \sqrt{\frac{n-m}{m(n-1)}} \sim \frac{1}{\sqrt{m}}$$

  when $m \ll n$. Equality holds iff:

  - M is **equiangular**: $|m_i^\top m_j| = c$ constant $\forall i \neq j$
  - M is a **tight frame**: $MM^\top \propto Id_m$

- (F&R, Thm 5.8):

$$\mu_1(s) \geq s\sqrt{\frac{n-m}{m(n-1)}}$$

  whenever $s \leq \sqrt{n-1}$. Equality holds in the same conditions.

Some properties of the coherence

(recall we take M with ℓ_2-normalized columns)

- (F&R, Thm 5.7):

$$\mu \geq \sqrt{\frac{n-m}{m(n-1)}} \sim \frac{1}{\sqrt{m}}$$

  when $m \ll n$. Equality holds iff:

  - M is **equiangular**: $|m_i^\top m_j| = c$ constant $\forall i \neq j$
  - M is a **tight frame**: $MM^\top \propto Id_m$

- (F&R, Thm 5.8):

$$\mu_1(s) \geq s\sqrt{\frac{n-m}{m(n-1)}}$$

  whenever $s \leq \sqrt{n-1}$. Equality holds in the same conditions.

- (F&R Prop 5.13): For each prime number $m \geq 5$, there is an explicit **complex** matrix with coherence $\mu = \frac{1}{\sqrt{m}}$.

Guarantees for OMP and BP

F&R, Thm 5.14, 5.15

Let M with normalized columns. If

$$\mu_1(s) + \mu_1(s-1) < 1$$

then every s-sparse vector x is recovered from $y = Mx$

- after at most s iteration of OMP
- via Basis Pursuit (ℓ_1 minimization)

Guarantees for OMP and BP

F&R, Thm 5.14, 5.15

Let M with normalized columns. If

$$\mu_1(s) + \mu_1(s-1) < 1$$

then every s-sparse vector x is recovered from $y = Mx$

- after at most s iteration of OMP
- via Basis Pursuit (ℓ_1 minimization)

- Since $\mu_1(s) \leq s\mu$, this condition is implied by the stronger but more classical condition

$$\mu \leq \frac{1}{2s-1}$$

  (which you have seen before)
- The proof uses ERC(s) and NSP$_1$(s)...

Guarantees for IHT

F&R, Thm 5.17

Let M with normalized columns. If

$$2\mu_1(s) + \mu_1(s-1) < 1$$

then every s-sparse vector x is recovered from $y = Mx$ after at most s iteration of IHT.

Guarantees for IHT

F&R, Thm 5.17

Let M with normalized columns. If

$$2\mu_1(s) + \mu_1(s-1) < 1$$

then every s-sparse vector x is recovered from $y = Mx$ after at most s iteration of IHT.

- This condition is implied by

$$\mu \leq \frac{1}{3s-1}$$

- stronger assumptions than OMP or BP !

Quadratic bottleneck

Recall that we have seen that

$$\mu \geq \sqrt{\frac{n-m}{m(n-1)}} \sim \frac{1}{\sqrt{m}}$$

and that the recovery condition reads

$$\mu < \frac{1}{2s-1}$$

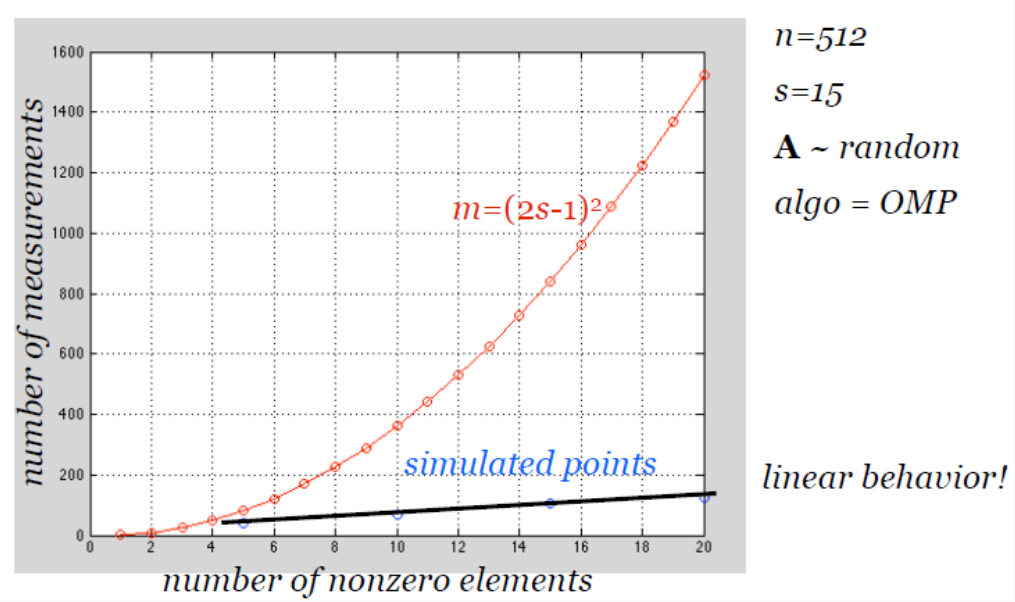
Quadratic Bottleneck

Hence, the condition yields

$$m \gtrsim O(s^2)$$

i.e. a **quadratic rate**. This is fine, but we can do **much better**: m **linear** in s up to log factors.

$\rightarrow$ For coherence-based methods, this condition is *tight*. This is the **quadratic bottleneck**. We need new tools!



Outline

The restricted isometry property (RIP) : definition

Definition (RIP)

Let $\epsilon > 0$, $s \in \mathbb{N}$. A matrix $M \in \mathbb{R}^{m,n}$ with $m \leq n$ is (ϵ, s)-RIP if

$$\forall x \in \Sigma_s,\ (1-\epsilon)||x||_2^2 \leq ||Mx||_2^2 \leq (1+\epsilon)||x||_2^2 \tag{6}$$

The restricted isometry property (RIP) : definition

Definition (RIP)

Let $\epsilon > 0$, $s \in \mathbb{N}$. A matrix $M \in \mathbb{R}^{m,n}$ with $m \leq n$ is (ϵ, s)-RIP if

$$\forall x \in \Sigma_s,\ (1-\epsilon)||x||_2^2 \leq ||Mx||_2^2 \leq (1+\epsilon)||x||_2^2 \tag{6}$$

- M "preserves" the Euclidean norm **of s-sparse vectors** $\rightarrow$ it is "almost" an isometry (hence the name)
- $M \in \mathbb{R}^{m,n}$ with $m \ll n$ is (ϵ, s)-RIP if

$$\forall x \in \Sigma_s \setminus \{0\},\ \left| \frac{||Mx||_2^2 - ||x||_2^2}{||x||_2^2} \right| \leq \epsilon \tag{7}$$

RIP constant

- the "RIP constant" ϵ_s of order s is the smallest constant ϵ such that (ϵ, s)-RIP holds: $...\epsilon_s \leq \epsilon_{s+1}...$
- If M has normalized columns, $\epsilon_1 = 0$, $\epsilon_2 = \mu$, $\epsilon_s \leq \mu_1(s-1)$ [F&R Prop 6.2]

RIP constant

- the "RIP constant" ϵ_s of order s is the smallest constant ϵ such that (ϵ, s)-RIP holds: $...\epsilon_s \leq \epsilon_{s+1}...$
- If M has normalized columns, $\epsilon_1 = 0$, $\epsilon_2 = \mu$, $\epsilon_s \leq \mu_1(s-1)$ [F&R Prop 6.2]

F&R, thm 6.8

One has

$$m \geq c\frac{s}{\epsilon_s^2}$$

provided $n \geq Cm$ and $\epsilon_s \leq \epsilon^*$, where c, C, ϵ^* are constant that only depend on each other.

$\rightarrow$ the RIP constant, sparsity, and m are linked.

Recovery with the RIP: ℓ_0 and BP

Note that, like EsR, we do not act on s columns of M, but at least $2s$, since we are looking at *differences* of s-sparse vectors.

Recovery with the RIP: ℓ_0 and BP

Note that, like EsR, we do not act on s columns of M, but at least $2s$, since we are looking at *differences* of s-sparse vectors.

- Exo: which $(?, ?)$-RIP implies EsR, aka ℓ_0 recovery?

Recovery with the RIP: ℓ_0 and BP

Note that, like EsR, we do not act on s columns of M, but at least $2s$, since we are looking at *differences* of s-sparse vectors.

- Exo: which $(?, ?)$-RIP implies EsR, aka ℓ_0 recovery?

Robustness and stability, F&R thm 6.13

If

$$\epsilon_{2s} < \frac{4}{\sqrt{41}} \approx 0.62$$

then M satisfies the ℓ_2-robust $\mathrm{NSP}_1(s)$ with constant ρ, τ that only depends on ϵ_{2s}.

As a consequence, we have all the NSP results of the previous section.

Recovery with the RIP: IHT

Theorem (Optimality of IHT for RIP matrices)

If M is $(\epsilon, 3s)$-RIP, then

$$\|x^{l+1} - x\| \leq 2\epsilon\|x^l - x\| \leq \ldots \leq (2\epsilon)^{l+1}\|x^0 - x\| \tag{8}$$

where x^l are the iterates of IHT.

Recovery with the RIP: IHT

Theorem (Optimality of IHT for RIP matrices)

If M is $(\epsilon, 3s)$-RIP, then

$$\|x^{l+1} - x\| \leq 2\epsilon\|x^l - x\| \leq \ldots \leq (2\epsilon)^{l+1}\|x^0 - x\| \tag{8}$$

where x^l are the iterates of IHT. In particular, if $\epsilon_{3s} < \frac{1}{2}$, the iterates x^l converge to x

Recovery with the RIP: IHT

Theorem (Optimality of IHT for RIP matrices)

If M is $(\epsilon, 3s)$-RIP, then

$$\|x^{l+1} - x\| \leq 2\epsilon\|x^l - x\| \leq \ldots \leq (2\epsilon)^{l+1}\|x^0 - x\| \tag{8}$$

where x^l are the iterates of IHT. In particular, if $\epsilon_{3s} < \frac{1}{2}$, the iterates x^l converge to x

- Here it's not "in s iterations"!
- Stability and Robustness: see F&R Thm 6.21. Condition: $\epsilon_{6s} < 1/\sqrt{3}$.

Recovery with the RIP: OMP

Here it becomes complicated... No simple ERC!

Recovery with the RIP: OMP

Here it becomes complicated... No simple ERC!

F&R thm 6.25

If

$$\epsilon_{13s} < 1/6$$

then the sequence x^l produced by OMP from $y = Ax + e$ satisfies, for any S,

$$\|y - Ax^{12s}\|_2 \lesssim \|Ax_{S^c} + e\|_2$$

Not even recovery of x!

Recovery with the RIP: OMP

Here it becomes complicated... No simple ERC!

F&R thm 6.25

If

$$\epsilon_{13s} < 1/6$$

then the sequence x^l produced by OMP from $y = Ax + e$ satisfies, for any S,

$$\|y - Ax^{12s}\|_2 \lesssim \|Ax_{S^c} + e\|_2$$

Not even recovery of x!
But if

$$\epsilon_{26s} < 1/6$$

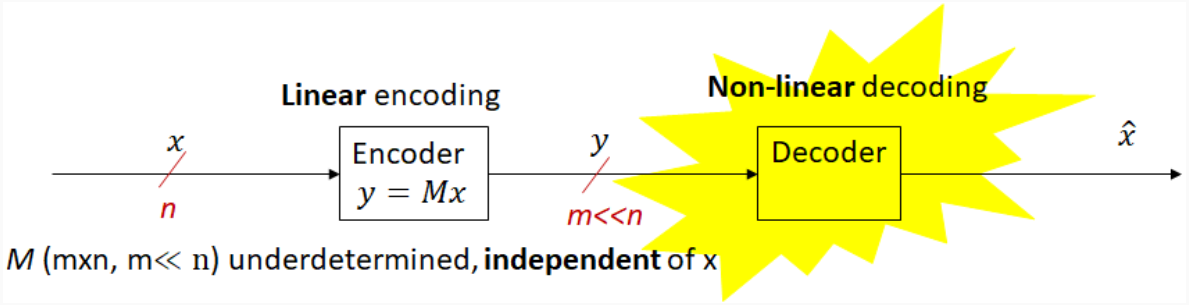
then

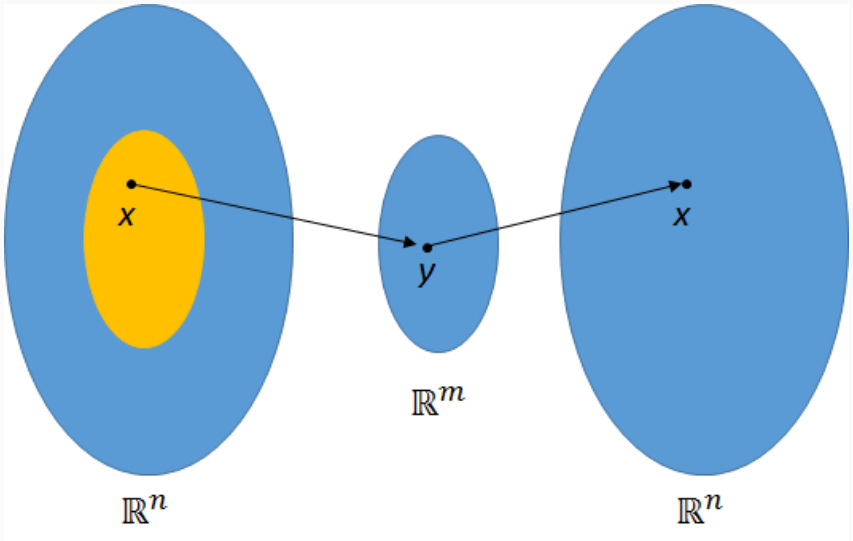
$$\|x - x^{24s}\|_1 \lesssim \sigma_s(x)_1 + \sqrt{s}\|e\|_2$$

In practice, generally much better than that! No need to do $24s$ iterations...

