Decoding in Compressed Sensing

Ronald DeVore

Discrete Compressed Sensing

- $x \in \mathbb{R}^N$ with N large

Discrete Compressed Sensing

- $x \in \mathbb{R}^N$ with N large
- We are able to ask n questions about x

Discrete Compressed Sensing

- $x \in \mathbb{R}^N$ with N large
- We are able to ask n questions about x
- Question means inner product $v \cdot x$ with $v \in \mathbb{R}^N$ - called sample

Discrete Compressed Sensing

- $x \in \mathbb{R}^N$ with N large
- We are able to ask n questions about x
- Question means inner product $v \cdot x$ with $v \in \mathbb{R}^N$ - called sample
- What are the best questions to ask??

Discrete Compressed Sensing

- $x \in \mathbb{R}^N$ with N large
- We are able to ask n questions about x
- Question means inner product $v \cdot x$ with $v \in \mathbb{R}^N$ - called sample
- What are the best questions to ask??
- Any such sampling is given by Φx where Φ is an $n \times N$ matrix

Discrete Compressed Sensing

- $x \in \mathbb{R}^N$ with N large
- We are able to ask n questions about x
- Question means inner product $v \cdot x$ with $v \in \mathbb{R}^N$ - called sample
- What are the best questions to ask??
- Any such sampling is given by Φx where Φ is an $n \times N$ matrix
- We are interested in the good / best matrices Φ

Discrete Compressed Sensing

- $x \in \mathbb{R}^N$ with N large
- We are able to ask n questions about x
- Question means inner product $v \cdot x$ with $v \in \mathbb{R}^N$ - called sample
- What are the best questions to ask??
- Any such sampling is given by Φx where Φ is an $n \times N$ matrix
- We are interested in the good / best matrices Φ
- Here good means the samples $y = \Phi x$ contain enough information to approximate x well

Encoder/Decoder

- We view Φ as an encoder

Encoder/Decoder

- We view Φ as an encoder
- Since $\Phi : \mathbb{R}^N \to \mathbb{R}^n$ many x are encoded with same y

Encoder/Decoder

- We view Φ as an encoder
- Since $\Phi : I\!R^N \rightarrow I\!R^n$ many x are encoded with same y
- $\mathcal{N} := \{\eta : \Phi\eta = 0\}$ the null space of Φ

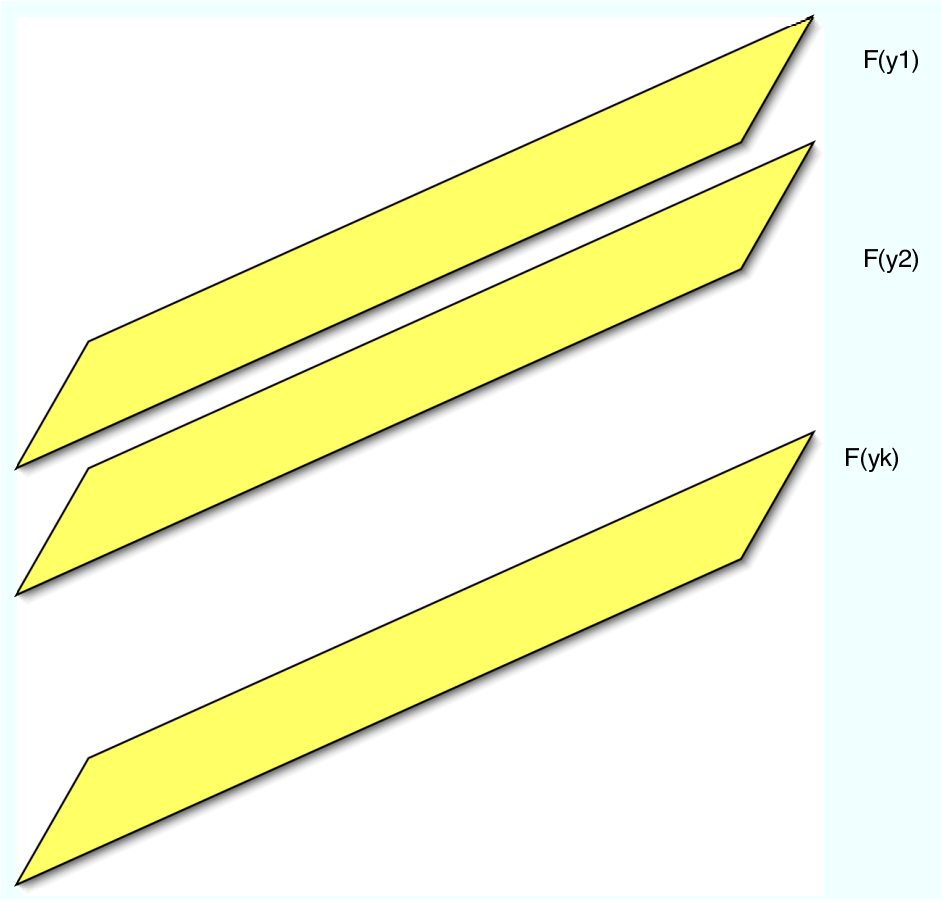
Encoder/Decoder

- We view Φ as an encoder
- Since $\Phi : I\!R^N \to I\!R^n$ many x are encoded with same y
- $\mathcal{N} := \{\eta : \Phi\eta = 0\}$ the null space of Φ
- $\mathcal{F}(y) := \{x : \Phi x = y\} = x_0 + \mathcal{N}$ for any $x_0 \in \mathcal{F}(y)$

Encoder/Decoder

- We view Φ as an encoder
- Since $\Phi : I\!R^N \to I\!R^n$ many x are encoded with same y
- $\mathcal{N} := \{\eta : \Phi\eta = 0\}$ the null space of Φ
- $\mathcal{F}(y) := \{x : \Phi x = y\} = x_0 + \mathcal{N}$ for any $x_0 \in \mathcal{F}(y)$
- The hyperplanes $\mathcal{F}(y)$ with $y \in I\!R^n$ stratify $I\!R^N$

The sets $\mathcal{F}(y)$



Encoder/Decoder

- We view Φ as an encoder
- Since $\Phi : I\!R^N \to I\!R^n$ many x are encoded with same y
- $\mathcal{N} := \{\eta : \Phi\eta = 0\}$ the null space of Φ
- $\mathcal{F}(y) := \{x : \Phi x = y\} = x_0 + \mathcal{N}$ for any $x_0 \in \mathcal{F}(y)$
- The hyperplanes $\mathcal{F}(y)$ with $y \in I\!R^n$ stratify $I\!R^N$
- Decoder is any (possibly nonlinear) mapping Δ from $I\!R^n \to I\!R^N$

Encoder/Decoder

- We view Φ as an encoder
- Since $\Phi : I\!R^N \to I\!R^n$ many x are encoded with same y
- $\mathcal{N} := \{\eta : \Phi\eta = 0\}$ the null space of Φ
- $\mathcal{F}(y) := \{x : \Phi x = y\} = x_0 + \mathcal{N}$ for any $x_0 \in \mathcal{F}(y)$
- The hyperplanes $\mathcal{F}(y)$ with $y \in I\!R^n$ stratify $I\!R^N$
- Decoder is any (possibly nonlinear) mapping Δ from $I\!R^n \to I\!R^N$
- $\bar{x} := \Delta(\Phi(x))$ is our approximation to x from the information extracted

Encoder/Decoder

- We view Φ as an encoder
- Since $\Phi : I\!R^N \to I\!R^n$ many x are encoded with same y
- $\mathcal{N} := \{\eta : \Phi\eta = 0\}$ the null space of Φ
- $\mathcal{F}(y) := \{x : \Phi x = y\} = x_0 + \mathcal{N}$ for any $x_0 \in \mathcal{F}(y)$
- The hyperplanes $\mathcal{F}(y)$ with $y \in I\!R^n$ stratify $I\!R^N$
- Decoder is any (possibly nonlinear) mapping Δ from $I\!R^n \to I\!R^N$
- $\bar{x} := \Delta(\Phi(x))$ is our approximation to x from the information extracted
- Let $\mathcal{A} := \mathcal{A}_{n,N} := \{(\Phi, \Delta) : \ \Phi \ \text{ is } \ n \times N\}$

GOAL: Instance-Optimal

- How should we measure performance of the encoding-decoding?

