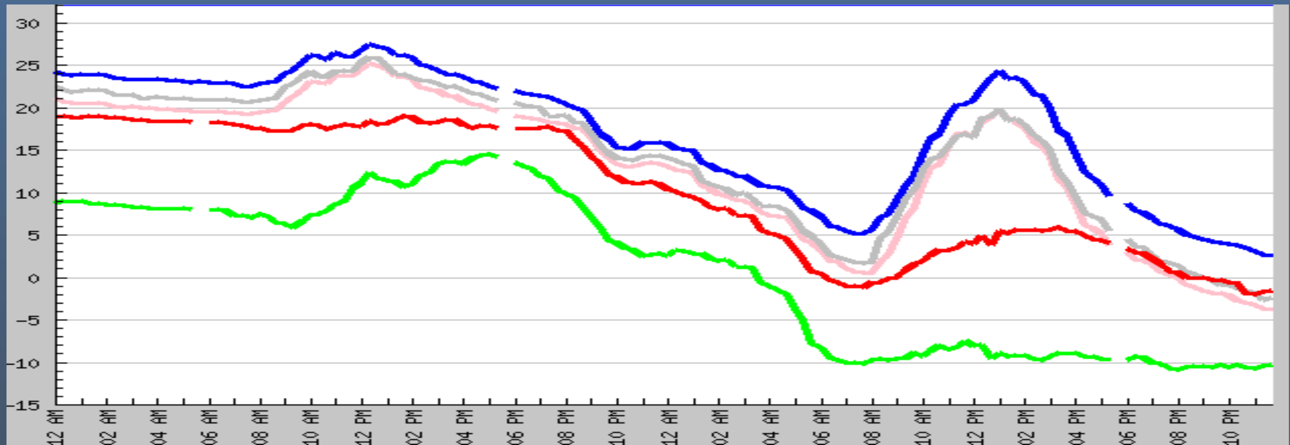
Dynamic Time Warping Algorithm

Slides from:

Elena Tsiporkova

INTRODUCTION

- Time Series
  - **collection of observations made sequentially in time**



  - **Occur in Medical, business, scientific domain**

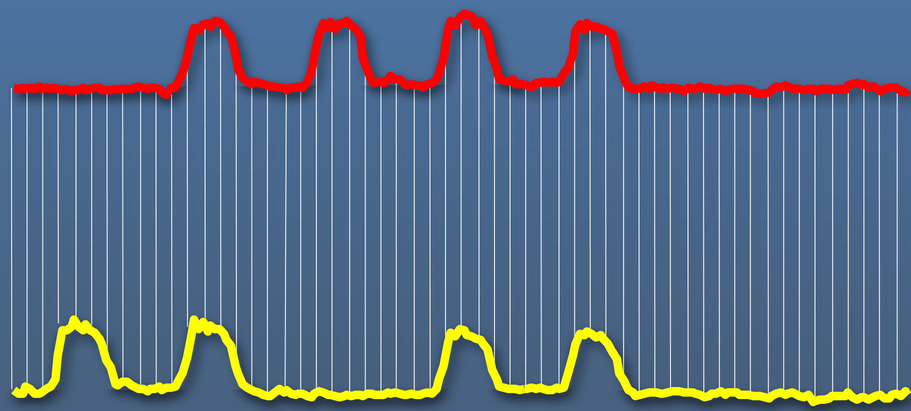
  - **Finding out similarities between two time series is required in many time series data mining applications**

CHALLENGES

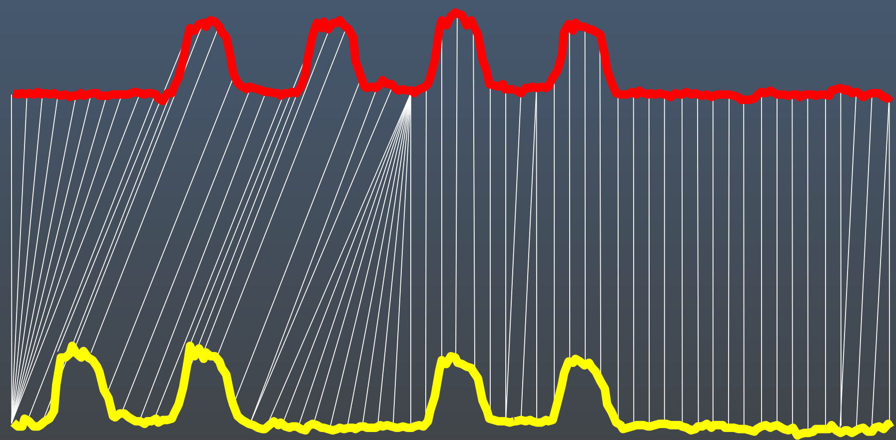
- How do we define similarity ?
- Need a method that allows elastic shifting of time axis to accommodate sequences that are similar but can be out of phase
- Large Amount of data
- How do we search quickly ?

SOLUTIONS

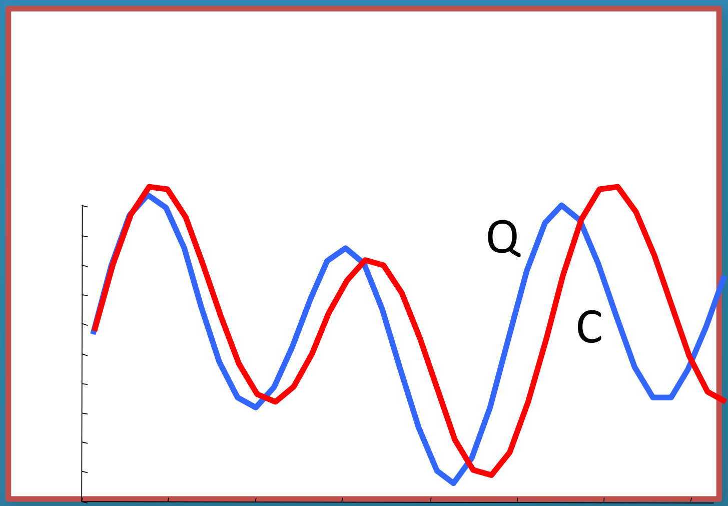
- Euclidean distance
  - Aligned one to one
  - Cannot find similarity b/w out of phase signals

