Solutions: Homework Set # 6

Problem 1

(a) $r \times (2^r - 1)$

(b) Note that

$$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \mathbf{0} \tag{1}$$

So, $d_{\min} \leq 3$.

Furthermore, if $\exists i, j$ such that $V_i \oplus V_j = 0$, necessarily $V_i = V_j$, which is not possible by construction of H. Thus $d_{min} > 2$. Hence $d_{\min} = 3$.

(c) $H_r X = 0$, and we are interested in the dimension of the kernel H.

H_r is a $r \times (2^r - 1)$ matrix and it is full rank. Thus we can set any $2^r - 1 - r$ elements of the vector X as desired and the rest of the elements will be determined by the system of equations $H_r X = 0$.

So, there are $2^{2^r - r - 1}$ such vectors in the kernel of H_r and thus its dimension is $2^r - r - 1$.

(d) To show that the code is linear, we should verify that

$$\text{if } X_1 \in C \text{ and } X_2 \in C \Rightarrow X_1 + X_2 \in C$$

This is true because

$$\begin{aligned} &\text{if } X_1 \in C \text{ and } X_2 \in C \\ &\Rightarrow H_r X_1 = 0 \text{ and } H_r X_2 = 0 \\ &\Rightarrow H_r(X_1 + X_2) = H_r X_1 + H_r X_2 = 0 \\ &\Rightarrow X_1 + X_2 \in C. \end{aligned}$$

So the code is linear.

Codeword length: $2^r - 1$.

Dimension of the code: $2^r - r - 1$.

$d_{min} = 3$.

Rate of the code $= \frac{k}{n} = \frac{2^r-r-1}{2^r-1}$.

Note that the rate of this code approaches 1 as $r \to \infty$.

(e) Enumerate the coins by binary sequences of length k. It can be shown that each trial can reveal at most one of the bits of the sequence assigned to the fake coin. Therefore, we have to use the device at least k times.

On the other hand, it is clear that the following algorithm finds the fake coin in exactly k trials.

```
1. $n = 0$.
2. $S_n = \{c_1, \ldots, c_{2^k}\}$
3. While $n < k$
    3.1. Partition $S_n$ into sets of same size, $A_n$ and $B_n$.
    3.2. Check $A_n$ by the detector.
        3.2.1. If fake coin $\in A_n$ then $S_{n+1} = A_n$.
        3.2.2. Otherwise $S_{n+1} = B_n$.
    3.3. $n = n + 1$
4. Fake coin $= S_k$.
```

(f) Enumerate the coins by integer numbers from 1 to 2^k. Take the smallest r such that $2^r - r - 1 \geq k$. It is clear that one can design a Hamming code $\mathcal{C}_1$ with parameters $(2^r - 1, 2^r - r - 1, 3)$. Therefore $\mathcal{C}_1$ has $2^{2^r-r-1} \geq 2^k = T$ codewords. Form a matrix $M_{2^r-1 \times 2^k}$ whose columns are 2^k arbitrary chosen codewords of C_1. We will use M as the measurement matrix. In fact we perform $2^r - 1$ measurements, each by a different detector (to make sure that at most one of the measurements are affected by the broken detector). In the i-th measurement, we chose a subset of the coins $A_i = \{j : M_{i,j} = 1\}$ and check this subset using a new detector. After $2^r - 1$ measurements, we obtain a vector v of length $2^r - 1$ whose i-th element shows the result of the i-th measurement, namely 1 if the fake coin is in A_i, and 0 otherwise. Note that if we have got the true answer from all the detectors, then v should be exactly equal to one of the columns of A (why?). Now, even if one detector has given us a wrong answer, only one element of v has been corrupted. Since the minimum distance of the columns of M is 3, the obtained v would be within distance 1 of a *unique* column of M. The number of that column declares the number associated to the fake coin.

It is easy to show that the smallest possible r to satisfy $2^r - r - 1 \geq k$, is $r = \log(k + \log k + o(1))$, and therefore the number of measurements, $2^r - 1$ would be $k + \log k - 1 + o(1)$, where $o(1)$ tends to zero as k grows.

