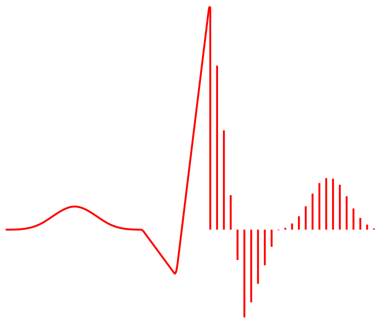
Biomedical Data Science Lab – EN.580.477
Fall 2019

Heart rate estimation, signal de-noising and mobile health



Benjamín Béjar Haro
Assistant Research Professor
Department of Biomedical Engineering
319B Clark Hall, Johns Hopkins University
E-mail: `bbejar@jhu.edu`

1 Lab Description

In this lab experience we will be dealing with the problem of heart rate estimation from noisy observations and we will illustrate its use in mobile health applications.

For a healthy individual, and under ideal conditions, this task might be easy to accomplish. However, different disorders together with different sources of error cause distortions to the signal rendering the estimation task more difficult. We will study how prior information about the signal and the interference can be exploited in order to improve the estimation task under less ideal conditions.

We will start detecting the QRS-complex assuming we have a template of the underlying waveform. Later, we will move to a more realistic scenario where we will be implementing a simplified version of the Pan-Tompkins algorithm [4] for the task of QRS-complex detection. Finally, we will illustrate how to estimate the heart rate from a video signal acquired using a smartphone.

1.1 Tasks

Task 1. *[40 points] (Signal generation and template matching)* In this task we will implement a simple version of a QRS-complex detector by using template matching, thresholding and non-maximum suppression on the observed ECG signal. We will then use those detections to provide an estimate of the heart rate.

(a) *(15 points)* In this first part we are going to generate a synthetic ECG signal. For that purpose we will be using the following model:

$$x(t) = \sum_{k=1}^{K} a_k \, \varphi(t - t_k) = \varphi(t) * \sum_{k=1}^{K} a_k \, \delta(t - t_k),$$

where a_k and t_k are the respective amplitudes and locations of the pulses defined by the waveform $\varphi(t)$. Generate a signal according to the above model consisting of $K = 10$ pulses equally spaced over a time window of 10 seconds and with a sampling rate of $f_s = 256$ Hz. As a canonical waveform use the function `ecg_wave` defined in your notebook.

(b) *(5 points)* Generate a noisy version of the synthetic ECG signal generated before by adding Gaussian noise with standard deviation $\sigma = 0.5$. Plot the noisy observations. Can you distinguish the locations of the QRS-complex?

(c) *(10 points)* Implement a QRS-complex detector on the noisy ECG signal you just generated. For that purpose we will use template matching, thresholding, and non-maximum suppression. We recommend that you normalize the signal before thresholding. For instance, you can normalize the signal to take values in the range $[0, 1]$ or you could standardize the signal by making it zero mean and unit variance. After normalization define a threshold value and keep only those values of the signal that are above the given threshold. Once you have done that, perform a non-maximum suppression (i.e., keep a value if it is greater than the previous and following values). To implement the thresholding and non-maximum suppression operations you can use the `find_peaks` function from the `scipy.signal` module.

(d) *(10 points)* Plot the original ECG signal and superimpose the locations of the peaks to verify your method. From the peak binary signal, estimate the RR interval sequence r_n and its average value as:

$$\bar{R} = \frac{1}{N}\sum_{n=0}^{N} r_n, \tag{1}$$

where N is the number of peaks detected. Based on the above quantity provide an estimate of the average heart rate in beats per minute.

Note: For the following two tasks you need to load data into Google Colab. Please, follow these steps in order to access your Google Drive from Colab.

1. Upload the datasets into your Google Drive (e.g., put them into "`My Drive/bmdslab/lab-01/`")
2. Run a cell with the following code:
```
from google.colab import drive
drive.mount('gdrive/')
```
3. Authorize access and copy-paste the authorization key
4. You can now access your Google Drive. Check by typing:
```
!ls gdrive/My Drive/bmdslab/lab-01/
```

After these steps, you should see the list of files at the specified location. Remember to use the correct path to your files when loading data.

Task 2. *[50 points] (QRS-complex detection)* Now let us consider a more realistic scenario where we have a noisy ECG signal and the waveform of the QRS-complex is unknown. There are different sources of noise that can be present during an ECG acquisition. In addition to high-frequency thermal noise, it is also common to observe the presence of low-frequency interference coming from breathing. When it comes to pathologies, different non-additive distortions might be present on the ECG signal that alter the shape of the QRS-complex itself but for the purpose of this task we just assume a healthy individual where signal distortion is additive and comes solely from the acquisition process. The procedure that we will employ in order to estimate the locations of the QRS-complex is based on the Pan-Tompkins algorithm [4].