Recovering with random matrices?

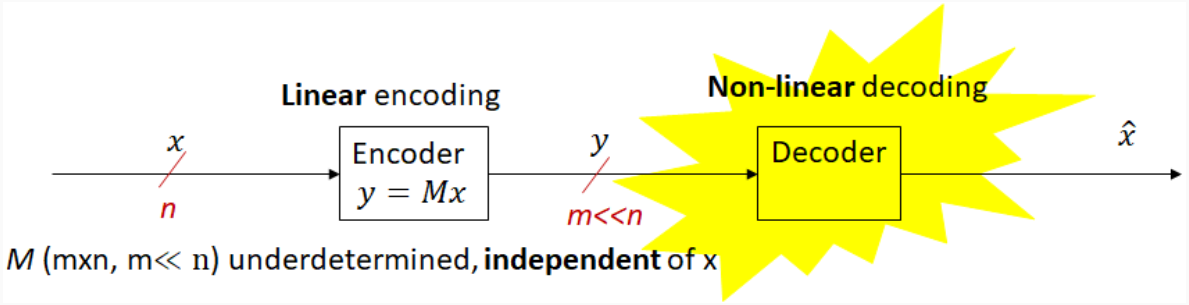
Compressive sensing: summary of what seen so far

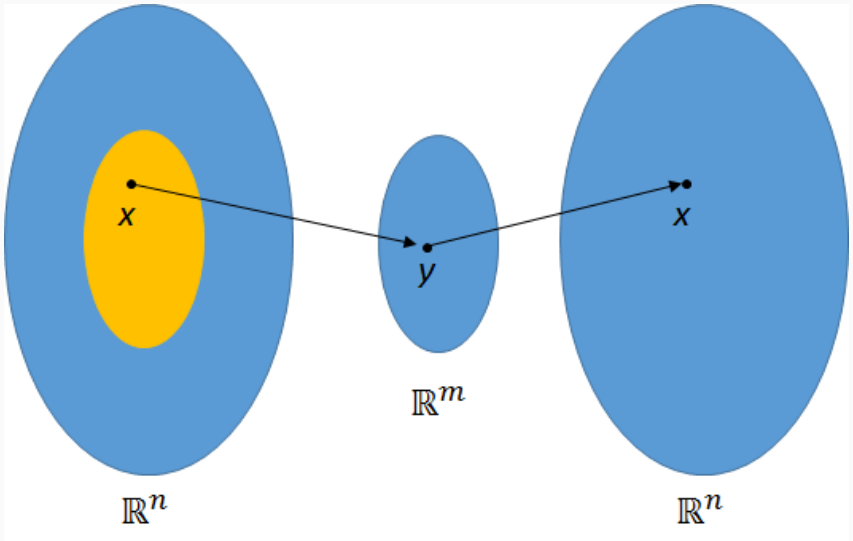




How do we choose a "good" matrix M with $m \ll n$?

Compressive sensing: summary of what seen so far

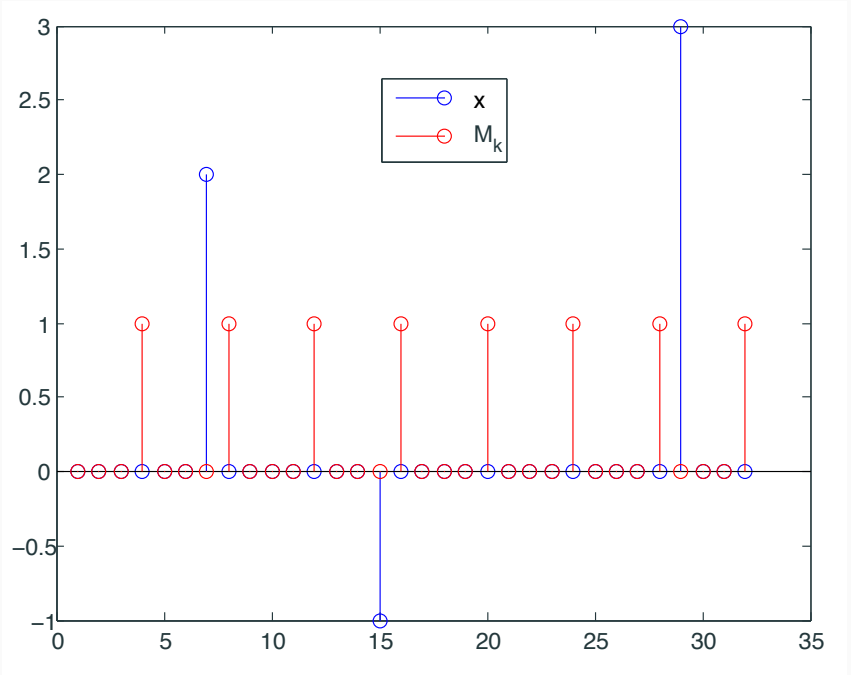




How do we choose a "good" matrix M with $m \ll n$?

Coherence leads to quadratic bottleneck, **can we have the RIP with (almost) linear number of measurements?**

Sensing matrices that are not good



Vector y is all zero!

$\rightarrow$ If x sparse, M must be **non-sparse**

Intuition

Recall **coherence** (for ℓ_2-normalized matrices), which must be as small as possible

$$\mu = \max_{i \neq j} |m_i^\top m_j|$$

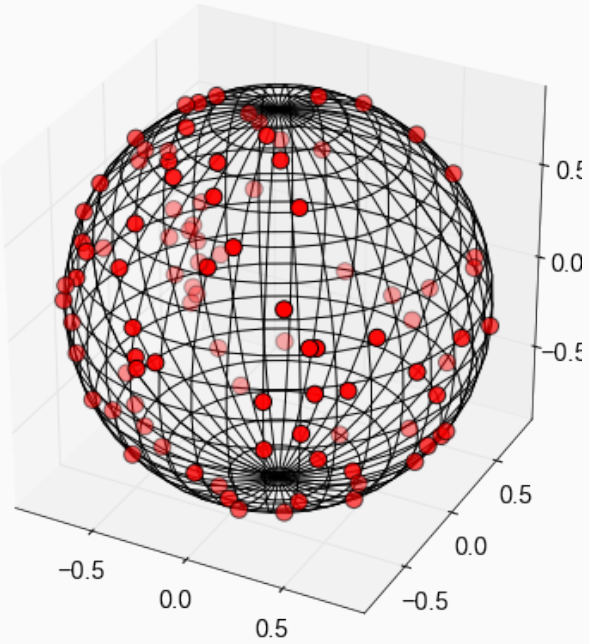
Intuition

Recall **coherence** (for ℓ_2-normalized matrices), which must be as small as possible

$$\mu = \max_{i \neq j} |m_i^\top m_j|$$

$\rightarrow$ we need n vectors that are "well-spread" on the m-dimensional sphere.

$\rightarrow$ Deterministic constructions are possible, but complicated... We will rely on **random matrices**, and show the desired property **with high probability**



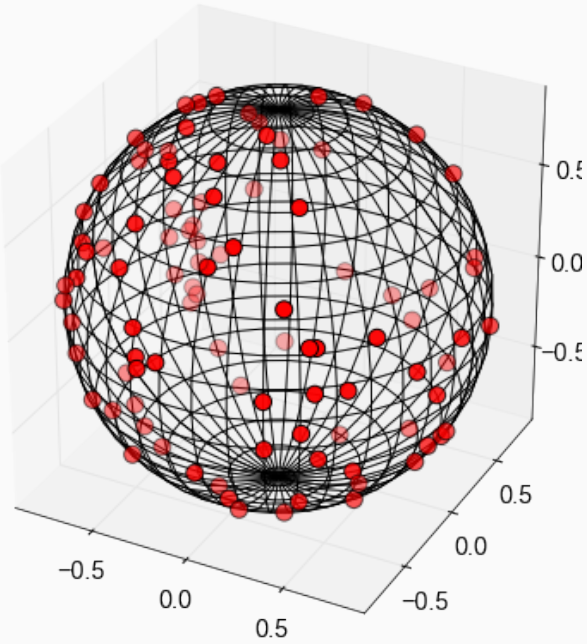
Intuition

Recall **coherence** (for ℓ_2-normalized matrices), which must be as small as possible

$$\mu = \max_{i \neq j} |m_i^\top m_j|$$

$\rightarrow$ we need n vectors that are "well-spread" on the m-dimensional sphere.

$\rightarrow$ Deterministic constructions are possible, but complicated... We will rely on **random matrices**, and show the desired property **with high probability**



$\rightarrow$ recall: coherence = quadratic bottleneck. We will study NSP/RIP instead.

$\rightarrow$ you have seen polynomial/exponential systems, that verify EsR. But not enough for RIP!

Gaussian matrices

One of the most used/convenient construction is **Gaussian matrices**. All entries are independent:

$$m_{ij} \sim \mathcal{N}(0, 1/m)$$

Gaussian matrices

One of the most used/convenient construction is **Gaussian matrices**. All entries are independent:

$$m_{ij} \sim \mathcal{N}(0, 1/m)$$

Variance is $1/m$ to have proper scaling

$$\mathbb{E}\|Mx\|_2^2 = \|x\|_2^2$$

Basis for proving the RIP: "show that $\|Mx\|_2^2$ is close to its expectation with high probability (for all sparse vectors)" see after

RIP for Gaussian matrices

F&R Thm 9.27

Let M be a Gaussian matrix (with variance $1/m$). For $\epsilon, \delta > 0$, assume

$$m \geq 2\epsilon^{-2}(s \log(en/s) + \log(2/\delta))$$

Then, with probability at least $1 - \delta$,

$$\delta_s \leq 2\left(1 + \frac{1}{\sqrt{2\log(en/s)}}\right)\epsilon + \left(1 + \frac{1}{\sqrt{2\log(en/s)}}\right)^2 \epsilon^2$$

RIP for Gaussian matrices

F&R Thm 9.27

Let M be a Gaussian matrix (with variance $1/m$). For $\epsilon, \delta > 0$, assume

$$m \geq 2\epsilon^{-2}(s \log(en/s) + \log(2/\delta))$$

Then, with probability at least $1 - \delta$,

$$\delta_s \leq 2\left(1 + \frac{1}{\sqrt{2\log(en/s)}}\right)\epsilon + \left(1 + \frac{1}{\sqrt{2\log(en/s)}}\right)^2 \epsilon^2$$

- Examine all the rates in m: which are fast, which are slow?

RIP for Gaussian matrices

F&R Thm 9.27

Let M be a Gaussian matrix (with variance $1/m$). For $\epsilon, \delta > 0$, assume

$$m \geq 2\epsilon^{-2}(s \log(en/s) + \log(2/\delta))$$

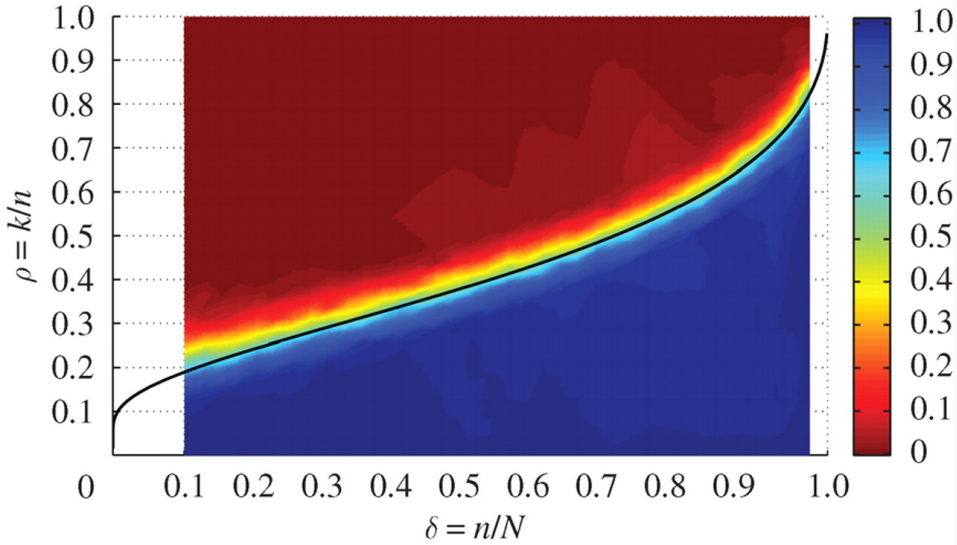
Then, with probability at least $1 - \delta$,

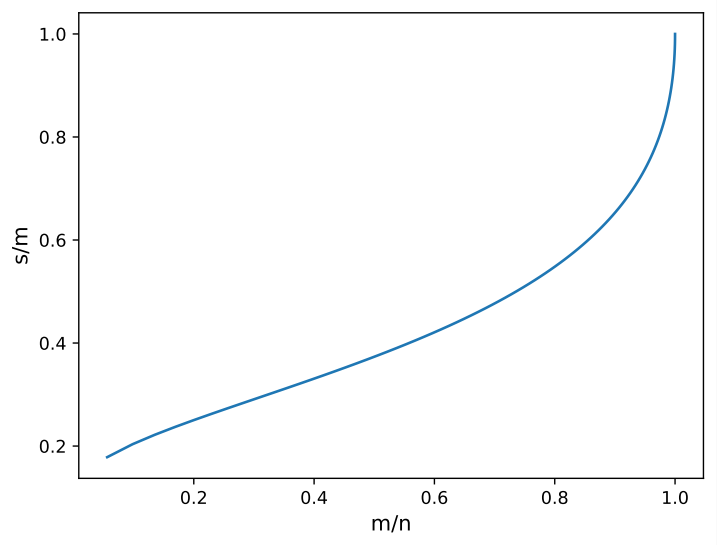
$$\delta_s \leq 2\left(1 + \frac{1}{\sqrt{2\log(en/s)}}\right)\epsilon + \left(1 + \frac{1}{\sqrt{2\log(en/s)}}\right)^2 \epsilon^2$$

- Examine all the rates in m: which are fast, which are slow?
- With more or less the same rate, one can also prove the robust NSP (Thm 9.29)

Phase transition

Almost linear rate $m \sim s \log(en/s)$. **This is the well-known rate of Compressed Sensing**.





