

Exercise Set 2

Error rates and error probabilities

Hans Georg Schaathun

Week 3–4 Spring Semester 2014

Session 1. The Binomial Distribution

1.1 The Binary Symmetric Channel

It is hard to imagine a world without digital telecommunications. Communication channels are always subject to noise. When you send a bit (0 or 1) over a digital channel, there is a certain probability p that the received bit is different from the one transmitted.

Definition 1 (Bit) *A bit is a variable over \mathbb{Z}_2 , i.e. with two possible values 0 or 1. Bit arithmetics is modulo 2. We will normally write $a \oplus b$ for $a + b \pmod{2}$.*

In communications it is rarely interesting to send a single bit. Usually, extremely long streams of bits are transmitted. It is often useful to group the bits together in blocks or *words* of fixed lengths.

Definition 2 (Word) *A word is tuple of bits. An n -bit word a vector in \mathbb{Z}_2^n . We use usual vector arithmetics on words.*

For the purpose of these exercises, we assume that the Channel is a so-called *Binary Symmetric Channel*, which means that each of the n bits of \mathbf{c} has independent probability p of changing into its opposite value.

Definition 3 (Binary Symmetric Channel) *The Binary Symmetric Channel (BSC) is a probabilistic function which takes one bit b as input. The output is b with probability $1 - p$, and $1 \oplus b$ with probability p .*

When a word \mathbf{c} is transmitted, the effect of the BSC is given as

$$\mathbf{r} = \mathbf{c} \oplus \mathbf{e}, \tag{1}$$

where \mathbf{e} is a binary vector where each bit is drawn independently at random with probability p for 1 and $1 - p$ for 0. We call \mathbf{e} the *error vector* (or error word). The error vector has a one bit for each bit error which occurred on the channel. This number is also as the Hamming weight $w(\mathbf{e})$.

Definition 4 (Hamming Weight) *The Hamming weight $w(\mathbf{w})$ of a word $\mathbf{w} = (w_1, w_2, \dots, w_n)$ is the number of indices i such that $w_i \neq 0$, i.e.*

$$w(\mathbf{w}) = \#\{i \mid i \in \{1, 2, \dots, n\}, w_i \neq 0\}.$$

Exercise 1.1 *Suppose you send a 7-bit word on the BSC with bit error probability 0.07. What is the probability of getting exactly ...*

1. zero errors?
2. one error?
3. two errors?

Exercise 1.2 *Suppose you send a 256-bit word on the BSC with bit error probability 0.01. What is the expected number of bit errors?*

Exercise 1.3 *Suppose you send 7-bit word on the BSC with bit error over $BSC(p)$. What is the mean μ of the number of bit errors per word? Give the answer as a function of p .*

Exercise 1.4 *Suppose again that you send a 256-bit word on the BSC with bit error probability 0.01. Let X be the number of bit errors in the received word. What is the variance $\text{var}(X)$?*

Exercise 1.5 *Send a 1024-bit word over the BSC with bit error probability 0.03. What is the expected number of bit errors in the received word?*

Exercise 1.6 *Let T be the number of bit errors when an n -bit word is transmitted over BSC with bit error probability p . What is the variance $\text{var}(T)$ when ...*

1. $n = 7$
2. $n = 1024$

1.2 Additional Exercises

Exercise 1.7 *What other examples of binomially distributed variables can you find? Review binomial distributions in the textbook (Frisvold and Moe).*

Exercise 1.8 (Frisvold and Moe) *Exercises 7.8 and 7.10, p. 104.*

Session 2. Starting on a simulation project

This session, we focus on getting our hands dirty. We will start considering a simulation project, which you have to complete by the end of Week 4 of teaching. An enquiry-based approach is recommended. Review matters of theory as you need it to understand the project. We return to fundamentals in the exercises in the next session, and it may be useful to look ahead.



Figure 1: Channel with coding.

2.1 Error-Control Coding

To reduce the error probability in communications, we use error-correcting codes, as depicted in Figure 1. The k -bit message word \mathbf{m} is encoded as an n -bit codeword \mathbf{c} . Because of noise on the Channel, the received word \mathbf{r} may or may not be equal to \mathbf{c} . The decoder function aims to recover the original message \mathbf{m} by returning an estimate $\hat{\mathbf{m}}$. In general the probability that $\hat{\mathbf{m}} \neq \mathbf{m}$ should be much smaller than the probability that $\mathbf{r} \neq \mathbf{c}$.

