Exercises 7

Exercise 1.
In this exercise, we consider the application of the Wiener filter in reducing additive noise. Consider a signal $X[n]$ embedded in additive zero mean white Gaussian noise. That is,

$$Y[n] = X[n] + W[n].$$

Assume that $X[n]$ and $W[n]$ are uncorrelated.

(a) Derive the transfer function of an optimal non-causal filter.

(b) We define the following signal to noise ratio:

$$\mathrm{a}(\omega) = \frac{R_{xx}(e^{j\omega})}{R_{ww}(e^{j\omega})}.$$

How is the Wiener filter response in the case of noise-free frequencies, i.e., $a(\omega_o) \gg 1$? and in the case of very high noise, i.e., $a(\omega_o) \approx 0$? what can you conclude?

Exercise 2. Wiener Filter
Suppose that a desired process $X[n]$ is generated by filtering the white gaussian noise $W[n]$ (centered with variance 1) using a filter $h[k]$, where

$$H(z) = \frac{1 + \frac{3}{4}z^{-1}}{1 + \frac{1}{2}z^{-1}}. \tag{1}$$

Consider now the signal $Y[n] = X[n] + V[n]$, where $V[n]$ is zero mean white Gaussian noise with variance $1/2$ and uncorrelated with $W[n]$.

(a) Design a Wiener filter for estimating $X[n]$ from $Y[n]$.

(b) Repeat (a) for the case when $V[n]$ is a random variable given by:

$$V[n] = \frac{1}{3}X[n-1] - \frac{1}{9}X[n-2].$$

Exercise 3.
In Markov chains, the probability that the chain is in a particular state k is given by π_k and the distribution of probabilities for states $1 \le k \le n$ can be represented by a vector π such that $0 \le \pi_k \le 1, \forall k$ and $\sum_{k=1}^{n} \pi_k = 1$. Markov chain is characterized by a matrix $\mathbf{M}$ such that the probability distribution π_{m+1} at instant $m+1$ can be computed from the distribution π_m at instant m as follows:

$$\pi_{m+1} = \mathbf{M}\pi_m\,.$$

The matrix $\mathbf{M} = [m_{ij}]$ has the following properties: $0 \leq m_{ij} \leq 1$ and $\sum_{i=1}^{n} m_{ij} = 1, \forall j$. An equilibrium distribution of the Markov chain π^e is a solution of the eigenvalue equation:

$$\mathbf{M}\pi^e = \pi^e$$

(a) Let $\|.\|_1$ be the 1-norm, defined as follows:

$$\|\mathbf{x}\|_1 = |x_1| + \ldots + |x_n|, \qquad \mathbf{x} \in \mathbb{C}^n \, .$$

Show that $\|\mathbf{M}\pi\|_1 = \|\pi\|_1$ for any probability distribution π.

(b) Does the sequence π_m always (for any initial vector π_0 with the above properties) converge to an equilibrium distribution π^e?

Exercise 4. Markov Chain

In this exercise you will generate Markov chain with 3 states. This could be for example the state of the weather: sunny, snowy and rainy. We can write transition probabilities in the matrix form:

$$\begin{bmatrix} 0.8 & 0.2 & 0.1 \\ 0.1 & 0.6 & 0.2 \\ 0.1 & 0.2 & 0.7 \end{bmatrix} \tag{2}$$

where the probabilities of the states at step $n+1$ given at step n the process was in the state i is defined by i-th column.

(a) *Warm up exercise:* Draw the diagram depicting this Markov Chain and it's transition probabilities.

(b) Generate a realisation of this Markov Chain in Python or Matlab, picking any state you like as the initial state. You can use provided Jupyter Notebook.

(c) Is your process (with a fixed starting point) stationary? What is the probability distribution after one, two and five steps? You can calculate the distribution in Python/Matlab.

(d) What equations the initial probabilities have to satisfy for the process to be stationary?

(e) Calculate such inital probabilities in Python/Matlab. Do it twice using different methods: using eigenvalue decomposition and estimate using your realisation of the process. Give example situations where you would use each method.

(f) Assume now that you don't know the the transition probabilities of your Markov Chain. Formulate the Maximum Likelihood as optimisation problem. What method would you use to solve it?

(g) *Additional Question*: Using the generated realisations of your signal solve your problem using `solve` in Matlab or `scipy.optimize.minimize` in Python. Is your estimation accurate?