Chapter 4

Sparse Recovery

In this chapter we will study algorithms for sparse recovery: given a matrix A and a vector b that is a sparse linear combination of its columns – i.e. $Ax = b$ and x is sparse – when can solve for x?

4.1 Basics

Throughout this section, we will consider only linear systems $Ax = b$ where A has more columns than rows. Hence there is more than one solution for x (if there is any solution at all), and we will be interested in finding the solution that has the smallest number of non-zeros:

Definition 4.1.1 *Let $\|x\|_0$ be the number of non-zero entries of x.*

Unfortunately finding the sparsest solution to a system of linear equations in full generality is computationally hard, but there will be a number of important examples where we can solve for x efficiently.

Question 7 *When can we find for the sparsest solution to $Ax = b$?*

A trivial observation is that we can recover x when A has full column rank. In this case we can set $x = A^+ b$, where A^+ is the left-pseudo inverse of A. Note that this procedure works regardless of whether or not x is sparse. In contrast, when A has more columns than rows we will need to take advantage of the sparsity of x. We will show that under certain conditions on A, if x is sparse enough then indeed it is the uniquely sparsest solution to $Ax = b$.

Our first goal is to prove that finding the sparsest solution to a linear system is hard. We will begin with the related problem:

Problem 1 (P) *Find the sparsest non-zero vector x in a given subspace S*

Khachiyan [81] proved that this problem is NP-hard, and this result has many interesting applications that we will discuss later.

Reduction from Subset Sum

We reduce from the following variant of subset sum:

Problem 2 (S) *Given distinct values $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$, does there exist a set $I \subseteq [m]$ such that $|I| = n$ and $\sum_{i \in I} \alpha_i = 0$?*

We will embed an instance of this problem into the problem of finding the sparsest non-zero vector in a given subspace. We will make use of the following mapping which is called the *weird moment curve*:

$$\Gamma^w(\alpha_i) \Rightarrow \begin{bmatrix} 1 \\ \alpha_i \\ \alpha_i^2 \\ \cdots \\ \alpha_i^{n-2} \\ \alpha_i^n \end{bmatrix} \in \mathbb{R}^n$$

Note that this differs from the standard moment curve since the weird moment curve has α_i^n instead of α_i^{n-1}.

Claim 4.1.2 *A set I with $|I| = n$ has $\sum_{i \in I} \alpha_i = 0$ if and only if the set of vectors $\{\Gamma^w(\alpha_i)\}_{i \in I}$ is linearly dependent.*

Proof: Consider the determinant of the matrix whose columns are $\{\Gamma^w(\alpha_i)\}_{i \in I}$. Then the proof is based on the following observations:

(a) The determinant is a polynomial in the variables α_i with total degree $\binom{n}{2} + 1$, which can be seen by writing the determinant in terms of its Laplace expansion (see e.g. [74]).

(b) Moreover the determinant is divisible by $\prod_{i<j} \alpha_i - \alpha_j$, since the determinant is zero if any $\alpha_i = \alpha_j$.

Hence we can write the determinant as

$$\Big(\prod_{\substack{i<j\\ i,j\in I}}(\alpha_i-\alpha_j)\Big)\Big(\sum_{i\in I}\alpha_i\Big)$$

We have assumed that the α_i's are distinct, and consequently the determinant is zero if and only if the sum of $\alpha_i = 0$. ■

We can now complete the proof that finding the sparsest non-zero vector in a subspace is hard: We can set A to be the $n \times m$ matrix whose columns are $\Gamma^w(\alpha_i)$, and let $S = \ker(A)$. Then there is a vector $x \in S$ with $\|x\|_0 = n$ if and only if there is a subset I with $|I| = n$ whose corresponding submatrix is singular. If there is no such set I then any $x \in S$ has $\|x\|_0 > n$. Hence if we could find the sparsest non-zero vector in S we could solve the above variant of subset sum.

In fact, this same proof immediately yields an interesting result in computational geometry (that was "open" several years after Khachiyan's paper).

Definition 4.1.3 *A set of m vectors in $\mathbb{R}^n$ is in* general position *if every set of at most n vectors is linearly independent.*

From the above reduction we get that it is hard to decide whether a set of m vectors in $\mathbb{R}^n$ is in general position or not (since there is an I with $|I| = n$ whose submatrix is singular if and only if the vectors $\Gamma^w(\alpha_i)$ are not in general position).

Now we return to our original problem:

Problem 3 (Q) *Find the sparsest solution x to $Ax = b$*

There is a subtle difference between **(P)** and **(Q)** since in **(P)** we restrict to *non-zero* vectors x but in **(Q)** there is no such restriction on x. However there is a simple many-to-one reduction from **(Q)** to **(P)**.

