

Calculating Weight Distribution of Our Ensemble

Raj and Hesam

March 6, 2010

1 Weight Distribution

In the previous report, we derived the following bound on mutual information:

$$I(x; z) \geq -\log \left[\left(\frac{e}{2}\right)^{n/2} \frac{1}{|C|} \sum_{w=0}^n B_w e^{-w\rho} \right] \quad (1)$$

If we denote the probability of having one bit of y equal to 1, then B_w is equal to $\binom{n}{w} (P_1)^w (1 - P_1)^{n-w}$. Replacing this relationship in equation (1), we get the following bound on mutual information.

$$\begin{aligned} I(x; z) &\geq -\log \left[\left(\frac{e}{2}\right)^{n/2} \frac{1}{|C|} \sum_{w=0}^n \binom{n}{w} (P_1)^w (1 - P_1)^{n-w} e^{-w\rho} \right] \\ &= -\log \left[\left(\frac{e}{2}\right)^{n/2} \frac{1}{|C|} (1 - P_1 + P_1 e^{-\rho})^n \right] \end{aligned} \quad (2)$$

Therefore, by noting $|C| = 2^k = 2^{Rn}$ where R is the code rate, we will have:

$$\frac{I(x; z)}{n} \geq -\log \left[\sqrt{\frac{e}{2}} \frac{1}{2^R} (1 - P_1 + P_1 e^{-\rho}) \right] \quad (3)$$

Therefore, all we have to do now is to calculate P_1 . To do so, we first calculate the probability P_1^j of having one bit of the codeword equal to 1 if the data sequence has weight j . We then multiply this value by the probability

of having a data word with weight j and sum up over all possible weights to get P_1 . In other words, if we assume $y = xA$, we have:

$$P_1 = \sum_{j=1}^n Pr(y_1 = 1|W(x = j))Pr(W(x = j)) \quad (4)$$

where y_1 is the first bit of y (one can choose any arbitrary bit as it does not change the problem).

In order to calculate P_1^j , without loss of generality we assume that the first j bits of the dataword x is equal to 1. Then, P_1^j is equal to having an odd number of 1's in first j positions of the first column of G . To derive this probability, we note that the total number of ways in which one can build a k -tuple with weight d is $\binom{k}{d}$. On the other hand, the total number of ways in which one can have $2i + 1$ ones in the first j positions of this k -tuple is $\binom{j}{2i+1}\binom{k-j}{d-(2i+1)}$. Hence, P_1^j can be calculated according to the following equation:

$$P_1^j = Pr(y_1 = 1|W(x = j)) = \sum_{i=L}^U \frac{\binom{j}{2i+1}\binom{k-j}{d-(2i+1)}}{\binom{k}{d}} \quad (5)$$

Where L and U are lower and upper bound on i . More specifically we note that:

$$L = \max\{0, \lceil \frac{j-1+d-k}{2} \rceil\} \quad (6)$$

and

$$U = \min\{\lfloor \frac{j-1}{2} \rfloor, \lfloor \frac{d-1}{2} \rfloor\} \quad (7)$$

Having P_1^j we will have P_1 which will give us B_w . Then, we can calculate the following probability:

$$Pr\left\{\frac{I(x; z)}{n} \leq Cap(C) - \epsilon\right\} \leq Pr\left\{-\log_2\left[\sqrt{\frac{e}{2}} \frac{1}{2^R} (1 - P_1 + P_1 e^{-\rho})\right] \leq Cap(C) - \epsilon\right\} \quad (8)$$

We would like to have the above probability go to zero as $n \rightarrow \infty$. Therefore, we would like to have the following condition on ρ and R :

$$\frac{2^R}{\sqrt{e/2}(1 - P_1 + P_1 e^{-\rho})} \geq 2^{Cap(C) - \epsilon} \quad (9)$$

Which of course depends on P_1 . I have performed numerical analysis on P_1 and it seems that $P_1 = .5$ as n goes to infinity, regardless of R and d ¹. Therefore, we will obtain the following condition on R and ρ :

$$\frac{2^{R+1}}{\sqrt{e/2}(1 + e^{-\rho})} \geq 2^{Cap(C) - \epsilon} \quad (10)$$

If ρ tends to infinity, then the above condition simplifies to $R \geq Cap(C) - \epsilon + \log(\sqrt{e/2}/2) \simeq Cap(C) - \epsilon - 0.78$. Therefore, we do not need to backoff $\log(\epsilon)$ bits to achieve capacity.

¹This is quite weird in my opinion.