EE5585 Data Compression April 9, 2013

Lecture 20

Instructor: Arya Mazumdar *Scribe: Aritra Konar*

1 Review

In the last class, we studied about Transform Coding, where we try to assign a new set of basis vectors to the data. If the data exhibits some form of correlation, we apply a linear unitary transformation to rotate the axes such that most of the data fits through one of the axes. The result is that the data is sparse, and along this axis. Meaning we can throw away the entries that have values near zero, thereby compressing the data. However, this approach suffers from two drawbacks.

i) It may happen that the data cannot be fit through a single (or a smaller set of) axis. In this case the linear transforms (using DFT matrix or DCT matrix for example) will not be able to compress the data. This is shown in the diagram below, where the data is fitted by a nonlinear function.

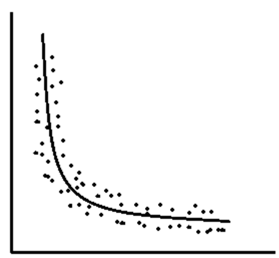

Figure 1: Example of data fitting by a non - linear function

ii) In this approach, the the data is acquired from a sensor array, following which a basis is determined in which the representation of the data is sparse, and then the sparse entries are thrown away to achieve data compression. Since a lot of the data samples obtained are thrown away, we may then seek an approach where we can compress the data by sensing only a small number of samples.

2 Compressed Sensing/Sampling

Suppose we have N pixels of an image we want to compress. We may capture the pixels using an array of N sensors to obtain a data vector, to which we then apply a linear unitary transformation (by multiplying it with the DFT matrix for example) to obtain a sparse representation of the data. We can then throw away the small entries to compress the data.

Alternatively, we may only use M ($\ll$N) sensors to capture the pixels. The diagram below illustrates this approach.

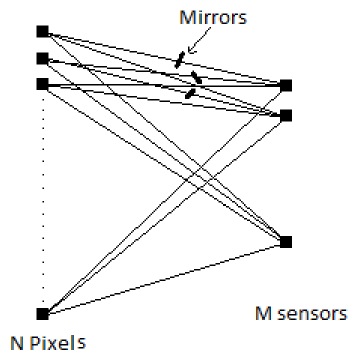

Figure 2: Single Pixel Camera

Here, each sensor senses a linear combination of N pixel values. The output of each sensor is given by

$$y_i = a_{1i}z_1 + a_{2i}z_2 + ... + a_{Ni}z_N$$

where the coefficients of the linear combination are determined from the reflection coefficients of the mirrors placed at an angle θ

Hence, the output of the sensor array consists of M linear samples drawn from z. The output vector y may be represented as

$$y = Az$$

where $A \to M \times N$, $y \to M \times 1$, $z \to N \times 1$

We observe that dimensionality reduction is achieved by mapping the $N \times 1$ vector of pixels into a $M \times 1$ vector of observations, since $M \ll N$.

Now, z has a property. We take the DFT matrix and use it to transform z into a sparse vector having only $k \ll N$ non - zero entires (i.e, z is compressible).

The next problem we face is how to recover z from the compressed data vector y. If the matrix A were square, full rank and hence invertible, we could have easily recovered z from y by the relation

$$z = A^{-1}y$$

However, since A is a fat matrix, we have an underdetermined system of linear equations with an infinte number of solutions. We will try to use the sparsity property of z(in some domain) to recover z given A, y.

Decompose

$$A = \Phi F$$

where $A \to M \times N$, $\Phi \to M \times N$, $F \to N \times N$ (Fourier Matrix in some domain where z has a sparse representation).

This is a valid matrix decomposition since F is an invertible matrix $\Longrightarrow \Phi = AF^{-1}$

Hence, we have $y = Az = \Phi Fz = \Phi x$ where $x = Fz$. x is an $N \times 1$vector having only $k \ll N$ non - zero values. Thus, in other words, x is k sparse.