Lemma 4.1.4 *Finding the sparsest solution x to $Ax = b$ is NP-hard.*

Proof: Suppose we are given a linear system $Ax = 0$ and we would like to find the sparsest non-zero solution x. Let A^{-i} be equal to the matrix A with he ith column deleted. Then for each i, let x^{-i} be the sparsest solution to $A^{-i}x^{-i} = A_i$. Let i^* be the index where x^{-i} is the sparsest, and suppose $\|x^{-i}\|_0 = k$. We can build a solution x to $Ax = 0$ with $\|x\|_0 = k + 1$ by setting the i^*th coordinate of x to be -1. Indeed, it is not hard to see that x is the sparsest solution to $Ax = 0$. ■

4.2 Uniqueness and Uncertainty Principles

Incoherence

Here we will define the notion of an *incoherent* matrix A, and prove that if x is sparse enough then it is the uniquely sparsest solution to $Ax = b$.

Definition 4.2.1 *The columns of $A \in \mathbb{R}^{n \times m}$ are μ-*incoherent* if for all $i \neq j$:*

$$|\langle A_i, A_j \rangle| \leq \mu \|A_i\| \cdot \|A_j\|$$

While the results we derive here can be extended to general A, we will restrict our attention to the case where $\|A_i\| = 1$, and hence a matrix is μ-incoherent if for all $i \neq j$, $|\langle A_i, A_j \rangle| \leq \mu$.

In fact, incoherent matrices are quite common. Suppose we choose m unit vetors at random in $\mathbb{R}^n$; then it is not hard to show that these vectors will be incoherent with $\mu = O(\sqrt{\frac{\log m}{n}})$. Hence even if $m = n^{100}$, these vectors will be $\widetilde{O}(1/\sqrt{n})$ incoherent. In fact, there are even better constructions of incoherent vectors that remove the logarithmic factors; this is almost optimal since for any $m > n$, any set of m vectors in $\mathbb{R}^n$ has incoherence at least $\frac{1}{\sqrt{n}}$.

We will return to the following example several times: Consider the matrix $A = [I, D]$, where $I \in \mathbb{R}^{n \times n}$ is the identity matrix and $D \in \mathbb{R}^{n \times n}$ is the DFT matrix. In particular, $D_{ij} = \frac{w^{(i-1)(j-1)}}{\sqrt{n}}$ where $w = e^{i\frac{2\pi}{n}}$. This is often referred to as the spikes-and-sines matrix. It is not hard to see that $\mu = \frac{1}{\sqrt{n}}$ here.

Uncertainty Principles

The important point is that if A is incoherent, then if x is sparse enough it will be the uniquely sparsest solution to $Ax = b$. These types of results were first established by the pioneering work of Donoho and Stark [53], and are based on establishing an *uncertainty principle.*

Lemma 4.2.2 *Suppose we have $A = [U, V]$, where U and V are orthogonal. If $b = U\alpha = V\beta$, then $\|\alpha\|_0 + \|\beta\|_0 \geq \frac{2}{\mu}$.*

The interpretation of this result for the spikes-and-sines matrix is that any signal must have at least $\sqrt{n}$ non-zeros in the standard basis, or in the Fourier basis.

Informally, a signal cannot be too localized in both the time and frequency domains simultaneously!

Proof: Since U and V are orthonormal we have that $\|b\|_2 = \|\alpha\|_2 = \|\beta\|_2$. We can rewrite b as either $U\alpha$ or $V\beta$ and hence $\|b\|_2^2 = |\beta^T(V^TU)\alpha|$. Because A is incoherent, we can conclude that each entry of V^TU has absolute value at most $\mu(A)$ and so $|\beta^T(V^TU)\alpha| \leq \mu(A)\|\alpha\|_1\|\beta\|_1$. Using Cauchy-Schwarz it follows that $\|\alpha\|_1 \leq \sqrt{\|\alpha\|_0}\|\alpha\|_2$ and thus

$$\|b\|_2^2 \leq \mu(A)\sqrt{\|\alpha\|_0\|\beta\|_0}\|\alpha\|_2\|\beta\|_2$$

Rearranging, we have $\frac{1}{\mu(A)} \leq \sqrt{\|\alpha\|_0\|\beta\|_0}$. Finally, applying the AM-GM inequality we get $\frac{2}{\mu} \leq \|\alpha\|_0 + \|\beta\|_0$ and this completes the proof. ■

Is this result tight? Indeed, returning to the spikes-and-sines example if choose b to be the comb function, where the signal has equally spaced spikes at distance $\sqrt{n}$, then b has $\sqrt{n}$ non-zeros in the standard basis. Moreover the comb function is its own discrete Fourier transform so it also has $\sqrt{n}$ non-zeros when represented using the Fourier basis.

