Mining Time Series:

Computing Similarity

Mining Massive Datasets

Prof. Carlos Castillo — https://chato.cl/teach

Universitat Pompeu Fabra Barcelona

Sources

- Data Mining, The Textbook (2015) by Charu Aggarwal (chapter 14)
- Introduction to Time Series Mining (2006) tutorial by Keogh Eamonn [alt. link]
- Time Series Data Mining (2006) slides by Hung Son Nguyen

Using Euclidean distance on time series

Euclidean distance for time series

- Euclidean distance between series y and z

$$d(y, z) = \sqrt{\sum_{i=1}^{n} (y_i - z_i)^2}$$

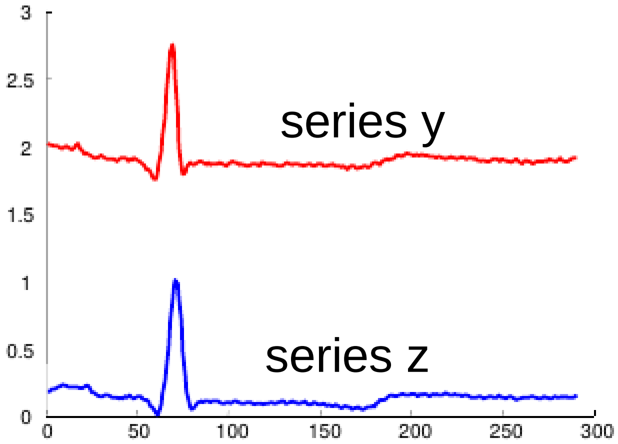
python

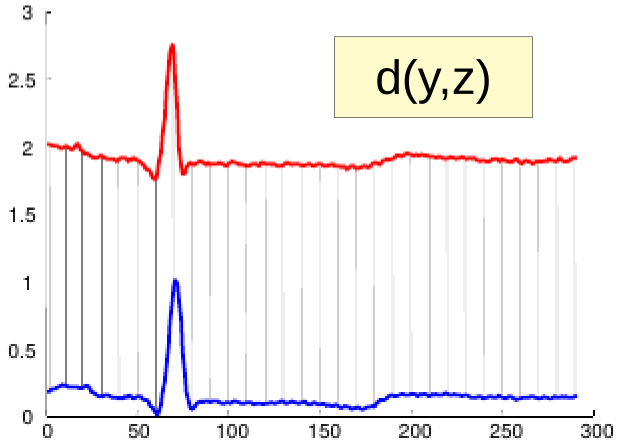
`numpy.linalg.norm(y-z)`

- Sensitive to **noise** (see previous slides on how to fix this)
- Sensitive to different **offsets**, **amplitudes**, and **trends**

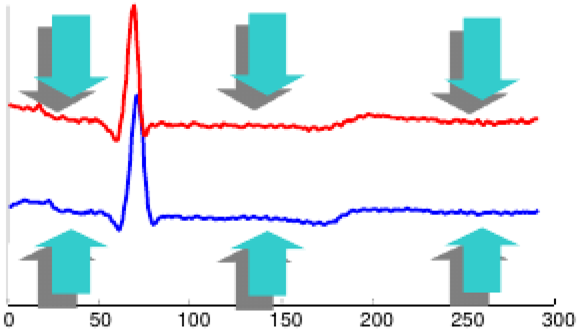
Slides on this section from: Introduction to Time Series Mining (2006) tutorial by Keogh Eamonn [alt. link]

