# Data Mining - Exercise 2

Luca Witte

2022-10-31

## Similarity measures on time series and graphs

### 1. Time series

Here the "ECG" dataset of heartbeat monitoring is analysed using dynamic time warping (DTW). Heartbeat data of the same length were manually annotated with "normal" and "abnormal". DTW is used to compare 100 samples.

More specifically, a $w$-constrained DTW variant is used. An additional user-defined hyperparameter constricts the number of possible warping paths, increasing the computational efficiency of the algorithm.

Table 1: Results obtained from running the attached python script. The script calculates distances via dynamic time warping (different w-constraints) and the Manhattan distance.

| Pair of classes | Manhattan | DTW (w = 0) | DTW (w = 10) | DTW (w = 25) | DTW (w = inf) |
|---|---|---|---|---|---|
| abnormal:abnormal | 67.77 | 67.77 | 38.65 | 26.48 | 25.37 |
| abnormal:normal | 67.52 | 67.52 | 34.20 | 26.94 | 26.35 |
| normal:normal | 45.65 | 45.65 | 24.42 | 22.17 | 21.87 |

#### 1 b) & c)

As can be seen in *table 1*, the Manhattan distance and the DTW algorithm with $w = 0$ yield the same results. This is expected, as $w = 0$ means, that only the values at the same time point are compared. As the Manhattan distance is used in the calculation of DTW, the results are equivalent. By increasing the parameter $w$, all distances decrease. As time warping is allowed over wider ranges, peaks and troughs can be matched more reliably between the different data series. Therefore distance is decreased. The biggest relative decrease is observed when $w$ is increased from 0 to 10, implementing time warping. Further increasing $w$ decreases the distance further, but the relative effect is smaller. Especially from 25 to infinity (meaning the distance between every possible combination of time points is considered), the effect is neglectable.

While the increase in $w$ appears to allow to match the time series better over time, it can be assumed that cost of calculation also increases with higher w (more combinations of elements need to be considered). Therefore, the choice of parameter $w$ constitutes a trade off between accuracy and efficiency. Depending on the data and required accuracy, higher values for $w$ should be chosen to increase accuracy. If the time scales between data sets are expected to be not very far off, lower $w$ values can be chosen to decrease computational complexity. In any case, an increase to $w = 10$ or $w = 25$ seems a good compromise in the given example.

Higher $w$ values show a lower relative deviation of the abnormal:normal group to the intra-group comparisons. This makes sense, considering the higher flexibility in the dynamic time warping with increased $w$. This allows to associate even highly different data sets more reliably by more extreme warping. This decreases the deviation between different sets. To compare specific properties, a high degree of warping might be helpful. Though, for a classification (e.g. “normal” and “abnormal”), higher distances allow to define separating conditions more exactly. In this case, a lower $w$ might result in a better separation. Therefore the implemented parameter value should be chosed depending on the desired purpose.

### 1 d)

As discussed in exercise set 1, a metric is defined by following four specific conditions. In the case of dynamic time warping, the condition of triangle inequality is not fulfilled:

$$d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$$

A simple example is given below:

$$\begin{aligned} x_1 &= [0, 8, 8, 12] \\ x_2 &= [0, 8, 12] \\ x_3 &= [0, 12, 12] \end{aligned}$$

Applying DTW yields:

$$\begin{aligned} DTW_d(x_1, x_2) &= 0 \\ DTW_d(x_2, x_3) &= 4 \\ DTW_d(x_1, x_3) &= 8 \end{aligned}$$

with:

$$DTW_d(x_1, x_3) > DTW_d(x_1, x_2) + DTW_d(x_2, x_3)$$

This shows that the dynamic time warping cannot be considered a metric.

### 1 e)

As the algorithm without constraints requires looping through both vectors (of lengths $n$ and $m$), the complexity is on the order of $\mathcal{O}(n * m)$. For vectors of the same length $n$, the complexity is $\mathcal{O}(n^2)$.

With w-constrained DTW, the number of elements is reduced depending on the hyperparameter $w$ in the range of $0 \leq w < n$ (for both vectors length $n$). For $w \geq n$, the complexity is $\mathcal{O}(n^2)$. For $w = 0$, we obtain a complexity of $\mathcal{O}(n)$.

## 2. Graphs

The used data set contains 188 chemical compounds that were screened and annotated for mutagenic effects. Using Floyd-Warshall’s algorithm, shortest paths within the graph-representations of those compounds are calculated. The groups “mutagenic” and “non-mutagenic” are compared using the shortest-paths Kernel.

Table 2: Results obtained from running the attached python script. The script calculates the shortest path matrix from an adjacency matrix using Floyd-Warshall’s algorithm. Matrices of the groups mutagenic and non-mutagenic were compared using the Shortest Path kernel.

| Pair of classes | SP Kernel similarity |
|---|---|
| mutagenic:mutagenic | 5268.30 |
| mutagenic:non-mutagenic | 2705.81 |
| non-mutagenic:non-mutagenic | 1471.34 |

It is evident that the mutagenic:mutagenic comparison has a higher average similarity than both other pairs. This could point to a single or a few shared structural components that are responsible for the mutagenic effects. Different molecules might have a variety of modifications, but are likely to share functional groups responsible for the biological effect. The non:mutagenic group therefore can be expected to be much more heterogeneous, as most chemical structures are not strongly mutagenic. Comparing both groups shows that some structures are shared, but less frequently than within the mutagenic group. These groups occuring in both groups might not influence mutagenicity, therefore they would not contribute to separation. The high similarity within the mutagenic group might point to specific molecular building blocks that confer effects like modification of the DNA backbone or bases, but also membrane permeability and other secondary factors.

### 1 c)

As we use three variables to iterate through all $n$ elements of the adjecency matrix three times, the complexity of Floyd-Warshall's algorithm is on the order of $\mathcal{O}(n * n * n) = \mathcal{O}(n^3)$. The runtime complexity of the shortest path Kernel algorithm is determined by the comparison of all $m$ edges in the two shortest path matrices, yielding a complexity of $\mathcal{O}(m^2)$.

The Floyd Warshall algorithm is already a highly efficient algorithm to identify all pairs of shortest paths in a graph. Other algorithms like Dijkstra or Johnson's algorithm offer an improved performance in specific cases, but from the authors understanding do not have a lower complexity in every case. As the Floyd Warshall algorithm is versatile and fast, it is a good choice for an introductory lecture.

The same case can be made for the shortest path kernel, which is already an improvement over other metrics. By abusing the symmetry of the matrix and only using the triangular matrix as an input, the runtime is already improved. As graphs in biology often have distinct properties (e.g. scale-free network structures), *(Borgwardt and Kriegel, 2005)* proposed it might be useful to develop algorithms that are adapted to a specific tasks that make use of those specific properties. They further propose to reduce runtime by storing additional information like the number of edges and setting the kernel value to 0 when the compared matrices vary in these metrics.

It needs to be considered that by analyzing shortest paths, a loss of information (stored in the whole path) can occur. Therefore, besides improving runtime, the accuracy might be considered when chosing an appropriate method.