Mathematical introduction to Compressed Sensing

Lesson 1 : measurements and sparsity

Guillaume Lecué

ENSAE

Organization of the course

Every Tuesday (2/02 – 9/02 – 23/02 – 30/02 – 2/03 – 9/03 – 16/03 – 23/03)

3 hours course (15:15 to 18:30) 15 minutes break.

No lesson (16/02)

Simulations using python + notebook + cvxopt + cvxpy (to **install before**)

All the course material (python and course notebooks) are available on my webpage

Evaluation:

- Python notebook or pdf report (see the details on my webpage)
- register by group of two students before 22/02 on my webpage in the comments section of my webpage
- Project defense between 29/03 and 2/04.

Aim of the course: analyze high-dimensional data

1. Understand low-dimensional structures in high-dimensional spaces
2. reveal this structure via appropriate measurements
3. construct efficient algorithms to learn this structure

Tools:

1. approximation theory
2. probability theory
3. convex optimization algorithms

First lesson is about:

Two central ideas:

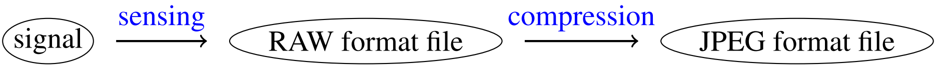
1. Sparsity
2. measurements

through three examples:

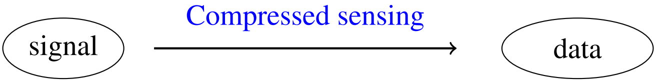
1. Single pixel camera
2. face recognition
3. Financial data

What is Compressed Sensing?

Classical data acquisition system in two steps:



Compressed sensing makes it in one step:



In french: Compressed Sensing = "acquisition comprimée"

Q: How is it possible? A: Construct clever measurements!

x: a signal (finite dimensional vector, say $x \in \mathbb{R}^N$)

Take m linear measurements of signal x:

$$y_i = \left\langle x, X_i \right\rangle, \quad i = 1, \ldots, m$$

where:

1. X_i : i-th measurement vector (in $\mathbb{R}^N$),
2. y_i : i-th measurement (= data = observation).

Problem: reconstruct x from the measurements $(y_i)_{i=1}^m$ and the measurement vectors $(X_i)_{i=1}^m$ with m as small as possible.

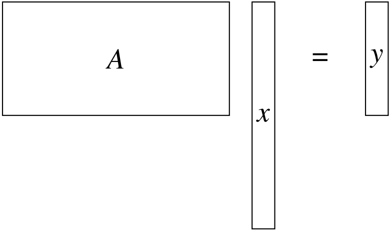
Matrix version of Compressed Sensing

We denote

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m \quad \text{and} \quad A = \begin{pmatrix} X_1^\top \\ \vdots \\ X_m^\top \end{pmatrix} \in \mathbb{R}^{m \times N}$$

y: measurements vector and A: measurements matrix

Problem: find x such that $y = Ax$ when $m << N$



CS = solve a highly undetermined linear system

Sparsity = low-dimensional structure

Since $m < N$ there is no unique solution to the problem $y = Ax \Rightarrow$ no hope to reconstruct x from the m measurements $y_i = \langle x, X_i \rangle$.

Idea: Signals to recover have some low-dimensional structure. We assume that x is sparse.

Definition

Support of $x = (x_1, \dots, x_N)^\top \in \mathbb{R}^N$:

$$\mathrm{supp}(x) = \left\{ j \in \{1, \dots, N\} : x_j \neq 0 \right\}$$

Size of the support of x:

$$\|x\|_0 = |\mathrm{supp}(x)|$$

x is **s-sparse** when $\|x\|_0 \leq s$ and $\Sigma_s = \{x \in \mathbb{R}^N : \|x\|_0 \leq s\}$.

Sparsity and the undetermined system $y = Ax$

Idea: Maybe the kernel of A is in such a position that the sparsest solution to $y = Ax$ is x itself?

Procedure: Look for the sparsest solution of the system $y = Ax$:

$$\boxed{\hat{x}_0 \in \underset{At=y}{\operatorname{argmin}} \|t\|_0} \tag{1}$$

which looks for vector(s) t with the shortest support in the affine set of solutions

$$\{t \in \mathbb{R}^N : At = y\} = x + \ker(A).$$

Idea: Denote $\Sigma_s = \{t \in \mathbb{R}^N : \|t\|_0 \leq s\}$. If $\Sigma_s \cap \big(x + \ker(A)\big) = \{x\}$ for $s = \|x\|_0$ then the sparsest element in $x + \ker(A)$ is x and so $\hat{x}_0 = x$

Definition

$\hat{x}_0$ is called the ℓ_0-minimization procedure

(cf. Second lesson)

Compressed sensing: problems statement

Problem 1: *Construct a minimal number of measurement vectors* $X_1, \dots, X_m$ *such that one can reconstruct any s-sparse signal x from the m measurements* $(\langle x, X_i \rangle)_{i=1}^m$.

