Homework 2

Exercise 1

Exercise 1b

Overall, the Manhattan distance takes the highest values, while for increasing w the DTW takes increasingly lower values. Considering all DTW and Manhattan distances computed, the lowest values are always observed when comparing two ECGs within the normal group. This is to be expected as a normal ECGs would fit a certain defined pattern, oppositely to ECGs in the abnormal group which can take different forms. This fits with the obtained results, as for most of them the highest distances are seen when comparing series within the abnormal group. However, comparisons between the normal and the abnormal heartbeats yield different results when reported to the comparisons for other groups. Namely, for the Manhattan and three of the DTW distances (w = 0, 25 or infinity) the value is close to the one obtained when looking at ECGs within the abnormal group. The DTW for w=10 is the only distance that can successfully separate ECGs in the abnormal group from the ones in the normal group. This renders DTW when the parameter w is 10 as the best distance to be used in practice since it would enable the investigator to assess the provenience of the ECGs compared (two normal, two abnormal, an abnormal and a normal one).

Exercise 1c

As the hyperparameter w is increased the average DTW distance over the different group comparisons decreases to a certain extent (in this case w=25). The effect of the hyperparameter w is related to restrictions imposing the maximum distance (number of time points) allowed between two mapped values. When w=0, the DTW distance is equal to the Manhattan distance as each entry is mapped to the respective one in the other time series which represents the case of maximum constraint. Similarly, w=infinity, no restriction is imposed on how points in one series are mapped on the other series as long as they respect the DTW norms which is the case of minimum constraint. When w=10, the DTW manages to best separate the normal and abnormal heartbeats. This moderate constraint allows the algorithm to search for the best alignment for a time point within a space of ten entries from the other time series. A less strict constraint (w=25, infinity) results in a worse alignment and thus separation which is also the result of a rigorous constraint.

Exercise 1d

The DTW distance is not a metric as it does not fulfill the triangle inequality: d(x,y)≤d(x,z)+d(z,y). Given x(0,1,1,2), y(0,2,2) and z(0,1,2), the DTW matrices would be the following:

- (x,y)

$$\begin{matrix} 0 & \infty & \infty & \infty \\ \infty & 0 & 2 & 4 \\ \infty & 1 & 1 & 2 \\ \infty & 2 & 2 & 2 \\ \infty & 4 & 2 & 2 \end{matrix}$$

DTW(x,y) = 2

- (x,z)

$$\begin{matrix} 0 & \infty & \infty & \infty \\ \infty & 0 & 1 & 3 \\ \infty & 1 & 0 & 1 \\ \infty & 2 & 0 & 1 \\ \infty & 4 & 1 & 0 \end{matrix}$$

DTW(x,z)=0

- (z,y)

$$\begin{matrix} 0 & \infty & \infty & \infty \\ \infty & 0 & 2 & 4 \\ \infty & 1 & 1 & 2 \\ \infty & 3 & 1 & 1 \end{matrix}$$

DTW(z,y)=1

Therefore, for the triangle inequality, the expression would be 2 ≤ 0+1 => 2 ≤ 1 which is false. As DTW does not fulfill the triangle inequality, DTW distance is not a metric. This is example also shows that the DTW distance between x and z is 0, even if the two are not equal so it does not fulfill d(x,z)=0 IIF x=z.

Exercise 1e

Considering w-constrained warping, the runtime is equal to O(n*(w+1)). However, if there are no constraints the runtime would be O(n^2) as any point in the first time series could be aligned to any other point in the second one which would require iteration through all options. In the case of w-constrained warping, the search is reduced to a smaller number of options which causes the reduction of the runtime.

Exercise 2

Exercise 2c

To compute the shortest path lengths using the Floyd-Warshall algorithm, a runtime of $O(n^3)$ is required as three iterations through the nodes of the graph are needed for calculating and updating the shortest path length. The runtime complexity of the SP is $O(n^2*m^2)$, with n and m being the number of nodes in the two graphs that are being compared. An option for improving the runtime of the SP would be iterating separately through the upper triangular matrix of the two Floyd-transformed matrices and using a dictionary to count the number of occurrences for each different shortest path length. For shortest paths lengths that appear in both matrices, the counts would be multiplied and added to the shortest path kernel value. In this way the runtime complexity would become $O(n^2+m^2)$.