DATA MINING 2

Time Series - Similarities & Distances

Riccardo Guidotti

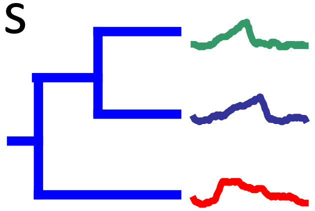
a.a. 2025/2026

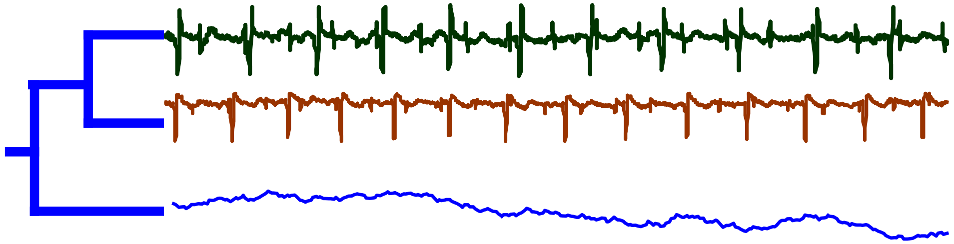
Slides edited from Keogh Eamonn's tutorial

UNIVERSITÀ DI PISA

Distances and Similarities

- Time series problems such as classification, forecasting, clustering, etc. require the usage of a notion of distance or similarity.
- What is similarity?
- It is the quality or state of being similar, likeness, resemblance, as a similarity of features.
- In TSA we recognize two types of similarity measures depending on the data representation considered:
  - **shape-based similarity**
  - **structural-based similarity**

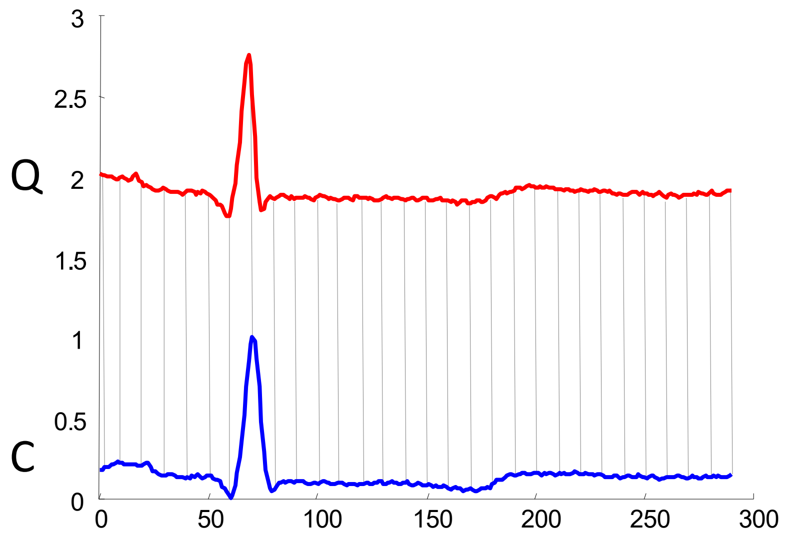




Shape vs Structural Similarities

Shape-based Similarity

- The original values of the time series are compared taking time into account.
- Better for short time series.



Structural Similarity

- Time series are transformed into an alternative representation where the novel features are time-independent.
- Better for long time series.

	min	max	mean	std
Q	1.8	2.9	2.0	1.3
C	0.0	1.0	0.2	1.2

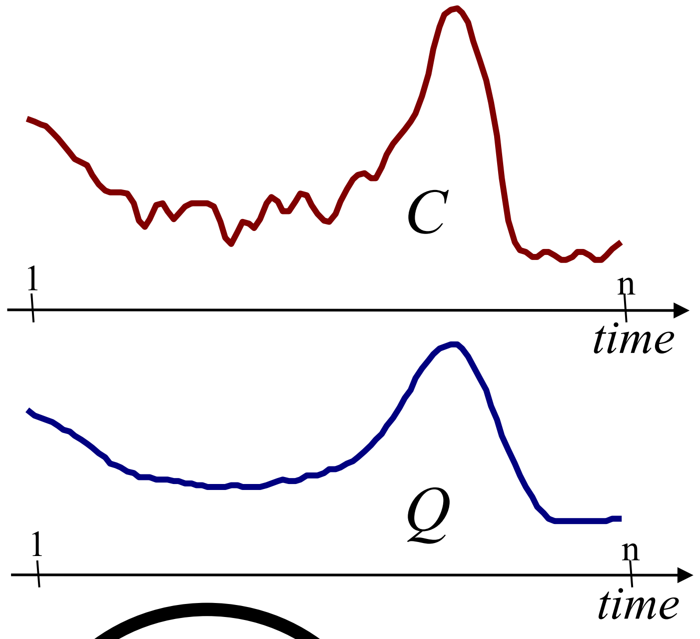
Euclidean Distance

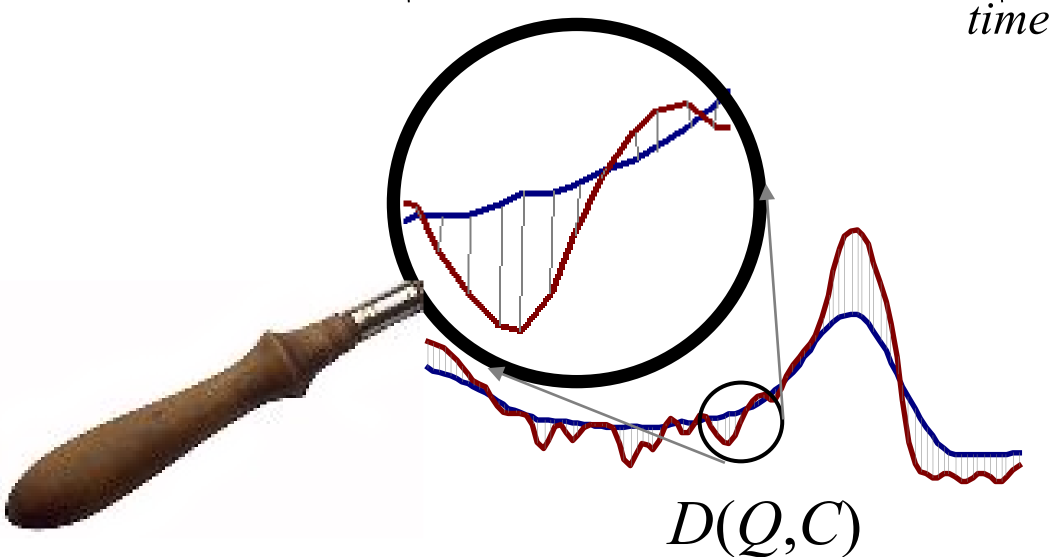
- Given two time series:
  - $Q = q_1 \dots q_n$
  - $C = c_1 \dots c_n$

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2}$$

- T1 = < 56, 176, 110, 95 >
- T2 = < 36, 126, 180, 80 >

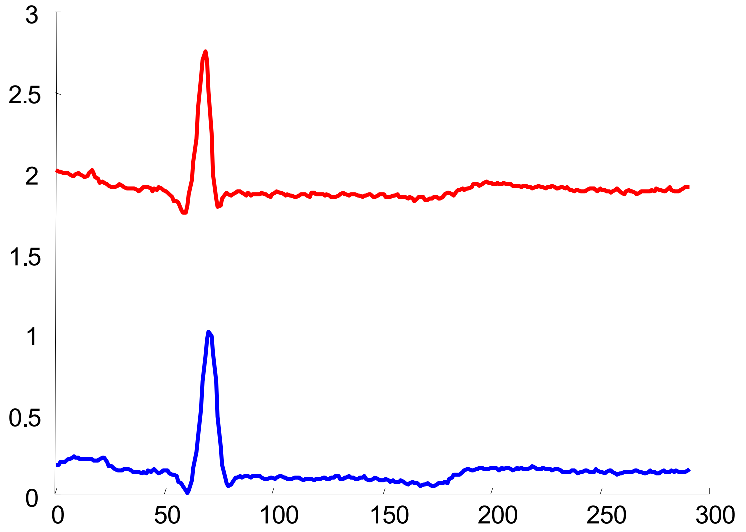
D(T1,T2) = sqrt [$(56\text{-}36)^2 + (176\text{-}126)^2 + (110\text{-}180)^2 + (95\text{-}80)^2$]





Problems with Euclidean Distance

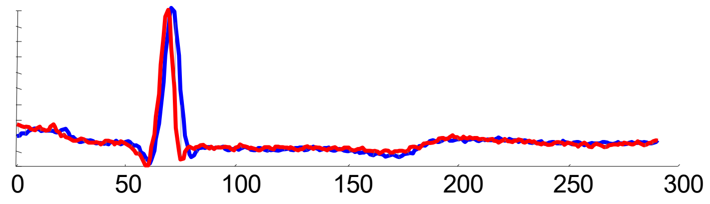
- Euclidean distance is very sensitive to "distortions" in the data.
- These distortions are dangerous and should be removed.
- Most common distortions:
    - Offset Translation
    - Amplitude Scaling
    - Linear Trend
    - Noise
- They can be removed by using the appropriate normalization.



$Q = Q - mean(Q)$

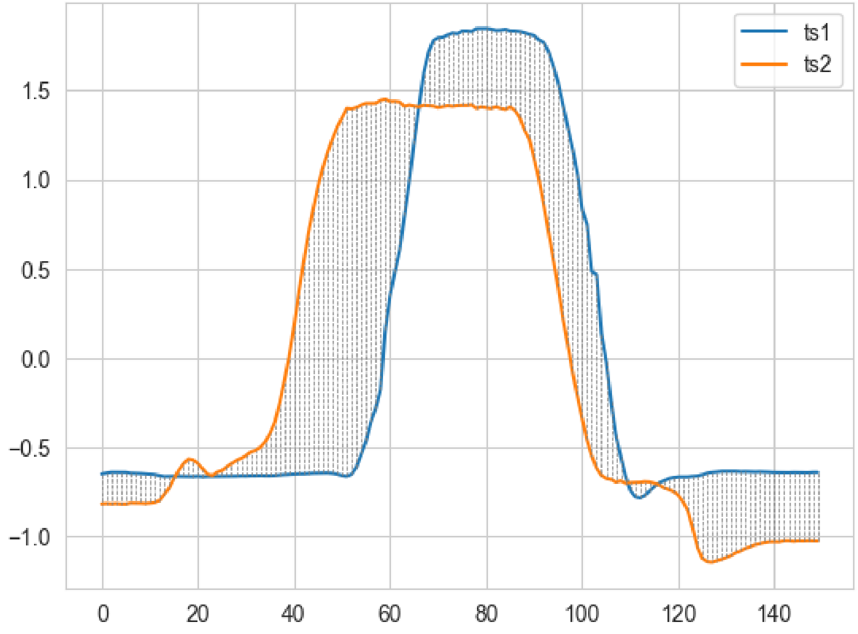
$C = C - mean(C)$

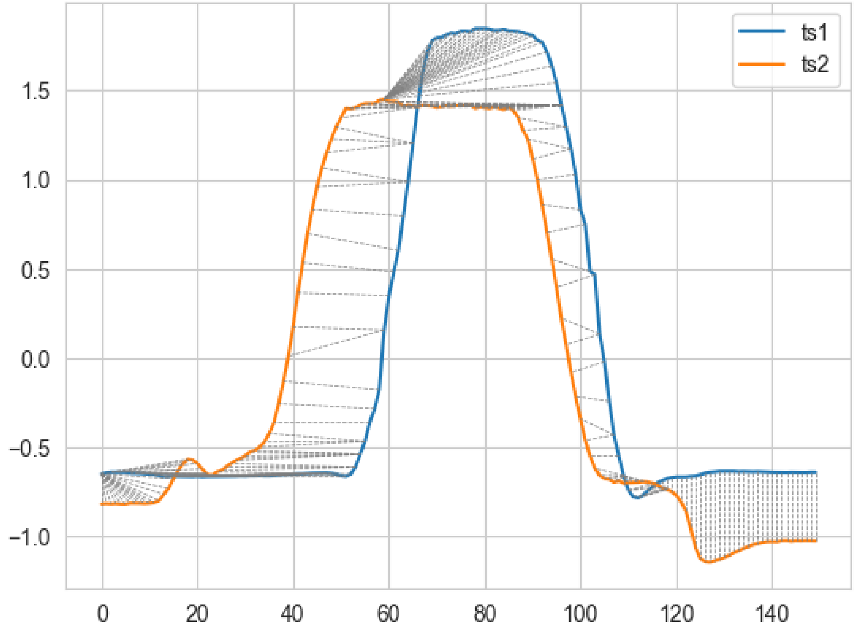
$D(Q,C)$



Further Problems with Euclidean Distance

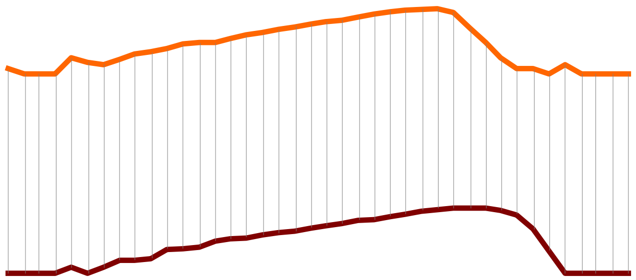
- Even after normalization, the Euclidean distance may still be unsuitable for some time series domains since it does not allow for acceleration and deceleration along the time axis.



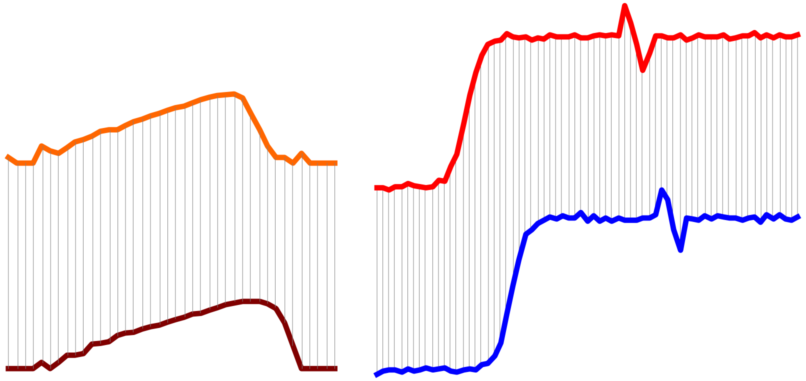


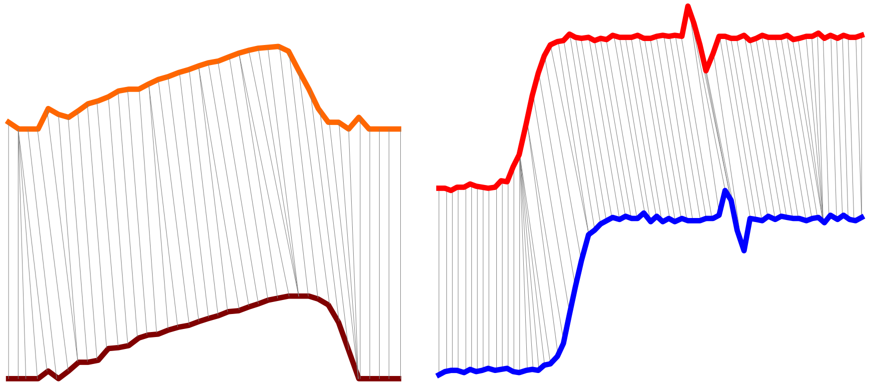
Dynamic Time Warping

- Sometimes two time series that are conceptually equivalent evolve at different speeds, at least in some moments.

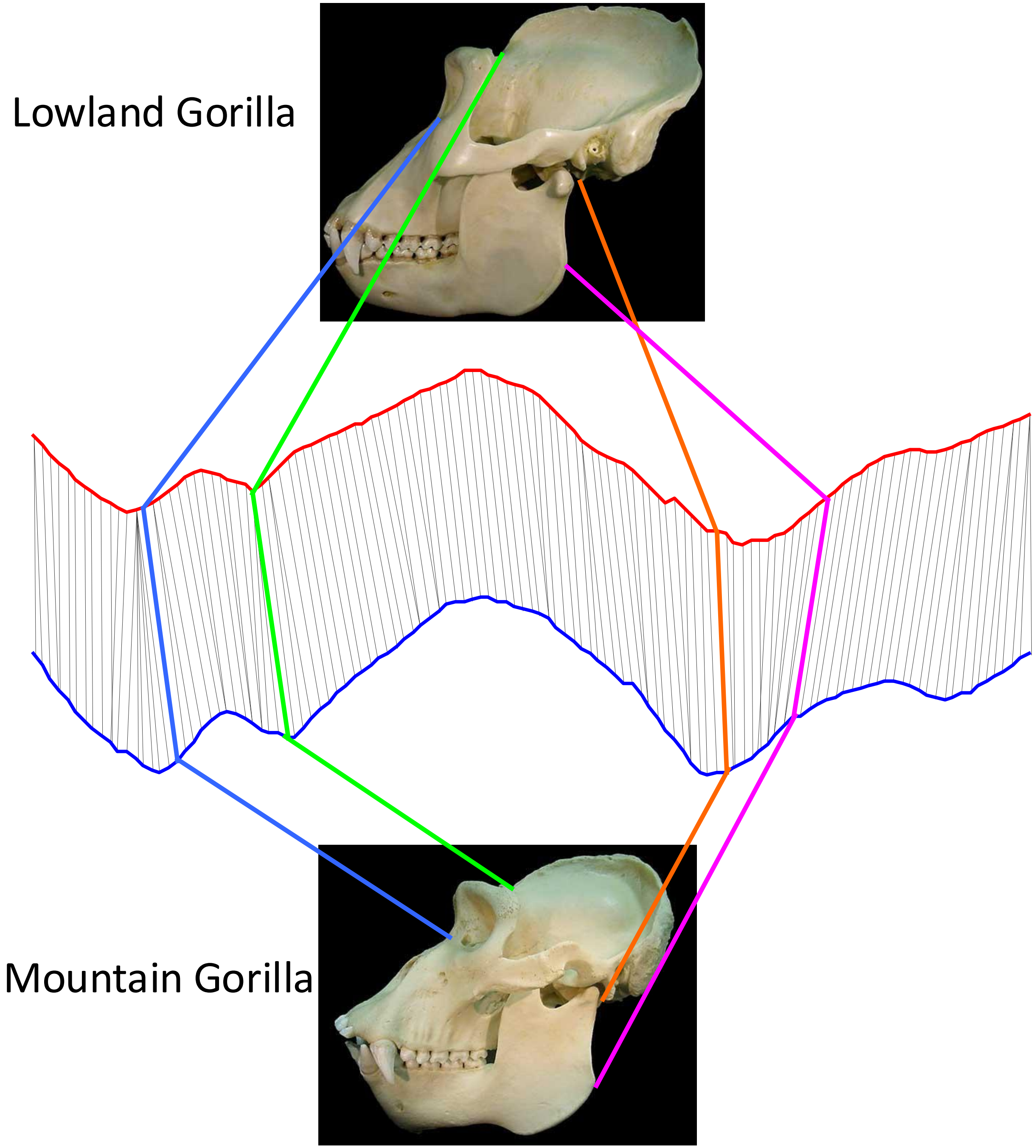


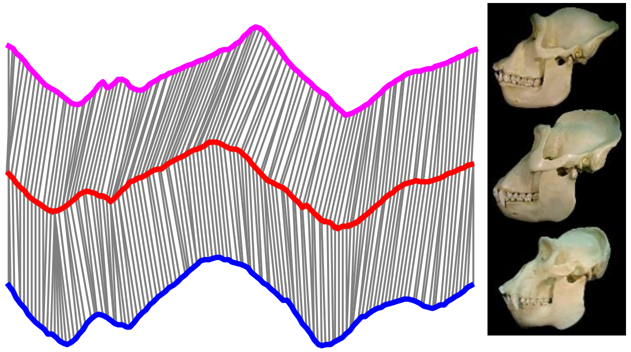
E.g. correspondence of peaks in two similar time series





- **Euclidean distance - Fixed Time Axis**: Sequences are aligned “one to one”. Greatly suffers from the misalignment in data.
- **Dynamic Time Warping - Warped Time Axis**: Nonlinear alignments are possible. Can correct misalignments in data.