Problem 2: *Construct efficient algorithms that can reconstruct exactly any sparse signal x from the measurements* $(\langle x, X_i \rangle)_{i=1}^m$.

Is signal x really sparse?

Sparsity of signal x is the main assumption in Compressed Sensing (and more generally in high-dimensional statistics).

Q.: Is it true that "real signals" are sparse?

Three examples:

1. images
2. face recognition
3. financial data

Compressed Sensing in images

Sparse representation of images

An image is a:

1. vector $f \in \mathbb{R}^{n\times n}$
2. function $f : \{0, \dots, n-1\}^2 \to \mathbb{R}$

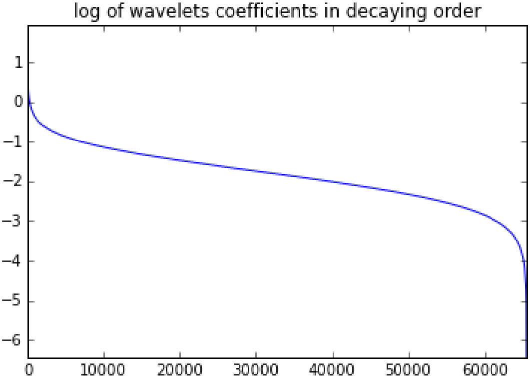
Images can be developed into basis: $f = \sum_{j=1}^{n^2} \langle f, \psi_j \rangle \psi_j$

Problem in approximation theory: Find basis (ψ_j) such that $(\langle f, \psi_j \rangle)_{j=1}^{n^2}$ is (approximatively) a sparse vector for real life images f.

Solution: Wavelets basis (cf. Gabriel Peyré course)

notebook: wavelet decomposition

Sparse representation of images



Graphics: Representation of $\left(\log|\langle f, \psi_j\rangle|\right)_{j=1}^{n^2}$ in a decreasing order for $n = 256$ ($256^2 = 65.536$ coefficients).

Conclusion: When developed in an appropriate basis, images have an *almost* sparse representation.

Sparse representation of images

Idea: Compression of images by thresholding small wavelets coefficients (JPEG 2000).

Remark: these are the only three slides about approximation theory in this course!

Compressed sensing and images

Two differences with the CS framework introduced above:

1. images are almost sparse
2. images are (almost) sparse not in the canonical basis but in some other (wavelet) basis.

Two consequences:

1. our procedures will be asked to "adapt" to this almost sparse situation: stability property
2. we need to introduce a stuctured sparsity: being sparse in some general basis.

Structured sparsity

Definition

Let $\mathcal{F} = \{f_1, \cdots, f_p\}$ be a dictionary in $\mathbb{R}^N$. A vector $x \in \mathbb{R}^N$ is said ***s*-sparse in** $\mathcal{F}$ when there exists $J \subset \{1, \ldots, p\}$ such that

$$|J| \leq s \text{ and } x = \sum_{j \in J} \theta_j f_j.$$

In that case,

$$x = F\theta \text{ where } F = [f_1 | \cdots, | f_p] \in \mathbb{R}^{N \times p}$$

and $\theta \in \mathbb{R}^p$ is a s-sparse in the canonical basis of $\mathbb{R}^p$.
For CS measurements, one has:

$$y = Ax = AF\theta$$

where $\theta \in \Sigma_s$ and so one just has to replace the measurement matrix A by AF.
Conclusion: All the course deals only with vectors that are sparse in the canonical basis.

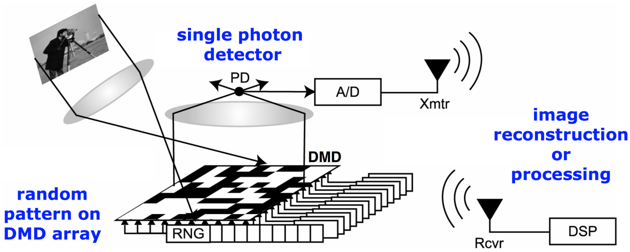
What is a photos machine using CS?

