UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Pattern Recognition

Exercise Session 7

Word Spotting with Dynamic Time Warping

Angelika Garz (angelika.garz@unifr.ch)
Marcel Würsch (marcel.wuersch@unifr.ch)

Exercise 4 – Recap

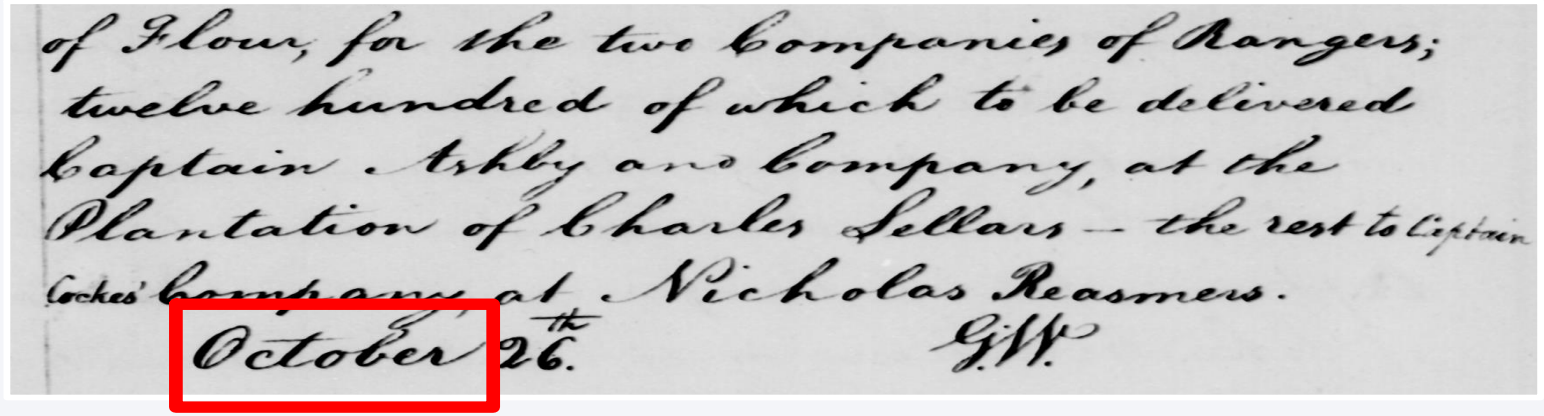
- Previous task
  - Analyze the data
  - Preprocess it
    - Binarize
    - Create word images
    - (Skew correction)
  - Compute some features and apply KNN

Keyword Spotting

Digitizing historical manuscripts for cultural heritage preservation

Textual content: searching and browsing scanned page images

- Widely unsolved for historical handwriting
  - too many writing styles and languages
- Keyword spotting is a “shortcut”: identify individual search terms

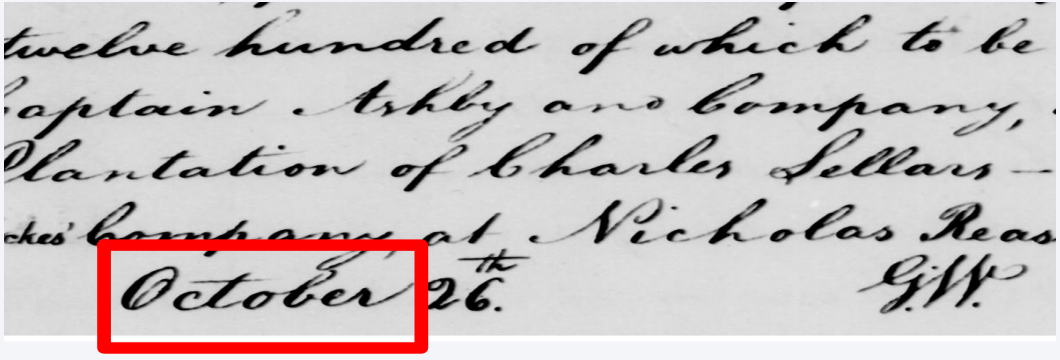


Query-By-Example

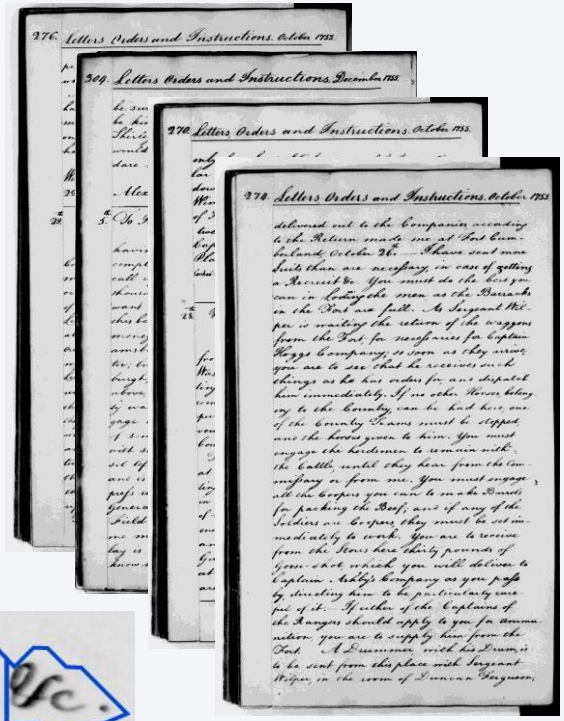
"one-shot learning": provide one example word image