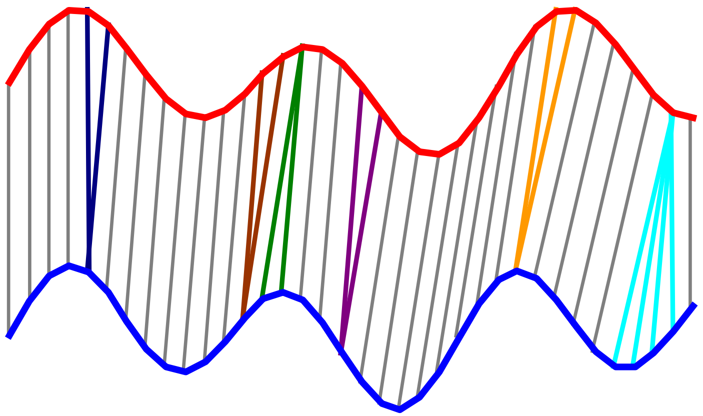


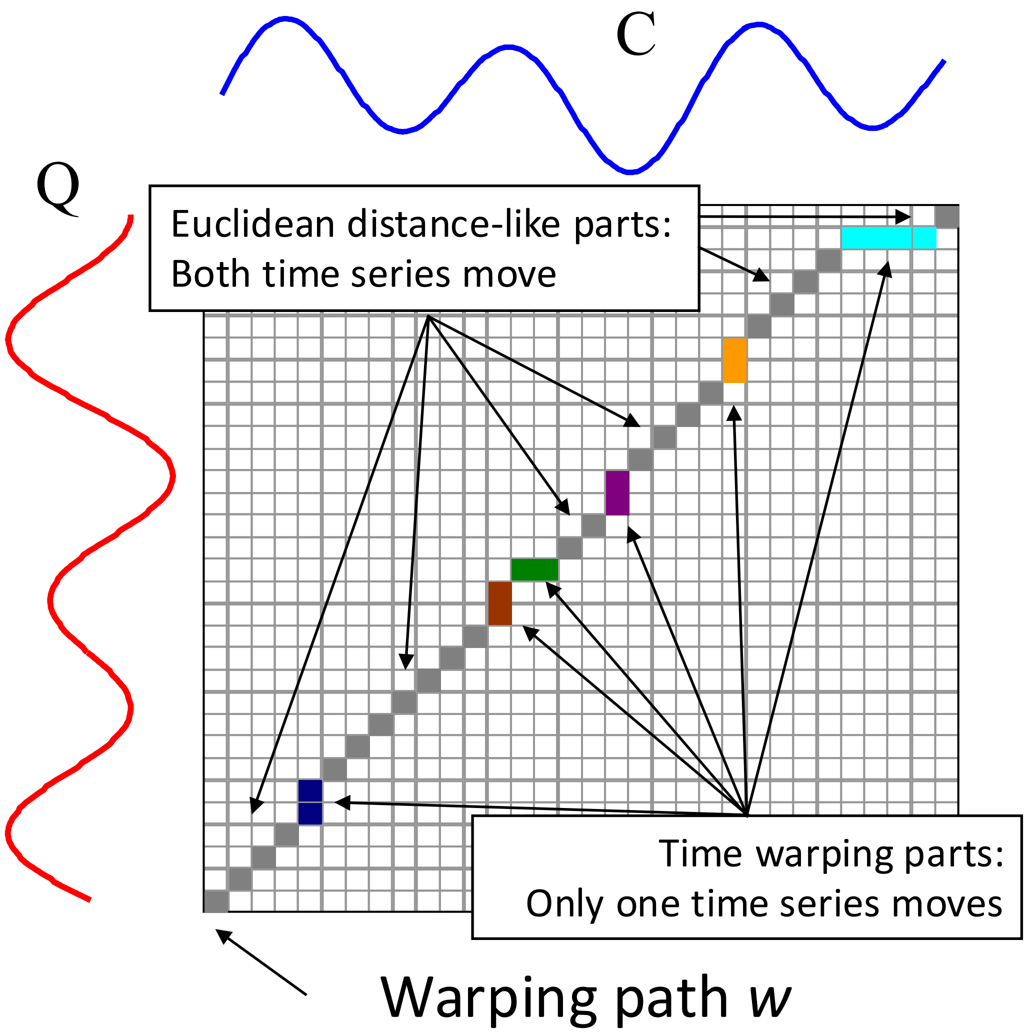


https://izbicki.me/blog/converting-images-into-time-series-for-data-mining.html

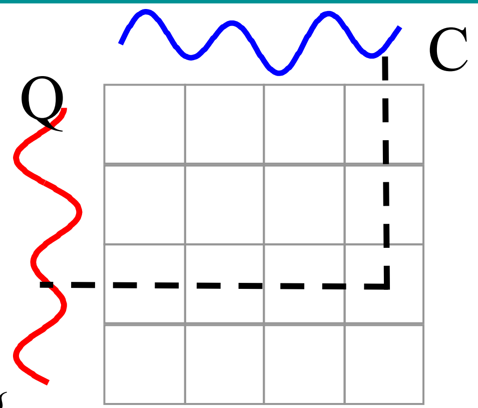
How is DTW Calculated?

- Every possible warping between two time series, is a path through the matrix.
- The constrained sequence of comparisons performed:
  - Start from pair of points *(0,0)*
  - After point *(i,j)*, either *i* or *j* increase by one, or both of them
  - End the process on *(n,m)*





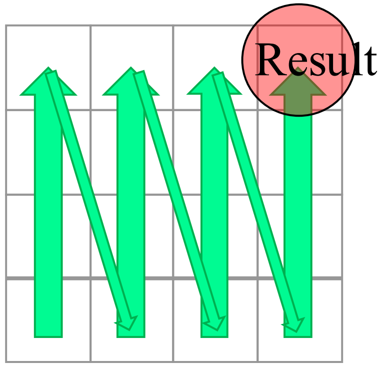
Dynamic Programming Approach



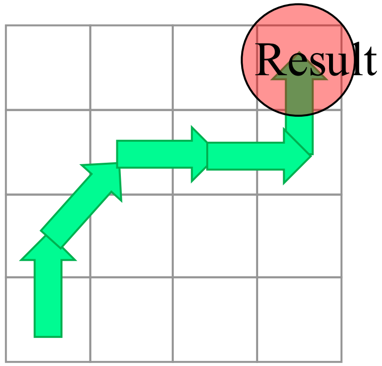
Step 1: Compute the matrix of all *point-to-point* distances $d(q_i,c_j) = | Q_i - C_j |$

Step 2: Compute the cumulative cost matrix as $\gamma(i,j) = d(q_i,c_j) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$

- Start from cell *(1,1)*
- Compute *(2,1), (3,1), …, (n,1)*
- Repeat for columns *2, 3, …, n*
- Final distance value is in the last cell computed



Step 3: find the path with the lowest values, i.e., the best alignment between Q and C



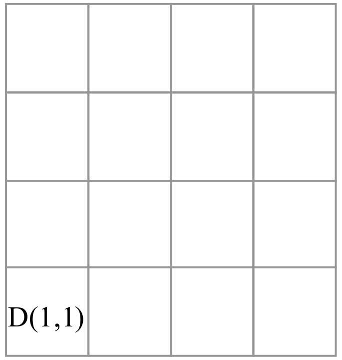
Dynamic Programming Approach

$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i-1,j-1), \gamma(i-1,j\,), \gamma(i,j-1) \,\}$

Step 2: compute the matrix of all path costs $\gamma(i,j)$

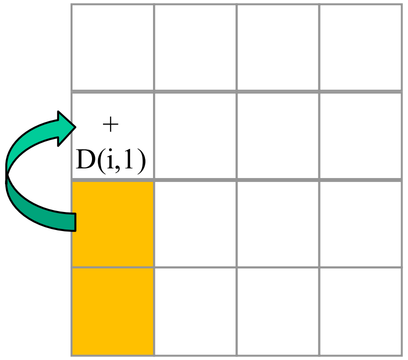
- Start from cell *(1,1)*

$$\begin{aligned}\gamma(1,1) &= d(q_1,c_1) + \min\{\cancel{\gamma(0,0)}, \cancel{\gamma(0,1)}, \cancel{\gamma(1,0)}\} \\ &= d(q_1,c_1) \\ &= D(1,1)\end{aligned}$$

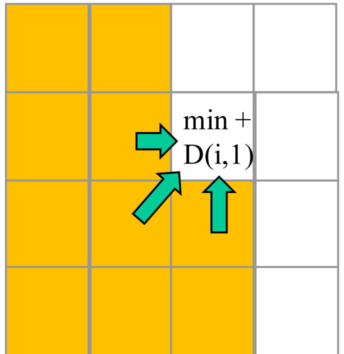


- Compute (2,1), (3,1), …, (n,1)

$$\begin{aligned}\gamma(i,1) &= d(q_i,c_1) + \min\{\cancel{\gamma(i-1,0)}, \gamma(i-1,1), \cancel{\gamma(i,0)}\} \\ &= d(q_i,c_1) + \gamma(i-1,1) \\ &= D(i,1) + \gamma(i-1,1)\end{aligned}$$

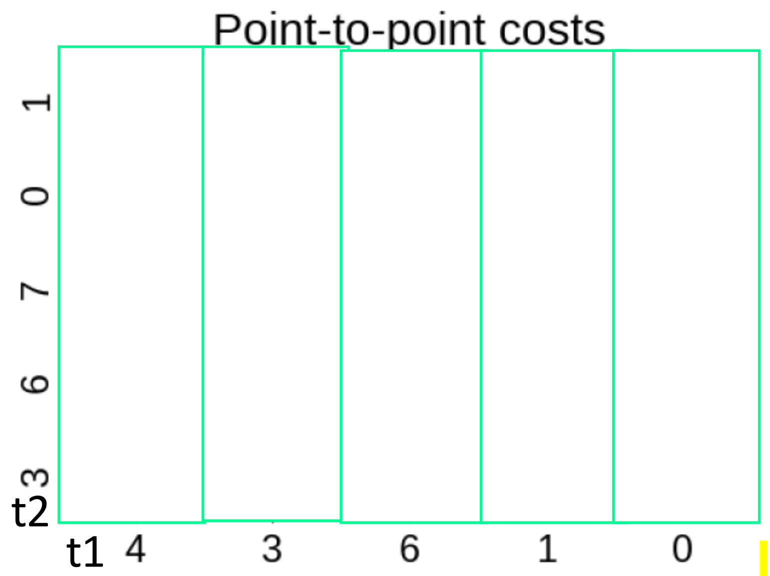


- Repeat for columns *2, 3, …, n*
  - The general formula applies



DTW – Example

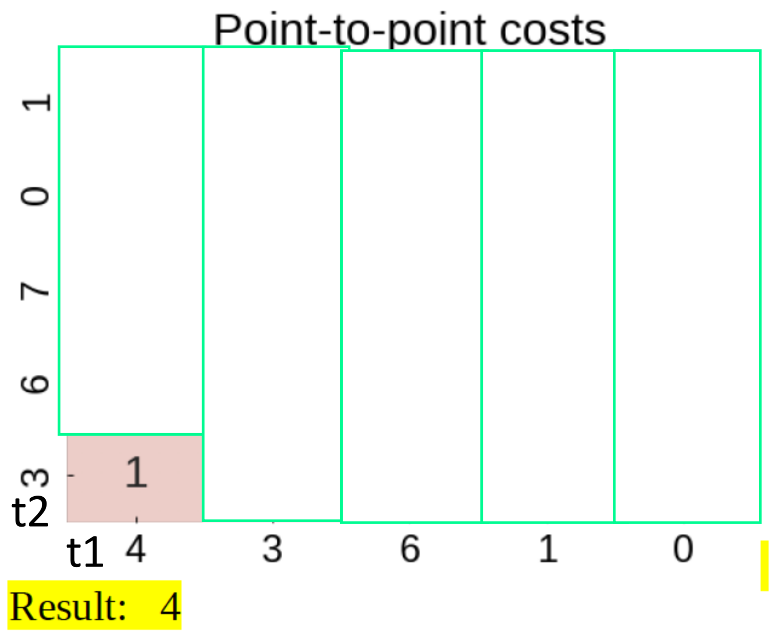
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



Result: 4

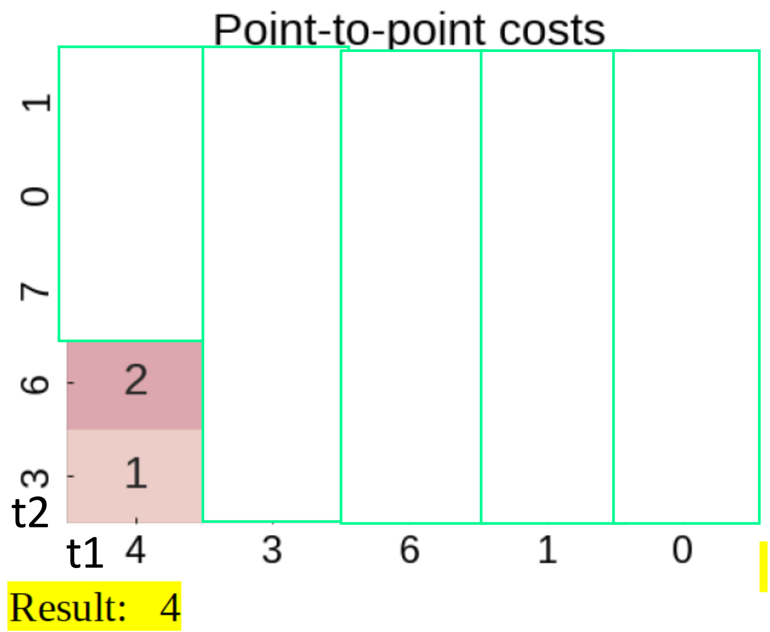
DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



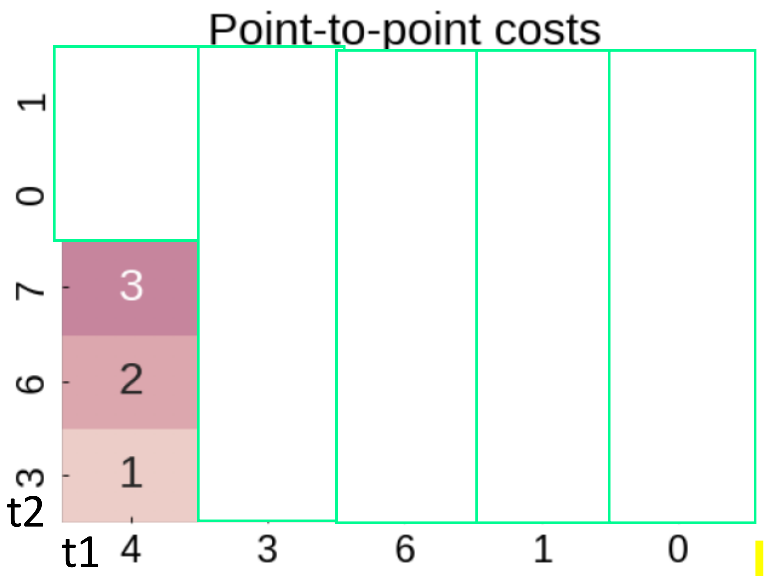
DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



DTW – Example

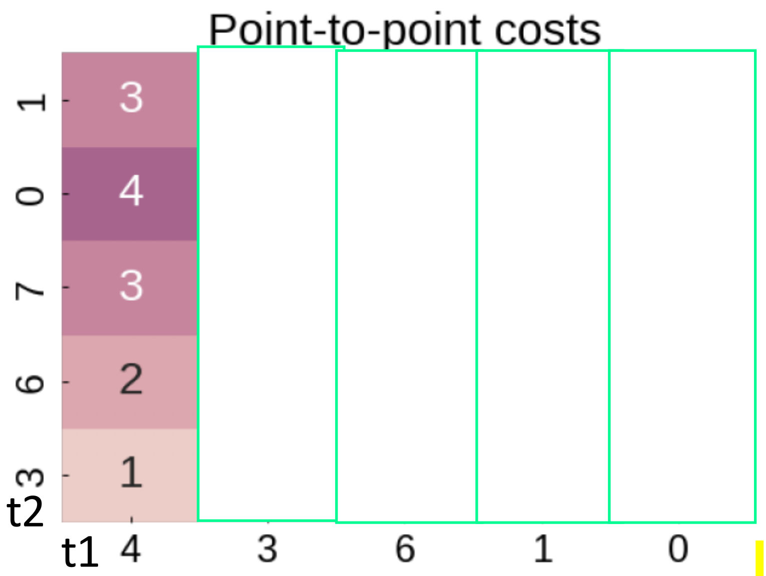
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



Result: 4

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

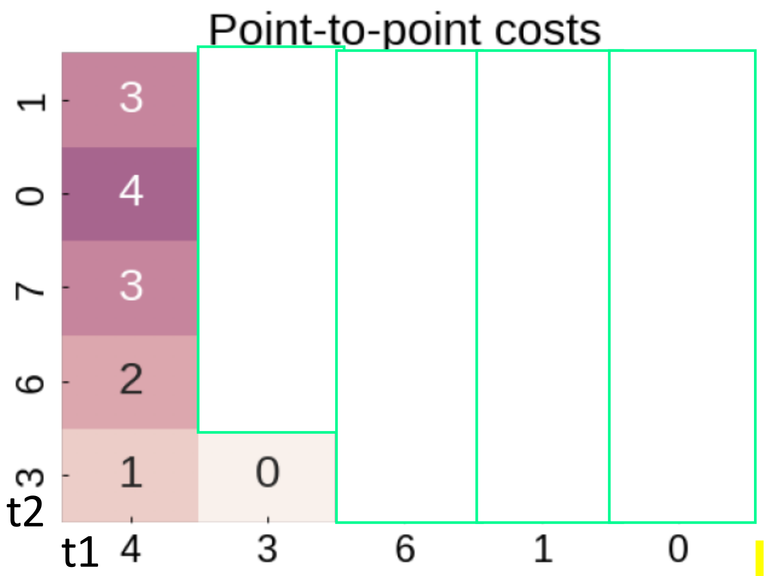
DTW – Example



Result: 4

DTW – Example

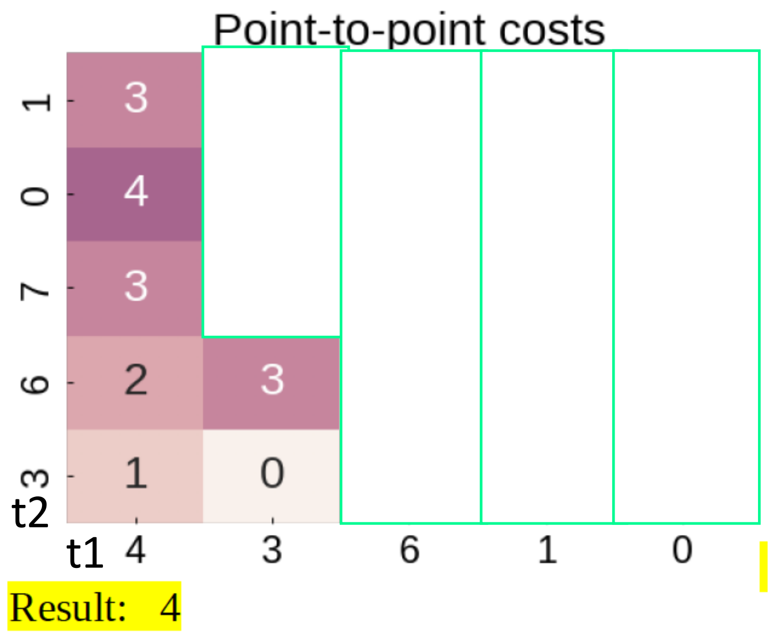
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



Result: 4

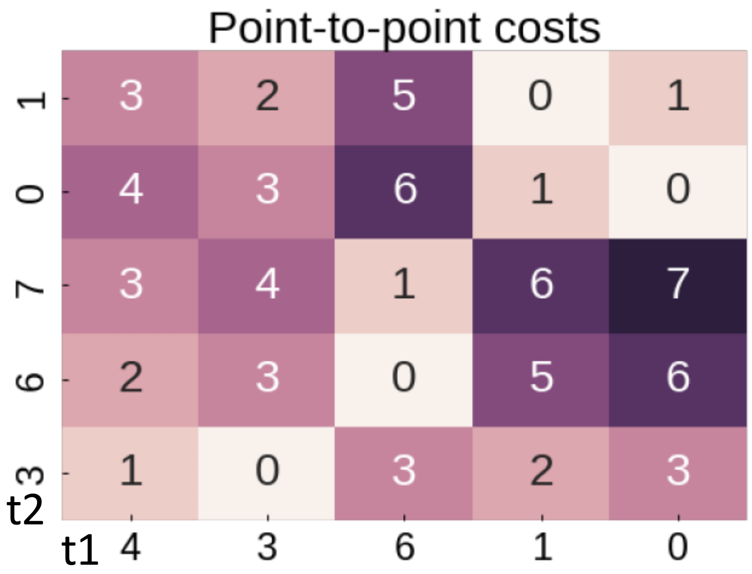
DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



DTW – Example

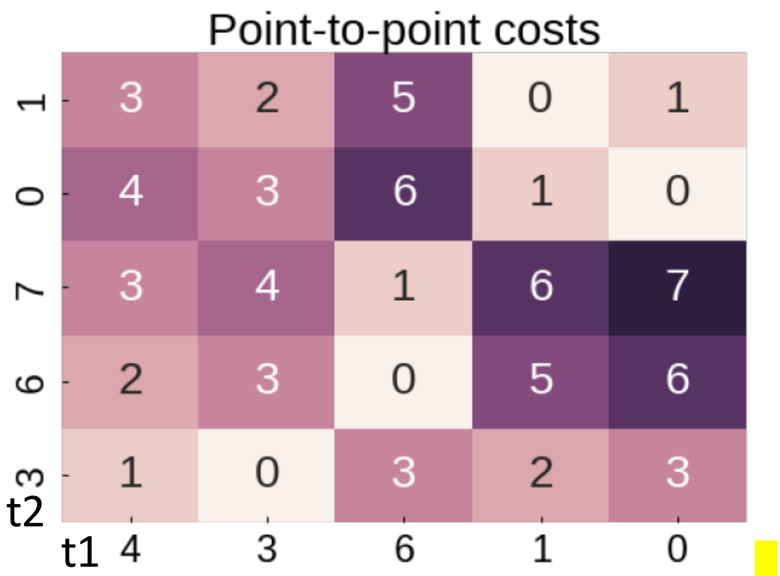
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



