Similarity measures on time series and graphs

Andreas Dahlberg

October 22, 2018

Exercise 1

Exercise 1c) and 1d)

In the file *timeseries_output.txt* we see that for each row the Manhattan distance between the time series is exactly the same as the DTW-distance with $w = 0$. This is also what we expect since the DTW-distance with the constraint $w = 0$ means that in the distance matrix we want to find the shortest path from the upper left corner to the bottom right corner while fulfilling $i = j$, i.e only staying on the diagonal. This shortest path is the same as the Manhattan distance. We can also note that for each row, the DTW-distances decreases with increasing w. This is because the constraint $|i - j| \leq w$ means that we want to find the shortest path form upper left corner to bottom right corner while staying close to the diagonal. So if we take w' such that $w' > w$ then the shortest path under the constraint $|i-j| \leq w$ is also a path under the constraint $|i-j| \leq w'$ which means that the length of shortest path under $|i - j| \leq w'$ is at most the length of the shortest path under $|i - j| \leq w$.

We can see that the distance between the abnormal group and the normal group is lower than the distance between the abnormal group and itself for DTW with $w = 0$ and DTW with $w = 10$. This is not a good result since we want small distances between the same type of groups and large distances between different types of groups. However, when $w = 25$ and $w = \infty$ the distance between the abnormal group and the normal group is larger than the distance between the abnormal group and itself and the distance between the normal group and itself. We can also see that it seems as if $w = \infty$ gives the best seperation between groups.

Exercise 1d)

In this exercise we want to determine if the dynamic time warping distance is a metric on the set of all time series with values in $\mathbb{R}$. We define two time series to be equal if and only if their corresponding vector representations are equal. First, it is easy to see that $DTW(t_1, t_2) = DTW(t_2, t_1)$ and that $DTW(t_1, t_2) \geq 0$. However, we do not have that $DTW(t_1, t_2) = 0$ if and only if $t_1 = t_2$. For instance, we can take $t_1 = (1, 1, 1, 1)$ and $t_2 = (1)$, we then have that $DTW(t_1, t_2) = 0$ even though $t_1 \neq t_2$. The DTW-distance doesn't either fulfil the triangle inequality. For example, we can take $X = (1, 2, 3)$, $Y = (1, 2, 2, 3)$ and $Z = (1, 3, 3)$. Then we have that $DTW(X, Y) = 0$, $DTW(X, Z) = 1$ and $DTW(Y, Z) = 2$. So we do not have that $DTW(Y, Z) \leq DTW(Y, X) + DTW(X, Z)$. This proves that the DTW-distance is not a metric.

Exercise 1f)

The time complexity of running normal DTW is $O(n^2)$ and it is the same for w-constrained DTW. This is because when creating the distance matrix it takes $O(n^2)$ operations. It is possible to determine $DTW[i, j]$ for all cells satisfying $|i - j| \leq w$ in $O(wn)$ if you only loop through the cells that satisfy this condition, i.e close to the diagonal. So w-constrained is faster than normal DTW but it is still $O(n^2)$ since you have to initialize the matrices.

1 Exercise 2)

Exercise 2c)

If A is the adjacency matrix of a graph with n nodes then Floyd-Warshall runs in $O(n^3)$. Initialisation of the matrix D is $O(n^2)$ and the following nested for-loops takes $O(n^3)$ since there are three nested for-loops and the time to get an element from the matrix D is $O(1)$.

Now suppose we want to calculate the time complexity of $spkernel(S1, S2)$ where S1 is an $n \times n$ matrix and S2 $m \times m$ matrix. To create the Counter-object for $S1$ takes $O(n^2)$ and to create the Counter-object for $S2$ takes $O(m^2)$. This is because one has to loop over the entire upper right part of the matrix. This is under the assumption that the operations for the underlying hashmap is about $O(1)$. The following for-loop is $O(k)$ where k is the number of keys in $c1$, the Counter object for $S1$. Since $k \leq n^2$ this loop runs in $O(n^2)$. So $spkernel(S1, S2)$ runs in $O(max(m, n)^2)$ and hence the time complexity to measure the similarity between two graphs using this method is $O(max(m, n)^3)$ (First Floyd-Warshall and then SPkernel).