Introductory Exercises
Error-Control Coding and the Binomial Distribution

Hans Georg Schaathun

Session 21 January 2015
Updated 20th January 2015

1 The binary symmetric channel

All digital communication systems can be described with similar generic models, whether they use radio waves, ultrasound, optical fibre, copper wire, or whatever. Messages are sent over a communication channel from Alice to Bob. Because of noise, the message received by Bob is *random* and probably not identical to the message submitted by Alice.

We will only consider one simple channel model, namely the *Binary Symmetric Channel* (BSC), characterised by the following

1. Each transmission accepts a single binary symbol (0 or 1).
2. Each transmission is independent.
3. For each transmission there are two mutually exclusive events: *(bit) error* or *correct transmission.*
4. The probability of an error p (*bit error probability*) is the same regardless of whether 0 or 1 was transmitted.

By using the channel repeatedly, we can send binary strings. Usually, we process strings of n bits at a time. An n-bit string of bits is called a *word.* Clearly, a transmission of a word can give anywhere from 0 to n bit error events. If there is at least one bit error, we say that we have a *word error.*

Exercise 1 *Suppose you send a 4-bit word on the BSC(p) with bit error probability* $p = 0.1$.

1. *What is the probability of getting exactly z errors for $z = 0, 1, 2, 3, 4$? Write the answer as a probability distribution table.*

2. *What is the expected number of bit errors?*

3. *What is the standard deviation of the number of bit errors?*

We write BSC(p) for the BSC with bit error probability p.

2 A few definitions

Write X_0 for the bit transmitted by Alice over a BSC, and Y_0 for the bit received by Bob. Note that the event of correct transmission can be described by $Y_0 = X_0$, and an error event by $Y_0 = X_0 + 1 \mod 2$. In other words, given the input X_0 the BSC draws a random bit E_0 and return $Y_0 = X_0 + E_0$.

Exercise 2 *What is the probability distribution of E_0 as drawn in the BSC(p)?*

When an n-bit word $\mathbf{c}$ is transimitted, the BSC is used n times and the error bit E_0 is drawn n times independently. We can write this as a vector $\mathbf{e} = (E_1, E_2, \ldots, E_n)$, where E_i are the different instances of E_0. We call $\mathbf{e}$ the *error word*.

Exercise 3 *Write* $\mathbf{s}$ *for the n-bit words transmitted by Alice, and* $\mathbf{e}$ *for the error word. Give an equation for the received word* $\mathbf{r}$.

The number of bit errors in $\mathbf{r}$ correspond to the number of one-bits in $\mathbf{e}$. This number is called the *Hamming Weight* $w(\mathbf{e})$ of $\mathbf{e}$.

3 Into Matlab

We want to simulate the BSC(p) in Matlab. We will assume transmission of n-bit words. (see the built-in help pages).

There is a standard technique to draw a random bit X where $P(X = 1) = p$. Draw first a number Y between 0 and 1 uniformly at random. If $Y < p$ then $X = 1$, otherwise $X = 0$. The `rand` function in Matlab gives you uniformly distributed vectors (check help pages). Note that if `X` is a vector, then a Boolean (binary) vector can be created in one operation with `Y = ( X<p )`.

Exercise 4 *Write an m-file with a function which produces a random error word* $\mathbf{e}$ *given the word length n and the error probability p.*

Exercise 5 (Optional) *In the function from the previous exercise, make the argument p optional, defaulting to $p = 0.5$.*

For the next exercise, you need the `sum` function in Matlab (check help pages).

Exercise 6 *Write an m-file with a function which calculates the Hamming weight of a binary word* $\mathbf{e}$.

4 Your first simulator

Simulation of random processes is known as Monte Carlo simulation, which is much used in physics and engineering. In the previous section, you have essentially written a Monte Carlo simulator for the BSC. By running the simulator repeatedly, we can generate data (a sample) for statistical analysis. This can be used to learn something about the probability distripution of errors on the channel.

We will focus on the number of bit errors Z per four-bit word transmitted. You can use the two functions you have just written to make *one observation* of Z.

Exercise 7 *Write a function which generates a data set (vector) of N observations of Z, by repeatedly drawing a random* $\mathbf{e}$ *and calculating the Hamming weight.*

Exercise 8 *Run the function from the previous exercise for $n = 100$. Draw a histogram to show the* empirical probability distribution *of z.*

Exercise 9 *Compare the empirical probability distribution from the previous exercise with the theoretical probability distribition from Exercise 1a. What do you see? Is the result reasonable?*