Next, we apply the above uncertainty principle to prove a uniqueness result:

Claim 4.2.3 *Suppose $A = [U, V]$ where U and V are orthonormal and A is μ-incoherent. If $Ax = b$ and $\|x\|_0 < \frac{1}{\mu}$, then x is the uniquely sparsest solution.*

Proof: Consider any alternative solution $A\widetilde{x} = b$. Set $y = x - \widetilde{x}$ in which case $y \in \ker(A)$. Write y as $y = [\alpha_y, \beta_y]^T$ and since $Ay = 0$, we have that $U\alpha_y = -V\beta_y$. We can now apply the uncertainty principle and conclude that $\|y\|_0 = \|\alpha_y\|_0 + \|\beta_y\|_0 \geq \frac{2}{\mu}$. It is easy to see that $\|\widetilde{x}\|_0 \geq \|y\|_0 - \|x\|_0 > \frac{1}{\mu}$ and so $\widetilde{x}$ has strictly more non-zeros than x does, and this completes the proof. ■

Indeed, a similar statement is true even if A is an arbitrary incoherent matrix (instead of a union of two orthonormal bases). We will discuss this extension further in the next section.

Kruskal Rank

We can also work with a more general condition that is more powerful when proving uniqueness; however this condition is computationally hard to verify, unlike incoherence.

Definition 4.2.4 *The* Kruskal rank *of a set of vectors* $\{A_i\}_i$ *is the maximum* r *such that all subsets of* r *vectors are linearly independent.*

In fact, we have already proven that it is NP-hard to compute the Kruskal rank of a given set of points, since deciding whether or not the Kruskal rank is n is precisely the problem of deciding whether the points are in general position. Nevertheless, the Kruskal rank of A is the right parameter for analyzing how sparse x must be in order for it to be the uniquely sparest solution to $Ax = b$. Suppose the Kruskal rank of A is r.

Claim 4.2.5 *If* $\|x\|_0 \leq r/2$ *then* x *is the unique sparsest solution to* $Ax = b$.

Proof: Consider any alternative solution $A\widetilde{x} = b$. Again, we can write $y = x - \widetilde{x}$ in which case $y \in \ker(A)$. However $\|y\|_0 \geq r + 1$ because every set of r columns of A is linearly independent, by assumption. Then $\|\widetilde{x}\|_0 \geq \|y\|_0 - \|x\|_0 \geq r/2 + 1$ and so $\widetilde{x}$ has strictly more non-zeros than x does, and this completes the proof. ■

In fact, if A is incoherent we can lower bound its Kruskal rank (and so the proof in the previous section can be thought of as a special case of the one in this).

Claim 4.2.6 *If* A *is* μ*-incoherent then the Kruskal rank of the columns of* A *is at least* $1/\mu$.

Proof: First we note that if there is a set I of r columns of A that are linearly dependent, then the $I \times I$ submatrix of $A^T A$ must be singular. Hence it suffices to prove that every set I of size r, the $I \times I$ submatrix of $A^T A$ is full rank for $r = 1/\mu$.

So consider any such a submatrix. The diagonals are one, and the off-diagonals have absolute value at most μ by assumption. We can now apply Gershgorin's disk theorem and conclude that the eigenvalues of the submatrix are strictly greater than zero provided that $r \leq 1/\mu$ (which implies that the sum of the absolute values of the off-diagonals in any row is strictly less than one). This completes the proof. ■

Hence we can extend the uniqueness result in the previous section to arbitrary incoherent matrices (instead of just ones that are the union of two orthonormal bases). Note that this bound differs from our earlier bound by a factor of two.

Corollary 4.2.7 *Suppose* A *is* μ*-incoherent. If* $Ax = b$ *and* $\|x\|_0 < \frac{1}{2\mu}$, *then* x *is the uniquely sparsest solution.*

There are a number of algorithms that recover x up to the uniqueness threshold in the above corollary, and we will cover one such algorithm next.

4.3 Pursuit Algorithms

Here we will cover algorithms for solving sparse recovery when A is incoherent. The first such algorithm is *matching pursuit* and was introduced by Mallat and Zhang [93]; we will instead analyze *orthogonal matching pursuit* [99]:

Orthogonal Matching Pursuit
Input: matrix $A \in \mathbb{R}^{n\times m}$, vector $b \in \mathbb{R}^n$, desired number of nonzero entries $k \in \mathbb{N}$.
Output: solution x with at most k nonzero entries.