One of the most famous codes is also known named after Hamming, and it is defined for any message length k . You can find a Matlab implementation for $k = 3$ on the web pages. A message word of k bits gives a code word of $n = 2^k - 1$ bits. The decoder correctly decodes the message if and only if at most *one* bit is flipped on the channel.

2.2 Matlab Warm-up

Exercise 2.1 Write an *m*-file with a function which produces a random binary word. It should take two arguments, the word length N , and the probability p of each bit being 1. Each bit should be drawn independently.

Make the argument p optional, defaulting to $p = 0.5$.

Exercise 2.2 Plot and compare the probability distribution functions of the stochastic variable X/n where $X \sim B(n, 0.05)$ for $n = 10, 100, 1000$. Any observations?

Exercise 2.3 Suppose you send a word of 1024 bits over a BSC bit error probability $p = 0.02$. What is the probability of getting more than ...

1. 2% bit errors?
2. 5% bit errors?

(Use matlab to find the answer.)

2.3 Coding Simulator Project

Later in the module, you will do three major simulation projects. The following project is smaller, but it goes through the same key phases of

1. Modelling
2. Implementation and Simulation
3. Analysis (Statistics)

The modelling stage is often the most difficult one, and it must be done properly lest you do not know what to implement or simulate. The model has to be an accurate representation of the real world that you want to simplify. It must be complex enough to say something useful about the real world, yet simplified so that the implementation and simulation is feasible.

With a good model and basic programming skills, the implementation and simulation are straight forward. With good output from the simulation and sound working knowledge of statistics, the analysis is also quite manageable.

Project 1 *Build a simulator to estimate the error probability of a coding system. Two encoder/decoder pairs are posted on the web page. Use a simulation to estimate the error probability of each of these systems. Assess the quality of your estimate.*

The project is elaborated as exercises and subprojects below.

2.3.1 Modelling

In this project, the model is already (almost) given above. There are in fact three key models.

1. the message model is a probability distribution for input messages.
2. the system model in Figure 1
3. the channel model in Definition 3.

The encoder/decoder components are provided on the web page and require no further elaboration. The message model (1) has not been discussed before. We assume that messages are uniformly distributed over the set of k -bit vectors.

The input \mathbf{m} is drawn uniformly at random from \mathbb{Z}_2^k . We are interested in whether we have an error $\hat{\mathbf{m}} \neq \mathbf{m}$ or correct transmission $\hat{\mathbf{m}} = \mathbf{m}$.

2.3.2 Implementation and Simulation

The encoder/decoder pairs have been provided in Matlab, so you will probably complete your project in Matlab as well. If you want to use a different language, you will need to reimplement the encoder/decoder functions.

Subproject 1.1 *Implement a simulator for communication with error-control. You need to implement each of the three models discussed above. In other words, you need the following constituent functions (modules):*

1. *A random message generator, drawing k -bit binary vectors from a uniform distribution.*
2. *A random error generator, drawing n -bit binary vectors \mathbf{e} (as in (1)) where each bit is 1 with probability p .*

Subproject 1.2 Build a Monte Carlo Simulator which uses the functions from the subproject above, to simulate the transmission of uncoded 7-bit words over the BSC(p). Run simulations for $p = 0.2, 0.05, 0.01$ and record the number of bit errors per word over $m = 250$ trials. (This should give you a 1×250 (or 250×1) matrix in matlab.)

Make sure to keep a copy of the simulator from Subproject 1.2 when you develop the new simulator below.

Subproject 1.3 Integrate error-control into the simulator from Subproject 1.2, so that you can test coded transmission with each of the two supplied encoder/decoder pairs. Run simulations with each of the encoder/decoder pairs and record the number of decoding errors for $p = 0.2, 0.05, 0.01$.

2.3.3 Analysis

Analysis of the simulation output will be the topic of the next session.