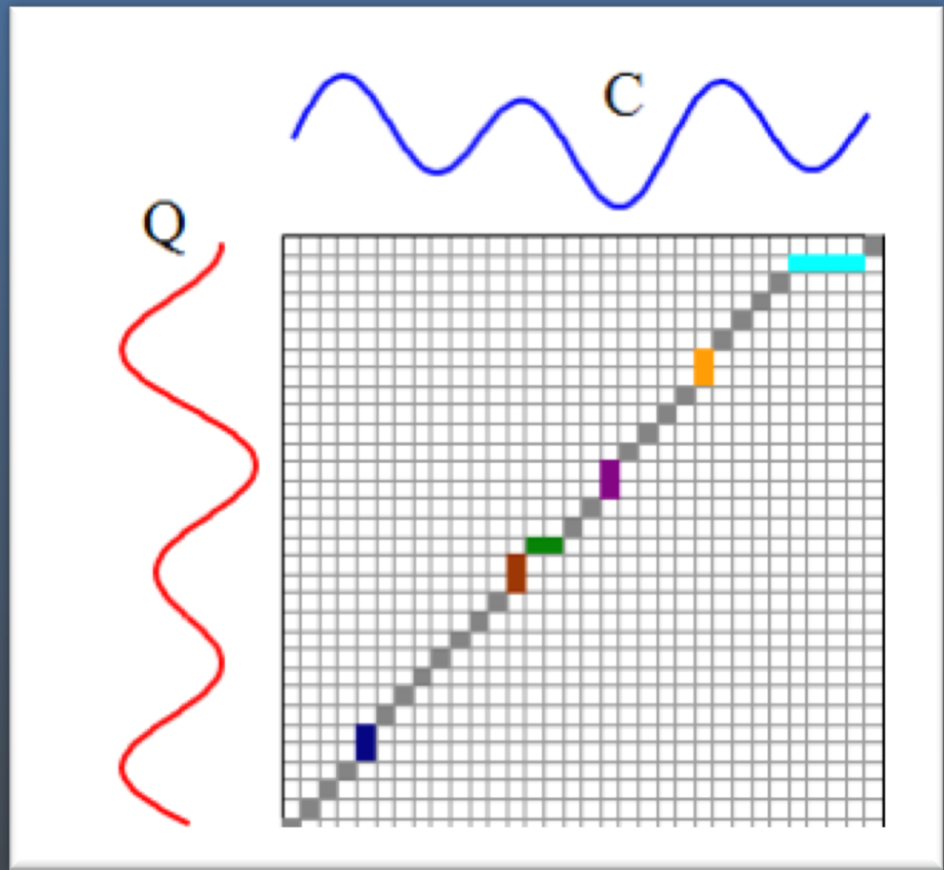


- Dynamic Time Warping
  - Can be non-linearly aligned

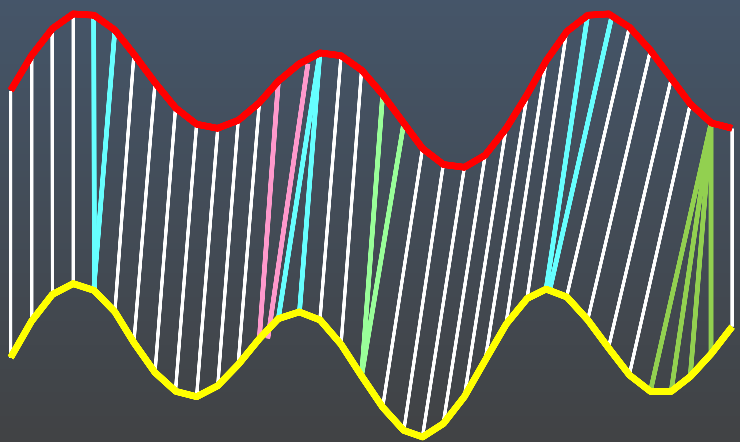


WHAT IS TIME WARPING



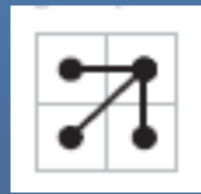


Warping Path

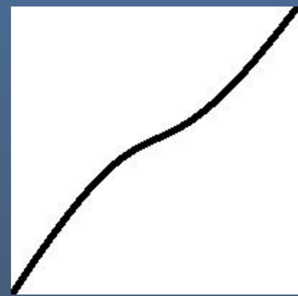


DYNAMIC TIME WARPING

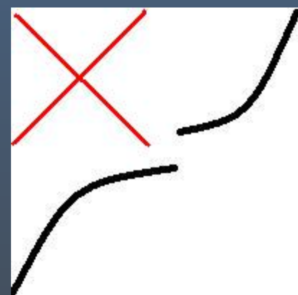
- $\gamma(i,j) = d(q_i,c_j) + \min\{\ \gamma(i\text{-}1,j\text{-}1)\ ,\ \gamma(i\text{-}1,j\ )\ ,\ \gamma(i,j\text{-}1)\ \}$



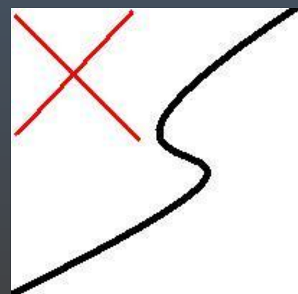
- Three Basic Constraints of Time Warping
  - Path should include beginning and ending