It should take measurements like:

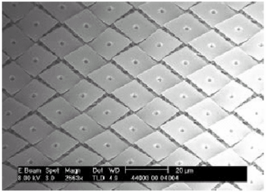
We take m measurements:

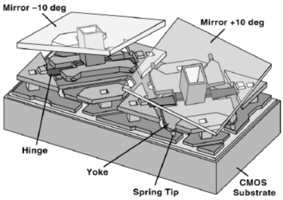
In particular, measurements $y_1, \ldots, y_m$ are real numbers. Each of them can be stored using only one pixel in the camera.

Single pixel camera from RICE University



DMD: digital micromirror device – randomly orientated





Single pixel camera from RICE University

Example of reconstruction of an image using the single pixel camera.

Two problems:

1. How do we choose the measurement vectors:  $, \cdots ,$  ?
2. Is there an efficient algorithm to reconstruct the signal from those few measurements?

CS in face Recognition

face recognition and Compressed Sensing

Database: $\mathcal{D} := \{(\phi_j, \ell_j) : 1 \leq j \leq N\}$ where :

1. $\phi_j \in \mathbb{R}^m$ is a vector representation of the j-th image, (for instance, concatenation of the images pixels value)
2. $\ell_j \in \{1, \ldots, C\}$ is a label refering to a person

A same person may be represented in $\mathcal{D}$ several times from various angles, luminosity, etc..

Problem: Given a new image $y \in \mathbb{R}^m$, we want to label it with an element from the set $\{\ell_j, j = 1, \ldots, C\}$

"Classical" solution: use multi-class classification algorithm.

Here: Face recognition as a CS problem.

The sparsity assumption in face recognition

Empirical observation: If for all of the C individuals one has:

1. a large enough number of images,
2. enough diversity in terms of angles and brightness

then for any new image $y \in \mathbb{R}^m$ of individual number $i \in \{1, \ldots, C\}$, one expect that

$$y \approx \sum_{j:\ell_j = i} \phi_j x_j.$$

Consequence: We assume that a new image $y \in \mathbb{R}^m$ can be written as

$$\boxed{y = \Phi x + \zeta}$$

where:

1. $\Phi = [\Phi_1 | \Phi_2 | \cdots | \Phi_C]$ and $\Phi_i = [\phi_j : \ell_j = i]$ for any $i \in \{1, \ldots, C\}$,
2. $x = [\mathbf{0}^\top | \mathbf{0}^\top | \cdots | \mathbf{x}_i^\top | \mathbf{0}^\top | \cdots | \mathbf{0}^\top]^\top$ where $\mathbf{x}_i$ is the restriction of x to the columns of Φ_i in Φ
3. $\zeta \in \mathbb{R}^m$ error due to linear approximation of y by columns in Φ.

Face recognition as a noisy CS problem

Compare with the benchmark CS setup, one has three difference:

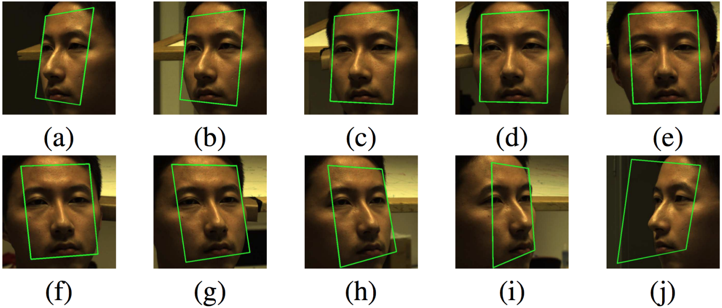
1. there is an additional noise term ζ
2. the sparsity assumption on x is stronger here: x is block-sparse
3. depending on the control one has on the database, we may or may not have the ability to choose (in a restricted way) the measurement matrix.

Three consequences:

1. our procedures will be asked to deal with noisy data: robustness property
2. we will design procedures taking advantages of more "advanced" sparsity like the block-sparsity one
3. when one is in a situation where there is no control on the choice of measurement vectors then one can try several algorithms and see how they behave.