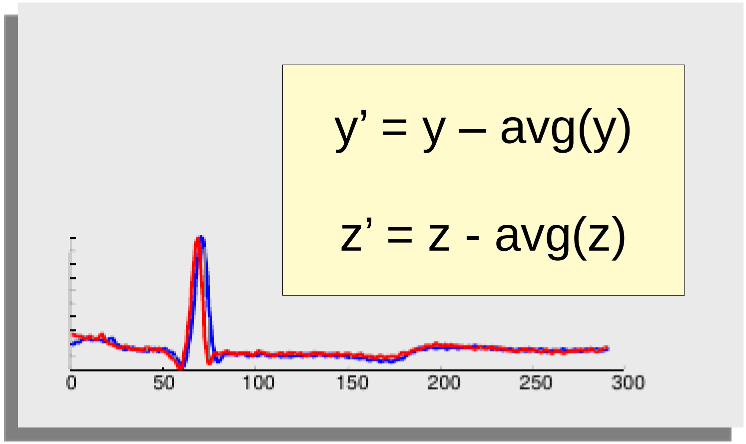
Offset translation: subtract the mean





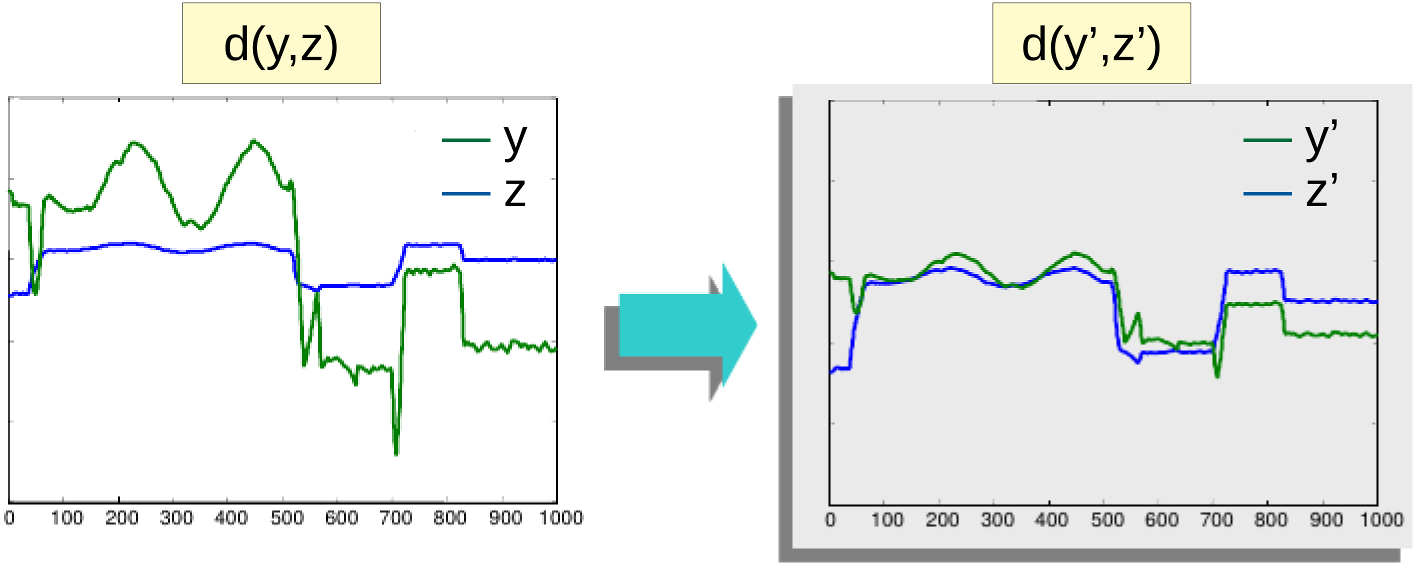
- Series look different





- Series look similar

Amplitude scaling: normalize



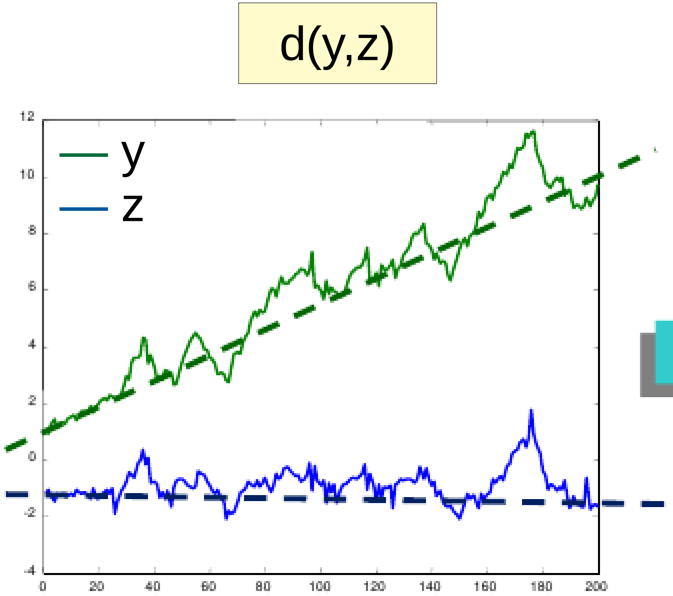
- Standardization

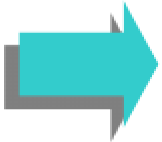
$$y_i' = \frac{y_i - \mathrm{avg}(y)}{\mathrm{std}(y)}$$

