EE5585 Data Compression April 16, 2013

Lecture 22

Instructor: Arya Mazumdar *Scribe: Cheng-Yu Hung*

Review of Compressed Sensing

Consider a general linear measurement process that produces an $m \times 1$ observation vector $\mathbf{y}$ in terms of a $N \times 1$ signal vector $\mathbf{x}$ and the columns of a $m \times N (m \ll N)$ measurement matrix Φ. It can be expressed as

$$\Phi \mathbf{x} = \mathbf{y} \tag{1}$$

It is noted that the signal vector $\mathbf{x}$ is an approximately k-sparse vector which has at most $k (\ll N)$ non-zero entries.

Basic Pursuit

Given $\mathbf{y}$ and Φ, recovery of the unknown signal vector $\mathbf{x}$ can be pursued by finding the sparsest estimate of $\mathbf{x}$ which has the constraint $\Phi \mathbf{x} = \mathbf{y}$, i.e.,

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & ||\mathbf{x}||_{l_0} \\ \text{subject to} \quad & \Phi \mathbf{x} = \mathbf{y}. \end{aligned}$$

However, this is an NP-hard problem. Convex relaxation methods cope with the intractability of the above formulation by approximating the l_0 norm by the convex l_1 norm. This is a well-known algorithm which is called basic pursuit. The above formulation is changed to

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & ||\mathbf{x}||_{l_1} \\ \text{subject to} \quad & \Phi \mathbf{x} = \mathbf{y}. \end{aligned} \tag{2}$$

Many linear programmings can solve this problem, such as the simplex method or interior point methods.

Stable Recovery and Restricted Isometry Property (RIP)

The RIP imposes that there exists a $0 < \delta_{2k} < 1$ such that for any $\mathbf{z}$ that has at most $2k$ nonzero entries,

$$(1 - \delta_{2k})||\mathbf{z}||_{l_2}^2 \leq ||\Phi \mathbf{z}||_{l_2}^2 \leq (1 + \delta_{2k})||\mathbf{z}||_{l_2}^2 \tag{3}$$

Candes et al. (2006) have shown that if Φ satisfies RIP with $\delta_{2k} \leq \sqrt{2} - 1$, then the solution $\hat{\mathbf{x}}$ to equation (2) will achieve the stable recovery,

$$||\hat{\mathbf{x}} - \mathbf{x}||_{l_2}^2 \leq \frac{c}{\sqrt{k}} ||\hat{\mathbf{x}} - \mathbf{x}_k||_{l_1}^2, \tag{4}$$

where $\mathbf{x}_k$ is the restriction of $\mathbf{x}$ to its k largest entries.

Design Good Measurement Matrices Φ

One of main issues is to find a matrix Φ with RIP: $\delta_{2k} < \sqrt{2} - 1$. Given the $N \times 1$ vector $\mathbf{x}$ and $m \times N (m \ll N)$ measurement matrix Φ, we denote a $m \times 2K$ matrix Φ_I, and a $2k \times 1$ vector $\mathbf{z} = \mathbf{x}_I$, where $I \subseteq \{1, \cdots, N\}$, $|I| = 2k$. Since $\delta_{2k} < \sqrt{2} - 1$, the RIP becomes as the following:

$$0.586 = (1 - \delta_{2k}) \leq \frac{||\Phi \mathbf{z}||_{l_2}^2}{||\mathbf{z}||_{l_2}^2} \leq (1 + \delta_{2k}) = 1.414. \tag{5}$$

We want any $2k$ columns of Φ satisfying the above inequalities (5). Actually, (5) is also equivalent to

$$0.586 \leq \text{eigenvalues of } \Phi_I^T \Phi_I \leq 1.414. \tag{6}$$

or

$$0.586 \leq \text{sigular values of } \Phi_I \leq 1.414. \tag{7}$$

But the question is what kind of matrix Φ has this property of (7)?
The answer is that the matrix Φ can be a random matrix chosen in the following way:

$$\Phi = \frac{1}{\sqrt{m}} \begin{bmatrix} \phi_{11} & \cdots & \phi_{1N} \\ \vdots & \ddots & \vdots \\ \phi_{m1} & \cdots & \phi_{mN} \end{bmatrix}, \text{where } \phi_{ij} \overset{iid}{\sim} \mathcal{N}(0,1). \tag{8}$$

