|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | **1** | **2** | **3** | **4** |
| Marks | **5** | **10** | **5** | **5** |
| Total | **25** | | | |

**Name (***as in registration***):** Kar Chaudhuri Anirban

**Course:** Intelligent Sense and Sense Making

**Date:** 06 July 2020

**Written Test**

**Q1.** A local startup is trying to invent a new medical device that can analyze a particular type physiological signal. However, they realize the signal captured by the device is distorted. Research reported that the particular type of physiological signal has a frequency range of 0 to 200Hz. And the device is designed to do only 100 times of sampling per second. What is the issue? How can that be resolved?

According to the Nyquist-Shannon theorem, the sampling rate must be at least twice the highest original analog frequency component. Otherwise, some of the highest frequency components in the analog input signal will not be correctly represented in the digitized output. In this case, ideally, the sampling frequency should be at least 400Hz, twice that of maximum 200Hz frequency. The device needs to be modified to sample at this rate.

**Q2.** Given any 1-dimensional signal, it is always possible to use forward difference, backward difference and central difference to determine peaks and trough. However, in general this strategy is not adopted. Why? How can we detect peaks and trough in Python?

The peaks can be false positives due to noise generated due to sensor problems. Unstable baselines in sensor values can be disturbed by external circumstances like temperature, humidity, etc. This distorts the actual trend supposed to be observed in signal.

In addition, the peaks identified using differencing not guaranteed to be relevant/desirable for analysis based on physical and business context.

The peaks identified at first sight may not be truly representing proper anomalous instances.

Baseline correction needed for obtaining ideal signal.

For peaks and throughs detection in python, run findPeaks function/model from scipy library after downloading the sensor data. Finetune findPeaks model. Take note of prominence of a peak that measure how much a peak stands out from baseline of the signal.

.

**Q3.** Baseline wandering is a common problem in ECG signals, and often baseline correction is needed. Why do we need to do baseline correction and what is the benefit of doing that?

Baseline correction is an important pre-processing technique used to separate true spectroscopic signals from interference effects or remove background effects and disturbances like temperature, humidity, stains, etc. Unstable baselines in sensor values can be disturbed by external circumstances. This distorts the actual trend supposed to be observed in signal.

By correcting for the baseline variation, improved agreements can be obtained for ECG readings that represent accurate heartbeat pattern of patient. Accurate results for heart disease analysis and diagnosis for a patient important to determine proper lifesaving approaches.

**Q4.** There are times when we need to compare signals, but in general we avoid using Euclidean distance or Manhattan distance as the measures to do comparison. Instead dynamic time warping is the preferred solution. Why?

Even if two signals have similar length, Euclidean or Manhattan distances cannot account for shape similarity if numeric distance similarity score is strong. Dynamic Time warping takes into account rate of change of signal with respect to time apart from numeric distance, that allows comparison of shape between reference and test signal patterns. This is achieved by DTW algorithm warping the signal path. Time alignment is reliable. The warping path step in algorithm stresses on distance, accumulated cost and optimal path that determines optimal signal path by penalizing high deviation.