Goal: find similar word images in the manuscript

Usually constrained to a single-writer scenario
(sample from the same manuscript)

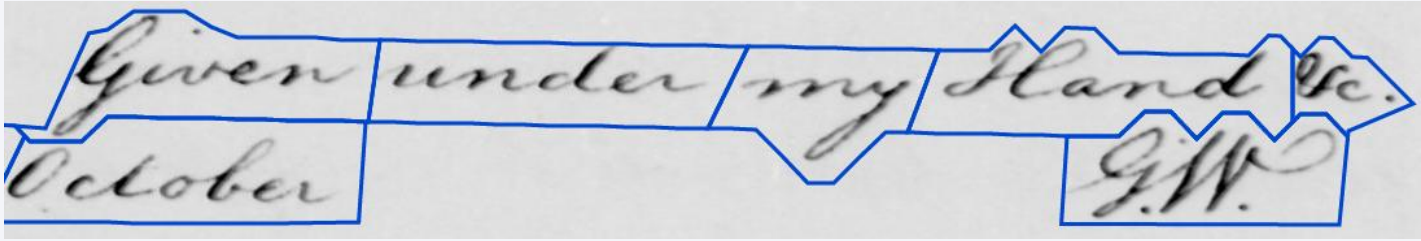


Data Set



WashingtonDB

- Letters of George Washington
- Library of Congress
- 18th century, longhand script

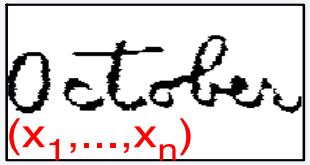


Exemplary Dissimilarity Approaches

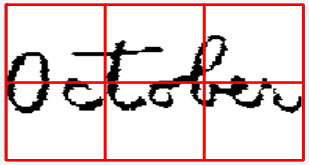
Global: extract global features, compute the Euclidean distance between the feature vectors

Grid-based: extract features for each cell, compute the sum of Euclidean distances over all cells

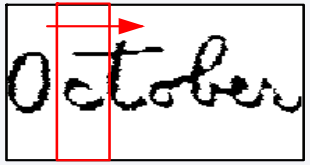
Sliding window-based: extract features for each window, compute the dynamic time warping (DTW) distance between two sequences of feature vectors



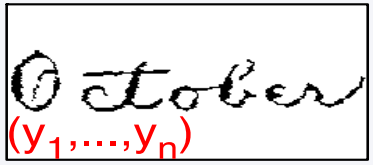
$d(x,y) = \|x-y\|$

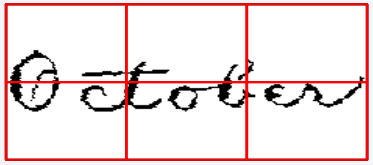


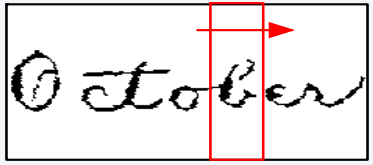
$d(x,y) = \sum \|x_i-y_i\|$



$d(x,y) = DTW(x,y)$







Dynamic Time Warping (DTW)

Dissimilarity between two feature vector sequences

$$Q = q_1, \ldots, q_N ; q_i \in R^n$$

$$C = c_1, \ldots, c_M ; c_i \in R^n$$

Dynamic time warping *aligns two sequences* ($q_i \rightarrow c_j$), along a common time axis usually with Euclidean cost:

$$\phi(q_i \rightarrow c_j) = \left\| q_i - c_j \right\| = \sqrt{\sum_{k=1}^{n} \left(q_{i,k} - c_{j,k} \right)^2}$$

DTW – How To (1)