GOAL: Instance-Optimal

- How should we measure performance of the encoding-decoding?
- Define $\Sigma_k := \{x \in I\!R^N : \#\mathrm{supp(x)} \leq \mathrm{k}\}$

$$\sigma_k(x)_{\ell_p} := \inf_{z \in \Sigma_k} \|x - z\|_{\ell_p^N}$$

GOAL: Instance-Optimal

- How should we measure performance of the encoding-decoding?
- Define $\Sigma_k := \{x \in I\!R^N : \#\mathrm{supp(x)} \leq \mathrm{k}\}$

$$\sigma_k(x)_{\ell_p} := \inf_{z \in \Sigma_k} \|x - z\|_{\ell_p^N}$$

- Given an encoding - decoding pair (Φ, Δ), we say that this pair is Instance-Optimal of order k for ℓ_p if for an absolute constant $C > 0$

$$\|x - \Delta(\Phi(x))\|_{\ell_p} \leq C\sigma_k(x)_{\ell_p}$$

GOAL: Instance-Optimal

- How should we measure performance of the encoding-decoding?
- Define $\Sigma_k := \{x \in I\!R^N : \#\mathrm{supp(x)} \leq \mathrm{k}\}$

$$\sigma_k(x)_{\ell_p} := \inf_{z \in \Sigma_k} \|x - z\|_{\ell_p^N}$$

- Given an encoding - decoding pair (Φ, Δ), we say that this pair is Instance-Optimal of order k for ℓ_p if for an absolute constant $C > 0$

$$\|x - \Delta(\Phi(x))\|_{\ell_p} \leq C\sigma_k(x)_{\ell_p}$$

- The encoding - decoding pairs which have the largest k are the best.

What are optimal systems

- Two issues: Good matrices and good decoders

What are optimal systems

- Two issues: Good matrices and good decoders
- We shall first discuss what are good matrices: everything is controlled by the null space

What are optimal systems

- Two issues: Good matrices and good decoders
- We shall first discuss what are good matrices: everything is controlled by the null space
- Theorem of Cohen-Dahmen-DeVore: We have instance optimality of order k in ℓ_p for some decoder Δ if and only if Φ has the Null Space Property (NSP) of order $2k$: for each $\#(T) \leq 2k$ we have

$$\|\eta_T\|_{\ell_p} \leq C\|\eta_{T^c}\|$$

What are optimal systems

- Two issues: Good matrices and good decoders
- We shall first discuss what are good matrices: everything is controlled by the null space
- Theorem of Cohen-Dahmen-DeVore: We have instance optimality of order k in ℓ_p for some decoder Δ if and only if Φ has the Null Space Property (NSP) of order $2k$: for each $\#(T) \leq 2k$ we have

$$\|\eta_T\|_{\ell_p} \leq C\|\eta_{T^c}\|$$

- Notation: x_T is the vector that agrees with x on T and is zero otherwise.

What are optimal systems

- Two issues: Good matrices and good decoders
- We shall first discuss what are good matrices: everything is controlled by the null space
- Theorem of Cohen-Dahmen-DeVore: We have instance optimality of order k in ℓ_p for some decoder Δ if and only if Φ has the Null Space Property (NSP) of order $2k$: for each $\#(T) \leq 2k$ we have

$$\|\eta_T\|_{\ell_p} \leq C\|\eta_{T^c}\|$$

- Notation: x_T is the vector that agrees with x on T and is zero otherwise.
- Here the decoder is pushed into the background: The above theorem gives no information about whether practical decoders will work

Restricted Isometry Property

- How to check NSP?

Restricted Isometry Property

- How to check NSP?
- A sufficient condition is the Restricted Isometry Property

Restricted Isometry Property

- How to check NSP?
- A sufficient condition is the Restricted Isometry Property
- We say Φ has the Restricted Isometry Property (RIP)of order k with constant $\delta \in (0, 1)$ if

$$(1 = \delta)\|x\|_{\ell_2} \leq \|\Phi x\|_{\ell_2} \leq (1 + \delta)\|x\|_{\ell_2} \quad x \in \Sigma_k$$

Restricted Isometry Property

- How to check NSP?
- A sufficient condition is the Restricted Isometry Property
- We say Φ has the Restricted Isometry Property (RIP)of order k with constant $\delta \in (0, 1)$ if

$$(1 = \delta)\|x\|_{\ell_2} \leq \|\Phi x\|_{\ell_2} \leq (1 + \delta)\|x\|_{\ell_2} \quad x \in \Sigma_k$$

- CDD show RIP is sufficient to guarantee NSP but the exact results depend very strongly on p

Sample Results: ℓ_1

- If Φ has RIP for $2k$ then Φ is instance-optimal of order k in ℓ_1:

$$\|x - \Delta(\Phi(x))\|_{\ell_1} \leq C\sigma_k(x)_{\ell_1}$$

Sample Results: ℓ_1

- If Φ has RIP for $2k$ then Φ is instance-optimal of order k in ℓ_1:

$$\|x - \Delta(\Phi(x))\|_{\ell_1} \leq C\sigma_k(x)_{\ell_1}$$

- This means that there is some decoder - not practical

Sample Results: ℓ_1

- If Φ has RIP for $2k$ then Φ is instance-optimal of order k in ℓ_1:

$$\|x - \Delta(\Phi(x))\|_{\ell_1} \leq C\sigma_k(x)_{\ell_1}$$

- This means that there is some decoder - not practical
- We know there are matrices Φ with RIP property for $k \leq c_0 n/\log(N/n)$

Sample Results: ℓ_1

- If Φ has RIP for $2k$ then Φ is instance-optimal of order k in ℓ_1:

$$\|x - \Delta(\Phi(x))\|_{\ell_1} \leq C\sigma_k(x)_{\ell_1}$$

- This means that there is some decoder - not practical
- We know there are matrices Φ with RIP property for $k \leq c_0 n/\log(N/n)$
- This range of k cannot be improved: from now on we refer to this as the largest range of k

Sample Results: ℓ_p

- If an $n \times N$ matrix Φ is instance-optimal of order $k = 1$ in ℓ_2 with constant C_0 then $n \geq N/C_0$

Sample Results: ℓ_p

- If an $n \times N$ matrix Φ is instance-optimal of order $k = 1$ in ℓ_2 with constant C_0 then $n \geq N/C_0$
- This shows that instance-optimal is not a viable concept for ℓ_2

Sample Results: ℓ_p

- If an $n \times N$ matrix Φ is instance-optimal of order $k = 1$ in ℓ_2 with constant C_0 then $n \geq N/C_0$
- This shows that instance-optimal is not a viable concept for ℓ_2
- For $1 < p < 2$ and any $k \leq c_0 N^{\frac{2-2/p}{1-2/p}} [n/\log(N/n)]^{\frac{p}{2-p}}$

Sample Results: ℓ_p

- If an $n \times N$ matrix Φ is instance-optimal of order $k = 1$ in ℓ_2 with constant C_0 then $n \geq N/C_0$
- This shows that instance-optimal is not a viable concept for ℓ_2
- For $1 < p < 2$ and any $k \leq c_0 N^{\frac{2-2/p}{1-2/p}} [n/\log(N/n)]^{\frac{p}{2-p}}$
- This bound cannot be improved

Sample Results: ℓ_p

- If an $n \times N$ matrix Φ is instance-optimal of order $k = 1$ in ℓ_2 with constant C_0 then $n \geq N/C_0$
- This shows that instance-optimal is not a viable concept for ℓ_2
- For $1 < p < 2$ and any $k \leq c_0 N^{\frac{2-2/p}{1-2/p}} [n/\log(N/n)]^{\frac{p}{2-p}}$
- This bound cannot be improved
- Matrices that satisfy instance-optimal for this range of k are obtained from matrices which satisfy RIP for $\bar{k} = k^{2/p-1} N^{2-2/p}$

What are good matrices

- How can we construct Φ satisfying RIP for the largest range of k?

