An Introduction to Sparse Recovery (and Compressed Sensing)

Massimo Fornasier

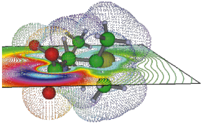
Department of Mathematics
massimo.fornasier@ma.tum.de
http://www-m15.ma.tum.de/

CoSeRa 2016
RWTH - Aachen
September 19, 2016

Motivation

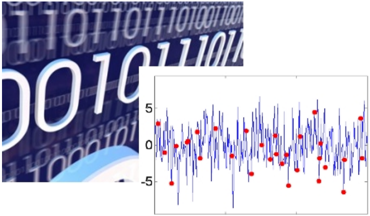
We collect a large amount of - *indirect measurements* - of any sort of relevant information:

Brain section acquired by Magnetic Resonance Imaging



Structure of a molecule recovered by X-ray crystallography

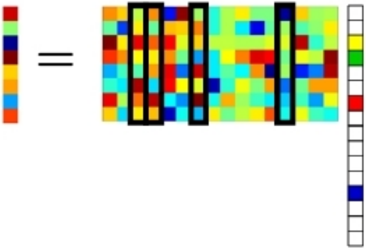
Aggregated statistical data (e.g. aggregated energy consumption)



Sampling streams of data

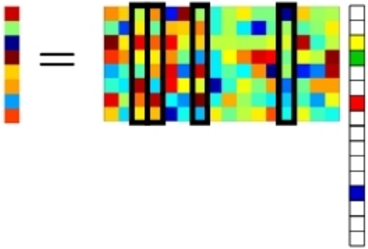
From large to small

The measurements of large amount of interesting data allow us to *distill* some aggregated information about the data ...



From large to small

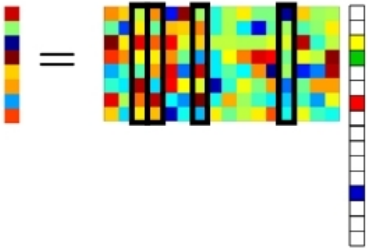
The measurements of large amount of interesting data allow us to *distill* some aggregated information about the data ...



- How much information are we able to capture?

From large to small

The measurements of large amount of interesting data allow us to *distill* some aggregated information about the data ...



- How much information are we able to capture?
- Is this information enough to allow us to get a sketch of the original data?

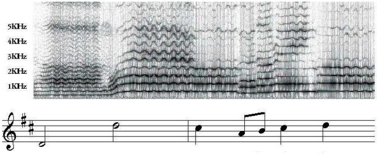
A few references

- *A Mathematical Introduction to Compressive Sensing* (Holger Rauhut and Simon Foucart), Birkhäuser-Springer, 2013.
- *Numerical methods for sparse recovery* book chapter in "Theoretical Foundations and Numerical Methods for Sparse Recovery", M. Fornasier (ed.) Radon Series in Applied and Computational Mathematics 9, de Gruyter, 2010
- *Compressive Sensing* (Massimo Fornasier and Holger Rauhut), book chapter in "Handbook of Mathematical Methods in Imaging" Springer.
- *An Overview on Algorithms for Sparse Recovery* (Massimo Fornasier and Steffen Peter) book chapter in "Sparse Reconstruction and Compressive Sensing in Remote Sensing", X. Zhu and R. Bamler (ed.), Springer, 2015.

Publications:
`https://www-m15.ma.tum.de/Allgemeines/PublicationsEN`
Software:
`https://www-m15.ma.tum.de/Allgemeines/SoftwareSite`

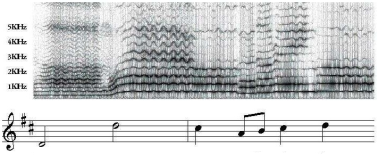
The world we wish to capture is sparse!

- A music is a stream of a finite number of notes: few frequencies (and their harmonics) are simultaneously active at each time;



The world we wish to capture is sparse!

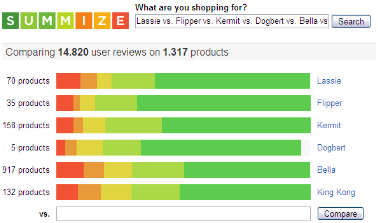
- A music is a stream of a finite number of notes: few frequencies (and their harmonics) are simultaneously active at each time;



- A natural image is made of piecewise smooth parts and a few edges;

The world we wish to capture is sparse!

- Consumer behaviors can often be subdivided into few categories;



What are you shopping for?

Lassie vs. Flipper vs. Kermit vs. Dogbert vs. Bella vs

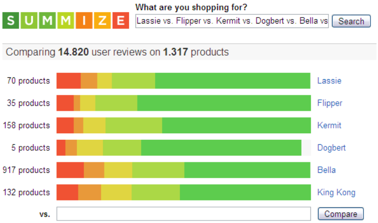
Comparing **14.820** user reviews on **1.317** products

70 products	Lassie
35 products	Flipper
158 products	Kermit
5 products	Dogbert
917 products	Bella
132 products	King Kong

vs.

The world we wish to capture is sparse!

- Consumer behaviors can often be subdivided into few categories;



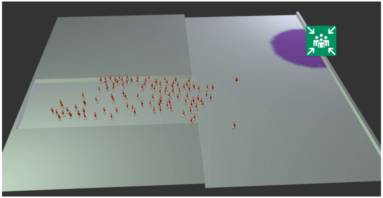
What are you shopping for?

Lassie vs. Flipper vs. Kermit vs. Dogbert vs. Bella vs

Comparing **14.820** user reviews on **1.317** products

70 products	Lassie
35 products	Flipper
158 products	Kermit
5 products	Dogbert
917 products	Bella
132 products	King Kong

- To optimally control the emergency evacuation of a crowd from a room, we just need only a few informed agents ...



Let us then assume that our data are sparse!

In many circumstances it is legitimate to assume that our data $\bar{x} \in \mathbb{R}^N$ are indeed "sparse", i.e., they can be described by using a few "words" of a given "dictionary" $\mathcal{D} = \{e_1, e_2, \dots e_N\} \subset \mathbb{R}^N$.

Let us then assume that our data are sparse!

In many circumstances it is legitimate to assume that our data $\bar{x} \in \mathbb{R}^N$ are indeed "sparse", i.e., they can be described by using a few "words" of a given "dictionary" $\mathcal{D} = \{e_1, e_2, \dots e_N\} \subset \mathbb{R}^N$. In mathematical terms we assume

$$\bar{x} = x_1 e_1 + x_2 e_2 + \dots x_N e_N,$$

for $x = (x_1, \dots, x_N)$ a *sparse* vector of $\mathbb{R}^N$. By sparse we mean that

$$\|x\|_{\ell_0^N} := \#\{x_i \neq 0\} \sim \log(N).$$

Let us then assume that our data are sparse!

In many circumstances it is legitimate to assume that our data $\bar{x} \in \mathbb{R}^N$ are indeed "sparse", i.e., they can be described by using a few "words" of a given "dictionary" $\mathcal{D} = \{e_1, e_2, \dots e_N\} \subset \mathbb{R}^N$. In mathematical terms we assume

$$\bar{x} = x_1 e_1 + x_2 e_2 + \dots x_N e_N,$$

for $x = (x_1, \dots, x_N)$ a *sparse* vector of $\mathbb{R}^N$. By sparse we mean that

$$\|x\|_{\ell_0^N} := \#\{x_i \neq 0\} \sim \log(N).$$

For some classes $\mathcal{C}$ of data, the dictionary is known (for instance for music, local Fourier bases are a good dictionary, while for images wavelets do the job).

Let us then assume that our data are sparse!

In many circumstances it is legitimate to assume that our data $\bar{x} \in \mathbb{R}^N$ are indeed "sparse", i.e., they can be described by using a few "words" of a given "dictionary" $\mathcal{D} = \{e_1, e_2, \dots e_N\} \subset \mathbb{R}^N$. In mathematical terms we assume

$$\bar{x} = x_1 e_1 + x_2 e_2 + \dots x_N e_N,$$

for $x = (x_1, \dots, x_N)$ a *sparse* vector of $\mathbb{R}^N$. By sparse we mean that

$$\|x\|_{\ell_0^N} := \#\{x_i \neq 0\} \sim \log(N).$$

For some classes $\mathcal{C}$ of data, the dictionary is known (for instance for music, local Fourier bases are a good dictionary, while for images wavelets do the job).

The problem of identifying a sparsifying dictionary $\mathcal{D}_{\mathcal{C}}$ given a class of data $\mathcal{C} = \{\bar{x}\}$ is called *dictionary learning*, but we will not address it here.

Recovering a sketch of our data from distilled information

Given a sparse vector $x \in \mathbb{R}^N$ representing our data, we assume to operate linear measurements $A \in \mathbb{R}^{m \times N}$on it, distilling aggregated information $y \in \mathbb{R}^m$ on x, for $m \sim \|x\|_{\ell_0^N}$ i.e.,

$$y = Ax.$$

Recovering a sketch of our data from distilled information

Given a sparse vector $x \in \mathbb{R}^N$ representing our data, we assume to operate linear measurements $A \in \mathbb{R}^{m \times N}$on it, distilling aggregated information $y \in \mathbb{R}^m$ on x, for $m \sim \|x\|_{\ell_0^N}$ i.e.,

$$y = Ax.$$

Can we recover x, at least approximatively?

Recovering a sketch of our data from distilled information

Given a sparse vector $x \in \mathbb{R}^N$ representing our data, we assume to operate linear measurements $A \in \mathbb{R}^{m \times N}$on it, distilling aggregated information $y \in \mathbb{R}^m$ on x, for $m \sim \|x\|_{\ell_0^N}$ i.e.,

$$y = Ax.$$

Can we recover x, at least approximatively? Since the matrix A is not squared (hence not invertible), this question is not trivial and we need to define a proper solution concept.

Recovering a sketch of our data from distilled information

Given a sparse vector $x \in \mathbb{R}^N$ representing our data, we assume to operate linear measurements $A \in \mathbb{R}^{m \times N}$on it, distilling aggregated information $y \in \mathbb{R}^m$ on x, for $m \sim \|x\|_{\ell_0^N}$ i.e.,

$$y = Ax.$$

Can we recover x, at least approximatively? Since the matrix A is not squared (hence not invertible), this question is not trivial and we need to define a proper solution concept.

Occam's razor: the simplest explanation is often the good one!

