SGN-21006 Advanced Signal Processing

Exercise 2: Random signals

We study the distribution of the values of various signals, namely audio and image data. First the basic MATLAB functions are introduced for generating data according to the uniform and normal distribution. Secondly we look at audio and image data and try to describe it in probabilistic terms. Your task is to run the provided code and understand its functionality and modify it when asked.

1 MATLAB generated data

In MATLAB the most used random data generators are the `rand` and `randn` functions. A short description is of the functions is given below:

- `rand`(n, m) generates $n \times m$ matrix containing pseudo random values drawn from the standard uniform distribution on the open interval $(0, 1)$;
- `randn`(n, m) returns $n \times m$ matrix containing pseudo random values drawn from the standard normal distribution with zero mean and unit variance.

The Laplace distribution is implemented using the inverse transform method that is presented in lecture 2, on the last page. We can generate n samples from any given cumulative distribution function $F(x)$ starting with n samples from the uniform distribution $\mathcal{U}(0, 1)$. If the uniform distributed samples are $u_1 \ldots u_n$ then the data $x_i = F^{-1}(u_i)$ is distributed according to $F(x)$, with

$$F^{-1}(u) = \arg\min_x \{F(x) \geqslant u, u \in [0,\ 1]\}. \tag{1}$$

For the Laplace distribution we have the cumulative distribution function,

$$F_L(x) = \frac{1}{2} + \frac{1}{2}\ \mathrm{sgn}(x - \mu)\left(1 - \exp\left(\frac{-|x - \mu|}{b}\right)\right), \tag{2}$$

which produces

$$F_L^{-1}(u) = \mu - b\ \mathrm{sgn}\left(u - \frac{1}{2}\right)\ln\left(1 - 2|u - \frac{1}{2}|\right). \tag{3}$$

Run the following matlab scripts to see the results and give a short answer to the following questions: (you can write your answer as a comment to the matlab file)

`uniform_distribution.m`

Q1. [2^{nd} figure] Why does the histogram for the uniform distributed data look different when we place the 1000 uniformly distributed variables in 50 bins? Why the values are not distributed more or less uniformly in the bins?

Q2. [3^{rd} figure] What interval corresponds to the distribution?

`normal_distribution.m`

Q3. [1^{st} figure] When looking at the histogram, can you find an estimate for the average value found in n? What about the variance? How?

Q4. [2^{nd} figure] What are the values of m and s such that the generated vector contains values from a normal distribution with mean 2 and variance 2.

`laplace_distribution.m`

Q5. What do the variables μ and b control?

2 Real data, audio and image

For this experiment we use two sources for the data, a fragment of audio data `BeeGees.wav` and an image `Barbara.tif`. The goal is to see if the data has a known distribution; to see if a probabilistic description can be found. In this case we try to fit a normal distribution on the data.

Run the `real_signals_are_random.m` placing breakpoints to stop the execution at key points. The file calls the function `real_signals_are_random_normal`; it is responsible for fitting a normal distribution on the data vector given as an input (The function uses the `pause` command so, to continue the execution after looking at the results, hit space.). Look over the code and answer the following questions:

`real_signals_are_random.m`

Q6. What is the length of the signal, its sampling frequency and the number of bits on which it is represented? Study the function *wavread*. Hint: type *help wavread* in the command window.

Q7. How do the zeros (silence zones) from the audio data influence the fitting to the normal distribution? Why this happens?

Q8. Repeat the fitting for the absolute values of the audio signal, assuming an exponential distribution. Is the fit better in this case than for the Gaussian distribution?

Q9. Is the normal distribution still describing one second of the audio data?

Q10. The usual hypothesis is 'quantization errors are uniformly distributed'. Does the hypothesis hold?

Q11. Comment the results for the image data.