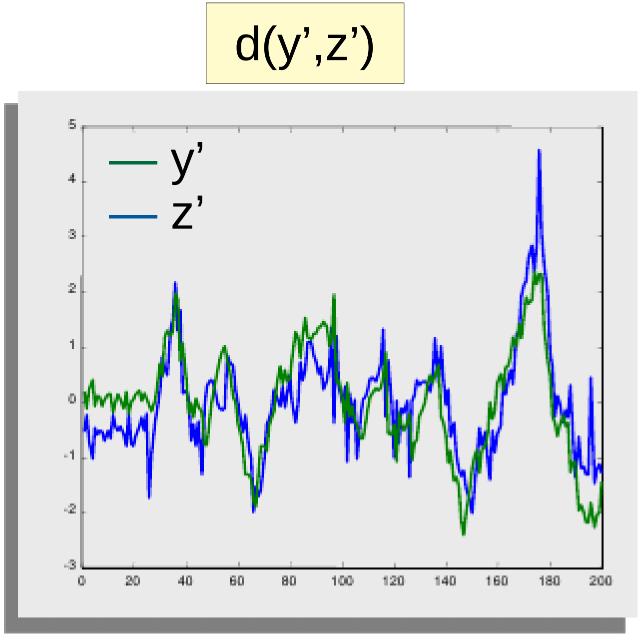
- Range-based normalization

$$y_i' = \frac{y_i - \min(y)}{\max(y) - \min(y)}$$

Trend removal: remove linear trend

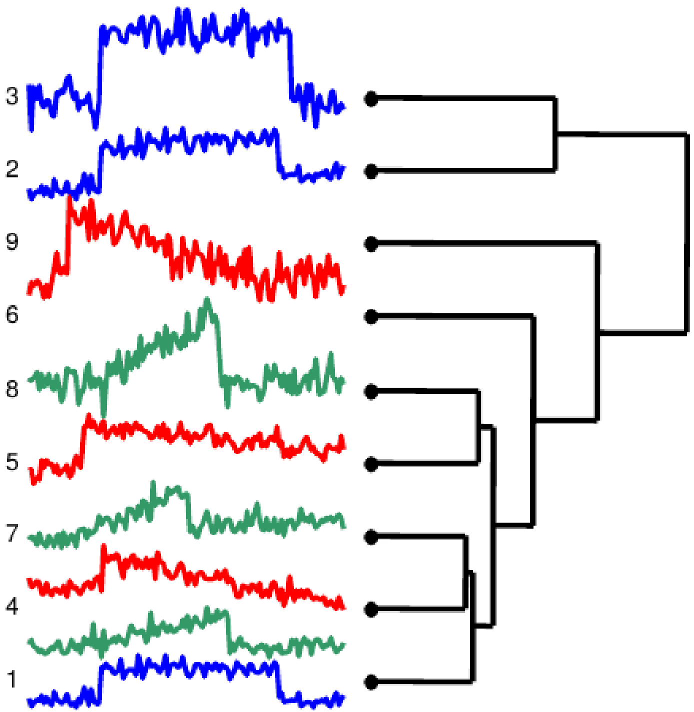




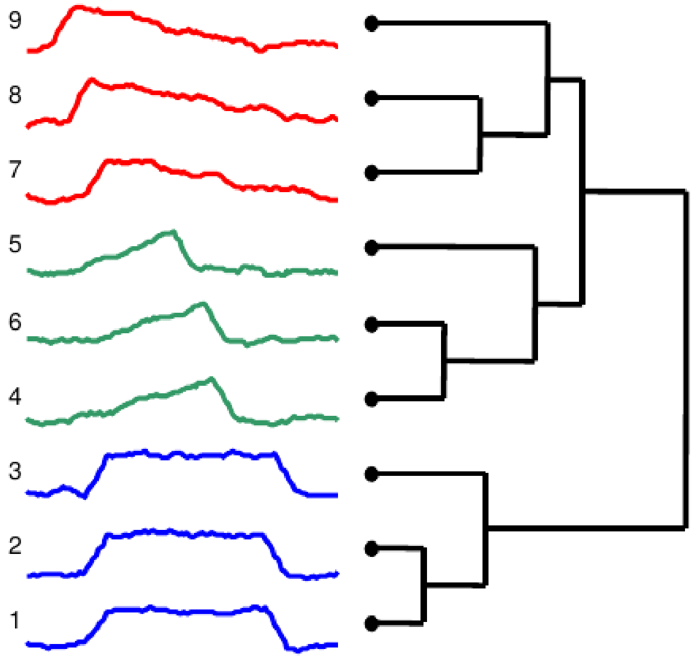


1. Find best straight line fitting data
2. Subtract that line from the data

Example: clustering of time series after using smoothing, offset translation, amplitude scaling, and trend removal



