KAUST **Project: Fall 2021**
AMCS241/STAT250 **Stochastic Processes**

Deadline: Wednesday, November 25 , 18:00

You should use R to solve the problems. The tasks should be performed individually or in pairs. You do not need to write a full project report with introduction and conclusions, it is enough with concise but sufficiently detailed answers to the questions.

We highly recommend you use the RMarkdown template *Template.Rmd*, which you can open in RStudio. Write the code and the answers in the designated areas and then compile by clicking on the *Knit* button. For a quick introduction to Rmarkdown watch the video in the webpage: `https://rmarkdown.rstudio.com/lesson-1.html`. Submit the answers in a single PDF file, and the code in Blackboard. No late projects will be considered.

The data files needed for the project can be found on the course homepage. Load the datasets with `load('data.RData')`.

Problem 1 (Estimation of the mean and covariance function)
The dataset `covid` contains the total Covid-positive cases $I(t)$ in the US.

(a) Calculate the logarithm infection rate $r(t) = \log\left(\frac{I(t)}{I(t-1)}\right)$. Then plot $r(t)$. Does the logarithm infection rate $r(t)$ seem stationary?

(b) A common trick when dealing with non-stationary processes is to work with differences $\Delta_r(t) = r(t) - r(t-1)$. Plot $\Delta_r(t)$, does it looks more stationary comparing to $r(t)$?

(c) Assume $\Delta_r(t)$ is a weekly stationary stochastic process. Create an `R` function that implements the mean function estimator and the covariance function estimator for lag $h \in [1, 30]$. What patterns do you find from the covariance function estimates?

(d) Compute a 95% confidence interval for the mean value, based on the covariance function estimates and the mean value estimate in (c).

Problem 2 (Gaussian Processes)

(a) Simulate a Gaussian process $X(t), t = 1, \ldots, 200$, with mean value function $m(t) = 0$ and covariance function $r(h) = Cov(t, t+h) = \exp(|h|)$. This process is sometimes referred to as the Ornstein-Uhlenbeck process. Plot the simulated process.

Hint: You can compute the mean and covarince matrix of $\mathbf{X} = (X_1, X_2 \ldots X_{200})$ *and draw a random vector from a multivariate normal distribution using the function* rmnorm *from the library* mnormt.

(b) Apply the mean and covariance function estimators (for $h = 0, 1, \ldots, 10$) developed in Problem 1 to the simulated data of the previous exercise. Check that your simulation is correct, by comparing with the true mean and covariance function of the process.

(c) Now simulate a Gaussian process $X(t), t = 1, \ldots, 200$ with $m(t) = \sqrt{t}$ and the same exponential covariance $r(h) = \exp(|h|)$ as before. Plot the simulated process.

(d) Apply the mean and covariance function estimators (for $h = 0, 1, \ldots, 10$) developed in Problem 1 to the simulated data in the previous exercise. Compare with the true mean and covariance function of the process. Why do you think the estimators fail for this data?

Consider the vector *sp* in the *data.Rdata* file. The data contains the S&P 500 stock market index from 01/10/2019 to 25/09/2020. The S&P 500 stock market index measures the stock performance of 500 large companies listed on stock exchanges in the United States. To stabilize the financial data people usually work with the log-returns. Given $X_1, X_2, \ldots, X_{249}$ the log-returns are given by $R_t = log(\frac{X_t}{X_{t-1}})$ for $t = 2, 3, \ldots, 249$.

(e) Compute the log-returns of the data. Plot the original data and the log-returns. Based on the plots, explain why it is better to model the log-returns and not the original data using a stationary Gaussian process.

We analyze the log-returns using a stationary Gaussian process with:

$$m(t) = 0 \quad \text{and} \quad r(h) = \frac{\sigma^2}{(1-a^2)} \times (-a)^{|h|}.$$

(f) Maximum likelihood estimation yields $\hat{\sigma} \approx 0.02$ and $\hat{a} \approx 0.4$. Using this result predict the values of R_{250}, R_{251}, and R_{252}, by computing their mean and variance.
Hint: Consider $\mathbf{X}_a = (R_2, \ldots, R_{249})$ *and* $\mathbf{X}_b = (R_{250}, R_{251}, R_{252})$ *and use the following result:*

If $\mathbf{X} = (\mathbf{X}_a^T, \mathbf{X}_b^T)^T$ *has a multivariate normal distribution,*

$$N\left(\begin{bmatrix}\boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb}\end{bmatrix}\right)$$

then $\mathbf{X}_b|\mathbf{X}_a = \mathbf{x}_a \sim N(\boldsymbol{\mu}_{b|a}, \boldsymbol{\Sigma}_{b|a})$*, where*

$$\boldsymbol{\mu}_{b|a} = \boldsymbol{\mu}_b + \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}(\mathbf{x}_a - \boldsymbol{\mu}_a), \quad \boldsymbol{\Sigma}_{b|a} = \boldsymbol{\Sigma}_{bb} - \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}\boldsymbol{\Sigma}_{ab}$$

Problem 3 (Periodogram and filtering)

(a) Compute the periodogram of the discrete-time process you simulated in 2.a) for $w \in (-0.5, 0.5]$. Compare it with the true spectral density function of the Ornstein-Uhlenbeck process, $R(\omega) = \frac{2}{1+(2\pi\omega)^2}$.
Hint: You can compute the Fourier transform of $X_0, X_1, \ldots, X_{n-1}$*, for frequency* w *by* $\sum_{t=0}^{n-1} X_t \exp(-i2\pi wt)$*.*

(b) Plot the periodogram for the same process, but now with $n = 1000$ simulation points. Why do you think the estimation does not improve?