What are good matrices

- How can we construct Φ satisfying RIP for the largest range of k?
- Choose at random N vectors from the unit sphere in $\mathbb{R}^n$ and use these as the columns of Φ

What are good matrices

- How can we construct Φ satisfying RIP for the largest range of k?
- Choose at random N vectors from the unit sphere in $\mathbb{R}^n$ and use these as the columns of Φ
- Choose each entry of Φ independently and at random from the Gaussian distribution $\mathcal{N}(0, 1/\sqrt{n})$

What are good matrices

- How can we construct Φ satisfying RIP for the largest range of k?
- Choose at random N vectors from the unit sphere in $\mathbb{R}^n$ and use these as the columns of Φ
- Choose each entry of Φ independently and at random from the Gaussian distribution $\mathcal{N}(0, 1/\sqrt{n})$
- We choose each entry of Φ independently and at random from the Bernouli distribution and then normalize columns to have length one.

What are good matrices

- How can we construct Φ satisfying RIP for the largest range of k?
- Choose at random N vectors from the unit sphere in $\mathbb{R}^n$ and use these as the columns of Φ
- Choose each entry of Φ independently and at random from the Gaussian distribution $\mathcal{N}(0, 1/\sqrt{n})$
- We choose each entry of Φ independently and at random from the Bernouli distribution and then normalize columns to have length one.
- With high probability on the draw the resulting matrix will have RIP for the largest range of k

What are good matrices

- How can we construct Φ satisfying RIP for the largest range of k?
- Choose at random N vectors from the unit sphere in $\mathbb{R}^n$ and use these as the columns of Φ
- Choose each entry of Φ independently and at random from the Gaussian distribution $\mathcal{N}(0, 1/\sqrt{n})$
- We choose each entry of Φ independently and at random from the Bernouli distribution and then normalize columns to have length one.
- With high probability on the draw the resulting matrix will have RIP for the largest range of k
- Problem: None of these are constructive. Can we put our hands on matrices??

What are good matrices

- How can we construct Φ satisfying RIP for the largest range of k?
- Choose at random N vectors from the unit sphere in $\mathbb{R}^n$ and use these as the columns of Φ
- Choose each entry of Φ independently and at random from the Gaussian distribution $\mathcal{N}(0, 1/\sqrt{n})$
- We choose each entry of Φ independently and at random from the Bernouli distribution and then normalize columns to have length one.
- With high probability on the draw the resulting matrix will have RIP for the largest range of k
- Problem: None of these are constructive. Can we put our hands on matrices??
- No constructions are known for largest range of k

Instance-Optimality in Probability

- We saw that Instance-Optimality for ℓ_2^N is not viable
- Suppose $\Phi(\omega)$ is a collection of random matrices
- We say this family satisfies RIP of order k with probability $1-\epsilon$ if a random draw $\{\Phi(\omega)\}$ will satisfy RIP of order k with probability $1-\epsilon$
- We say $\{\Phi(\omega)\}$ is bounded with probability $1-\epsilon$ if given any $x \in I\!R^N$ with probability $1-\epsilon$ a random draw $\{\Phi(\omega)\}$ will satisfy

$$\|\Phi(\omega)(x)\|_{\ell_2^N} \leq C_0 \|x\|_{\ell_2^N}$$

  with C_0 an absolute constant
- Our earlier analysis showed that Gaussian and Bernouli random matrices have these properties with $\epsilon = e^{-cn}$

Theorem: Cohen-Dahmen-DeVore

- If $\{\Phi(\omega)\}$ satisfies RIP of order $3k$ and boundedness each with probability $1-\epsilon$ then there are decoders $\Delta(\omega)$ such that given any $x \in \ell_2^N$ we have with probability $1-2\epsilon$

$$\|x - \Delta(\omega)\Phi(\omega)(x)\|_{\ell_2^N} \leq C_0 \sigma_k(x)_{\ell_2^N}$$

- Instance-optimality in probability
- Range of k is $k \leq c_0 n/\log(N/n)$
- Decoder is impractical

Decoding

- By far the most intriguing part of Compressed Sensing is the decoding

Decoding

- By far the most intriguing part of Compressed Sensing is the decoding
- There are continuing debates as to which decoding is numerically fastest

Decoding

- By far the most intriguing part of Compressed Sensing is the decoding
- There are continuing debates as to which decoding is numerically fastest
- Some common decoders

Decoding

- By far the most intriguing part of Compressed Sensing is the decoding
- There are continuing debates as to which decoding is numerically fastest
- Some common decoders
  - ℓ_1 minimization: Long history (Donoho; Candes-Romberg)

Decoding

- By far the most intriguing part of Compressed Sensing is the decoding
- There are continuing debates as to which decoding is numerically fastest
- Some common decoders
  - ℓ_1 minimization: Long history (Donoho; Candes-Romberg)
  - Greedy algorithms - find support of a good approximation vector and then decode using ℓ_2 minimization (Gilbert-Tropp; Needel-Vershynin(ROMP), Donoho (STOMP))

Decoding

- By far the most intriguing part of Compressed Sensing is the decoding
- There are continuing debates as to which decoding is numerically fastest
- Some common decoders
  - ℓ_1 minimization: Long history (Donoho; Candes-Romberg)
  - Greedy algorithms - find support of a good approximation vector and then decode using ℓ_2 minimization (Gilbert-Tropp; Needel-Vershynin(ROMP), Donoho (STOMP))
  - Iterative Reweighted Least Squares (Osborne, Daubechies-DeVore-Fornasier-Gunturk)

Decoding

- By far the most intriguing part of Compressed Sensing is the decoding
- There are continuing debates as to which decoding is numerically fastest
- Some common decoders
  - ℓ_1 minimization: Long history (Donoho; Candes-Romberg)
  - Greedy algorithms - find support of a good approximation vector and then decode using ℓ_2 minimization (Gilbert-Tropp; Needel-Vershynin(ROMP), Donoho (STOMP))
  - Iterative Reweighted Least Squares (Osborne, Daubechies-DeVore-Fornasier-Gunturk)
- We shall make some remarks on these decoders emphasizing the last one

Issues in Decoding

- Range of Instance Optimality: When combined with encoder does it give full range of instance optimality?

Issues in Decoding

- Range of Instance Optimality: When combined with encoder does it give full range of instance optimality?
- Number of computations to decode?

Issues in Decoding

- Range of Instance Optimality: When combined with encoder does it give full range of instance optimality?
- Number of computations to decode?
- Robustness to noise?