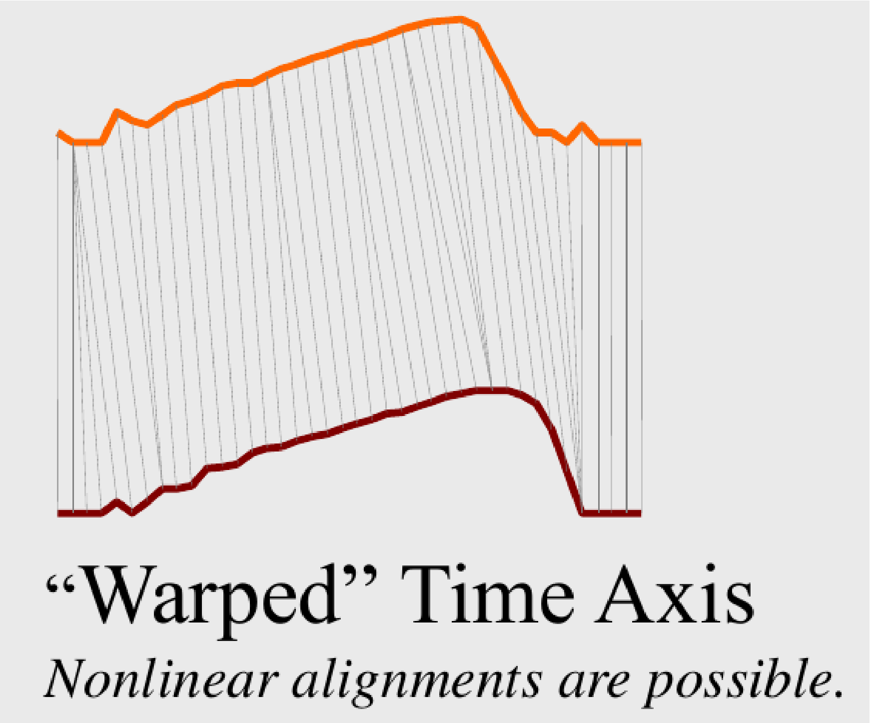
Clustering using euclidean distance on original series



Clustering using euclidean distance on processed series

Dynamic time warping

Dynamic time warping



Slides on this section from: Introduction to Time Series Mining (2006) tutorial by Keogh Eamonn [alt. link]