Introduction to sparse recovery

We address the numerical solution of problems of the type:

$$(\ell_0) \qquad \min \|x\|_{\ell_0^N} \text{ subject to } \|Ax - y\|_{\ell_2^m} \leq \varepsilon;$$

Introduction to sparse recovery

We address the numerical solution of problems of the type:

$$(\ell_0) \qquad \min \|x\|_{\ell_0^N} \text{ subject to } \|Ax - y\|_{\ell_2^m} \leq \varepsilon;$$

$$(\ell_1) \qquad \min \|x\|_{\ell_1^N} \text{ subject to } \|Ax - y\|_{\ell_2^m} \leq \varepsilon;$$

Introduction to sparse recovery

We address the numerical solution of problems of the type:

$$(\ell_0) \qquad \min \|x\|_{\ell_0^N} \text{ subject to } \|Ax - y\|_{\ell_2^m} \leq \varepsilon;$$

$$(\ell_1) \qquad \min \|x\|_{\ell_1^N} \text{ subject to } \|Ax - y\|_{\ell_2^m} \leq \varepsilon;$$

where

$$x \in \mathbb{R}^N, \quad A \in \mathbb{R}^{m \times N}, \quad m \leq N$$

$$\|x\|_{\ell_0^N} := \#\operatorname{supp}(x), \quad \|x\|_{\ell_p^N} = \begin{cases} \left(\sum_{i=1}^N |x_j|^p\right)^{1/p}, & 0 < p < \infty, \\ \max_{j=1,\dots,N} |x_j|, & p = \infty. \end{cases}$$

Introduction to sparse recovery

We address the numerical solution of problems of the type:

$$(\ell_0) \qquad \min \|x\|_{\ell_0^N} \text{ subject to } \|Ax - y\|_{\ell_2^m} \leq \varepsilon;$$

$$(\ell_1) \qquad \min \|x\|_{\ell_1^N} \text{ subject to } \|Ax - y\|_{\ell_2^m} \leq \varepsilon;$$

where

$$x \in \mathbb{R}^N, \quad A \in \mathbb{R}^{m \times N}, \quad m \leq N$$

$$\|x\|_{\ell_0^N} := \# \operatorname{supp}(x), \quad \|x\|_{\ell_p^N} = \begin{cases} \left(\sum_{i=1}^N |x_j|^p\right)^{1/p}, & 0 < p < \infty, \\ \max_{j=1,\ldots,N} |x_j|, & p = \infty. \end{cases}$$

The matrix A is called the *measurement matrix*, which is distilling from x an aggregated (compressed) information y.

Adaptive VS Nonadaptive

These optimizations have been greatly popularized by the development of the field of nonadaptive compressed acquisition of data, the so-called *compressed sensing* (Donoho, Candés-Romberg-Tao '06).

Adaptive VS Nonadaptive

These optimizations have been greatly popularized by the development of the field of nonadaptive compressed acquisition of data, the so-called *compressed sensing* (Donoho, Candés-Romberg-Tao '06).

($\odot\odot$) Adaptive compressed acquisition

Adaptive VS Nonadaptive

These optimizations have been greatly popularized by the development of the field of nonadaptive compressed acquisition of data, the so-called *compressed sensing* (Donoho, Candés-Romberg-Tao '06).

(⊙⊙) Adaptive compressed acquisition

(◎◎) Nonadaptive compressed acquisition

Adaptive compressed acquisition

Let $k \in \mathbb{N}$, $k \leq N$ and

$$\Sigma_k := \{x \in \mathbb{R}^N : \# \operatorname{supp}(x) \leq k\},$$

is the set of vectors with at most k nonzero entries, which we will call *k-sparse vectors*.

Adaptive compressed acquisition

Let $k \in \mathbb{N}$, $k \leq N$ and

$$\Sigma_k := \{x \in \mathbb{R}^N : \#\operatorname{supp}(x) \leq k\},$$

is the set of vectors with at most k nonzero entries, which we will call *k-sparse vectors*.The *best k-term approximation error* that we can achieve in this set to a vector $x \in \mathbb{R}^N$ with respect to a suitable space quasi-norm $\|\cdot\|_X$ is defined by

$$\sigma_k(x)_X = \inf_{z \in \Sigma_k} \|x - z\|_X.$$

Adaptive compressed acquisition

Example

Let $r(x)$ be the non-increasing rearrangement of x, i.e., $r(x) = (|x_{i_1}|, \ldots, |x_{i_N}|)^T$ and $|x_{i_j}| \geq |x_{i_{j+1}}|$ for $j = 1, \ldots, N-1$.

Adaptive compressed acquisition

Example

Let $r(x)$ be the non-increasing rearrangement of x, i.e., $r(x) = (|x_{i_1}|, \ldots, |x_{i_N}|)^T$ and $|x_{i_j}| \geq |x_{i_{j+1}}|$ for $j = 1, \ldots, N-1$.Then it is straightforward to check that

$$\sigma_k(x)_{\ell_p^N} := \left(\sum_{j=k+1}^{N} r_j(x)^p \right)^{1/p}, \quad 1 \leq p < \infty.$$

Adaptive compressed acquisition

Example

Let $r(x)$ be the non-increasing rearrangement of x, i.e., $r(x) = (|x_{i_1}|, \ldots, |x_{i_N}|)^T$ and $|x_{i_j}| \geq |x_{i_{j+1}}|$ for $j = 1, \ldots, N-1$.Then it is straightforward to check that

$$\sigma_k(x)_{\ell_p^N} := \left(\sum_{j=k+1}^{N} r_j(x)^p \right)^{1/p}, \quad 1 \leq p < \infty.$$

In particular, the vector $x_{[k]}$ derived from x by setting to zero all the $N-k$ smallest entries in absolute value is called *the best k-term approximation* to x and it coincides with

$$x_{[k]} = \arg \min_{z \in \Sigma_k} \|x - z\|_{\ell_p^N}.$$

for any $1 \leq p < \infty$.

Adaptive compressed acquisition

The computation the best k-term approximation of $x \in \mathbb{R}^N$, in general requires the search of the largest entries of x in absolute value, and therefore the testing of all the entries of the vector x:

- I go to Aachen

Adaptive compressed acquisition

The computation the best k-term approximation of $x \in \mathbb{R}^N$, in general requires the search of the largest entries of x in absolute value, and therefore the testing of all the entries of the vector x:

- I go to Aachen
- I get a 10M pixel picture $\bar{x}$ of the Super C

Adaptive compressed acquisition

The computation the best k-term approximation of $x \in \mathbb{R}^N$, in general requires the search of the largest entries of x in absolute value, and therefore the testing of all the entries of the vector x:

- I go to Aachen
- I get a 10M pixel picture $\bar{x}$ of the Super C
- I go back home

Adaptive compressed acquisition

The computation the best k-term approximation of $x \in \mathbb{R}^N$, in general requires the search of the largest entries of x in absolute value, and therefore the testing of all the entries of the vector x:

- I go to Aachen
- I get a 10M pixel picture $\bar{x}$ of the Super C
- I go back home
- I realize that the picture $\bar{x}$ is too big

Adaptive compressed acquisition

The computation the best k-term approximation of $x \in \mathbb{R}^N$, in general requires the search of the largest entries of x in absolute value, and therefore the testing of all the entries of the vector x:

- I go to Aachen
- I get a 10M pixel picture $\bar{x}$ of the Super C
- I go back home
- I realize that the picture $\bar{x}$ is too big
- I compute a complete, i.e., *wavelet* or Fourier decomposition x of the image $\bar{x}$

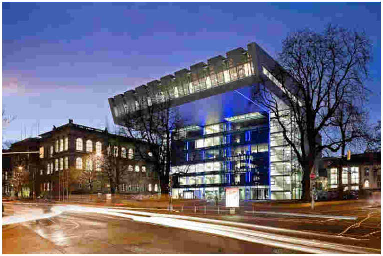
Adaptive compressed acquisition

The computation the best k-term approximation of $x \in \mathbb{R}^N$, in general requires the search of the largest entries of x in absolute value, and therefore the testing of all the entries of the vector x:

- I go to Aachen
- I get a 10M pixel picture $\bar{x}$ of the Super C
- I go back home
- I realize that the picture $\bar{x}$ is too big
- I compute a complete, i.e., *wavelet* or Fourier decomposition x of the image $\bar{x}$
- I eventually keep ONLY the best k-term approximation $x_{[k]}$ w.r.t. to wavelet or Fourier coordinates (JPEG)

This procedure is *adaptive*, since it depends on the particular vector.

Compressing Super C



Nonadaptive and compressed acquisition: compressed sensing

We would like to describe a *linear encoder* which allows to compute approximatively k measurements $(y_1, \ldots, y_k)^T$ and a *nearly optimal* approximation of x:

Nonadaptive and compressed acquisition: compressed sensing

We would like to describe a *linear encoder* which allows to compute approximatively k measurements $(y_1, \ldots, y_k)^T$ and a *nearly optimal* approximation of x:

Provided a set $K \subset \mathbb{R}^N$, there exists a linear map $A : \mathbb{R}^N \to \mathbb{R}^m$, with $m \approx k$ and a possibly nonlinear map $\Delta : \mathbb{R}^m \to \mathbb{R}^N$ such that

$$\|x - \Delta(Ax)\|_X \leq C\sigma_k(x)_X$$

for all $x \in K$.

Note that

- the way we encode x is via a prescribed map A which is independent of x as well as the decoding procedure Δ. This is why we call this strategy a *nonadaptive (or universal) and compressed acquisition* of x;

Note that

- the way we encode x is via a prescribed map A which is independent of x as well as the decoding procedure Δ. This is why we call this strategy a *nonadaptive (or universal) and compressed acquisition* of x;
- we recover an approximation to x from nearly k-linear measurements which is of the order of the k-best approximation error. In this sense we say that the performances of the encoder/decoder system (A, Δ) is nearly optimal.

Note that

- the way we encode x is via a prescribed map A which is independent of x as well as the decoding procedure Δ. This is why we call this strategy a *nonadaptive (or universal) and compressed acquisition* of x;
- we recover an approximation to x from nearly k-linear measurements which is of the order of the k-best approximation error. In this sense we say that the performances of the encoder/decoder system (A, Δ) is nearly optimal.

$$\|x - \Delta(Ax)\|_X \leq C\sigma_k(x)_X$$

An optimal decoder

Under a certain property, called the *Null Space Property* (NSP), depending on $k \leq N$, for a matrix A,

An optimal decoder

Under a certain property, called the *Null Space Property* (NSP), depending on $k \leq N$, for a matrix A,

the decoder, which we call ℓ_1-*minimization*,

$$\Delta(y) = \arg \min_{Az=y=Ax} \|x\|_{\ell_1^N}$$

An optimal decoder

Under a certain property, called the *Null Space Property* (NSP), depending on $k \leq N$, for a matrix A,

the decoder, which we call ℓ_1-*minimization*,

$$\Delta(y) = \arg \min_{Az=y=Ax} \|x\|_{\ell_1^N}$$

performs

$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N},$$

An optimal decoder

Under a certain property, called the *Null Space Property* (NSP), depending on $k \leq N$, for a matrix A,

the decoder, which we call ℓ_1-*minimization*,

$$\Delta(y) = \arg \min_{Az=y=Ax} \|x\|_{\ell_1^N}$$

performs

$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N},$$

as well as

$$\|x - \Delta(y)\|_{\ell_2^N} \leq C_2 \frac{\sigma_k(x)_{\ell_1^N}}{k^{1/2}},$$

for all $x \in \mathbb{R}^N$.

An optimal decoder

Under a certain property, called the *Null Space Property* (NSP), depending on $k \leq N$, for a matrix A,

the decoder, which we call ℓ_1-*minimization*,

$$\Delta(y) = \arg \min_{Az=y=Ax} \|x\|_{\ell_1^N}$$

performs

$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N},$$

as well as

$$\|x - \Delta(y)\|_{\ell_2^N} \leq C_2 \frac{\sigma_k(x)_{\ell_1^N}}{k^{1/2}},$$

for all $x \in \mathbb{R}^N$.

The next question we will address is the existence of matrices A with NSP for which k is optimal, i.e.,

$$k \approx m.$$

Surprising result

Property

$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N},$$

ensures that if the vector $x \in \Sigma_k$, then ℓ_1-minimization will be able to recover it *exactly*, as $x = x_{[k]}$ and $\sigma_k(x_{[k]}) = 0$.

Surprising result

Property

$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N},$$

ensures that if the vector $x \in \Sigma_k$, then ℓ_1-minimization will be able to recover it *exactly*, as $x = x_{[k]}$ and $\sigma_k(x_{[k]}) = 0$. This result is quite surprising because the problem of recovering a sparse vector, or the solution of the following optimization

$$\min \|x\|_{\ell_0^N} \text{ subject to } Ax = y,$$

is know to be *NP-complete* (Mallat-Zhang '93, Natarajan '95) .

Surprising result

Property

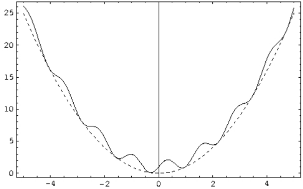
$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N},$$

ensures that if the vector $x \in \Sigma_k$, then ℓ_1-minimization will be able to recover it *exactly*, as $x = x_{[k]}$ and $\sigma_k(x_{[k]}) = 0$. This result is quite surprising because the problem of recovering a sparse vector, or the solution of the following optimization

$$\min \|x\|_{\ell_0^N} \text{ subject to } Ax = y,$$

is know to be *NP-complete* (Mallat-Zhang '93, Natarajan '95) . Instead interior-point methods are guaranteed to solve the ℓ_1-minimization problem to a fixed precision in time $\mathcal{O}(m^2 N^{1.5})$ (Nesterov-Nemirovskii '94).

Convexification

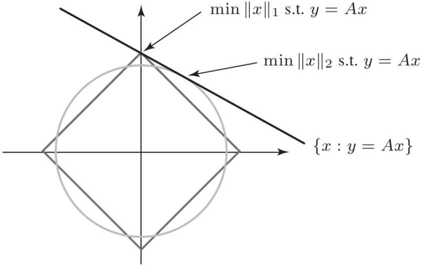


One rewrites

$$\|x\|_{\ell_0^N} := \sum_{j=1}^{N} |x_j|_0, \quad |t|_0 := \begin{cases} 0, & t = 0 \\ 1, & 0 < t \leq 1 \end{cases} .$$

Its convex envelope in $B_{\ell_\infty^N}(R) \cap \{z : Az = y\}$ is bounded below by $\frac{1}{R}\|x\|_{\ell_1^N} := \frac{1}{R}\sum_{j=1}^{N} |x_j|$.

Geometry



Assume $N = 2$ and $m = 1$. Hence $\mathcal{F}(y) = \{z : Az = y\}$ is just a line in $\mathbb{R}^2$. If we exclude that there exists $\eta \in \ker A$ such that $|\eta_1| = |\eta_2|$ or, equivalently,

$$|\eta_i| < |\eta_{\{1,2\}\setminus\{i\}}|$$

for all $\eta \in \ker A$ and for one $i = 1, 2$, then the solution to (ℓ_1) is a solution of (ℓ_0).

Null Space Property (NSP)

Definition

One says that $A \in \mathbb{R}^{m \times N}$ has the *Null Space Property* (NSP) of order k for $0 < \gamma < 1$ if

$$\|\eta_T\|_{\ell_1^N} \leq \gamma \|\eta_{T^c}\|_{\ell_1^N},$$

for all sets $T \subset \{1, \ldots, N\}$, $\# T \leq k$ and for all $\eta \in \mathcal{N} = \ker A$.

Null Space Property (NSP)

Definition

One says that $A \in \mathbb{R}^{m \times N}$ has the *Null Space Property* (NSP) of order k for $0 < \gamma < 1$ if

$$\|\eta_T\|_{\ell_1^N} \leq \gamma \|\eta_{T^c}\|_{\ell_1^N},$$

for all sets $T \subset \{1, \ldots, N\}$, $\# T \leq k$ and for all $\eta \in \mathcal{N} = \ker A$.

- The NSP is equivalent to stable recovery, i.e.,

$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N} \Rightarrow NSP;$$

- the NSP is used in algorithms to prove convergence rates and stability.

Restricted Isometry Property (RIP)

Definition

One says that $A \in \mathbb{R}^{m \times N}$ has the RIP of order K if there exists $0 < \delta_K < 1$ such that

$$(1 - \delta_K)\|z\|_{\ell_2^N} \leq \|Az\|_{\ell_2^m} \leq (1 + \delta_K)\|z\|_{\ell_2^N},$$

for all $z \in \Sigma_K$.

Restricted Isometry Property (RIP)

Definition

One says that $A \in \mathbb{R}^{m \times N}$ has the RIP of order K if there exists $0 < \delta_K < 1$ such that

$$(1 - \delta_K)\|z\|_{\ell_2^N} \leq \|Az\|_{\ell_2^m} \leq (1 + \delta_K)\|z\|_{\ell_2^N},$$

for all $z \in \Sigma_K$.

- The RIP turns out to be very useful in the analysis of stability of certain algorithms;
- the RIP is also introduced because it implies the *Null Space Property*, and when dealing with random matrices it is more easily addressed.

Restricted Isometry Property (RIP)

Definition

One says that $A \in \mathbb{R}^{m\times N}$ has the RIP of order K if there exists $0 < \delta_K < 1$ such that

$$(1-\delta_K)\|z\|_{\ell_2^N} \leq \|Az\|_{\ell_2^m} \leq (1+\delta_K)\|z\|_{\ell_2^N},$$

for all $z \in \Sigma_K$.

- The RIP turns out to be very useful in the analysis of stability of certain algorithms;
- the RIP is also introduced because it implies the *Null Space Property*, and when dealing with random matrices it is more easily addressed.

Lemma

Assume that $A \in \mathbb{R}^{m\times N}$ has the RIP of order $K = k + h$ with $0 < \delta_K < 1$. Then A has the NSP of order k and constant $\gamma = \sqrt{\frac{k}{h}\frac{1+\delta_K}{1-\delta_K}}$.

Stability results

Theorem

Let $A \in \mathbb{R}^{m \times N}$ which satisfies the RIP of order $2k$ with $\delta_{2k} \leq \delta$ small enough, then the decoder $\Delta =$ "ℓ_1-minimization" satisfies

$$\|x - \Delta(y)\|_{\ell_1^N} \leq C_1 \sigma_k(x)_{\ell_1^N}.$$

Stability results: noise case

Theorem

Let $A \in \mathbb{R}^{m\times N}$ which satisfies the RIP of order $2k$ with δ_{2k} sufficiently small. Assume further that $Ax + e = y$ where e is a measurement error. Then the decoder Δ as the further enhanced stability property:

$$\|x - \Delta(y)\|_{\ell_2^N} \leq C_3 \left(\sigma_k(x)_{\ell_2^N} + \frac{\sigma_k(x)_{\ell_1^N}}{k^{1/2}} + \|e\|_{\ell_2^N} \right).$$

The class of optimal RIP matrices is not empty

We would like to mention how for different classes of random matrices it is possible to show that the RIP property can hold with optimal constants, i.e.,

$$k \asymp \frac{m}{\log N/m + 1}.$$

at least with high probability. This implies in particular, that such matrices exist, they are frequent, but they are given to us only with an uncertainty.

Random matrices with concentration properties

Let (Ω, ρ) be a probability measure space and X a random variable on (Ω, ρ). One can define a random matrix $A(\omega)$, $\omega \in \Omega^{mN}$, as the matrix whose entries are independent realizations of X. We assume further that $\|A(\omega)x\|_{\ell_2^N}^2$ has expected value $\|x\|_{\ell_2^N}^2$ and

$$\mathbb{P}\left(\left| \|A(\omega)x\|_{\ell_2^N}^2 - \|x\|_{\ell_2^N}^2 \right| \geq \varepsilon \|x\|_{\ell_2^N}^2 \right) \leq 2e^{-mc_0(\varepsilon)}, \quad 0 < \varepsilon < 1.$$

Classical examples

Example

Here we collect two of the most relevant examples for which the concentration property holds:

1. One can choose, for instance, the entries of A as i.i.d. Gaussian random variables, $A_{ij} \sim \mathcal{N}(0, \frac{1}{m})$, and $c_0(\varepsilon) = \varepsilon^2/4 - \varepsilon^3/6$. This can be shown by using Chernoff inequalities and a comparison of the moments of a Bernoulli random variable to those of a Gaussian random variable;

2. One can also use matrices where the entries are independent realizations of ±1 Bernoulli random variables

$$A_{ij} = \begin{cases} +1/\sqrt{m}, & \text{with probability } \frac{1}{2} \\ -1/\sqrt{m}, & \text{with probability } \frac{1}{2} \end{cases} .$$

RIP with high probability

Theorem

Suppose that m, N *and* $0 < \delta < 1$ *are fixed. If* $A(\omega), \omega \in \Omega^{mN}$ *is a random matrix of size* $m \times N$ *with the concentration property, then there exist constants* $c_1, c_2 > 0$ *depending on* δ *such that the RIP holds for* $A(\omega)$ *with constant* δ *and* $k \leq c_1 \frac{m}{\log(N/m)+1}$ *with probability exceeding* $1 - 2e^{-c_2 m}$.

Numerical methods for compressed sensing

The ℓ_1-minimization problem

$$\min \|x\|_{\ell_1} \quad \text{subject to } Ax = y$$

is equivalent to the linear program

$$\min \sum_{j=1}^{2N} v_j \quad \text{subject to} \quad v \geq 0,\ (A| - A)v = y.$$

The solution x^* is obtained from the solution v^* via $x^* = (I| - I)v^*$. Any linear programming method may therefore be used. Interior point methods apply in particular with complexity $\mathcal{O}(m^2 N^{1.5})$.