  - Path should not have any jumps

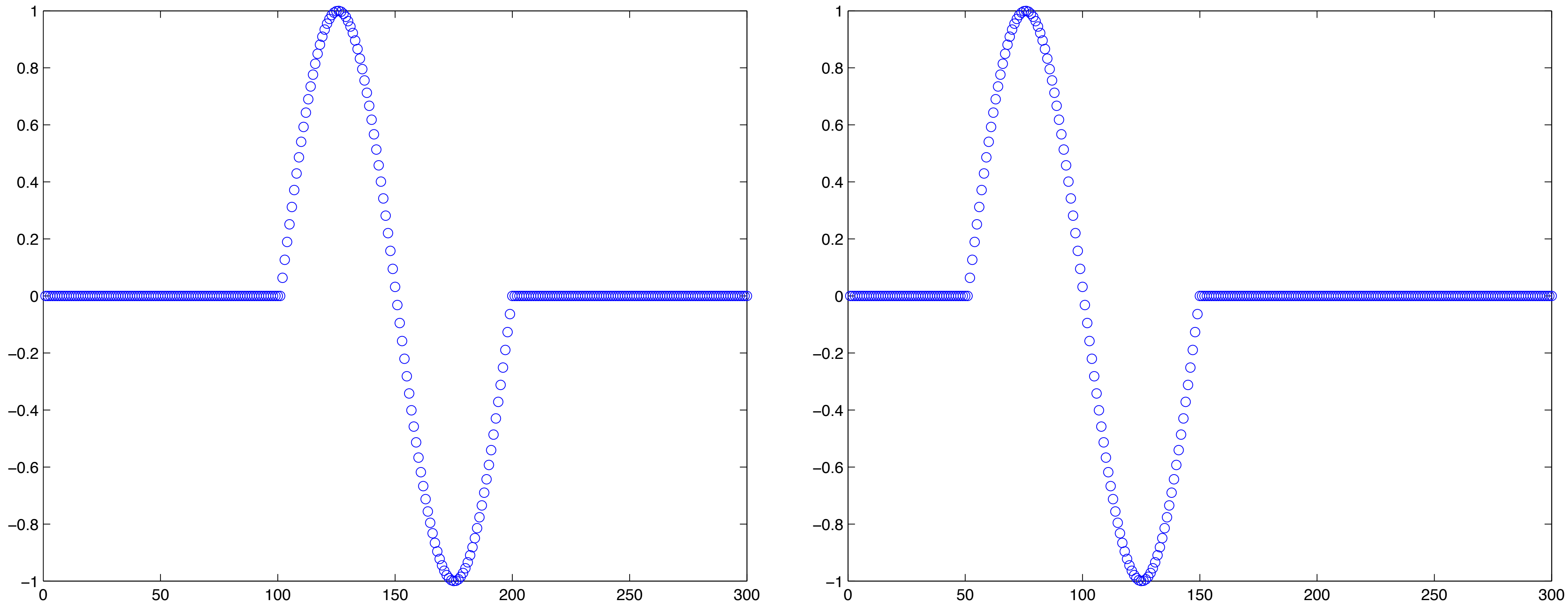


  - Path cannot go back in time



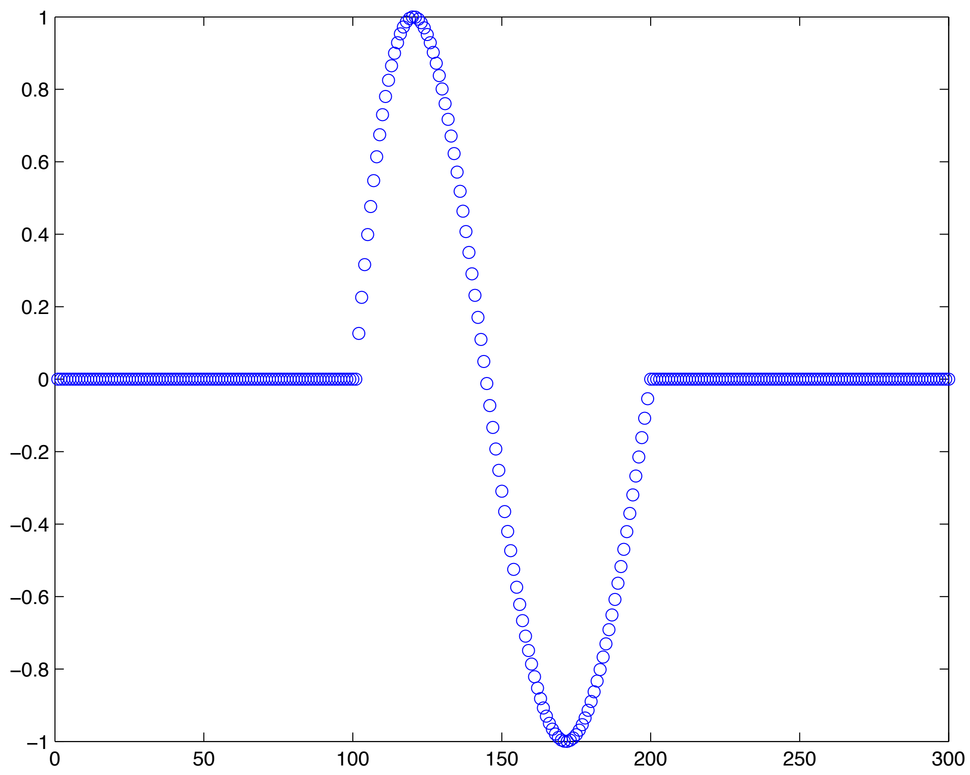
Shift variance

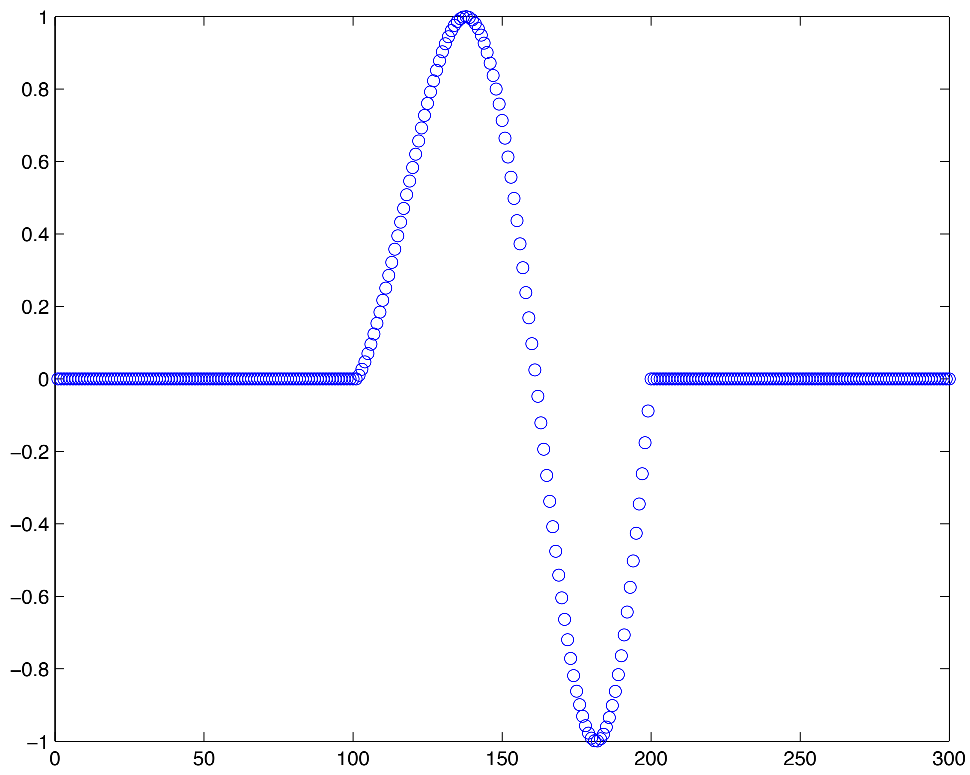
- Time series have shift variance
  - Are these two points close?



Time warp variance

- Slight changes in timing are not relevant
  - Are these two point close?

