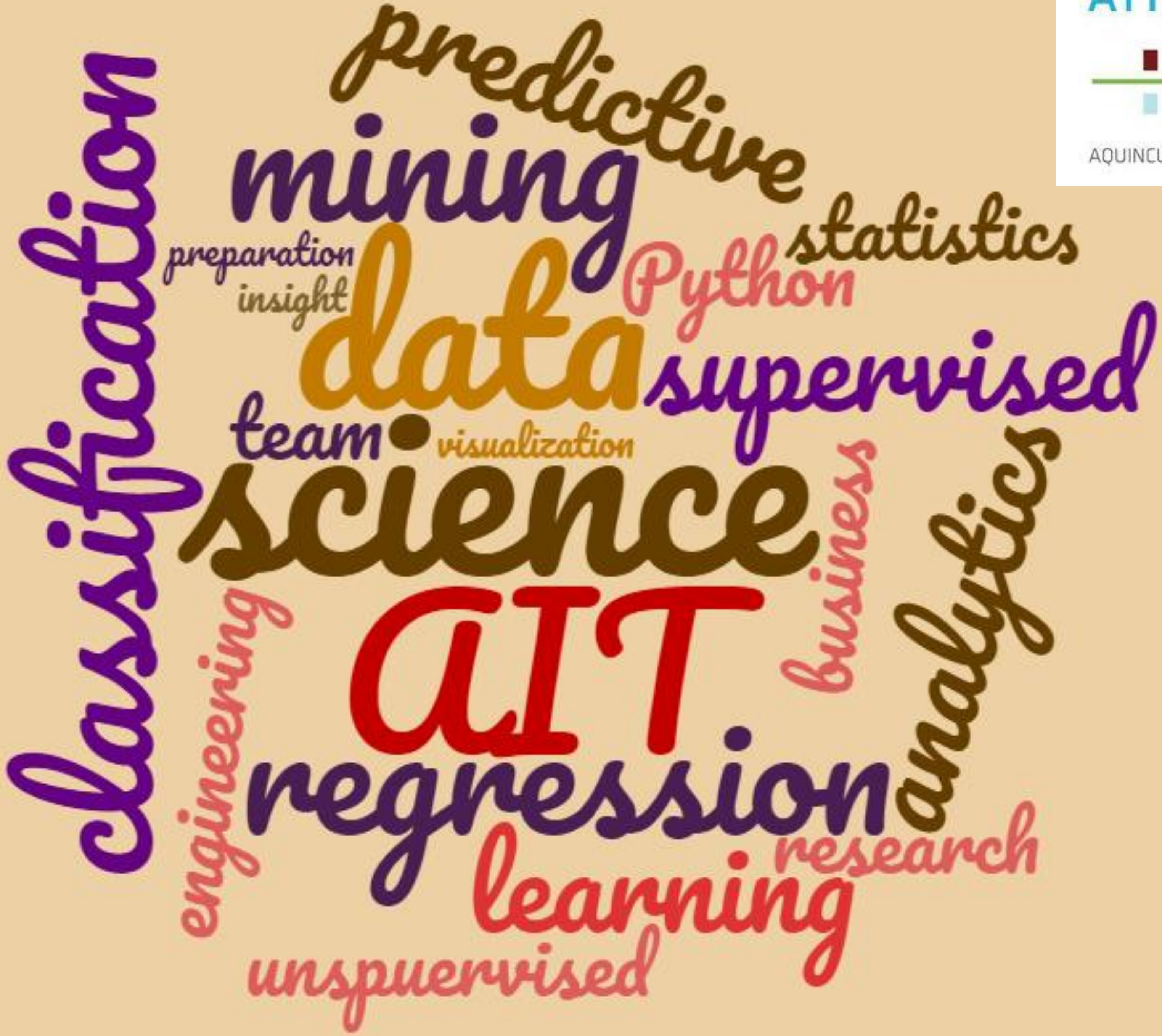
AIT-BUDAPEST

AQUINCUM INSTITUTE OF TECHNOLOGY

Data Science

February 14,
Similarity measures

Roland Molontay



Similarity, dissimilarity

- For many algorithms, it is necessary to quantify the (dis)similarity of two records (rows)
  - E.g. the goal of clustering is to group similar objects together
- There are several (dis)similarity measures
- The used metric depends on the feature vector, what type of attributes (continuous, categorical, etc.) the row consists of

Similarity vs. dissimilarity

Similarity

The higher it is, the more similar the records are

Usually symmetric

Typically takes values from [0 , 1] (or [0, ∞])

Dissimilarity

The higher it is, the more dissimilar the records are

Usually symmetric

Usually the value is 0 if the records are identical

Distance metric

- A special case of dissimilarity measures is distance (or metric):

$d : X \times X \rightarrow [0, \infty)$

1.	$d(x, y) \geq 0$	(non-negativity)
2.	$d(x, y) = 0 \Leftrightarrow x = y$	(identity of indiscernibles)
3.	$d(x, y) = d(y, x)$	(symmetry)
4.	$d(x, z) \leq d(x, y) + d(y, z)$	(triangular inequality or subadditivity)

$x, y, z \in X$.