Issues in Decoding

- Range of Instance Optimality: When combined with encoder does it give full range of instance optimality?
- Number of computations to decode?
- Robustness to noise?
- Theorems versus numerical examples

Issues in Decoding

- Range of Instance Optimality: When combined with encoder does it give full range of instance optimality?
- Number of computations to decode?
- Robustness to noise?
- Theorems versus numerical examples
- Instance Optimality in Probability

Issues in Decoding

- Range of Instance Optimality: When combined with encoder does it give full range of instance optimality?
- Number of computations to decode?
- Robustness to noise?
- Theorems versus numerical examples
- Instance Optimality in Probability
- Given that we cant construct best encoding matrices it seems that the best results would correspond to random draws of matrices

ℓ_1 minimization

- ℓ_1 minimization: $x^* := \underset{z \in \mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_1}$

ℓ_1 minimization

- ℓ_1 minimization: $x^* := \underset{z \in \mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_1}$
- $x^* = x - \eta^*$ where $\eta^* := \underset{\eta \in \mathcal{N}}{\operatorname{Argmin}} \|x - \eta\|_{\ell_1}$

ℓ_1 minimization

- ℓ_1 minimization: $x^* := \underset{z \in \mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_1}$
- $x^* = x - \eta^*$ where $\eta^* := \underset{\eta \in \mathcal{N}}{\operatorname{Argmin}} \|x - \eta\|_{\ell_1}$
- Can be solved by Linear Programming

ℓ_1 minimization

- ℓ_1 minimization: $x^* := \underset{z \in \mathcal{F}(y)}{\text{Argmin}} \|z\|_{\ell_1}$
- $x^* = x - \eta^*$ where $\eta^* := \underset{\eta \in \mathcal{N}}{\text{Argmin}} \|x - \eta\|_{\ell_1}$
- Can be solved by Linear Programming
- Let $T := \text{supp(x)}$

ℓ_1 minimization

- ℓ_1 minimization: $x^* := \underset{z \in \mathcal{F}(y)}{\text{Argmin}} \|z\|_{\ell_1}$
- $x^* = x - \eta^*$ where $\eta^* := \underset{\eta \in \mathcal{N}}{\text{Argmin}} \|x - \eta\|_{\ell_1}$
- Can be solved by Linear Programming
- Let $T := \text{supp(x)}$
- Recall that x is an ℓ_1 minimizer if and only if

$$|\sum_{i \in T} \text{sign}(x_i)\eta_i| \leq \sum_{i \in T^c} |\eta_i|, \quad \forall \eta \in \mathcal{N}$$

ℓ_1 minimization

- ℓ_1 minimization: $x^* := \underset{z \in \mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_1}$
- $x^* = x - \eta^*$ where $\eta^* := \underset{\eta \in \mathcal{N}}{\operatorname{Argmin}} \|x - \eta\|_{\ell_1}$
- Can be solved by Linear Programming
- Let $T := \operatorname{supp}(\mathrm{x})$
- Recall that x is an ℓ_1 minimizer if and only if

$$|\sum_{i \in T} \operatorname{sign}(x_i)\eta_i| \leq \sum_{i \in T^c} |\eta_i|, \quad \forall \eta \in \mathcal{N}$$

- If $\mathcal{N}$ has the following null space property: there is a $\gamma < 1$ with $\|\eta_T\|_{\ell_1} \leq \gamma \|\eta_{T^c}\|_{\ell_1}, \quad \forall \eta \in \mathcal{N},\ \#(T) \leq k$

ℓ_1 minimization

- ℓ_1 minimization: $x^* := \underset{z \in \mathcal{F}(y)}{\text{Argmin}} \|z\|_{\ell_1}$
- $x^* = x - \eta^*$ where $\eta^* := \underset{\eta \in \mathcal{N}}{\text{Argmin}} \|x - \eta\|_{\ell_1}$
- Can be solved by Linear Programming
- Let $T := \text{supp(x)}$
- Recall that x is an ℓ_1 minimizer if and only if

$$|\sum_{i \in T} \text{sign}(x_i)\eta_i| \leq \sum_{i \in T^c} |\eta_i|, \quad \forall \eta \in \mathcal{N}$$

- If $\mathcal{N}$ has the following null space property: there is a $\gamma < 1$ with $\|\eta_T\|_{\ell_1} \leq \gamma \|\eta_{T^c}\|_{\ell_1}, \quad \forall \eta \in \mathcal{N},\ \#(T) \leq k$
- then all $x \in \Sigma_k$ have unique ℓ_1 minimizers equal to x

Orhtogonal Matching Pursuit (OMP)

- We seek approximations to y from the dictionary $\mathcal{D} := \{\phi_1, \ldots, \phi_N\}$ consisting of the columns of Φ

Orhtogonal Matching Pursuit (OMP)

- We seek approximations to y from the dictionary $\mathcal{D} := \{\phi_1, \ldots, \phi_N\}$ consisting of the columns of Φ
- $j_1 := \underset{j=1,\cdots,N}{\text{Argmax}} \, |\langle y, \phi_j \rangle|$

Orhtogonal Matching Pursuit (OMP)

- We seek approximations to y from the dictionary $\mathcal{D} := \{\phi_1, \ldots, \phi_N\}$ consisting of the columns of Φ
- $j_1 := \underset{j=1,\cdots,N}{\text{Argmax}} \, |\langle y, \phi_j \rangle|$
- $y^1 := z^1_{j_1} \phi_{j_1}$ with $z^1_{j_1} := \langle y, \phi_{j_1} \rangle / \|\phi_{j_1}\|^2$

Orhtogonal Matching Pursuit (OMP)

- We seek approximations to y from the dictionary $\mathcal{D} := \{\phi_1, \ldots, \phi_N\}$ consisting of the columns of Φ
- $j_1 := \underset{j=1,\cdots,N}{\mathrm{Argmax}} |\langle y, \phi_j \rangle|$
- $y^1 := z^1_{j_1} \phi_{j_1}$ with $z^1_{j_1} := \langle y, \phi_{j_1} \rangle / \|\phi_{j_1}\|^2$
- i-th step: $\{j_1, \cdots, j_i\}$ and $y^i = \sum_{l=1}^{i} z^i_{j_l} \phi_{j_i}$ the orthogonal projection of y onto $\mathrm{Span}\{\phi_{j_1}, \cdots, \phi_{j_l}\}$

Orhtogonal Matching Pursuit (OMP)

- We seek approximations to y from the dictionary $\mathcal{D} := \{\phi_1, \ldots, \phi_N\}$ consisting of the columns of Φ
- $j_1 := \underset{j=1,\cdots,N}{\text{Argmax}} |\langle y, \phi_j \rangle|$
- $y^1 := z^1_{j_1} \phi_{j_1}$ with $z^1_{j_1} := \langle y, \phi_{j_1} \rangle / \|\phi_{j_1}\|^2$
- i-th step: $\{j_1, \cdots, j_i\}$ and $y^i = \sum_{l=1}^{i} z^i_{j_l} \phi_{j_i}$ the orthogonal projection of y onto $\text{Span}\{\phi_{j_1}, \cdots, \phi_{j_l}\}$
- $j_{i+1} := \underset{j=1,\cdots,N}{\text{Argmax}} |\langle r^i, \phi_j \rangle|$ where $r^i := y - y^i$ is the residual

Orhtogonal Matching Pursuit (OMP)