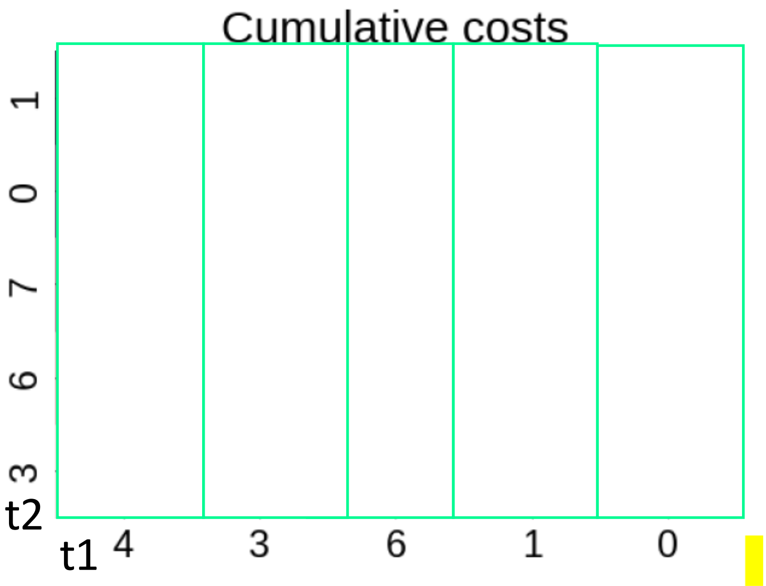
Result: 4

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Example



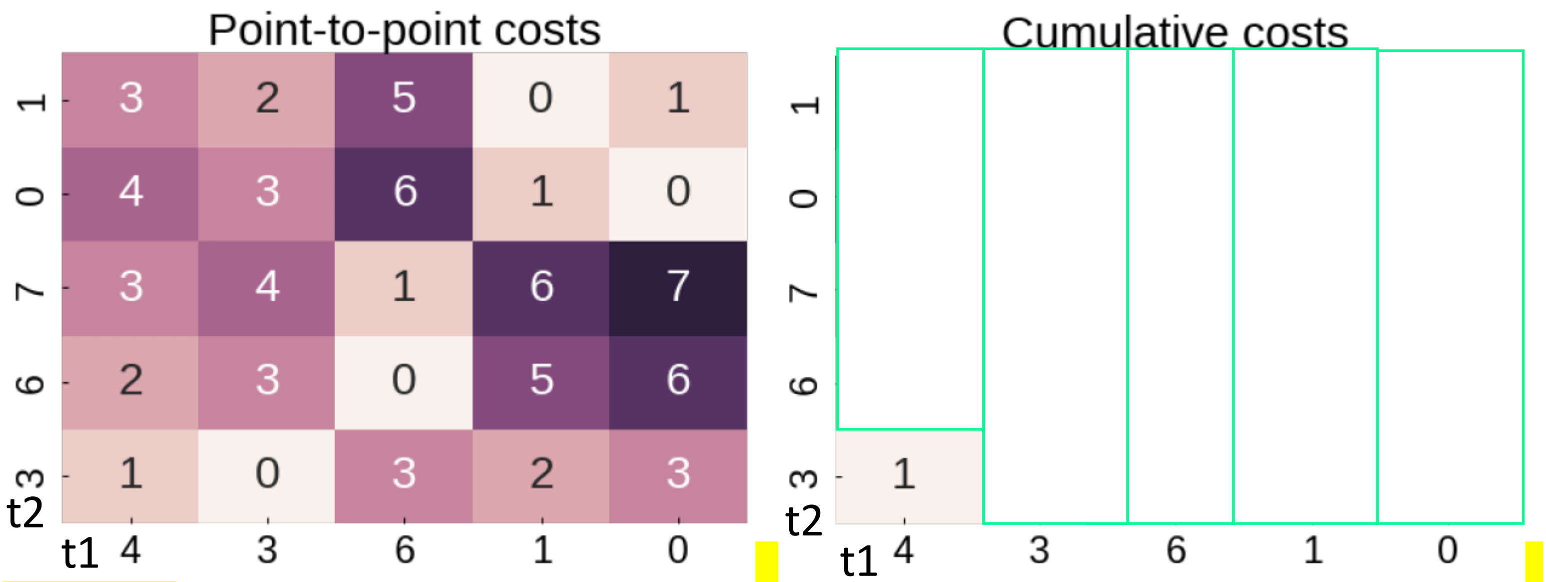
Result: 4



$$\gamma(i,j) = d(q_i,c_j) + \min\{\, \gamma(i-1,j-1),\ \gamma(i-1,j),\ \gamma(i,j-1) \,\}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

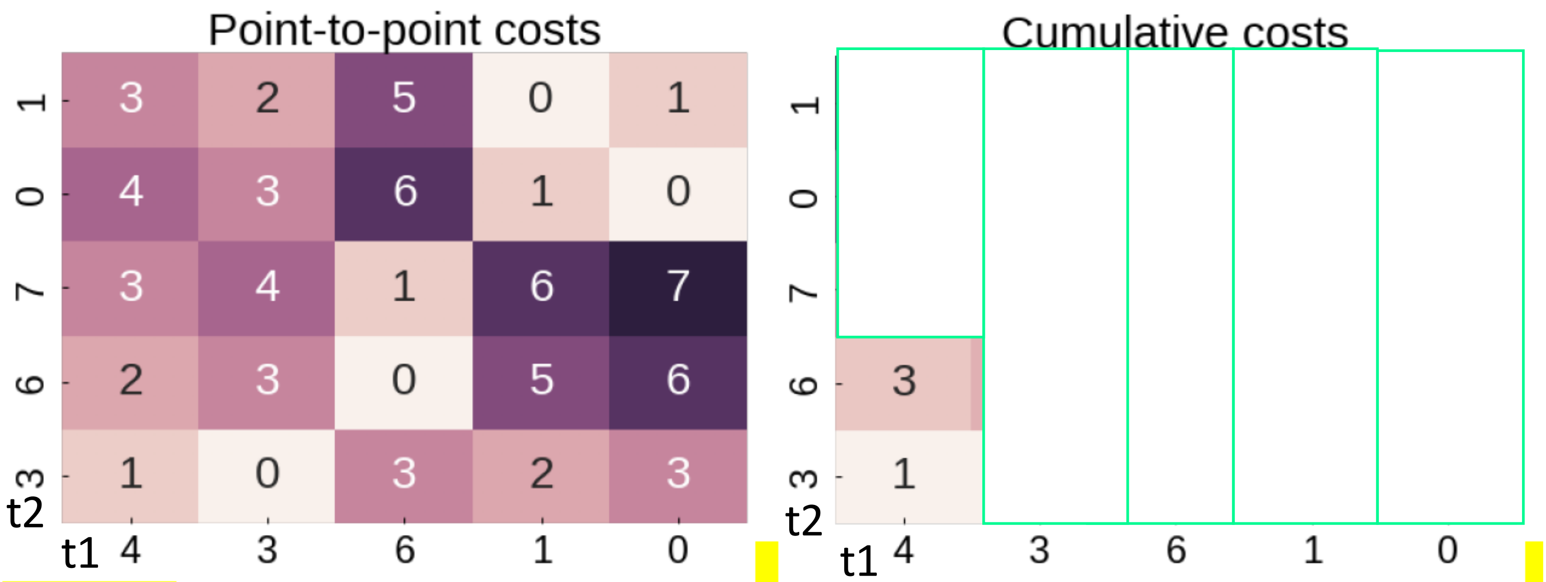


Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

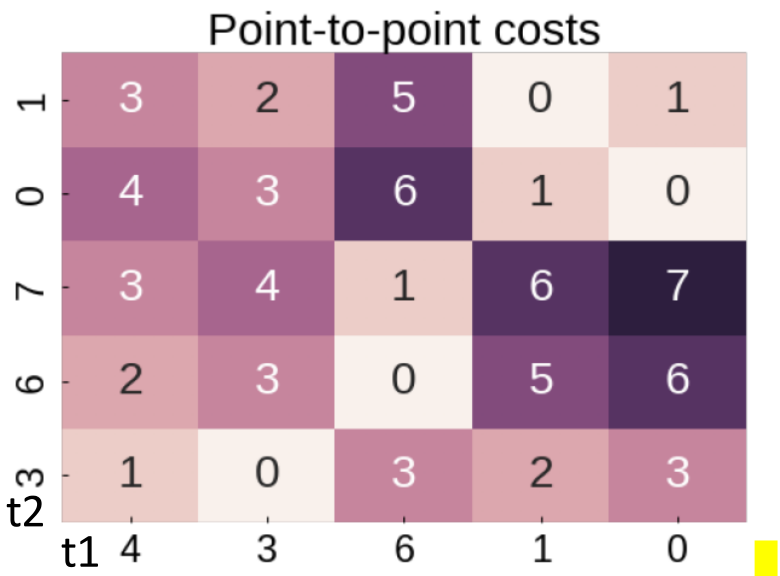


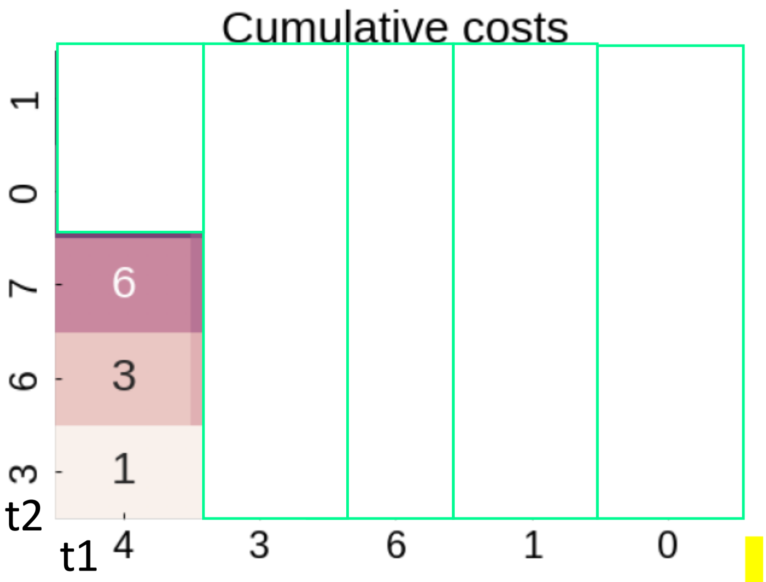
Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{\, \gamma(i-1,j-1),\ \gamma(i-1,j),\ \gamma(i,j-1) \,\}$$

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Example



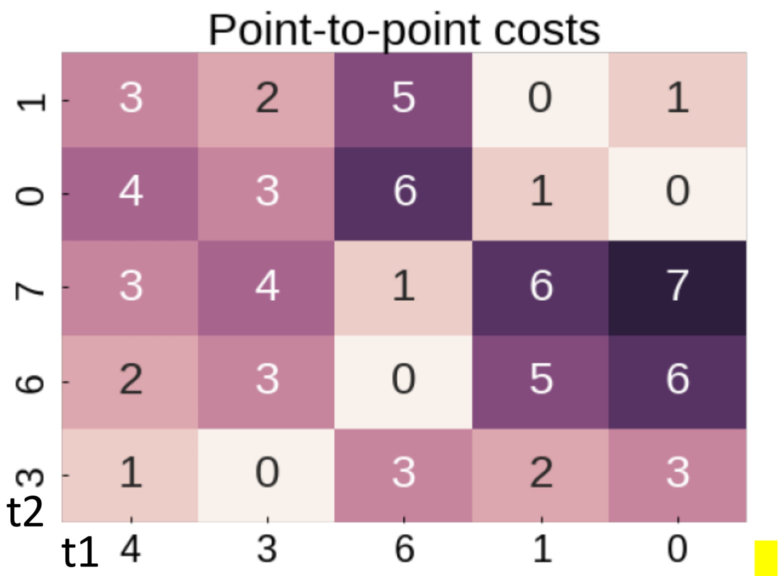


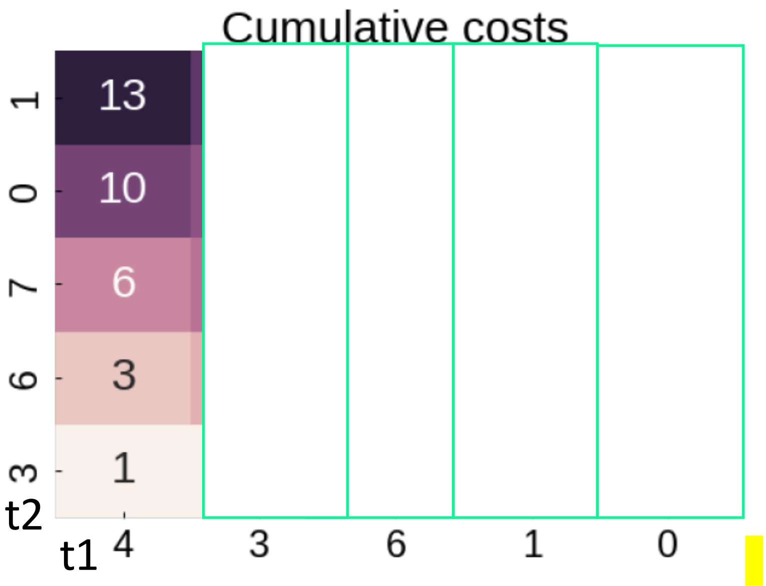
Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{\ \gamma(i\text{-}1,j\text{-}1),\ \gamma(i\text{-}1,j\ ),\ \gamma(i,j\text{-}1)\ \}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



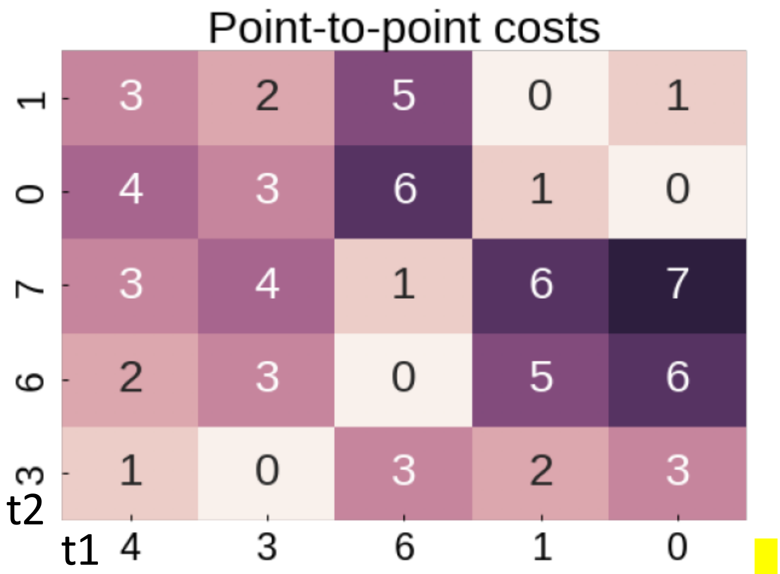


Result: 4

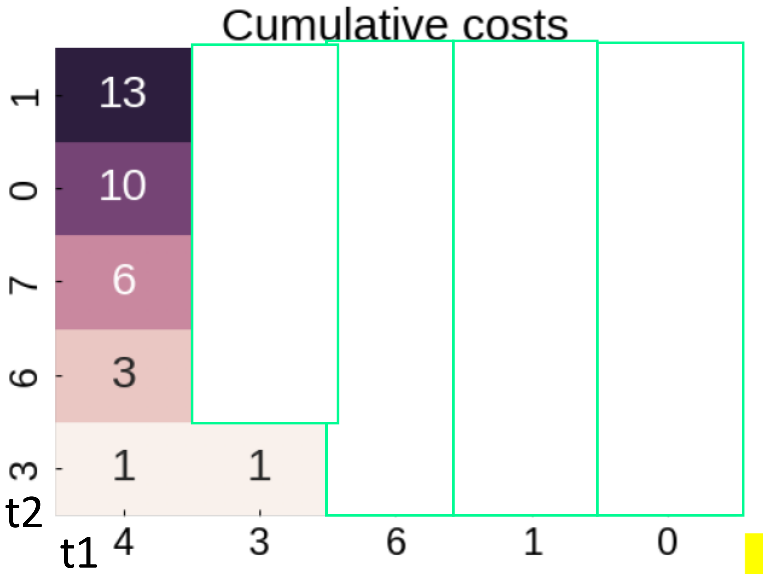
$$\gamma(i,j) = d(q_i,c_j) + \min\{\ \gamma(i\text{-}1,j\text{-}1),\ \gamma(i\text{-}1,j\ ),\ \gamma(i,j\text{-}1)\ \}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



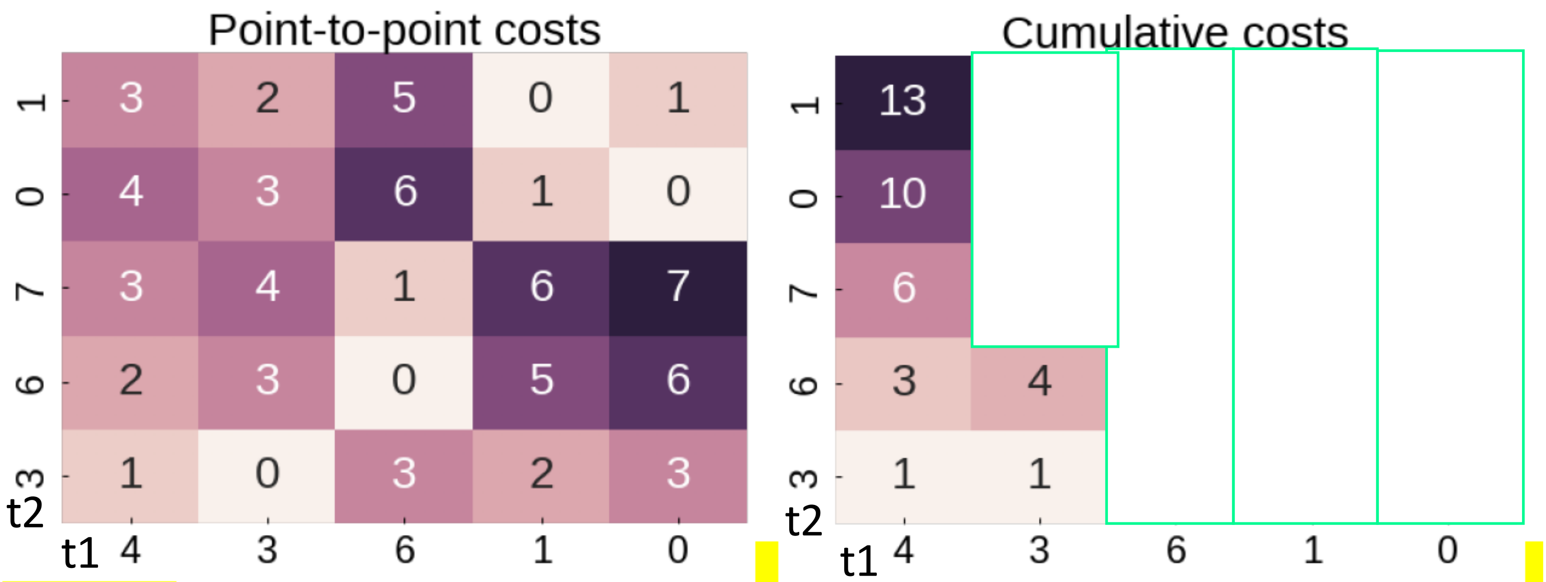
Result: 4



$$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i\text{-}1,j\text{-}1),\ \gamma(i\text{-}1,j\,),\ \gamma(i,j\text{-}1)\,\}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

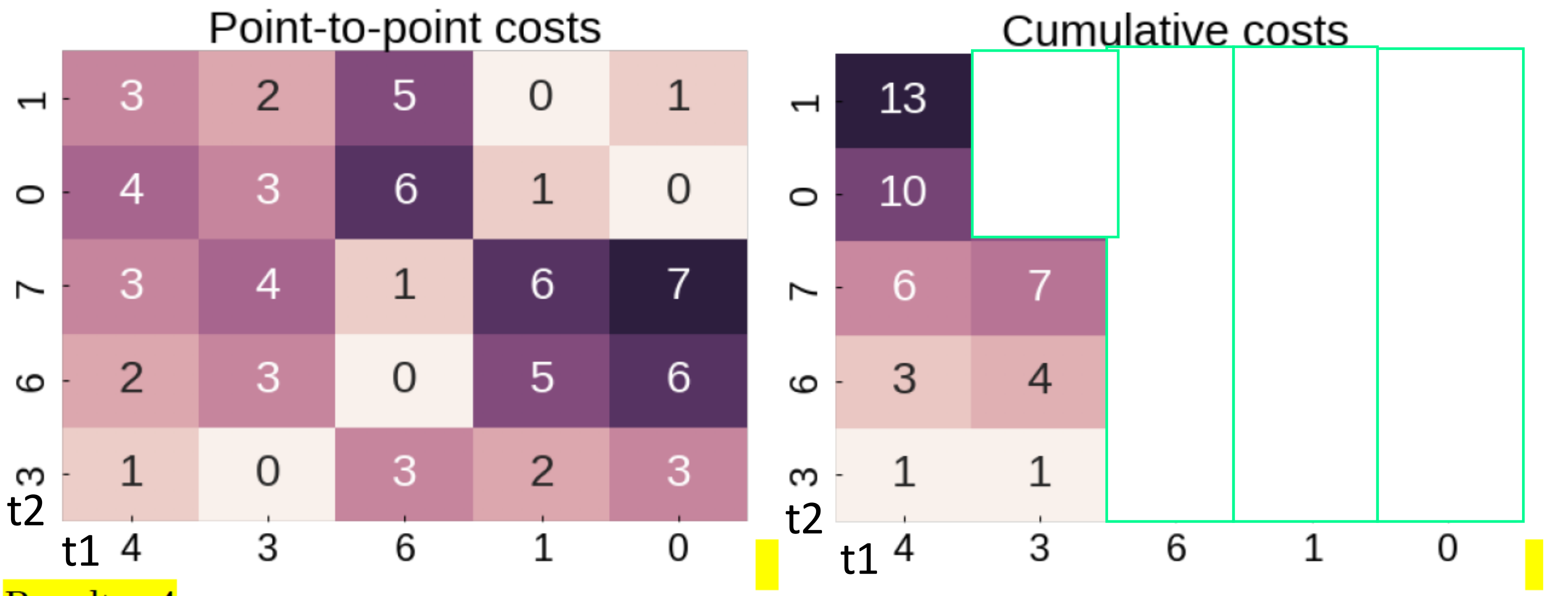


Result: 4

$$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i\text{-}1,j\text{-}1),\, \gamma(i\text{-}1,j\,),\, \gamma(i,j\text{-}1)\,\}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

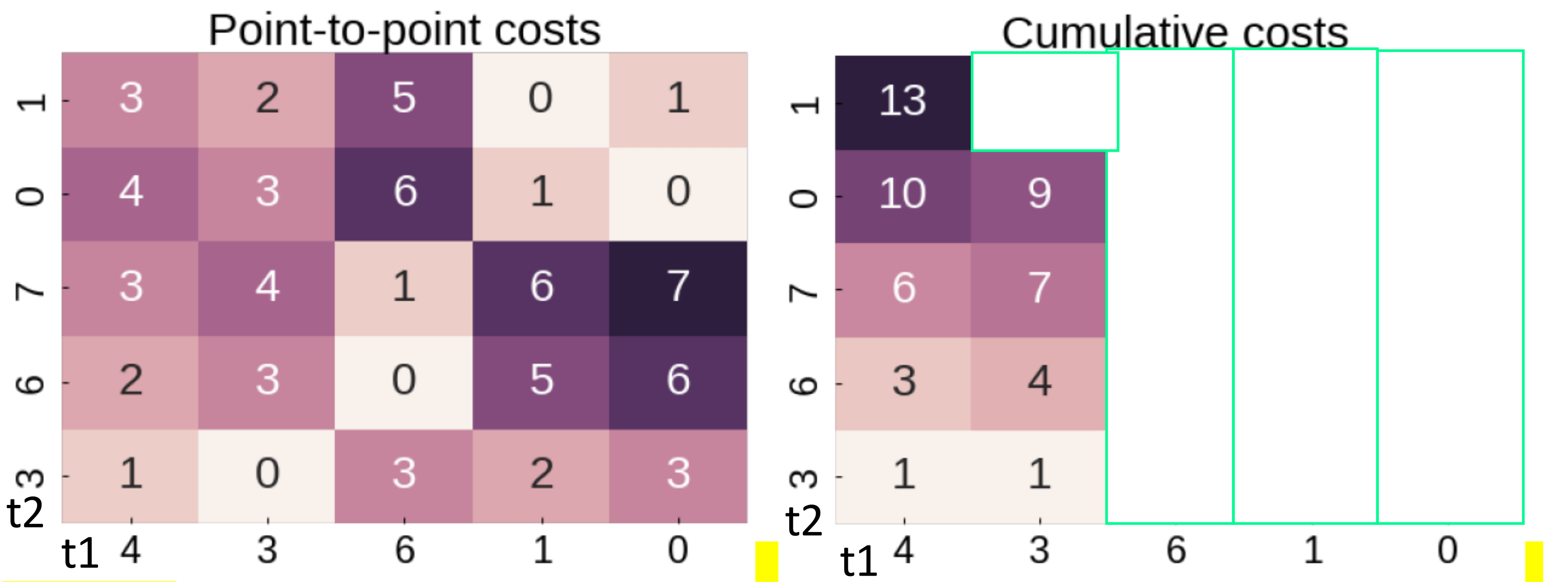


Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{\, \gamma(i-1,j-1),\ \gamma(i-1,j),\ \gamma(i,j-1) \,\}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