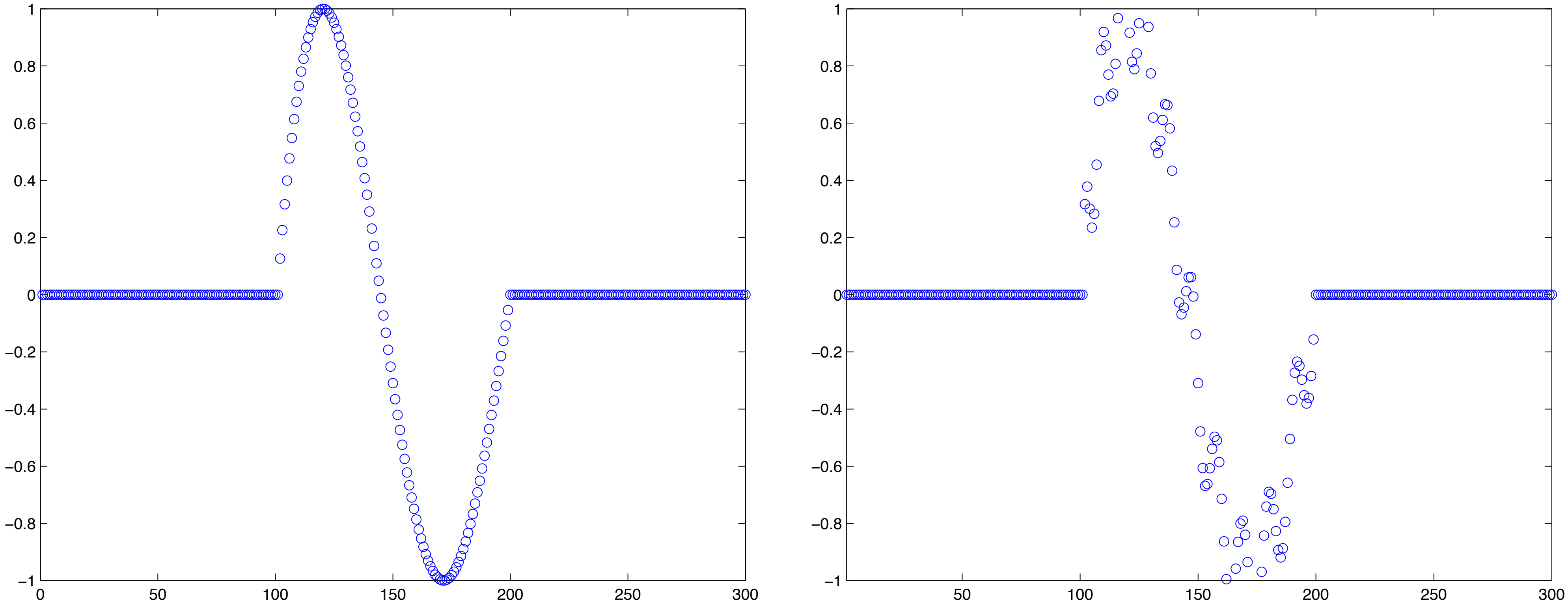




ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

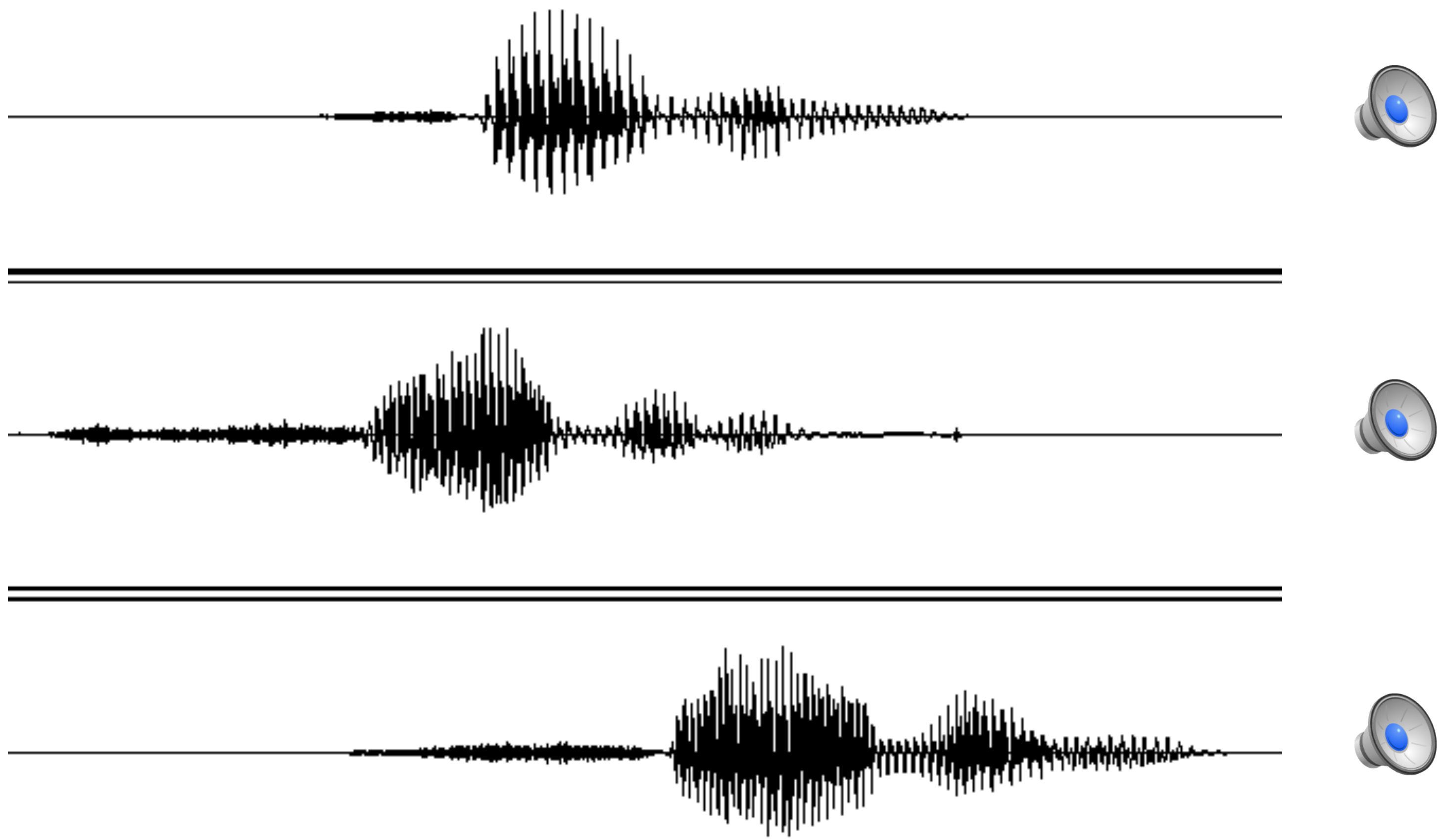
Noise/filtering variance

- Small changes can look serious
  - How about these two points?