Faster iterative methods

In applications it is important to have fast(er) methods for actually solving ℓ_1-minimization and to have similar guarantees of stability. We present

- the iteratively reweighted least square method (IRLS)
- a variant of IRLS for low-rank matrix recovery
- iterative hard thresholding
- iterative soft-thresholding and its variations

A rough description

Denote $\mathcal{F}(y) = \{x : Ax = y\}$ and $\mathcal{N} = \ker A$. Let us start with a few non-rigorous observations; next we will be more precise. For $t \neq 0$ we simply have

$$|t| = \frac{t^2}{|t|}.$$

A rough description

Denote $\mathcal{F}(y) = \{x : Ax = y\}$ and $\mathcal{N} = \ker A$. Let us start with a few non-rigorous observations; next we will be more precise. For $t \neq 0$ we simply have

$$|t| = \frac{t^2}{|t|}.$$

Hence, an ℓ_1-minimization can be recasted into a weighted ℓ_2-minimization, and we may expect

$$\arg \min_{x \in \mathcal{F}(y)} \sum_{j=1}^{N} |x_j| \approx \arg \min_{x \in \mathcal{F}(y)} \sum_{j=1}^{N} x_j^2 |x_j^*|^{-1},$$

as soon as x^* is the wanted ℓ_1-norm minimizer.

A rough description

Denote $\mathcal{F}(y) = \{x : Ax = y\}$ and $\mathcal{N} = \ker A$. Let us start with a few non-rigorous observations; next we will be more precise. For $t \neq 0$ we simply have

$$|t| = \frac{t^2}{|t|}.$$

Hence, an ℓ_1-minimization can be recasted into a weighted ℓ_2-minimization, and we may expect

$$\arg \min_{x \in \mathcal{F}(y)} \sum_{j=1}^{N} |x_j| \approx \arg \min_{x \in \mathcal{F}(y)} \sum_{j=1}^{N} x_j^2 |x_j^*|^{-1},$$

as soon as x^* is the wanted ℓ_1-norm minimizer.

- the advantage is that minimizing a smooth quadratic function $|t|^2$ is better than addressing the minimization of the nonsmooth function $|t|$;

A rough description

Denote $\mathcal{F}(y) = \{x : Ax = y\}$ and $\mathcal{N} = \ker A$. Let us start with a few non-rigorous observations; next we will be more precise. For $t \neq 0$ we simply have

$$|t| = \frac{t^2}{|t|}.$$

Hence, an ℓ_1-minimization can be recasted into a weighted ℓ_2-minimization, and we may expect

$$\arg \min_{x \in \mathcal{F}(y)} \sum_{j=1}^{N} |x_j| \approx \arg \min_{x \in \mathcal{F}(y)} \sum_{j=1}^{N} x_j^2 |x_j^*|^{-1},$$

as soon as x^* is the wanted ℓ_1-norm minimizer.

- the advantage is that minimizing a smooth quadratic function $|t|^2$ is better than addressing the minimization of the nonsmooth function $|t|$;
- the obvious drawbacks are that neither we dispose of x^* a priori nor we can expect that $x_j^* \neq 0$ for all $i = 1, \ldots, N$, since we hope for k-sparse solutions.

A rough description

We start by assuming that we dispose of a good approximation w_j^n of $|(x_j^*)^2 + \epsilon_n^2|^{-1/2} \approx |x_j^*|^{-1}$ and we compute

$$x^{n+1} = \arg \min_{x \in \mathcal{F}(y)} \sum_{j=1}^{N} x_j^2 w_j^n,$$

then we up-date $\epsilon_{n+1} \leq \epsilon_n$, we define

$$w_j^{n+1} = |(x_j^n)^2 + \epsilon_{n+1}^2|^{-1/2},$$

and we iterate the process. The hope is that a proper choice of $\epsilon_n \to 0$ will allow for the computation of an ℓ_1-minimizer.

Variational interpretation

Our analysis of the algorithm starts from the observation that

$$|t| = \min_{w>0} \frac{1}{2}\left(wt^2 + w^{-1}\right),$$

the minimum being reached for $w = \frac{1}{|t|}$.

Variational interpretation

Our analysis of the algorithm starts from the observation that

$$|t| = \min_{w>0} \frac{1}{2}\left(wt^2 + w^{-1}\right),$$

the minimum being reached for $w = \frac{1}{|t|}$. Given a real number $\epsilon > 0$ and a weight vector $w \in \mathbb{R}^N$, with $w_j > 0$, $j = 1, \ldots, N$, we define

$$\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N}(\epsilon^2 w_j + w_j^{-1})\right], \quad z \in \mathbb{R}^N.$$

Variational interpretation

The algorithm can be recasted as an alternating method for choosing minimizers and weights based on the functional $\mathcal{J}$.

Algorithm 1. We initialize by taking $w^0 := (1, \dots, 1)$. We also set $\epsilon_0 := 1$. We then recursively define for $n = 0, 1, \dots,$

$$x^{n+1} := \arg \min_{z \in \mathcal{F}(y)} \mathcal{J}(z, w^n, \epsilon_n) = \arg \min_{z \in \mathcal{F}(y)} \|z\|_{\ell_2(w^n)}$$

and

$$\epsilon_{n+1} := \min(\epsilon_n, \frac{r(x^{n+1})_{K+1}}{N}),$$

where K is a fixed integer that will be described more fully later. We also define

$$w^{n+1} := \arg \min_{w > 0} \mathcal{J}(x^{n+1}, w, \epsilon_{n+1}).$$

We stop the algorithm if $\epsilon_n = 0$; in this case we define $x^j := x^n$ for $j > n$. However, in general, the algorithm will generate an infinite sequence $(x^n)_{n \in N}$ of distinct vectors.

Sketched convergence properties

Note that for each $n = 1, 2, \ldots,$ we have

$$\mathcal{J}(x^{n+1}, w^{n+1}, \epsilon_{n+1}) = \sum_{j=1}^{N} [(x_j^{n+1})^2 + \epsilon_{n+1}^2]^{1/2}.$$

Sketched convergence properties

Note that for each $n = 1, 2, \ldots$, we have

$$\mathcal{J}(x^{n+1}, w^{n+1}, \epsilon_{n+1}) = \sum_{j=1}^{N} [(x_j^{n+1})^2 + \epsilon_{n+1}^2]^{1/2}.$$

We also have the following monotonicity property which holds for all $n \geqslant 0$:

$$\begin{aligned} & \mathcal{J}(x^{n+1}, w^{n+1}, \epsilon_{n+1}) \leqslant \mathcal{J}(x^{n+1}, w^{n}, \epsilon_{n+1}) \\ \leqslant \quad & \mathcal{J}(x^{n+1}, w^{n}, \epsilon_{n}) \leqslant \mathcal{J}(x^{n}, w^{n}, \epsilon_{n}). \end{aligned}$$

Sketched convergence properties

Note that for each $n = 1, 2, \ldots$, we have

$$\mathcal{J}(x^{n+1}, w^{n+1}, \epsilon_{n+1}) = \sum_{j=1}^{N} [(x_j^{n+1})^2 + \epsilon_{n+1}^2]^{1/2}.$$

We also have the following monotonicity property which holds for all $n \geqslant 0$:

$$\begin{aligned} & \mathcal{J}(x^{n+1}, w^{n+1}, \epsilon_{n+1}) \leqslant \mathcal{J}(x^{n+1}, w^{n}, \epsilon_{n+1}) \\ \leqslant \quad & \mathcal{J}(x^{n+1}, w^{n}, \epsilon_{n}) \leqslant \mathcal{J}(x^{n}, w^{n}, \epsilon_{n}). \end{aligned}$$

Lemma

For each $n \geqslant 1$ we have

$$\|x^n\|_{\ell_1} \leqslant \mathcal{J}(x^1, w^0, \epsilon_0) =: \mathcal{A}$$

and

$$w_j^n \geqslant \mathcal{A}^{-1}, \quad j = 1, \ldots, N.$$

Sketched convergence properties

As

$$\mathcal{A}^{-1}\|x^{n+1}-x^n\|_{\ell_2}^2 \leq 2(\mathcal{J}(x^n,w^n,\epsilon_n)-\mathcal{J}(x^{n+1},w^{n+1},\epsilon_{n+1})), \forall n$$

we obtain (by telescopic sum)

$$\sum_{n=1}^{\infty}\|x^{n+1}-x^n\|_{\ell_2}^2 \leqslant 2\mathcal{A}^2.$$

In particular, we have

$$\lim_{n\to\infty}(x^n-x^{n+1})=0.$$

(Super)linear rate of convergence under NSP

The linear rate can be improved significantly, by a very simple modification of the rule of updating the weight:

$$w_j^{n+1} = \left((x_j^{n+1})^2 + \epsilon_{n+1}^2\right)^{-\frac{2-\tau}{2}}, \quad j = 1, \ldots, N, \text{ for any } 0 < \tau < 1.$$

(Super)linear rate of convergence under NSP

The linear rate can be improved significantly, by a very simple modification of the rule of updating the weight:

$$w_j^{n+1} = \left((x_j^{n+1})^2 + \epsilon_{n+1}^2\right)^{-\frac{2-\tau}{2}}, \quad j = 1, \ldots, N, \text{ for any } 0 < \tau < 1.$$

This corresponds to the substitution of the function $\mathcal{J}$ with

$$\mathcal{J}_\tau(z, w, \epsilon) := \frac{\tau}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N}\left(\epsilon^2 w_j + \frac{2-\tau}{\tau}\frac{1}{w_j^{\frac{\tau}{2-\tau}}}\right)\right].$$

Surprisingly the rate of local convergence of this modified algorithm is superlinear.

(Super)linear rate of convergence under NSP

The rate is larger for smaller τ, increasing to approach a quadratic regime as $\tau \to 0$. More precisely the local error $E_n := \|x^n - x^*\|_{\ell_\tau^N}^\tau$ satisfies

$$E_{n+1} \leqslant \mu(\gamma, \tau) E_n^{2-\tau},$$

where $\mu(\gamma, \tau) < 1$ for $\gamma > 0$ sufficiently small.