Exo: produce the theoretical curve above (right)

Rademacher and subgaussian matrices

Gaussian rv may not be easily found in physical measurement systems!
More "natural": Rademacher variables

$$m_{ij} \sim \text{Unif}(\{1/\sqrt{m}, -1/\sqrt{m}\})$$

$\rightarrow$ much, **much** more convenient on a computer! (just addition substraction)

Rademacher and subgaussian matrices

Gaussian rv may not be easily found in physical measurement systems!
More "natural": Rademacher variables

$$m_{ij} \sim \text{Unif}(\{1/\sqrt{m}, -1/\sqrt{m}\})$$

$\rightarrow$ much, **much** more convenient on a computer! (just addition substraction)

More generally, *centered* "subgaussian" random variables w/ parameters κ, β (includes Gaussian and bounded rv like rademacher)

$$\mathbb{P}(|m_{ij}| \geq t) \leq \beta e^{-\kappa t^2}$$

RIP for subgaussian matrices

F&R Thm 9.2

Let M be a subgaussian matrix of centered rv with variance $1/m$. We have $\epsilon_s \leq \epsilon$ with probability at least $1 - \delta$, provided

$$m \gtrsim \epsilon^{-2}(s \log(en/s) + \log(2/\delta))$$

RIP for subgaussian matrices

F&R Thm 9.2

Let M be a subgaussian matrix of centered rv with variance $1/m$. We have $\epsilon_s \leq \epsilon$ with probability at least $1 - \delta$, provided

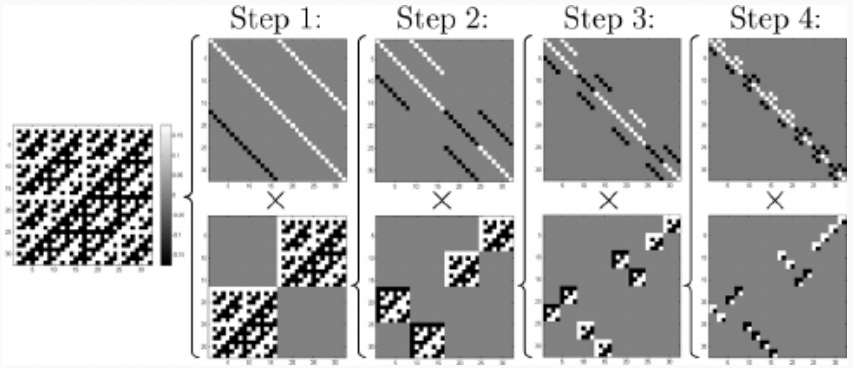
$$m \gtrsim \epsilon^{-2}(s \log(en/s) + \log(2/\delta))$$

- Basically the same result than Gaussian random matrices

Beyond iid Matrices

Random matrices with iid entries are convenient theoretically, but **not really in practice**. **Active research field!**

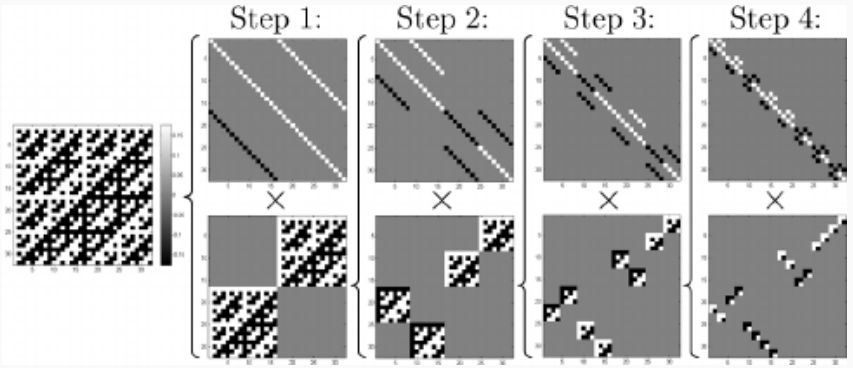
- Dense, unstructured matrices (costly!). Could we have "FFT"-like instead? Maybe, but structure is bad for the RIP! (trade-off)



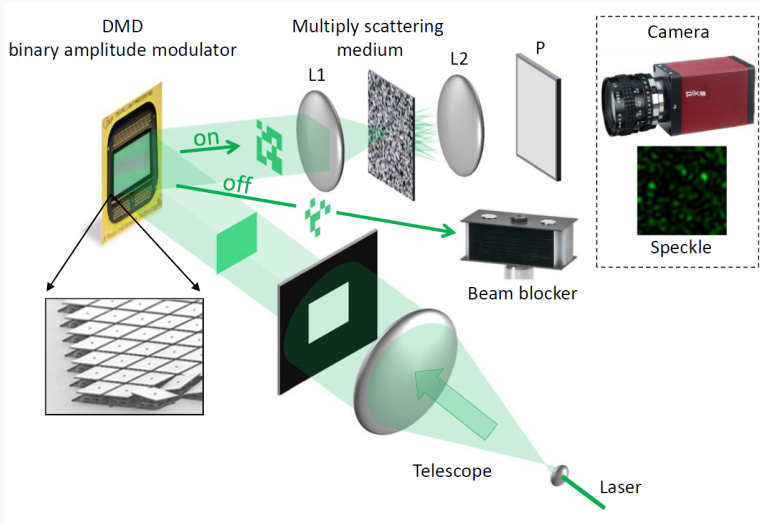
Beyond iid Matrices

Random matrices with iid entries are convenient theoretically, but **not really in practice**. **Active research field!**

- Dense, unstructured matrices (costly!). Could we have "FFT"-like instead? Maybe, but structure is bad for the RIP! (trade-off)



- Multiplication by large matrices is costly . Can we have "physical" random transforms? Maybe optics...



Concentration inequalities and proving the RIP

Concentration inequalities

We have seen that, with iid centered rv with variance $1/m$, we have $\mathbb{E}\|Mx\|_2^2 = \|x\|_2^2$. *Exo: show it*.

Concentration inequalities

We have seen that, with iid centered rv with variance $1/m$, we have $\mathbb{E}\|Mx\|_2^2 = \|x\|_2^2$. *Exo: show it.*

$$\mathbb{E}\|Mx\|_2^2 = \mathbb{E}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} m_{ij}x_j\right)^2 = \mathbb{E}\sum_{i=1}^{m}\sum_{j,k=1}^{n} m_{ij}m_{ik}x_j x_k$$

$$= \sum_{i=1}^{m}\sum_{j,k=1}^{n}\mathbb{E}(m_{ij}m_{ik})x_j x_k = \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbb{E}(m_{ij}^2)x_j^2$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{m}\frac{1}{m}x_j^2 = \sum_{j} x_j^2 = \|x\|_2^2$$

Concentration inequalities

We have seen that, with iid centered rv with variance $1/m$, we have $\mathbb{E}\|Mx\|_2^2 = \|x\|_2^2$. *Exo: show it.*

$$\mathbb{E}\|Mx\|_2^2 = \mathbb{E}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} m_{ij}x_j\right)^2 = \mathbb{E}\sum_{i=1}^{m}\sum_{j,k=1}^{n} m_{ij}m_{ik}x_j x_k$$
$$= \sum_{i=1}^{m}\sum_{j,k=1}^{n}\mathbb{E}(m_{ij}m_{ik})x_j x_k = \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbb{E}(m_{ij}^2)x_j^2$$
$$= \sum_{j=1}^{n}\sum_{i=1}^{m}\frac{1}{m}x_j^2 = \sum_{j} x_j^2 = \|x\|_2^2$$

The main proof technique for the RIP is to **bound the deviation of $\|Mx\|_2^2$ from its expectation** with high probability. This is done with **concentration inequalities**.

Basic concentration inequalities

Markov's inequality: Given a non-negative random variable X with finite mean

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}, \quad \forall t > 0. \qquad \text{Decay in } \mathcal{O}(\frac{1}{t})$$

Basic concentration inequalities

Markov's inequality: Given a non-negative random variable X with finite mean

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}, \quad \forall t > 0. \qquad \text{Decay in } \mathcal{O}(\frac{1}{t})$$

Chebyshev's inequality: Given a random variable X with finite variance

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq t) \leq \frac{\text{var}(X)}{t^2}, \quad \forall t > 0. \qquad \text{Decay in } \mathcal{O}(\frac{1}{t^2})$$

Basic concentration inequalities

Markov's inequality: Given a non-negative random variable X with finite mean

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}, \quad \forall t > 0. \qquad \text{Decay in } \mathcal{O}(\frac{1}{t})$$

Chebyshev's inequality: Given a random variable X with finite variance

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq t) \leq \frac{\text{var}(X)}{t^2}, \quad \forall t > 0. \qquad \text{Decay in } \mathcal{O}(\frac{1}{t^2})$$

Chernoff bound: Given a random variable X with mean μ

$$\mathbb{P}(X - \mu \geq t) \leq \frac{\mathbb{E}[e^{\lambda|X-\mu|}]}{e^{\lambda t}}, \quad \forall t, \lambda > 0. \quad \text{Decay in } \mathcal{O}(e^{-\lambda t})$$

as soon as $\mathbb{E}[e^{\lambda|X-\mu|}]$ is finite.

Basic concentration inequalities

Markov's inequality: Given a non-negative random variable X with finite mean

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}, \quad \forall t > 0. \qquad \text{Decay in } \mathcal{O}(\frac{1}{t})$$

Chebyshev's inequality: Given a random variable X with finite variance

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq t) \leq \frac{\text{var}(X)}{t^2}, \quad \forall t > 0. \qquad \text{Decay in } \mathcal{O}(\frac{1}{t^2})$$

Chernoff bound: Given a random variable X with mean μ

$$\mathbb{P}(X - \mu \geq t) \leq \frac{\mathbb{E}[e^{\lambda|X-\mu|}]}{e^{\lambda t}}, \quad \forall t, \lambda > 0. \quad \text{Decay in } \mathcal{O}(e^{-\lambda t})$$

as soon as $\mathbb{E}[e^{\lambda|X-\mu|}]$ is finite.

Also see Hoeffding's inequality, Bernstein inequality, McDiarmid inequality...

[book Boucheron Lugosi Massart]

Concentration inequality

Theorem (Concentration of Gaussian Matrices)

Let $x \in \mathbb{R}^n$. For a Gaussian matrix M,

$$\forall 0 \leq \epsilon \leq 3, \quad \mathbb{P}_M \left(\left| \frac{||Mx||_2^2}{||x||_2^2} - 1 \right| > \epsilon \right) \leq 2e^{-\frac{m\epsilon^2}{6}} \tag{9}$$

Concentration inequality

Theorem (Concentration of Gaussian Matrices)

Let $x \in \mathbb{R}^n$. For a Gaussian matrix M,

$$\forall 0 \leq \epsilon \leq 3, \quad \mathbb{P}_M \left(\left| \frac{||Mx||_2^2}{||x||_2^2} - 1 \right| > \epsilon \right) \leq 2e^{-\frac{m\epsilon^2}{6}} \tag{9}$$

Equivalently: with probability $1 - \delta$, we have

$$(1 - \epsilon)\|x\|_2^2 \leq \|Mx\|_2^2 \leq (1 + \epsilon)\|x\|_2^2$$

provided $m \gtrsim \epsilon^{-2} \log(1/\delta)$.

Concentration inequality

Theorem (Concentration of Gaussian Matrices)

Let $x \in \mathbb{R}^n$. For a Gaussian matrix M,

$$\forall 0 \leq \epsilon \leq 3, \quad \mathbb{P}_M \left(\left| \frac{||Mx||_2^2}{||x||_2^2} - 1 \right| > \epsilon \right) \leq 2e^{-\frac{m\epsilon^2}{6}} \tag{9}$$

Equivalently: with probability $1 - \delta$, we have

$$(1 - \epsilon)\|x\|_2^2 \leq \|Mx\|_2^2 \leq (1 + \epsilon)\|x\|_2^2$$

provided $m \gtrsim \epsilon^{-2} \log(1/\delta)$.

We have the RIP **for one vector** x. We need this **for all s-sparse vectors simultaneously!** This will be done using union bounds and covering numbers of finite-dimensional compact sets (!)