Session 3. Point estimation

3.1 Reviewing simulator output

Exercise 3.1 Using the simulator from Subproject 1.2, send 7-bit words on the BSC and count the number X of bit errors per word, with bit flip probabilities $p = 0.2, 0.05, 0.01$. What is the mean \bar{x} over 250 trials?

Exercise 3.2 Using the simulator from Subproject 1.3, give an estimate for the probability of decoding error for each bit flip probability $p = 0.2, 0.05, 0.01$.

Exercise 3.3 Reviewing Exercise 3.1, what parameters of the probability distribution do you think we could reasonably estimate using \bar{x} ?

3.1.1 Cumulative Probabilities

Exercise 3.4 Suppose we send a 7-bit word over the BSC with bit error probability $p = 0.05$. What is the probability that two or more bits are incorrectly received?

Exercise 3.5 Given a BSC with bit flip probability p , what is the probability that the message is incorrectly received ($\mathbf{m} \neq \hat{\mathbf{m}}$) when it is encoded with the Hamming code for $k = 3$.

Exercise 3.6 Suppose we use the $[7, 4, 3]$ Hamming code to detect up to two bit errors on the BSC with bit flip probability 0.05. What is the probability of undetected errors?

Exercise 3.7 Suppose we send a n -bit word over the BSC with bit error probability p . What is the probability that 2 or more bits are incorrectly received? Give a general formula.

Exercise 3.8 (Frisvold and Moe) Exercises 7.9, 7.11, and 7.12, p. 104.

Session 4. Interval Estimation

4.1 Interval estimation for the simulator

Exercise 4.1 Let p be the probability of decoding error on the error-control system you have simulated. Find a 99% confidence interval for p for each of the two encoder/decoder pairs. How many trials does it take, in your opinion, to get a reasonable estimate?

Exercise 4.2 Let μ_p be the expected number of bit errors per word when sending a seven-bit uncoded word over $BSC(p)$. Let $p = 0.2$, and use your simulator to find a 95% confidence interval for μ_p . Choose the number of samples n such that the width of the confidence interval is at most 0.002.

Exercise 4.3 Let μ_p be as in the previous exercises. Calculate μ_p theoretically.

4.2 Other exercises

Exercise 4.4 (See video for solution) Suppose you are testing a system with error probability of 0.01. How many trials do you need to make your estimator \hat{p}_e fall between 0.011 and 0.009 99.75% of the time?

Exercise 4.5 Find the following probabilities using either software (e.g. Matlab) or a z -table (e.g. in Frisvold and Moe):

1. $P(0 \leq Z \leq 1)$ when $Z \sim N(0, 1)$
2. $P(0.5 \leq Z \leq 0.5)$ when $Z \sim N(0, 1)$
3. $P(2 \leq Z \leq 5)$ when $Z \sim N(3, 2)$

Exercise 4.6 Suppose you want to find out the percentage p of Norwegian students who think they have made a bad choice of degree programme. You poll 1000 students and 177 say they think their choice was bad. Give a 95% confidence interval for p .

Exercise 4.7 Exercise 9.11 p. 165 in Frisvold and Moe.

Exercise 4.8 Exercise 9.1 p. 150 in Frisvold and Moe.

Exercise 4.9 Exercise 9.2 p. 153 in Frisvold and Moe.

Session 5. A channel with memory

Exercise 5.1 *Implement a simulator generating random 10-bit error vectors with the probability distribution of the previous slide. Run a simulation and estimate the mean number of bit errors per 10-bit word. Run enough tests to get a 95% confidence interval smaller than ± 0.3 .*

At this stage, please make sure to complete the tutorial from Part 1 (Week 1-2 of teaching).

Appendix A. The π from Monte Carlo

This is a set of bonus exercises to illustrate another application of Monte Carlo simulation.

Exercise A.1 *Consider a circle inscribed in a square of size $2r \times 2r$. A Bernoulli trial consists of drawing a random point p within this square and noting success if p is also inside the circle and failure otherwise. What is the success probability of this trial?*

Project 2 *Use the observation in the previous exercise to devise an experiment to estimate π . Implement and run the simulation. What does the result tell you about π ?*

Exercise A.2 *Consider your experiment from the previous exercise. What is the probability that your simulation gives an estimate $\hat{\pi} \leq 3$.*