EE 121 – Intro to Digital Communication Spring 2016

Homework 4

Assigned: Thur, 2/25/16 Due: Thur, 3/3/16

Instructor: Omur Ozel *GSI: Preetum Nakkiran*

1 Singleton Bound

In this problem, we will see two ways of deriving a simple bound on the minimum-distance of a binary linear code.

(a) Warm-up: Recall that the *hamming weight* $w_H(c)$ of a codeword $c \in \{0,1\}^n$ is the number of ones in c. Show that for binary linear codes $\mathcal{C}$, the minimum-distance is equal to the codeword of minimum hamming weight. That is,

$$\min_{c_1 \neq c_2 \in C} d_H(c_1, c_2) = \min_{c \in C} w_H(c)$$

(b) Let $\mathcal{C}$ be an (n,k) binary linear code, with $(k \times n)$ generator matrix G. Using the previous part, show that the minimum-distance is bounded by

$$d_{min} \leq n - k + 1$$

(Hint: Consider the generator matrix G in systematic form. How many ones can each row have?) This is known as the *Singleton bound*.

(c) Now let us see a (formally) different way of deriving the same bound. Let H be the $n \times (n-k)$ parity-check matrix for $\mathcal{C}$ (recall, this means $v \in \mathcal{C} \iff vH = 0$). Clearly, any set of $n-k+1$ rows of H are linearly dependent. What does this imply about the minimum distance? *(Hint: Linear dependency of rows means there is a v such that $vH = 0$...)*

2 Capacity Lower Bound for Binary Symmetric Channel

We have seen a way of obtaining an achievable rate in the Gaussian channel with antipodal signaling at points $\pm\sqrt{E_0}$ as follows:

(a) We start by noting that for a fixed codebook $\mathcal{C}$, the probability of error is:

$$P[\mathcal{E}|\mathcal{C}] = \frac{1}{2^{Rt}} \sum_{m=1}^{2^{Rt}} P[\mathcal{E}|m, \mathcal{C}] \tag{1}$$

where we assume that each message $m \in \{1, \ldots, 2^{Rt}\}$ is equally likely. We note that by the union bound,

$$P[\mathcal{E}|m, \mathcal{C}] \leq \sum_{\hat{m}=1, \hat{m} \neq m}^{2^{Rt}} P[\hat{m}|m, \mathcal{C}] \tag{2}$$

We have

$$P[\mathcal{E}|m] = \mathbb{E}_{\mathcal{C}}[P[\mathcal{E}|m, \mathcal{C}]] \tag{3}$$

and

$$P[\mathcal{E}] \leq \frac{1}{2^{Rt}} \sum_{m=1}^{2^{Rt}} \sum_{\hat{m}=1, \hat{m} \neq m}^{2^{Rt}} \mathbb{E}_{\mathcal{C}}[P[\hat{m}|m, \mathcal{C}]] \tag{4}$$

where $\mathbb{E}_{\mathcal{C}}$ denotes expectation with respect to the randomly generated codebook $\mathcal{C}$. Here, we assume that all possible codebooks are equally likely.

(b) Assume that the Hamming distance between the codewords m and $\hat{m}$ are mapped is ℓ. That is, there are l indices in $\{1, 2, \ldots, t\}$ in which these codewords are different. We have

$$\mathbb{E}_{\mathcal{C}}[P[\hat{m}|m, \mathcal{C}]] = \sum_{\ell=1}^{t} f_{m,\hat{m}}^{\ell} Q(\frac{\sqrt{\ell E_0}}{\sigma}) \tag{5}$$

where $f_{m,\hat{m}}^{\ell}$ is the fraction of codes that allow ℓ Hamming distance between the codewords m and $\hat{m}$ are mapped. The term $Q(\frac{\sqrt{\ell E_0}}{\sigma})$ is due to the fact that probability of confusing m and $\hat{m}$ depends only on the Euclidean distance between these two codewords. We note that since the codebook $\mathcal{C}$ is generated randomly and equally likely, $f_{m,\hat{m}}^{\ell}$ is independent of m and $\hat{m}$ and is given by:

