IMA Lecture 7

Ronald DeVore

Last Time

- We showed instance optimality of order k in ℓ_2 with high probability is possible

Last Time

- We showed instance optimality of order k in ℓ_2 with high probability is possible
- We are fighting for practical decoders that achieve this

Last Time

- We showed instance optimality of order k in ℓ_2 with high probability is possible
- We are fighting for practical decoders that achieve this
- We introduced the Orthogonal Matching Pursuit (OMP) algorithm for finding an approximation to x from the information $y = \Phi x$ and showed that for random families $\Phi(\omega),\ \omega \in \Omega$, then with high probability it recovers x exactly if this vector is in Σ_k

Computational Complexity of OMP

- Let us analyze the number of computations needed to implement the OMP algorithm

Computational Complexity of OMP

- Let us analyze the number of computations needed to implement the OMP algorithm
- Suppose after step i, we have computed y^i, the residual r^i and an orthonormal system $\phi^*_{j_1}, \ldots, \phi^*_{j_i}$ (by Gramm-Schmidt)

Computational Complexity of OMP

- Let us analyze the number of computations needed to implement the OMP algorithm
- Suppose after step i, we have computed y^i, the residual r^i and an orthonormal system $\phi^*_{j_1}, \ldots, \phi^*_{j_i}$ (by Gramm-Schmidt)
- At the next iteration, we have to compute $N - i \leq N$ inner products $\langle r^i, \phi_j \rangle$, $j \neq j_1, \ldots, j_i$

Computational Complexity of OMP

- Let us analyze the number of computations needed to implement the OMP algorithm
- Suppose after step i, we have computed y^i, the residual r^i and an orthonormal system $\phi^*_{j_1}, \ldots, \phi^*_{j_i}$ (by Gramm-Schmidt)
- At the next iteration, we have to compute $N - i \leq N$ inner products $\langle r^i, \phi_j \rangle$, $j \neq j_1, \ldots, j_i$
- This takes $\leq 2Nn$ arithmetic operations

Computational Complexity of OMP

- Let us analyze the number of computations needed to implement the OMP algorithm
- Suppose after step i, we have computed y^i, the residual r^i and an orthonormal system $\phi^*_{j_1}, \ldots, \phi^*_{j_i}$ (by Gramm-Schmidt)
- At the next iteration, we have to compute $N - i \le N$ inner products $\langle r^i, \phi_j \rangle$, $j \neq j_1, \ldots, j_i$
- This takes $\le 2Nn$ arithmetic operations
- We have to do a comparison of these inner products

Computational Complexity of OMP

- Let us analyze the number of computations needed to implement the OMP algorithm
- Suppose after step i, we have computed y^i, the residual r^i and an orthonormal system $\phi^*_{j_1}, \ldots, \phi^*_{j_i}$ (by Gramm-Schmidt)
- At the next iteration, we have to compute $N - i \leq N$ inner products $\langle r^i, \phi_j \rangle$, $j \neq j_1, \ldots, j_i$
- This takes $\leq 2Nn$ arithmetic operations
- We have to do a comparison of these inner products
- This can be done with N comparisons

OMP Complexity Continued

- We need to update our orthogonalization :

$$\phi^*_{j_{i+1}} := \frac{\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}}{\|\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}\|_{\ell_2}}, \quad Y_i := \mathrm{span}\{\phi_{j_1}, \dots, \phi_{j_i}\}$$

OMP Complexity Continued

- We need to update our orthogonalization :
  $\phi^*_{j_{i+1}} := \frac{\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}}{\|\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}\|_{\ell_2}}, \quad Y_i := \operatorname{span}\{\phi_{j_1}, \ldots, \phi_{j_i}\}$
- $P_{Y_i} f = \sum_{\nu=1}^{i} \langle f, \phi^*_{j_\nu} \rangle \phi^*_{j_\nu}$

OMP Complexity Continued

- We need to update our orthogonalization :

  $\phi^*_{j_{i+1}} := \frac{\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}}{\|\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}\|_{\ell_2}}, \quad Y_i := \mathrm{span}\{\phi_{j_1}, \dots, \phi_{j_i}\}$

- $P_{Y_i} f = \sum_{\nu=1}^{i} \langle f, \phi^*_{j_\nu} \rangle \phi^*_{j_\nu}$
- This takes $\leq 4in \leq 4nN$ operations

OMP Complexity Continued

- We need to update our orthogonalization :
  $\phi^*_{j_{i+1}} := \frac{\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}}{\|\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}\|_{\ell_2}}, \quad Y_i := \mathrm{span}\{\phi_{j_1}, \ldots, \phi_{j_i}\}$
- $P_{Y_i} f = \sum_{\nu=1}^{i} \langle f, \phi^*_{j_\nu} \rangle \phi^*_{j_\nu}$
- This takes $\leq 4in \leq 4nN$ operations
- To compute $y - P_{Y_{i+1}} y$ needs again at most $5nN$ operations