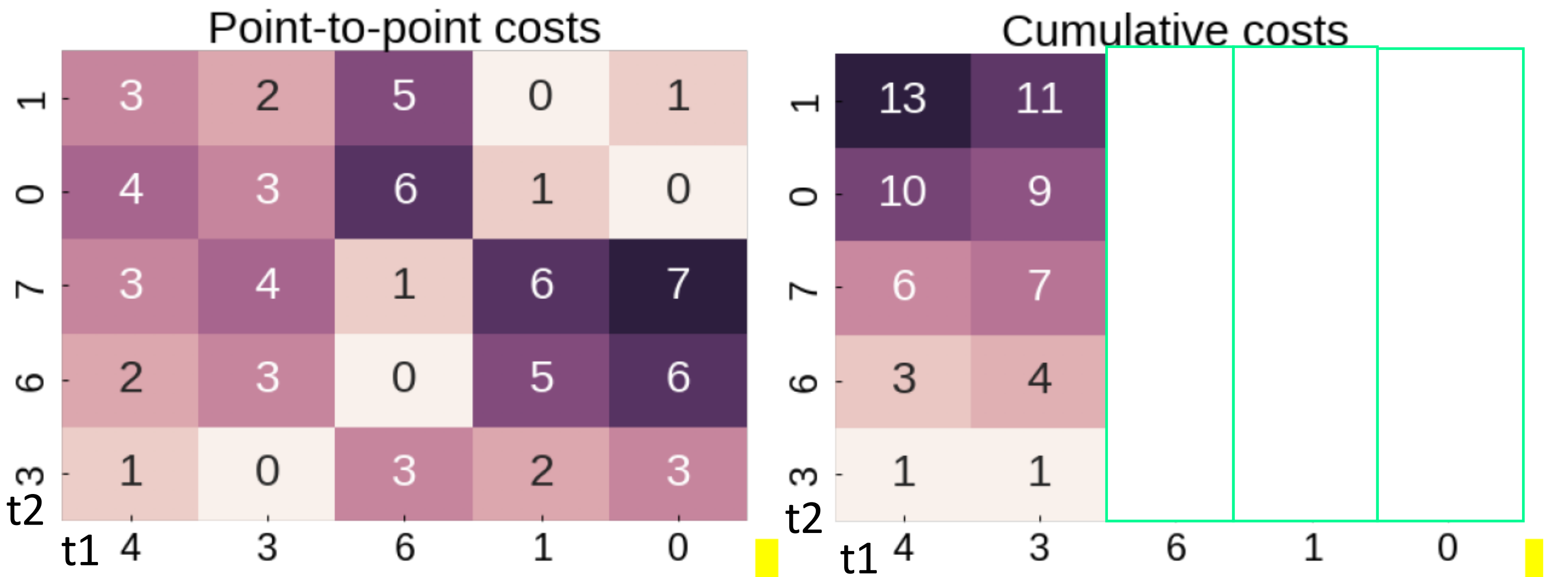


Result: 4

$$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i\text{-}1,j\text{-}1),\ \gamma(i\text{-}1,j\,),\ \gamma(i,j\text{-}1) \,\}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

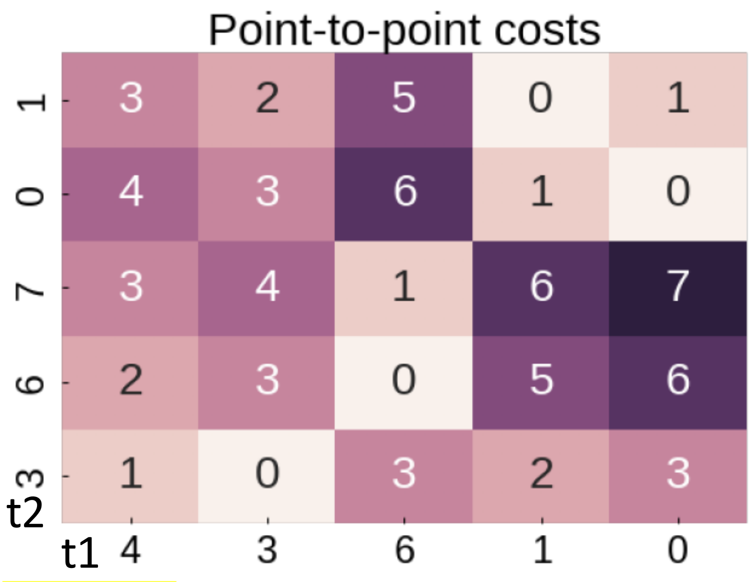


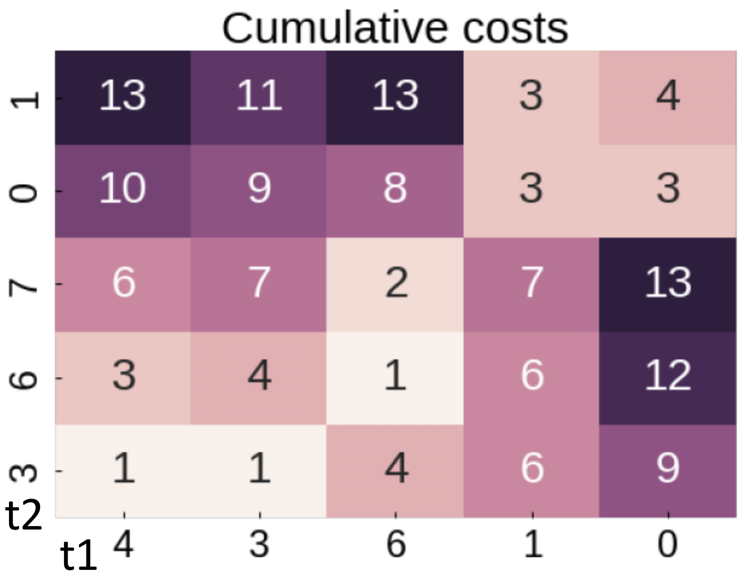
Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

DTW – Example

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



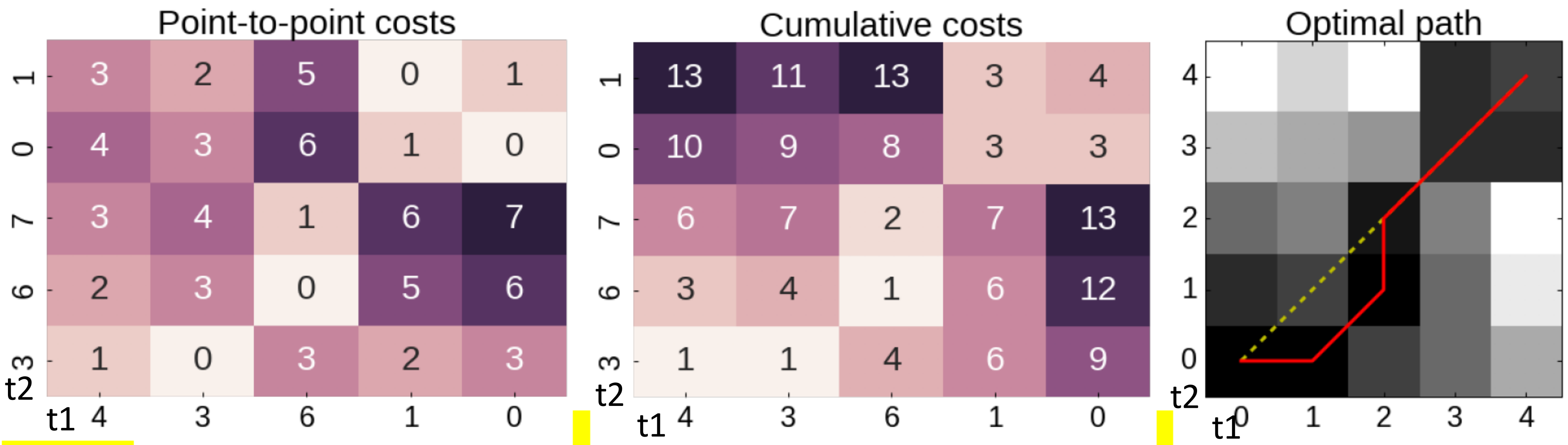


Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

DTW – Example

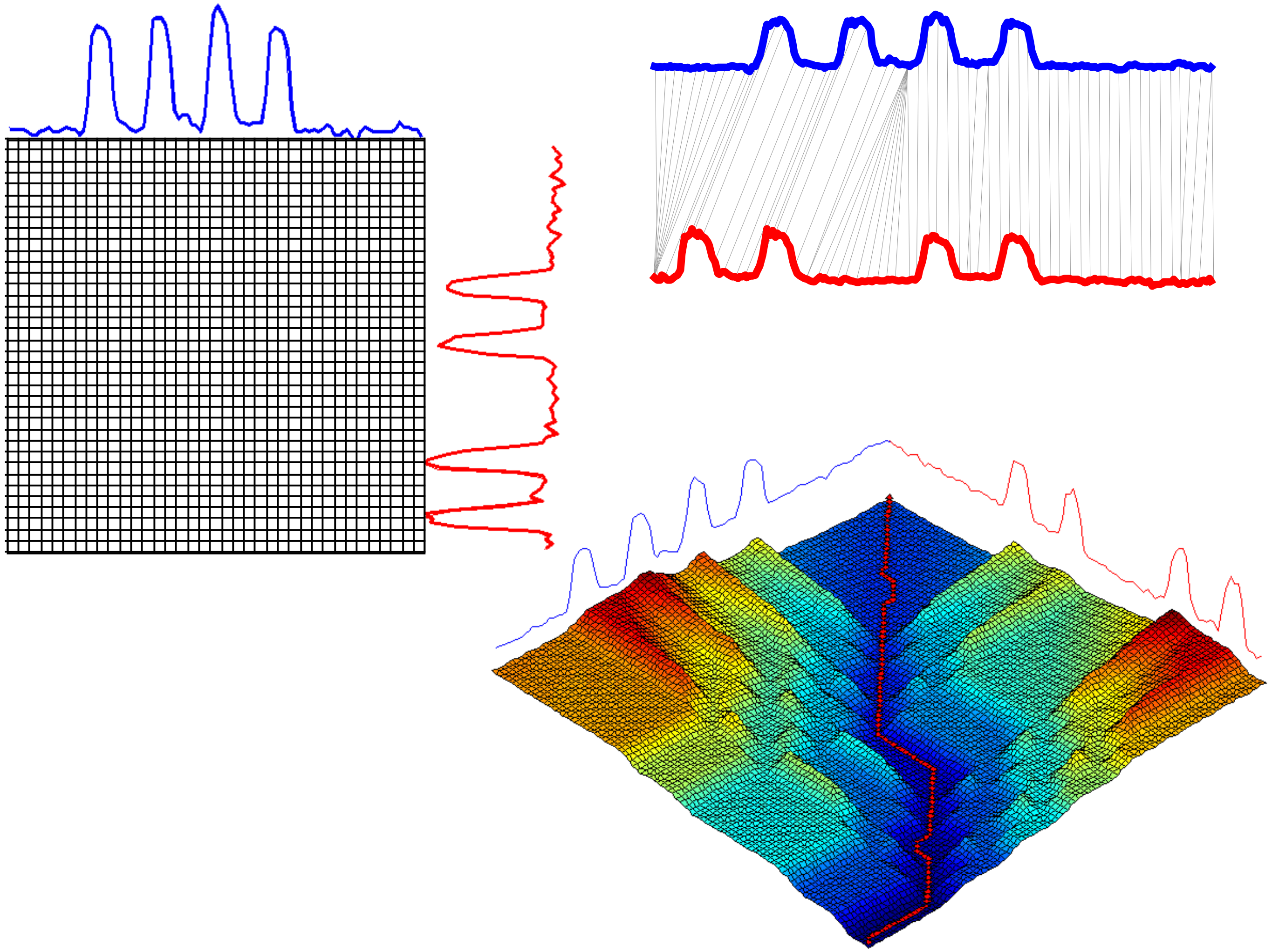
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



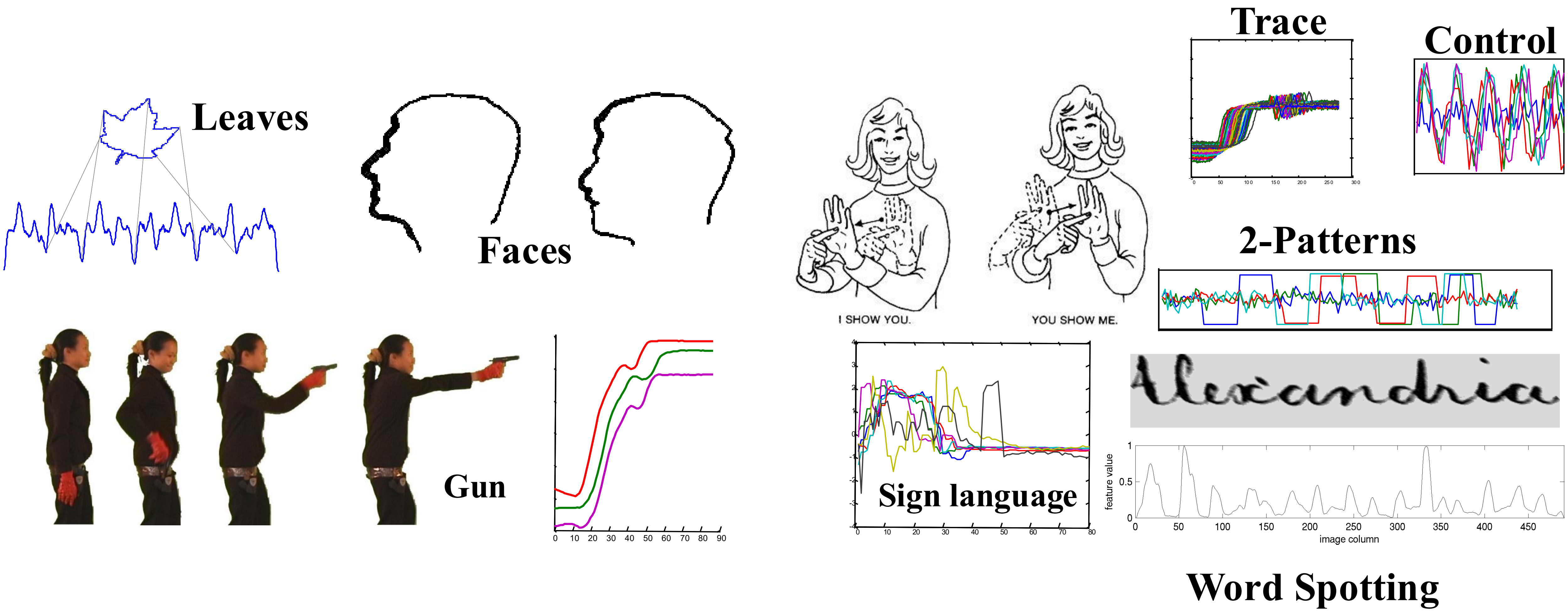
Result: 4

DTW – A Real Example

- This example shows 2 one-week periods from the power demand time series.
- Note that although they both describe 4-day work weeks, the blue sequence had Monday as a holiday, and the red sequence had Wednesday as a holiday.



Comparison of Euclidean Distance and DTW



Comparison of Euclidean Distance and DTW

- Classification using 1-NN
- Class(x) = class of most similar training object
- Leaving-one-out evaluation
- For each object: use it as test set, return overall average

Accuracy

Dataset	Euclidean	DTW
Word Spotting	0.95	0.99
Sign language	0.71	0.74
GUN	0.95	0.99
Nuclear Trace	0.89	1.00
Leaves#	0.67	0.96
(4) Faces	0.94	0.97
Control Chart*	0.93	1.00
2-Patterns	0.99	1.00

Comparison of Euclidean Distance and DTW

- Classification using 1-NN
- Class(x) = class of most similar training object
- Leaving-one-out evaluation
- For each object: use it as test set, return overall average
- DTW is two to three orders of magnitude slower than Euclidean distance.

Milliseconds

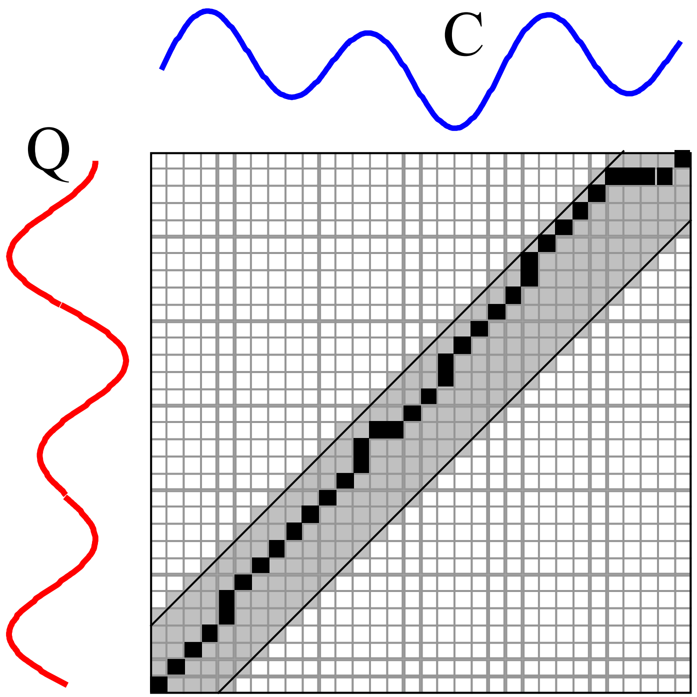
Dataset	Euclidean	DTW
Word Spotting	40	8,600
Sign language	10	1,110
GUN	60	11,820
Nuclear Trace	210	144,470
Leaves	150	51,830
(4) Faces	50	45,080
Control Chart	110	21,900
2-Patterns	16,890	545,123

Problems with Dynamic Time Warping

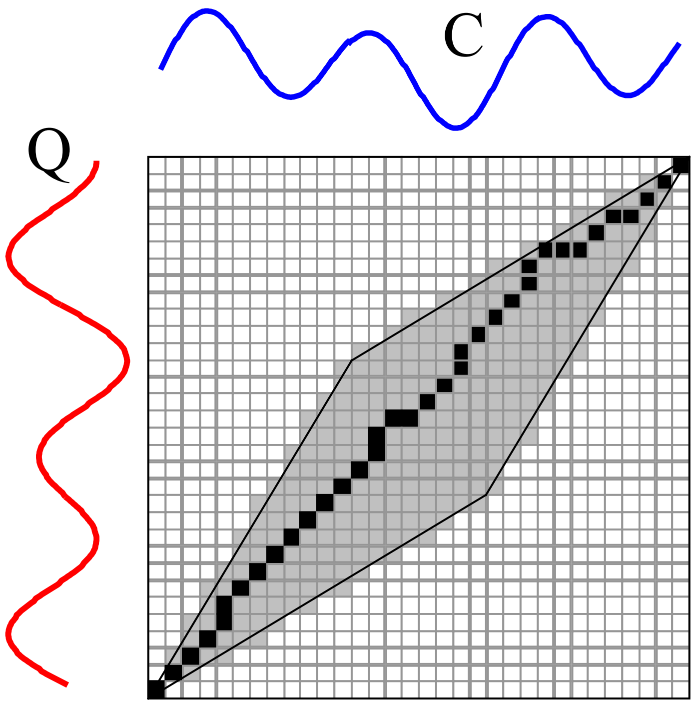
- Dynamic Time Warping gives much better results than Euclidean distance on many problems.
- Dynamic Time Warping is very very slow to calculate!
- Is there anything we can do to speed up similarity search under DTW?

