EE5585 Data Compression	May 7, 2013

Lecture 28

Instructor: Arya Mazumdar *Scribe: Yijun Lin*

1 Distributed Data Compression

From the previous lectures we have learn that for two data sources X and Y, one can acheive rates at $R_1 + R_2 \geq H(X + Y)$, with two seperated encoders which ignore the source correlation. However, if the two sources X and Y are correlated with each other, an optimal joint encoder can acheive compression rates at H(X,Y), by applying the Slepian Wolf Theorem.

Now let us consider a similar problem, supoose there are two set of data X and Y, which both have n number of elements (x1, x2, x3...xn) and (y1, y2, y3,... yn). These two set of data are corelated and not independent. The data sets are sent to two different encoders before being compressed together and being sent to a single joint decoder, as the relation shown in Figure 1.

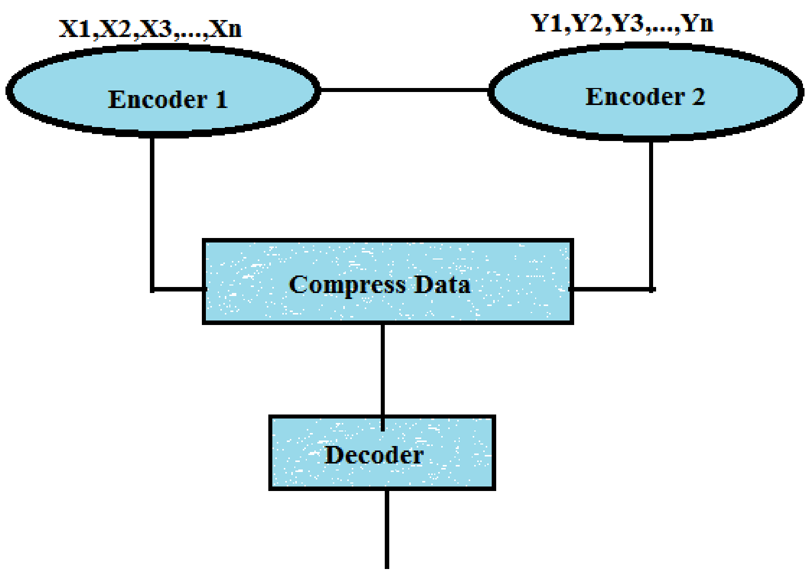

Figure 1: Block diagram of two correlated data sets

Such system is measured by the compress rate in bits per source symbol of the output streams of encoders. The single decoder is designed to be able to reconstruct the correlated data streams after compression in an optimal way. As we can see, encoder 1 has a rate of R1 = nH(X) while encoder 2 has a rate of R2 = nH(Y). The claim is if there is a communication channel between Encoder 1 and Encoder 2, by Slepian Wolf Theorem, the achievable rate of compression is:

$$nH(X) + nH(Y) \geq nH(X,Y) \quad \cdots\cdots \quad (1)$$

which means it can be compressed to H(X,Y) bits/symbol.

1.1 Distributed Data Compression with Bernoulli Source

Now let us consider the following example: assume there are two correlated data sources X ($x_1, x_2, x_3 \ldots x_n$) and Y($y_1, y_2, y_3 \ldots y_n$). X is Bernoulli(q) with the following properties:

$$\Pr(xi = yi) = 1 - p; \quad \Pr(xi \neq yi) = p.$$

Hence, Y is also Bernoulli(p + q +2pq) since:

$$\begin{aligned} \Pr(Y=1) &= \Pr(X = 1 \text{ and } Y = X) + \Pr(X = 0 \text{ and } Y \neq X) \\ &= q(1 - p) + (1 - q)p \\ &= p + q + 2pq \end{aligned}$$

The rate of compression we can achieve is to equal to: $nh(q) + nh(p + q + 2pq)$ bits. However, by the inequality (1) the optimal solution is to achieve $nh(q) + nh(p)$ bits.

Since: $$\begin{aligned} n[\, H(X,Y) \,] &= n[\, H(X) + H(Y|X) \,] \\ &= n[\, H(X) + H(Z) \,] \end{aligned}$$

Where $Z_i = X_i + Y_i \bmod 2$, for i = 1 to n

$$= n[\, h(q) + h(p) \,]$$

Notice it is always true that $p < \frac{1}{2}$, implies that:

$$q(1 - 2p) > 0 \quad \cdots\cdots\cdots\gg \quad p + q + 2pq > p$$

and $nh(q) + nh(p)$ is a increasing function.