Dynamic time warping example

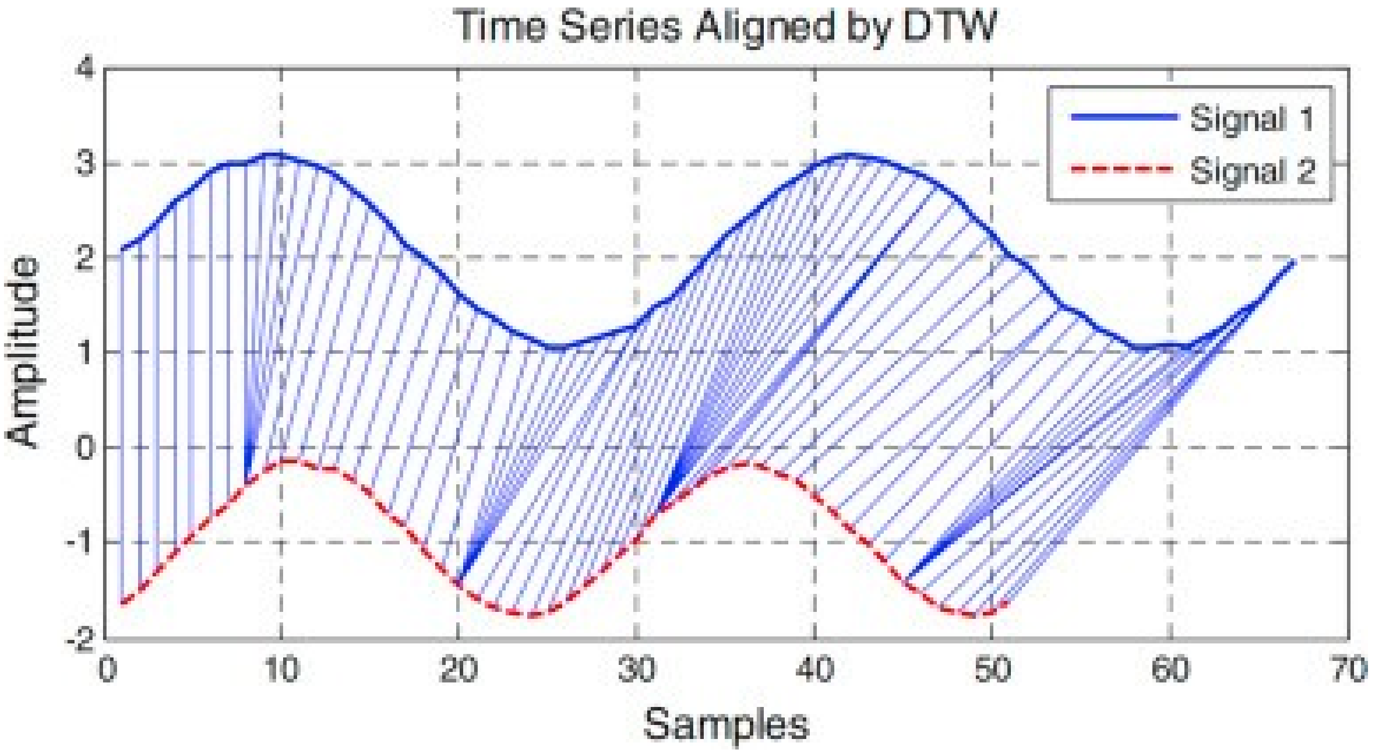
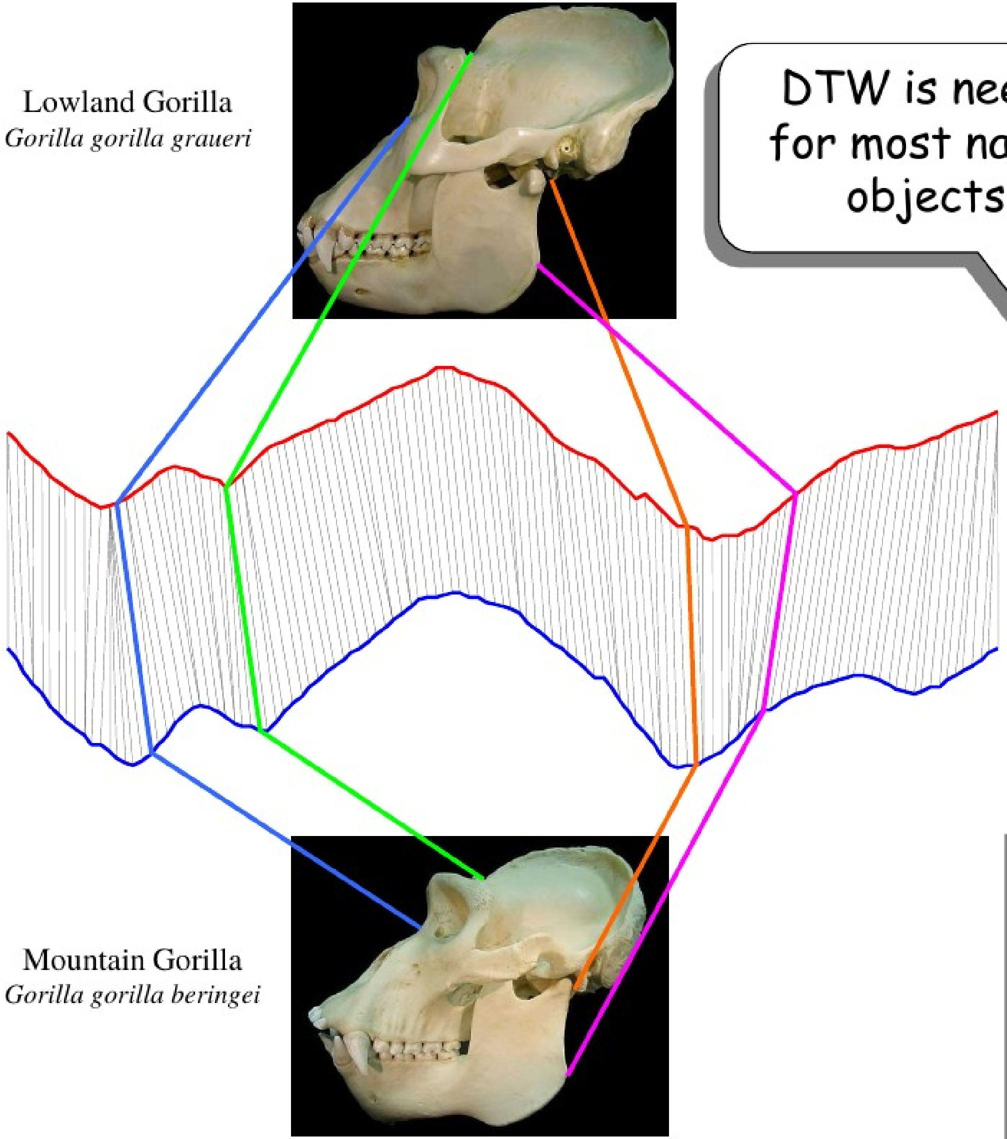
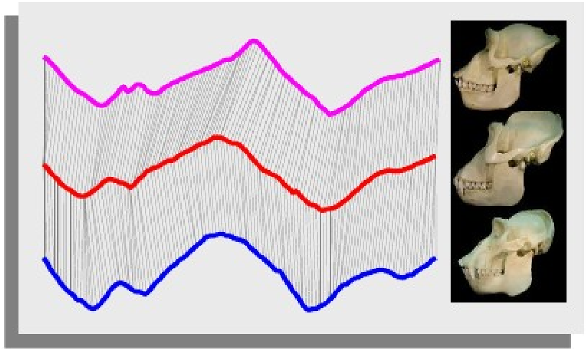