- We don’t necessarily require dissimilarity measures to satisfy the above axioms of distance metric but in many cases we work with distance functions (i.e. measures that satisfy the above conditions)

Categorical attributes

- Similarity: 1 if they are equal, 0 otherwise
  - Dissimilarity: just the opposite (0 if they are equal, 1 otherwise)
- Sometimes there are categories that are „closer” to each other than others, so the similarity function can be non-binary as well
  - We can use a scoring matrix that codes the similarities between the categories
  - E.g. professions

Values from a given interval

- Possible values from an intervsl [*1, n*]
- The following dissimilarity notion is natural
    - Dissimilarity of *x* and *y* : $\frac{|x-y|}{n-1}$
    - Dissimilarity values between 0 and 1 (0, if they are equal)
- Similarity measure can be defined similarly
    - Similarity *of x* and *y:* $1 - \frac{|x-y|}{n-1}$
    - Similarity values between 0 and 1 (1, if they are equal)

Values from a non-finite interval

- The following dissimilarity notion
  - Dissimilarity of *x* and *y*: $|x - y|$
  - Dissimilarity values between 0 and ∞ (0, if they are equal)
- We can derive a similarity notion from the above dissimilarity:
  - We can take its negative (a value between - ∞ and 0)
  - We can take its rational function (a value between 0 and 1)

$$\frac{1}{1 + |x - y|}$$

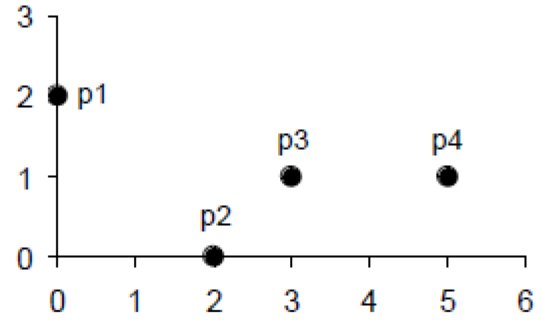
Comparing records having the same attribute types

- For each attribute (column) we take the (dis)similarity
- Somehow we aggregate the results
  - Summing up for all columns
  - Summing up for all columns than dividing by the number of columns
  - We can use weighted sum
  - It is practical to rescale the columns before calculating the aggregated (dis)similarity in order to have attribute values on a comparable scale

Euclidean distance

- The most frequently uses distance if the records can be interpreted as points in the *n*-dimensional space
- $p = (p_1, \ldots, p_n)$ and $q = (q_1, \ldots, q_n)$ are two points in the space, their Euclidean distance:

$$d(p,q) = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

- It is practical to rescale the range (max-min normalization, standardization)

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Minkowski distance

- Minkowski-distance (L_p distance): generalization of Euclidean distance
- $p = (p_1, \ldots, p_n)$ and $q = (q_1, \ldots, q_n)$ are two points, their Minkowski distance:

$$d(p,q) = \sqrt[r]{\sum_{k=1}^{n} |p_k - q_k|^r}$$

- If *r ≥ 1* it is a distance metric
- The greater the *r* parameter is, the more important feature scaling is
- Special cases:
  - r = 1, Manhattan distance
  - r = 2, Euclidean distance
  - r = ∞, Chebisev distance

It can be weighted as well:

$$\sqrt[r]{\sum_{k=1}^{n} w_k \cdot |p_k - q_k|^r}$$

Special cases of Minkowski distance

- Manhattan distance (L_1 distance, taxicab distance):
  - The Manhattan distance of points (1,2) and (7,0) is 8, they are 8 „blocks” from each other

$$\sum_{k=1}^{n} |p_k - q_k|$$

- Euclidean distance (L_2)
- Chebisev distance (L_{max}, L_∞)
  - Two equivalent definitions:

$$\lim_{r\to\infty} \sqrt[r]{\sum_{k=1}^{n} |p_k - q_k|^r} \qquad \max_{k\in\{1,2,\dots,n\}} |p_k - q_k|$$

