BEENG-320 Fall-2017

Homework #2 Due: Sep. 28, 2017

Instructions: Please submit a **SINGLE DOCUMENT (PDF)** on Blackboard. You are allowed to scan in handwritten work, as long as it is legible, but please be sure to combine everything, including MATLAB scripts and plots, into a single document.

Points: [10+20+15+15] = 60

Question 1

Consider a noisy ECG signal provided in the file ecg_noisy.mat. Load the signal in MATLAB and plot it as a function of time. The signal contains multiple R-peaks that have been masked by the a very high noise. We are interested in computing the RR-interval (interval between successive RR-peaks). Write a script that could be used to identify this RR-interval using a correlation based approach.

Question 2

a. Compute the Fourier Series coefficients of the periodic signal *x(t)* whose one period is defined as:

$$x(t) = \begin{cases} 1 & 0 < t < T/2 \\ -1 & T/2 < t < T/ \end{cases}$$

Consider now that T = 1 seconds. Plot five cycles of the signal as a function of time (x(t)) is essentially repeated five times), you can define a time axis from 0-5 seconds with increments of 0.0001 seconds for seeing a continuous plot.

As you would notice in your equation, you have infinite number of frequency coefficients as obtained from your solution above.

Now consider that you want to reconstruct the signal using the synthesis equation of the Fourier series but by limiting only to selected number of coefficients.

i) Reconstruct your signal using the first coefficient ($a_{\pm 1}$, i.e. k=1 and k=-1) of the Fourier Series. Write a MATLAB script that does this reconstruction. You can feed in the equation for Fourier Series coefficients you derived for the first part here to generate the synthesized signal $x_c(t)$. Plot $x_c(t)$ on the same curve as $x(t)$ and comment what do you observe.

ii) Repeat the same for the following cases, plot on the same curve using the hold on command in MATLAB.

   a. k = [-3:3], omit k = 0;
   b. k = [-10:10], omit k = 0;
   c. k = [-40:40], omit k = 0;

b. Find the Continuous-time Fourier Transform of the following signals and plot the magnitude spectrum.

i. $x(t) = \cos(8t) + \cos(20t)$

ii. Use linearity and frequency shifting properties of Fourier Transform to obtain FT of the following signal:

$$y(t) = \begin{cases} 1 + cos4t & -T_1 < t < T_1 \\ 0 & otherwise \end{cases}$$

Question 3

In this problem, we want to illustrate the process of abnormality (anomaly) detection for a biomedical signal. We consider the example of epileptic seizure detection using a EEG signal and some very basic processing to obtain a simplified block. This illustrative example is too simplified to work for detecting real epileptic signatures but will be modified later during the course with more complex processing.

Load the file "HW2_Q3_epilepsy_eeg.mat" which contains two EEG example signal variables *eeg_sig1* and *eeg_sig2*. The sampling frequency, f_s, used for acquiring these signals was 173.6 Hz.

You are provided a function file "eeg_epilepsy_detection_illustrator.m" which you need to complete for the tasks stated below.

Write a MATLAB script, HW2_Q3_sol.m which incorporates the following:

a) Your main file, will call the function "eeg_epilepsy_detection_illustrator(), so your need to complete the file, "eeg_epilepsy_detection_illustrator.m", containing a function that takes as its inputs two variables, "raw_sig" and "alphas" and outputs a variable, "labels". The goal of this function is to detect the epileptic seizure events at each sample of the input by setting a standard deviation based threshold. Your code will compute the standard deviation (sigma) of the "raw_sig" input to this function. Then it will compute an array of thresholds, one for each element of the input variable "alphas" by multiplying the "sigma" with "alphas" to obtain the variable of thresholds "thr". Now you will use the variable "thr" to detect the epileptic events by comparing the raw_signal at each sample against the "thr". The output variable "labels" is a binary number, 0 or 1, and should be set to "1" everytime the raw_signal goes beyond ±threshold (e.g if the threshold is 2, any time the raw signal is greater than 2 or less than -2, the labels is set to 1) and will be set to "0" otherwise. The expected size of the output variable is mentioned in the m-file (eeg_epilepsy_detection_illustrator.m) provided to you.

b) Your code should also plot on the same graph the raw_input (as function of time) and the different thresholds (as dotted constant lines for each of the thresholds). Incorporate this in the file "eeg_epilepsy_detection_illustrator.m".

c) Finally, you should output a graph of epileptic seizure events (number of 1s) detected for each threshold as a function of the "thr". Incorporate this in the file "eeg_epilepsy_detection_illustrator.m".

d) Once you have completed the file "eeg_epilepsy_detection_illustrator.m", call this function from your main file HW2_Q2_sol.m, as

   [labels] = eeg_epilepsy_detection_illustrator(eeg_sig1', alphas);

   for the following vectors alphas: alphas = [0.25 0.5 0.75 1 1.25 1.5 2 2.5 3 4 6];

Copy your code and the resulting plots and put them in the PDF file to be submitted with your homework.

Question 4: Template matching using cross-correlation in images

Cross-correlation can be used in images to automatically identify where in an image a given template exists. In this exercise, you will try and extract lesions in MR images automatically by matching (cross-correlating) a template to given MR images. You are provided a MATLAB file mr_lesions_sample.mat. It contains 4 MR images stored in variables: mr, mr_noise1, mr_noise2 and mr_noise3, where mr is the original MR image containing some lesions and others are the noise added versions of mr, where the SNR is progressively decreased. The template is stored in variable tmp. You would need to automatically locate this template within your 4 images using cross-correlation.

a. Plot these 4 MR images and comment on the relative SNR levels of the 4 images. (you may use imagesc() command in MATLAB, represent them as gray scale color intensities by setting the colormap property of imagesc()).

You will write a MATLAB function (or complete the provided function hw_q4_sample() that takes as its input one MR image and the template. The function does not need to return an output. Your function script will implement the following steps:

b. Compute the cross-correlation (2D) between the template and the MR image, you can write your own cross-correlation function or use MATLAB's xcorr2() function (see help xcorr2).
c. xcorr2 does not return normalized correlation, normalize your result in (b) by the maximum value of cross-correlation obtained in (b) so that the resultant normalized cross-correlation falls between 0 and 1. Plot this image of the normalized cross-correlation.
d. Set a threshold of 0.9 on this cross-correlation, and generate a binary mask of the normalized cross-correlation (part (c)) by applying this threshold. Anything above this threshold is set to 1 and below this threshold to 0. Display this mask as an image and analyze with respect to your original MR image where the non-zero points of this mark and what information they reveal about the original MR image that was input to your function.
e. Compute the centroids of the objects in your thresholded binary image (see the sample code provided using regionprops () function of MATLAB).
f. Find the number of objects detected by regionprops () using the length function.
g. If the number of objects is more than 2, display the statement "Input image may be too noisy". Note this limit is set for our problem here but not generically true.
h. Else if the number of objects is 1 or 2, xxtract the centroids for each of the object identified and subtract from the centroids half of the length of your template 'tmp'. Plot the original MR image provided as input to your function and overlay on the same image the two centroids with a pointer (**+**, see MATLAB plot () help for Markers.)

Test your completed function hw_q4_sample() on each of the 4 MR images. Include your code and all the figures generated in your submission.

Comment on the effect of noise on detecting the template using your method.