Union bound for finite set

Union Bound

For two events A, B, it is obvious that

$$\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$$

Union bound for finite set

Union Bound

For two events A, B, it is obvious that

$$\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$$

Exo: how to apply this to obtain concentration for **several** *vectors?*

Union bound for finite set

Union Bound

For two events A, B, it is obvious that

$$\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$$

Exo: how to apply this to obtain concentration for **several** *vectors?*

We obtain the so-called Johnson-Lindenstrauss lemma.

Lemma (Johnson-Lindenstrauss)

For a Gaussian matrix M, let $0 \leq \epsilon \leq 3, \delta > 0$.
Let $\mathcal{Q}$ be a **finite set** *of vectors $\subset \mathbb{R}^n$. If $m \geq \frac{6}{\epsilon^2} \log \frac{2|\mathcal{Q}|}{\delta}$, then*

$$\mathbb{P}_M \left(\sup_{x \in \mathcal{Q}} \left| \frac{||Mx||_2^2}{||x||_2^2} - 1 \right| \leq \epsilon \right) \geq 1 - \delta$$

Covering numbers

Covering numbers

A compact set in dimension d can be covered with $O(r^{-d})$ balls of radius r.

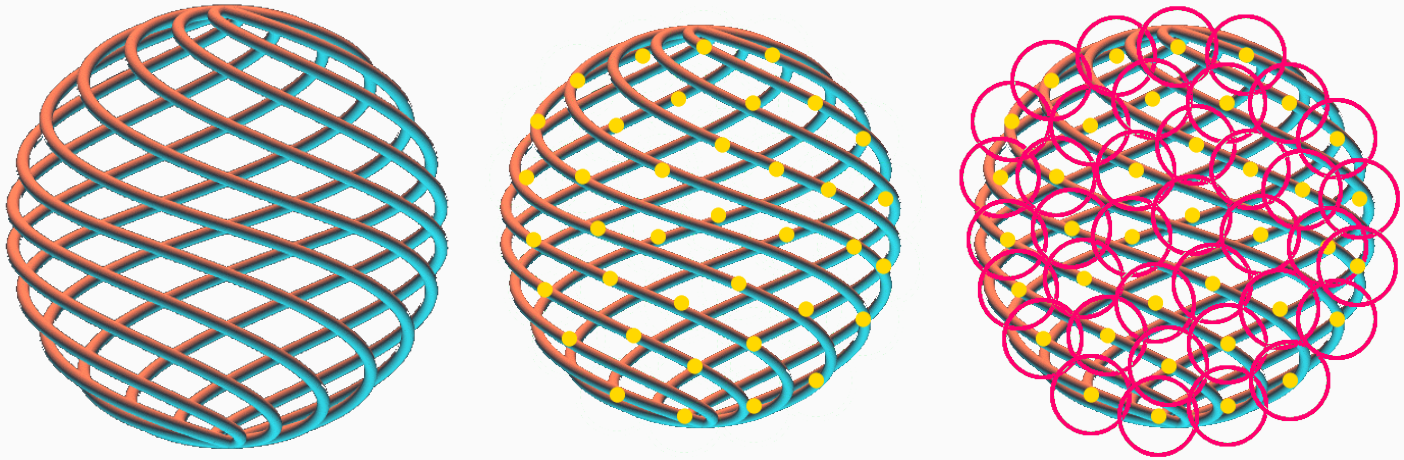
Covering numbers

Covering numbers

A compact set in dimension d can be covered with $O(r^{-d})$ balls of radius r.

Finishing the proof of the RIP *(exo)*:

1. Covering the set $\{x/\|x\|_2, x \in \Sigma_s\}$ with balls of appropriate size



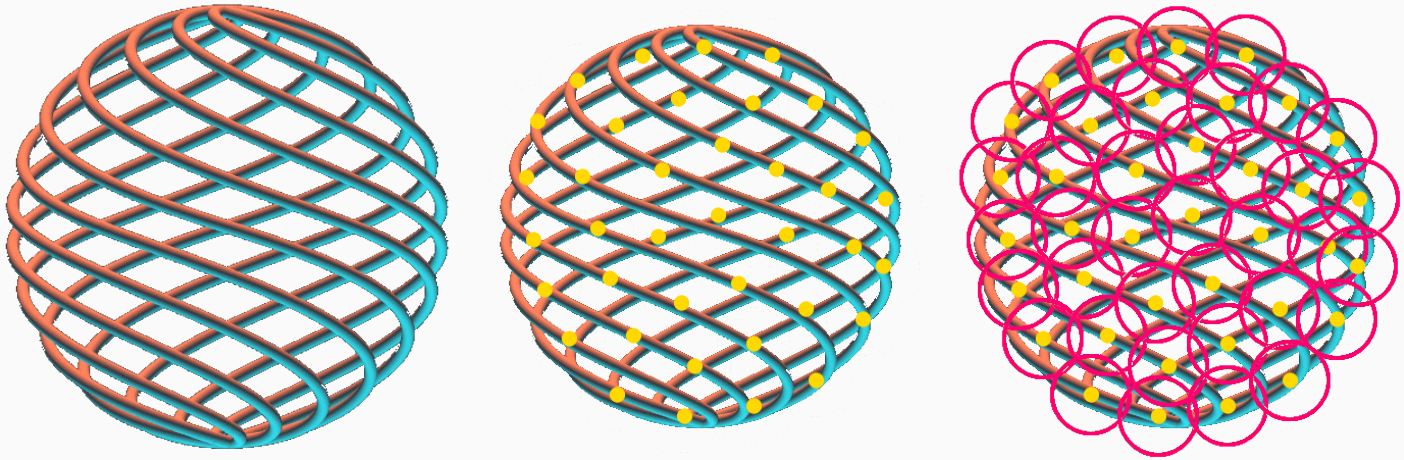
Covering numbers

Covering numbers

A compact set in dimension d can be covered with $O(r^{-d})$ balls of radius r.

Finishing the proof of the RIP *(exo)*:

1. Covering the set $\{x/\|x\|_2, x \in \Sigma_s\}$ with balls of appropriate size
2. Applying the JL lemma for the center of each ball



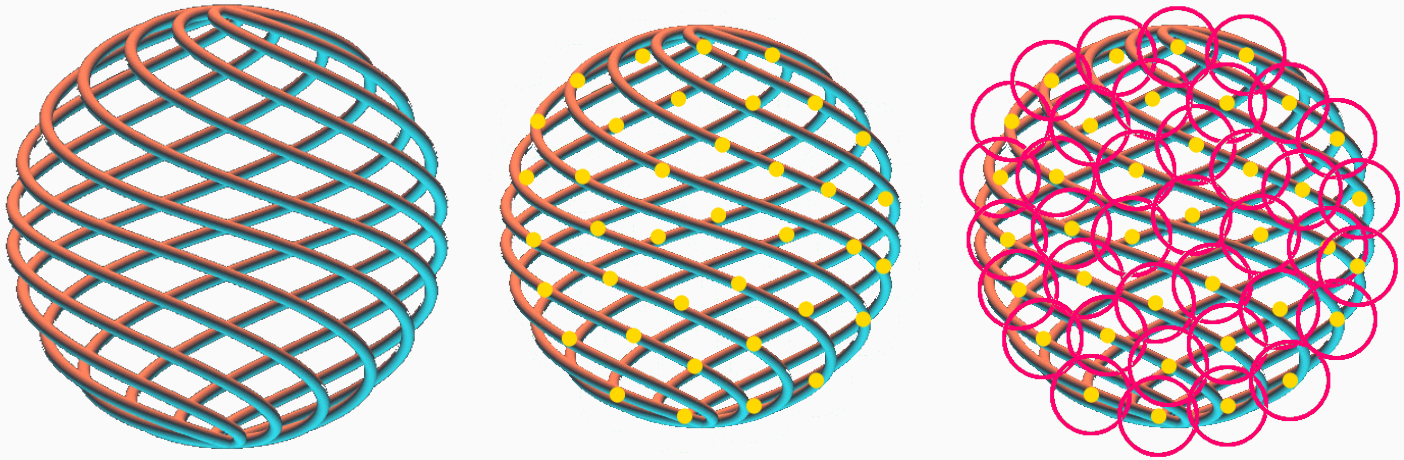
Covering numbers

Covering numbers

A compact set in dimension d can be covered with $O(r^{-d})$ balls of radius r.

Finishing the proof of the RIP *(exo)*:

1. Covering the set $\{x/\|x\|_2, x \in \Sigma_s\}$ with balls of appropriate size
2. Applying the JL lemma for the center of each ball
3. Conclude with triangular inequality.



Beyond Sparsity

Outline

Total variation model

- Reminder: images are sparse in wavelet/DCT basis.

Total variation model

- Reminder: images are sparse in wavelet/DCT basis.

- Other model: images are formed by constant-by-part values.

$\rightarrow$ the **difference of neighboring pixels** is a sparse signal

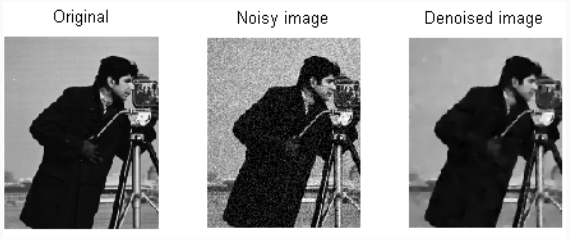
This is the "Rudin-Osher-Fatemi" (ROF) model [1992]. It uses the **total variation**:

$$\|x\|_{TV} = \|\nabla x\|_1 = \sum_{i,j \text{ "neighbours"}} |x_i - x_j|$$

$\rightarrow$ Several possibilities for "neighbours" (4-neightbours, 8-neighbours, graph...) $\rightarrow$ discrete "gradient"

Total variation application

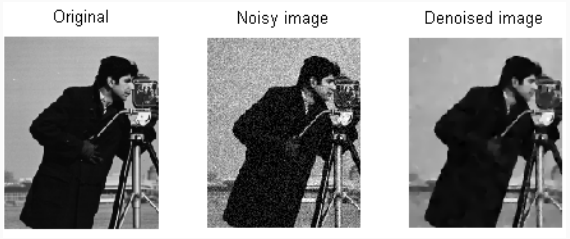
- Denoising: $\min_x \|y - x\|^2 + \lambda \|\nabla x\|_1$



$\rightarrow$ directly proximal operator of TV (... no closed-form)

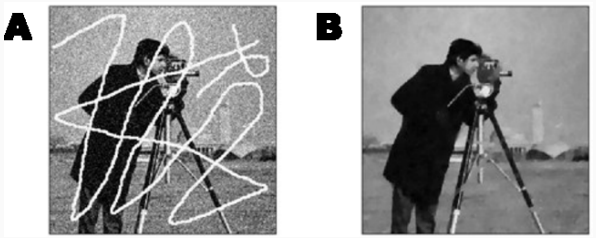
Total variation application

- Denoising: $\min_x \|y - x\|^2 + \lambda \|\nabla x\|_1$



$\rightarrow$ directly proximal operator of TV (... no closed-form)

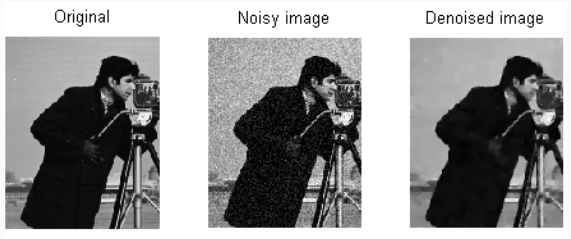
- Inpainting: $\min_x \|y - Mx\|^2 + \lambda \|\nabla x\|_1$



$\rightarrow$ Even more complicated (Prox-GD asks for the computation of proximal operator at each operation)

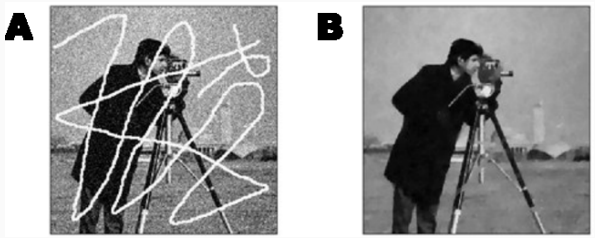
Total variation application

- Denoising: $\min_x \|y - x\|^2 + \lambda \|\nabla x\|_1$



  $\rightarrow$ directly proximal operator of TV (... no closed-form)

- Inpainting: $\min_x \|y - Mx\|^2 + \lambda \|\nabla x\|_1$



  $\rightarrow$ Even more complicated (Prox-GD asks for the computation of proximal operator at each operation)

- There exists powerful convex optimization algorithms (Chambolle-Pock)

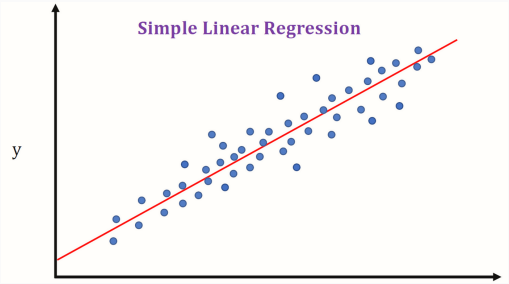
Outline

Linear Regression and Model selection

The term "LASSO" comes from **Statistics**. There, and in Machine Learning, sparsity is used for **model selection**. Just different notations and interpretation!