1.2 Distributed Data Compression over Mod-2 Arithmetic

Suppose there are two data sources X ($x_1, x_2, x_3 \ldots x_n$) and Y($y_1, y_2, y_3 \ldots y_n$) over vector space $F^n = \{0, 1\}^n$. Let Z = X + Y. Notice that Z is also equal to X - Y or Y - X, due to fact that they are in mod-2 arithmetic.

Figure 2 shows the block diagram of sources X, Y and Z, where Z is Bernoulli(p).

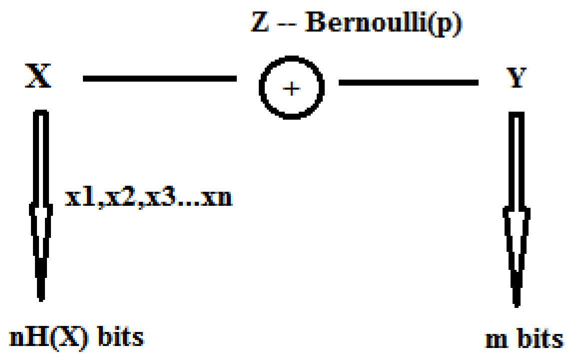

Figure 2

The rates of compression of X and Y is nH(X) and m bits, respectively. From what was discussed above, the achievable number of bits per symbol for such problem is:

$$H(X) + \frac{m}{n} = h(q) + \frac{m}{n}$$

Then we claim that there exists an m×n (m < n) matrix Φ, such that m ≈ $nh(p)$. With such matrix Φ, source data Y can be recovered from ΦY.

Here is the process of how to decode in the decoder: when we looking at the output of decoder, we have nH(X) and ΦY. Then do the following:

(1) decode X from nH(X);

(2) find ΦX simply by multiplying with each other;

(3) find ΦZ = Φ(X+Y) = ΦX + ΦY;

(4) recover Z from ΦZ;

(5) find Y from Y = X + Z.

However, when recovering Z from ΦZ from step (4), there is no unique solution Z. But since any solution of Z would work for the problem purpose, we turn to the typical set of Z, with arbitrarily small error probability. First, let us put some definitions around here:

We know that Z is a Bernoulli(p), let ΦZ = b for simple. For the typical set of Z: $A_\varepsilon^{(n)}(p)$ and $|A_\varepsilon^{(n)}| \le 2^{n(h(p)+\varepsilon}$. Now the goal is to find a vector $\hat{Z}$ from the typical set $A_\varepsilon^{(n)}$ that satisfies $\Phi\hat{Z} = b$. Such job could be complicated and require computer as a companion. But once we have the output $\hat{Z}$ from whatever the tools we may used, we can recover Y from $Y = X + \hat{Z}$.

The probability of error $P_e^{(n)}$can be calculated in bounds of:

$$P_e^{(n)} \le \Pr(Z \notin A_\varepsilon^{(n)}(p)) + \Pr(\exists Z' \neq Z;\ Z \in A_\varepsilon^{(n)}(p) \text{ and } \Phi Z' = b)$$

$$\le \sum_{Z' \in A\varepsilon(n)(p) \text{and } Z' \neq Z} \Pr(\Phi Z' = b)$$

To calculate Pr ($\Phi Z' = b$), we first choose Φ to be randomly uniformly with iid Bernoulli($\frac{1}{2}$) entries.

$$\begin{bmatrix} \Phi 11 & \cdots & \Phi 1n \\ \vdots & \ddots & \vdots \\ \Phi m1 & \cdots & \Phi mn \end{bmatrix} \times \begin{matrix} Z1' \\ \vdots \\ Zn' \end{matrix} = \begin{matrix} b1 \\ \vdots \\ bm \end{matrix}$$

Φ matrix: m× n Z' b

$$\Phi_{11}Z_1' + \Phi_{12}Z_2' + \cdots + \Phi_{1n}Z_n' = b_1$$

Since Φ is Bernoulli(1/2), the probability of making Z' to equal b_i is equal to $\frac{1}{2}$, implies that:

Pr ($\Phi Z' = b$) = $\frac{1}{2} \times \frac{1}{2} \times \cdots \times \frac{1}{2} = \frac{1}{2^m}$. Substitute it back the error inequality:

$$P_e^{(n)} \leq |A_\varepsilon^{(n)}(p)| \times \frac{1}{2^m}$$

$$\leq 2^{n(h(p)+\varepsilon)} \times \frac{1}{2^m}$$

$P_e^{(n)}$ goes to $\frac{1}{2}n\varepsilon'$ if $m = n(h(p) + \varepsilon + \varepsilon'$.

Now that $nh(q) + m = n[h(q) + \frac{m}{n}] \approx n[h(q) + h(p)]$.

1.3 Distribution Data Compression when two sources are closed

Consider a similar problem with the case that the two sources X ($x_1, x_2, x_3 \ldots x_n$) and Y($y_1, y_2, y_3 \ldots y_n$) are very closed to each other (ie. X-Y to be minimum). The process diagram is shown in Figure 3 below:

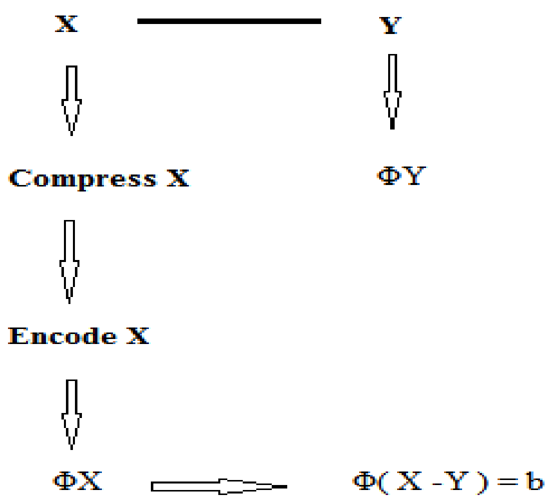

Figure 3

Let Z = X - Y, which can be considered as a noise function $\mathcal{N}(0, \sigma^2)$. The job once again is to find Z from ΦZ = b. Since the noise figure should be as small as possible, we want to find min $||Z||_{l2}^2$, such that ΦZ = b.

♠ ***Claim:*** Assuming that $\Phi\Phi^{-1}$ is nonsingular, $\hat{Z} = \Phi^T (\Phi\Phi^T)^{-1} b$

♠ ***Proof:*** suppose that Z satisfies ΦZ = b.

$$\text{Then } ||Z||_{l2}^2 = ||Z - \hat{Z} + \hat{Z}||_{l2}^2$$

$$= ||Z - \hat{Z}||_{l2}^2 + Z\,\hat{Z}^T(Z - \hat{Z}) + ||Z||_{l2}^2$$

The term of $Z\,\hat{Z}^T(Z - \hat{Z})$ goes to zero due to:

$$\hat{Z}^T(Z - \hat{Z}) = b^T(\Phi\Phi^T)^{-1}\Phi(Z - \hat{Z})$$

$$= b^T(\Phi\Phi^T)^{-1}(b - b)$$

$$= 0$$

$$\text{Hence, } ||Z||_{l2}^2 \geq ||\hat{Z}||_{l2}^2$$

Now we have: $\Phi\hat{Z} = \Phi\Phi^T (\Phi\Phi^T)^{-1} b$, which could be used as a solution.

2 Linear Error-Correcting Code

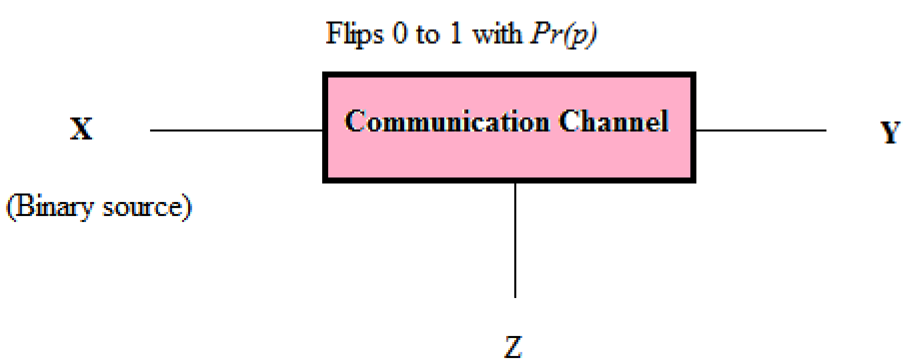

Figure 4

In general, as Figure 4 shown, suppose we have a data sources $X \in F^n = \{0, 1\}^n$ going through a communication channel, which flips 0 to 1 with probability of Pr(p),with some noises Z. the output Y of the channel is known. The process we discussed above is used to recover X by finding Z and calculating ΦY. Such processing code is called Linear Error-Correcting code.