

ECS 452: In-Class Exercise #11

Instructions

1. Separate into groups of no more than three persons.
Only one submission is needed for each group.
2. **Do not panic.**

Date: 04 / 04 / 2019		
Name	ID (last 3 digits)	
Prapun	5	5

Consider a transmission over a BSC using block encoding.

Suppose the crossover probability of the BSC is $p = 0.1$. Note that this is assumed for the whole exercise.

Use the provided MATLAB function **PE_minDist** to help answer these problems.

```
function PE = PE_minDist(C,p)
% Function PE_minDist computes the error probability P(E) when code C
% is used for transmission over BSC with crossover probability p.
% Code C is defined by putting all its (valid) codewords as its rows.
M = size(C,1); % the number of (valid) codewords
k = log2(M);
n = size(C,2);

% Generate all possible n-bit received vectors
Y = dec2bin(0:(2^n-1));

% Normally, we need to construct an extended Q matrix. However, because
% each conditional probability in there is a decreasing function of the
% Hamming distance, we can work with the distances instead of the
% conditional probability. In particular, instead of selecting the max in
% each column of the Q matrix, we consider min distance in each column.
dmin = zeros(1,2^n); % preallocation
for j = 1:(2^n)
    % for each received vector y,
    y = Y(j,:);
    % find the minimum distance
    % (the distance from y to the closest codeword)
    d = sum(mod(bsxfun(@plus,y,C),2),2);
    dmin(j) = min(d);
end

% From the distances, calculate the conditional probabilities.
% Note that we compute only the values that are to be selected (instead of
% calculating the whole Q first).
n1 = dmin; n0 = n-dmin;
Qmax = (p.^n1).*(1-p).^n0;
% Scale the conditional probabilities by the input probabilities and add
% the values. Note that we assume equally likely input.
PC = sum((1/M)*Qmax);
PE = 1-PC;
end
```

```
close all; clear all;

% ECS315 Example 6.58
% ECS452 Example 3.66
C = [0 0 0 0 0; 1 1 1 1 1]; % repetition code

p = (1/10);
PE_minDist(C,p)

>> PE_minDist_demo1
ans =
    3.8506e-06
```

1. Suppose repetition code with $n = 5$ is used. Find the corresponding error probability $P(\mathcal{E})$.

In class, we have shown examples of how to use the function **PE_minDist** to find $P(\mathcal{E})$ for repetition code with $n = 5$. See, for example, the script **PE_minDist_demo1**. Here, the difference is simply that the crossover probability is changed from $1/100$ to $1/10$. So, by changing the value of p in the provided example to 0.1 , we can find the new $P(\mathcal{E})$.

$$P(\mathcal{E}) = 0.0086$$

2. Here, as in Problem 1, we will use $k = 1$ and $n = 5$.

This implies that any code must contain two 5-bit codewords.

Assume that these two codewords are distinct.

- a. Give an example of a code that performs worse (gives larger $P(\mathcal{E})$) than repetition code.

Method 1: We can use trial and error.

Method 2: Alternatively, for BSC with $p < 0.5$, the transmitted bits are more likely to stay the same than switch to their complements. The received vectors that have less bit errors are more likely to occur than ones with many bit errors. Therefore, code whose codewords are far away from each other seems to give small $P(\mathcal{E})$. When $n = 5$, the maximum "distance" between the codewords is 5. So, the repetition code in problem 1 already achieves this maximum. To find a code that performs worse, we try to make the distance between the two codewords smaller. This implies making some bits (in the corresponding positions) of the two codewords the same.

Some examples are provided below:

\mathcal{C}	Distance between the two codewords	$P(\mathcal{E})$
$\{0\ 0\ 0\ 0\ 0; 1\ 1\ 1\ 1\ 1\}$	5	0.0086
$\{0\ 0\ 0\ 0\ 0; 0\ 1\ 1\ 1\ 1\}$	4	0.0280
$\{0\ 0\ 0\ 0\ 0; 0\ 0\ 1\ 1\ 1\}$	3	0.0280
$\{0\ 0\ 0\ 0\ 0; 0\ 0\ 0\ 1\ 1\}$	2	0.1000
$\{0\ 0\ 0\ 0\ 0; 0\ 0\ 0\ 0\ 1\}$	1	0.1000

Note that smaller distance between the two codewords does not automatically correspond to smaller $P(\mathcal{E})$.