Global Constraints

- Slightly speed up the calculations
- Prevent pathological warpings



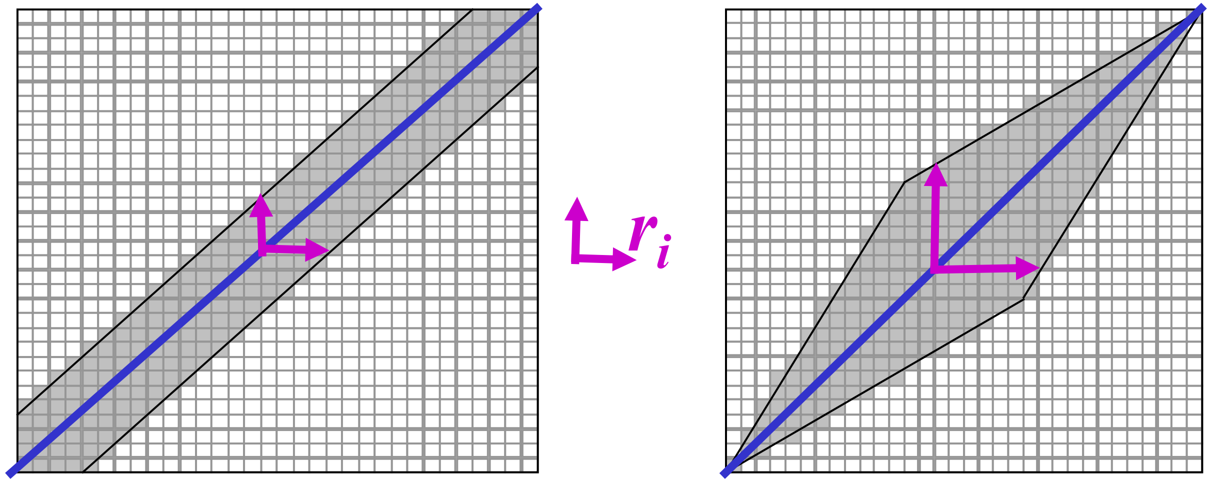
Sakoe-Chiba Band



Itakura Parallelogram

Global Constraints

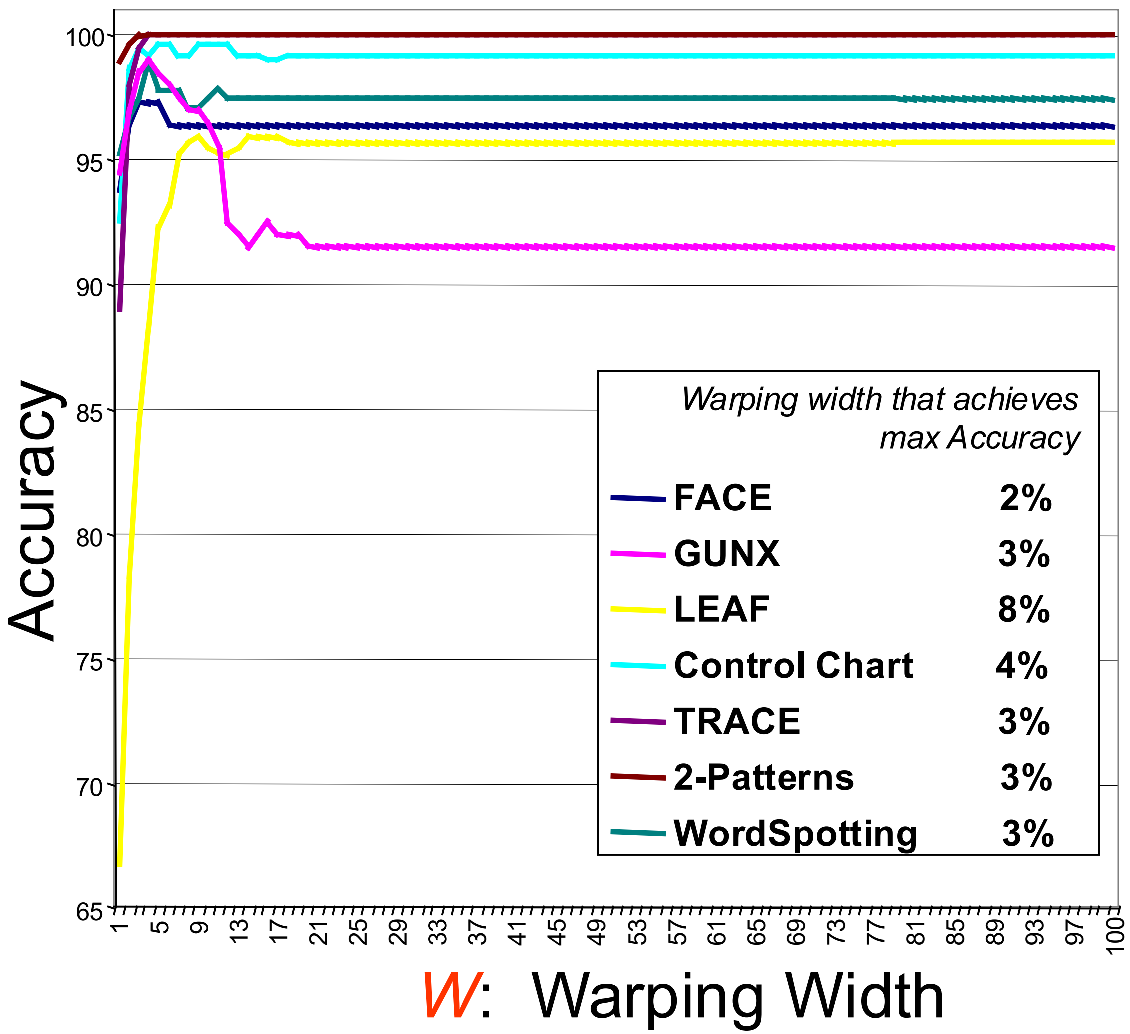
- A global constraint constrains the indices of the warping path $w_k = (i,j)_k$ such that $j\text{-}r \leq i \leq j\text{+}r$, where r is a term defining allowed range of warping for a given point in a sequence.
- r can be considered as a *window* that reduces the number of calculus.

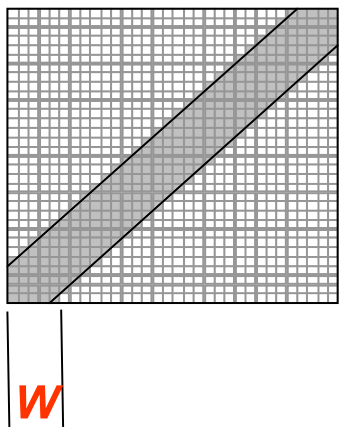


Sakoe-Chiba Band

Itakura Parallelogram

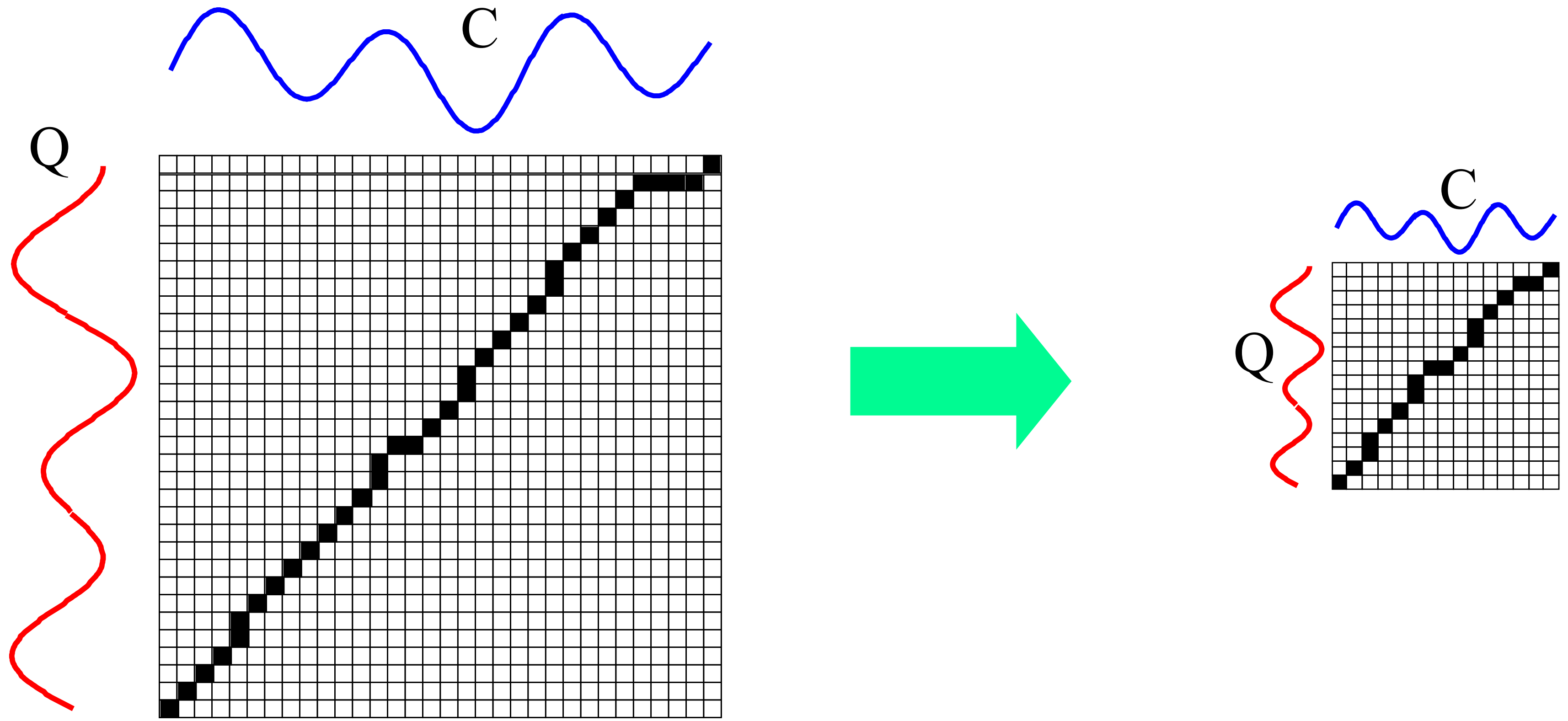
Accuracy vs. Width of Warping Window





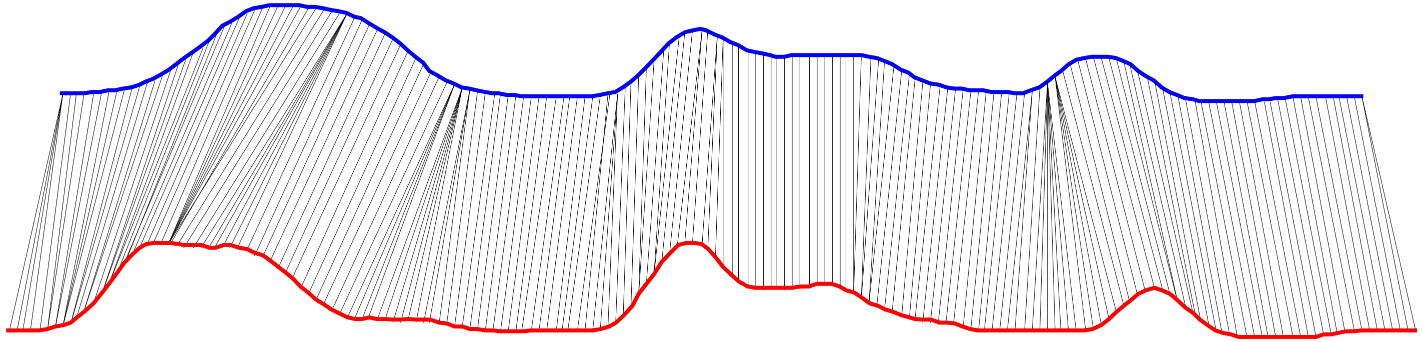
Fast Approximations to DTW

- Approximate the time series with some compressed or downsampled representation and do DTW on the new representation.

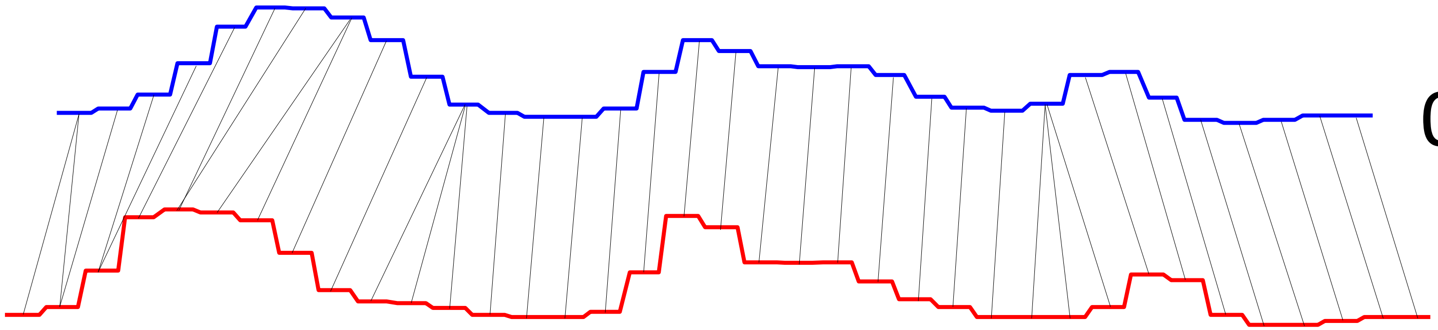


Fast Approximations to DTW

- There is strong visual evidence to suggests it works well
- In the literature there is good experimental evidence for the utility of the approach on clustering, classification, etc.



1.03 sec



0.07 sec

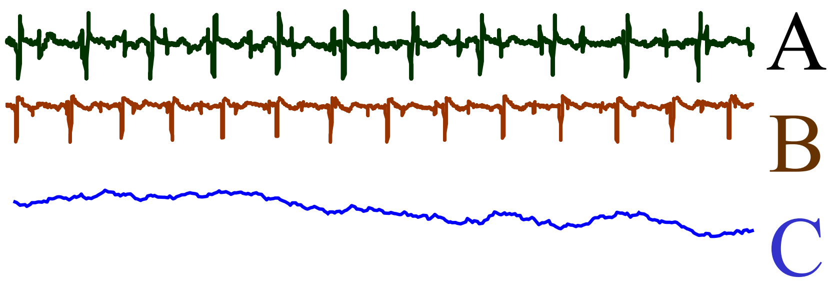
Distances and Normalizations

- If measuring a distance to account for a shape-based similarity it is important to consider the level then the level, i.e., the mean, should not be removed.
- This kind of reasoning applies also to other features of the TS.

Global Structural Features

Structure-based Similarity

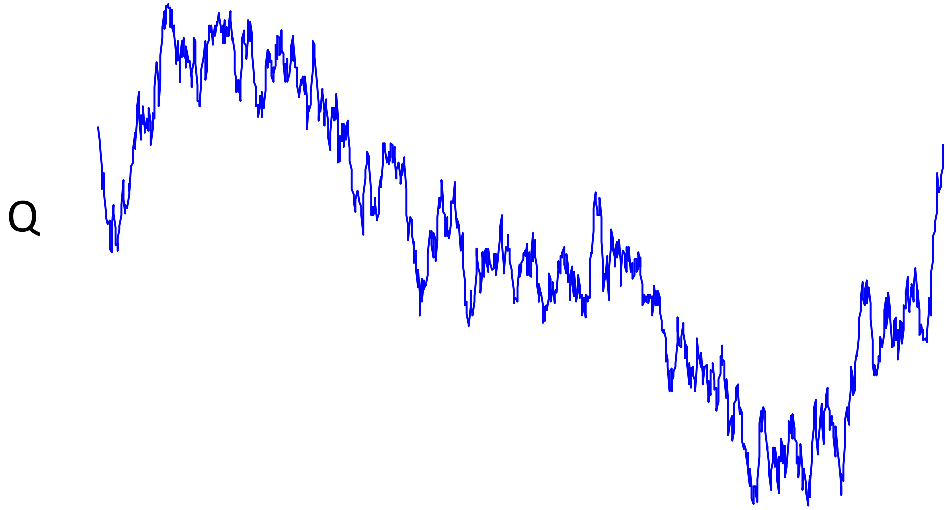
- For long time series, shape-based similarity typically give poor results.
- Structure-based similarity measure similarly of TS based on high level structure.
- The basic idea is to:
  1. extract ***global*** features from the time series,
  2. create a feature vector, and
  3. use it to measure similarity with Euclidean distance
- Example of features:
  - mean, variance, skewness, kurtosis,
  - 1st derivative mean, 1st derivative variance, …
  - parameters of regression, forecasting, Markov model



Feature\Time Series	A	B	C
Max Value	11	12	19
Mean	5.3	6.4	4.8
Min Value	3	2	5
Autocorrelation	0.2	0.3	0.5
…	…	…	…

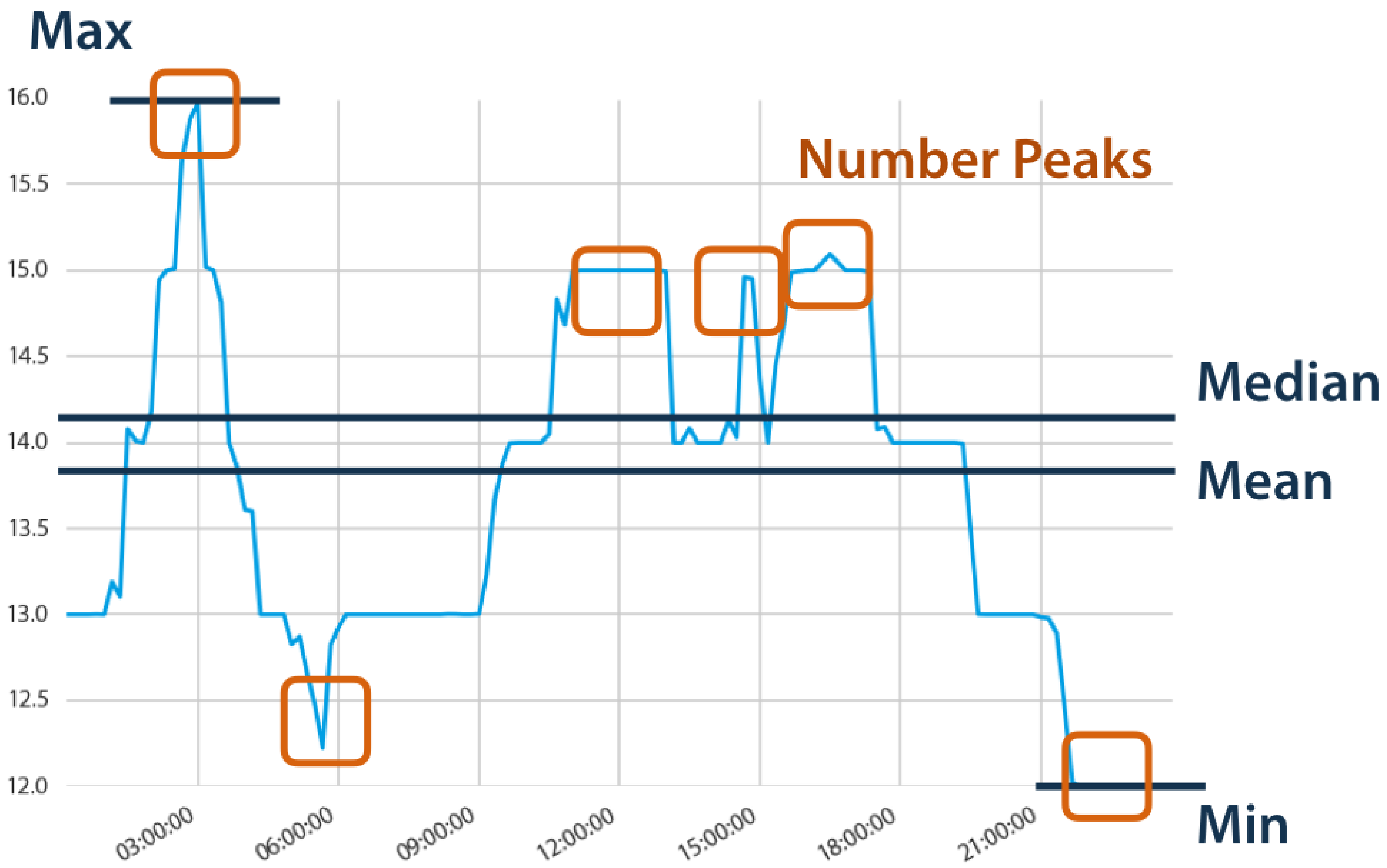
Simple Standard Features

- Mean
- Standard Deviation
- Variance
- Median
- 10th Percentile
- 25th Percentile
- 75th Percentile
- 90th Percentile
- IQR
- Covariance
- Skewness
- Kurtosis
- Min
- Max



	mean	std	var	med	…	max
Q	1.8	2.9	8.4	1.3	…	

TSFresh Features



TSFresh Features

- abs_energy - Returns the absolute energy of the time series which is the sum over the squared values
- absolute_maximum - Calculates the highest absolute value of the time series x.
- absolute_sum_of_changes - Returns the sum over the absolute value of consecutive changes in the series x
- agg_autocorrelation - Descriptive statistics on the autocorrelation of the time series.
- agg_linear_trend - Calculates a linear least-squares regression for values of the time series that were aggregated over chunks versus the sequence from 0 up to the number of chunks minus one.
- approximate_entropy - Implements a vectorized Approximate entropy algorithm.
- ar_coefficient - This feature calculator fits the unconditional maximum likelihood of an autoregressive AR(k) process.
- augmented_dickey_fuller - Does the time series have a unit root?
- autocorrelation - Calculates the autocorrelation of the specified lag
- benford_correlation - Useful for anomaly detection applications. Returns the correlation from first digit distribution when
- binned_entropy - First bins the values of x into max_bins equidistant bins.
- c3 - Uses c3 statistics to measure non linearity in the time series
- change_quantiles - First fixes a corridor given by the quantiles ql and qh of the distribution of x.
- cid_ce - This function calculator is an estimate for a time series complexity.
- count_above - Returns the percentage of values in x that are higher than t
- count_above_mean - Returns the number of values in x that are higher than the mean of x
- count_below - Returns the percentage of values in x that are lower than t
- count_below_mean - Returns the number of values in x that are lower than the mean of x
- cwt_coefficients - Calculates a Continuous wavelet transform for the Ricker wavelet, also known as the "Mexican hat wavelet" which is defined by
- energy_ratio_by_chunks - Calculates the sum of squares of chunk i out of N chunks expressed as a ratio with the sum of squares over the whole series.
- fft_aggregated - Returns the spectral centroid (mean), variance, skew, and kurtosis of the absolute fourier transform spectrum.

TSFresh Features

- fft_coefficient - Calculates the fourier coefficients of the one-dimensional discrete Fourier Transform for real input by fast fourier transformation algorithm
- first_location_of_maximum - Returns the first location of the maximum value of x.
- first_location_of_minimum - Returns the first location of the minimal value of x.
- fourier_entropy - Calculate the binned entropy of the power spectral density of the time series (using the welch method).
- friedrich_coefficients - Coefficients of polynomial, which has been fitted to the deterministic dynamics of Langevin model
- has_duplicate - Checks if any value in x occurs more than once
- has_duplicate_max - Checks if the maximum value of x is observed more than once
- has_duplicate_min - Checks if the minimal value of x is observed more than once
- index_mass_quantile - Calculates the relative index i of time series x where q% of the mass of x lies left of i.
- kurtosis - Returns the kurtosis of x.
- large_standard_deviation - Does time series have large standard deviation?
- last_location_of_maximum - Returns the relative last location of the maximum value of x.
- last_location_of_minimum - Returns the last location of the minimal value of x.
- lempel_ziv_complexity - Calculate a complexity estimate based on the Lempel-Ziv compression algorithm.
- length - Returns the length of x
- linear_trend - Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one.
- linear_trend_timewise - Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one.
- longest_strike_above_mean - Returns the length of the longest consecutive subsequence in x that is bigger than the mean of x
- longest_strike_below_mean - Returns the length of the longest consecutive subsequence in x that is smaller than the mean of x

TSFresh Features

- fast fourier matrix_profile - Calculates the 1-D Matrix Profile[1] and returns Tukey's Five Number Set plus the mean of that Matrix Profile.
- max_langevin_fixed_point - Largest fixed point of dynamics :math:argmax_x {h=0}` estimated from polynomial, which has been fitted to the deterministic dynamics of Langevin model
- maximum - Calculates the highest value of the time series x.
- mean - Returns the mean of x
- mean_abs_change - Average over first differences.
- mean_change - Average over time series differences.
- mean_n_absolute_max - Calculates the arithmetic mean of the n absolute maximum values of the time series.
- mean_second_derivative_central - Returns the mean value of a central approximation of the second derivative
- median - Returns the median of x
- minimum - Calculates the lowest value of the time series x.
- number_crossing_m - Calculates the number of crossings of x on m.
- number_cwt_peaks - Number of different peaks in x.
- number_peaks - Calculates the number of peaks of at least support n in the time series x.
- partial_autocorrelation - Calculates the value of the partial autocorrelation function at the given lag.
- percentage_of_reoccurring_datapoints_to_all_datapoints - Returns the percentage of non-unique data points.
- percentage_of_reoccurring_values_to_all_values - Returns the percentage of values that are present in the time series more than once.
- permutation_entropy - Calculate the permutation entropy.
- quantile - Calculates the q quantile of x.
- query_similarity_count - This feature calculator accepts an input query subsequence parameter, compares the query (under z-normalized Euclidean distance) to all subsequences within the time series, and returns a count of the number of times the query was found in the time series (within some predefined maximum distance threshold).