Construction of a measurement matrix in face recognition

Various angles:

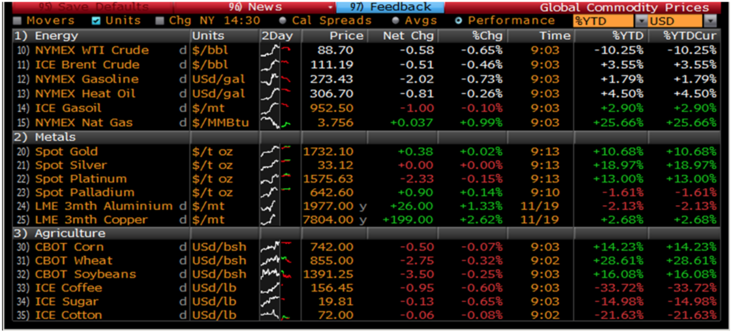


Various brightness:

CS in Finance

Finance and CS

Problem: We observe the performances of a portfolio every minute: $y_1, \ldots, y_m$. We would like to know how it is structured (shares and quantity).
Data: In addition to $y_1, \ldots, y_m$, we know the values of all shares at any time:


95) Save Defaults | 96) News | 97) Feedback | Global Commodity Prices
Movers | Units | Chg NY 14:30 | Cal Spreads | Avgs | Performance | %YTD | USD

		Units	2Day	Price	Net Chg	%Chg	Time	%YTD	%YTDCur
1) Energy									
10) NYMEX WTI Crude	d	$/bbl		88.70	-0.58	-0.65%	9:03	-10.25%	-10.25%
11) ICE Brent Crude	d	$/bbl		111.19	-0.51	-0.46%	9:03	+3.55%	+3.55%
12) NYMEX Gasoline	d	USd/gal		273.43	-2.02	-0.73%	9:03	+1.79%	+1.79%
13) NYMEX Heat Oil	d	USd/gal		306.70	-0.81	-0.26%	9:03	+4.50%	+4.50%
14) ICE Gasoil	d	$/mt		952.50	-1.00	-0.10%	9:03	+2.90%	+2.90%
15) NYMEX Nat Gas	d	$/MMBtu		3.756	+0.037	+0.99%	9:03	+25.66%	+25.66%
2) Metals									
20) Spot Gold		$/t oz		1732.10	+0.38	+0.02%	9:13	+10.68%	+10.68%
21) Spot Silver		$/t oz		33.12	+0.00	+0.00%	9:13	+18.97%	+18.97%
22) Spot Platinum		$/t oz		1575.63	-2.33	-0.15%	9:13	+13.00%	+13.00%
23) Spot Palladium		$/t oz		642.60	+0.90	+0.14%	9:10	-1.61%	-1.61%
24) LME 3mth Aluminium	d	$/mt		1977.00 y	+26.00	+1.33%	11/19	-2.13%	-2.13%
25) LME 3mth Copper	d	$/mt		7804.00 y	+199.00	+2.62%	11/19	+2.68%	+2.68%
3) Agriculture									
30) CBOT Corn	d	USd/bsh		742.00	-0.50	-0.07%	9:03	+14.23%	+14.23%
31) CBOT Wheat	d	USd/bsh		855.00	-2.75	-0.32%	9:02	+28.61%	+28.61%
32) CBOT Soybeans	d	USd/bsh		1391.25	-3.50	-0.25%	9:03	+16.08%	+16.08%
33) ICE Coffee	d	USd/lb		156.45	-0.95	-0.60%	9:03	-33.72%	-33.72%
34) ICE Sugar	d	USd/lb		19.81	-0.13	-0.65%	9:03	-14.98%	-14.98%
35) ICE Cotton	d	USd/lb		72.00	-0.06	-0.08%	9:02	-21.63%	-21.63%

Finance and CS

$x_{i,j}$: value of share j at time i. We have the following data:

$$
\begin{array}{llll}
t = 1: & y_1 : \text{ portfolio value} & (x_{1,j})_{j=1}^N : & \text{shares values} \\
t = 2: & y_2 : \text{ portfolio value} & (x_{2,j})_{j=1}^N : & \text{shares values} \\
\cdots\cdots & & & \\
t = m: & y_m : \text{ portfolio value} & (x_{m,j})_{j=1}^N : & \text{shares values}
\end{array}
$$

Sparsity assumption: The portfolio contains only a limited number of shares and its structure did not change during the observation time.
Problem formulation: find $x \in \mathbb{R}^N$ such that $y = Ax$ where

$$y = (y_i)_{i=1}^m \text{ and } A = \big(x_{i,j} : 1 \leq i \leq m, 1 \leq j \leq N\big)$$

and x is supposed to be sparse.

CS and high-dimensional statistics

Definition

We say that a statistical problem is a **high-dimensional statistical problem** when one has to estimate a N-dimensional parameter / vector / object using m observations and $m < N$.

1. CS is therefore a high-dimensional statistical problem.
2. Noisy CS is exactly the linear regression statistical model when the noise is assumed to be random.