- b. Give an example of a code that performs as good as the repetition code.

From the discussion in the previous part, it seems that, to be as good as the repetition code, we should try the code whose distance between the two codewords is 5. Such code can be found easily: set the first codeword to be any 5-bit binary vector; invert all the bits to get the second codeword. An example is given below:

$$\mathcal{C} = \{00111, 11000\}$$

Remark: There are $2^5 = 32$ 5-bit binary vectors. The number of ways to choose two distinct codewords from this collection is $\binom{32}{2} = 496$. We can write a MATLAB script to try them all. The following table shows the number of codes for each $P(\mathcal{E})$ value:

$P(\mathcal{E})$	Number of codes
0.0086	16
0.0280	240
0.1000	240

So, there are 480 different solutions for part (a) and $16 - 1 = 15$ different solutions for part (b). The “-1” is there because we want a code that is different from the given repetition code. For part (b), in the discussion above, it seems that there are $2^5 = 32$ choices for the first codeword. However, because we treat the collection as a set, swapping the roles of these two codewords is considered the same code. Therefore, the actual number of choices is $\frac{32}{2} = 16$ choices.

3. In this problem, we will use $k = 2$ and $n = 5$.

This implies that any code must contain four 5-bit codewords.

Assume that these four codewords are distinct.

- a. Find the error probability $P(\mathcal{E})$ of the following code: $\mathcal{C} = \{00000, 11000, 00111, 11111\}$

$$P(\mathcal{E}) = 0.1252$$

- b. The best value of the error probability $P(\mathcal{E})$ for $k = 2$ and $n = 5$ is 0.0669.

Find at least one code that can achieve this $P(\mathcal{E})$.



Submit your answer of this part at <https://goo.gl/forms/Ka5pula4ekFNrZz13>.

You will get a penalty of “-1” if your answer is the same as the answer from another group who submitted before you. (Codes whose codewords are just “reordering” of another code will be

considered the same. For example, $\mathcal{C} = \{11000, 00000, 00111, 11111\}$ is considered the same as the code in part 3a.)



Here are the solutions found in the submitted in-class exercises:

1. [1 0 0 0 0; 0 1 1 0 0; 1 0 1 1 1; 0 1 0 1 1];
2. [1 0 0 0 1; 1 1 0 1 0; 0 1 1 1 1; 0 0 1 0 0];
3. [1 0 1 0 0; 0 1 0 0 1; 1 0 0 1 1; 0 1 1 1 0];
4. [0 0 1 1 0; 0 1 1 0 1; 1 1 0 1 1; 1 0 0 0 0];
5. [0 0 0 0 0; 0 1 1 1 0; 1 0 0 1 1; 1 1 1 0 1];
6. [0 1 1 0 1; 0 0 0 1 1; 1 0 1 0 0; 1 1 0 1 0];
7. [0 1 0 1 0; 1 0 0 0 1; 1 0 1 1 0; 0 1 1 0 1];
8. [0 1 0 0 0; 1 0 0 1 0; 0 0 1 1 1; 1 1 1 0 1];
9. [1 0 0 1 0; 0 1 0 0 1; 1 0 1 0 1; 0 1 1 1 0];
10. [0 0 1 1 0; 1 1 0 1 1; 0 0 0 0 1; 1 1 1 0 0];

Remark: It may be quite difficult to find one solution. However, with many groups working on the search, we expect some groups to be able to find solutions within reasonable time. Now, because the solutions must be posted on the web, other groups may look at the posted solutions and try to apply some “modification” to get their own solutions. The following operations do not change the distances between codewords and hence do not change the corresponding $P(\mathcal{E})$:

- Permute the columns
- Invert all the bits in some columns

For example,

$$\begin{array}{ccc}
 \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} & \xrightarrow{\text{permute columns}} & \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} & \xrightarrow{\text{Invert all the bits in some columns}} & \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \\
 \text{solution \#1} & & & & \text{solution \#2}
 \end{array}$$

Similarly, solutions #3 to #10 can be obtained from solution #1.