Problem 2

(a)

$$C = \max_{p(x)} I(X;Y)$$

The mutual information is maximized when all the inputs are equally likely, and there are 27 symbols ($\mathcal{X} = \{a, b, c, \cdots, y, z, -\}$).

So

$$p(x) = \frac{1}{27}, \; x \in \mathcal{X},$$

which means that

$$p(y) = \frac{1}{27}, \quad y \in \mathcal{Y}, \qquad \mathcal{Y} = \mathcal{X}$$

The mutual information can then be calculated as follows:

$$\begin{aligned}
&I(X;Y) = H(X) - H(X|Y) \\
&H(X) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} = \sum \frac{1}{27} \log_2 27 = \log_2 27 \\
&H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) H(X|Y=y) = \frac{1}{27} \sum_{y \in \mathcal{Y}} H(X|Y=y) = \frac{1}{27} \sum_{y \in \mathcal{Y}} \log_2 3 = \log_2 3
\end{aligned}$$

$H(X|Y=y) = \log_2 3$, since, when given y, X can be from only 3 possible values, and it can be either of them with equal probability 1/3.

Hence

$$C_T = I(X_{\text{uniform}}; Y) = \log_2 27 - \log_2 3 = \log_2 9 \qquad \text{bits/symbol}$$

(b) If we restrict ourselves to $\mathcal{S} = \{a, d, g, j, m, p, s, v, y\}$, then we can always reconstruct the input from the output. However, if we add another symbol, then we no longer have this property. Hence, we can of most use 9 symbols. We call these "supersymbols".

(c) Assume now $p(x) = \frac{1}{27}, \; x \in \mathcal{X}, \; \mathcal{X} = \{a, b, c, \cdots, y, z, -\}$.

Let $\mathcal{S} = \{a, d, g, j, m, p, s, v, y\}$ and consider the following mapping:

$$\begin{array}{lll}
aa & \rightarrow & aaa \\
ab & \rightarrow & aad \\
ac & \rightarrow & aag \\
ad & \rightarrow & aaj \\
\vdots & & \\
ba & & \\
bb & & \\
\vdots & & \\
-a & & \\
-b & & \\
\vdots & & \\
-x & & \\
-y & & \\
-z & &
\end{array}$$

Then for each of the 3^6 two-letter sequences (all equally likely) on the left, we need to use 3 supersymbols. Since we are using the supersymbols, we can always decode at the output of our noisy typewriter the original 2 bits.

The information that we are transmitting in this way is $\log_2 3^6 = 3 \log_2 9$ bits, and we are using 3 symbols. Hence the rate of communication is

$$\frac{3 \log_2 9 \, \text{bits}}{3 \, \text{symbols}} = \log_2 9 \qquad \text{bits/symbols.}$$

In other words this simple scheme achieves capacity of the noisy typewriter channel.

Problem 3

(a) No.

(b) The information sequence is encoded to

$$(u_0 + u_1 x_1 + \cdots + u_{k-1} x_1^{k-1}, \cdots, u_0 + u_1 x_n + \cdots + u_{k-1} x_n^{k-1})$$
$$= (U(x_1), U(x_2), \cdots, U(x_{n-1}))$$

Where $U(D) = u_0 + u_1 D + \cdots + u_{k-1} D^{k-1}$.

This is exactly a Reed-Solomon code evaluated at $x_1, x_2, \cdots, x_n$ and the rate of the code is k/n.

(c) Reed-Solomon codes of rate k/n are shown to have $d_{\min} = n - k + 1$.

(d) The codewords of minimum Hamming weight $d_{\min}$ have exactly $n-k+1$ non-zero elements or equivalently $n - (n - k + 1) = k - 1$ zero elements. At the same time, as the parity check matrix is a full rank matrix we know that after choosing k elements of a codeword, the $n - k$ other are determined by the system of equations $\mathbf{H}\,\mathbf{x} = 0$. So we conclude, we count the number of codewords of Hamming weight $d_{\min}$ as follows:
$\binom{n}{k-1}$ is the number f possible ways to choose $k - 1$ elements of the codeword of length n and set all to zero. As described we are free to fix k elements of a codeword vector; $k - 1$ already set to zero; thus, we are left with one position and we have $|\mathcal{X}| - 1$ poddiblr non-zero elements of the field for it.

