ECE 302, A Hard Drive Example — Error Control by Repetition

Question 1:

A hard drive designer found a way to reduce the physical dimension of a hard drive by a factor of 10 so that it can be used in many portable devices. The only problem is that with the dimension being so small, the stored data are easily contaminated by environmental noises. To be more specific, there is a 10% probability that a bit 0 stored in the hard drive may sometimes be mis-read as 1. Similarly, there is a 10% probability that a bit 1 will be mis-read as 0. The 10% error rate is too large and beyond the tolerable range. So the hard drive designer tries to reduce the error probability by using some signal processing to protect the data.

The following question demonstrates how to use probability arguments to analyze the performance of a simple scheme, termed the "repetition code."

Consider the following case: a bit X is chosen equally likely from $\{0, 1\}$ and is then stored in the hard drive. When you want to read the data, the hard drive outputs Y. By the description above, we know that

$$\begin{aligned} P(Y = 0|X = 0) &= 1 - \epsilon \\ P(Y = 1|X = 0) &= \epsilon \\ P(Y = 0|X = 1) &= \epsilon \\ P(Y = 1|X = 1) &= 1 - \epsilon, \end{aligned}$$

where $\epsilon = 0.1$. (For those of you who are interested, this conditional probability assignment is very popular and is generally named a binary symmetric channel (BSC), since both the input and output are binary and the "cross-over" probability are the same / symmetric.)

1. What is the sample space when considering jointly the values of X and Y?
2. What is the corresponding weight assignment for this example?
3. What is the probability that bit X is contaminated. (Namely, find the probability that $P(X \neq Y)$.)

A "repetition coding" scheme is as follows. For any X, instead of storing X, we store three identical bits XXX (basically repeating it three times.) We are hoping that this repetition can "protect" the data. (What is the price/cost of this data protection scheme?)

When we want to retrieve the data, we read three times (once for each of the three repeated bits respectively). Use (Y_1, Y_2, Y_3) to denote the outputs of the three "read" operations respectively. Then the hard drive uses a majority vote to determine $\hat{Y}$, the

majority of (Y_1, Y_2, Y_3). For example, if $(Y_1, Y_2, Y_3) = (0, 1, 0)$, the majority vote is $\hat{Y} = 0$. In the end, the hard drive outputs $\hat{Y}$ and we are hoping that this $\hat{Y}$ is a more reliable replica of the original bit X.

1. What is the sample space of this repetition scheme?

2. Suppose the noise for each stored bit is *independent*. What is the corresponding weight assignment?

3. Even with this repetition data protection, there is still some chance of error. What is the *error probability* of the repetition code?