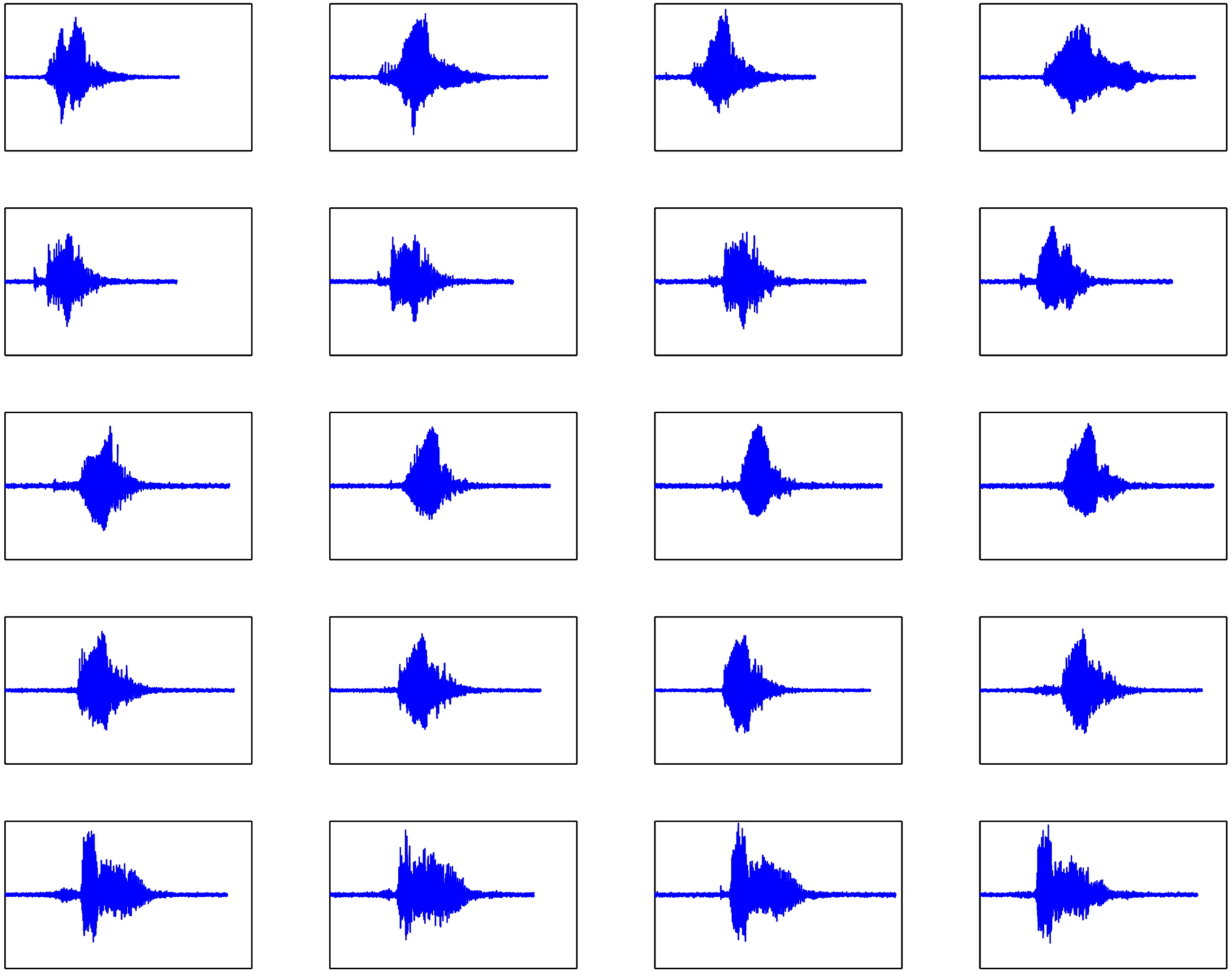


A real-world case

- Spoken digits

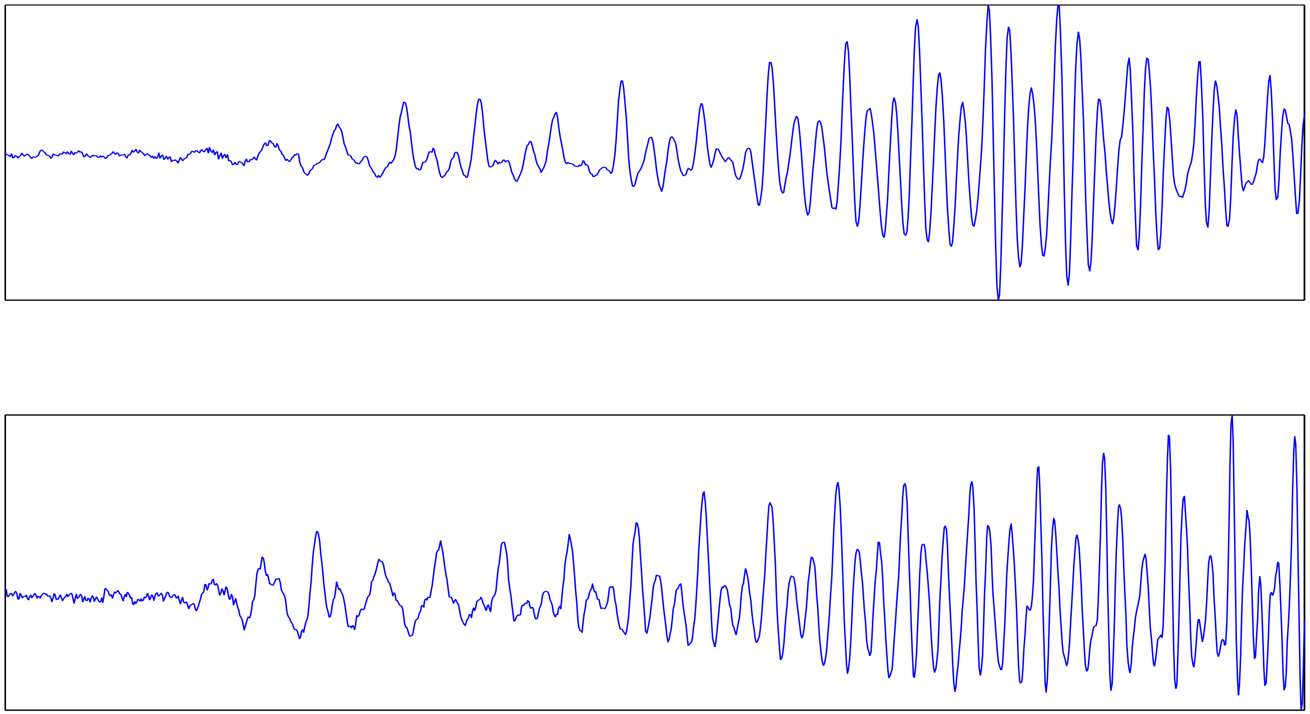


Example data



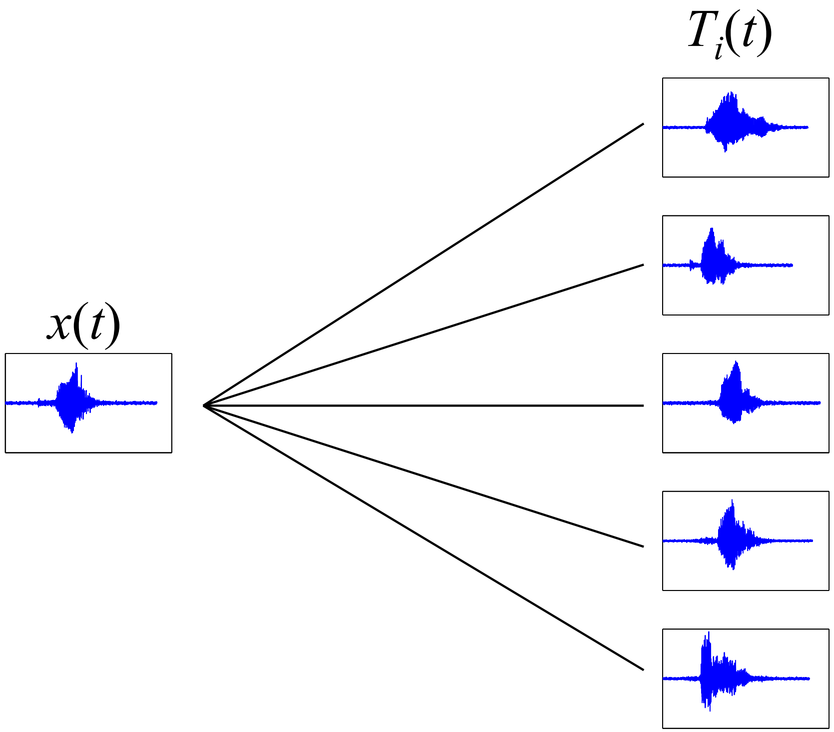
Going from fine to coarse

- Small differences are not important
  - Find features that obscure them



A basic speech recognizer

- Collect template spoken words $T_i(t)$
- Get their DTW distances from input $x(t)$
  - Smallest distance wins



