ETH zürich

Exercise 1: More Probability and AM Demodulation

© 2024
Christoph Studer (studer@ethz.ch)
Stefan M. Moser (moser@isi.ee.ethz.ch)

- **Exercise: Thursday, 10 October 2024**
- This exercise will be assisted in CAB G11 from 8:15 h to 10:00 h.
- Problem 2 requires a computer with MATLAB (or Octave) installed.
- **Due date:** Monday, 21 October 2024, 08:00 h
- **Submission instructions:** Upload your solutions (answers and MATLAB code) to Moodle as a *single* zip-file with a file name `⟨last-name⟩-⟨first-name⟩-e1.zip`. Your homework should consist of a single pdf file for the written answers and one MATLAB script per problem part. The MATLAB file must be named according to the syntax `E1_P2_1.m` (for "Exercise 1 Problem 2 Part 1").
- **Rules:** You may collaborate with other students on this exercise, but each student must turn in their own assignment. You must specify the names of your collaborators! Copying answers from previous years' exercises is strictly forbidden.

Problem	**Maximum Points**	**Points Received**
1	26	
2	28	
3	22	
4	24	
Total	**100**	

Problem 1: Simple Expected Values and Probability Mass Functions (26pts)

Assume that you have two *fair* dice, each can assume six values $\{1, 2, 3, 4, 5, 6\}$. In what follows, we model the outcome of dice one and two using the random variables x_1 and x_2, respectively. The probabilities of observing the numbers $\{1, 2, \ldots, 6\}$ on dice one is given by the probability mass function (PMF) $p_{1,i} = \mathbb{P}[\mathsf{x}_1 = i] = \frac{1}{6}$ for $i = 1, 2, \ldots, 6$; the same PMF also applies to dice number two. The goal of this problem is to analyze some basic probabilistic properties of these two dice.

Part 1 (6pts). Assume that dice one and two are *statistically independent*. Define the random variables $\mathsf{y} = \mathsf{x}_1 + \mathsf{x}_2$ and $\mathsf{z} = \mathsf{x}_1\mathsf{x}_2$. Compute the expected values $\mathbb{E}[\mathsf{y}]$ and $\mathbb{E}[\mathsf{z}]$.

Part 2 (10pts). Compute the PMF of $\mathsf{y} = \mathsf{x}_1 + \mathsf{x}_2$ and of $\mathsf{z} = \mathsf{x}_1\mathsf{x}_2$.

Part 3 (10pts). Consider the situation of Part 1, but assume that the random variables x_1 and x_2 are statistically *dependent*, i.e., the outcomes of dice one and two are somehow "entangled." To keep our discussion general, assume that the statistical dependence can be arbitrarily complicated. Assume that the marginal distribution of each of the dice separately remains fair.

Can you compute the expected values of $\mathsf{y} = \mathsf{x}_1 + \mathsf{x}_2$ and $\mathsf{z} = \mathsf{x}_1\mathsf{x}_2$, i.e., $\mathbb{E}[\mathsf{y}]$ and $\mathbb{E}[\mathsf{z}]$, respectively? Write down the result if you can compute it, and explain why not in case you cannot compute it.

Problem 2: The Basics of Monte–Carlo Sampling (28pts)

Remember Part 2 from Problem 1, which was quite annoying to compute by hand. Our goal is now to verify the PMF expressions for the random variables $\mathsf{y} = \mathsf{x}_1 + \mathsf{x}_2$ and of $\mathsf{z} = \mathsf{x}_1\mathsf{x}_2$ using MATLAB simulations.

We will use a technique called Monte–Carlo sampling to estimate the PMF expressions. Monte–Carlo sampling is a common approach to extract PMFs that are otherwise complicated to compute analytically. We will use this technique here to (i) demonstrate the basic concept of Monte–Carlo sampling and (ii) to numerically verify Part 2 of Problem 1. In future exercises, we will use this technique in order to numerically evaluate the error-rate performance of simple and complicated communication systems.

The principle of Monte–Carlo sampling for our purpose is as follows. Remember the fact that $\mathbb{P}[\mathsf{x} = x] = \mathbb{E}[\mathbb{1}(\mathsf{x} = x)]$ for a given x, where $\mathbb{1}(\mathsf{x} = x)$ is the indicator function that is 1 if the random variable x is equal to x and 0 otherwise.

For a fixed x, one can now numerically approximate the expectation operator

$\mathbb{E}[\mathbb{1}(\mathrm{x}=x)]$ with its sample mean, i.e.,

$$\mathbb{P}[\mathrm{x}=x]=\mathbb{E}[\mathbb{1}(\mathrm{x}=x)]\approx\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\big(\mathrm{x}[t]=x\big), \tag{1}$$

where the random variables $\mathrm{x}[t]$, $t=1,2,\ldots,T$, are independent and identically (IID) distributed as x, i.e., $\mathrm{x}[t]\sim\mathrm{x}$, $t=1,2,\ldots,T$, and T is the number of Monte–Carlo trials.