Problem

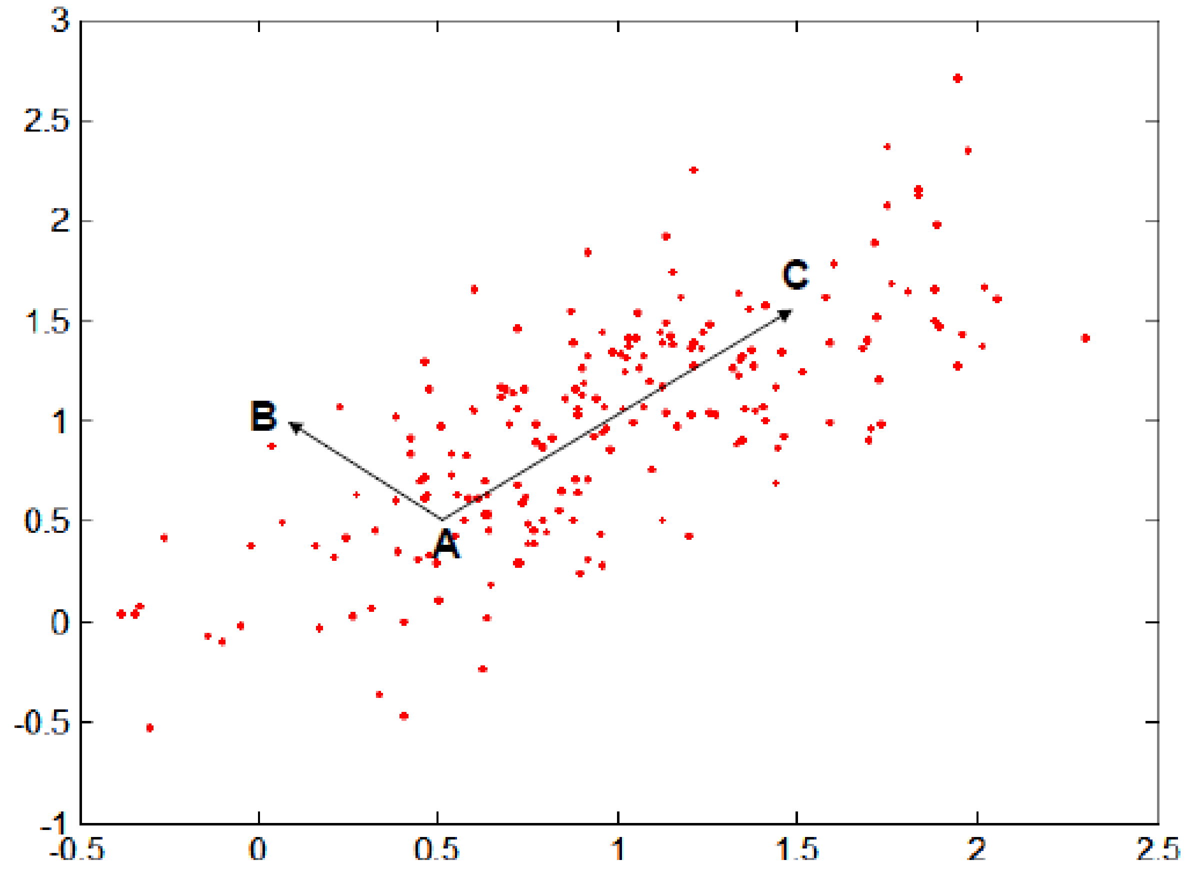
Prove the following statements:

1. $L_1(x, y) = \sum_{i=1}^{d} |x_i - y_i|$ is a distance metric
2. $L_2^2(x, y) = \sum_{i=1}^{d} (x_i - y_i)^2$ is **not** a metric.

Mahalanobis distance

- Minkowski distance does not take into consideration that attributes are not necessarily independent from each other
  - In the extreme case, we can have two identical columns, then it counts double in the distance
  - A possible solution is that we filter out these attributes (reducing the number attributes)
  - Or we can use a distance that compensate for the bias coming from the correlatedness of the attributes
- Mahal(p, q) = $(p - q)\Sigma^{-1}(p - q)^T$
  - Where ∑ is the sample covariance matrix, that is: $\Sigma_{j,k} = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ij} - \overline{X}_j)(X_{ik} - \overline{X}_k)$

Mahalanobis distance, an example



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

Check!

Mahal(A, B) =5

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

Mahal(A,C) = 4

Problem

Exercise. *Covariance matrix:*

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

Calculate the following quantities!

1. $\mathrm{Corr}(X, Y)$
2. $\mathrm{Mahal}(A, B)$, *where* $A = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 1 \end{bmatrix}$

Problem

Let two feature vectors contain the following attributes:

- person's height (between 5 and 6 feet)
- person's weight (between 90 and 260 lbs)
- person's annual income (between 10,000 and 1 Million dollars)

How would you calculate the distance between the two vectors? What kind of transformations would you apply, and which distance would you chose?