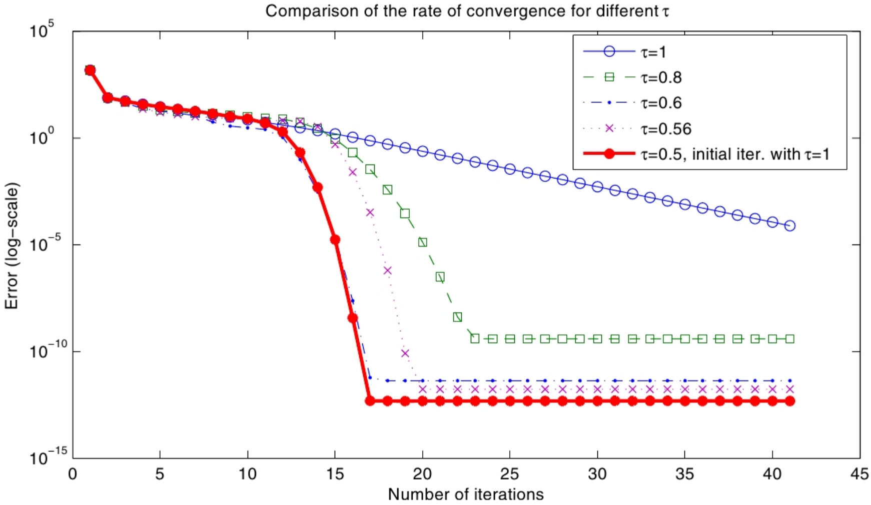
(Super)linear rate of convergence under NSP

The rate is larger for smaller τ, increasing to approach a quadratic regime as $\tau \to 0$. More precisely the local error $E_n := \|x^n - x^*\|_{\ell_\tau^N}^\tau$ satisfies

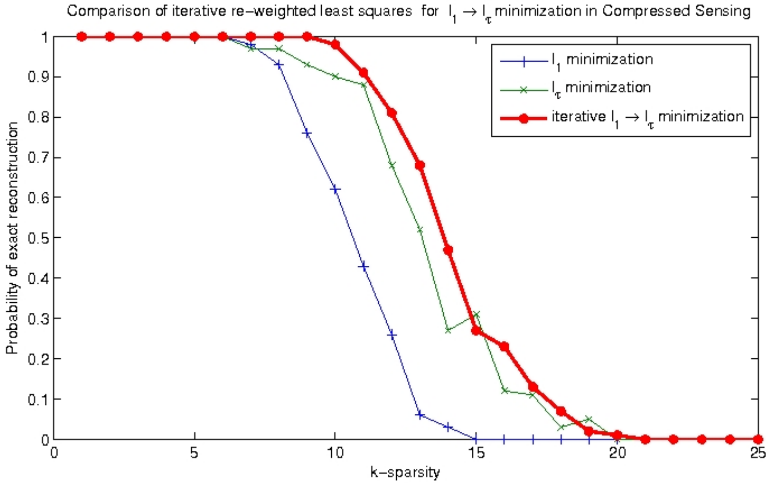
$$E_{n+1} \leqslant \mu(\gamma, \tau) E_n^{2-\tau},$$

where $\mu(\gamma, \tau) < 1$ for $\gamma > 0$ sufficiently small. The validity is restricted to x^n in a (small) ball centered at x^*. In particular, if x^0 is close enough to x^* then the estimate ensures the convergence of the algorithm to the k-sparse solution x^*.

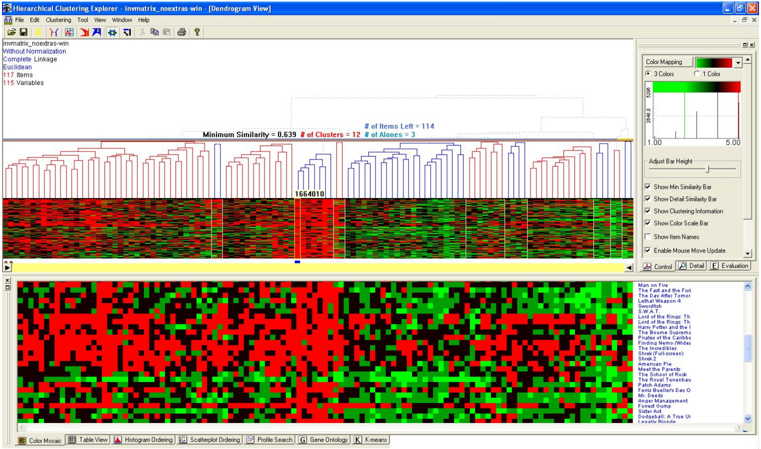
Reweighted iterative least squares: ℓ_τ minimization $\tau < 1$



Re-weighted Iterative Least Squares: ℓ_τ minimization $\tau < 1$



Rating movies as low-rank matrix completion problem



Hierarchical Clustering Explorer - invmatrix_noextras-win - [Dendrogram View]

invmatrix_noextras-win
Without Normalization
Complete Linkage
Euclidean
117 Items
115 Variables

# of Items Left = 114
Minimum Similarity = 0.639 # of Clusters = 12 # of Alones = 3

1664010

Color Mapping
● 3 Colors ○ 1 Color

Adjust Bar Height

☑ Show Min Similarity Bar
☑ Show Detail Similarity Bar
☑ Show Clustering Information
☑ Show Color Scale Bar
☐ Show Item Names
☑ Enable Mouse Move Update

Man on Fire
The Fast and the Furi
The Day After Tomor
Lethal Weapon 4
Swordfish
S.W.A.T.
Lord of the Rings: Th
Lord of the Rings: Th
Harry Potter and the (
The Bourne Suprema
Pirates of the Caribbe
Finding Nemo (Wides
The Incredibles
Shrek (Full-screen)
Shrek 2
American Pie
Meet the Parents
The School of Rock
The Royal Tenenbau
Patch Adams
Ferris Bueller's Day O
Mr. Deeds
Anger Management
Forrest Gump
Sister Act
Dodgeball: A True Ur
Legally Blonde

Low-rank matrix completion

Low-rank matrix identification from few linear measurements: nuclear norm minimization.

$$\text{argmin}_{\{X_{ij}=M_{ij}:ij\in\Omega\}} \text{rank}(X) \quad \Leftrightarrow \quad \text{argmin}_{\{X_{ij}=M_{ij}:ij\in\Omega\}} \sum_{i=1}^{n} \sigma_i(X).$$

Theorem

Let M be a "generic" $n_1 \times n_2$ matrix of rank r and $n = \max(n_1, n_2)$. Suppose we observe m entries of M uniformly at random on Ω. Then there exist $C, c > 0$ such that if

$$m \geq C n^{5/4} r (\beta \log n),$$

then the solution X^* to

$$\text{argmin}_{\{X_{ij}=M_{ij}:ij\in\Omega\}} \sum_{i=1}^{n} \sigma_i(X).$$

is unique and concides with M with probability $1 - cn^{-\beta}$, for $\beta > 0$.

The IRLS algorithm adapted to matrices

Algorithm 2. We initialize by $W^{(0)} = 1$, $\gamma < 1$, and $\varepsilon_0 = 1$. Then, recursively

$$
\begin{aligned}
X^{(n+1)} &:= \operatorname{argmin}_{\{X_{ij}=M_{ij}:ij\in\Omega\}} \|XW^{(n)}\|_F, \\
\varepsilon_{n+1} &:= \min\{\varepsilon_n, \gamma\sigma_{K+1}(X^{(n+1)})\} \\
W^{(n+1)} &:= [(X^{(n+1)})^* X^{(n+1)} + I \cdot \varepsilon_{n+1}^2]^{-1/4}
\end{aligned}
$$

Theorem

Let M be a "generic" $n_1 \times n_2$ matrix of rank r and $n = \max(n_1, n_2)$. Suppose we observe m entries of M uniformly at random on Ω. Then there exist $C, c > 0$ such that if

$$m \geq Cn^{5/4} r(\beta \log n).$$

and $\varepsilon_n \to 0$, then the algorithm converges to a matrix $\bar{X}$ which concides with M with probability $1 - cn^{-\beta}$, for $\beta > 0$.

Iterative Hard Thresholding

Algorithm 3. We initialize by taking $x^0 = 0$. We iterate

$$x^{n+1} = \mathbb{H}_k(x^n + A^*(y - Ax^n)),$$

where

$$\mathbb{H}_k(x) = x_{[k]},$$

is the operator which returns the best k-term approximation to x.

Note that if x^* is k-sparse and $Ax^* = y$, then x^* is a fixed point of

$$x^* = \mathbb{H}_k(x^* + A^*(y - Ax^*)).$$

This algorithm can be seen as a minimizing method for the functional

$$\mathcal{J}(x) = \|y - Ax\|_{\ell_2^N}^2 + 2\alpha\|x\|_{\ell_0^N},$$

for a suitable $\alpha = \alpha(k) > 0$.

Convergence properties

Theorem

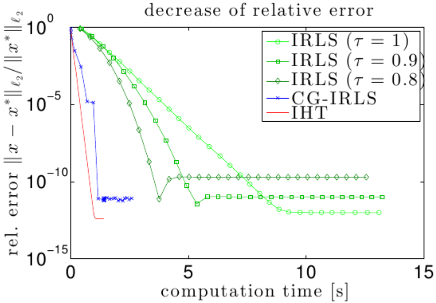
Let us assume that $y = Ax + e$ is a noisy encoding of x via A, where x is an arbitrary vector. If A has the RIP of order $3k$ and constant $\delta_{3k}^2 < \frac{1}{\sqrt{32}}$, then after at most

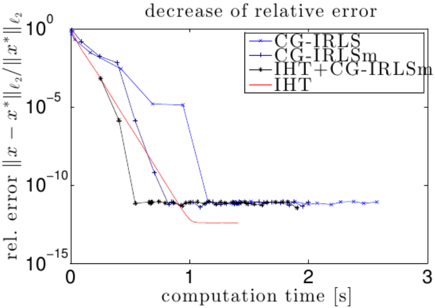
$$n^* = \left\lceil \log_2 \left(\frac{\|x\|_{\ell_2^N}}{\|e\|_{\ell_2^N}} \right) \right\rceil$$

iterations, the algorithm estimates x with accuracy

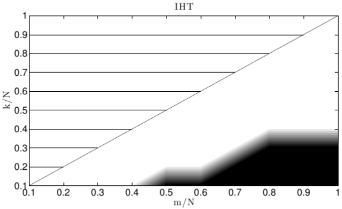
$$\|x - x^{n^*}\|_{\ell_2^N} \leq 7 \left(\sigma_k(x)_{\ell_2^N} + \frac{\sigma_k(x)_{\ell_1^N}}{\sqrt{k}} + \|e\|_{\ell_2^N} \right).$$

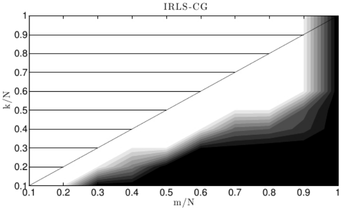
Some numerical comparisons





Phase transition diagrams: empirical rate of success





ℓ_1 minimization as a regularization for inverse problems

We are interested in

$$J(u) := \|Ku - y\|_Y^2 + 2\|(\langle u, \tilde{\psi}_\lambda \rangle)_{\lambda \in \mathcal{I}}\|_{\ell_{1,\alpha}(\mathcal{I})},$$

where $K : X \to Y$ is a bounded linear operator acting between two separable Hilbert spaces X and Y, $y \in Y$ is a given datum, and $\Psi := \{\psi_\lambda\}_{\lambda \in \mathcal{I}}$ is a prescribed countable basis for X with associated dual $\tilde{\Psi} := \{\tilde{\psi}_\lambda\}_{\lambda \in \mathcal{I}}$.

ℓ_1 minimization as a regularization for inverse problems

Associated to the basis, we are given the synthesis map $F : \ell_2(\mathcal{I}) \to X$ defined by

$$Fu := \sum_{\lambda \in \mathcal{I}} u_\lambda \psi_\lambda, \quad u \in \ell_2(\mathcal{I}).$$

ℓ_1 minimization as a regularization for inverse problems

Associated to the basis, we are given the synthesis map $F : \ell_2(\mathcal{I}) \to X$ defined by

$$Fu := \sum_{\lambda \in \mathcal{I}} u_\lambda \psi_\lambda, \quad u \in \ell_2(\mathcal{I}).$$

We can re-formulate equivalently the functional in terms of sequences in $\ell_2(\mathcal{I})$ as follows:

$$J(u) := J_\alpha(u) = \|(K \circ F)u - y\|_Y^2 + 2\|u\|_{\ell_{1,\alpha}(\mathcal{I})}.$$

For ease of notation let us write $A := K \circ F$.