- We seek approximations to y from the dictionary $\mathcal{D} := \{\phi_1, \ldots, \phi_N\}$ consisting of the columns of Φ
- $j_1 := \underset{j=1,\cdots,N}{\text{Argmax}} |\langle y, \phi_j \rangle|$
- $y^1 := z^1_{j_1} \phi_{j_1}$ with $z^1_{j_1} := \langle y, \phi_{j_1} \rangle / \|\phi_{j_1}\|^2$
- i-th step: $\{j_1, \cdots, j_i\}$ and $y^i = \sum_{l=1}^{i} z^i_{j_l} \phi_{j_i}$ the orthogonal projection of y onto $\text{Span}\{\phi_{j_1}, \cdots, \phi_{j_l}\}$
- $j_{i+1} := \underset{j=1,\cdots,N}{\text{Argmax}} |\langle r^i, \phi_j \rangle|$ where $r^i := y - y^i$ is the residual
- x^i is $z^i_{j_1}, \cdots, z^i_{j_i}$ augmented by zeros in the other coordinates, $x^i_j = z^i_j, \quad \text{if} \quad j \in \{j_1, \cdots, j_i\}, \quad 0 \quad \text{otherwise.}$

Results for Decoding for Random Draws

- Gilbert and Tropp proved that for Bernouli random matrices, OMP captures a k sparse vector with high probability for the large range of k

Results for Decoding for Random Draws

- Gilbert and Tropp proved that for Bernouli random matrices, OMP captures a k sparse vector with high probability for the large range of k
- Cohen-Dahmen-DeVore extend this to general random families of matrices

Results for Decoding for Random Draws

- Gilbert and Tropp proved that for Bernouli random matrices, OMP captures a k sparse vector with high probability for the large range of k
- Cohen-Dahmen-DeVore extend this to general random families of matrices
- Are there practical decoders that give instance optimality in ℓ_2?

Results for Decoding for Random Draws

- Gilbert and Tropp proved that for Bernouli random matrices, OMP captures a k sparse vector with high probability for the large range of k
- Cohen-Dahmen-DeVore extend this to general random families of matrices
- Are there practical decoders that give instance optimality in ℓ_2?
- Wojtaszcek has shown that for Gaussian matrices ℓ_1 minimization does the job

Results for Decoding for Random Draws

- Gilbert and Tropp proved that for Bernouli random matrices, OMP captures a k sparse vector with high probability for the large range of k
- Cohen-Dahmen-DeVore extend this to general random families of matrices
- Are there practical decoders that give instance optimality in ℓ_2?
- Wojtaszcek has shown that for Gaussian matrices ℓ_1 minimization does the job
- This result rests on a geometric property of Gaussian matrices

Results for Decoding for Random Draws

- Gilbert and Tropp proved that for Bernouli random matrices, OMP captures a k sparse vector with high probability for the large range of k
- Cohen-Dahmen-DeVore extend this to general random families of matrices
- Are there practical decoders that give instance optimality in ℓ_2?
- Wojtaszcek has shown that for Gaussian matrices ℓ_1 minimization does the job
- This result rests on a geometric property of Gaussian matrices
- Namely the image of the unit ℓ_1 ball under such matrices will with high probability contain an ℓ_2 ball of radius $1/\sqrt{k}$

General Random Families

- Cohen-Dahmen-DeVore

General Random Families

- Cohen-Dahmen-DeVore
- Given an arbitrary $\epsilon > 0$ we can obtain

$$\|x - \Delta(\Phi(x))\|_{\ell_2} \leq C\sigma_k(x)_{\ell_2} + \epsilon$$

with high probability

General Random Families

- Cohen-Dahmen-DeVore
- Given an arbitrary $\epsilon > 0$ we can obtain

$$\|x - \Delta(\Phi(x))\|_{\ell_2} \leq C\sigma_k(x)_{\ell_2} + \epsilon$$

  with high probability
- Encoders that attain this

General Random Families

- Cohen-Dahmen-DeVore
- Given an arbitrary $\epsilon > 0$ we can obtain

$$\|x - \Delta(\Phi(x))\|_{\ell_2} \leq C\sigma_k(x)_{\ell_2} + \epsilon$$

  with high probability
- Encoders that attain this
  - ℓ_1 minimization

General Random Families

- Cohen-Dahmen-DeVore
- Given an arbitrary $\epsilon > 0$ we can obtain

$$\|x - \Delta(\Phi(x))\|_{\ell_2} \leq C\sigma_k(x)_{\ell_2} + \epsilon$$

  with high probability
- Encoders that attain this
  - ℓ_1 minimization
  - Greedy thresholding: at each iteration take all coordinates for which the inner product $\langle r, \phi_\nu \rangle \geq \delta \|r\|_{\ell_2}/\sqrt{k}$ where r is the residual

General Random Families

- Cohen-Dahmen-DeVore
- Given an arbitrary $\epsilon > 0$ we can obtain

$$\|x - \Delta(\Phi(x))\|_{\ell_2} \leq C\sigma_k(x)_{\ell_2} + \epsilon$$

  with high probability
- Encoders that attain this
  - ℓ_1 minimization
  - Greedy thresholding: at each iteration take all coordinates for which the inner product $\langle r, \phi_\nu \rangle \geq \delta \|r\|_{\ell_2}/\sqrt{k}$ where r is the residual
  - Here $\delta > 0$ is a fixed threshold parameter

Other Possible Decoders

- We seek other possible decoding algorithms which may be faster

Other Possible Decoders

- We seek other possible decoding algorithms which may be faster
- Here we add in the knowledge of our Compressed Sensing setting - the RIP property for Φ

Other Possible Decoders

- We seek other possible decoding algorithms which may be faster
- Here we add in the knowledge of our Compressed Sensing setting - the RIP property for Φ
- We have also discussed the least squares problem

$$\bar{x} := \underset{z \in \mathcal{F}(y)}{\text{Argmin}} \|z\|_{\ell_2} = \underset{\eta \in \mathcal{N}}{\text{Argmin}} \|x - \eta\|_{\ell_2}$$

Other Possible Decoders

- We seek other possible decoding algorithms which may be faster
- Here we add in the knowledge of our Compressed Sensing setting - the RIP property for Φ
- We have also discussed the least squares problem

$$\bar{x} := \underset{z \in \mathcal{F}(y)}{\text{Argmin}} \|z\|_{\ell_2} = \underset{\eta \in \mathcal{N}}{\text{Argmin}} \|x - \eta\|_{\ell_2}$$

- We know this does not work well

Other Possible Decoders

- We seek other possible decoding algorithms which may be faster
- Here we add in the knowledge of our Compressed Sensing setting - the RIP property for Φ
- We have also discussed the least squares problem

$$\bar{x} := \underset{z \in \mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_2} = \underset{\eta \in \mathcal{N}}{\operatorname{Argmin}} \|x - \eta\|_{\ell_2}$$

- We know this does not work well
- However it is easy to compute
  $\bar{x} = \Phi^t[\Phi\Phi^t]^{-1}\Phi x = \Phi^t[\Phi\Phi^t]^{-1}y$

Other Possible Decoders

- We seek other possible decoding algorithms which may be faster
- Here we add in the knowledge of our Compressed Sensing setting - the RIP property for Φ
- We have also discussed the least squares problem

$$\bar{x} := \underset{z \in \mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_2} = \underset{\eta \in \mathcal{N}}{\operatorname{Argmin}} \|x - \eta\|_{\ell_2}$$

- We know this does not work well
- However it is easy to compute $\bar{x} = \Phi^t[\Phi\Phi^t]^{-1}\Phi x = \Phi^t[\Phi\Phi^t]^{-1}y$
- $O(Nn^2)$ arithmetic operations

Weighted Least Squares

- Consider weighted ℓ_2 minimization.