Cosine similarity

- Compares the direction of vectors (the magnitude does not play a role here)
  - The similarity of two vectors in the same direction is 1
  - The similarity of two orthogonal (perpendicular) vector is 0
  - The similarity of two vectors in opposite direction is -1

$$cos(p, q) = \frac{p \cdot q}{\|p\| \cdot \|q\|}$$

- It is mostly used for high dimensional positive vectors
  - E.g. for document-term matrices with frequencies
- We can form dissimilarity as well: 1- cos(*p*, *q*)
  - It is not a distant metric (triangular inequality is not satisfied)

Cosine similarity - example

d_1 = 3 2 0 5 0 0 0 2 0 0

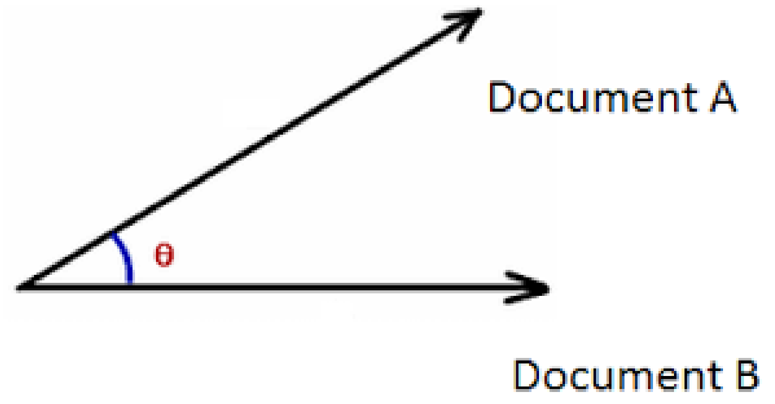
d_2 = 1 0 0 0 0 0 0 1 0 2

$d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$

$||d_1|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

$||d_2|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.245$

$\cos(d_1, d_2) = .3150$



Problem

Consider the following three documents:

- d_1: "ant bee"
- d_2: "dog bee hog ant"
- d_3: "cat gnu dog eel fox"

These documents can be represented by 8-dimensional vectors in a so-called *document-term matrix*, which describes the frequency of terms that occur in a collection of documents:

	ant	bee	cat	dog	eel	fox	gnu	hog
d_1	1	1	0	0	0	0	0	0
d_2	1	1	0	1	0	0	0	1
d_3	0	0	1	1	1	1	1	0

1. Calculate simple matching coefficient (SMC) and Jaccard-coefficient of d_1 and d_2.
2. Determine distances derived from these coefficients. Which is the better coefficient to handle the problem? Why?

Note.: Of course, this approach cannot distinguish "John is quicker than Mary" and "Mary is quicker than John" documents.

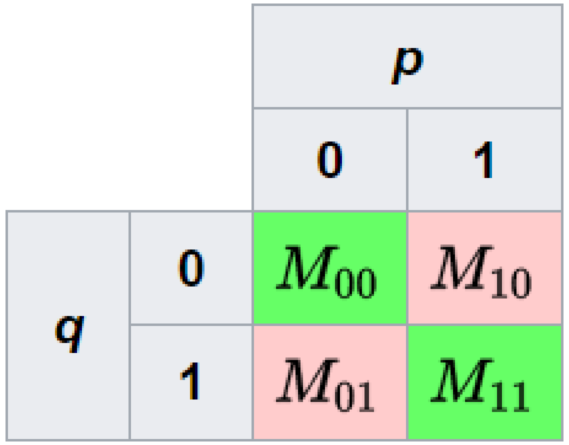
Similarity of binary vectors

- Binary data are frequently correspond to sparse data matrices (almost all entries are 0s and there are few 1s)
  - E.g. document term matrix, transaction matrix
- The previously studied (dis)similarities are not informative in this case, every record looks similar
  - Special (dis)similarity notions are needed
- Let *p* and *q* be binary vectors of length *n* (the entries are 0 or 1)

Simple Matching Coefficient (SMC)

- SMC: simple matching coefficient

$$\text{SMC} = \frac{\text{number of matching attributes}}{\text{number of attributes}} = \frac{M_{00} + M_{11}}{M_{00} + M_{01} + M_{10} + M_{11}}$$



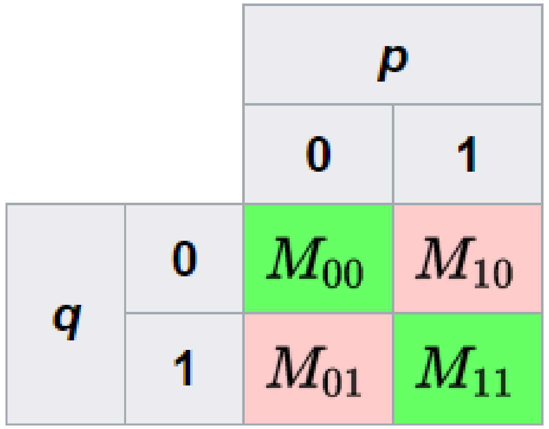
		p	
		0	1
q	0	M_{00}	M_{10}
	1	M_{01}	M_{11}

- SMD (simple matching distance) = 1 – SMC
- Practical to use if the „information content" of 0 and 1 are equivalent (symmetric), i.e. the data matrix is not sparse

Jaccard index

- Jaccard index (similarity coefficient)

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$



		p	
		0	1
q	0	M_{00}	M_{10}
	1	M_{01}	M_{11}

- The common 0s do not play a role
- It is practical for sparse data matrix
- Jaccard distance = 1- J