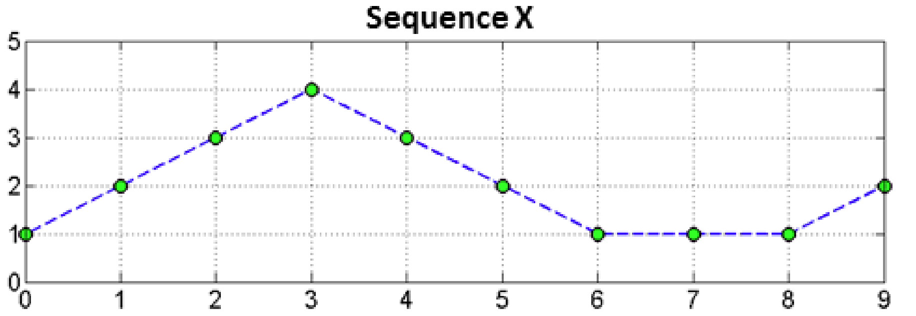
Image credits: Lu et al. 2016



DTW is needed for most natural objects…

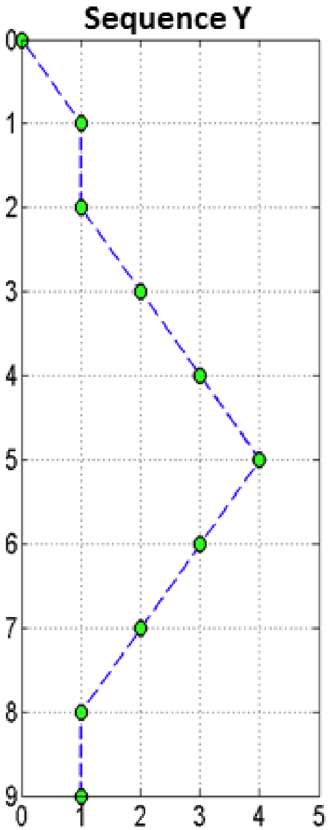


Computing DTW(X,Y)



1) Create a matrix M of size |X|×|Y|

2) Fill-in the matrix using dynamic programming



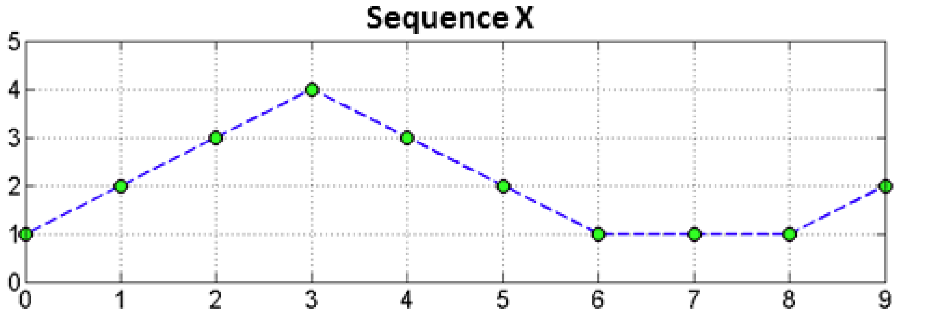
1	5	14	30
1	2	6	15

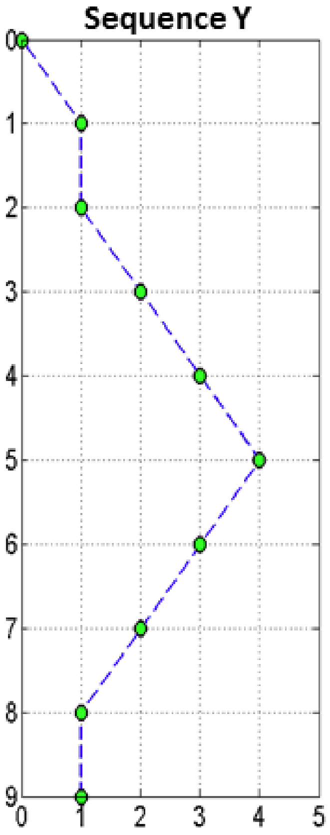
$$
\begin{aligned}
M(i,j) = \quad & d(y_i, x_j) + \\
& \min\{M(i-1, j-1), \\
& \quad M(i-1, j), \\
& \quad M(i, j-1)\}
\end{aligned}
$$

[Source]

Computing DTW(X,Y) (cont.)

1) Create a matrix M of size $|X| \times |Y|$
2) Fill-in the matrix using dynamic programming
3) Find lighter path
4) Cell (a,b) in path $\Rightarrow$ points a,b should be aligned





1	5	14	30	39	43	44	45	46	50
1	2	6	15	19	20	20	20	20	21
1	2	6	15	19	20	20	20	20	21
2	1	2	6	7	7	8	9	10	10
6	2	1	2	2	3	7	11	13	11
15	6	2	1	2	6	12	16	20	15
19	7	2	2	1	2	6	10	14	15
20	7	3	6	2	1	2	3	4	4
20	8	7	12	6	2	1	1	1	2
20	9	11	16	10	3	1	1	1	2

[Source]