Initialize: $x^0 = 0$, $r^0 = Ax^0 - b$, $S = \emptyset$.
For $\ell = 1, 2, \dots, k$
  Choose column j that maximizes $\frac{|\langle A_j, r^{\ell-1}\rangle|}{\|A_j\|_2^2}$.
Add j to S.
  Set $r^\ell = \mathrm{proj}_{U^\perp}(b)$, where $U = \mathrm{span}(A_S)$.
  If $r^\ell = 0$, break.
End
Solve for x_S: $A_S x_S = b$. Set $x_{\bar{S}} = 0$.

Let A be μ-incoherent and suppose that there is a solution x with $k < 1/(2\mu)$ nonzero entries, and hence x is the uniquely sparsest solution to the linear system. Let $T = \mathrm{supp}(x)$. We will prove that orthogonal matching pursuit recovers the true solution x. Our analysis is based on establishing the following two invariants for our algorithm:

(a) Each index j the algorithm selects is in T .

(b) Each index j gets chosen at most once.

These two properties immediately imply that orthogonal matching pursuit recovers the true solution x, because the residual error r^ℓ will be non-zero until $S = T$, and moreover the linear system $A_T x_T = b$ has a unique solution (since otherwise x would not be the uniquely sparsest solution, which contradicts the uniqueness property that we proved in the previous section).

Property **(b)** is straightforward, because once $j \in S$ at every subsequent step in the algorithm we will have that $r^\ell \perp U$, where $U = \mathrm{span}(A_S)$, so $\langle r^\ell, A_j\rangle = 0$ if $j \in S$. Our main goal is to establish property **(a)**, which we will prove inductively. The main lemma is:

Lemma 4.3.1 *If $S \subseteq T$ at the start of a stage, then the algorithm selects $j \in T$.*

We will first prove a helper lemma:

Lemma 4.3.2 *If $r^{\ell-1}$ is supported in T at the start of a stage, then the algorithm selects $j \in T$.*

Proof: Let $r^{\ell-1} = \sum_{i\in T} y_i A_i$. Then we can reorder the columns of A so that the first k' columns correspond to the k' nonzero entries of y, in decreasing order of magnitude:

$$\underbrace{|y_1| \geq |y_2| \geq \cdots \geq |y_{k'}| > 0}_{\text{corresponds to first } k' \text{ columns of } A}, \quad |y_{k'+1}| = 0, |y_{k'+2}| = 0, \ldots, |y_m| = 0.$$

where $k' \leq k$. Hence $\text{supp}(y) = \{1, 2, \ldots, k'\} \subseteq T$. Then to ensure that we pick $j \in T$, a sufficient condition is that

$$|\langle A_1, r^{\ell-1}\rangle| > |\langle A_i, r^{\ell-1}\rangle| \qquad \text{for all } i \geq k'+1. \tag{4.1}$$

We can lower-bound the left-hand side of (4.1):

$$|\langle r^{\ell-1}, A_1\rangle| = \left|\left\langle \sum_{\ell=1}^{k'} y_\ell A_\ell, A_1 \right\rangle\right| \geq |y_1| - \sum_{\ell=2}^{k'} |y_\ell||\langle A_\ell, A_1\rangle| \geq |y_1| - |y_1|(k'-1)\mu \geq |y_1|(1-k'\mu+\mu),$$

which, under the assumption that $k' \leq k < 1/(2\mu)$, is strictly lower-bounded by $|y_1|(1/2+\mu)$.

We can then upper-bound the right-hand side of (4.1):

$$|\langle r^{\ell-1}, A_i\rangle| = \left|\left\langle \sum_{\ell=1}^{k'} y_\ell A_\ell, A_i \right\rangle\right| \leq |y_1| \sum_{\ell=1}^{k'} |\langle A_\ell, A_i\rangle| \leq |y_1|k'\mu,$$

which, under the assumption that $k' \leq k < 1/(2\mu)$, is strictly upper-bounded by $|y_1|/2$. Since $|y_1|(1/2+\mu) > |y_1|/2$, we conclude that condition (4.1) holds, guaranteeing that the algorithm selects $j \in T$ and this completes the proof of the lemma. ■

Now we can prove the main lemma:

Proof: Suppose that $S \subseteq T$ at the start of a stage. Then the residual $r^{\ell-1}$ is supported in T because we can write it as

$$r^{\ell-1} = b - \sum_{i\in S} z_i A_i, \text{ where } z = \arg\min \|b - A_S z_S\|_2$$

Applying the above lemma, we conclude that the algorithm selects $j \in T$. ■

This establishes property **(a)** inductively, and completes the proof of correctness for orthogonal matching pursuit. Note that this algorithm works up to exactly the threshold where we established uniqueness. However in the case where $A = [U, V]$ and U and V are orthogonal, we proved a uniqueness result that is better by a factor of two. There is no known algorithm that matches the best known uniqueness bound there, although there are better algorithms than the one above (see e.g. [55]).