Clustering Time Series

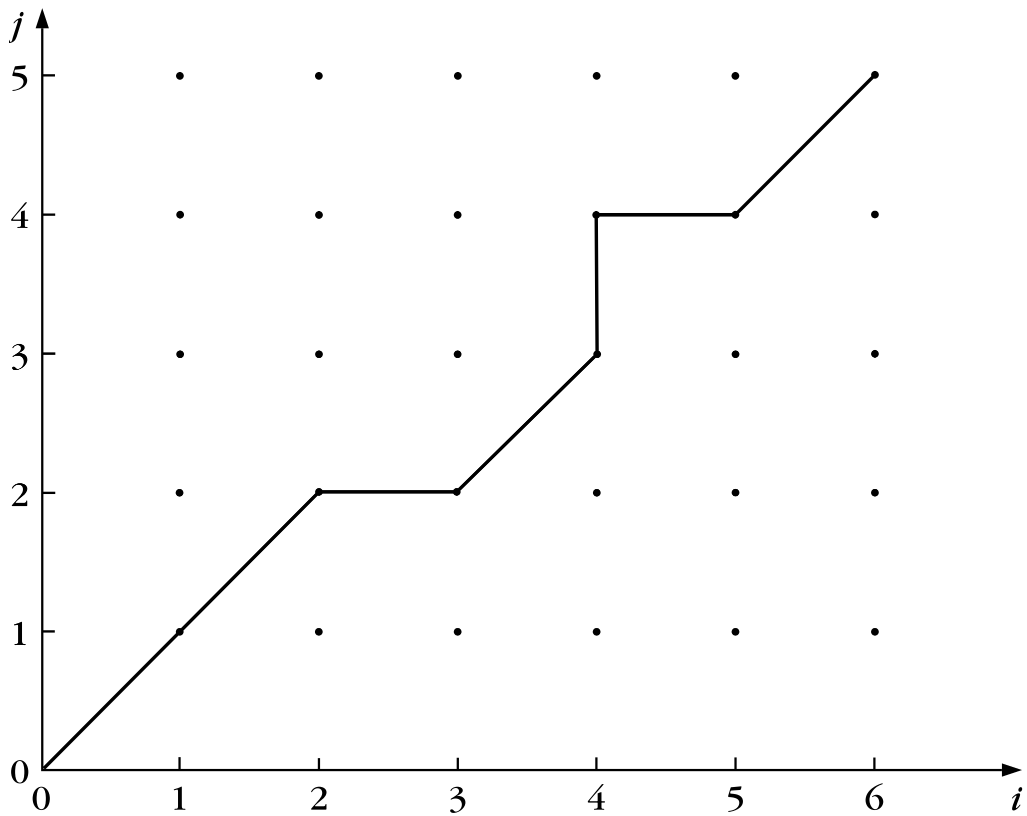
- How do we cluster time series?
  - We can't just use k-means …

- We can use DTW for this

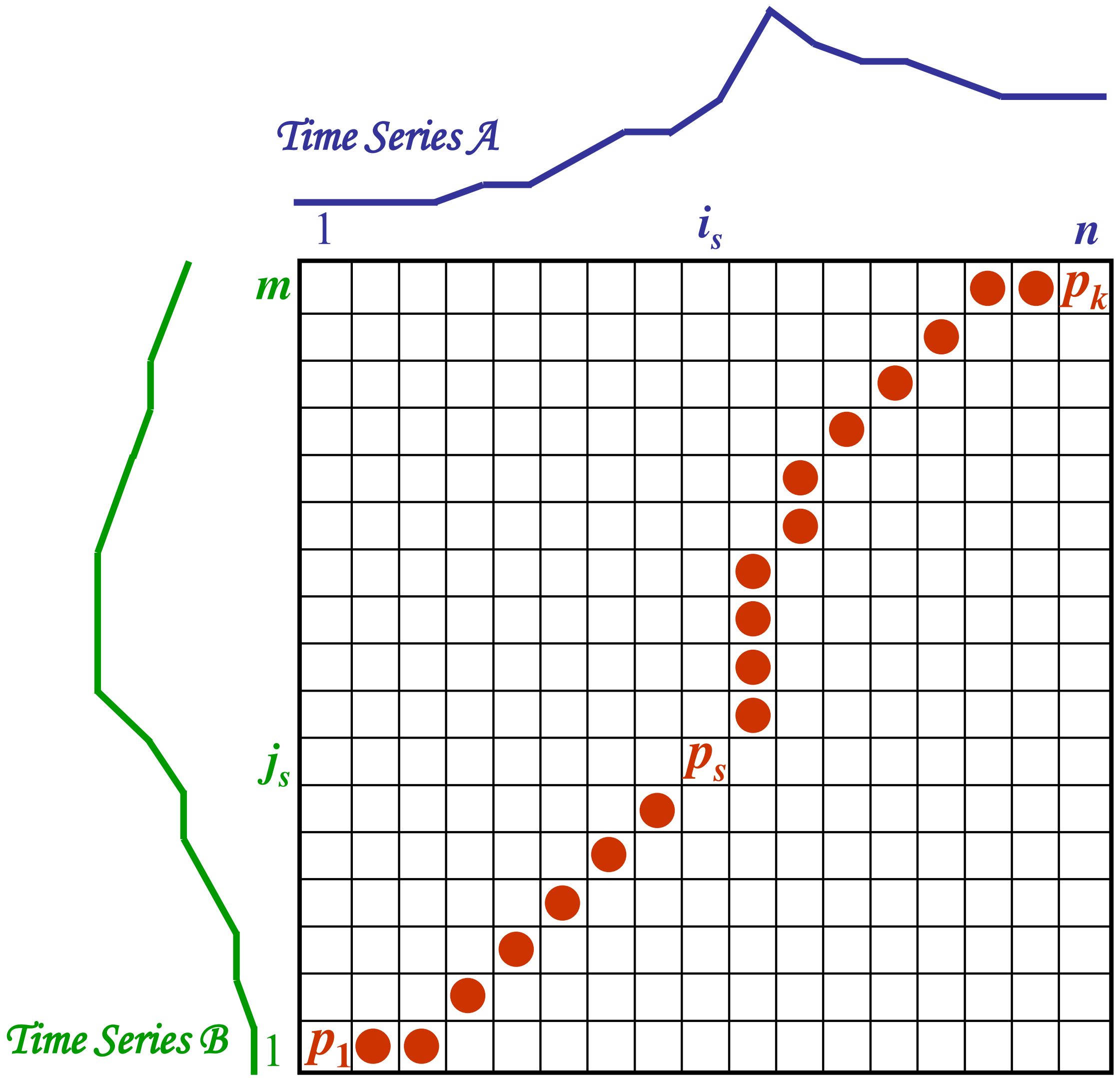
Matching warped series

- Represent the warping with a path

$$\mathbf{r}(i), i = 1, 2, \ldots, 6 \qquad \mathbf{t}(j), j = 1, 2, \ldots, 5$$



Warping Function



To find the *best alignment* between $\mathcal{A}$ and $\mathcal{B}$ one needs to find the path through the grid

$$P = p_1, \ldots, p_s, \ldots, p_k$$

$$p_s = (i_s, j_s)$$

which *minimizes* the total distance between them.