A minimizing algorithm: iterative thresholding

Several authors have proposed an iterative soft-thresholding algorithm to approximate a minimizer $u^* := u_\alpha^*$, which is the limit of sequences $u^{(n)}$ defined recursively by

$$u^{(n+1)} = \mathbb{S}_\alpha \left[u^{(n)} + A^* y - A^* A u^{(n)} \right] \quad ,$$

starting from an arbitrary $u^{(0)}$,

A minimizing algorithm: iterative thresholding

Several authors have proposed an iterative soft-thresholding algorithm to approximate a minimizer $u^* := u^*_\alpha$, which is the limit of sequences $u^{(n)}$ defined recursively by

$$u^{(n+1)} = \mathbb{S}_\alpha \left[u^{(n)} + A^* y - A^* A u^{(n)} \right] \quad ,$$

starting from an arbitrary $u^{(0)}$, where $\mathbb{S}_\alpha$ is the soft-thresholding $\mathbb{S}_\alpha(u)_\lambda = S_{\alpha_\lambda}(u_\lambda)$ with

$$S_\alpha(x) = \begin{cases} x - \alpha & x > \alpha \\ 0 & |x| \le \alpha \\ x + \alpha & x < -\alpha \end{cases} \quad .$$

The surrogate functional

The algorithm can be recasted into an iterated minimization of a properly augmented functional, which we call the *surrogate functional* of $\mathcal{J}$,

$$\mathcal{J}^S(u,a) := \|Au - y\|_Y^2 + 2\|u\|_{\ell_{1,\alpha}(\mathcal{I})} + \|u-a\|_{\ell_2(\mathcal{I})}^2 - \|Au - Aa\|_Y^2.$$

The surrogate functional

The algorithm can be recasted into an iterated minimization of a properly augmented functional, which we call the *surrogate functional* of $\mathcal{J}$,

$$\mathcal{J}^S(u,a) := \|Au-y\|_Y^2 + 2\|u\|_{\ell_{1,\alpha}(\mathcal{I})} + \|u-a\|_{\ell_2(\mathcal{I})}^2 - \|Au-Aa\|_Y^2.$$

Assume here and later that $\|A\| < 1$.

The surrogate functional

The algorithm can be recasted into an iterated minimization of a properly augmented functional, which we call the *surrogate functional* of $\mathcal{J}$,

$$\mathcal{J}^S(u,a) := \|Au-y\|_Y^2 + 2\|u\|_{\ell_{1,\alpha}(\mathcal{I})} + \|u-a\|_{\ell_2(\mathcal{I})}^2 - \|Au-Aa\|_Y^2.$$

Assume here and later that $\|A\| < 1$. Observe that

$$\|u-a\|_{\ell_2(\mathcal{I})}^2 - \|Au-Aa\|_Y^2 \geq C\|u-a\|_{\ell_2(\mathcal{I})}^2,$$

for $C = (1 - \|A\|^2) > 0$.

The surrogate functional

The algorithm can be recasted into an iterated minimization of a properly augmented functional, which we call the *surrogate functional* of $\mathcal{J}$,

$$\mathcal{J}^S(u,a) := \|Au-y\|_Y^2 + 2\|u\|_{\ell_{1,\alpha}(\mathcal{I})} + \|u-a\|_{\ell_2(\mathcal{I})}^2 - \|Au-Aa\|_Y^2.$$

Assume here and later that $\|A\| < 1$. Observe that

$$\|u-a\|_{\ell_2(\mathcal{I})}^2 - \|Au-Aa\|_Y^2 \geq C\|u-a\|_{\ell_2(\mathcal{I})}^2,$$

for $C = (1-\|A\|^2) > 0$. Hence

$$\mathcal{J}(u) = \mathcal{J}^S(u,u) \leq \mathcal{J}^S(u,a),$$

and

$$\mathcal{J}^S(u,a) - \mathcal{J}^S(u,u) \geq C\|u-a\|_{\ell_2(\mathcal{I})}^2.$$

The surrogate functional

We can express the optimization of $\mathcal{J}^S(u,a)$ with respect to u explicitly by

$$\mathbb{S}_\alpha(a + A^*(y - Aa)) = \arg\min_{u\in\ell_2(\mathcal{I})} \mathcal{J}^S(u,a).$$

Algorithm 4. We initialize by taking any $u^{(0)} \in \ell_2(\mathcal{I})$. We iterate

$$\begin{aligned} u^{(n+1)} &= \mathbb{S}_\alpha\left[u^{(n)} + A^*y - A^*Au^{(n)}\right] \\ &= \arg\min_{u\in\ell_2(\mathcal{I})} \mathcal{J}^S(u,u^{(n)}). \end{aligned}$$

Sketch on convergence

As

$$\mathcal{J}(u^{(n)}) - \mathcal{J}(u^{(n+1)}) \geq C \| u^{(n+1)} - u^{(n)} \|_{\ell_2(\mathcal{I})}^2$$

we obtain the numerical convergence

$$\lim_{n \to \infty} \| u^{(n+1)} - u^{(n)} \|_{\ell_2(\mathcal{I})}^2 = 0.$$

Acceleration principles

We illustrate acceleration principles. We emphasize three main ingredients in particular:

- *the problem is set in infinite dimensions*;

Acceleration principles

We illustrate acceleration principles. We emphasize three main ingredients in particular:

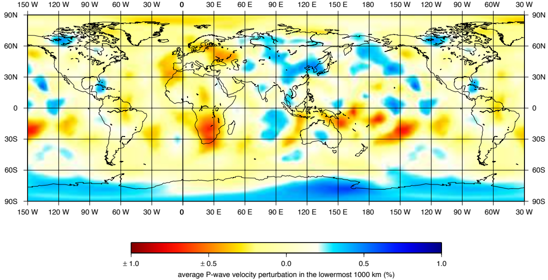
- *the problem is set in infinite dimensions*;
- *multiscale preconditioning*;

Acceleration principles

We illustrate acceleration principles. We emphasize three main ingredients in particular:

- *the problem is set in infinite dimensions*;
- *multiscale preconditioning*;
- *adaptivity*.

Global terrestrial seismic tomography (matrices $31m \times 200k$)



Finite frequency model + sparsity recovery methods = a new generation Earth model;

Rate of convergence results

Exponential (linear rate) convergence, i.e.,

$$\max\left\{\|u^{(n)} - u^*\|, J_\alpha(u^{(n)}) - J_\alpha(u^*)\right\} \leq C\gamma^n, \quad \gamma < 1,$$

can be ensured, e.g., when

A fulfills the so-called *finite basis injectivity* (FBI) condition (K. Bredies and D. Lorenz), i.e., for any finite set $\Lambda \subset \mathcal{I}$, the restriction A_Λ is injective.

Rate of convergence results

- We have strong convergence of the $u^{(n)}$ to a finitely supported limit sequence u^*.

Rate of convergence results

- We have strong convergence of the $u^{(n)}$ to a finitely supported limit sequence u^{*}.
- There exists a finite index set $\Lambda \subset \mathcal{I}$ such that all iterates $u^{(n)}$ and u^{*} are supported in Λ.

Rate of convergence results

- We have strong convergence of the $u^{(n)}$ to a finitely supported limit sequence u^*.
- There exists a finite index set $\Lambda \subset \mathcal{I}$ such that all iterates $u^{(n)}$ and u^* are supported in Λ.
- By the FBI condition, A_Λ is injective and hence $A^*A|_{\Lambda\times\Lambda}$ is boundedly invertible, so that $I - A_\Lambda^* A_\Lambda$ is a contraction on $\ell_2(\Lambda)$.

Rate of convergence results

- We have strong convergence of the $u^{(n)}$ to a finitely supported limit sequence u^*.
- There exists a finite index set $\Lambda \subset \mathcal{I}$ such that all iterates $u^{(n)}$ and u^* are supported in Λ.
- By the FBI condition, A_Λ is injective and hence $A^*A|_{\Lambda\times\Lambda}$ is boundedly invertible, so that $I - A_\Lambda^* A_\Lambda$ is a contraction on $\ell_2(\Lambda)$.
- Using
$$u_\Lambda^{(n+1)} = \mathbb{S}_\alpha\big(u_\Lambda^{(n)} + A_\Lambda^*(y - A_\Lambda u_\Lambda^{(n)})\big)$$

Rate of convergence results

- We have strong convergence of the $u^{(n)}$ to a finitely supported limit sequence u^*.
- There exists a finite index set $\Lambda \subset \mathcal{I}$ such that all iterates $u^{(n)}$ and u^* are supported in Λ.
- By the FBI condition, A_Λ is injective and hence $A^*A|_{\Lambda\times\Lambda}$ is boundedly invertible, so that $I - A_\Lambda^* A_\Lambda$ is a contraction on $\ell_2(\Lambda)$.
- Using
$$u_\Lambda^{(n+1)} = \mathbb{S}_\alpha\big(u_\Lambda^{(n)} + A_\Lambda^*(y - A_\Lambda u_\Lambda^{(n)})\big)$$

it follows $\|u^* - u^{(n+1)}\|_{\ell_2(\mathcal{I})} \leq \gamma \|u^* - u^{(n)}\|_{\ell_2(\mathcal{I})}$, where
$\gamma = \max\{|1 - \|(A^*A|_{\Lambda\times\Lambda})^{-1}\|^{-1}|, |\|A^*A|_{\Lambda\times\Lambda}\| - 1|\} \in (0,1)$.

Rate of convergence results

- We have strong convergence of the $u^{(n)}$ to a finitely supported limit sequence u^*.
- There exists a finite index set $\Lambda \subset \mathcal{I}$ such that all iterates $u^{(n)}$ and u^* are supported in Λ.
- By the FBI condition, A_Λ is injective and hence $A^*A|_{\Lambda\times\Lambda}$ is boundedly invertible, so that $I - A_\Lambda^* A_\Lambda$ is a contraction on $\ell_2(\Lambda)$.
- Using
$$u_\Lambda^{(n+1)} = \mathbb{S}_\alpha\left(u_\Lambda^{(n)} + A_\Lambda^*(y - A_\Lambda u_\Lambda^{(n)})\right)$$

it follows $\|u^* - u^{(n+1)}\|_{\ell_2(\mathcal{I})} \leq \gamma \|u^* - u^{(n)}\|_{\ell_2(\mathcal{I})}$, where $\gamma = \max\{|1 - \|(A^*A|_{\Lambda\times\Lambda})^{-1}\|^{-1}|, |\|A^*A|_{\Lambda\times\Lambda}\| - 1|\} \in (0,1)$.

We can have
$$\|u^{(n)} - u^*\| \leq C\gamma^n, \quad \gamma < 1,$$
with C arbitrarily large and γ arbitrarily small.