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \ldots, N$ be a positive weight

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \ldots, N$ be a positive weight
- $\|u\|_{\ell_2(w)} := \left[\sum_{j=1}^{N} w_j u_j^2\right]^{1/2}$

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \ldots, N$ be a positive weight
- $\|u\|_{\ell_2(w)} := \left[\sum_{j=1}^{N} w_j u_j^2\right]^{1/2}$
- $\langle u, v\rangle_w := \sum_{j=1}^{N} w_j u_j v_j$

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \ldots, N$ be a positive weight
- $\|u\|_{\ell_2(w)} := \left[\sum_{j=1}^N w_j u_j^2\right]^{1/2}$
- $\langle u, v\rangle_w := \sum_{j=1}^N w_j u_j v_j$
- Define $x(w) := \underset{z\in\mathcal{F}(y)}{\text{Argmin}} \|z\|_{\ell_2(w)}$

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \ldots, N$ be a positive weight
- $\|u\|_{\ell_2(w)} := \left[\sum_{j=1}^N w_j u_j^2\right]^{1/2}$
- $\langle u, v\rangle_w := \sum_{j=1}^N w_j u_j v_j$
- Define $x(w) := \underset{z\in\mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_2(w)}$
- $x(w) = x - \eta(w)$ where $\eta(w) := \underset{\eta\in\mathcal{N}}{\operatorname{Argmin}} \|x - \eta\|_{\ell_2(w)}$

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \ldots, N$ be a positive weight
- $\|u\|_{\ell_2(w)} := \left[\sum_{j=1}^N w_j u_j^2\right]^{1/2}$
- $\langle u, v\rangle_w := \sum_{j=1}^N w_j u_j v_j$
- Define $x(w) := \underset{z\in\mathcal{F}(y)}{\operatorname{Argmin}} \|z\|_{\ell_2(w)}$
- $x(w) = x - \eta(w)$ where $\eta(w) := \underset{\eta\in\mathcal{N}}{\operatorname{Argmin}} \|x - \eta\|_{\ell_2(w)}$
- Note that this solution is characterized by the orthogonality conditions $\langle x(w), \eta\rangle_w = 0, \quad \eta \in \mathcal{N}$

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \ldots, N$ be a positive weight
- $\|u\|_{\ell_2(w)} := \left[\sum_{j=1}^N w_j u_j^2\right]^{1/2}$
- $\langle u, v\rangle_w := \sum_{j=1}^N w_j u_j v_j$
- Define $x(w) := \operatorname*{Argmin}_{z \in \mathcal{F}(y)} \|z\|_{\ell_2(w)}$
- $x(w) = x - \eta(w)$ where $\eta(w) := \operatorname*{Argmin}_{\eta \in \mathcal{N}} \|x - \eta\|_{\ell_2(w)}$
- Note that this solution is characterized by the orthogonality conditions $\langle x(w), \eta\rangle_w = 0, \quad \eta \in \mathcal{N}$
- We can again solve for $x(w)$ in $O(Nn^2)$ operations

Connections

- Suppose $x^* \in I\!R^N$ is the ℓ_1 minimizer from $\mathcal{F}(y)$ and $T = \mathrm{supp}(\mathrm{x}^*)$

Connections

- Suppose $x^* \in I\!R^N$ is the ℓ_1 minimizer from $\mathcal{F}(y)$ and $T = \mathrm{supp(x^*)}$
- If $w_j = |x_j^*|^{-1}$, $j \in T$ then

$$x(w) = x^*$$

Connections

- Suppose $x^* \in I\!R^N$ is the ℓ_1 minimizer from $\mathcal{F}(y)$ and $T = \mathrm{supp}(\mathrm{x}^*)$
- If $w_j = |x_j^*|^{-1}$, $j \in T$ then

$$x(w) = x^*$$

- If x^* has full support then ℓ_1 minimization means that $(\mathrm{sign}(x_i))_{i=1}^N$ is orthogonal to $\mathcal{N}$

Connections

- Suppose $x^* \in I\!R^N$ is the ℓ_1 minimizer from $\mathcal{F}(y)$ and $T = \mathrm{supp}(\mathrm{x}^*)$
- If $w_j = |x_j^*|^{-1}$, $j \in T$ then

$$x(w) = x^*$$

- If x^* has full support then ℓ_1 minimization means that $(\mathrm{sign}(x_i))_{i=1}^N$ is orthogonal to $\mathcal{N}$
- For least squares we have $w_i x_i^* = \mathrm{sign}(x_i^*)$ and so x^* satisfies the weighted ℓ_2 orthogonality conditions.

Connections

- Suppose $x^* \in I\!R^N$ is the ℓ_1 minimizer from $\mathcal{F}(y)$ and $T = \mathrm{supp}(\mathrm{x}^*)$
- If $w_j = |x_j^*|^{-1}$, $j \in T$ then

$$x(w) = x^*$$

- If x^* has full support then ℓ_1 minimization means that $(\mathrm{sign}(x_i))_{i=1}^N$ is orthogonal to $\mathcal{N}$
- For least squares we have $w_i x_i^* = \mathrm{sign}(x_i^*)$ and so x^* satisfies the weighted ℓ_2 orthogonality conditions.
- Unique minimizer for these problems shows $x(w) = x^*$

Iterative Weighted Least Squares

- We would like to iteratively choose weights with the result convergence to the ℓ_1 minimizer $x^* \in \mathcal{F}(y)$

Iterative Weighted Least Squares

- We would like to iteratively choose weights with the result convergence to the ℓ_1 minimizer $x^* \in \mathcal{F}(y)$
- In the process, we want to always deal with strictly convex problems and strictly positive weights

Iterative Weighted Least Squares

- We would like to iteratively choose weights with the result convergence to the ℓ_1 minimizer $x^* \in \mathcal{F}(y)$
- In the process, we want to always deal with strictly convex problems and strictly positive weights
- To do this we introduce the following functional

Iterative Weighted Least Squares

- We would like to iteratively choose weights with the result convergence to the ℓ_1 minimizer $x^* \in \mathcal{F}(y)$
- In the process, we want to always deal with strictly convex problems and strictly positive weights
- To do this we introduce the following functional
- $$\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N}(\epsilon^2 w_j + w_j^{-1})\right].$$

Iterative Weighted Least Squares

- We would like to iteratively choose weights with the result convergence to the ℓ_1 minimizer $x^* \in \mathcal{F}(y)$
- In the process, we want to always deal with strictly convex problems and strictly positive weights
- To do this we introduce the following functional
- $$\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N}(\epsilon^2 w_j + w_j^{-1})\right].$$
- We will now describe a recursive algorithm

Iterative Weighted Least Squares

- We would like to iteratively choose weights with the result convergence to the ℓ_1 minimizer $x^* \in \mathcal{F}(y)$
- In the process, we want to always deal with strictly convex problems and strictly positive weights
- To do this we introduce the following functional
- $$\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N}(\epsilon^2 w_j + w_j^{-1})\right].$$
- We will now describe a recursive algorithm
- If $z \in I\!R^N$ we let $r(z)_K$ be its K-th largest entry (in absolute value)

The Algorithm

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

The Algorithm

- $\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N} (\epsilon^2 w_j + w_j^{-1})\right].$
- Initialize: $w^0 := (1, \ldots, 1), \quad \epsilon_0 := 1$

The Algorithm

- $\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N} (\epsilon^2 w_j + w_j^{-1})\right].$
- Initialize: $w^0 := (1, \ldots, 1), \quad \epsilon_0 := 1$
- $x^{m+1} := \underset{z \in \mathcal{F}(y)}{\text{Argmin}}\ \mathcal{J}(z, w^m, \epsilon_m), \quad m = 0, 1, \ldots$