Linear Regression and Model selection

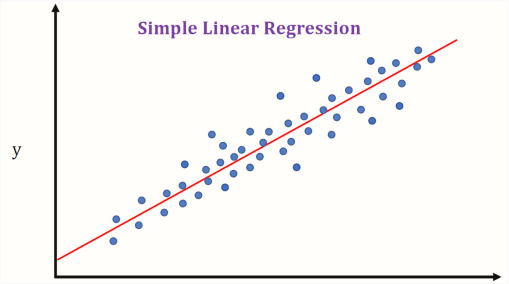
The term "LASSO" comes from **Statistics**. There, and in Machine Learning, sparsity is used for **model selection**. Just different notations and interpretation!



- seek to linearly regress the variable y from high-dimensional covariates $x^1, \ldots, x^d$, but we hypothesize that not all covariates have influence on y.

Linear Regression and Model selection

The term "LASSO" comes from **Statistics**. There, and in Machine Learning, sparsity is used for **model selection**. Just different notations and interpretation!



- seek to linearly regress the variable y from high-dimensional covariates $x^1, \dots, x^d$, but we hypothesize that not all covariates have influence on y.
- We have examples $(y_i, x_i)_{i=1}^n$ with $x_i = [x_i^1, \dots, x_i^d]$, but potentially $d > n$
- We solve:

$$\min_{\beta \in \mathbb{R}^d} \|Y - X^\top \beta\|_2^2 + \lambda \|\beta\|_1$$

  where x_i are the columns of X. β is the regression vector.
- We **never** have exact recovery here. Different type of guarantees...

Group Sparsity

Sometimes we know in advance that **group of covariates** have influence **together** or not.

Group Sparsity

Sometimes we know in advance that **group of covariates** have influence **together** or not.

- Group coordinates of β together: $\beta_{g_1}, \ldots, \beta_{g_p}$ with group $\beta_{g_l} = [\beta_1^l, \ldots, \beta_{i_l}^l]$

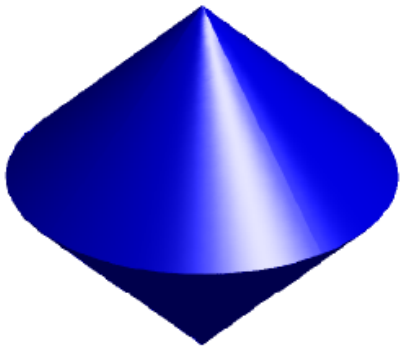
Group Sparsity

Sometimes we know in advance that **group of covariates** have influence **together** or not.

- Group coordinates of β together: $\beta_{g_1}, \ldots, \beta_{g_p}$ with group $\beta_{g_l} = [\beta_1^l, \ldots, \beta_{i_l}^l]$
- Group Lasso: new regularization term

$$\|\beta\|_{gL} = \sum_{l=1}^{p} \|\beta_{g_l}\|_2$$

$\rightarrow$ ℓ_1-norm of ℓ_2-norms of groups. Group of size 1: regular sparsity.



$$\Omega(\beta) = \|\beta_{\{1,2\}}\|_2 + |\beta_3|.$$

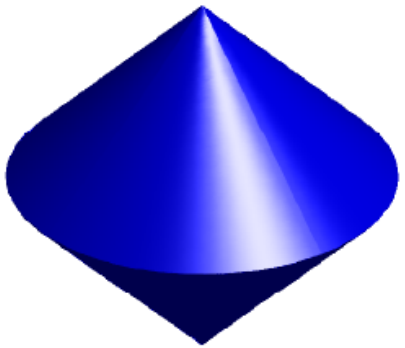
Group Sparsity

Sometimes we know in advance that **group of covariates** have influence **together** or not.

- Group coordinates of β together: $\beta_{g_1}, \ldots, \beta_{g_p}$ with group $\beta_{g_l} = [\beta_1^l, \ldots, \beta_{i_l}^l]$
- Group Lasso: new regularization term

$$\|\beta\|_{gL} = \sum_{l=1}^{p} \|\beta_{g_l}\|_2$$

$\rightarrow \ell_1$-norm of ℓ_2-norms of groups. Group of size 1: regular sparsity.

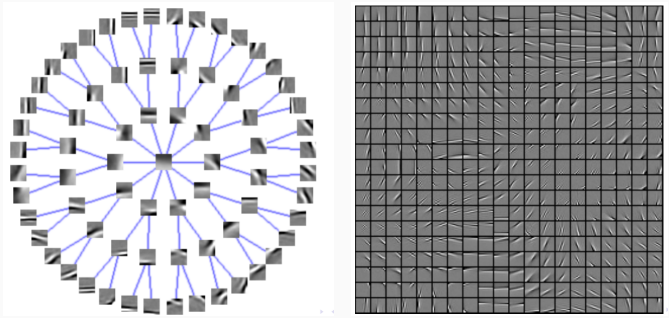


$$\Omega(\beta) = \|\beta_{\{1,2\}}\|_2 + |\beta_3|.$$

- Need different optimization algorithms...

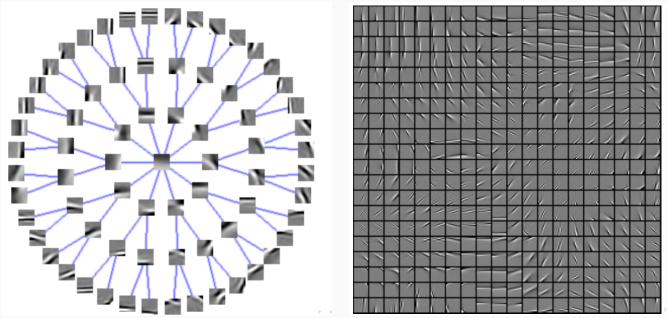
Beyond group sparsity: structured sparsity

- Structured dictionary for image patches

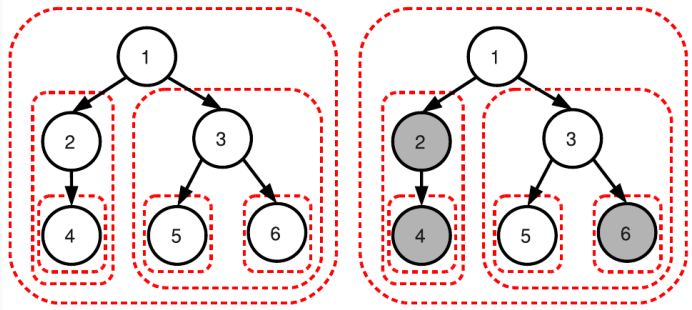


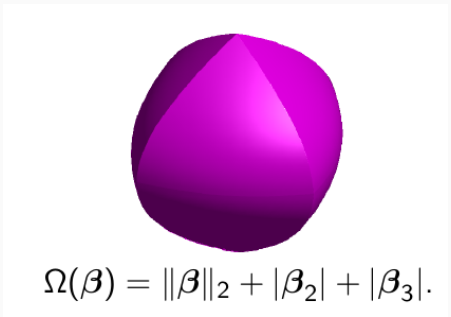
Beyond group sparsity: structured sparsity

- Structured dictionary for image patches

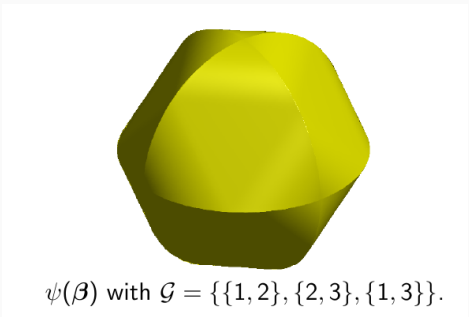


- Hierarchical norm, overlapping groups, union of groups...





$\Omega(\beta) = \|\beta\|_2 + |\beta_2| + |\beta_3|.$



$\psi(\beta)$ with $\mathcal{G} = \{\{1,2\},\{2,3\},\{1,3\}\}$.

Outline

Matrix completion

- **Recommender system**: given users as row and items as columns, fill with preferences. Known preference $\rightarrow$ missing values. Also called "collaborative filtering"

The "Netflix prize": a huge competion (100 480 507 ratings, 480 189 users, 17 770 movies) in 2008, with a 1M dollars prize. Greatly boosted the interest in collaborative filtering...

Matrix completion

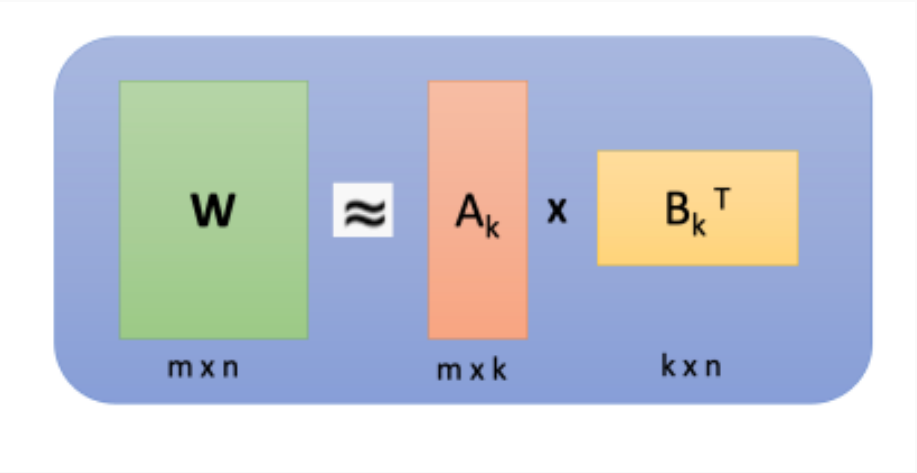
- **Recommender system**: given users as row and items as columns, fill with preferences. Known preference $\rightarrow$ missing values. Also called "collaborative filtering"

The "Netflix prize": a huge competion (100 480 507 ratings, 480 189 users, 17 770 movies) in 2008, with a 1M dollars prize. Greatly boosted the interest in collaborative filtering...

- An instance of **matrix completion** and **missing data inputation**, two important fields

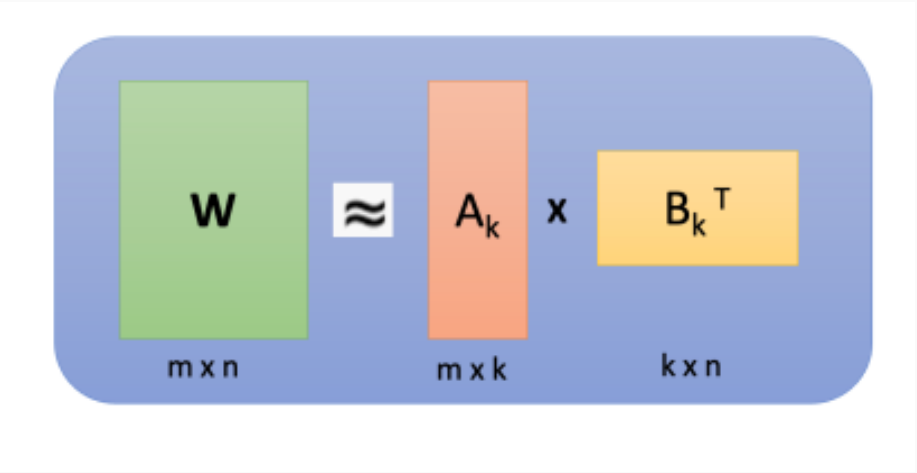
Low-rank

- For such problems, the **low-rank assumption** can seem natural.



Low-rank

- For such problems, the **low-rank assumption** can seem natural.



→ Each **item** i is represented by a "feature vector" $a_i \in \mathbb{R}^k$, each **user** j by a "preference vector" $b_j \in \mathbb{R}^k$, and the score is the scalar product between the two:

$$w_{ij} = a_i^\top b_j$$

Nuclear norm

- As with regural sparsity, choosing in advance the rank k can be problematic: we'd rather prefer a **regularization term!** (like ℓ_0 or ℓ_1)

$$\min_W \| \underbrace{\hat{W}}_{\text{Obs.}} - \underbrace{\mathcal{M}}_{\text{Mask}}(W) \|_F^2 + \lambda \| W \|_?$$

Nuclear norm

- As with regural sparsity, choosing in advance the rank k can be problematic: we'd rather prefer a **regularization term!** (like ℓ_0 or ℓ_1)

$$\min_W \| \underbrace{\hat{W}}_{\text{Obs.}} - \underbrace{\mathcal{M}}_{\text{Mask}}(W) \|_F^2 + \lambda \| W \|_?$$

- For low-rank matrices, it is the pseudo-norm: $\|W\|_0 = \sharp nnz\{\sigma_i(W) \mid 1 \leq i \leq \min(n, m)\}$, where $\sigma_i(W) \geq 0$ are the **singular values** of W.

Nuclear norm

- As with regural sparsity, choosing in advance the rank k can be problematic: we'd rather prefer a **regularization term!** (like ℓ_0 or ℓ_1)

$$\min_W \| \underbrace{\hat{W}}_{\text{Obs.}} - \underbrace{\mathcal{M}}_{\text{Mask}}(W) \|_F^2 + \lambda \| W \|_?$$

- For low-rank matrices, it is the pseudo-norm: $\|W\|_0 = \sharp nnz\{\sigma_i(W) \mid 1 \leq i \leq \min(n, m)\}$, where $\sigma_i(W) \geq 0$ are the **singular values** of W.
- Its convex relaxation is the **nuclear norm**:

$$\|W\|_* = \sum_{i=1}^{\min(n,m)} \sigma_i(W)$$

Nuclear norm

- As with regural sparsity, choosing in advance the rank k can be problematic: we'd rather prefer a **regularization term!** (like ℓ_0 or ℓ_1)

$$\min_W \| \underbrace{\hat{W}}_{\text{Obs.}} - \underbrace{\mathcal{M}}_{\text{Mask}}(W) \|_F^2 + \lambda \| W \|_?$$

- For low-rank matrices, it is the pseudo-norm: $\|W\|_0 = \sharp nnz\{\sigma_i(W) \mid 1 \leq i \leq \min(n, m)\}$, where $\sigma_i(W) \geq 0$ are the **singular values** of W.
- Its convex relaxation is the **nuclear norm**:

$$\|W\|_* = \sum_{i=1}^{\min(n,m)} \sigma_i(W)$$

- Nuclear norm minimization (and variants) is still an active research field. Main problem: **computational complexity**. The SVD is super costly! Many (many) alternative approach...