Non-linear mapping between 2 sequences minimizing the distance between them

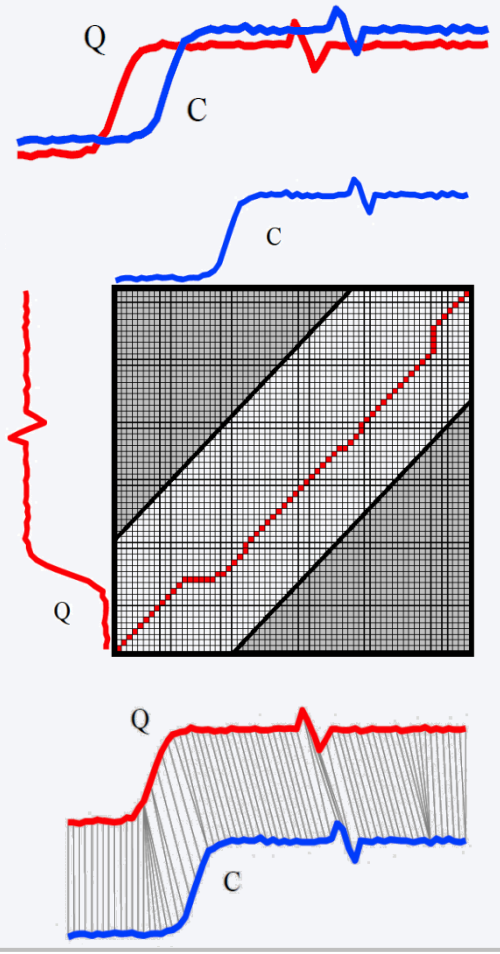
$$Q = q_1, \ldots, q_N ; q_i \in R^n$$

$$C = c_1, \ldots, c_M ; c_i \in R^n$$

N-by-M matrix, where (i^{th}, j^{th}) element alignment between points q_i and c_j

$$d(q_i, c_j) = \sqrt{(q_i - c_j)^2}$$

Find the best match: retrieve a path through the matrix that minimizes the total cumulative distance



DTW – How To (2)

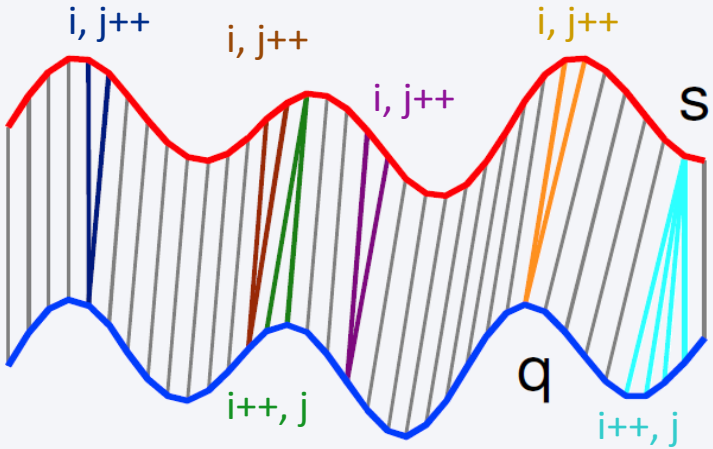
Start from (1,1) and end in (n,m)

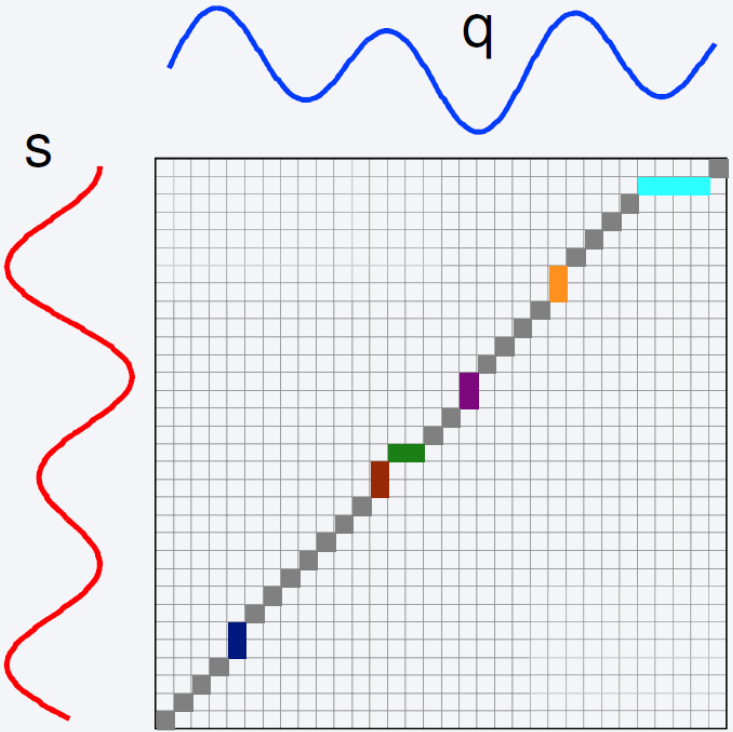
At each step, increase *i*, *j*, or both

(never go back)

Jumping not allowed!