Example: SMC vs. Jaccard

p = 1 0 0 0 0 0 0 0 0 0

q = 0 0 0 0 0 0 1 0 0 1

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)

$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)

$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)

$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

SMC = $(M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00})$ = (0+7) / (2+1+0+7) = 0.7

J = $(M_{11}) / (M_{01} + M_{10} + M_{11})$ = 0 / (2 + 1 + 0) = 0

Problem

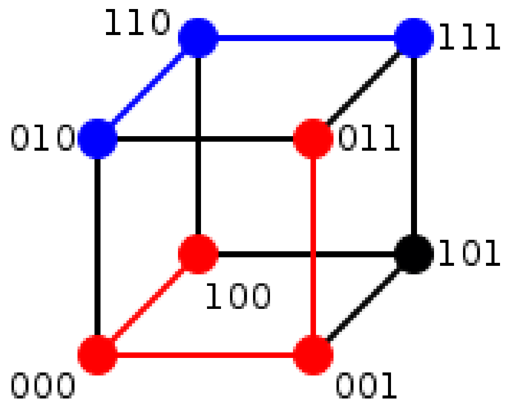
The following table's rows correspond to customers (A, B, C), and the columns correspond to products $(a, b, \ldots, h)$. The table contains 1 if a given costumer bought the given item, 0 otherwise. Determine the Jaccard similarity and the Cosine similarity of A and B.

	a	b	c	d	e	f	g	h
A	1	1	0	1	1	0	1	1
B	0	1	1	1	1	1	1	0
C	1	0	1	1	0	1	1	1

Hamming distance

- H = $M_{01} + M_{10}$
  - It is equal to the number of ones in (p XOR q)
- If *p* and *q* are considered to be points in R^n, then *p* and *q* are the vertices of an *n*-dimensional hypercube
- The Hamming distance corresponds to the Manhattan distance of the vertices
  - The Hamming distance of 010 and 111 is 2
  - The Hamming distance of 100 and 011 is 3

		p	
		0	**1**
q	**0**	M_{00}	M_{10}
	1	M_{01}	M_{11}



Comparing records having different attribute types

- The previously studied methods are applicable if the records (vectors) consist of attributes of the same type in every coordinate
- Sometimes it is not the case
  - Transfer all the features to same type (e.g. to numeric)
  - OR group together coordinates with the same type (categorical, numerical)
  - Calculate the (dis)similarities in each group
    - Do not mix similarities with dissimilarities
  - Define the resultant (dis)similarity by aggregating (with proper weights) the values in each group

Problem

Let $tf_{i,j}$ denote the entry in the i-th row and j-th column of the document-term matrix from the previous task. For instance, $tf_{1,1} = 1$, where row $= d_1$ and column $=$ ant. Consider the following *tf–idf* tranformation.

- Let m be the number of documents.
- Let df_j denote the *document frequency*, i.e. the number of non-zero elements in the j-th column (number of documents containing the j-th word). E.g. $df_1 = 2$.
- Let idf_j denote the *inverse document frequency*, which is defined as follows:

$$idf_j = \log\left(\frac{m}{df_j}\right)$$

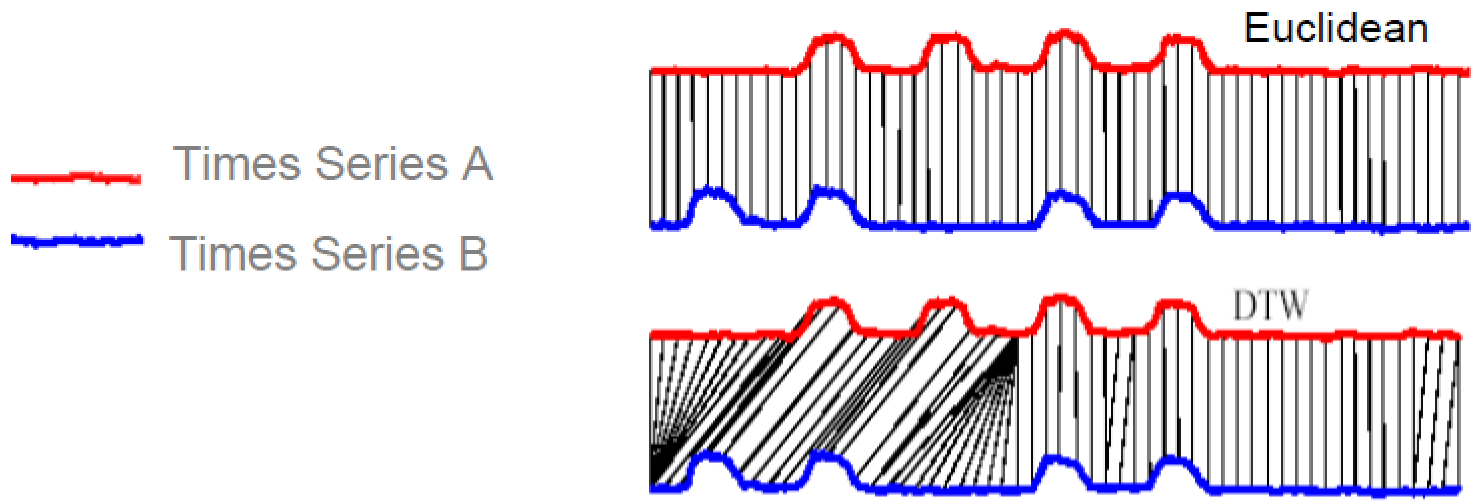
With these notations the *tf–idf* tranformation is defined as follows:

$$tf\text{–}idf_{i,j} = tf_{i,j} \cdot idf_j = tf_{i,j} \cdot \log\left(\frac{m}{df_j}\right)$$

The tf–idf abbreviation stands for term frequency – inverse document frequency. What is the impact of this transformation? In case of a concrete, real document-term matrix, what could be the purpose of this transformation?

Distance of time series

- (In this course there will be little emphasis on time series.)
- First approach: we compare the the time series index wise (e.g. using Minkowski distance where the coordinates correspond to indices)
    - A problem arises if there is any discrepancy in the alignment of the signals
        - E.g the signal is stretched or compressed compared to the other
        - How do you decide which points to compare with each other?
- A more sophisticated method: dynamic time warping (DTW)



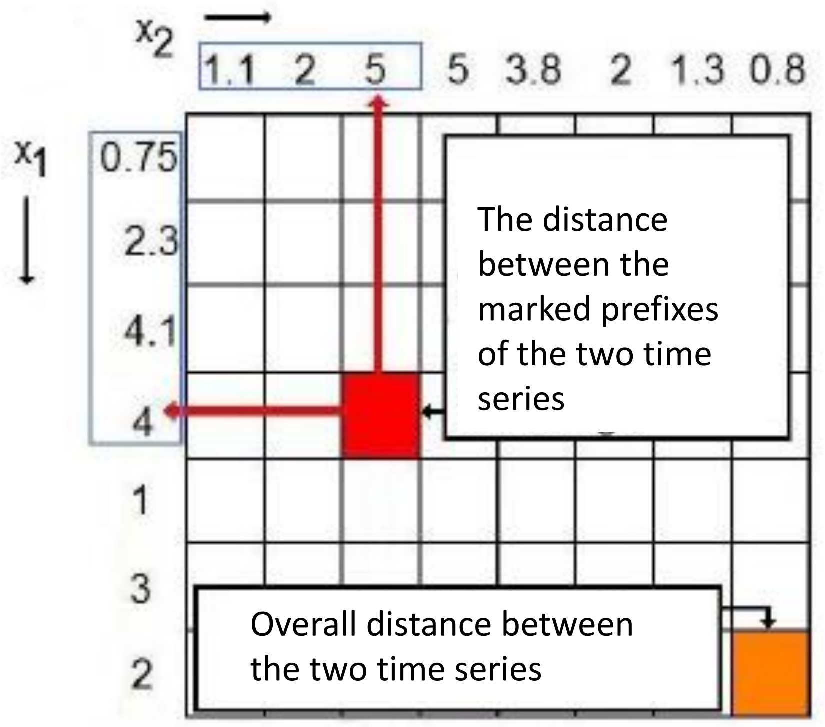
Dynamic Time Warping (DTW)

- In text mining: Levensthein distance, in bioinformatics Smith-Watermann distance
- It is insensitive to local compression and stretches
- An edit distance that calculates the „cost” to transform a time series to another one
- First a distance measure is needed, comparing the corresponding elements of the two sequences, in the simplest form:

$$c_{tr}^{DTW}(x_1[i], x_2[j]) = |x_1[i] - x_2[j]|$$

DTW II.

- We have to define the cost of insertion/deletion: c_{el}^{DTW}.
  - In the simplest case it is 0
- To calculate the distance, we fill the entries of a matrix (using dynamic programming)
  - We write one of the time series above the first row of the matrix and write the second time series next to the first column
  - The values in each entry of the matrix correspond to DTW distance of the corresponding prefixes of the time series



x2
1.1 2 5 5 3.8 2 1.3 0.8
x1
0.75
2.3
4.1
4
1
3
2
The distance between the marked prefixes of the two time series
Overall distance between the two time series

DTW III.