The Algorithm

- $\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N}(\epsilon^2 w_j + w_j^{-1})\right].$
- Initialize: $w^0 := (1, \ldots, 1), \quad \epsilon_0 := 1$
- $x^{m+1} := \underset{z \in \mathcal{F}(y)}{\text{Argmin}}\ \mathcal{J}(z, w^m, \epsilon_m), \quad m = 0, 1, \ldots$
- $\epsilon_{m+1} := \min(\epsilon_m, \frac{r(x^{m+1})_K}{N}),$

The Algorithm

- $\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N} (\epsilon^2 w_j + w_j^{-1})\right].$
- Initialize: $w^0 := (1, \ldots, 1), \quad \epsilon_0 := 1$
- $x^{m+1} := \underset{z \in \mathcal{F}(y)}{\text{Argmin}}\ \mathcal{J}(z, w^m, \epsilon_m), \quad m = 0, 1, \ldots$
- $\epsilon_{m+1} := \min(\epsilon_m, \frac{r(x^{m+1})_K}{N}),$
- K is a fixed integer to be described

The Algorithm

- $\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N} (\epsilon^2 w_j + w_j^{-1})\right].$
- Initialize: $w^0 := (1, \ldots, 1), \quad \epsilon_0 := 1$
- $x^{m+1} := \underset{z \in \mathcal{F}(y)}{\text{Argmin}}\ \mathcal{J}(z, w^m, \epsilon_m), \quad m = 0, 1, \ldots$
- $\epsilon_{m+1} := \min(\epsilon_m, \frac{r(x^{m+1})_K}{N}),$
- K is a fixed integer to be described
- $w^{m+1} := \underset{w>0}{\text{Argmin}}\ \mathcal{J}(x^{m+1}, w, \epsilon_{m+1})$

The Algorithm

- $\mathcal{J}(z, w, \epsilon) := \frac{1}{2}\left[\sum_{j=1}^{N} z_j^2 w_j + \sum_{j=1}^{N} (\epsilon^2 w_j + w_j^{-1})\right].$
- Initialize: $w^0 := (1, \ldots, 1), \quad \epsilon_0 := 1$
- $x^{m+1} := \underset{z \in \mathcal{F}(y)}{\text{Argmin}}\ \mathcal{J}(z, w^m, \epsilon_m), \quad m = 0, 1, \ldots$
- $\epsilon_{m+1} := \min(\epsilon_m, \frac{r(x^{m+1})_K}{N}),$
- K is a fixed integer to be described
- $w^{m+1} := \underset{w>0}{\text{Argmin}}\ \mathcal{J}(x^{m+1}, w, \epsilon_{m+1})$
- $w_j^{m+1} = [(x_j^{m+1})^2 + \epsilon_{m+1}^2]^{-1/2}, \quad j = 1, \ldots, N$

Daubechies-DeVore-Fornasier-Gunturk

- These authors prove several results on the convergence of the above algorithm

Daubechies-DeVore-Fornasier-Gunturk

- These authors prove several results on the convergence of the above algorithm
- The main result we prove is the following theorem

Theorem

Let $k \geq 1$ and define $K = k + 6$. We assume that Φ satisfies the Null Space Property for ℓ_1 of order $3K$ for $\gamma \leq 1/2$. Then, for each $x \in I\!R^N$, $y = \Phi(x)$, the Algorithm converges and its limit $\bar{x}$ satisfies

$$\|x - \bar{x}\|_{\ell_1} \leq C_1 \sigma_k(x)_{\ell_1}, \quad C_1 := \frac{5(1+\gamma)}{1-\gamma}.$$

In particular if x is k-sparse then x^m converges to x.

Exponential Convergence

- A second theorem shows that the algorithm converges exponentially to x^* if $x^* \in \Sigma_k$ and the starting point is close enough

  Theorem For a given $0 < \rho < 1$, assume Φ satisfies NSP of order $3K$ with constant γ such that

  $\mu := \frac{\gamma}{1-\rho}\left(1 + \frac{1}{K-k}\right) < 1$. Let m_0 be such that

  $$\|x^{m_0} - x^*\|_{\ell_1} \leq R^* := \rho \min_{i \in T} |x_i| = \rho\, r(x)_k.$$

  Then for all $m \geq m_0$, we have

  $$\|x^{m+1} - x^*\|_{\ell_1} \leq \mu \|x^m - x^*\|_{\ell_1}$$

  Consequently x^m converges to x^* exponentially.

Exponential Convergence

- A second theorem shows that the algorithm converges exponentially to x^* if $x^* \in \Sigma_k$ and the starting point is close enough

  Theorem For a given $0 < \rho < 1$, assume Φ satisfies NSP of order $3K$ with constant γ such that

  $\mu := \frac{\gamma}{1-\rho}\left(1 + \frac{1}{K-k}\right) < 1$. Let m_0 be such that

  $$\|x^{m_0} - x^*\|_{\ell_1} \leq R^* := \rho \min_{i \in T} |x_i| = \rho\, r(x)_k.$$

  Then for all $m \geq m_0$, we have

  $$\|x^{m+1} - x^*\|_{\ell_1} \leq \mu \|x^m - x^*\|_{\ell_1}$$

  Consequently x^m converges to x^* exponentially.

Exponential Convergence 2

- It is an interesting question whether the algorithm actually converges exponentially to $x^* \in \Sigma_k$ from the get go

Exponential Convergence 2

- It is an interesting question whether the algorithm actually converges exponentially to $x^* \in \Sigma_k$ from the get go
- This is observed in practice but only proved for one or two supported vectors

Theorem Assume Φ satisfies NSP of order $3K$ with constant γ sufficiently small. Then for any vector x^* which has support $k = 1, 2$ there is an absolute constant C_0 and a $\rho < 1$ such that

$$\|x^m - x^*\|_{\ell_1} \leq C_0 \rho^m \|x\|_{\ell_1}, \quad m = 1, 2, \ldots$$

Proofs of these Results

- The proofs of these results are interesting in how they utilize RIP in the form of the Null Space Property

Proofs of these Results

- The proofs of these results are interesting in how they utilize RIP in the form of the Null Space Property
- We shall begin with the proof of the convergence of the algorithm

Proofs of these Results

- The proofs of these results are interesting in how they utilize RIP in the form of the Null Space Property
- We shall begin with the proof of the convergence of the algorithm
- Before embarking on the proof we want to bring out a certain geometric result on ℓ_1 minimization which we shall utilize

Proofs of these Results

- The proofs of these results are interesting in how they utilize RIP in the form of the Null Space Property
- We shall begin with the proof of the convergence of the algorithm
- Before embarking on the proof we want to bring out a certain geometric result on ℓ_1 minimization which we shall utilize
- As a preliminary we consider the operation of rearrangements

Proofs of these Results

- The proofs of these results are interesting in how they utilize RIP in the form of the Null Space Property
- We shall begin with the proof of the convergence of the algorithm
- Before embarking on the proof we want to bring out a certain geometric result on ℓ_1 minimization which we shall utilize
- As a preliminary we consider the operation of rearrangements
- We define $r(z)$ as the rearrangement of the sequence $|z_j|$ into decreasing order. In other words $r(z)_j$ is the j-th largest of the $|z_\nu|$