Matching Pursuit

We note that matching pursuit differs from orthogonal matching pursuit in a crucial way: In the latter, we recompute the coefficients x_i for $i \in S$ at the end of each stage because we project b perpendicular to U. However we could hope that these coefficients do not need to be adjusted much when we add a new index j to S. Indeed, matching pursuit does not recompute these coefficients and hence is faster in practice, however the analysis is more involved because we need to keep track of how the error accumulates.

4.4 Prony's Method

The main result in this section is that any k-sparse signal can be recovered from just the first $2k$ values of its discrete Fourier transform, which has the added benefit that we can compute Ax quickly using the FFT. This is algorithm is called Prony's method, and dates back to 1795. This is optimal optimal relationship between the number of rows in A and the bound on the sparsity of x; however this method is very unstable since it involves inverting a Vandermonde matrix.

Properties of the DFT

In Prony's method, we will make crucial use of some of the properties of the DFT. Recall that DFT matrix has entries:

$$F_{a,b} = \left(\frac{1}{\sqrt{n}}\right) \exp\left(\frac{i2\pi(a-1)(b-1)}{n}\right)$$

We can write $\omega = e^{i2\pi/n}$, and then the first row is $\frac{1}{\sqrt{n}}[1, 1, \ldots, 1]$; the second row is $\frac{1}{\sqrt{n}}[1, \omega, \omega^2, \ldots]$, etc.

We will make use of following basic properties of F:

(a) F is orthonormal: $F^H F = FF^H$, where F^H is the Hermitian transpose of F

(b) F diagonalizes the convolution operator

In particular, we will define the convolution operation through its corresponding linear transformation:

Definition 4.4.1 (Circulant matrix) *For a vector $c = [c_1, c_2, \ldots, c_n]$, let*

$$M^c = \begin{bmatrix} c_n & c_{n-1} & c_{n-2} & \ldots & c_1 \\ c_1 & c_n & c_{n-1} & \ldots & c_2 \\ \vdots & & & & \vdots \\ c_{n-1} & \ldots & \ldots & \ldots & c_n \end{bmatrix}.$$

Then we can define $M^c x$ as the result of convolving c and x, denoted by $c * x$. It is easy to check that this coincides with the standard definition of convolution.

In fact, we can diagonalize M^c using F. We will use the following fact, without proof:

Claim 4.4.2 *$M^c = F^H diag(\widehat{c}) F$, where $\widehat{c} = Fc$.*

Hence we can think about convolution as coordinate-wise multiplication in the Fourier representation:

Corollary 4.4.3 *Let $z = c * x$; then $\widehat{z} = \widehat{c} \odot \widehat{x}$, where $\odot$ indicates coordinate-wise multiplication.*

Proof: We can write $z = M^c x = F^H \text{diag}(\widehat{c}) F x = F^H \text{diag}(\widehat{c}) \widehat{x} = F^H(\widehat{c} \odot \widehat{x})$, and this completes the proof. ■

We introduce the following helper polynomial, in order to describe Prony's method:

Definition 4.4.4 (Helper polynomial)

$$\begin{aligned} p(z) &= \prod_{b \in \text{supp}(x)} \omega^{-b}(\omega^b - z) \\ &= 1 + \lambda_1 z + \ldots + \lambda_k z^k. \end{aligned}$$

Claim 4.4.5 *If we know $p(z)$, we can find $supp(x)$.*

Proof: In fact, an index b is in the support of x if and only if $p(\omega^b) = 0$. So we can evaluate p at powers of ω, and the exponents where p evaluates to a non-zero are exactly the support of x. ■

The basic idea of Prony's method is to use the first $2k$ values of the discrete Fourier transform to find p, and hence the support of x. We can then solve a linear system to actually find the values of x.

Finding the Helper Polynomial

Our first goal is to find the Helper polynomial. Let

$$v = [1, \lambda_1, \lambda_2, \dots, \lambda_k, 0, \dots, 0], \text{ and } \widehat{v} = Fv$$

It is easy to see that the value of $\widehat{v}$ at index $b + 1$ is exactly $p(\omega^b)$.