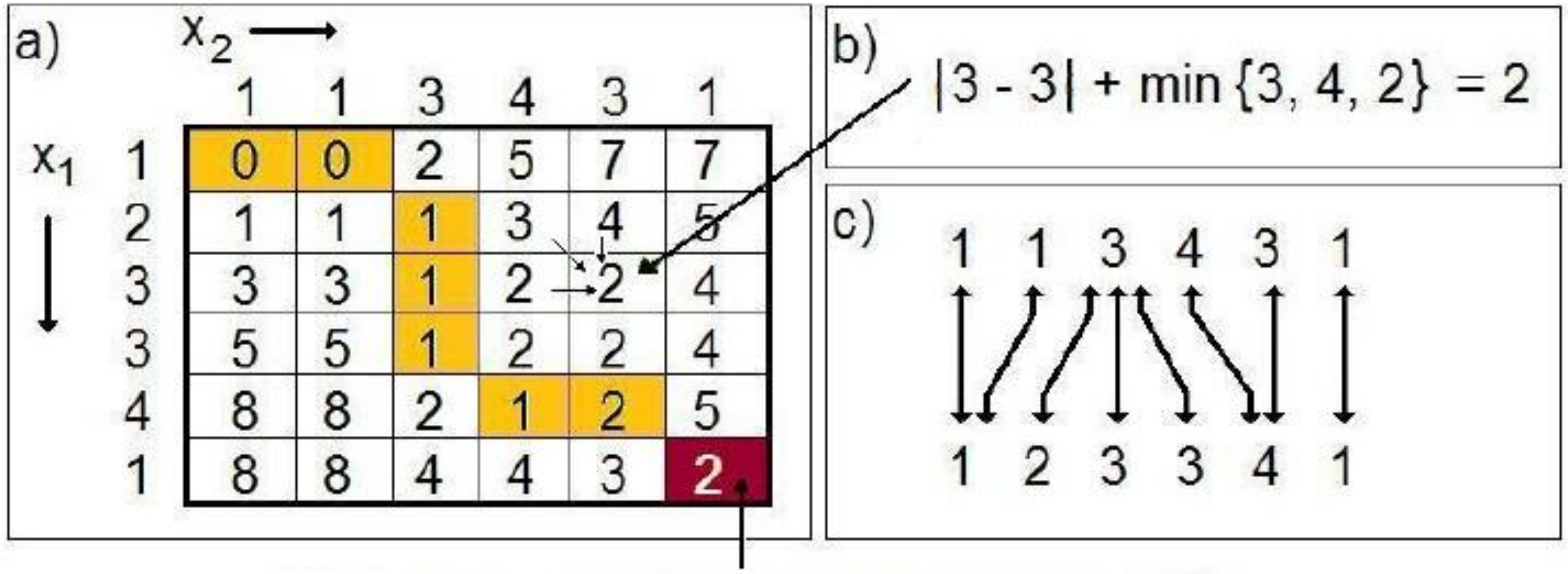
- Let $d_0^{DTW}(i,j)$ denote the value in the *i*th row and *j*th column of the matrix
- The entry in the upper left corner of the matrix: $d_0^{DTW}(0,0) = c_{tr}^{DTW}(x_1[0], x_2[0])$
- Calculating other elements using:

$$d_0^{DTW}(i,j) = c_{tr}^{DTW}(x_1[i], x_2[j]) + \min \left\{ \begin{array}{l} d_0^{DTW}(i, j-1) + c_{el}^{DTW} \\ d_0^{DTW}(i-1, j) + c_{el}^{DTW} \\ d_0^{DTW}(i-1, j-1) \end{array} \right\}$$

- The terms in the minimum expression respectively corresponds to
  - deletion (compression),
  - insertion (stretch),
  - match

DTW - example

- Warping path: which optimally deforms one of the two input series onto the other
- It works for time series of differing length as well
- Running time: *O(nm) /n* and m: the length of the time series/
  - It can be faster if we only calculate elements near the main diagonal