TSFresh Features

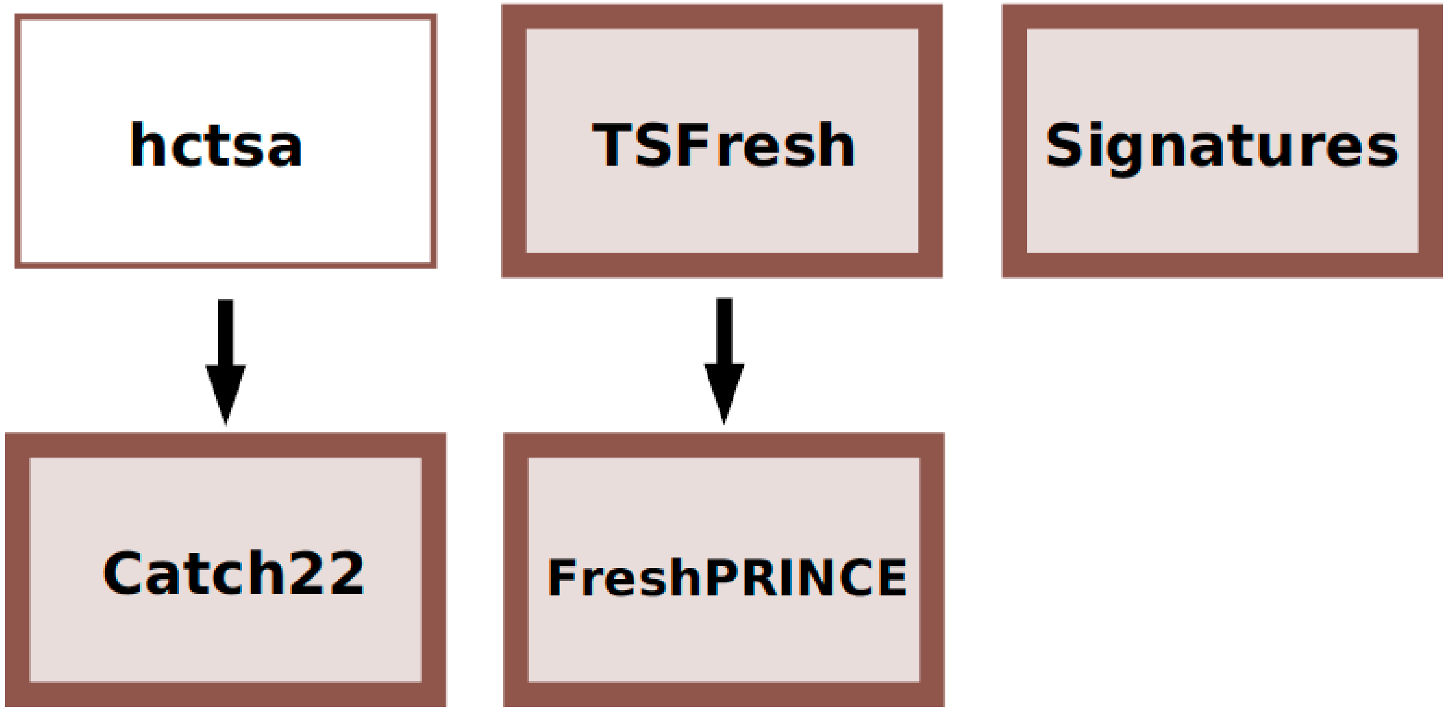
- range_count - Count observed values within the interval [min, max).
- ratio_beyond_r_sigma - Ratio of values that are more than r * std (so r times sigma) away from the mean of x.
- ratio_value_number_to_time_series_length - Returns a factor which is 1 if all values in the time series occur only once, and below one if this is not the case.
- root_mean_square - Returns the root mean square (rms) of the time series.
- sample_entropy - Calculate and return sample entropy of x.
- set_property - This method returns a decorator that sets the property key of the function to value
- skewness - Returns the sample skewness of x (calculated with the adjusted Fisher-Pearson standardized moment coefficient G1).
- spkt_welch_density - This feature calculator estimates the cross power spectral density of the time series x at different frequencies.
- standard_deviation - Returns the standard deviation of x
- sum_of_reoccurring_data_points - Returns the sum of all data points, that are present in the time series more than once.
- sum_of_reoccurring_values - Returns the sum of all values, that are present in the time series more than once.
- sum_values - Calculates the sum over the time series values
- symmetry_looking - Boolean variable denoting if the distribution of x looks symmetric.
- time_reversal_asymmetry_statistic - Returns the time reversal asymmetry statistic.
- value_count - Count occurrences of value in time series x.
- variance - Returns the variance of x
- variance_larger_than_standard_deviation - Is variance higher than the standard deviation?
- variation_coefficient - Returns the variation coefficient (standard error / mean, give relative value of variation around mean) of x

catch22: CAnonical Time-series CHaracteristics

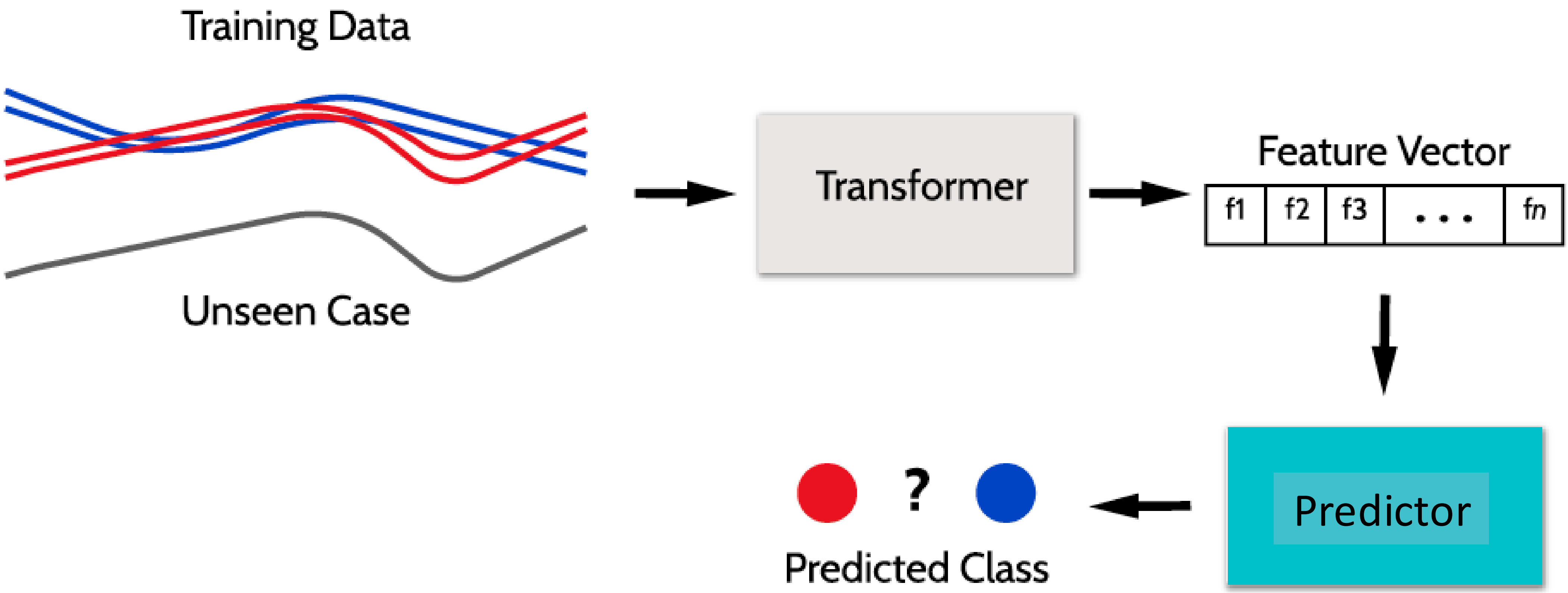
- The catch22 feature set spans a diverse range of time-series characteristics representative of the diversity of interdisciplinary methods for TSA.
- Features in catch22 capture TS properties of the distribution of values in the TS, linear and nonlinear temporal autocorrelation properties, scaling of fluctuations, and others.
- Selected by applying the procedure describe in [Lubba 2019] to a set of 93 datasets containing over 147k TS and using a filtered version of the HCTSA feature library (4791 features).
- The reduction from 4791 to 22 features is associated with a 1000-fold reduction in computation time and near linear scaling with TS length, despite an average reduction in classification accuracy of just 7%.

Distribution	
`DN_HistogramMode_5`	Mode of z-scored distribution (5-bin histogram)
`DN_HistogramMode_10`	Mode of z-scored distribution (10-bin histogram)
Simple temporal statistics	
`SB_BinaryStats_mean_longstretch1`	Longest period of consecutive values above the mean
`DN_OutlierInclude_p_001_mdrmd`	Time intervals between successive extreme events above the mean
`DN_OutlierInclude_n_001_mdrmd`	Time intervals between successive extreme events below the mean
Linear autocorrelation	
`CO_f1ecac`	First $1/e$ crossing of autocorrelation function
`CO_FirstMin_ac`	First minimum of autocorrelation function
`SP_Summaries_welch_rect_area_5_1`	Total power in lowest fifth of frequencies in the Fourier power spectrum
`SP_Summaries_welch_rect_centroid`	Centroid of the Fourier power spectrum
`FC_LocalSimple_mean3_stderr`	Mean error from a rolling 3-sample mean forecasting
Nonlinear autocorrelation	
`CO_trev_1_num`	Time-reversibility statistic, $\langle (x_{t+1} - x_t)^3 \rangle_t$
`CO_HistogramAMI_even_2_5`	Automutual information, $m = 2, \tau = 5$
`IN_AutoMutualInfoStats_40_gaussian_fmmi`	First minimum of the automutual information function
Successive differences	
`MD_hrv_classic_pnn40`	Proportion of successive differences exceeding 0.04σ [20]
`SB_BinaryStats_diff_longstretch0`	Longest period of successive incremental decreases
`SB_MotifThree_quantile_hh`	Shannon entropy of two successive letters in equiprobable 3-letter symbolization
`FC_LocalSimple_mean1_tauresrat`	Change in correlation length after iterative differencing
`CO_Embed2_Dist_tau_d_expfit_meandiff`	Exponential fit to successive distances in 2-d embedding space
Fluctuation Analysis	
`SC_FluctAnal_2_dfa_50_1_2_logi_prop_r1`	Proportion of slower timescale fluctuations that scale with DFA (50% sampling)
`SC_FluctAnal_2_rsrangefit_50_1_logi_prop_r1`	Proportion of slower timescale fluctuations that scale with linearly rescaled range fits
Others	
`SB_TransitionMatrix_3ac_sumdiagcov`	Trace of covariance of transition matrix between symbols in 3-letter alphabet
`PD_PeriodicityWang_th0_01`	Periodicity measure of [31]

Overview of Global Features and Relationships



Global Feature-based Predictor



Features, Approximations, Distances and Normalizations

- Normalizations can be applied before global features extraction depending on the objective of your TSA task.
- Time-Dependent approximations can be applied before global features extraction depending on the objective of your TSA task.
- It does not make any sense to use Time-Independent approximation after that global features have been extracted.
- It does not make any sense to use a distance function accounting for time like DTW after that global features have been extracted.

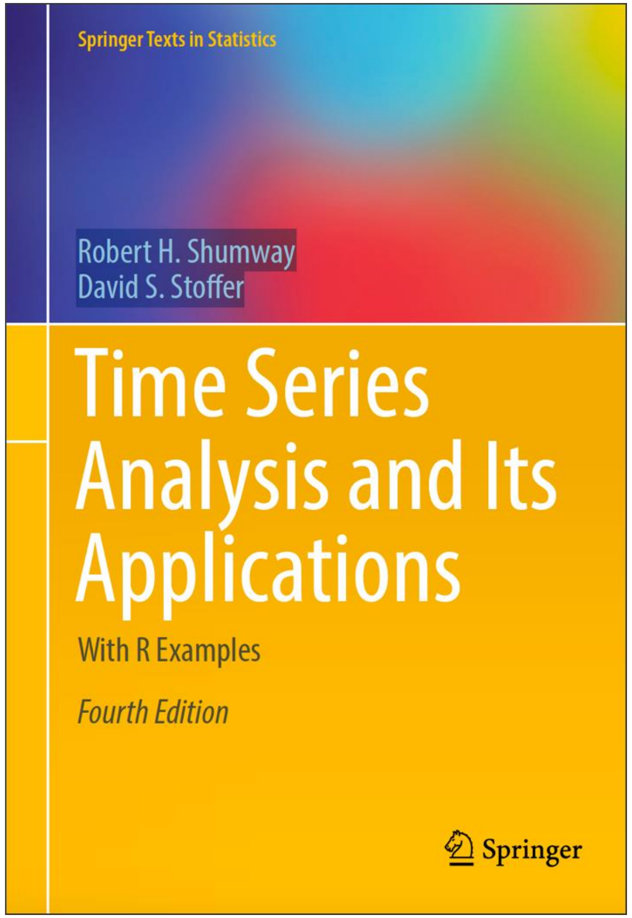
Summary of Time Series Similarity

- If you have short time series
  - use DTW after searching over the warping window size
  - try also to approximate to speed up the calculus

- If you have long time series
  - if you do know something about your data =>
    extract features
  - (and you know nothing about your data =>
    try compression/approximation-based dissimilarity)

References

- Forecasting: Principles and Practic. Rob J Hyndman and George Athanasaopoulus. (https://otexts.com/fpp2/)
- Time Series Analysis and Its Applications. Robert H. Shumway and David S. Stoffer. 4th edition.(http://www.stat.ucla.edu/~frederic/415/S23/tsa4.pdf)
- Mining Time Series Data. Chotirat Ann Ratanamahatana et al. 2010. (https://www.researchgate.net/publication/227001229_Mining_Time_Series_Data)
- Dynamic Programming Algorithm Optimization for Spoken Word Recognition. Hiroaki Sakode et al. 1978.
- Experiencing SAX: a Novel Symbolic Representation of Time Series. Jessica Line et al. 2009
- Compression-based data mining of sequential data. Eamonn Keogh et al. 2007.



Exercises DTW

DTW – Exercise 1

- Given the following input time series:

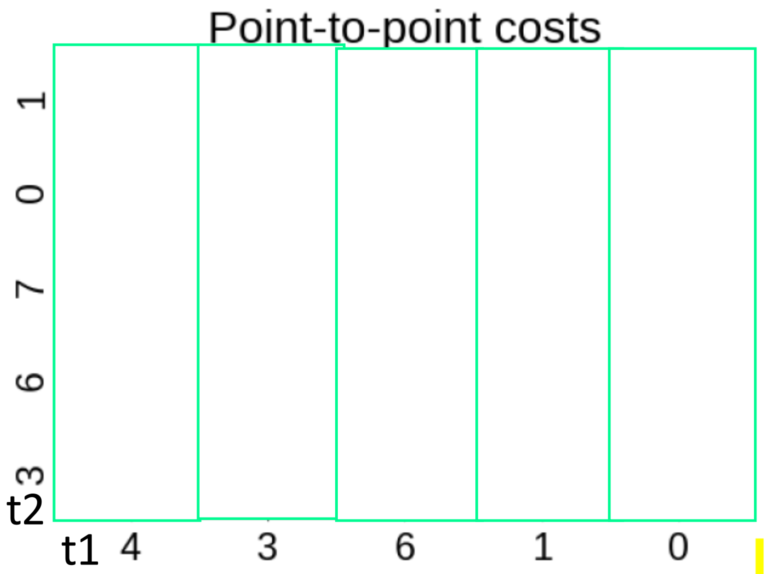
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A) Compute the distance between “t1” and “t2”, using the DTW with distance between points computed as $d(x,y) = |x - y|$.
- B) If we repeat the computation of point (A) above, this time with a Sakoe-Chiba band of size r=1, does the result change? Why?
- C) If we compute DTW(T1,T2), where T1 is equal to t1 in reverse order (namely T1=<0,1,6,3,4>) and similarly for T2 (namely T2=<1,0,7,6,3>), is it true that DTW(T1,T2) = DTW(t1,t2)? Discuss the problem without providing any computation.

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)

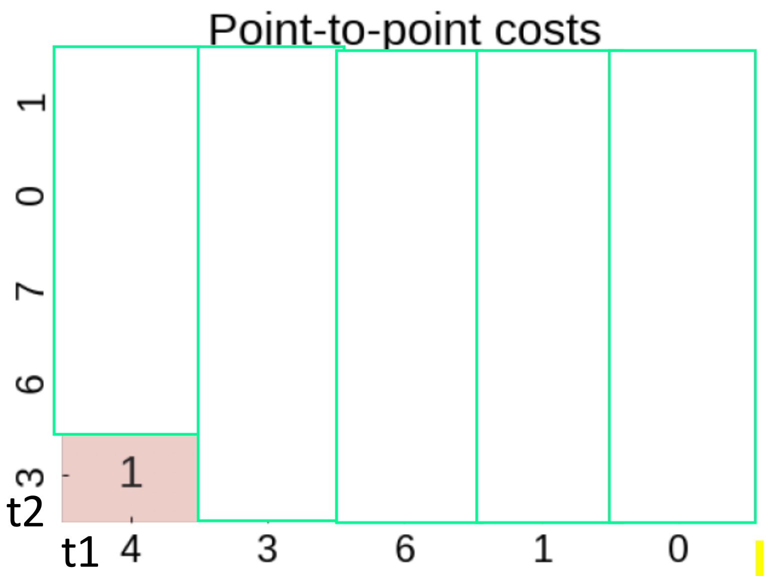


Result: 4

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)

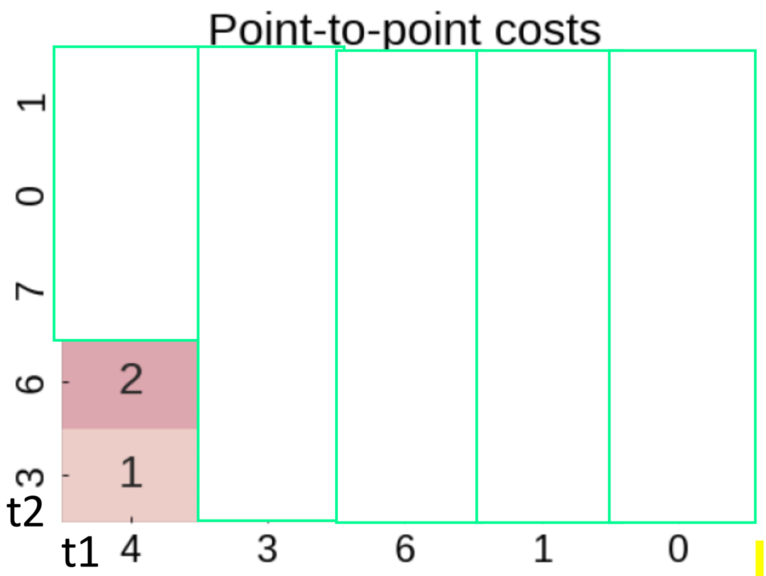


Result: 4

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)

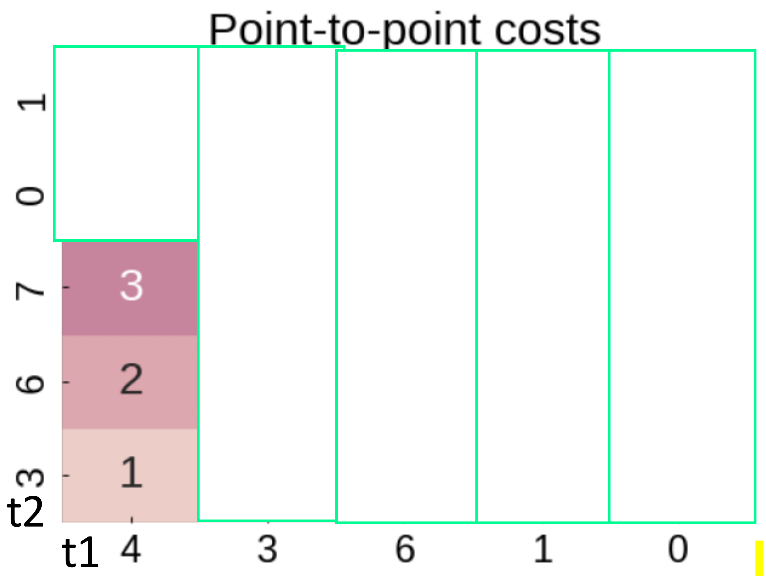


Result: 4

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)