Claim 4.4.6 $supp(\widehat{v}) = \overline{supp(x)}$

That is, the zeros of $\widehat{v}$ correspond roots of p, and hence non-zeros of x. Conversely, the non-zeros of $\widehat{v}$ correspond to zeros of x. We conclude that $x \odot \widehat{v} = 0$, and so:

Corollary 4.4.7 $M^{\widehat{x}} v = 0$

Proof: We can apply Claim 4.4.2 to rewrite $x \odot \widehat{v} = 0$ as $\widehat{x} * v = \widehat{0} = 0$, and this implies the corollary. ■

Let us write out this linear system explicitly:

$$M^{\widehat{x}} = \begin{bmatrix} \widehat{x}_n & \widehat{x}_{n-1} & \dots & \widehat{x}_{n-k} & \dots & \widehat{x}_1 \\ \widehat{x}_1 & \widehat{x}_n & \dots & \widehat{x}_{n-k+1} & \dots & \widehat{x}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \widehat{x}_{k+1} & \widehat{x}_k & \dots & \widehat{x}_1 & \dots & \widehat{x}_{k+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \widehat{x}_{2k} & \widehat{x}_{2k-1} & \dots & \widehat{x}_k & \dots & \widehat{x}_{2k+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Recall, we do not have access to all of the entries of this matrix since we are only given the first $2k$ values of the DFT of x. However consider the $k \times k+1$ submatrix whose top left value is $\widehat{x}_{k+1}$ and whose bottom right value is $\widehat{x}_k$. This matrix only involves the values that we do know!

Consider

$$\begin{bmatrix} \widehat{x}_k & \widehat{x}_{k-1} & \dots & \widehat{x}_1 \\ \vdots & & & \\ \widehat{x}_{2k-1} & \widehat{x}_{2k-1} & \dots & \widehat{x}_k \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{bmatrix} = - \begin{bmatrix} \widehat{x}_{k+1} \\ \vdots \\ \vdots \\ \widehat{x}_{2k} \end{bmatrix}$$

It turns out that this linear system is full rank, so λ is the unique solution to the linear system (the proof is left to the reader[1]). The entries in λ are the coefficients of p, so once we have solved for λ we can evaluate the helper polynomial on ω^b to find the support of x. All that remains is to find the values of x. Indeed, let M be the restriction of F to the columns in S and its first $2k$ rows. M is a Vandermonde matrix, so again $Mx_S = \widehat{x}_{1,2,\ldots 2k}$ has a unique solution, and we can solve this linear system to find the non-zero values of x.

4.5 Compressed Sensing

Here we will give *stable* algorithms for recovering a signal x that has an almost linear (in the number of rows of the sensing matrix) number of non-zeros. Recall that the Kruskal rank of the columns of A is what determines how many non-zeros we can allow in x and yet have x be the uniquely sparsest solution to $Ax = b$. A random matrix has large Kruskal rank, and what we will need for compressed sensing is a robust analogue of Kruskal rank:

Definition 4.5.1 *A matrix A is RIP with constant δ_k if for all k-sparse vectors x we have:*

$$(1-\delta_k)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\delta_k)\|x\|_2^2$$

If A is a random $m \times n$ matrix where each entry is an independent Gaussian ($N(0,1)$) then we can choose $m \approx k \log n/k$ and $\delta_k = 1/3$. Next we formalize the goals of sparse recovery:

Definition 4.5.2 $\sigma_k(x) = \min_{w \ s.t. \ \|w\|_0 \le k} \|x - w\|_1$

i.e. $\sigma_k(x)$ is the ℓ_1 sum of all but the k largest entries of x. In particular, if $\|x\|_0 \le k$ then $\sigma_k(x) = 0$.

Our goal is to find a w where $\|x - w\|_1 \leq C\sigma_k(x)$ from a few ($\tilde{O}(k)$) measurements. Note that we will note require that w is k sparse. However if x is exactly k sparse, then any w satisfying the above condition must be exactly equal to x and hence this new recovery goal subsumes our exact recovery goals from previous lectures (and is indeed much stronger).

The natural (but intractable) approach is:

$$(P0) \qquad \min \|w\|_0 \text{ s.t. } Aw = b$$

Since this is computationally hard to solve (for all A) we will work with the ℓ_1 relaxation:

$$(P1) \qquad \min \|w\|_1 \text{ s.t. } Aw = b$$

and we will prove conditions under which the solution w to this optimization problem satisfies $w = x$ (or $\|x - w\|_1 \leq C\sigma_k(x)$).

Theorem 4.5.3 *[35] If $\delta_{2k} + \delta_{3k} < 1$ then if $\|x\|_0 \leq k$ we have $w = x$.*

Theorem 4.5.4 *[34] If $\delta_{3k} + 3\delta_{4k} < 2$ then*

$$\|x - w\|_2 \leq \frac{C}{\sqrt{k}}\sigma_k(x)$$

Note that this bounds the ℓ_2 norm of the error $x - w$ in terms of the ℓ_1 error of the best approximation.