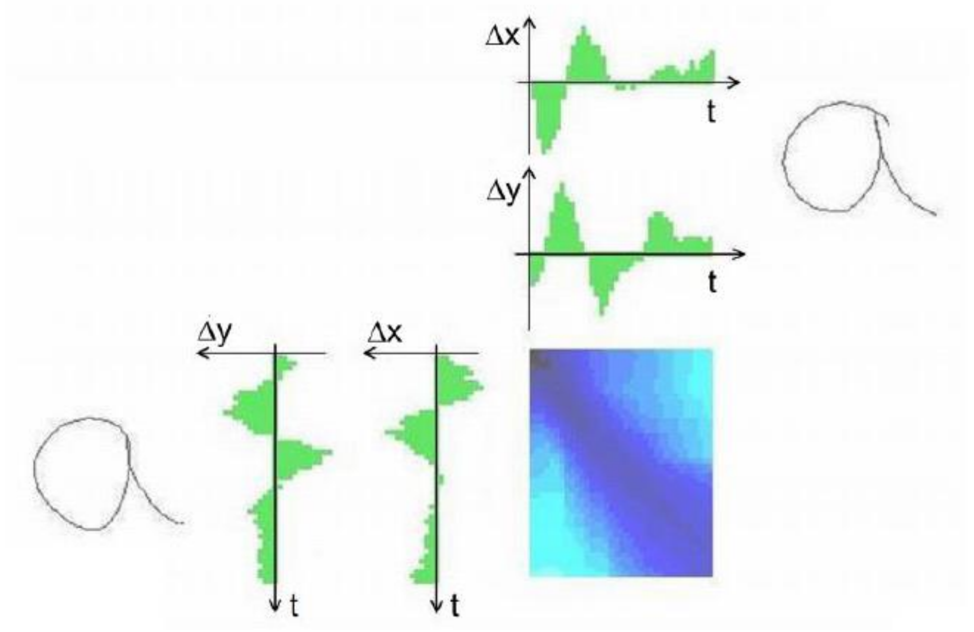


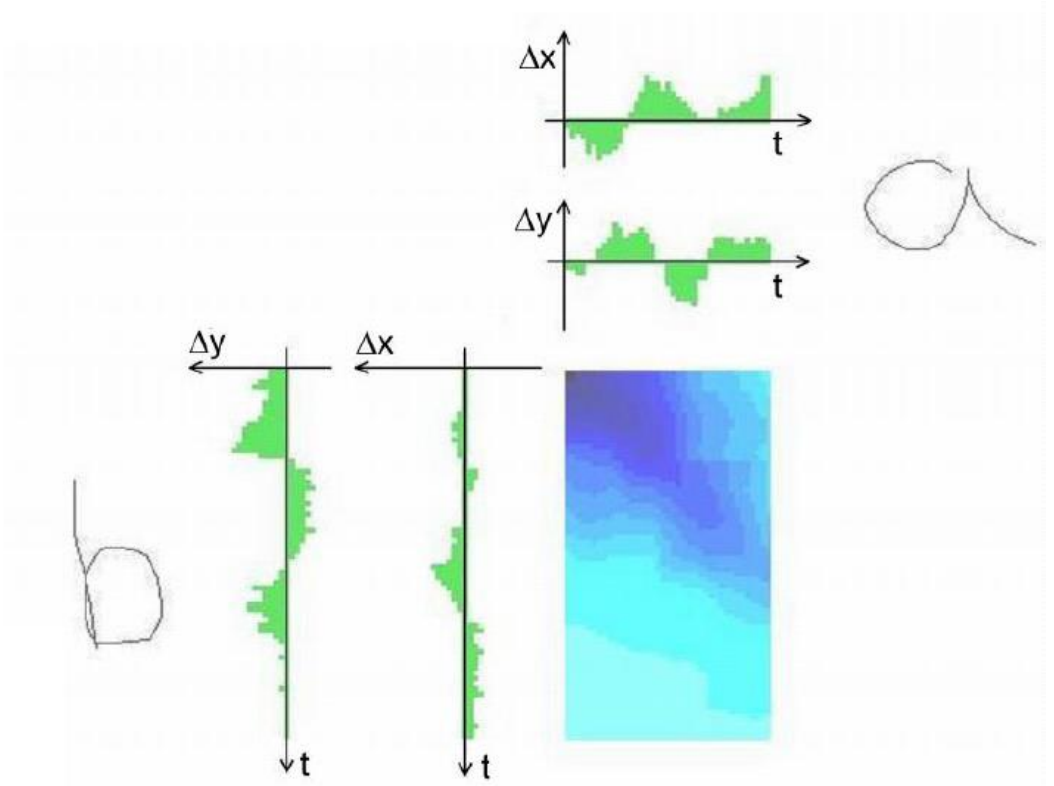
	1	1	3	4	3	1
1	0	0	2	5	7	7
2	1	1	1	3	4	5
3	3	3	1	2	2	4
3	5	5	1	2	2	4
4	8	8	2	1	2	5
1	8	8	4	4	3	2

$$|3 - 3| + \min\{3, 4, 2\} = 2$$

DTW distance of time series x_1 and x_2

Handwritten character recognition using DTW





Problem

Assuming that the cost of compression/stretching is 0, determine the DTW distance of the following time series (let the inner distance function be the absolute distance)! Find optimal alignment between two time series (the warping path)!

$$t_1 = (3, 2, 5, 7, 8, 9), \qquad t_2 = (2, 3, 2, 3, 6, 8)$$

Acknowledgement

- András Benczúr, Róbert Pálovics, SZTAKI-AIT, DM1-2
- Krisztián Buza, MTA-BME, VISZJV68
- Bálint Daróczy, SZTAKI-BME, VISZAMA01
- Judit Csima, BME, VISZM185
- Gábor Horváth, Péter Antal, BME, VIMMD294, VIMIA313
- Lukács András, ELTE, MM1C1AB6E
- Tim Kraska, Brown University, CS195
- Dan Potter, Carsten Binnig, Eli Upfal, Brown University, CS1951A
- Erik Sudderth, Brown University, CS142
- Joe Blitzstein, Hanspeter Pfister, Verena Kaynig-Fittkau, Harvard University, CS109
- Rajan Patel, Stanford University, STAT202
- Andrew Ng, John Duchi, Stanford University, CS229