We have now reduced the problem to the one of recovering x from the under-determined system of equations $y = \Phi x$ where x is k sparse. The matrix Φ is to be designed such that unambiguous recovery of x is possible. This happens if and only if, for any two k sparse vectors x_1 and x_2

$$\Phi x_1 \neq \Phi x_2$$

$$\Rightarrow \Phi(x_1 - x_2) \neq 0$$

This statement implies that any two k sparse vectors should not get mapped to the same obeservation vector y. Otherwise, unique recovery will not be possible. Since $x_1 - x_2$ is atmost $2k$ sparse, the left hand side of the above expression is the weighted sum of any $2k$ columns of the matrix Φ. Since the linear combination of the columns $\neq 0$, it follows that any $2k$ columns of Φ must be linearly independent.

$$\Rightarrow M \geq 2k$$

The above expression implies that for stable recovery, the number of samples required for compressive sensing must be greater than equal to twice the sparsity of the data vector x. Note that this statement is somewhat analogous to the Nyquist Sampling Theorem.

Also, since we do not know the locations of the k sparse samples in the $N \times 1$data vector x, we must use $M \geq 2k$ samples. If we knew the exact locations of the sparse entries, then we would require only k columns of Φ to be linearly independent, implying that only $M= k$ samples would be sufficient for stable recovery. Hence, we see that we must pay a penalty for not knowing the locations of the sparse entries of x by requiring a greater number of samples.

Our ultimate goal, now, is to determine a matrix Φ that is $2k \times N$ dimensional and any $2k$ columns are linearly independent.

Let us consider an example of signal recovery where x is 1 sparse 6×1 vector and Φ is given by

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

If the observation vector $y = \Phi x$ is also known, then we can easily recover x. We know that y is obtained from a linear combination of the columns of Φ with the coefficients of the combination being the elements of the vector x. Since x is 1 sparse, with only a single non - zero entry, then y is simply the scaled column of Φ whose coefficient is the non - zero entry of x. Hence, knowing y, we can determine the non -zero entry of x and from the position of the corresponding column in Φ, we can determine the position of the non - zero entry in x.

Note however, in this case, we used $M = 3 > 2$ samples in the sampling matrix Φ. In general, if we wanted to use $M = 2k$ samples, we could use a *Vandermonde matrix* given by

$$\Phi = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha_1 & \alpha_2 & . & . & . & \alpha_N \\ \alpha_{1^2} & \alpha_{2^2} & . & . & . & \alpha_{N^2} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \alpha_{1^{N-1}} & \alpha_{2^{N-1}} & . & . & . & \alpha_{N^{N-1}} \end{bmatrix}$$

where $\alpha_i \in \mathbb{R}$, such that $\alpha_i \neq \alpha_j \ \forall i \neq j$

We can choose any $M \times M = 2k \times 2k$ submatrix of Φ which will be non - singular, full rank with m linearly independent columns.

$$\Psi = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha_{i_1} & \alpha_{i_2} & . & . & . & \alpha_{i_M} \\ \alpha_{i_1}^2 & \alpha_{i_2}^2 & . & . & . & \alpha_{i_M}^2 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \alpha_{i_1}^{M-1} & \alpha_{i_2}^{M-1} & . & . & . & \alpha_{i_M}^{M-1} \end{bmatrix}$$

where we choose $i_1, i_2, ..., i_M$from the matrix Φ

Let $\alpha_{i_j} = \beta_j$. Then,

$$\Psi = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \beta_1 & \beta_2 & . & . & . & \beta_M \\ \beta_1^2 & \beta_2^2 & . & . & . & \beta_M^2 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \beta_1^{M-1} & \beta_2^{M-1} & . & . & . & \beta_M^{M-1} \end{bmatrix}$$

Ψ is a square Vandermonde matrix with

$$det\Psi = \prod_{1 \leq j < i \leq m} (\beta_i - \beta_j)$$

Since $\beta_i \neq \beta_j$ for $i \neq j$,Ψ is guaranteed to be non singular, full rank with M= $2k$ linearly independent columns.

Thus, the Vandermonde matrix Φ can be used to recover x from the observation y. However, the entire success of this scheme hinges on the assumption made at the beginning that x is k sparse. In reality, x may only be approximately k sparse, with k prominent values and another $N - k$ values that are close to, but not zero. If this is the case, then this scheme fails, because only $M = 2k$ columns are not enough to guarantee recovery. The contributions from the other columns of will be non - zero in this case since x is not exactly k sparse. We now seek a scheme that guarentees stable recovery even if x is approximately k sparse.

