ERC Report - March 3, 2010

Amir Hesam Salavati
E-mail hesam.salavati@epfl.ch

Supervisor: Prof. Amin Shokrollahi
E-mail amin.shokrollahi@epfl.ch
Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)

March 9, 2010

1 Weight Distribution

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

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

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.

$$\begin{aligned} I(x;z) &\geqslant -\log\left[\left(\frac{e}{2}\right)^{n/2}\frac{1}{|C|}\sum_{w=0}^{n}\binom{n}{w}(P_1)^w(1-P_1)^{n-w}e^{-w\rho}\right] \\ &= -\log[(\frac{e}{2})^{n/2}\frac{1}{|C|}(1-P_1+P_1e^{-\rho})^n] \end{aligned} \tag{2}$$

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

$$\frac{I(x;z)}{n} \geqslant -\log[\sqrt{\tfrac{e}{2}}\tfrac{1}{2^R}(1-P_1+P_1e^{-\rho})] \tag{3}$$

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:

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

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:

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

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

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

and

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

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

$$Pr\{\frac{I(x;z)}{n} \leqslant Cap(C)-\epsilon\} \leqslant Pr\{-\log_2[\sqrt{\frac{e}{2}}\frac{1}{2^R}(1-P_1+P_1e^{-\rho})] \leqslant Cap(C)-\epsilon\} \tag{8}$$

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

$$\frac{2^R}{\sqrt{e/2}(1-P_1+P_1e^{-\rho})} \geqslant 2^{Cap(C)-\epsilon} \tag{9}$$

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:

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

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

[1] This is quite weird in my opinion.