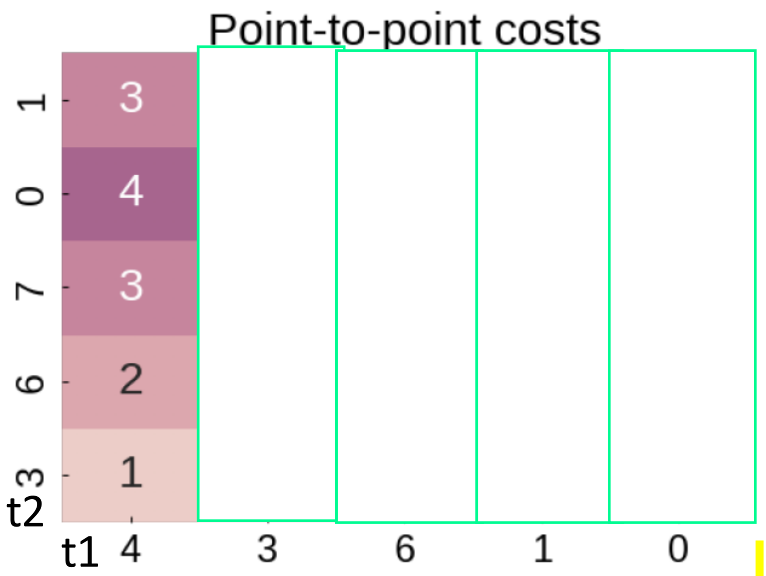


Result: 4

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)

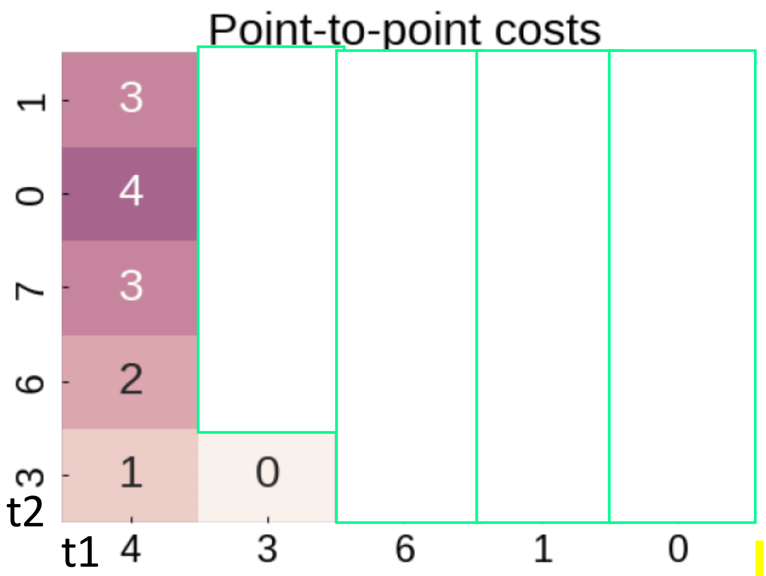


Result: 4

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)

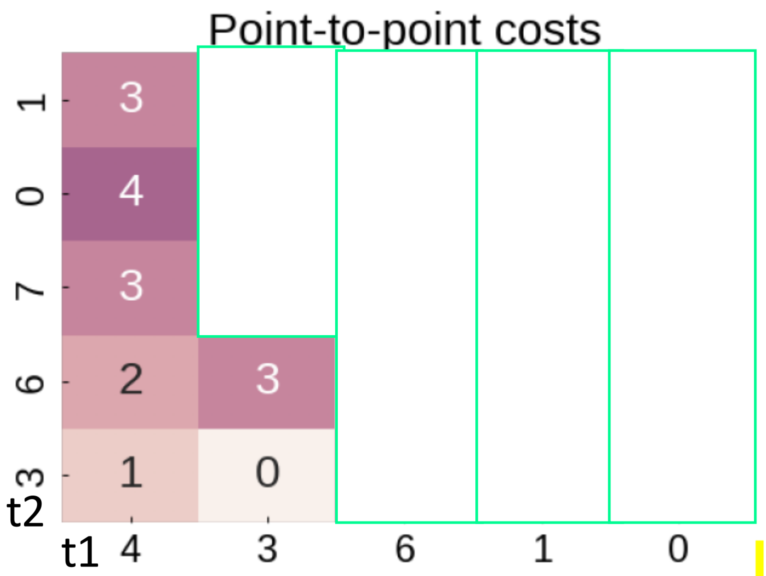


Result: 4

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)

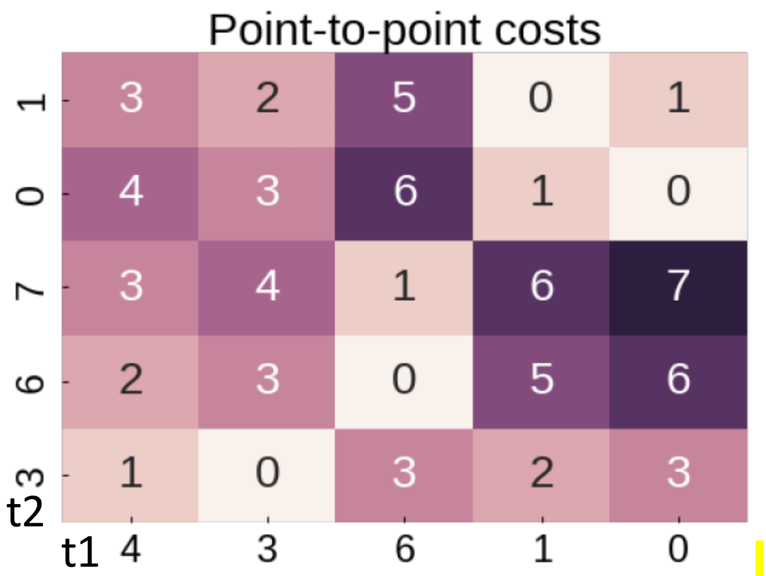


Result: 4

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)

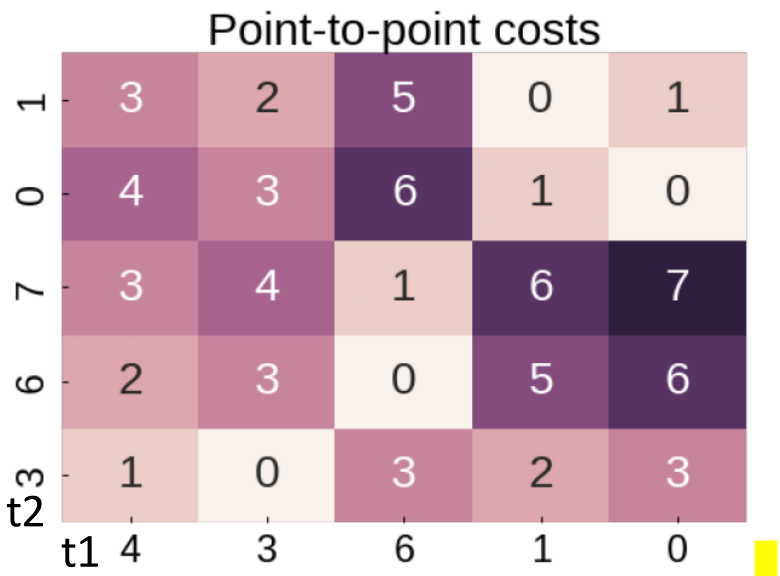


Result: 4

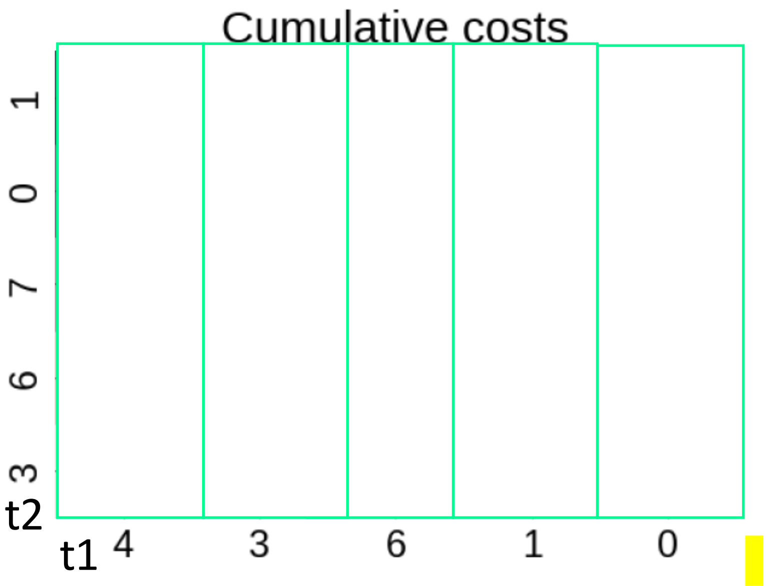
DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)



Result: 4

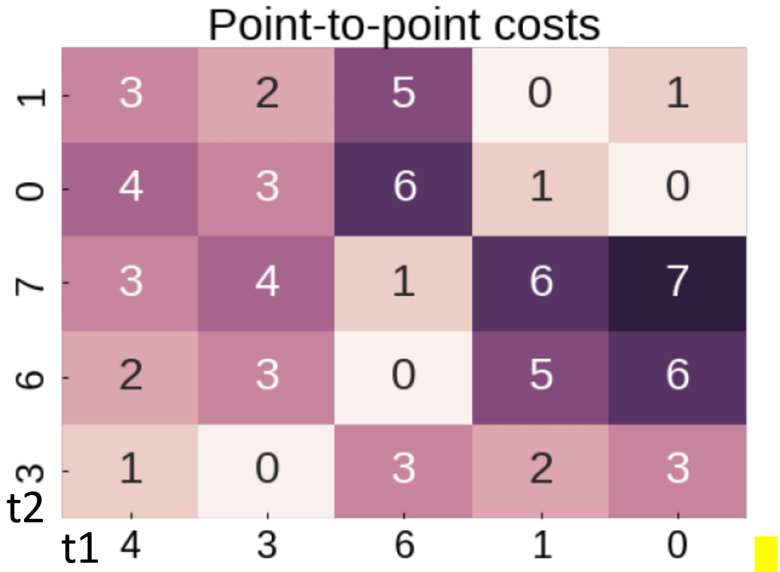


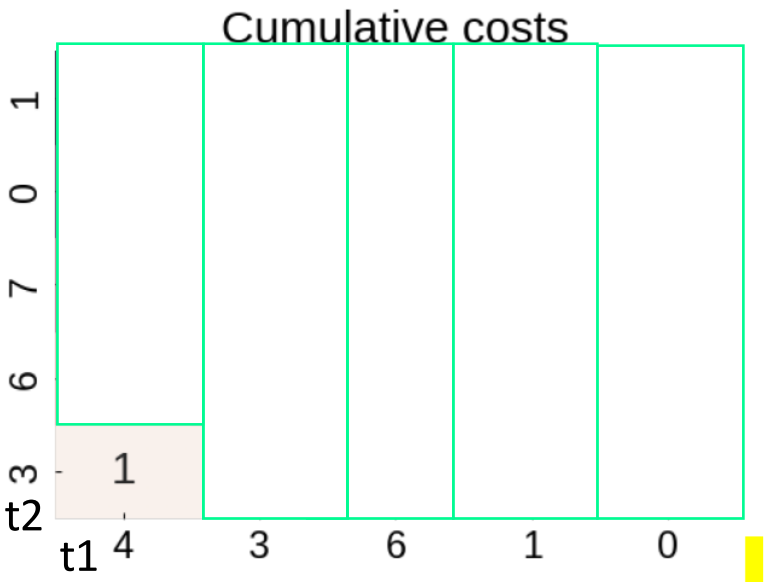
$$\gamma(i,j) = d(q_i,c_j) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)





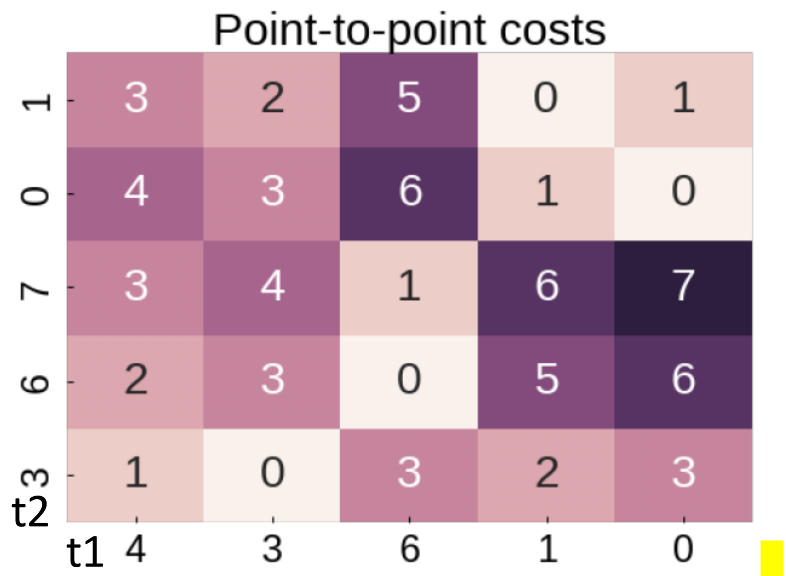
Result: 4

$$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i-1,j-1),\, \gamma(i-1,j\,),\, \gamma(i,j-1)\,\}$$

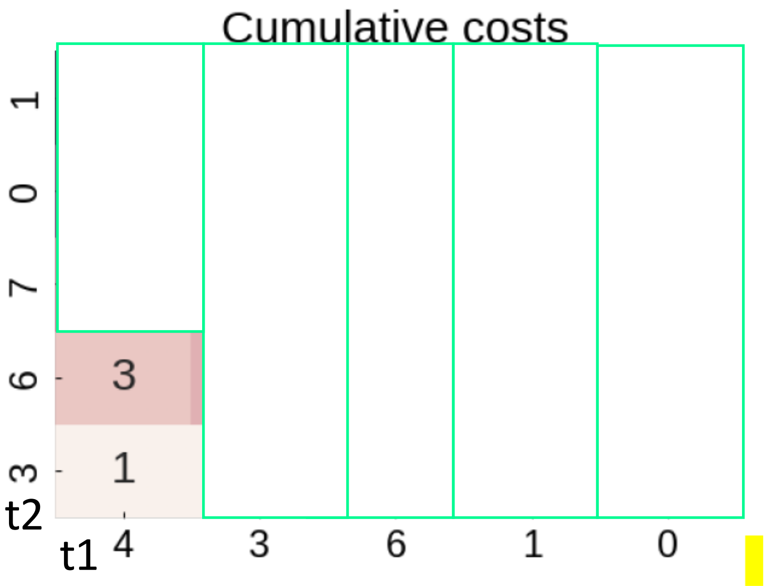
DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)



Result: 4

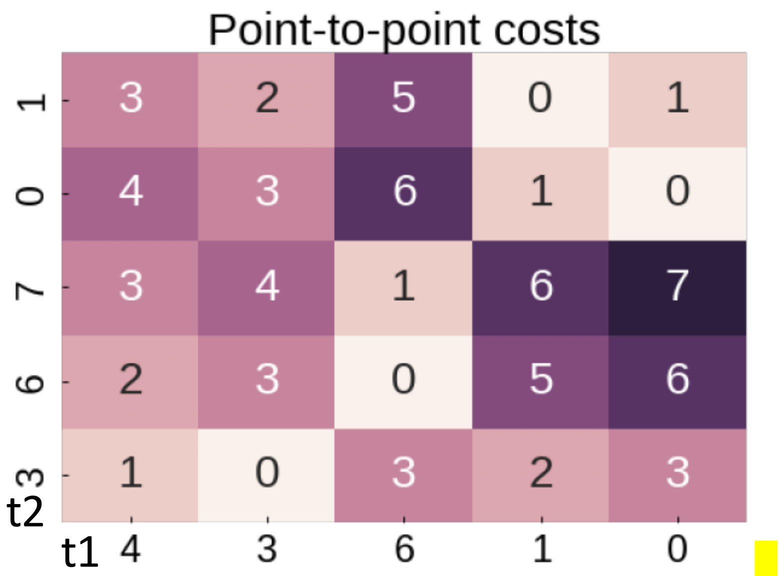


$$\gamma(i,j) = d(q_i,c_j) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

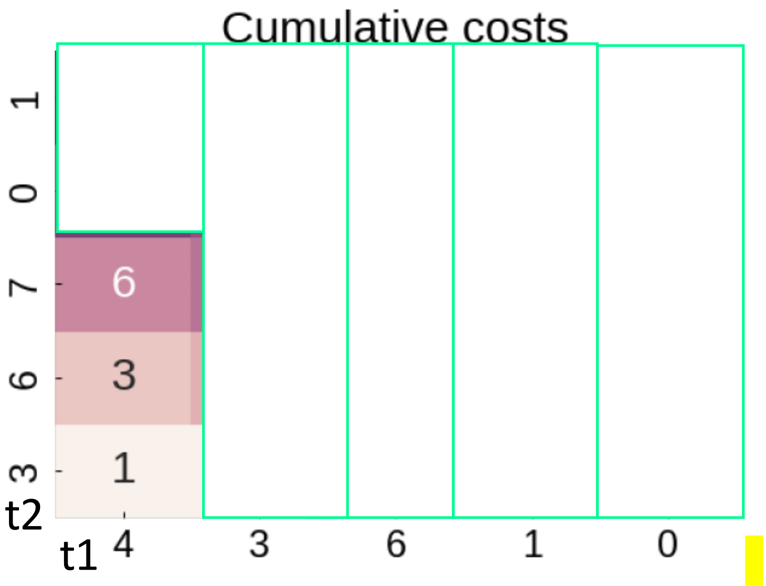
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)



Result: 4

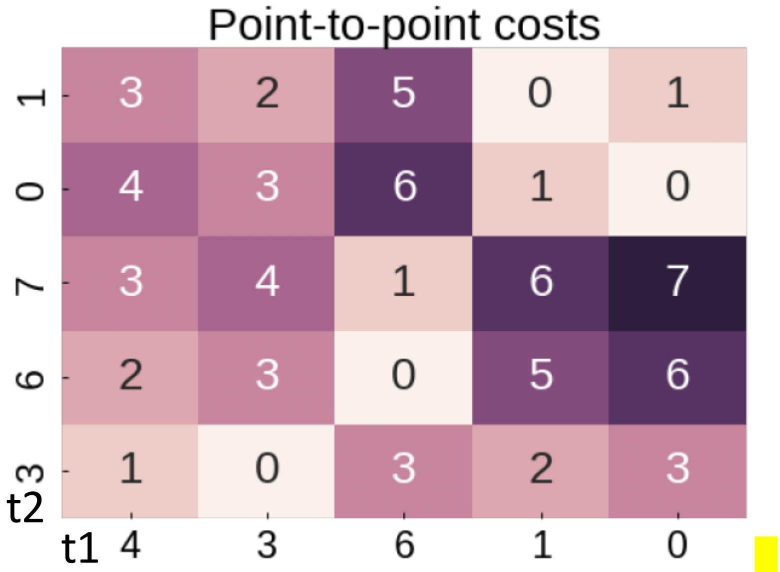


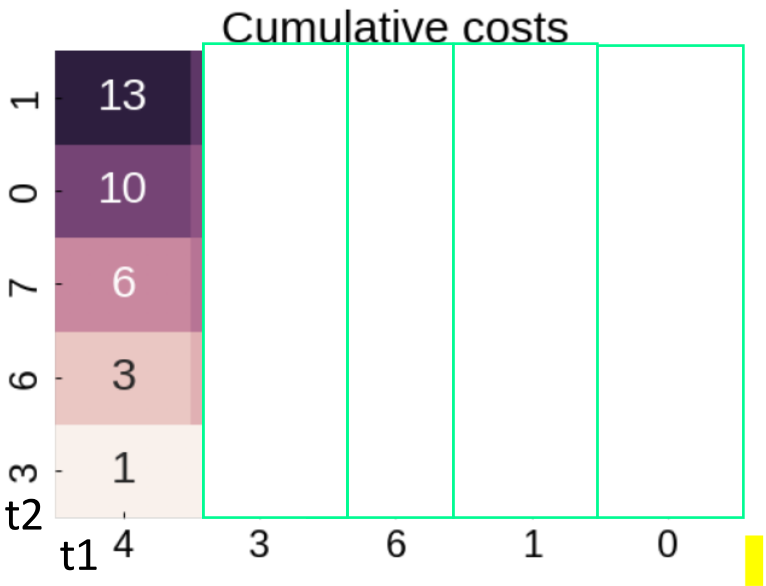
$$\gamma(i,j) \quad = \quad d(q_i,c_j) + \min\{\ \gamma(i\text{-}1,j\text{-}1),\ \gamma(i\text{-}1,j\ ),\ \gamma(i,j\text{-}1)\ \}$$

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)





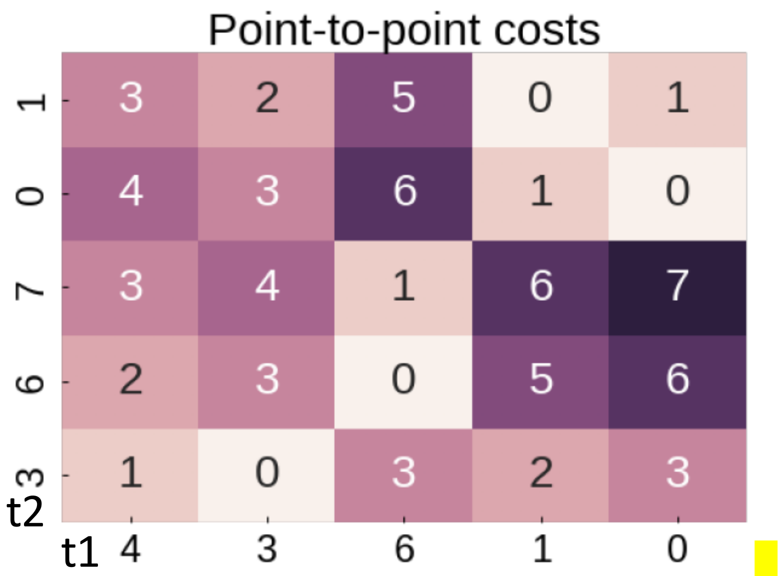
Result: 4

$$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i\text{-}1,j\text{-}1),\, \gamma(i\text{-}1,j\,),\, \gamma(i,j\text{-}1)\,\}$$

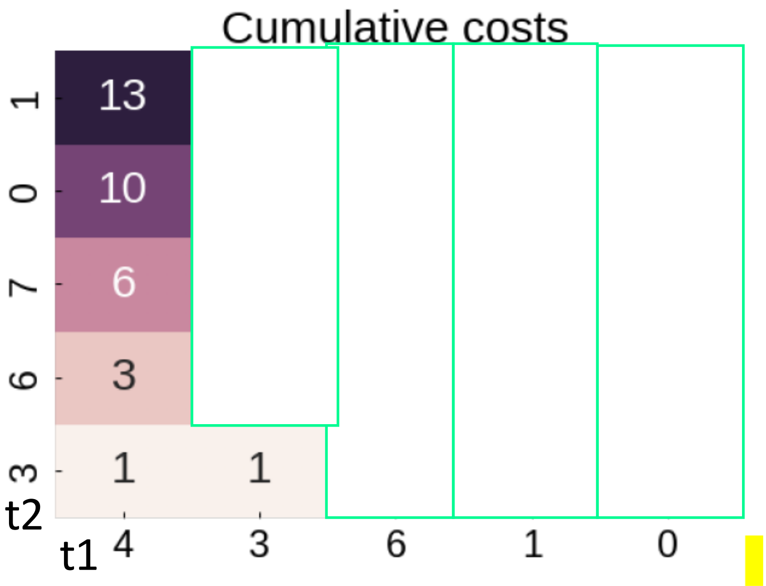
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)