Beyond Compressed Sensing

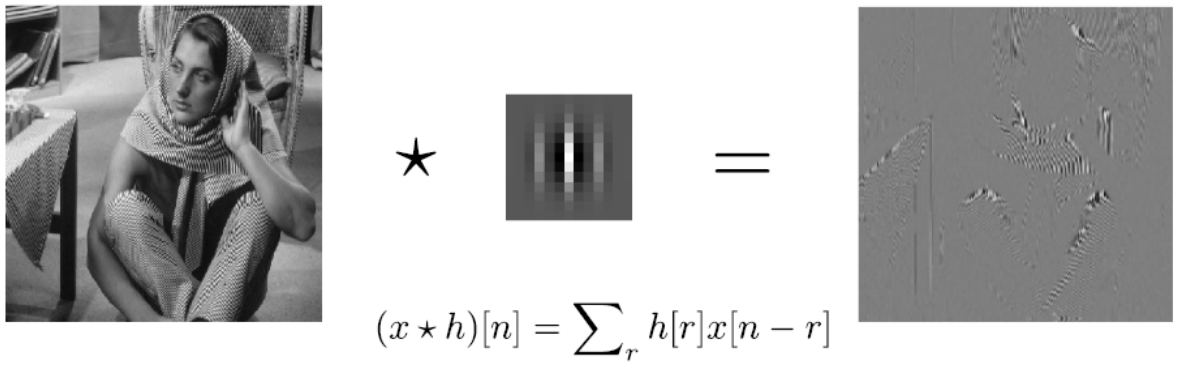
Outline

CNN: historic origin

- Convolutional Neural Networks (CNN) come from **image processing** (mostly).

CNN: historic origin

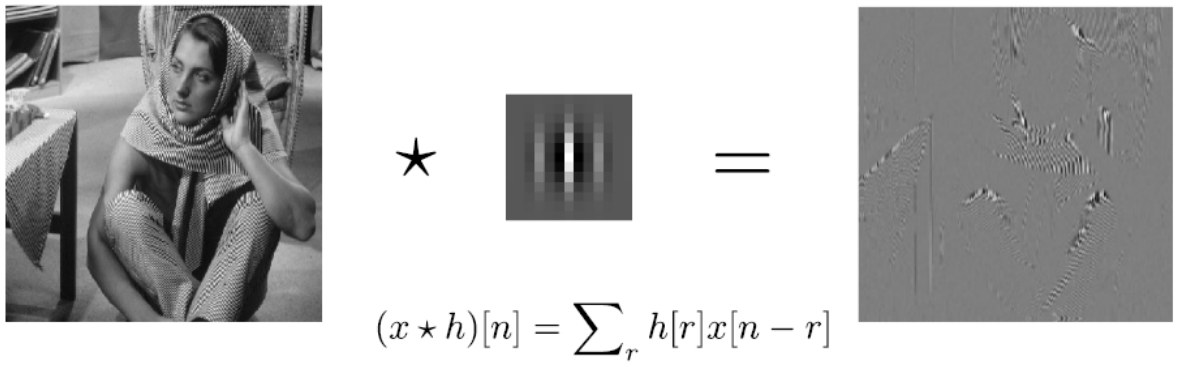
- Convolutional Neural Networks (CNN) come from **image processing** (mostly).
- Since the 80s - 90s, the heart of image processing are convolutions, aka **pattern matching**.



$$(x \star h)[n] = \sum_r h[r]x[n-r]$$

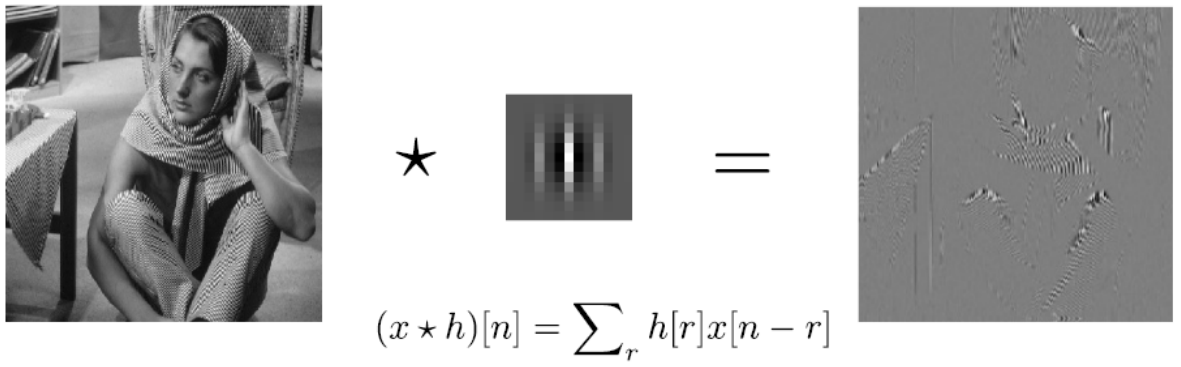
CNN: historic origin

- Convolutional Neural Networks (CNN) come from **image processing** (mostly).
- Since the 80s - 90s, the heart of image processing are convolutions, aka **pattern matching**.
  - how much the image locally correlate with a small kernel (pattern, patch...)
  - Ex: **Fourier Transform, wavelets**...



CNN: historic origin

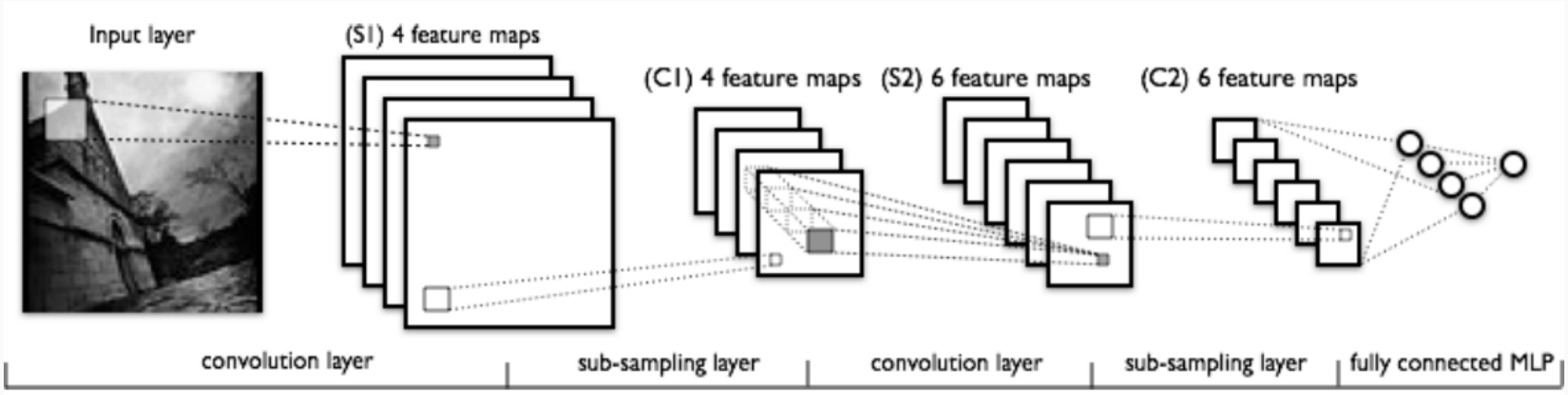
- Convolutional Neural Networks (CNN) come from **image processing** (mostly).
- Since the 80s - 90s, the heart of image processing are convolutions, aka **pattern matching**.
  - how much the image locally correlate with a small kernel (pattern, patch...)
  - Ex: **Fourier Transform, wavelets**...



- Impossible to try every pattern!
- Patterns have a **hierarchical structure**: small patterns are "combined" to make bigger objects, etc.

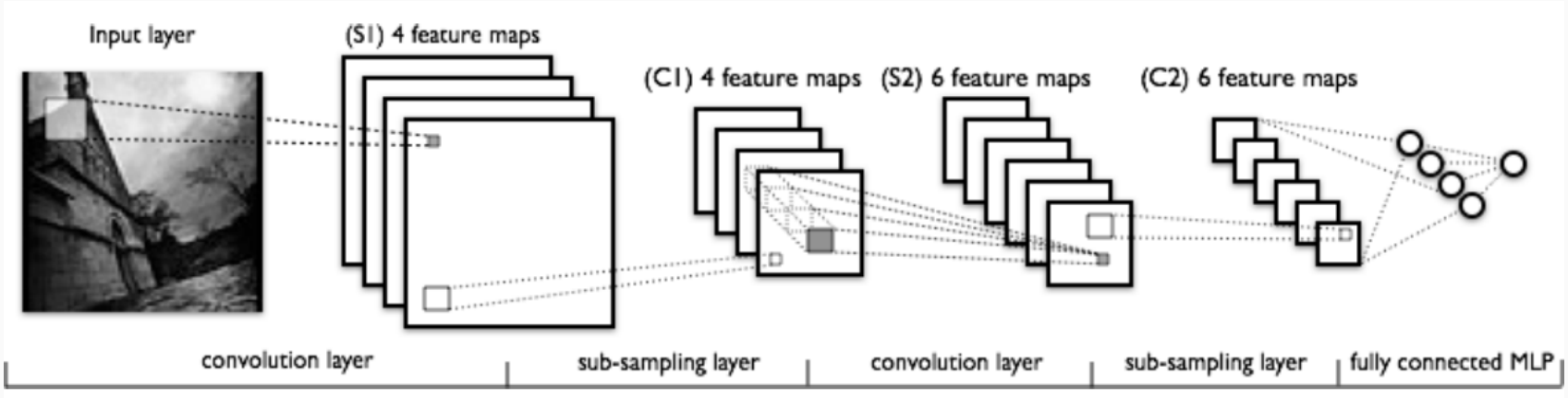
Ingredients of a CNN

- Convolution filters: the convolution kernels are **learned** (unlike wavelets)



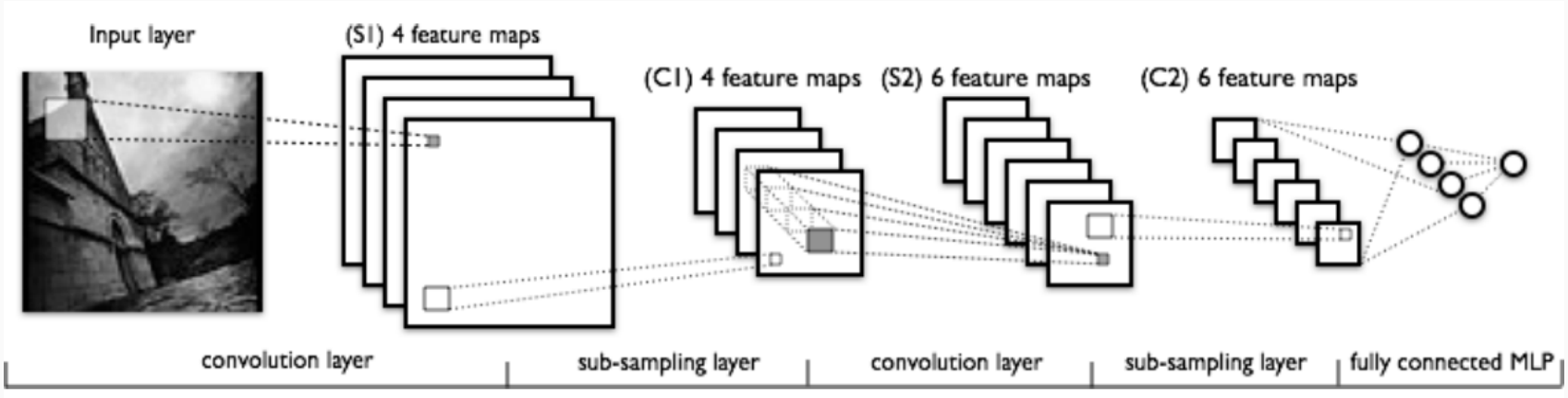
Ingredients of a CNN

- Convolution filters: the convolution kernels are **learned** (unlike wavelets)
- Non-linearity: without it, stacking convolution would still be linear!