Attempts of faster algorithms

(a) the *GPSR-algorithm* (gradient projection for sparse reconstruction), another iterative projection method, in the auxiliary variables $x, y \geq 0$ with $u = x - y$ (Figueredo, Nowak, and Wright)

Attempts of faster algorithms

(a) the *GPSR-algorithm* (gradient projection for sparse reconstruction), another iterative projection method, in the auxiliary variables $x, y \geq 0$ with $u = x - y$ (Figueredo, Nowak, and Wright)

(b) the $\ell_1 - \ell_s$ *algorithm*, an interior point method using preconditioned conjugate gra dient substeps (this method solves a linear system in each outer iteration step) (S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky)

Attempts of faster algorithms

(a) the *GPSR-algorithm* (gradient projection for sparse reconstruction), another iterative projection method, in the auxiliary variables $x, y \geq 0$ with $u = x - y$ (Figueredo, Nowak, and Wright)

(b) the $\ell_1 - \ell_s$ *algorithm*, an interior point method using preconditioned conjugate gra dient substeps (this method solves a linear system in each outer iteration step) (S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky)

(c) *FISTA* (fast iterative soft-thresholding algorithm) is a variation of the iterative soft-thresholding (Beck and Teboulle). Define the operator $\Gamma(u) = \mathbb{S}_\alpha(u + A^*(y - Au))$. The FISTA is defined as the iteration, starting for $u^{(0)} = 0$,

$$u^{(n+1)} = \Gamma\left(u^{(n)} + \frac{t^{(n)} - 1}{t^{(n+1)}}\left(u^{(n)} - u^{(n-1)}\right)\right),$$

where $t^{(n+1)} = \frac{1+\sqrt{1+4(t^{(n)})^2}}{2}$ and $t^{(0)} = 1$.

Innovation

For several operators K and for certain choices of Ψ, the matrix A^*A can be preconditioned by a matrix $D^{-1/2}$, resulting in the matrix $D^{-1/2}A^*AD^{-1/2}$, in such a way that any restriction $(D^{-1/2}A^*AD^{-1/2})_{\Lambda\times\Lambda}$ turns out to be well-conditioned as soon as $\Lambda \subset \mathcal{I}$ is a small set, but independently of its "location" within $\mathcal{I}$.

Which operators?

Typically we consider (non local) compact operators K

$$Ku(x) = \int_\Omega \kappa(x,\xi)u(\xi)d\xi, \quad x \in \tilde{\Omega},$$

for $\tilde{\Omega}, \Omega \subset \mathbb{R}^d$, $u \in X := H^t(\Omega)$, and

$$|\partial_x^\alpha \partial_\xi^\beta \kappa(x,\xi)| \leq c_{\alpha,\beta}|x-\xi|^{-(d+2t+|\alpha|+|\beta|)}, \quad t \in \mathbb{R}, \quad \alpha, \beta \in \mathbb{N}^d.$$

Which bases/frames?

Moreover, for the proper definition of the discrete matrix $A^*A := F^*K^*KF$, we use multiscale tight (wavelet) frame $\{\psi_\lambda\}_{\lambda\in\mathcal{I}}$ on Ω. We assume:

(i) the index $\lambda = (|\lambda|, k, e)$ encodes several different properties, respectively, the scale $|\lambda|$, the spatial location $k \in \mathbb{R}^d$, and the wavelet label e (without loss of cogency in the following we ignore this latter parameter);

(ii) $\Omega_\lambda := \mathrm{supp}(\psi_\lambda)$, $|\Omega| \sim 2^{-|\lambda|}$; we can assume for simplicity that $\Omega_\lambda \subset 2^{-|\lambda|}k + 2^{-|\lambda|}Q$, where Q is a suitable cube centered at the origin;

(iii) $\int_\Omega \xi^\alpha\, \psi_\lambda(\xi)\, d\xi = 0$, $\alpha = 0, \ldots, d^* \in \mathbb{N}$;

(iv) $\|\psi_\lambda\|_\infty \leq C\, 2^{d/2|\lambda|}$.

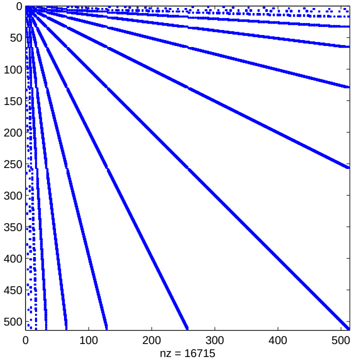
Instructive numerical experiments in an infinite-dimensional setting

Consider the integral operator $K : L_2(0,1) \to L_2(0,1)$,

$$Ku(t) = \int_0^t u(s)\,\mathrm{d}s, \quad K^*Ku(t) = \int_0^1 \big(1 - \max(s,t)\big)\,u(s)\,\mathrm{d}s.$$

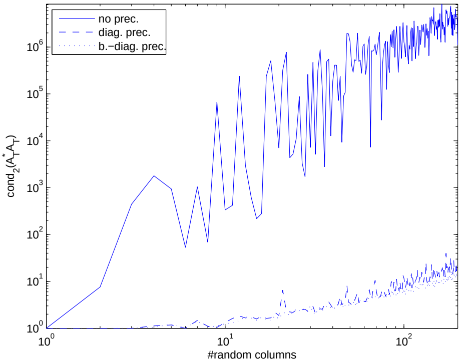
- The integration operator can be regarded as a model case for more general Fredholm-type integral operators;
- K is injective and bounded with norm $\|K\| = 2/\pi \approx 0.64$.
- The nonzero eigenvalues of K^*K are explicitly available as $\lambda_n = 1/(\pi(n+\frac{1}{2}))^2$.

Compressibility of the matrix



Nonzero pattern of the system matrix A^*A, using piecewise linear spline wavelets up to level 8. The discretization of K is performed using a biorthogonal, piecewise linear spline wavelet basis for $L_2(0,1)$ with 2 vanishing moments.

RIP for infinite matrices



Average spectral condition numbers $\mathcal{K}(A^*A|_\Lambda)$ of small $N \times N$-submatrices of A^*A, without preconditioning (*solid line*), with diagonal preconditioning (*dashed line*), and with blockdiagonal preconditioning (*dotted line*).

Adaptive thresholding parameter

For threshold parameters $\alpha, \alpha^{(n)} \in \mathbb{R}_{+}^{\mathcal{I}}$, where $\alpha^{(n)} \geq \alpha$, i.e., $\alpha_{\lambda}^{(n)} \geq \alpha_{\lambda}$ for all $\lambda \in \Lambda$, and $\bar{\alpha} = \inf_{\lambda \in \mathcal{I}} \alpha_{\lambda} > 0$, we consider the iteration

$$u^{(0)} = 0, \quad u^{(n+1)} = \mathbb{S}_{\alpha^{(n)}}\big(u^{(n)} + A^{*}(y - Au^{(n)})\big), \quad n = 0, 1, \ldots$$

which we called the *decreasing iterative soft-thresholding algorithm* (D-ISTA).

Prescribed linear convergence

Theorem (Dahlke, Fornasier, and Raasch '09)

Let $\|A\| < \sqrt{2}$ *and let* $\bar{u} := (I - A^*A)u^* + A^*y \in \ell_\alpha^w(\mathcal{I})$ *for some* $0 < \alpha < 2$*. Moreover, let* $L = L(\alpha) := \frac{4\|u^*\|^2_{\ell_2(\mathcal{I})}}{\bar{\alpha}^2} + 4C\|\bar{u}\|^\alpha_{\ell_\alpha^w(\mathcal{I})}\bar{\alpha}^{-\alpha}$, *and assume that for* $S^* := \mathrm{supp}(u^*)$ *and all finite subsets* $\Lambda \subset \mathcal{I}$ *with at most* $\#\Lambda \leq 2L$ *elements,*

$$\|(I - A^*A)|_{S^*\cup\Lambda \times S^*\cup\Lambda}\| \leq \gamma_0,$$

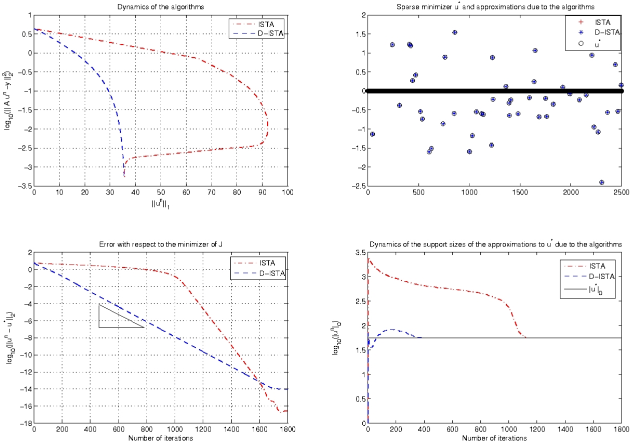
where $0 < \gamma_0 < 1$*. Then, for any* $\gamma_0 < \gamma < 1$*, the iterates* $u^{(n)}$ *fulfill* $\#\,\mathrm{supp}\,u^{(n)} \leq L$ *and they converge to* u^* *at a linear rate*

$$\|u^* - u^{(n)}\|_{\ell_2(\mathcal{I})} \leq \gamma^n \|u^*\|_{\ell_2(\mathcal{I})} =: \epsilon_n$$

whenever the $\alpha^{(n)}$ *are chosen according to*

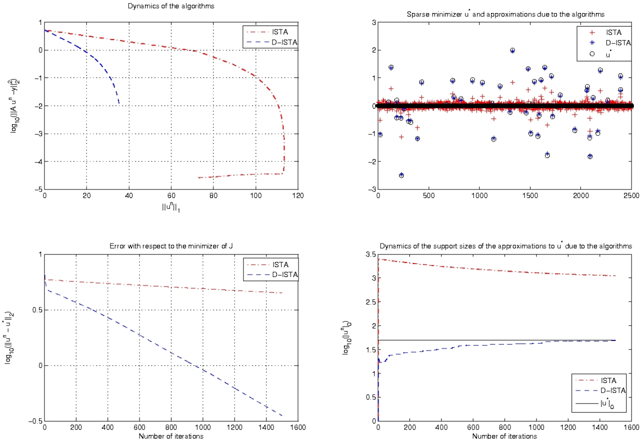
$$\alpha_\lambda \leq \alpha_\lambda^{(n)} \leq \alpha_\lambda + (\gamma - \gamma_0)L^{-1/2}\epsilon_n, \text{ for all } \lambda \in \Lambda.$$

Gaussian matrices and rates of convergence



Computation of a sparse minimizer u^* for A being a 500×2500 matrix with i.i.d. Gaussian entries, $\alpha = 10^{-3}$, $\gamma_0 = 0.1$ and $\gamma = 0.95$.

Gaussian matrices and rates of convergence



Computation of a sparse minimizer u^* for A being a 500×2500 matrix with i.i.d. Gaussian entries, $\alpha = 10^{-4}$, $\gamma_0 = 0.01$ and $\gamma = 0.998$.