$$f_{m,\hat{m}}^{\ell} = \binom{t}{\ell} \frac{1}{2^t} \tag{6}$$

(c) By using $Q(\frac{\sqrt{\ell E_0}}{\sigma}) \leq \frac{1}{2} e^{-\frac{\ell E_0}{2\sigma^2}}$, we have

$$\mathbb{E}_{\mathcal{C}}[P[\hat{m}|m, \mathcal{C}]] \leq \sum_{\ell=1}^{t} \binom{t}{\ell} \frac{1}{2^t} \frac{1}{2} e^{-\frac{\ell E_0}{2\sigma^2}} \tag{7}$$

We use the formula

$$\sum_{\ell=1}^{t} \binom{t}{\ell} \alpha^{\ell} = (1+\alpha)^t \tag{8}$$

and we get

$$\mathbb{E}_{\mathcal{C}}[P[\hat{m}|m, \mathcal{C}]] \leq \frac{1}{2}(1 + e^{-\frac{E_0}{2\sigma^2}})^t \frac{1}{2^t} = 2^{-R^* t} \tag{9}$$

where $R^* = 1 - \log_2(1 + e^{-\frac{E_0}{2\sigma^2}})$.

(d) From (4), we have

$$P[\mathcal{E}] \leq \frac{1}{2} 2^{(R-R^*)t} \tag{10}$$

Hence, if $R < R^*$, then the average error probability over all possible codebooks vanishes as $t \to \infty$.

You are asked to apply the same method to find a capacity lower bound for the binary symmetric channel. Plot the capacity and the lower bound you find for all possible p_e and verify that the capacity is above the lower bound. Recall the hint given in the class. Also note that this method yields a lower bound for the capacity rather than the exact capacity. Determine which step(s) is(are) the bottleneck and could be improved to achieve a higher rate.

3 A Ternary Code

Consider a ternary code $\mathcal{C}$ over the usual $\mathbb{F}_3$ field spanned by the following rows

$$012$$

$$110$$

$$211$$

(a) What is the size of $\mathcal{C}$? What is the rate of $\mathcal{C}$? What is the minimum weight of $\mathcal{C}$? What is the minimum distance of $\mathcal{C}$?

(b) Is this code maximum distance separable?

(c) Is this code a perfect code?

(d) Determine the generator matrix for $\mathcal{C}$ in the form $G = [\ I \mid P\ ]$.

(e) Determine a parity check matrix for $\mathcal{C}$.

(f) Is it possible to add another codeword to $\mathcal{C}$ and still have the same minimum distance? If yes, determine which codewords can be added. If no, then Gilbert-Varshamov bound should hold for this code (why?); verify that this inequality holds.

4 Project: Linear Codes

Implement an encoder and decoder for binary linear codes, which will be useful for your project. **This problem is not optional and you have two weeks to complete.** *Your project group can work together and submit one solution.*

(a) Write the code constructor and encoder: For any (n, k), construct a binary linear code by selecting a random generator matrix G (optional: put G in systematic form, and check that it has full rank). Write the encoder, that takes k message bits to n encoded bits.

(b) Given G, write a syndrome decoder. This should work by pre-computing a lookup-table mapping the 2^{n-k} possible syndromes to the most-likely noise consistent with the syndrome. If you didn't put G in systematic form, you will also need to decode the original message bits.

(c) Test your encoder and decoder, assuming no noise, for parameters $(n, k) = (30, 20)$, and $(64, 54)$. *(Note: Try to be intelligent about how to construct the lookup-table, or it would be not be feasible to compute.)* Comment on the computational advantage of using a syndrome decoder, as opposed to the brute-force decoder (searching all possible codewords to find the one closest to the received vector).

(d) For $(n, k) = (30, 20)$, test your code in a simulated noisy channel, BSC_p for $p = 0.1$. What is (experimentally) the probability of error? Plot the simulated error probability for various values of p, and comment.