Theorem 4.5.5 *[40] If $\delta_{2k} < 1/3$ then*

$$\|x - w\|_1 \leq \frac{2 + 2\delta_{2k}}{1 - 3\delta_{2k}}\sigma_k(x)$$

We will follow the proof of [80] that greatly stream-lined the types of analysis and made the connection between compressed sensing and *almost Euclidean subsections* explicit. From this viewpoint it will be much easier to draw an analogy with error correcting codes.

Almost Euclidean Subsections

Set $\Gamma = \ker(A)$. We will make use of certain geometric properties of Γ (that hold almost surely) in order to prove that basis pursuit works:

Definition 4.5.6 *$\Gamma \subseteq \mathbb{R}^n$ is an almost Euclidean subsection if for all $v \in \Gamma$,*

$$\frac{1}{\sqrt{n}}\|v\|_1 \leq \|v\|_2 \leq \frac{C}{\sqrt{n}}\|v\|_1$$

Note that the inequality $\frac{1}{\sqrt{n}}\|v\|_1 \leq \|v\|_2$ holds for all vectors, hence the second inequality is the important part. What we are requiring is that the ℓ_1 and ℓ_2 norms are almost equivalent after rescaling.

Question 8 *If a vector v has $\|v\|_0 = o(n)$ then can v be in Γ?*

No! Any such vector v would have $\|v\|_1 = o(\sqrt{n})\|v\|_2$ using Cauchy-Schwartz.

Let us think about these subsections geometrically. Consider the unit ball for the ℓ_1 norm:

$$B_1 = \{v | \|v\|_1 \leq 1\}$$

This is called the cross polytope and is the convex hull of the vectors $\{\pm e_i\}_i$ where e_i are the standard basis vectors. Then Γ is a subspace which when intersected with B_1 results in a convex body that is close to the sphere B_2 after rescaling.

In fact it has been known since the work of [63] that choosing Γ uniformly at random with $\dim(\Gamma) \geq n - m$ we can choose $C = O(\sqrt{\log n/m})$ almost surely (in which case it is the kernel of an $m \times n$ matrix A, which will be our sensing matrix). In the remainder of the lecture, we will establish various geometric properties of Γ that will set the stage for compressed sensing.

Properties of Γ

Throughout this section, let $S = n/C^2$.

Claim 4.5.7 *Let $v \in \Gamma$, then either $v = 0$ or $|supp(v)| \geq S$.*

Proof:

$$\|v\|_1 = \sum_{j \in \text{supp}(v)} |v_j| \leq \sqrt{|\text{supp}(v)|} \cdot \|v\|_2 \leq \sqrt{|\text{supp}(v)|}\frac{C}{\sqrt{n}}\|v\|_1$$

where the last inequality uses the property that Γ is almost Euclidean. The last inequality implies the claim. ■

Now we can draw an analogy with error correcting codes. Recall that here we want $\mathcal{C} \subseteq \{0,1\}^n$. And the rate R is $R = \log |\mathcal{C}|/n$ and the relative distance δ is

$$\delta = \frac{\min_{x \neq y \in \mathcal{C}} d_H(x,y)}{n}$$

where d_H is the Hamming distance. The goal is to find a code where $R, \delta = \Omega(1)$ and that are easy to encode and decode. In the special case of linear codes, e.g. $\mathcal{C} = \{y | y = Ax\}$ where A is an $n \times Rn$ $\{0,1\}$-valued matrix and $x \in \{0,1\}^{Rn}$. Then

$$\delta = \frac{\min_{x \neq 0 \in \mathcal{C}} \|x\|_0}{n}$$

So for error correcting codes we want to find large (linear) dimensional subspaces where each vector has a linear number of non-zeros. In compressed sensing we want Γ to have this property too, but moreover we want that its ℓ_1 norm is also equally spread out (e.g. most of the non-zero coordinates are large).

Definition 4.5.8 *For $\Lambda \subseteq [n]$, let v_Λ denote the restriction of v to coordinates in Λ. Similarly let v^Λ denote the restriction of v to $\bar{\Lambda}$.*

Claim 4.5.9 *Suppose $v \in \Gamma$ and $\Lambda \subseteq [n]$ and $|\Lambda| < S/16$. Then*

$$\|v_\Lambda\|_1 < \frac{\|v\|_1}{4}$$

Proof:

$$\|v_\Lambda\|_1 \leq \sqrt{|\Lambda|}\|v_\Lambda\|_2 \leq \sqrt{|\Lambda|}\frac{C}{\sqrt{n}}\|v\|_1$$

■

Hence not only do vectors in Γ have a linear number of non-zeros, but in fact their ℓ_1 norm is spread out. Now we are ready to prove that $(P1)$ exactly recovers x when $\|x\|_0$ is sufficiently small (but nearly linear). Next lecture we will prove that it is also stable (using the properties we have established for Γ above).