with high probability for any I such that Φ_I satisfies

$$0.586 \leq \text{eigenvalues of } \Phi_I^T \Phi_I \leq 1.414.$$

In particular the high probability is $1 - e^{-\alpha m}$, for some $\alpha > 0$.
Then,

$$\begin{aligned}
&\Pr(\text{RIP-2k isn't satisfied}) && (9)\\
&= \Pr(\exists \text{ a set of 2k columns for which (7) is not satisfied}) && (10)\\
&\leq \binom{N}{2k} e^{-\alpha m} && (11)\\
&= \left(\frac{Ne}{2k}\right)^{2k} e^{-\alpha m} && (12)\\
&= e^{2k \log\left(\frac{Ne}{2k}\right) - \alpha m} && (13)\\
&\text{if } \alpha m \geq 2 \cdot 2k \log\left(\frac{Ne}{2k}\right) && (14)\\
&\leq e^{\frac{-\alpha m}{2}} \to 0 \text{ as } m \to \infty && (15)
\end{aligned}$$

In summary, suppose that the entries of the $m \times 2k$ matrix Φ_I are i.i.d. Gaussian with zero mean and variance $\frac{1}{m}$. Then, with high probability, Φ satisfies the required RIP condition for stable recovery to hold, provided that

$$m > \frac{4}{\alpha} k \log\left(\frac{Ne}{2k}\right). \tag{16}$$

Moreover, sensing matrices whose entries are iid from a Bernoulli distribution ($+\frac{1}{\sqrt{m}}$ with prob. 1/2 and $-\frac{1}{\sqrt{m}}$ with prob. 1/2), columns normalized to a unit norm, also obey the RIP given equation (16). It is noted that since $m > \frac{4}{\alpha} k \log\left(\frac{Ne}{2k}\right)$, the observation signal's dimension is $m = \mathcal{O}(k \log \frac{N}{k})$, while the dimension of input signal is N.

Differential Encoding

In many sources we are interested in, the sampled source output $\{x_n\}$ does not change a lot from one sample to the next. This means that the variance of the sequence of difference $\{d_n = x_n - x_{n-1}\}$ are

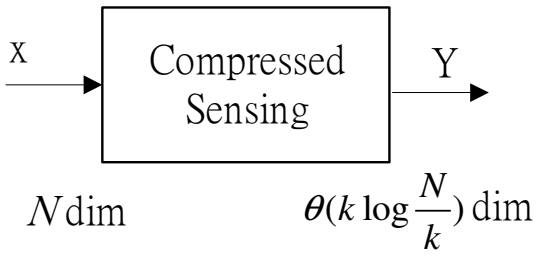

Figure 1: Compressed sensing

substantially smaller than that of the source output sequence. In other words, for the correlated data the distribution of d_n is highly peaked at zero. It is useful to encode the difference from the sample to the next rather than encoding the actual sample values since the variance of quantization error in the differences is less than in the actula output samples. This technque is called *differential encoding*.

Quantize with Differential Encoding

Consider a sequence $\{x_n\}$ in Figure 2. A difference sequence $\{d_n\}$ is generated by taking the difference $x_n - x_{n-1}$. The difference sequence is quantized to obtain the sequence $\{\hat{d}_n\}$. So,

$$\begin{aligned} d_0 &= x_0 && (17)\\ d_1 &= x_1 - x_0 && (18)\\ d_2 &= x_2 - x_1 && (19)\\ &\vdots && (20)\\ d_n &= x_n - x_{n-1}. && (21) \end{aligned}$$

The above equations can be written as a matrix form, i.e.,

$$\begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & & \ddots & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_0 \\ \vdots \\ d_n \end{bmatrix} \text{ or } \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ 1 & 1 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \begin{bmatrix} d_0 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix} \quad (22)$$

And

$$\hat{d}_n = Q[d_n] = d_n + q_n, \text{ where } q_n \text{ is the quantization error.} \quad (23)$$

At the receiver, the reconstructed sequence $\{\hat{x}_n\}$ is obtained by adding $\hat{d}_n$ to the previous reconstructed value $\hat{x}_{n-1}$:

$$\hat{x}_n = \hat{x}_{n-1} + \hat{d}_n. \quad (24)$$

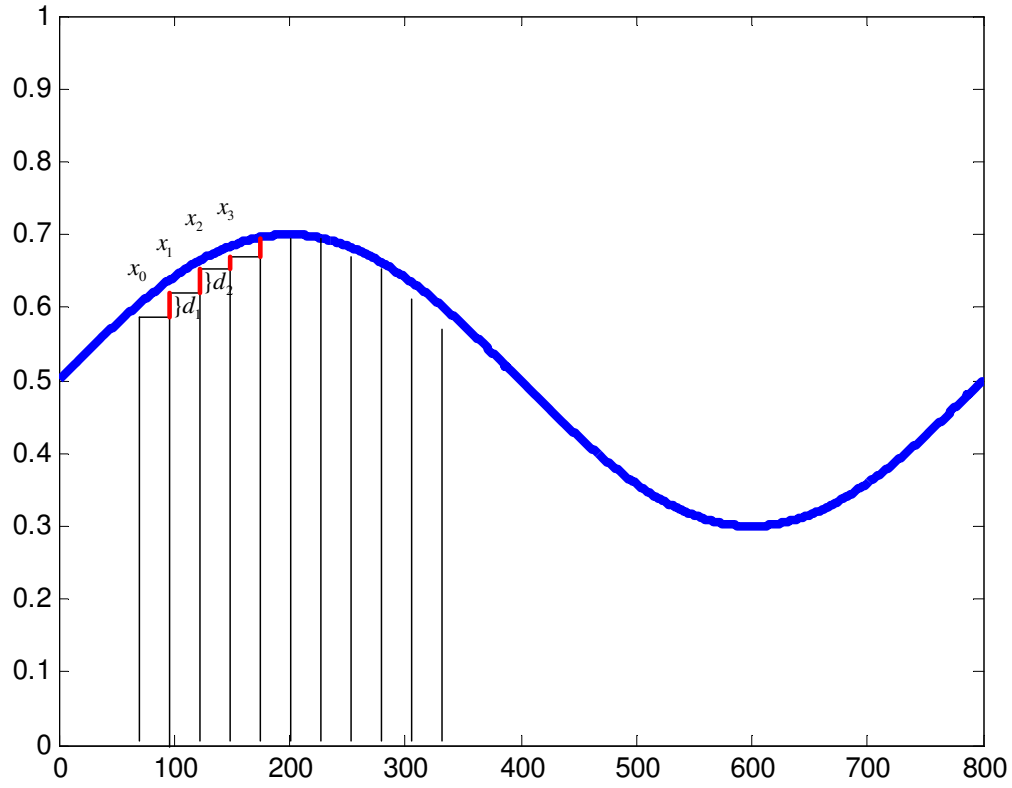

Figure 2: Output samples $\{x_n\}$ and difference samples $\{d_n\}$

Let us assume that both transmitter ans receiver start with the same value x_0, that is, $\hat{x}_0 = x_0$. Then, follow the quantization and reconstruction process:

$$\hat{x}_0 = d_0 + q_0 \tag{25}$$

$$\hat{x}_1 = \hat{x}_0 + \hat{d}_1 = d_0 + q_0 + d_1 + q_1 = x_1 + q_0 + q_1 \tag{26}$$

$$\hat{x}_2 = \hat{x}_1 + \hat{d}_2 = d_0 + d_1 + q_0 + q_1 + d_2 + q_2 = x_2 + q_0 + q_1 + q_2 \tag{27}$$

$$\vdots \tag{28}$$

$$\hat{x}_n = d_0 + d_1 + \cdots + d_n + q_0 + q_1 + \cdots + q_n = x_n + \sum_{i=0}^{n} q_i \tag{29}$$

So, at the ith iteration we get

$$\hat{x}_i = x_i + \sum_{j=0}^{i} q_j. \tag{30}$$

We can see that the quantization error accumulates as the process continues. According to central limit