Result: 4

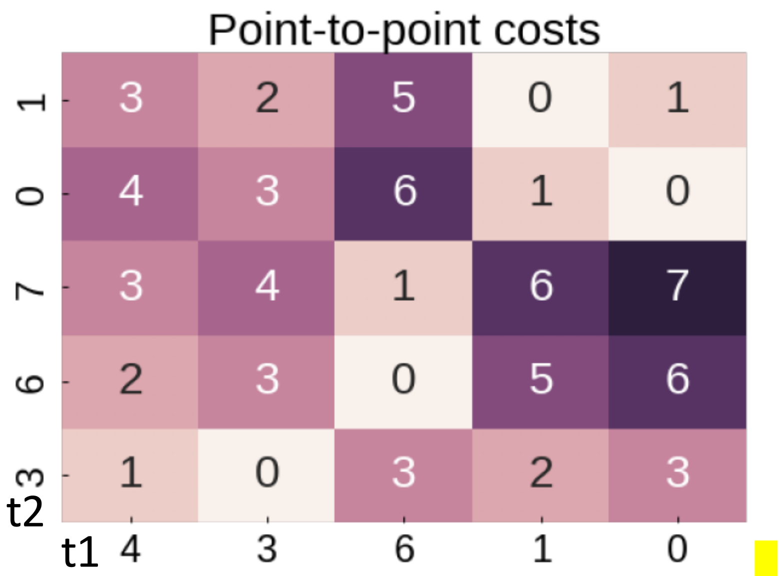


$$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

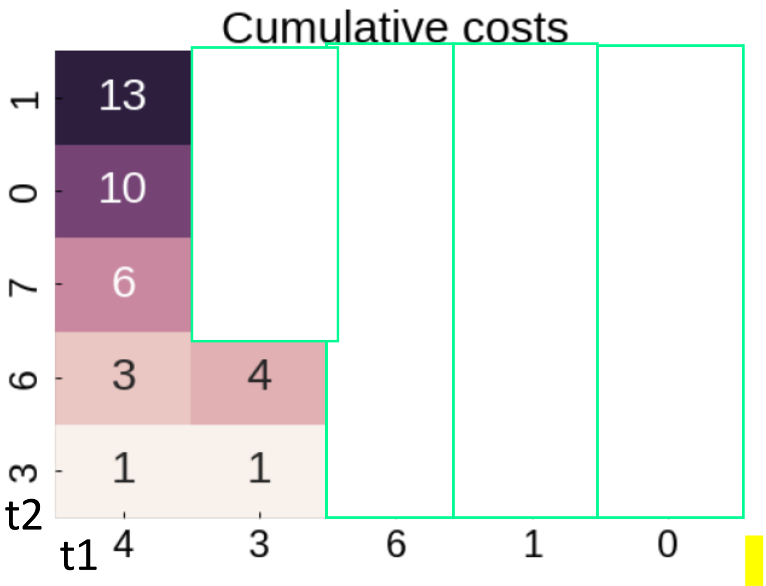
DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)



Result: 4

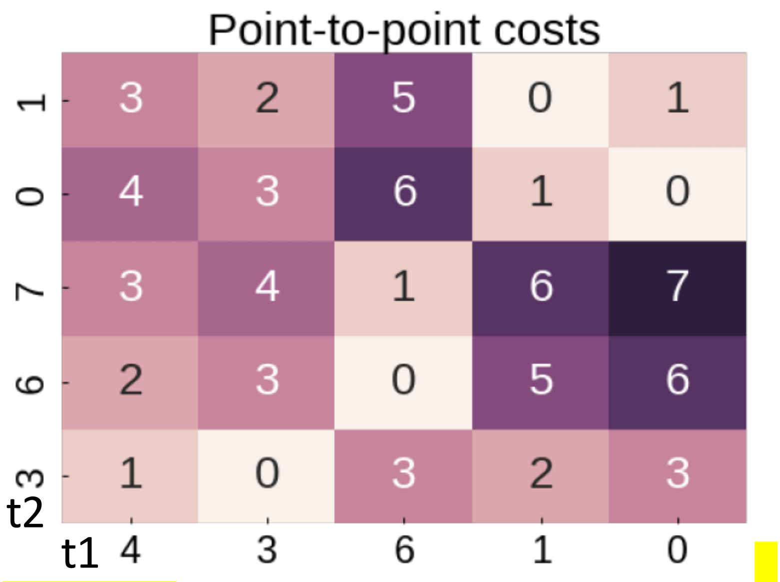


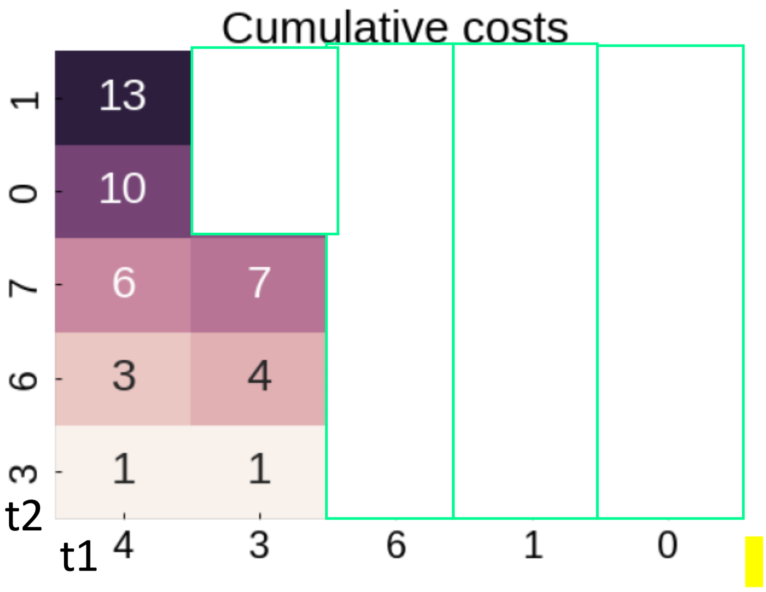
$$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i\text{-}1,j\text{-}1),\ \gamma(i\text{-}1,j\,),\ \gamma(i,j\text{-}1)\,\}$$

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)





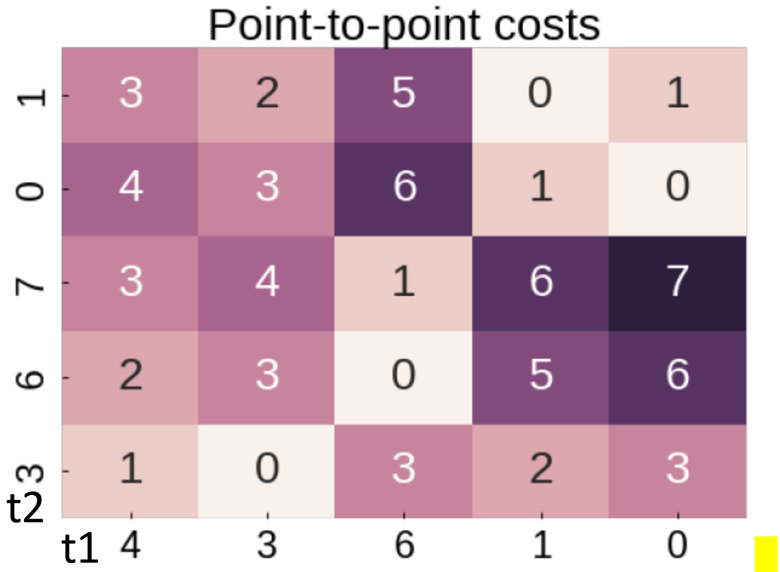
Result: 4

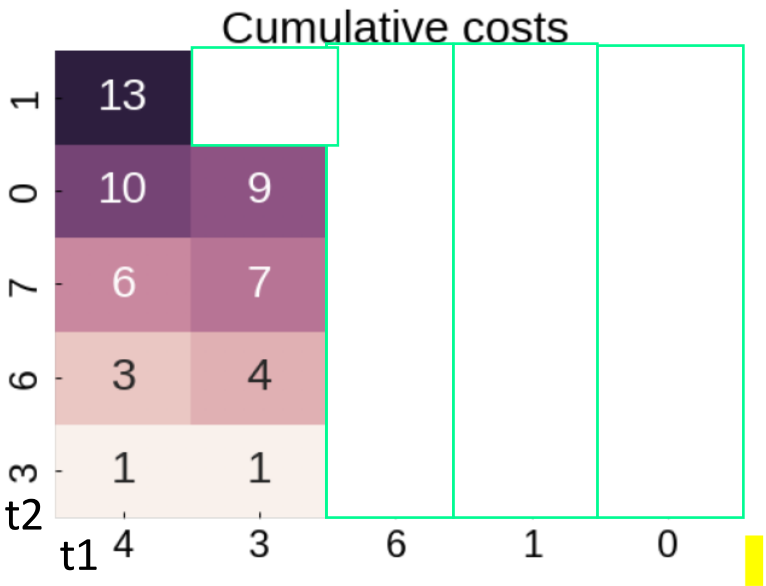
$$\gamma(i,j) \;=\; d(q_i,c_j) + \min\{\, \gamma(i\text{-}1,j\text{-}1),\, \gamma(i\text{-}1,j\,),\, \gamma(i,j\text{-}1)\,\}$$

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)





Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)

Point-to-point costs

t2 \ t1	4	3	6	1	0
1	3	2	5	0	1
0	4	3	6	1	0
7	3	4	1	6	7
6	2	3	0	5	6
3	1	0	3	2	3

Result: 4

Cumulative costs

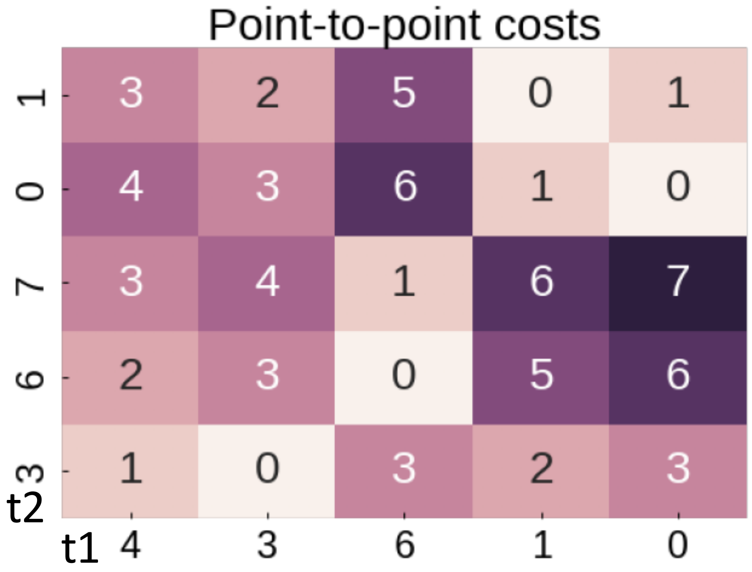
t2 \ t1	4	3	6	1	0
1	13	11			
0	10	9			
7	6	7			
6	3	4			
3	1	1			

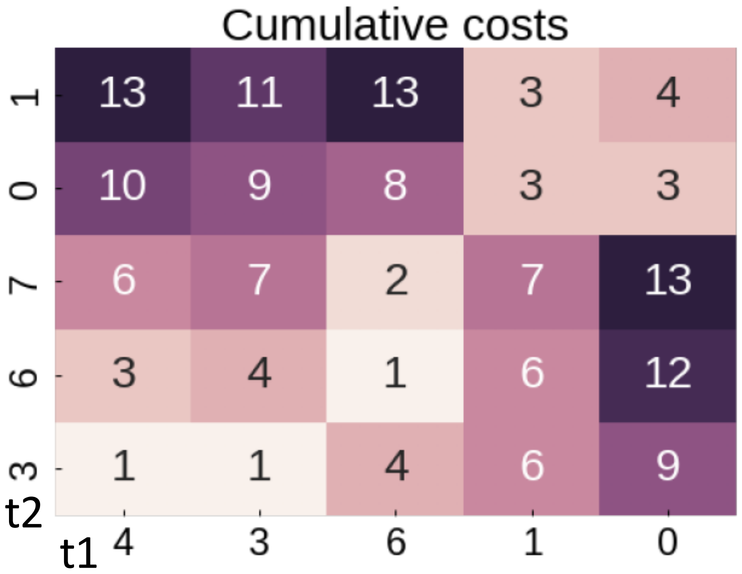
$$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

DTW – Exercise 1 - Solution

- A)





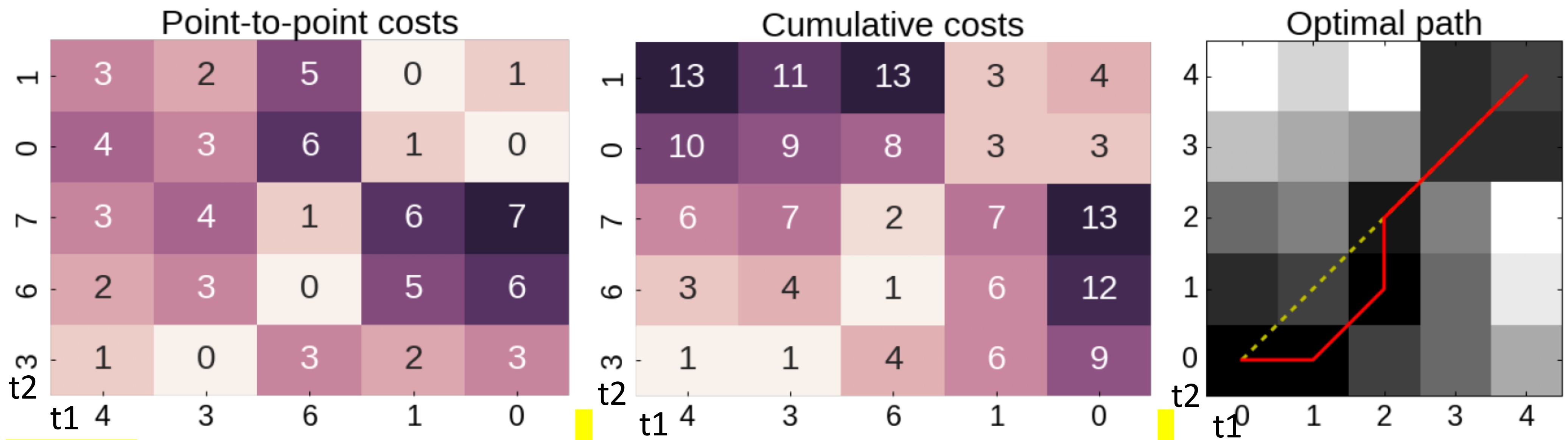
Result: 4

$$\gamma(i,j) = d(q_i,c_j) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >

- A)



Result: 4

- B) No. Because the DTW optimal path remains inside the band of size r=1
- C) Yes. The optimal path in one direction is the same in the opposite direction. Though, the cumulative costs matrix might look different.

DTW – Exercise 2

- Given the following time series:

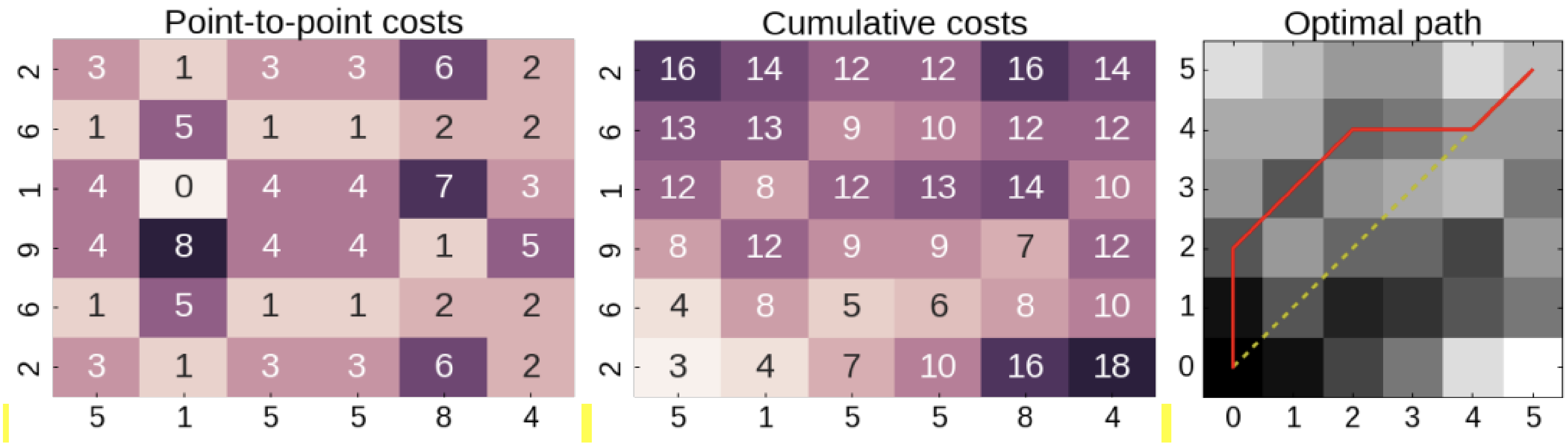
t	=	< 2, 6, 9, 1, 6, 2 >
q	=	< 5, 1, 5, 5, 8, 4 >

compute

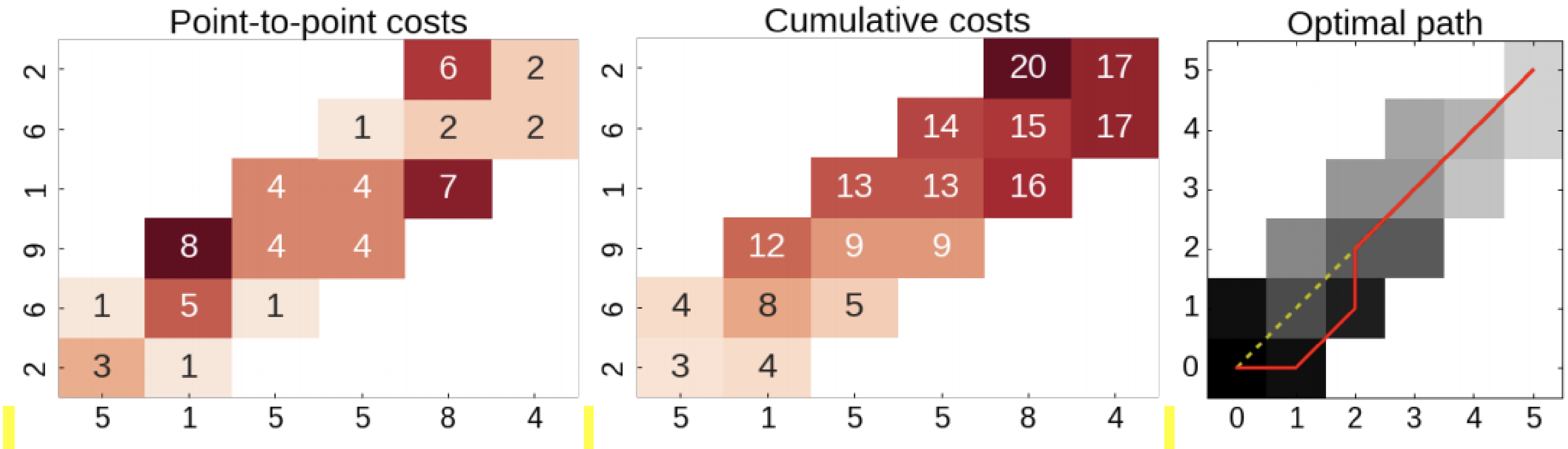
- (i) their Manhattan and Euclidean distance,
- (ii) their DTW, and (iii) their DTW with Sakoe-Chiba band of size r=1 (i.e. all cells at distance <= 1 from the diagonal are allowed).
- For points (ii) and (iii) show the cost matrix and the optimal path found.

DTW – Exercise 2 - Solution

- Euclidean = sqrt(74) = 8.6, Manhattan = 20
- DTW = 14



- DTW r=1 = 17



DTW – Exercise 3

- Given the following time series:

ID	Time series
W	< 6, 11, 13, 15 >
X	< 10, 7, 7, 12, 14, 17 >
Y	< 9, 11, 14, 13, 20 >

- Compute the distances among all pairs of time series adopting a Dynamic Time Warping distance, and computing the distances between single points as $d(x,y) = | x - y |$. For each pair of time series compared also show the matrix used to compute the final result.

ID	Time series
W	< 6, 11, 13, 15 >
X	< 10, 7, 7, 12, 14, 17 >
Y	< 9, 11, 14, 13, 20 >

DTW – Exercise 3 - Solution

W – X

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	(4) 4	(1) 5	(1) 6	(6) 12	(8) 20	(11) 31
[2,]	(1) 5	(4) 8	(4) 9	(1) 7	(3) 10	(6) 16
[3,]	(3) 8	(5) 11	(5) 14	(1) 8	(1) 8	(4) 12
[4,]	(5) 13	(8) 16	(8) 19	(3) 11	(4) 9	(2) 10

W – Y

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	(3) 3	(5) 8	(8) 16	(7) 23	(14) 37
[2,]	(2) 5	(0) 3	(3) 6	(2) 8	(9) 17
[3,]	(5) 9	(2) 5	(1) 4	(0) 4	(7) 11
[4,]	(6) 15	(4) 9	(1) 5	(2) 6	(5) 9

X – Y

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	(1) 1	(1) 2	(4) 6	(3) 9	(10) 19
[2,]	(2) 3	(4) 5	(7) 9	(6) 12	(13) 22
[3,]	(2) 5	(4) 7	(7) 12	(6) 15	(13) 25
[4,]	(3) 8	(1) 6	(2) 8	(1) 9	(8) 17
[5,]	(5) 13	(3) 9	(0) 6	(1) 7	(6) 13
[6,]	(8) 21	(6) 15	(3) 9	(4) 10	(3) 10