Exercise: Dynamic Time Warping

- Compute DTW these two series
- Create the matrix using the formula (remember first row and first column will be different)
- Mark with color the minimum path

t	1	2	3	4	5	6
Y_t	2	5	2	5	3	--
X_t	0	3	6	0	6	1

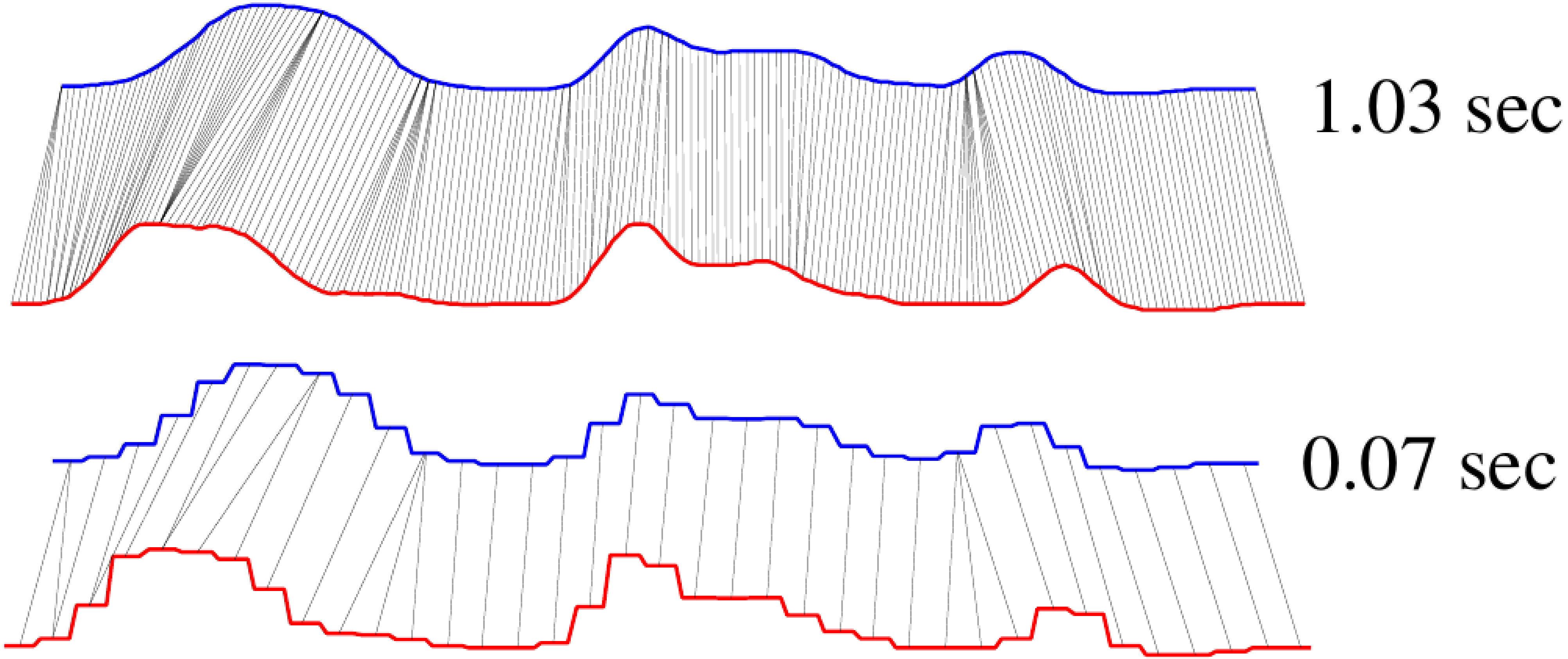
$$M(i,j) = d(y_i, x_j) + \min\{M(i-1,j-1), M(i-1,j), M(i,j-1)\}$$

Spreadsheet links: https://upfbarcelona.padlet.org/chato/hogch321o6pws1fd

Faster DTW through size reduction

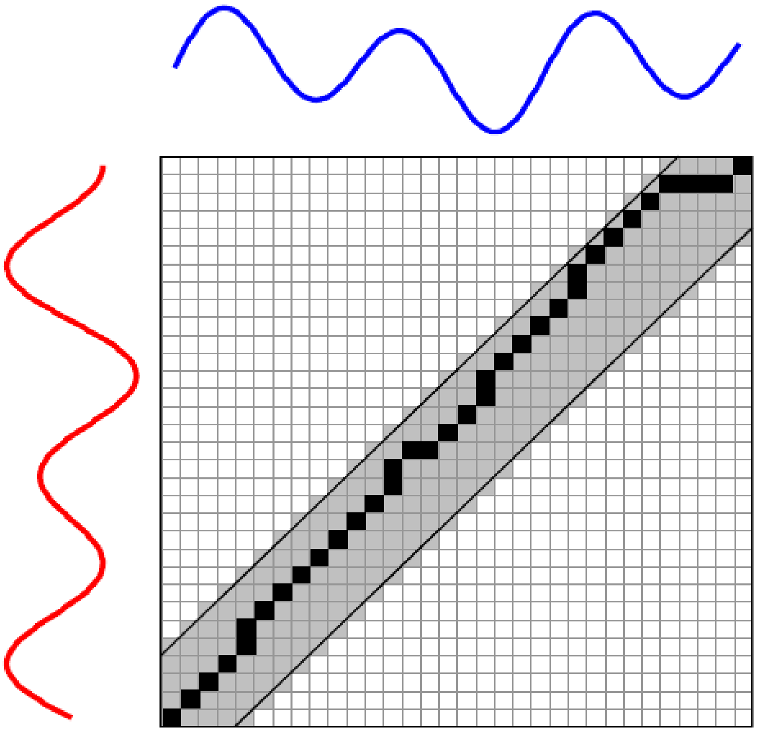
- How to avoid having a large matrix?
- Use less points
  - Sub-sample from original series
  - Bin the original series
- If sampling was done, after doing DTW:
  - Interpolate warpings for intermediate points