Suppose $\hat{x}$ is our estimate of x. The error ε is given by

$$\varepsilon = \| \hat{x} - x \|_{l_2}^2 = (\hat{x} - x)^T (\hat{x} - x)$$

Now, we would like to bound this error even if x is not exactly k sparse.

Suppose x_k is a vector that has the k largest co-ordinates of x with all others zero. Then, we may write an approximately k sparse vector x in terms of x_k as

$$x = x_k + \varepsilon$$

$$\Longrightarrow \varepsilon = x - x_k$$

where x is an exact k sparse vector and ε is the error in x not being exactly k sparse.

We would like a recovery guarantee of the form

$$\| \hat{x} - x \|_{l_2} \leq c \| x - x_k \|_{l_2} \text{ ....}(\nabla)$$

The above statement implies that when x is exactly k sparse, then $x_k = x$ and the right hand side of the above equation is zero, from which we get $\hat{x} = x$. In this case, perfect recovery is achieved. However, when x is only approximately k sparse, even then $\hat{x}$ is not too far from x! We are now ready to state a theorem which gives the recovery guarantee given by (∇) subject to some conditions.

3 Theorem

Recovery with guarantee given by (∇) will require $M = c_1 k \log(N/k)$ samples. Then, there exists a $M \times N$ sampling matrix Φ with $M = c_1 k \log(N/k)$ which gives guarantee

$$\| \hat{x} - x \|_{l_2} \leq \frac{c}{\sqrt{k}} \| x - x_k \|_{l_1} \text{ ....(\$)}$$

Note that, since we are dividing by $\sqrt{k}$, the error bound obtained is quite tight. This scheme guarantees stable recovery even when x is only approxumately k sparse. We can determine $\hat{x}$ which satisfies the above error bound by a linear program called *Basis Pursuit* which is a polynomial time algorithm.

3.1 Basis Pursuit

Given $y = \Phi x$ where x is approximately k sparse, then we can obtain $\hat{x}$ which gives guarantee ($) by solving the following optimization problem with a linear constraint

$$\min_{s.t. \Phi\hat{x}=y} \parallel \hat{x} \parallel_{l_1}$$

3.2 Fun Facts about Norms

If q is an integer, then the L^q norm of a $N \times 1$ vector x is given by

$$\parallel x \parallel_{l_q} = (\sum_{i=1}^{N} \mid x_i \mid^q)^{\frac{1}{q}}$$

When $q \to \infty$, $L^\infty = max \mid x_i \mid$
When $q \to 0$, L^0 =number of non zero entries of x

4 Restricted Isometry Property

A sampling matrix $\Phi$is said to possess δ_k RIP, if, for any k sparse vector v,

$$(1 - \delta_k) \| v \parallel_2^2 \leq \parallel \Phi v \parallel_2^2 \leq (1 + \delta_k) \parallel v \parallel_2^2$$

We choose the smallest δ_k such that the above expression is true. Note that, if δ_k is 0, then

$$\parallel v \parallel_2^2 = \parallel \Phi v \parallel_2^2$$

Thus, Φ is an orthogonal transform that preserves the norm (energy) of the k sparse vector v.However, even if $\delta_k \neq 0$, even then Φ is approximately orthogonal since the energy in the signal is only changed by a factor δ_k. In 2005, Candes and Tao showed that if

$$\delta_{2k} \leq \sqrt{2} - 1$$

That is, if for any $2k$ sparse vector the energy is preserved by a factor of $\sqrt{2} - 1 = 0.414$,then, $\hat{x}$ obtained from Basis Pursuit will give guarantee

$$\parallel \hat{x} - x \parallel_{l_2} \leq \frac{c}{\sqrt{k}} \parallel x - x_k \parallel_{l_1}$$

Later on, it will be shown that if Φ is a random matrix with iid Gausssian or Bernoulli random variables as it's entries, then with very high probability, Φ satisfies RIP.