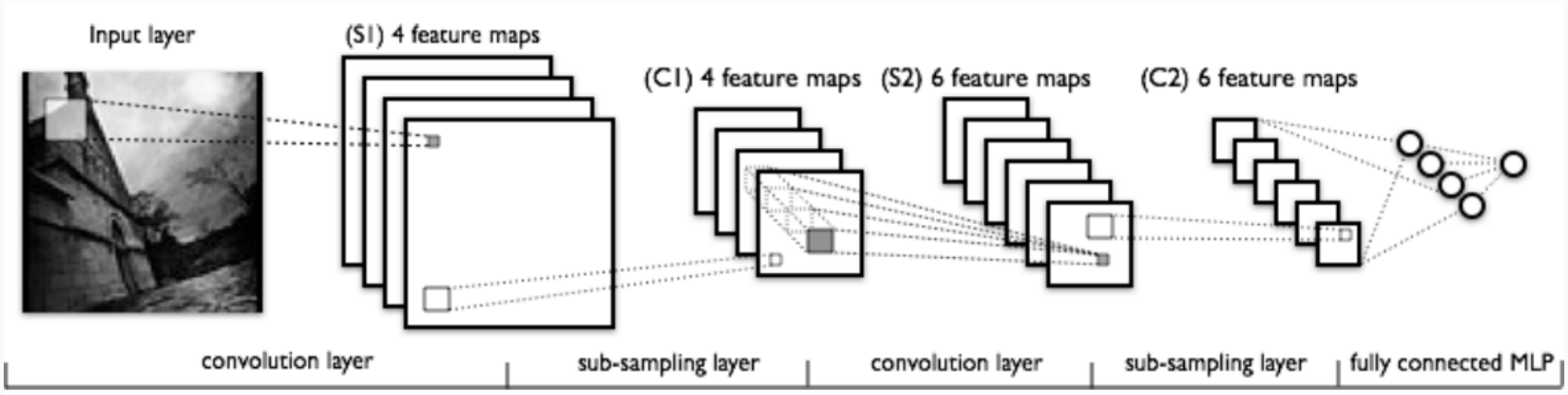
Ingredients of a CNN

- Convolution filters: the convolution kernels are **learned** (unlike wavelets)
- Non-linearity: without it, stacking convolution would still be linear!
- Pooling (downsampling, subsampling...) = **local aggregation**: reduce dimension, **make the model hierarchical**



Ingredients of a CNN

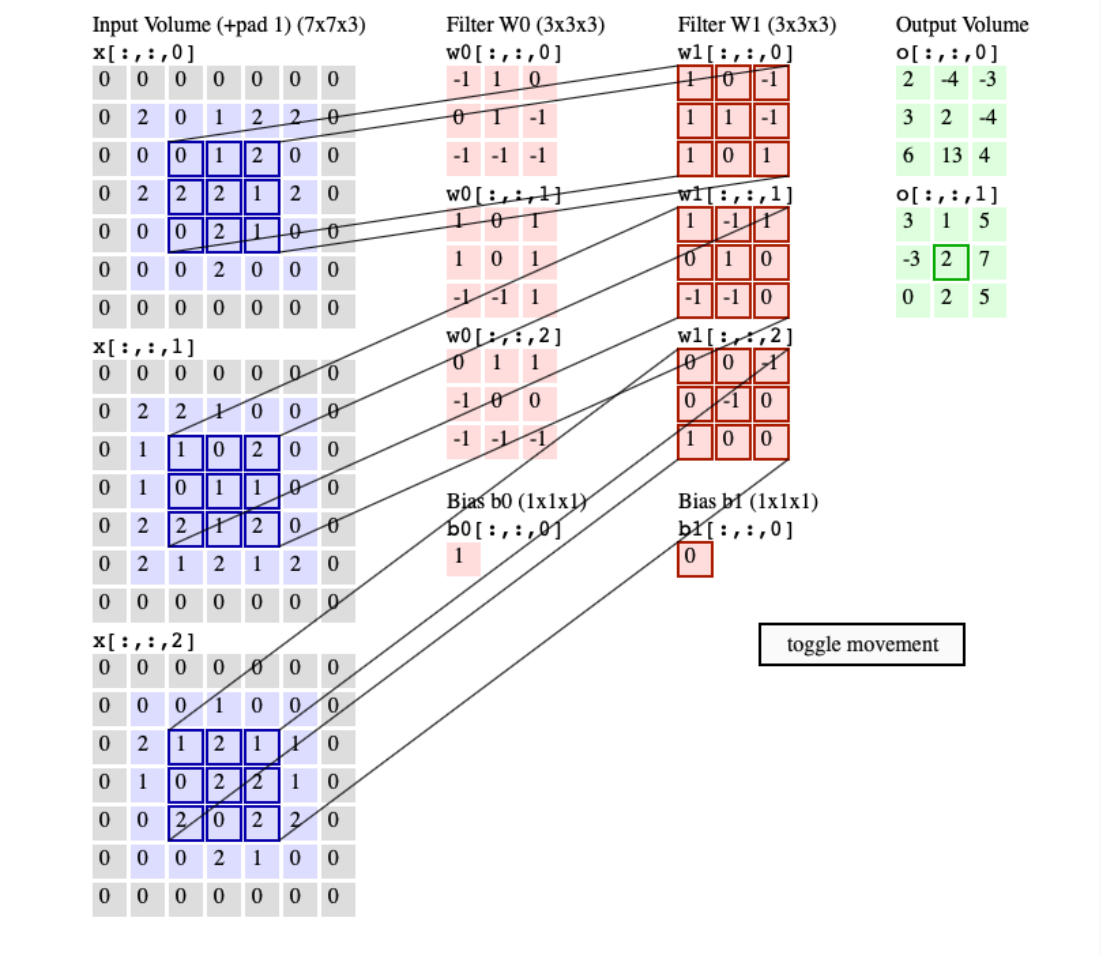
- Convolution filters: the convolution kernels are **learned** (unlike wavelets)
- Non-linearity: without it, stacking convolution would still be linear!
- Pooling (downsampling, subsampling...) = **local aggregation**: reduce dimension, **make the model hierarchical**



- Similar to an MLP, but with significant **zeroed weights** and weight-sharing (instead of dense parameter matrices, use block matrices with same, structured blocks)

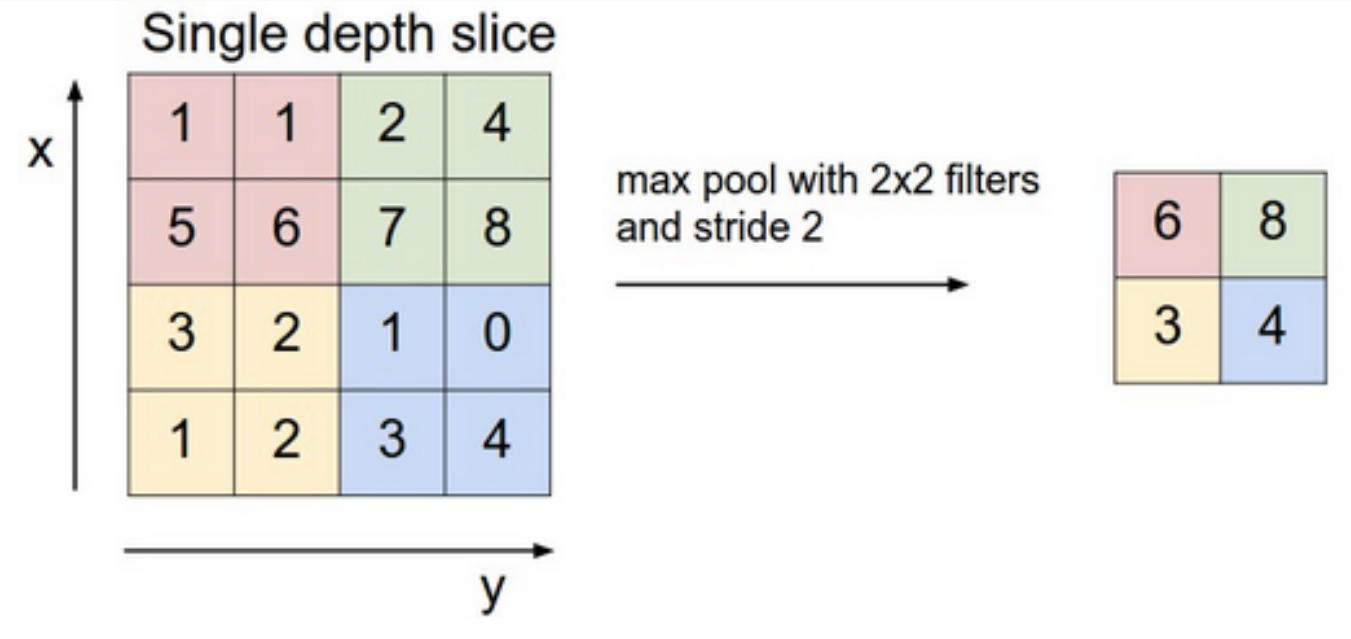
CNN Convolutive Layer example

RGB Channels Filter 1 Filter 2 Output Stack



Pooling / Downsampling within CNNs

Example: Max Pooling



- Only the locations on the image that shows the strongest correlation to each feature (the max value) are preserved, and those max values combine to form a lower-dimensional space
- Other examples: mean pooling, softmax...

CNNs: remark

- Depth is important to make the model **hierarchical**: it is impossible to enumerate all objects, but **small patterns are combined to learn bigger objects**

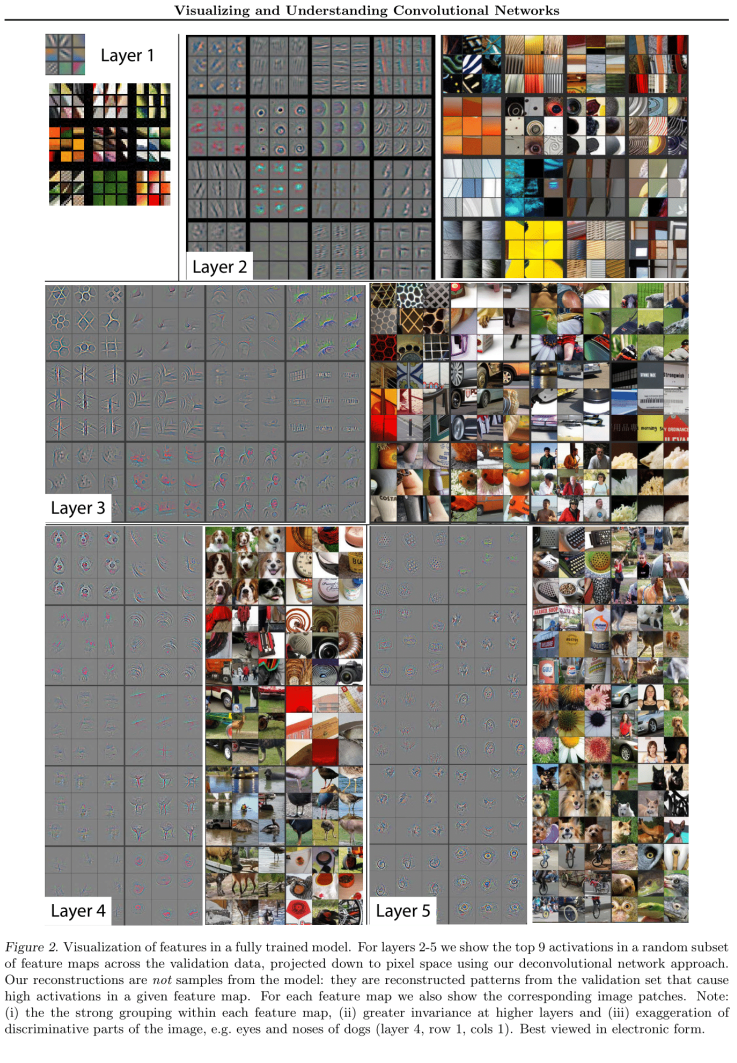
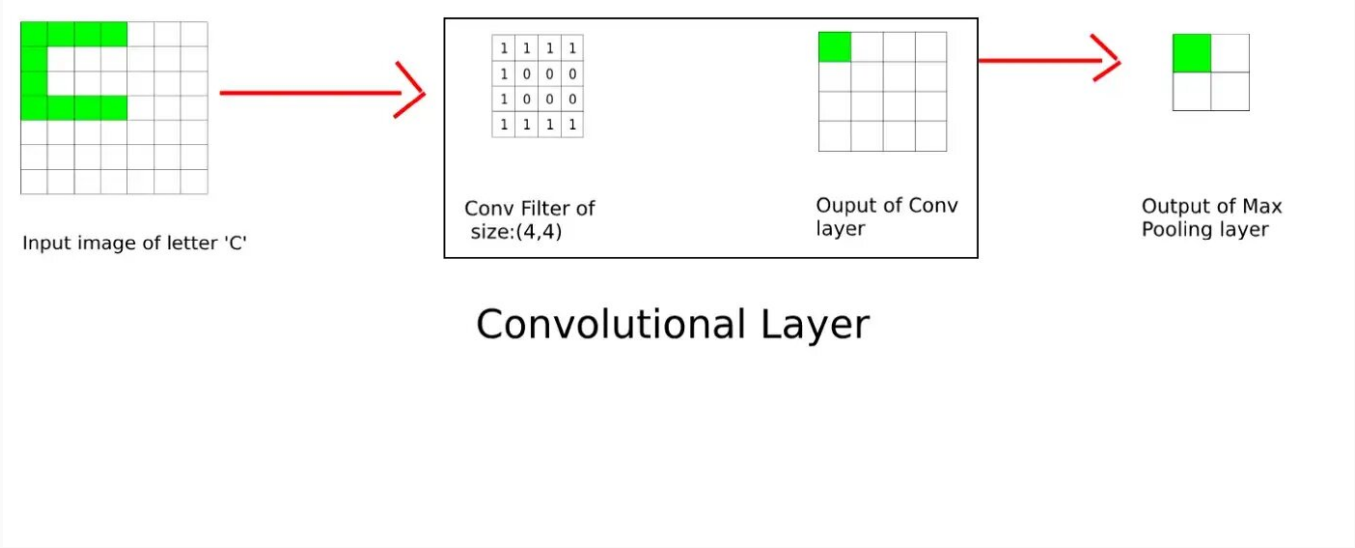
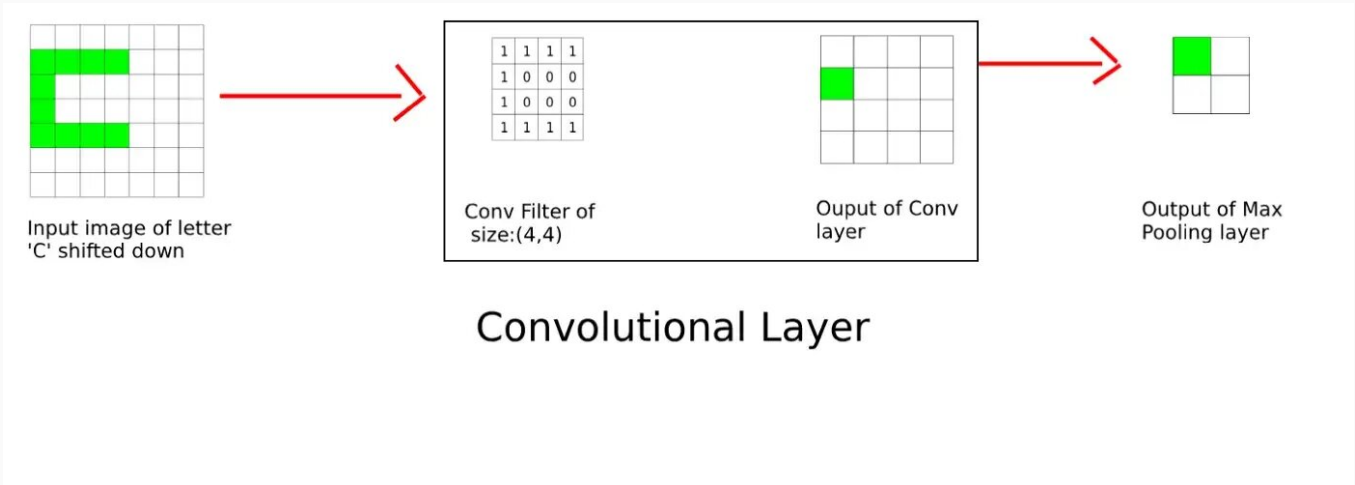