Example: faster DTW through sub-sampling

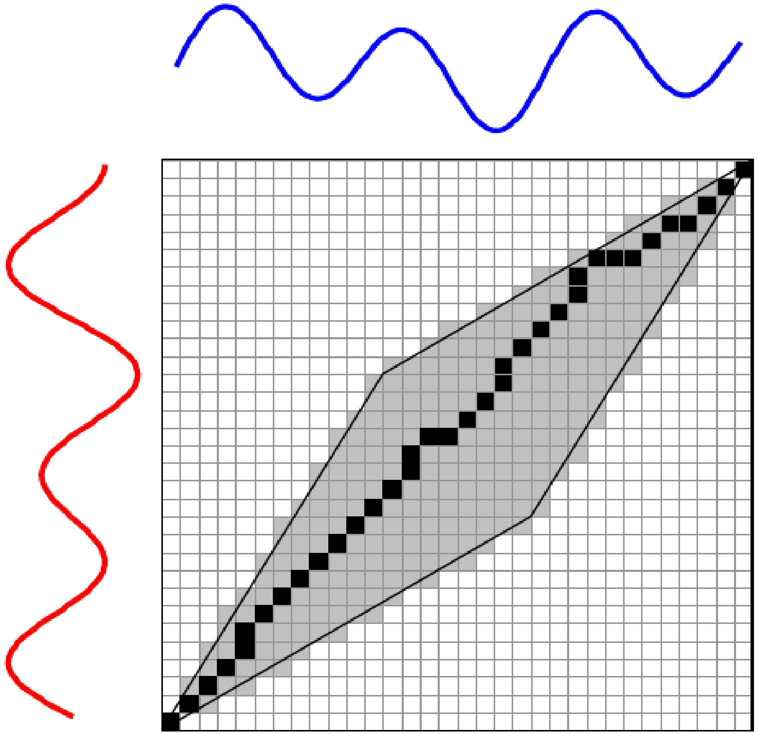


How to avoid pathological warpings?

Assume original series cannot be so far apart from each other, using domain knowledge



Sakoe-Chiba Band



Itakura Parallelogram

Summary

Things to remember

- Dynamic time warping

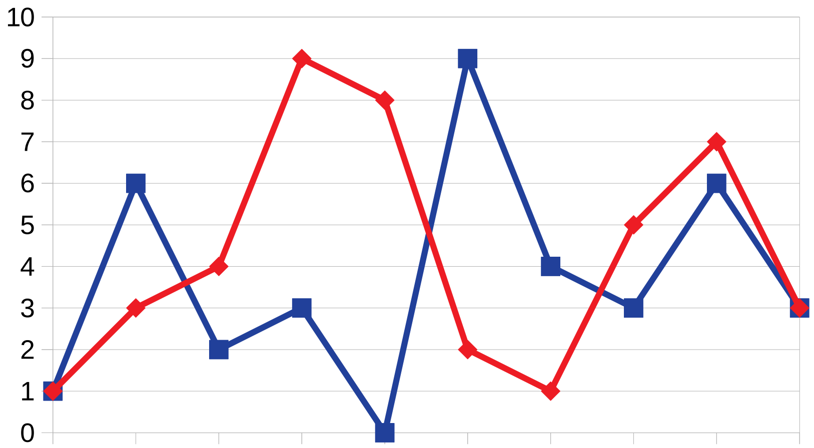
Solved exercise on DTW

- Blue series:

  1, 6, 2, 3, 0, 9, 4, 3, 6, 3

- Red series:

  1, 3, 4, 9, 8, 2, 1, 5, 7, 3



- First try to do it on your own, then you can watch the solution:

  https://youtu.be/_K1OsqCicBY?t=125

Exercises for TT27-TT29

- Data Mining, The Textbook (2015) by Charu Aggarwal
  - Exercises 14.10 $\rightarrow$ 1-6