Lemma 4.5.10 *Let $w = x + v$ and $v \in \Gamma$ where $\|x\|_0 \leq S/16$. Then $\|w\|_1 > \|x\|_1$.*

Proof: Set $\Lambda = \text{supp}(x)$.

$$\|w\|_1 = \|(x+v)_\Lambda\|_1 + \|(x+v)^\Lambda\|_1 = \|(x+v)_\Lambda\|_1 + \|v^\Lambda\|_1$$

Now we can invoke triangle inequality:

$$\|w\|_1 \geq \|x_\Lambda\|_1 - \|v_\Lambda\|_1 + \|v^\Lambda\|_1 = \|x\|_1 - \|v_\Lambda\|_1 + \|v^\Lambda\|_1 = x_\Lambda\|_1 - 2\|v_\Lambda\|_1 + \|v\|_1$$

However $\|v\|_1 - 2\|v_\Lambda\|_1 \geq \|v\|_1/2 > 0$ using the above claim. This implies the lemma. ■

Hence we can use almost Euclidean subsections to get exact sparse recovery up to

$$\|x\|_0 = S/16 = \Omega(n/C^2) = \Omega\Big(\frac{n}{\log n/m}\Big)$$

Next we will consider stable recovery. Our main theorem is:

Theorem 4.5.11 *Let $\Gamma = ker(A)$ be an almost Euclidean subspace with parameter C. Let $S = \frac{n}{C^2}$. If $Ax = Aw = b$ and $\|w\|_1 \leq \|x\|_1$ we have*

$$\|x - w\|_1 \leq 4\,\sigma_{\frac{S}{16}}(x)\,.$$

Proof: Let $\Lambda \subseteq [n]$ be the set of $S/16$ coordinates maximizing $\|x_\Lambda\|_1$. We want to upper bound $\|x - w\|_1$. By the repeated application of the triangle inequality, $\|w\|_1 = \|w^\Lambda\|_1 + \|w_\Lambda\|_1 \leq \|x\|_1$ and the definition of $\sigma_t(\cdot)$, it follows that

$$\begin{aligned}\|x - w\|_1 &= \|(x-w)_\Lambda\|_1 + \|(x-w)^\Lambda\|_1 \\ &\leq \|(x-w)_\Lambda\|_1 + \|x^\Lambda\|_1 + \|w^\Lambda\|_1 \\ &\leq \|(x-w)_\Lambda\|_1 + \|x^\Lambda\|_1 + \|x\|_1 - \|w_\Lambda\|_1 \\ &\leq 2\|(x-w)_\Lambda\|_1 + 2\|x^\Lambda\|_1 \\ &\leq 2\|(x-w)_\Lambda\|_1 + 2\,\sigma_{\frac{S}{16}}(x)\,.\end{aligned}$$

Since $(x - w) \in \Gamma$, we can apply Claim 4.5.9 to conclude that $\|(x - w)_\Lambda\|_1 \leq \frac{1}{4}\|x - w\|_1$. Hence

$$\|x - w\|_1 \leq \frac{1}{2}\|x - w\|_1 + 2\sigma_{\frac{S}{16}}(x)\,.$$

This completes the proof. ■

Notice that in the above argument, it was the geometric properties of Γ which played the main role. There are a number of proofs that basis pursuit works, but the advantage of the one we presented here is that it clarifies the connection between the classical theory of error correction over the finite fields, and the sparse recovery. The matrix A here plays the role *parity check matrix* of error correcting code, and hence its kernel corresponds to the *codewords*. There is more subtlety in the real case though: as opposed to the finite field setting where the Hamming distance is essentially the only reasonable way of measuring the magnitude of errors, in the real case there is an interplay among many different norms, giving rise to some phenomenon not present in the finite field case.

In fact, one of the central open question of the field is to give a *deterministic* construction of RIP matrices:

Question 9 (Open) *Is there an explicit construction of RIP matrices, or equivalently an almost Euclidean subsection* Γ*?*

In contrast, there are many explicit constructions of asymptotically good codes. The best known deterministic construction is due to Guruswami, Lee and Razborov:

Theorem 4.5.12 *[69] There is a polynomial time deterministic algorithm for constructing an almost Euclidean subspace* Γ *with parameter* $C \sim (\log n)^{\log \log \log n}$

We note that it is easy to achieve weaker guarantees, such as $\forall 0 \neq v \in \Gamma$, $\mathrm{supp}(v) = \Omega(n)$, but these do not suffice for compressed sensing since we also require that the ℓ_1 weight is spread out too.