From the weak law of large numbers we have

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\big(\mathrm{x}[t]=x\big)\xrightarrow{P}\mathbb{E}[\mathbb{1}(\mathrm{x}=x)], \tag{2}$$

which means that for any nonzero margin $\varepsilon>0$ the following holds:

$$\mathbb{P}\left[\left|\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\big(\mathrm{x}[t]=x\big)-\mathbb{E}[\mathbb{1}(\mathrm{x}=x)]\right|>\varepsilon\right]=0. \tag{3}$$

The above expression means that the sample mean will converge in probability (thus the notation $\xrightarrow{P}$) to the true mean within an arbitrarily small margin $\varepsilon>0$ and with overwhelmingly high probability.

To summarize Monte–Carlo sampling, we have the following procedure: Fix x and perform the following two steps for $t=1,2,\ldots,T$:

1. Generate a realization of the random variable $\mathrm{x}[t]$
2. Evaluate $\mathbb{1}\big(\mathrm{x}[t]=x\big)$

One can then approximate the probability mass $\mathbb{P}[\mathrm{x}=x]$ via the sample mean on the right-hand side of (1).

The more Monte–Carlo trials T one performs, the more accurate your probability mass estimate will be. (You will learn more details on this aspect in later exercises.) As a rule of thumb, you want to observe about 100 ones generated by the indicator function before you stop the Monte–Carlo simulation; if you use too few trials, then the obtained results will be inaccurate.

Part 1 (24pts). Write a MATLAB script named "`E1_P2_1.m`" that performs a Monte–Carlo simulation that evaluates the PMFs of the random variables y and z from Part 2 of Problem 1 with independent x_1 and x_2. Perform $T=10,000$ Monte–Carlo trials. Generate bar-plots of the resulting PMFs.

Remark: Monte–Carlo sampling is a standard tool in communication-system design in order to numerically evaluate error-rate expressions, especially in situations for which analytical expressions are either too complicated or simply unavailable.

Part 2 (4pts). Compare the simulated results to the analytical results from Part 2 in Problem 1. Do they match?

Problem 3: Bit- and Packet-Error Rates (22pts)

Consider a packet-based communication system, which transmits B bits per packet. Each bit per transmitted packet has a bit-error rate (BER) of P_b, which indicates the probability of a bit being in error. In what follows, we assume that all of the bit errors are statistically independent.

Part 1 (4pts). Compute the packet error rate (PER) in dependence of P_b and B. The PER indicates the probability of at least one bit per packet is transmitted with an error.

Part 2 (4pts). Numerically compute the PER assuming that $P_\mathrm{b} = 10^{-3}$ and $B = 128$.

Part 3 (6pts). Assume a communication system that retransmits the entire packet if the packet was received with an error. Compute the expected number of required (re)trans-mis-sions in dependence of P_b and B. Assume that all errors (bit and packet errors) are statistically independent.

Part 4 (4pts). Numerically compute the number of expected (re)transmissions assuming $P_\mathrm{b} = 10^{-3}$ and $B = 128$.

Part 5 (4pts). For the assumptions made in Part 4, what is the worst-case number of (re)transmissions? What are the implications of this observation?

Problem 4: AM Demodulation with Rectification (24pts)

Let us assume that we demodulate an AM signal using a full-wave rectifier, i.e., by taking the absolute value of the received signal

$$\tilde{m}(t) = |r(t)| = |m(t)|\,|\cos(2\pi f_\mathrm{c} t)|, \quad t \in \mathbb{R}. \tag{4}$$

In order to see why this approach performs demodulation, we will use the Fourier series. In contrast to the complex-valued Fourier series discussed in the signals and systems tutorial, it is easier to use the following real-valued Fourier series expansion:

$$x(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2\pi n t}{T}\right) + B_n \sin\left(\frac{2\pi n t}{T}\right), \tag{5}$$

where the signal $x(t) = x(t-T)$ for $t \in \mathbb{R}$ is periodic with period T. Here, the Fourier series coefficients are calculated as follows:

$$A_0 = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\,\mathrm{d}t, \tag{6}$$

$$A_n = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\cos\left(\frac{2\pi n t}{T}\right)\mathrm{d}t, \quad n \in \mathbb{N}, \tag{7}$$

$$B_n = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\sin\left(\frac{2\pi n t}{T}\right)\mathrm{d}t, \quad n \in \mathbb{N}. \tag{8}$$

Part 1 (16pts). Expand the rectified cosine signal $x(t) = |\cos(2\pi f_c t)|$, $t \in \mathbb{R}$, into its Fourier series.

Part 2 (8pts). Show that by performing full-wave rectification

$$\tilde{m}(t) = |r(t)| = |m(t)|\,|\cos(2\pi f_c t)|, \quad t \in \mathbb{R}, \tag{9}$$

followed by an ideal lowpass filter, one can perfectly recover the message signal up to a scale factor.