So $\binom{n}{k-1}(|\mathcal{X}| - 1)$ is the number of possible different codewords of Hamming weight $d_{\min}$ (Note that this proof works not only for Reed Solomon codes, but for any maximum distance code).

Problem 4

(a) As we have seen in the course, the capacity of BSC(p) is given by $C = 1 - H_2(p)$.

(b) Assume that n is the block length of the code, and np is an integer. We claim that the typical weight is np. Let j be the weight for which $f(i) = \Pr(w_H(z) = i)$ is maximum. It is easy to show that $f(\cdot)$ increases up to a certain i and then decreases. such i would be the most probable weight. We have $f(i) = \binom{n}{i} p^i (1-p)^{n-i}$, and so

$$\frac{f(i-1)}{f(i)} = \frac{\binom{n}{i-1} p^{i-1} (1-p)^{n-i+1}}{\binom{n}{i} p^i (1-p)^{n-i}}$$
$$= \frac{i(1-p)}{(n-i+1)p}$$

which is less than or equal to 1 for $i \leq (n+1)p$. Similarly one can show that the function is decreasing for $i \geq (n+1)p$. In fact $i = (n+1)p$ is the maximizing point of the function. So, for large enough n and $p \in (0, 1)$, np would be closest integer number to the maximizing weight.

Note that $\mathbb{E}[w_H(z)] = \mathbb{E}[\sum_i z_i] = \sum_i \mathbb{E}[z_i] = np$.

(c) Let x is transmitted and corrupted by a noise of weight np. Therefore the distance between the received signal y and x is np. The decoder can decode y correctly if and only if x is the unique codeword lies in the ball of radius. Therefore, there does not exist any codeword of distance less than $2np$ of x ($d_{\min} = 2np$).

More precisely, let x and x' be two codewords of distance $2np - 1$, *i.e.,* the weight of $z' = x \oplus x'$ is $2np - 1$. Define z as a binary sequence of length n and weight np which satisfies the following property: if $z'_i = 0$ then $z_i = 0$. Now the received signal $y = x + z$ is within distance np and $np - 1$ of the codewords x and x', respectively. This causes confusion for the decoder and it declares an error.

(d) The Gilbert-Varshamov bound says that there exists binary linear codes of length n and minimum distance d which satisfy

$$|C_n| \geq \frac{2^n}{\sum_{i=0}^{d-1} \binom{n}{i}}.$$

Approximating the denominator by $2^{nH_2(d/n)}$, we have

$$2^{nr} \geq \frac{2^n}{2^{nH_2(d/n)}},$$

or

$$r \geq 1 - H_2(\delta).$$

(e) Replacing $d = 2np$ from (c) in the GV bound, we get $r \geq 1 - H_2(2p)$. For $p = 1/4$, we have $r \geq 1 - H_2(1/2) = 0$, which is a trivial result. However, the capacity of the channel $C = 1 - H_2(1/4) = 0.188$. That means the GV does not guarantee any non-trivial performance for designing code for BSC(1/4), while one can communicate up to rate $C = 0.188$ using good (linear) codes.

(f) The rate of a capacity achieving code is arbitrary close to the capacity, $C = 1 - H_2(p)$. Using the GV bound the minimum distance would be $d_{\min} = n\delta = nH_2^{-1}(1 - r) = np$. Therefore it can tolerate up to $\frac{d_{\min}}{2} = \frac{1}{2}np$ errors, using the bounded distance decoder. However, as we have seen in part (b) the typical weight of an error is np. It can be shown that this decoder can only correct a very small portion of the error events. The main point to study is that in order to have capacity achieving codes, it is not enough to only concentrate on the minimum distance of the code, *i.e.,* the minimum distance is not everything! See Sections 13.5 and 13.8 of MacKay's book (Information Theory, Inference, and Learning Algorithms) for more details.