OMP Complexity Continued

- We need to update our orthogonalization :
  $\phi^*_{j_{i+1}} := \frac{\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}}{\|\phi_{j_{i+1}} - P_{Y_i}\phi_{j_{i+1}}\|_{\ell_2}}, \quad Y_i := \mathrm{span}\{\phi_{j_1}, \dots, \phi_{j_i}\}$
- $P_{Y_i} f = \sum_{\nu=1}^{i} \langle f, \phi^*_{j_\nu} \rangle \phi^*_{j_\nu}$
- This takes $\leq 4in \leq 4nN$ operations
- To compute $y - P_{Y_{i+1}} y$ needs again at most $5nN$ operations
- Since there are k steps we have $O(Nnk)$ operations in total

Outstanding Problems

- Will OMP give ℓ_2 instance optimality of order k in probability

Outstanding Problems

- Will OMP give ℓ_2 instance optimality of order k in probability
- Cohen-Dahmen-DeVore

Outstanding Problems

- Will OMP give ℓ_2 instance optimality of order k in probability
- Cohen-Dahmen-DeVore
- Can we use ℓ_1 minimization to obtain ℓ_2 instance optimality of order k in probability?

Outstanding Problems

- Will OMP give ℓ_2 instance optimality of order k in probability
- Cohen-Dahmen-DeVore
- Can we use ℓ_1 minimization to obtain ℓ_2 instance optimality of order k in probability?
- Cohen-Dahmen-DeVore have another algorithm that almost does this

ℓ_1 minimization

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

ℓ_1 minimization

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

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

ℓ_1 minimization

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

ℓ_1 minimization

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

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

ℓ_1 minimization

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

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

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

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

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

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

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

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

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

- 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

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

- 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

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

- 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 $x(w) := \operatorname*{Argmin}_{z \in \mathcal{F}(y)} \|z\|_{\ell_2(w)}$

Weighted Least Squares

- Consider weighted ℓ_2 minimization.
- Let $w_j > 0$, $j = 1, \dots, N$ 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 $\eta(w) := \operatorname*{Argmin}_{\eta \in \mathcal{N}} \|x - \eta\|_{\ell_2(w)}$

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 $\langle x(w), \eta\rangle_w = 0, \quad \eta \in \mathcal{N}$

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 $\langle x(w), \eta\rangle_w = 0, \quad \eta \in \mathcal{N}$
- We can again solve for $x(w)$ in $O(Nn^2)$ operations

Connections

- Suppose $x^* \in \mathbb{R}^N$ 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 $T = \mathrm{supp}(\mathrm{x}^*)$
- 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 $T = \mathrm{supp(x^*)}$
- 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 $x^* \in \mathbb{R}^N$ 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 $T = \mathrm{supp(x^*)}$
- 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

Iterative Weighted Least Squares

- We would like to iteratively choose weights with the result convergence to the ℓ_1 minimizer
- 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
- 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
- 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
- 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
- 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 $r(z)_K$ as 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: $w^0 := (1,\dots,1), \quad \epsilon_0 := 1$
- $x^{m+1} := \underset{z\in\mathcal{F}(y)}{\operatorname{Argmin}}\ \mathcal{J}(z,w^m,\epsilon_m), \quad m = 0,1,\dots$

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: $w^0 := (1,\dots,1), \quad \epsilon_0 := 1$
- $x^{m+1} := \underset{z\in\mathcal{F}(y)}{\operatorname{Argmin}}\ \mathcal{J}(z,w^m,\epsilon_m), \quad m = 0,1,\dots$
- $\epsilon_{m+1} := \min(\epsilon_m, \frac{r(x^{m+1})_K}{N}),$
- K is a fixed integer to be described
- $w^{m+1} := \underset{w>0}{\operatorname{Argmin}}\ \mathcal{J}(x^{m+1},w,\epsilon_{m+1})$

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 $\gamma \leq 1/2$. Let x^* be the unique minimum ℓ_1 minimizer from $\mathcal{F}(y)$. Then, for each $y \in I\!R^n$, 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 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 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

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

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

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

  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

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

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

Proof of Rearrangement Lemma

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

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

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'
- $$r(z)_j \le \max_{i \in \Lambda^c} |z_i| \le \max_{i \in \Lambda^c} |z'_i| + \|z - z'\|_{\ell_\infty} \le r(z')_j + \|z - z'\|_{\ell_\infty}$$
- Reverse the roles of z and z'
- Next we approximate z by the j-term approximation u of z' in ℓ_1

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

- Again reverse roles of z, z'

Proof of Rearrangement Lemma

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

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

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

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

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

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