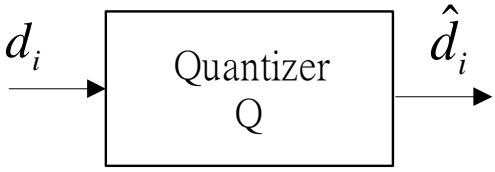

Figure 3: Quantizer of differential encoding

theorem, the sum of these quantization error is Gaussian noise and its mean is zero, but in fact, before that happens, the finite precision of machines causes the reconstructed value to overflow.
Therefore, instead of $d_n = x_n - x_{n-1}$, we use $d_n = x_n - \hat{x}_{n-1}$. By use of this new differencing operation, we repeat the quantization and reconstruction process. Assume that $\hat{x}_0 = x_0$,

$$\begin{aligned} d_1 &= x_1 - x_0 && (31)\\ \hat{d}_1 &= d_1 + q_1 && (32)\\ \hat{x}_1 &= x_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_1 + q_1 && (33)\\ d_2 &= x_2 - \hat{x}_1 && (34)\\ \hat{d}_2 &= d_2 + q_2 && (35)\\ \hat{x}_2 &= \hat{x}_1 + \hat{d}_2 = \hat{x}_1 + d_2 + q_2 = x_2 + q_2 && (36) \end{aligned}$$

So, at the ith iteration, we have

$$\hat{x}_i = x_i + q_i, \quad (37)$$

and there is no accumulation of the quantization error. The quantization error q_i is the quantization noise incurred by the quantization of the ith difference and it is significantly less than the quantization error for the original sequence. Thus, this procedure leads to an overall reduction of the quantization error and then we can use fewer bits with a differential encoding to attain the same distortion.

Subband Coding

Consider the sequence $\{x_n\}$ in the Figure 4. We see that while there is a significant amount of sample-to-sample variations, there's also an underlying long-term trend shown by the blue line that varied slowly. One way to extract this trend is to use moving window to average the sample value. Let us use a window of size two and generate a new sequence $\{y_n\}$ by averaging neighboring values of x_n:

$$y_n = \frac{x_n + x_{n-1}}{2}. \quad (38)$$

The consecutive values of y_n will be closer to each other than the consecutive values of x_n. Thus, the $\{y_n\}$ can be coded more efficiently using differential encoding than the sequence $\{x_n\}$. However, we want to encode the sequence $\{x_n\}$, not $\{y_n\}$. Therefire, we need another sequence $\{z_n\}$:

$$z_n = x_n - y_n = x_n - \frac{x_n + x_{n-1}}{2} = \frac{x_n - x_{n-1}}{2}. \quad (39)$$

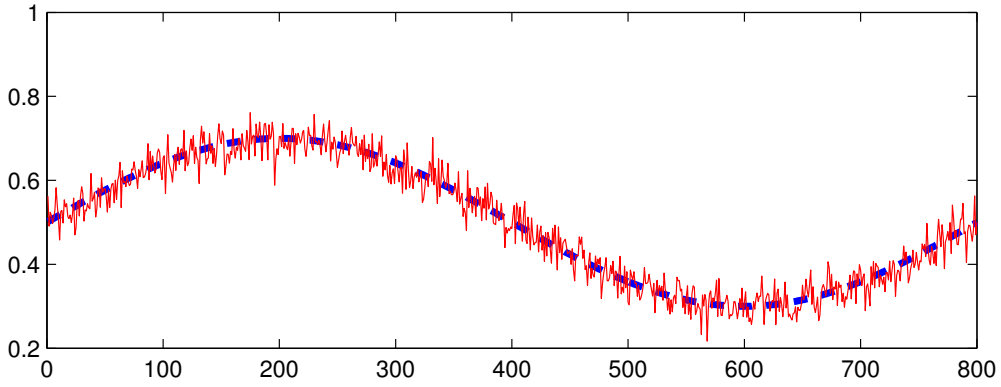

Figure 4: A rapidly changong source output that contains a long-run component with slow variations.

Then, the sequences $\{y_n\}$ and $\{z_n\}$ can be coded independently of each other.
Notice that we use the same number of bits for each value of y_n and z_n, but the number of elements in each of the sequences $\{y_n\}$ and $\{z_n\}$ is the same as the number of elements in the original sequence $\{x_n\}$. Although we are using the same number of bits of bits per sample, we are transmitting twice as many samples ans doubling the bit rate. We can avoid this by sending every other value of y_n and z_n. Let's divide the sequences $\{y_n\}$ into $\{y_{2n}\}$ and $\{y_{2n-1}\}$, and similarly, divide $\{z_n\}$ into $\{z_{2n}\}$ and $\{z_{2n-1}\}$. If we transmit either the even-numbered subsequencse or odd-numbered subsequences, we would transmit only as many elements as in the original sequence.