Rearrangements

- Rearrangement is a Lipschitz map on $\|\cdot\|_{\ell_\infty}$

Rearrangements

- Rearrangement is a Lipschitz map on $\|\cdot\|_{\ell_\infty}$
- More precisely $\|r(z) - r(z')\|_{\ell_\infty} \leq \|z - z'\|_{\ell_\infty}$

Rearrangements

- Rearrangement is a Lipschitz map on $\|\cdot\|_{\ell_\infty}$
- More precisely $\|r(z) - r(z')\|_{\ell_\infty} \leq \|z - z'\|_{\ell_\infty}$
- Moreover, for any j, we have

$$|\sigma_j(z)_{\ell_1} - \sigma_j(z')_{\ell_1}| \leq \|z - z'\|_{\ell_1}$$

Rearrangements

- Rearrangement is a Lipschitz map on $\|\cdot\|_{\ell_\infty}$
- More precisely $\|r(z) - r(z')\|_{\ell_\infty} \leq \|z - z'\|_{\ell_\infty}$
- Moreover, for any j, we have

$$|\sigma_j(z)_{\ell_1} - \sigma_j(z')_{\ell_1}| \leq \|z - z'\|_{\ell_1}$$

- For any $J > j$, we have

$$(J - j)r(z)_J \leq \|z - z'\|_{\ell_1} + \sigma_j(z')_{\ell_1}$$

Proof of Rearrangement Lemma

- Given z, z' and $j \in \{1, \ldots, N\}$, let Λ be a set corresponding to the $j-1$ largest entries in z'

Proof of Rearrangement Lemma

- Given z, z' and $j \in \{1, \ldots, N\}$, let Λ be a set corresponding to the $j-1$ largest entries in z'
- $$r(z)_j \leq \max_{i \in \Lambda^c} |z_i| \leq \max_{i \in \Lambda^c} |z'_i| + \|z - z'\|_{\ell_\infty} \leq r(z')_j + \|z - z'\|_{\ell_\infty}$$

Proof of Rearrangement Lemma

- Given z, z' and $j \in \{1, \ldots, N\}$, let Λ be a set corresponding to the $j-1$ largest entries in z'
- $$r(z)_j \leq \max_{i \in \Lambda^c} |z_i| \leq \max_{i \in \Lambda^c} |z'_i| + \|z - z'\|_{\ell_\infty} \leq r(z')_j + \|z - z'\|_{\ell_\infty}$$
- Reverse the roles of z and z'

Proof of Rearrangement Lemma

- Given z, z' and $j \in \{1, \ldots, N\}$, let Λ be a set corresponding to the $j-1$ largest entries in z'
- $$r(z)_j \leq \max_{i \in \Lambda^c} |z_i| \leq \max_{i \in \Lambda^c} |z'_i| + \|z - z'\|_{\ell_\infty} \leq r(z')_j + \|z - z'\|_{\ell_\infty}$$
- Reverse the roles of z and z'
- Next we approximate z by the j-term approximation u of z' in ℓ_1

$$\sigma_j(z)_{\ell_1} \leq \|z - u\|_{\ell_1} \leq \|z - z'\|_{\ell_1} + \sigma_j(z')_{\ell_1},$$

Proof of Rearrangement Lemma

- Given z, z' and $j \in \{1, \ldots, N\}$, let Λ be a set corresponding to the $j-1$ largest entries in z'
- $$r(z)_j \leq \max_{i \in \Lambda^c} |z_i| \leq \max_{i \in \Lambda^c} |z'_i| + \|z - z'\|_{\ell_\infty} \leq r(z')_j + \|z - z'\|_{\ell_\infty}$$
- Reverse the roles of z and z'
- Next we approximate z by the j-term approximation u of z' in ℓ_1

$$\sigma_j(z)_{\ell_1} \leq \|z - u\|_{\ell_1} \leq \|z - z'\|_{\ell_1} + \sigma_j(z')_{\ell_1},$$

- Again reverse roles of z, z'

Proof of Rearrangement Lemma

- Given z, z' and $j \in \{1, \ldots, N\}$, let Λ be a set corresponding to the $j-1$ largest entries in z'
- $$r(z)_j \leq \max_{i \in \Lambda^c} |z_i| \leq \max_{i \in \Lambda^c} |z'_i| + \|z - z'\|_{\ell_\infty} \leq r(z')_j + \|z - z'\|_{\ell_\infty}$$
- Reverse the roles of z and z'
- Next we approximate z by the j-term approximation u of z' in ℓ_1

$$\sigma_j(z)_{\ell_1} \leq \|z - u\|_{\ell_1} \leq \|z - z'\|_{\ell_1} + \sigma_j(z')_{\ell_1},$$

- Again reverse roles of z, z'
- $(J - j) r(z)_J \leq \sigma_j(z)_{\ell_1}$

A Geometric Property

- Assume that NSP holds for some k and $\gamma < 1$

A Geometric Property

- Assume that NSP holds for some k and $\gamma < 1$
- For any $z, z' \in \mathcal{F}(y)$

$$\|z' - z\|_{\ell_1} \leq \frac{1+\gamma}{1-\gamma} \left(\|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1} \right).$$

Proof of Geometric Property

- T the set of indices of the k largest entries in z

$$\begin{aligned}
\|(z' - z)_{T^c}\|_{\ell_1} &\leq \|z'_{T^c}\|_{\ell_1} + \|z_{T^c}\|_{\ell_1} \\
&= \|z'\|_{\ell_1} - \|z'_T\|_{\ell_1} + \sigma_k(z)_{\ell_1} \\
&= \|z\|_{\ell_1} + \|z'\|_{\ell_1} - \|z\|_{\ell_1} - \|z'_T\|_{\ell_1} + \sigma_k(z)_{\ell_1} \\
&\leq \|z_T\|_{\ell_1} - \|z'_T\|_{\ell_1} + \|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1} \\
&\leq \|(z' - z)_T\|_{\ell_1} + \|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1}
\end{aligned}$$

Proof of Geometric Property

- T the set of indices of the k largest entries in z

$$\begin{aligned}
\|(z'-z)_{T^c}\|_{\ell_1} &\leq \|z'_{T^c}\|_{\ell_1} + \|z_{T^c}\|_{\ell_1} \\
&= \|z'\|_{\ell_1} - \|z'_T\|_{\ell_1} + \sigma_k(z)_{\ell_1} \\
&= \|z\|_{\ell_1} + \|z'\|_{\ell_1} - \|z\|_{\ell_1} - \|z'_T\|_{\ell_1} + \sigma_k(z)_{\ell_1} \\
&\leq \|z_T\|_{\ell_1} - \|z'_T\|_{\ell_1} + \|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1} \\
&\leq \|(z'-z)_T\|_{\ell_1} + \|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1}
\end{aligned}$$

- Using NSP $\|(z'-z)_T\|_{\ell_1}$

$$\leq \gamma\|(z'-z)_{T^c}\|_{\ell_1} \leq \gamma(\|(z'-z)_T\|_{\ell_1} + \|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1})$$

Proof Continued

- In other words,

$$\|(z' - z)_T\|_{\ell_1} \leq \frac{\gamma}{1-\gamma}(\|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1}).$$

Proof Continued

- In other words,

$$\|(z' - z)_T\|_{\ell_1} \leq \frac{\gamma}{1-\gamma}(\|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1}).$$

- Finally

$$\|z' - z\|_{\ell_1} = \|(z' - z)_{T^c}\|_{\ell_1} + \|(z' - z)_T\|_{\ell_1}$$

$$\leq \frac{1+\gamma}{1-\gamma}(\|z'\|_{\ell_1} - \|z\|_{\ell_1} + 2\sigma_k(z)_{\ell_1}),$$