Figure 2. Visualization of features in a fully trained model. For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using our deconvolutional network approach. Our reconstructions are *not* samples from the model: they are reconstructed patterns from the validation set that cause high activations in a given feature map. For each feature map we also show the corresponding image patches. Note: (i) the the strong grouping within each feature map, (ii) greater invariance at higher layers and (iii) exaggeration of discriminative parts of the image, e.g. eyes and noses of dogs (layer 4, row 1, cols 1). Best viewed in electronic form.

CNNs: remark

- Due to convolution+pooling, CNNs are (approximately) **translation-invariant**: moving the object around does not change the output $h(\tau(x)) = h(x)$.





Outline

AutoEncoder

- Autoencoders are Unsupervised Neural Networks, designed for **representation learning** and/or dimension reduction

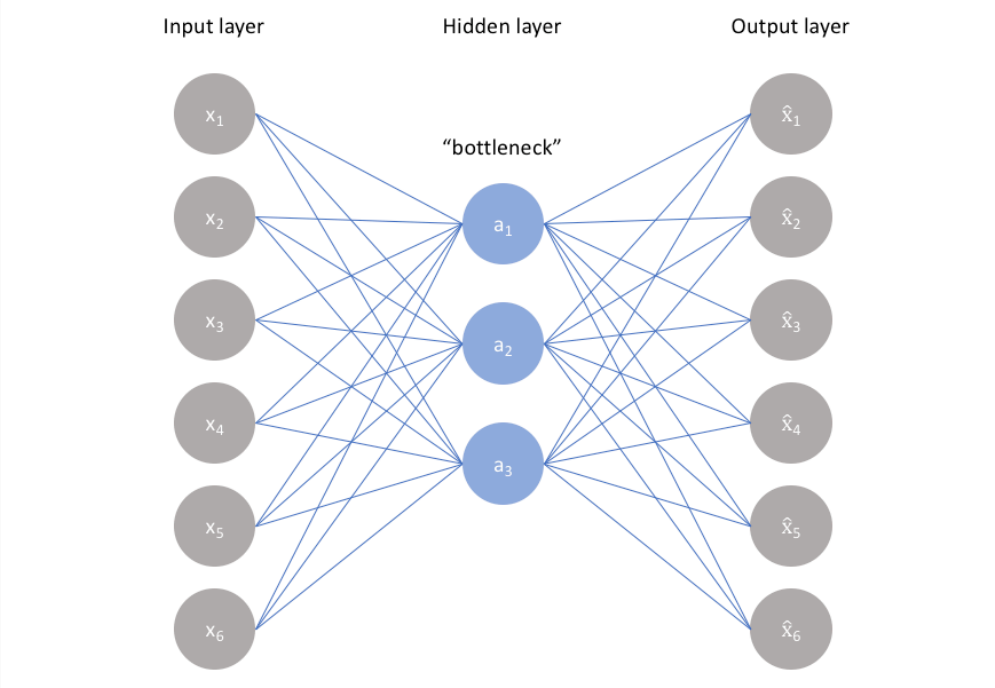
AutoEncoder

- Autoencoders are Unsupervised Neural Networks, designed for **representation learning** and/or dimension reduction
- Two parts: an **encoder** $f_\theta : x \to z$, and a **decoder** $g_\pi : z \to \tilde{x} \approx x$

AutoEncoder

- Autoencoders are Unsupervised Neural Networks, designed for **representation learning** and/or dimension reduction
- Two parts: an **encoder** $f_\theta : x \to z$, and a **decoder** $g_\pi : z \to \tilde{x} \approx x$
- main idea : impose a **bottleneck** in the network to **compressed** knowledge representation of the input.

Rk : This assumes that the data are structured (like images!), otherwise such compression will be very difficult if not impossible without loosing much information.

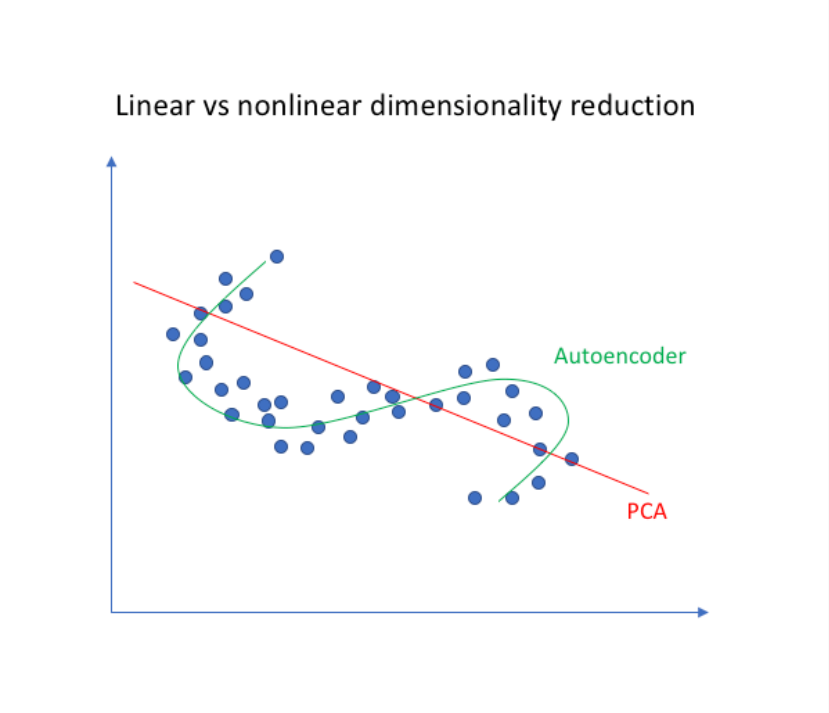


AutoEncoder principle

$\Rightarrow$ Formulate the problem as a **supervised learning** problem whose output is $\{x_\ell\}$

$\Rightarrow$ The empirical risk to minimize is thus $L(f(x), x)$: **the bottleneck plays a key role** (otherwise the network simply passes the values to the output.)

$\Rightarrow$ if linear activation function were used, that would perform **PCA like** dimension reduction

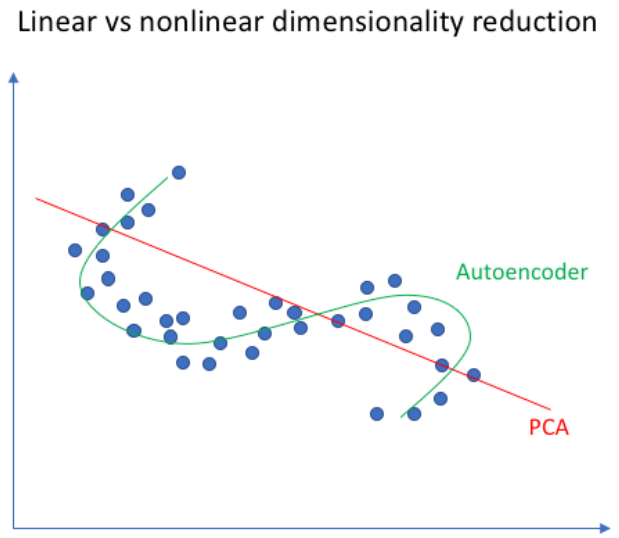


AutoEncoder principle

$\Rightarrow$ Formulate the problem as a **supervised learning** problem whose output is $\{x_\ell\}$

$\Rightarrow$ The empirical risk to minimize is thus $L(f(x), x)$: **the bottleneck plays a key role** (otherwise the network simply passes the values to the output.)

$\Rightarrow$ if linear activation function were used, that would perform **PCA like** di-

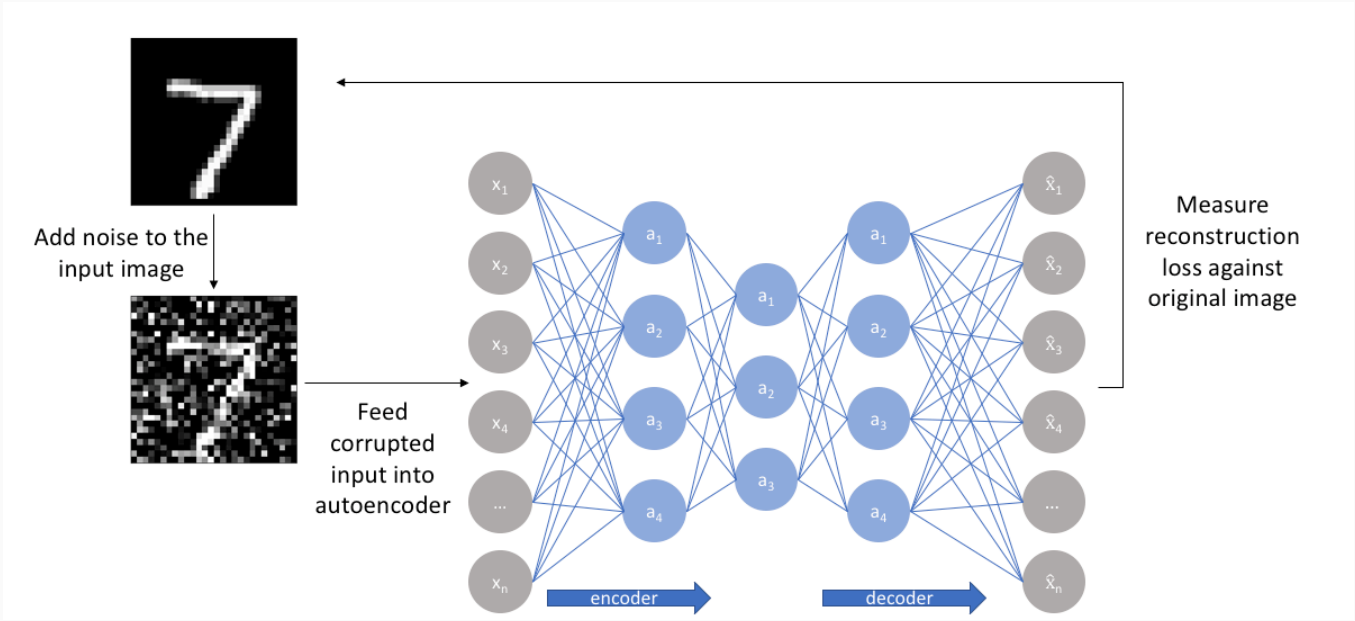


The AutoEncoder must be :

- sensitive enough to the inputs, to built an accurate reconstruction
- insensitive enough to the inputs to avoid overfitting

This requires to regularize the loss function of the form
$L(f(x), x) + \textit{regularization}$

Example of application: AE for denoising



The low-dimensional representation has a **regularizing effect** (like wavelet thresholding).

Sparse AE

How many nodes?

- **If too many nodes**, the AE may be capable of learning a way to **simply memorize the data**.

Sparse AE

How many nodes?

- **If too many nodes**, the AE may be capable of learning a way to **simply memorize the data**.
- **Taking inspiration from ℓ_1 sparse regularization (like LASSO)**: keep the number of nodes in hidden layers **quite large**, but regularize the loss function by **penalizing activations within a layer**. For layer k :

$$L(f(x), x) + \lambda \sum_{j=1}^{r_k} |o_j^k|$$