A domain decomposition method for large scale computing

- We consider the minimization of $\mathcal{J} = \mathcal{J}_\alpha$, by alternating subspace corrections.

A domain decomposition method for large scale computing

- We consider the minimization of $\mathcal{J} = \mathcal{J}_\alpha$, by alternating subspace corrections.
- We start by decomposing the "domain" of the sequences $\mathcal{I}$ into two disjoint sets Λ_1, Λ_2 so that $\mathcal{I} = \Lambda_1 \cup \Lambda_2$.

A domain decomposition method for large scale computing

- We consider the minimization of $\mathcal{J} = \mathcal{J}_\alpha$, by alternating subspace corrections.
- We start by decomposing the "domain" of the sequences $\mathcal{I}$ into two disjoint sets Λ_1, Λ_2 so that $\mathcal{I} = \Lambda_1 \cup \Lambda_2$.
- Associated to a decomposition $\mathcal{C} = \{\Lambda_1, \Lambda_2\}$ we define the *extension operators* $E_i : \ell_2(\Lambda_i) \to \ell_2(\mathcal{I})$, $(E_i v)_\lambda = v_\lambda$, if $\lambda \in \Lambda_i$, $(E_i v)_\lambda = 0$, otherwise, $i = 1, 2$.

A domain decomposition method for large scale computing

- We consider the minimization of $\mathcal{J} = \mathcal{J}_\alpha$, by alternating subspace corrections.
- We start by decomposing the "domain" of the sequences $\mathcal{I}$ into two disjoint sets Λ_1, Λ_2 so that $\mathcal{I} = \Lambda_1 \cup \Lambda_2$.
- Associated to a decomposition $\mathcal{C} = \{\Lambda_1, \Lambda_2\}$ we define the *extension operators* $E_i : \ell_2(\Lambda_i) \to \ell_2(\mathcal{I})$, $(E_i v)_\lambda = v_\lambda$, if $\lambda \in \Lambda_i$, $(E_i v)_\lambda = 0$, otherwise, $i = 1, 2$. The adjoint operator, which we call the *restriction operator*, is denoted by $R_i := E_i^*$.

With these operators we define the functional $\mathcal{J}(x_1, x_2)$, $\mathcal{J} : \ell_2(\Lambda_1) \times \ell_2(\Lambda_2) \to \mathbb{R}$, given by

$$\mathcal{J}(x_1, x_2) := \mathcal{J}(E_1 x_1 + E_2 x_2).$$

With these operators we define the functional $\mathcal{J}(x_1, x_2)$, $\mathcal{J} : \ell_2(\Lambda_1) \times \ell_2(\Lambda_2) \to \mathbb{R}$, given by

$$\mathcal{J}(x_1, x_2) := \mathcal{J}(E_1 x_1 + E_2 x_2).$$

In analogy to the Schwartz multiplicative algorithm in domain decoposition in numerics for PDEs, we analyze the following algorithm:

$$\begin{cases} x_1^{(n+1)} = \operatorname{argmin}_{v_1 \in \ell_2(\Lambda_1)} \mathcal{J}(v_1, x_2^{(n)}) \\ x_2^{(n+1)} = \operatorname{argmin}_{v_2 \in \ell_2(\Lambda_2)} \mathcal{J}(x_1^{(n+1)}, v_2) \\ x^{(n+1)} := E_1 x_1^{(n+1)} + E_2 x_2^{(n+1)}. \end{cases}$$

Let us observe that
$\|E_1 x_1 + E_2 x_2\|_{\ell_1(\Lambda)} := \|x_1\|_{\ell_1(\Lambda_1)} + \|x_2\|_{\ell_1(\Lambda_2)}$, hence

$$\begin{aligned} & \operatorname{argmin}_{v_1 \in \ell_2(\Lambda_1)} \mathcal{J}(v_1, x_2^{(n)}) \\ = \quad & \operatorname{argmin}_{v_1 \in \ell_2(\Lambda_1)} \|(y - AE_2 x_2^{(n)}) - AE_1 v_1\|_2^2 + \alpha \|v_1\|_1. \end{aligned}$$

Let us observe that
$\|E_1 x_1 + E_2 x_2\|_{\ell_1(\Lambda)} := \|x_1\|_{\ell_1(\Lambda_1)} + \|x_2\|_{\ell_1(\Lambda_2)}$, hence

$$\begin{aligned} & \operatorname{argmin}_{v_1 \in \ell_2(\Lambda_1)} \mathcal{J}(v_1, x_2^{(n)}) \\ = \; & \operatorname{argmin}_{v_1 \in \ell_2(\Lambda_1)} \|(y - AE_2 x_2^{(n)}) - AE_1 v_1\|_2^2 + \alpha \|v_1\|_1. \end{aligned}$$

A similar formulation holds for $\operatorname{argmin}_{v_2 \in \ell_2(\Lambda_2)} \mathcal{J}(x_1^{(n+1)}, v_2)$.

Let us observe that $\|E_1 x_1 + E_2 x_2\|_{\ell_1(\Lambda)} := \|x_1\|_{\ell_1(\Lambda_1)} + \|x_2\|_{\ell_1(\Lambda_2)}$, hence

$$\begin{aligned} & \operatorname{argmin}_{v_1 \in \ell_2(\Lambda_1)} \mathcal{J}(v_1, x_2^{(n)}) \\ = \; & \operatorname{argmin}_{v_1 \in \ell_2(\Lambda_1)} \|(y - AE_2 x_2^{(n)}) - AE_1 v_1\|_2^2 + \alpha \|v_1\|_1. \end{aligned}$$

A similar formulation holds for $\operatorname{argmin}_{v_2 \in \ell_2(\Lambda_2)} \mathcal{J}(x_1^{(n+1)}, v_2)$. This means that the solution of the local problems on Λ_i is of the *same* kind as the original problem $\operatorname{argmin}_{x \in \ell_2(\Lambda)} \mathcal{J}(x)$, but the dimension for each has been reduced.

This leads to the following sequential algorithm

Algorithm 5.

$$\begin{cases} \begin{cases} x_1^{(n+1,0)} = x_1^{(n,L)} \\ x_1^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(x_1^{(n+1,\ell)} + R_1 A^* ((y - AE_2 x_2^{(n,M)}) - AE_1 x_1^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, L-1 \end{cases} \\ \begin{cases} x_2^{(n+1,0)} = x_2^{(n,M)} \\ x_2^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(x_2^{(n+1,\ell)} + R_2 A^* ((y - AE_1 x_1^{(n+1,L)}) - AE_2 x_2^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, M-1 \end{cases} \\ x^{(n+1)} := E_1 x_1^{(n+1,L)} + E_2 x_2^{(n+1,M)}. \end{cases}$$

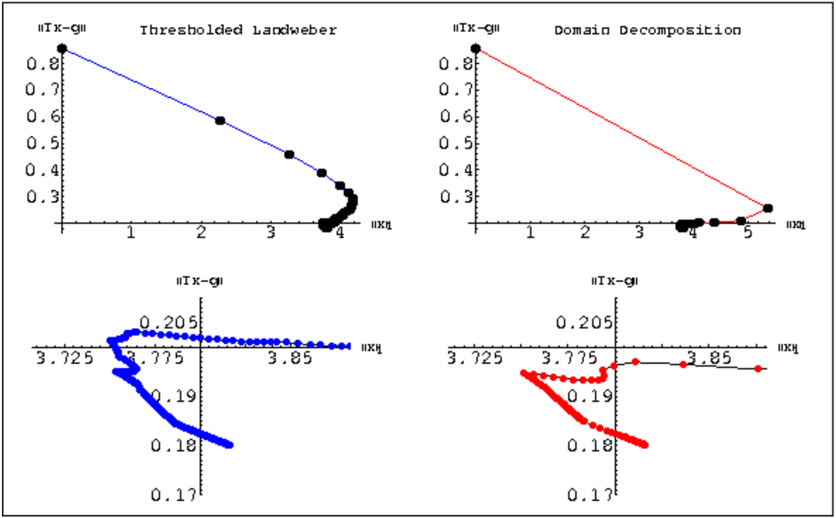
This leads to the following parallel algorithm

Algorithm 6.

$$\begin{cases} \begin{cases} x_1^{(n+1,0)} = x_1^{(n,L)} \\ x_1^{(n+1,\ell+1)} = \mathbb{S}_\alpha\left(x_1^{(n+1,\ell)} + R_1 A^*((y - AE_2 R_2 x^{(n)}) - AE_1 x_1^{(n+1,\ell)})\right) \\ \ell = 0, \dots, L-1 \end{cases} \\ \begin{cases} x_2^{(n+1,0)} = x_2^{(n,M)} \\ x_2^{(n+1,\ell+1)} = \mathbb{S}_\alpha\left(x_2^{(n+1,\ell)} + R_2 A^*((y - AE_1 R_1 x^{(n)}) - AE_2 x_2^{(n+1,\ell)})\right) \\ \ell = 0, \dots, M-1 \end{cases} \\ x^{(n+1)} := \frac{E_1 x_1^{(n+1,L)} + E_2 x_2^{(n+1,M)} + x^{(n)}}{2}. \end{cases}$$

Theorem

The (sequential and parallel) subspace correction algorithms produce a sequence $(x^{(n)})_{n\in\mathbb{N}}$ *in* $\ell_2(\mathcal{I})$ *whose strong accumulation points are minimizers of the functional* $\mathcal{J}$*. In particular, the set of strong accumulation points is non-empty. If the minimizer is unique then the whole sequence* $(x^{(n)})_{n\in\mathbb{N}}$ *converges to it.*



||Tx-g||
Thresholded Landweber
0.8
0.7
0.6
0.5
0.4
0.3
1
2
3
4
||x||
||Tx-g||
Domain Decomposition
0.8
0.7
0.6
0.5
0.4
0.3
1
2
3
4
5
||x||
||Tx-g||
0.205
3.725
3.775
3.85
||x||
0.19
0.18
0.17
||Tx-g||
0.205
3.725
3.775
3.85
||x||
0.19
0.18
0.17

A few references

- *A Mathematical Introduction to Compressive Sensing* (Holger Rauhut and Simon Foucart), Birkhäuser-Springer, 2013.
- *Numerical methods for sparse recovery* book chapter in "Theoretical Foundations and Numerical Methods for Sparse Recovery", M. Fornasier (ed.) Radon Series in Applied and Computational Mathematics 9, de Gruyter, 2010
- *Compressive Sensing* (Massimo Fornasier and Holger Rauhut), book chapter in "Handbook of Mathematical Methods in Imaging" Springer.
- *An Overview on Algorithms for Sparse Recovery* (Massimo Fornasier and Steffen Peter) book chapter in "Sparse Reconstruction and Compressive Sensing in Remote Sensing", X. Zhu and R. Bamler (ed.), Springer, 2015.

Publications:
`https://www-m15.ma.tum.de/Allgemeines/PublicationsEN`
Software:
`https://www-m15.ma.tum.de/Allgemeines/SoftwareSite`