An electroencephalogram (EEG) is a test that detects electrical activity in your brain using small, metal discs (electrodes) attached to your scalp. One of the most widely used methods to analyze EEG data is to decompose the signal into functionally distinct frequency bands, such as delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (12-30 Hz), and gamma (30-100 Hz). Deep sleep is characterized by its predominance of slow-waves with a frequency range comprised between 0.5 to 4 Hz (i.e., delta band), which reflects synchronized brain

activity. Conversely, wakefulness is characterized by very little delta activity and much more higher-frequencies activity.

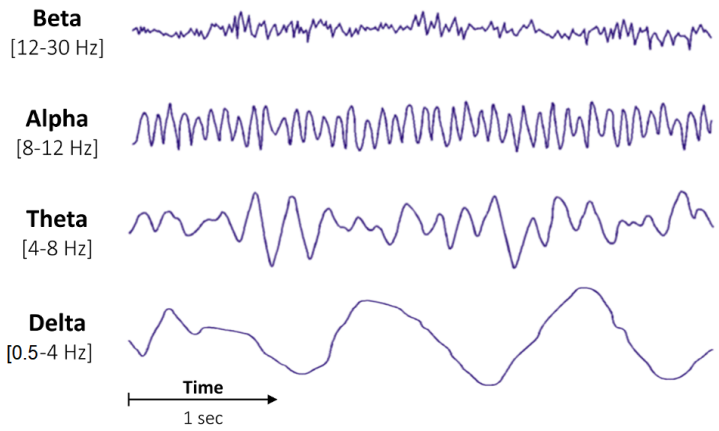

Fig. 1. The EEG signal can be decomposed into different frequency bands.

Consider the vector *eeg* in the *data.Rdata* file. The data consists of a 30-second extract of an EEG signal in Volts. The sampling frequency is 100 Hz.

(c) Compute the periodogram of the EEG signal for $w \in [0, 50)$. Plot the results.

Hint: The estimator in (a) assumes the data is sampled at $t = 0, 1, 2, \ldots, (n-1)$. *Since the data is sampled at* $t = 0, d, 2d, \ldots, (n-1)d$ *with* $d = 1/100$ *you can consider* $\mathcal{F}(X)(w) = \sum_{k=0}^{n-1} X(dk) \exp(-i2\pi wdk)$, and then multiply the periodogram by d.

The Welch's periodogram attemps to reduce the noise of the periodogram estimates (you can read more in Chapter 9.5 in the Lindgren book). The Welch's periodogram for a time window of 4s is obtained with the following code:

```
>> #install.packages("bspec")
>> library(bspec)
>> welsh = welchPSD(ts(eeg, frequency = 100), seglength = 4)
>> plot(welsh$frequency, welsh$power, type = 'l')
```

The contribution of each frequency band to the overall signal can be computed with the average band power. As an example, the average band power of the Delta signal is the area under the periodogram between the frequencies 0.5Hz and 4Hz.

(d) Using the Welch's periodogram compute the average band power of the delta, theta, alpha, and beta bands. Do you think the EEG data was measured while the person was awake or asleep?

Hint: To compute the average band power for a given band, you can make a simple approximation of respective area under the periodogram in the object welsh.

We now want to decompose the original EEG signal into the different bands as displayed in the image. To filter the delta band we should consider the transfer function $H(\omega) = I_{0.5<|\omega|<4}(\omega)$, and likewise for the other bands.

(e) Show that the impulse response $h(t)$ of the transfer function $H(\omega) = I_{a<|\omega|<b}(\omega)$ is:

$$h(t) = \frac{\sin(2\pi tb) - \sin(2\pi ta)}{\pi t}.$$

(f) Considering the previous impulse response $h(t)$, apply filters to the EEG signal to obtain the delta, theta, alpha, and beta signals. Plot the results.

Hint: Since the data is sampled at time points $t = 0, d, 2d, \ldots, (n-1)d$ with $d = 1/100$ you can consider:

$$\mathcal{L}X(t) = \sum_{i=0,\ t-di\geq 0}^{n-1} h(di)X(t-di)d.$$

In the previous expression you need to evaluate $h(0)$. Doing so directly wields an indetermination (0/0). Thus, for $i = 0$, consider $h(0) = 2(b-a)$ (obtained by L'Hôpital's rule).

Problem 4 (Simulation of Markov Chain)
During the orientation week in KAUST, there are many new students that need to open bank accounts in SAMBA bank in order to receive their stipends. For their convenience, SAMBA decide to open a special counter just for opening bank accounts for new students. Suppose the number of new students arrive at SAMBA bank at time t (starts from 0), $N(t)$ follows a Poisson process with rate λ. Moreover, suppose the bank operates efficiently without stop and follows "first come, first serve" principle. If the counter is busy with one customer, the other new arrivals have to wait in queue. The time needed to finish with just one customer follows $\text{Exp}(1/\mu)$ independently (Exponential distribution with mean $\frac{1}{\mu}$). The data frame `samba` contains two variables, `customer` and `processing_time` represent the arriving time of each customer and the processing time needed for the bank counter associated with that customer.

(a) Estimate μ and λ using the data.

(b) Simulate the whole process (customers arriving-queuing-leaving) described above using the estimates from (a), and use the samples to compute the average waiting time for the customers. Make sure to simulate enough customers so that you get a good estimate of the distribution. Can you guess the distribution of the waiting time for each customer?

(c) Suppose there are 4 counters with the same efficiency specialized for opening bank accounts for the new students. If any counters are idle, they would serve the next customer immediately. Simulate the whole process and calculate the average waiting time.

Good luck!