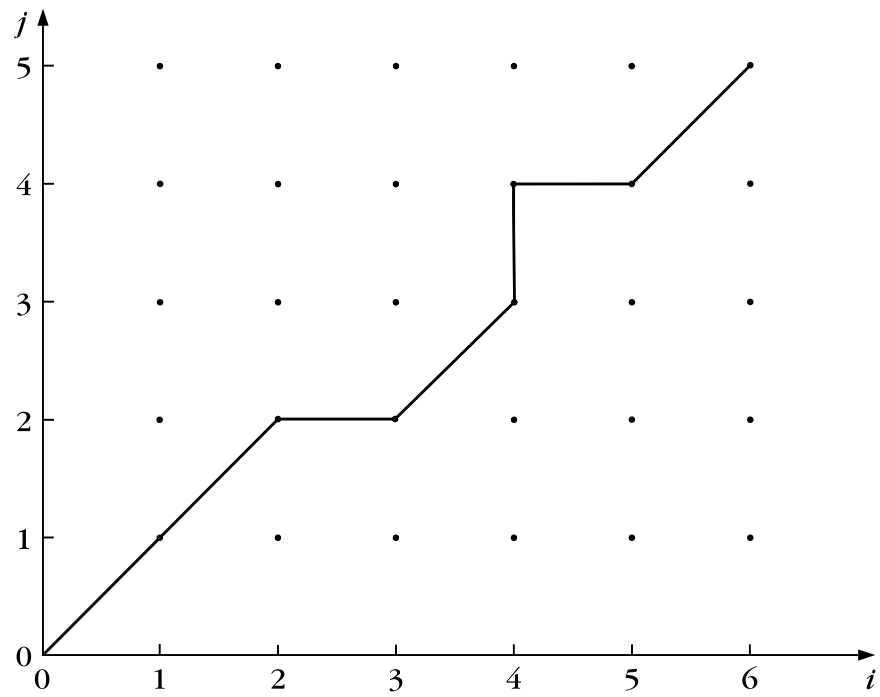
P is called a *warping function*.

Finding the overall "distance"

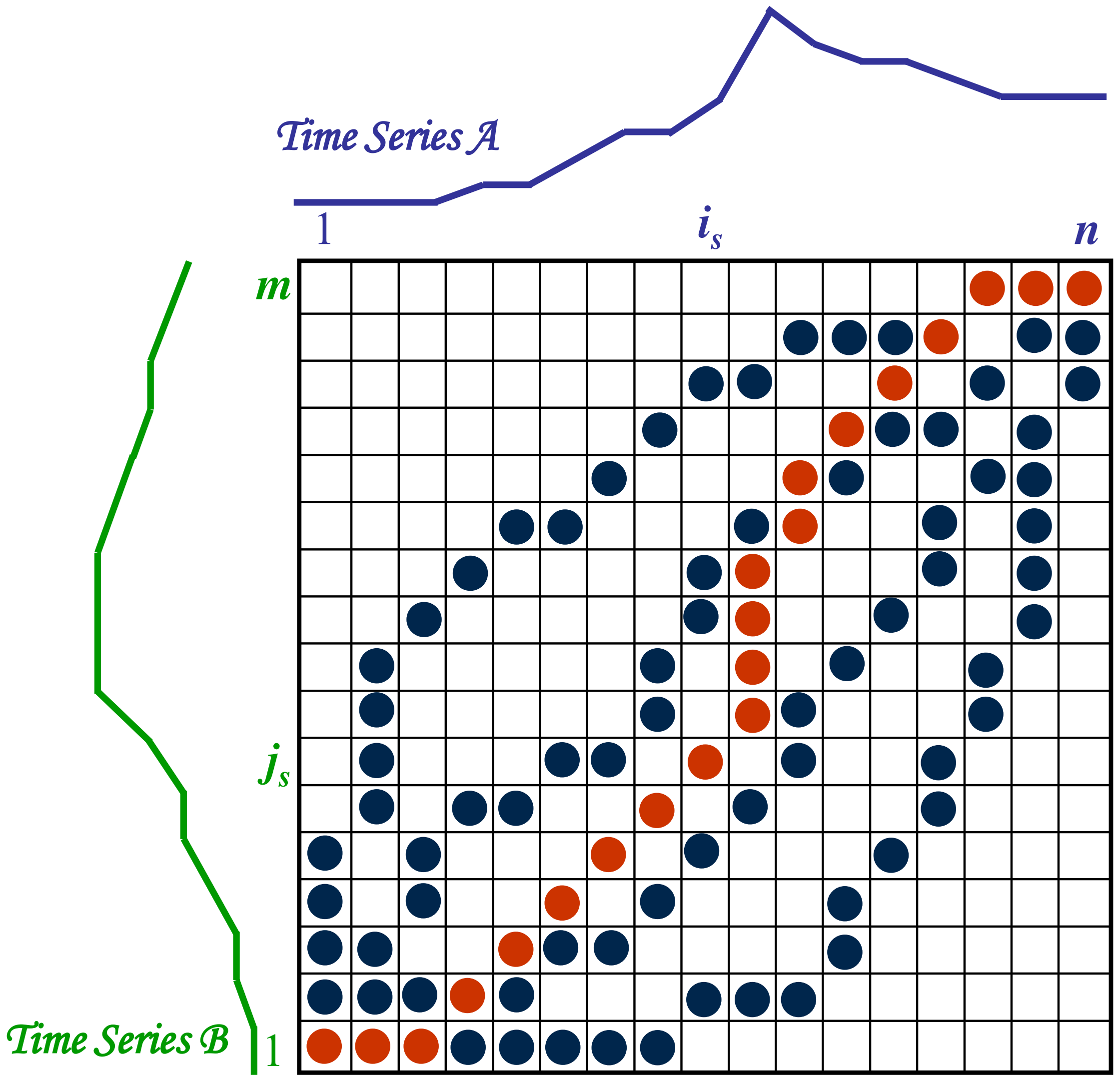
- Each node will have a *cost*
  - e.g., $d(i, j) = \left\| \mathbf{r}(i) - \mathbf{t}(j) \right\|$
- Overall path *cost* is:

$$D = \sum_k d(i_k, j_k)$$

- Optimal D path defines the "distance" between two given sequences



Optimisations to the DTW Algorithm



The number of possible warping paths through the grid is exponentially explosive!

reduction of the search space

Restrictions on the warping function:

- monotonicity
- continuity
- boundary conditions
- warping window
- slope constraint.

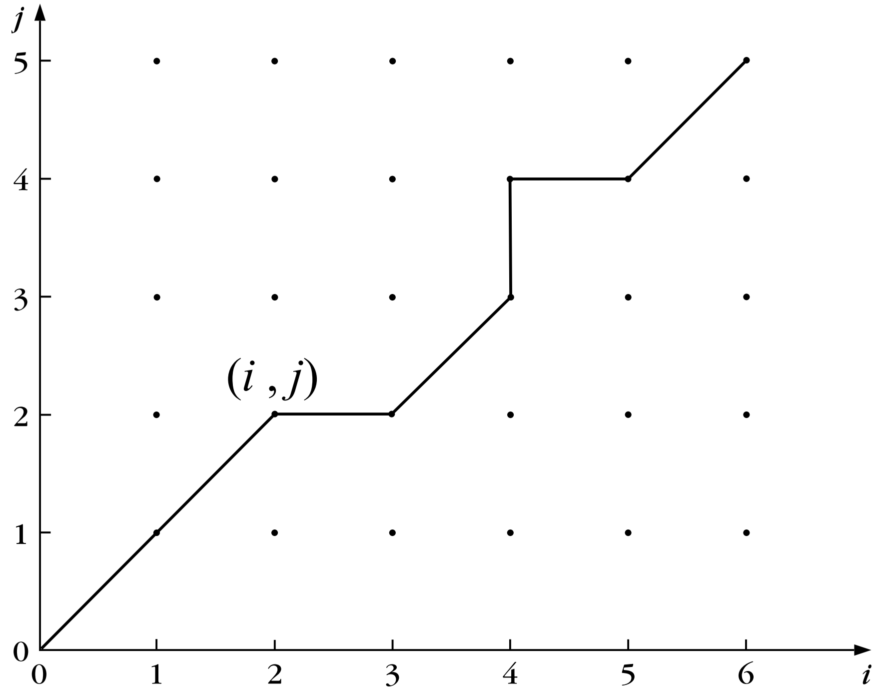
Bellman's optimality principle

- For an optimal path passing through $(i\ ,\ j)$:

$$(i_0, j_0) \overset{opt}{\longrightarrow} (i_f, j_f)$$

- Then:

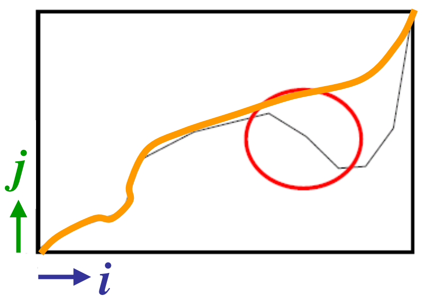
$$(i_0, j_0) \overset{opt}{\longrightarrow} (i_f, j_f) = \left\{ (i_0, j_0) \overset{opt}{\longrightarrow} (i, j), (i, j) \overset{opt}{\longrightarrow} (i_f, j_f) \right\}$$



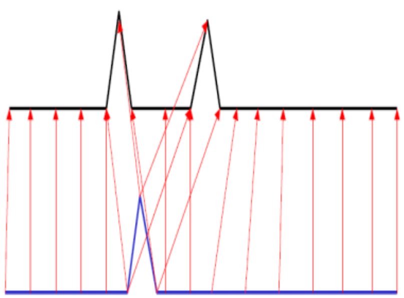
Restrictions on the Warping Function

Monotonicity: $i_{s-1} \leq i_s$ and $j_{s-1} \leq j_s$.

The alignment path does not go back in "*time*" index.

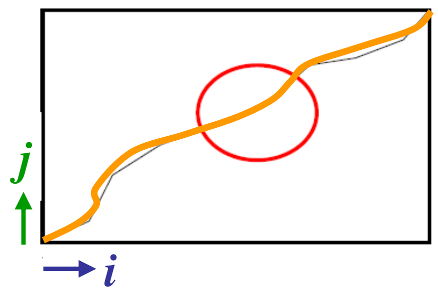


Guarantees that features are not repeated in the alignment.

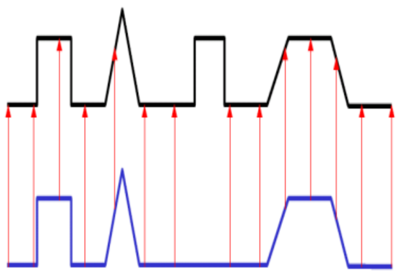


Continuity: $i_s - i_{s-1} \leq 1$ and $j_s - j_{s-1} \leq 1$.

The alignment path does not jump in "*time*" index.



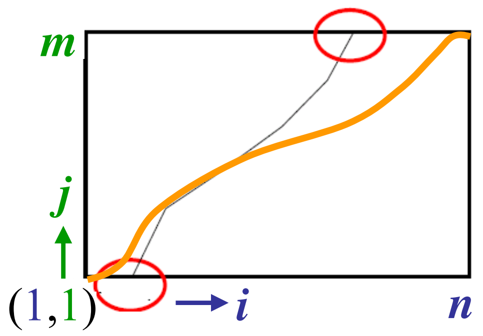
Guarantees that the alignment does not omit important features.



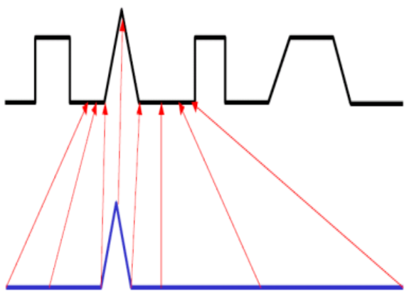
Restrictions on the Warping Function

Boundary Conditions: $i_1 = 1,\ i_k = n$ and $j_1 = 1,\ j_k = m$.

The alignment path starts at the bottom left and ends at the top right.

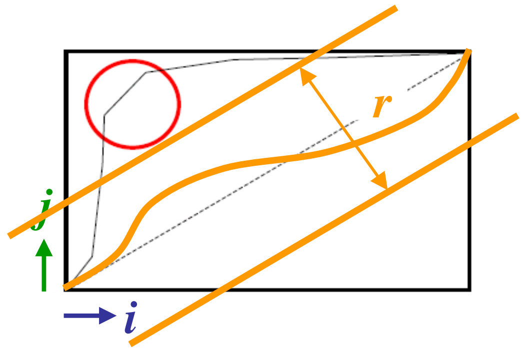


Guarantees that the alignment does not consider partially one of the sequences.

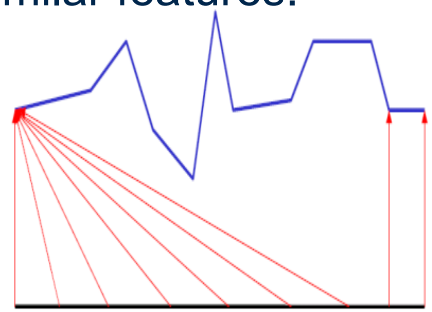


Warping Window: $|i_s - j_s| \leq r$, where $r > 0$ is the window length.

A good alignment path is unlikely to wander too far from the diagonal.



Guarantees that the alignment does not try to skip different features and gets stuck at similar features.



DTW Algorithm: Example

Time Series B ↑ / *Time Series A* →	-0.87 -0.88	-0.84 -0.91	-0.85 -0.84	-0.82 -0.82	-0.23 -0.24	1.95 1.92	1.36 1.41	0.60 0.51	0.0 0.03	-0.29 -0.18
1.94 1.97	0.51	0.51	0.49	0.49	0.35	0.17	0.21	0.33	0.41	**0.49**
0.77 0.74	0.27	0.27	0.26	0.25	0.16	**0.18**	**0.23**	**0.25**	**0.31**	0.68
-0.17 -0.32	0.13	0.13	0.13	0.12	**0.08**	0.26	0.40	0.47	0.49	0.49
-0.58 -0.63	0.08	0.08	0.08	**0.08**	0.10	0.31	0.47	0.57	0.62	0.65
-0.71 -0.68	0.06	0.06	**0.06**	0.07	0.11	0.32	0.50	0.60	0.65	0.68
-0.65 -0.62	0.04	**0.04**	0.06	0.08	0.11	0.32	0.49	0.59	0.64	0.66
-0.60 -0.46	**0.02**	0.05	0.08	0.11	0.13	0.34	0.49	0.58	0.63	0.66

Euclidean distance between vectors