(a) *(5 points)* Load the signal `ecg_mitnst.json` and plot the signal over time. The signal corresponds to a sample ECG from the MIT noise stress dataset [1] that has been downloaded from physionet.org [2]. Plot the signal and observe the presence of a strong low-frequency component.

(b) *(10 points)* Bandpass filtering. In this first step we want to eliminate as much as possible interference that is present in our acquired signal. In order to do that we can filter the signal and only leave the frequency range that is of interest for our task (e.g. $5-15$ Hz as suggested in [4]). We will split the bandpass filtering operation as the concatenation of a lowpass and a highpass filtering steps:

- *Highpass filtering*: A highpass filter can be thought of as removing the lower frequency component of the signal. That means that we could use a lowpass filter to get a lowpass version of the original signal and then subtract the result from the original series. We will use a lowpass filter of triangular shape and length L. Determine the length of the filter based on the required passband (i.e. the filter should attenuate significantly beyond 5 Hz) or empirically by adjusting the value of L. Plot the original signal together with the lowpass filtered version. The latter one should describe the trend in the ECG recording. Remember to normalize the filters (i.e., weights add up to one) to prevent signal amplification or attenuation. In an additional plot also display the signal with the lowpass version removed (high-pass filtered).
- *Lowpass filtering*: Use an averaging (box) filter to reduce high-frequency noise. Filter the signal with a filter of length $L = 10$ taps.

(c) *(5 points)* Differentiate the resulting signal in order to localize the region of steepest slope in the QRS-complex. You can use the finite difference filter suggested in [4] which is given by:

$$d_n = \frac{1}{8}\left(\delta_{n-2} + 2\delta_{n-1} - 2\delta_{n+1} - \delta_{n+2}\right)$$

(d) *(5 points)* Square the obtained signal after differentiation and plot the obtained waveform. Why was squaring helpful in revealing the peaks of the QRS complex?

(e) *(5 points)* Integrate the resulting signal from the squaring operation with a box window of length $L = 50$. Display the resulting signal.

(f) *(10 points)* Use a peak detector to estimate the locations of the peaks. Estimate the sequence of estimated RR intervals and plot the corresponding heart beat rate over time. Plot the locations of your estimated peak sequence and compare it with the ground truth.

Task 3. *[20 points] (Mobile health)* In this last task we will consider the estimation of the heart rate in a less controlled scenario. In this case, the signal comes from a mobile device and it is not a one-dimensional time-series but a video recording. Those with a smartphone can record and use their own recordings for the experiment. Alternatively, you can use the video provided for this purpose. The recorded signal consists of a video of the tip of the finger placed right in front of the camera while the camera flash is on. Play this video on your notebook and try to observe the intensity variations due to the heart beat.

(a) *(20 points)* Using the video signal propose a method to estimate the heart rate from it. Plot your estimated heart beat rate signal.

1.2 Python cheat sheet

Here goes a list of relevant python functions that you might want to use for this lab experience. This list in only intended to be illustrative. For a detailed description of the different functions, please refer to the particular package documentation.

Command	Description
`import numpy as np`	Imports numpy module with name np
`import matplotlib.pyplot as plt`	Imports module with name plt
`plt.step(x,y)`	Staircase-style plot.
`np.convolve(x,y,mode='same')`	Convolves x and y and keeps central part of convolution (same length as x).
`np.correlation(x,y,mode='same')`	Correlates x and y and keeps central part of correlation sequence (same length as x).
`np.mean(x), np.std(x)`	Computes mean/std value of x.
`np.where(x)`	Returns non-zero support of x.
`np.diff(x)`	Computes the one-sample finite difference of x.
`np.arange(N)`	Gives a list of numbers $0, \ldots, N-1$.
`np.fft.fft(x), np.fft.ifft(x)`	Computes the (inverse) FFT of x.
`np.abs(x)`	Computes the absolute value of x.
`scipy.signal.find_peaks(x,T)`	Find peaks in x above threshold T.

References

[1] W. E. M. G. B. Moody and R. G. Mark. A noise stress test for arrythmia detectors. *Computers in Cardiology*, 11:381–384, 1984.

[2] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. I. andn RG Mark, J. Mietus, G. Moody, C.-K. Peng, and H. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101(23):e215–e220, June 2000.

[3] G. B. Moody and R. G. Mark. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, May 2001.

[4] J. Pan and W. J. Tompkins. A Real-Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, March 1985.