Sum distances in the path



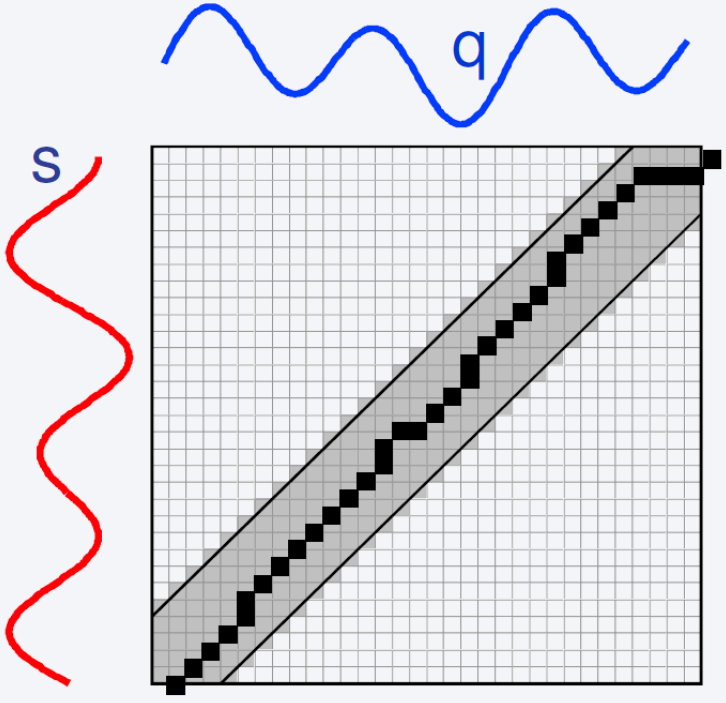


DTW – Computational Efficiency

Sakoe-Chiba Band: Reduce the number of paths to consider

- Excludes abnormal edit paths
- Speeds up the computation
- Sequences of same length

s \ q						
7				43	24	17
3			7	11	8	8
4		6	9	18	8	7
5	10	5	11	18	7	
2	1	2	2	3		
1	1	5	6			
DTW	2	3	2	1	3	4



Features for DTW

Normalize

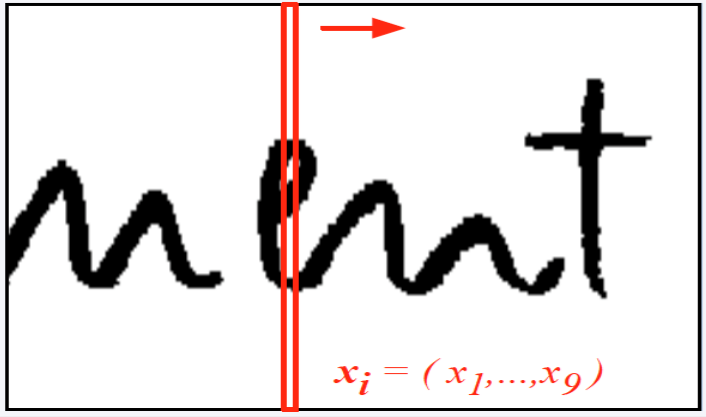
- Image dimensions (scale to same size, e.g. *100 px x 100 px*)
  → same-length sequence

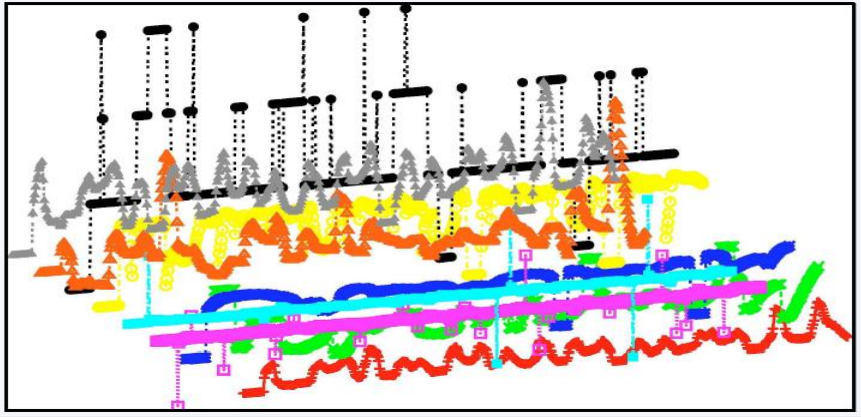
- Feature vectors (e.g. $\frac{x_i - \mu}{\sigma}$)

Features for DTW – Suggestions …

Sliding window (suggestion: width 1 px, offset 1px)

- Lower contour (LC)
- Upper contour (UC)
- # b/w transitions
- Fraction of black px in the window
- Fraction of black px between LC and UC
- Gradient: difference LC_i, UC_i to LC_{i+1}, UC_{i+1}





UNI
FR

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG