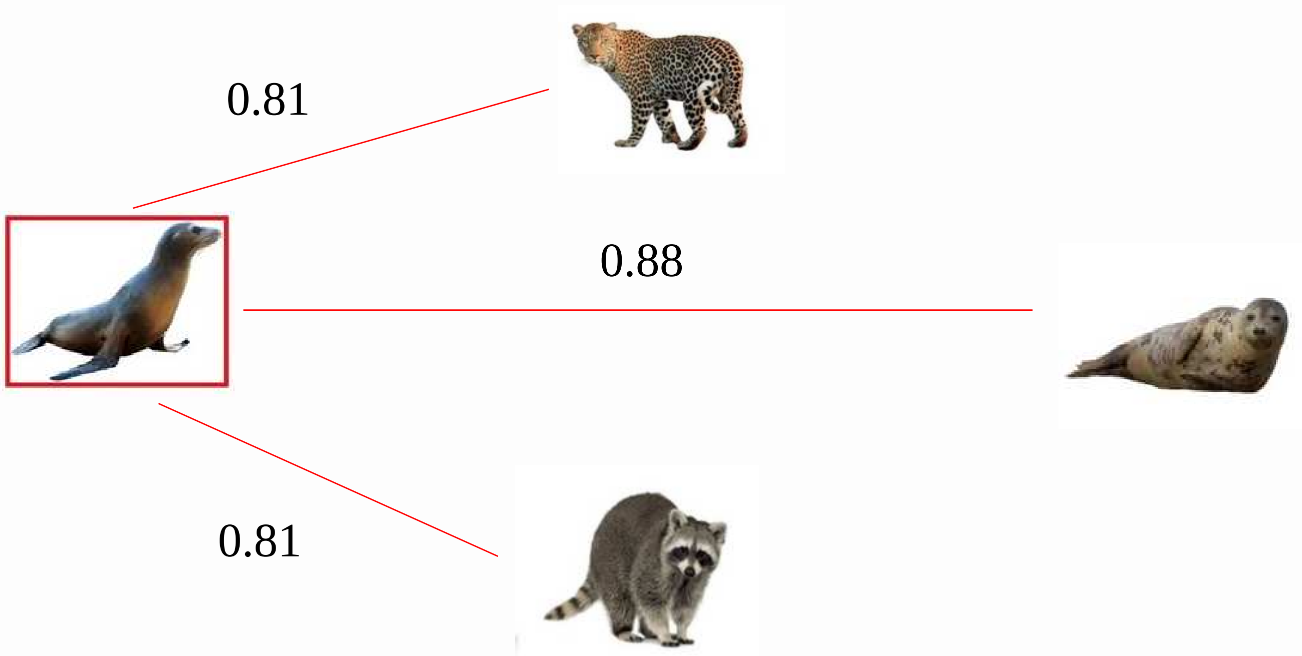
Lecture 3: Similarity and distance measures



Book chapter 3

---

image source Lin 2018. `https://www.linkedin.com/pulse/cosine-similarity-classification-michael-lin`

Contents

- Concepts of distance, similarity, metric
- Measures for numerical data (L_p-norms, similarity measures, accounting for distribution)
- Measures for categorical and mixed data
- Measures for sets, strings, text

What is distance?

Let $\mathcal{S}$ be a space of data objects. A distance function has the type

$$d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^{+} \cup \{0\}$$

Ituitively: Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{S}$ be objects.

- if $d(\mathbf{x}, \mathbf{y})$ small, $\mathbf{x}$ and $\mathbf{y}$ are close or similar
- If $d(\mathbf{x}, \mathbf{y}) < d(\mathbf{x}, \mathbf{z})$, $\mathbf{x}$ is closer/more similar to $\mathbf{y}$ than $\mathbf{z}$

Similarity vs. distance

Similarity function $s : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$

- $s(\mathbf{x}, \mathbf{y})$ large when $\mathbf{x}$ and $\mathbf{y}$ similar (and $d(\mathbf{x}, \mathbf{y})$ small)
- often $s : \mathcal{S} \times \mathcal{S} \to [0, 1]$
- $\Rightarrow$ possible to induce distance $d_s = 1 - s$
- if $d : \mathcal{S} \times \mathcal{S} \to [0, 1]$, possible to induce similarity $s_d = 1 - d$
- if not, then e.g.,
  $s_d = 1 - \frac{d}{D}$ (D=maximal possible distance) or
  $s_d = \frac{1}{1+d}$

Metric: distance d that satisfies 4 properties

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ (non-negativity or separation)
2. $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$ (coincidence axiom)
3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (symmetry)
4. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ **(triangle inequality)**

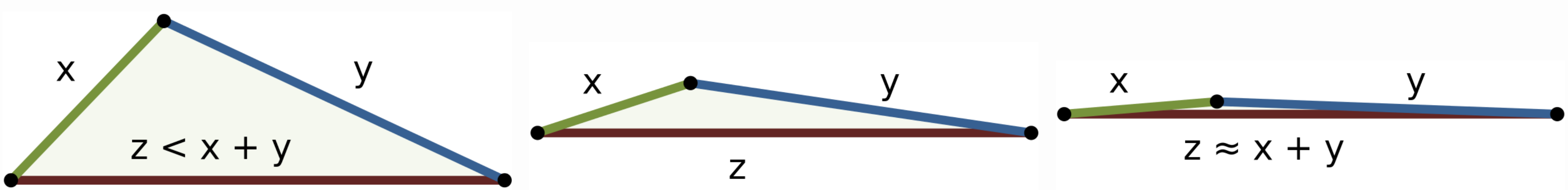


Image source `https://en.wikipedia.org/wiki/Triangle_inequality`

Metric space

Metric space $(\mathcal{S}, d)$ = data space equipped with a metric

- e.g., 3-D Euclidean space or any normed vector space
- no need to be a vector space! (e.g., space of strings + suitable metric)

Why they are so nice?

- many tasks can be performed more efficiently!
- especially similarity search (find nearest neighbours, closest cluster centers, similar documents,...)

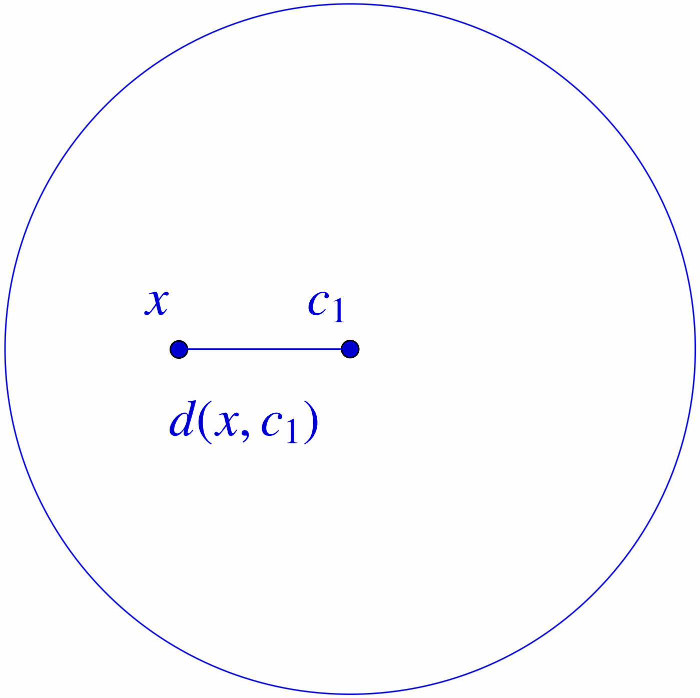
Example how $\triangle$ inequality can speed up things

Problem: Given cluster centroids $\mathbf{c}_1, \ldots, \mathbf{c}_K$, find the closest $\mathbf{c}_i$ for all data points $\mathbf{x}$. (d is a metric)

1. Naive solution: calculate all $d(\mathbf{x}, \mathbf{c}_i)$. ($nK$ calculations)
2. **Pruning trick**: Given $d(\mathbf{c}_i, \mathbf{c}_j)$ for all i, j and $d(\mathbf{x}, \mathbf{c}_1)$ to the currently closest $\mathbf{c}_1$.

   Test: If $d(\mathbf{c}_1, \mathbf{c}_2) > 2d(\mathbf{x}, \mathbf{c}_1)$, then $\mathbf{c}_2$ cannot be closer to $\mathbf{x}$!

   If $\mathbf{c}_2$ was closest to $\mathbf{c}_1$, then $\mathbf{c}_1$ is closest to $\mathbf{x}$.



c_2 cannot be inside the circle

since $d(c_1, c_2) > 2d(x, c_1)$

Example how $\triangle$ inequality can speed up things

More pruning by utilizing upper and lower bounds of distances!

Further reading:

- Elkan: Using the triangle inequality to accelerate k-means. ICML 2003.
- Hamerly: Making k-means even faster. SDM 2010.

Do you know distance or similarity measures for these data types?

- numerical
- categorical
- mixed
- sets
- binary
- strings
- text
- graphs

Multidimensional numerical: L_p*-norm*

Objects are $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$, $x_i, y_i \in \mathbb{R}$

Most common measure L_p**-norm** or **Minkowski distance**:

$$\boxed{L_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{k} |x_i - y_i|^p \right)^{\frac{1}{p}}}$$

- different variants by setting p
- e.g., **Euclidean distance** $L_2(\mathbf{x}, \mathbf{y}) = \left(\sum_i |x_i - y_i|^2 \right)^{1/2}$
- **metric, if** $p \geq 1$

Manhattan ("city block") distance L_1

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_i |x_i - y_i|$$

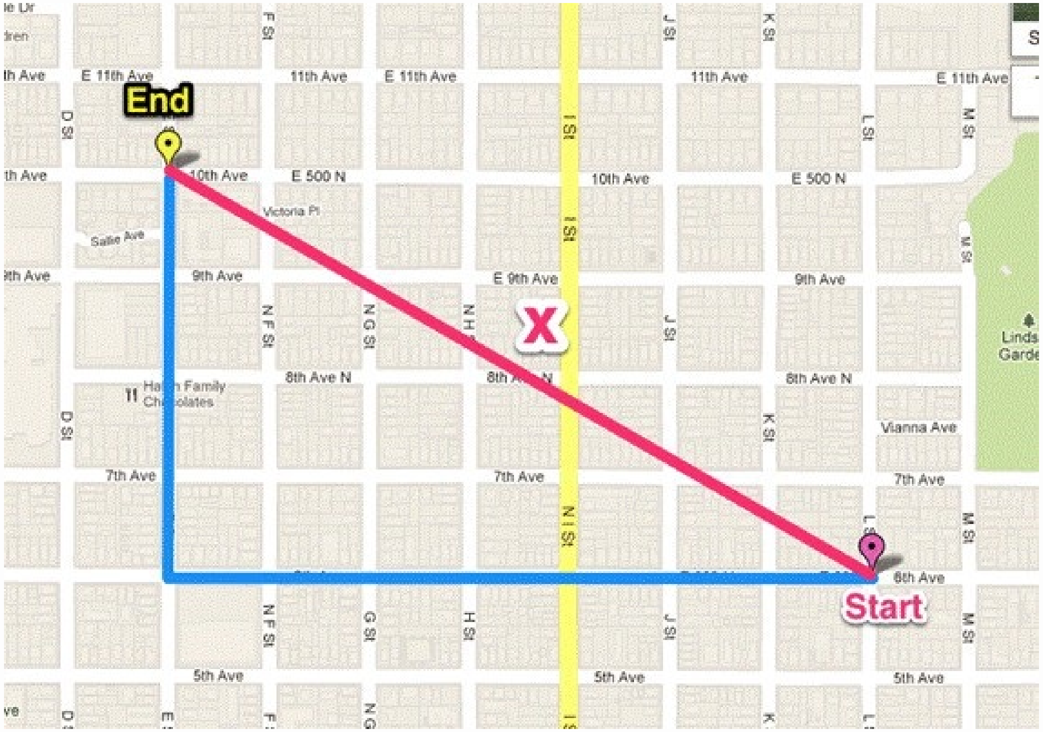
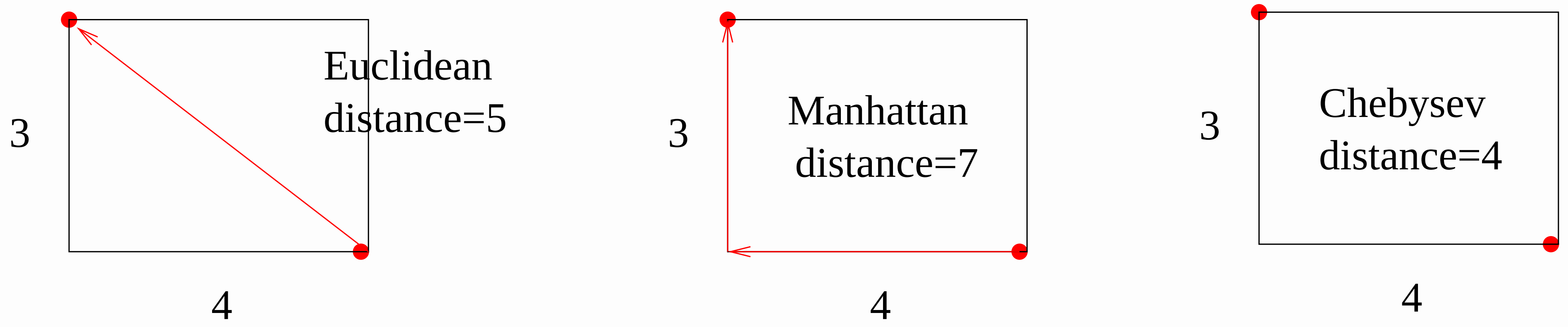


image source `https://medium.com/@paubric/the-square-circle-exploiting-distance-cef434f7f550`

L_p-norms

- $p = 1$: Manhattan distance $L_1(\mathbf{x}, \mathbf{y}) = \sum_i |x_i - y_i|$
- $p = 2$: Euclidean distance $L_2(\mathbf{x}, \mathbf{y}) = \left(\sum_i |x_i - y_i|^2\right)^{1/2}$
- $p \to \infty$: Chebyshev distance $L_\infty(\mathbf{x}, \mathbf{y}) = \max_i |x_i - y_i|$



L_p-norms do not work well in high dimensions

Curse of dimensionality: Contrasts $\frac{D_{\max}-D_{\min}}{D_{avg}}$ between largest and smallest distances disappear. Behaviour in random data:

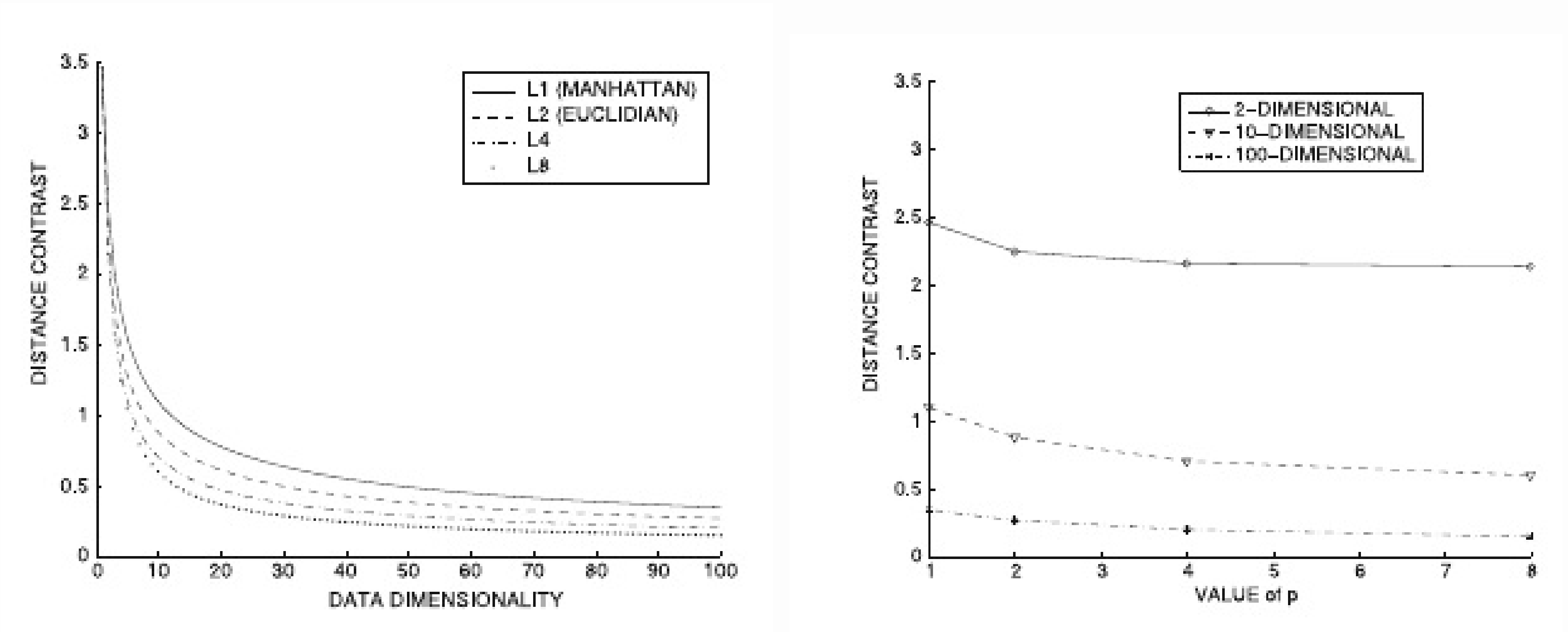


Image source: Aggarwal 2015, ch 3.2

L_p-norms do not work well in high dimensions

- irrelevant features tend to dominate $L_2, \ldots, L_\infty$
- Consider $L_\infty(\mathbf{x}, \mathbf{y})$, when $\mathbf{x}$ and $\mathbf{y}$ have similar value in 999 dimensions but dissimilar in 1 irrelevant attribute!

$$\Rightarrow$$

- generalized Minkowski distance give weights a_i reflecting importance: $L_p(\mathbf{x}, \mathbf{y}) = \left(\sum_i a_i |x_i - y_i|^p\right)^{\frac{1}{p}}$
- fractional L_p quasinorms set $p \in]0, 1[$ (**not metrics**)
- match-based similarity with proximity thresholding

Match-based similarity with proximity thresholding

Observations:

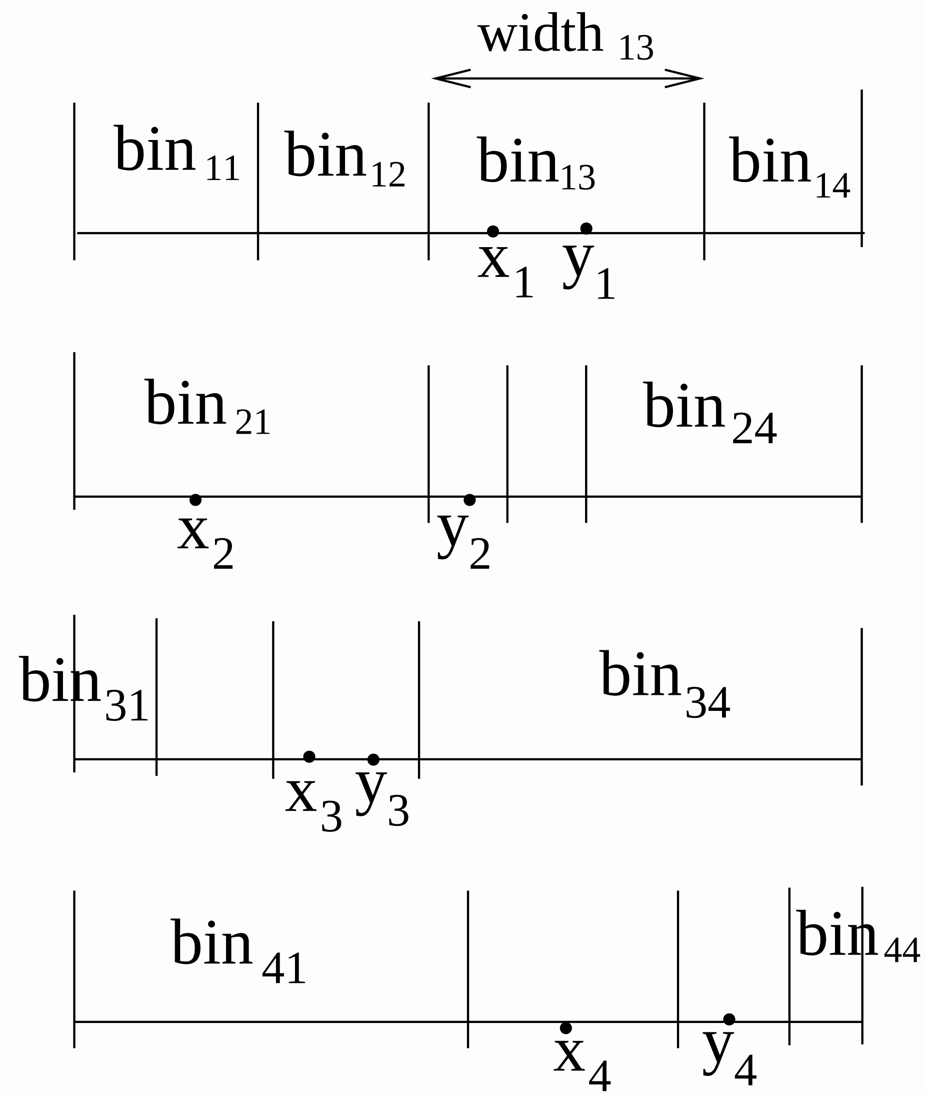
1. Features may be only **locally relevant** (e.g., blood glucose for diabetic patients but not for epileptic).
2. In large dimensions, two objects are unlikely to have similar values, unless the feature is relevant.

$\Rightarrow$ **emphasize dimensions where objects are close/similar!**

(Euclidean and pals do the opposite)

Match-based similarity with proximity thresholding

- discretize all dimensions to m equi-depth bins, bin_{ij} (i=dimension, j=bin number)
- $\mathbf{x}$ and $\mathbf{y}$ are **in proximity on dimension** i, if $x_i, y_i \in bin_{ij}$ for some j
- **proximity set** $S(\mathbf{x}, \mathbf{y}, m)$ = list of dimensions, where x_i and y_i in the same bin e.g., here $S(\mathbf{x}, \mathbf{y}, 4) = \{1, 3\}$



Match-based similarity with proximity thresholding

Similarity measure

$$PSelect(\mathbf{x}, \mathbf{y}, m) = \left[\sum_{i \in S(\mathbf{x},\mathbf{y},m); x_i \in bin_{i,j}} \left(1 - \frac{|x_i - y_i|}{width_{i,j}} \right)^p \right]^{1/p}$$

- ignores dimensions where $\mathbf{x}$ and $\mathbf{y}$ not in proximity
- value when i) $\mathbf{x} = \mathbf{y}$? ii) $S(\mathbf{x}, \mathbf{y}, m) = \emptyset$?
- how to choose parameters? ($m \propto k$ + e.g., $p = 1$ or $p = 2$)

Aggarwal & Yu (2000): The IGrid Index: Reversing the Dimensionality Curse For Similarity Indexing in High Dimensional Space.

Cosine similarity and distance

Cosine similarity:

$$cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$$

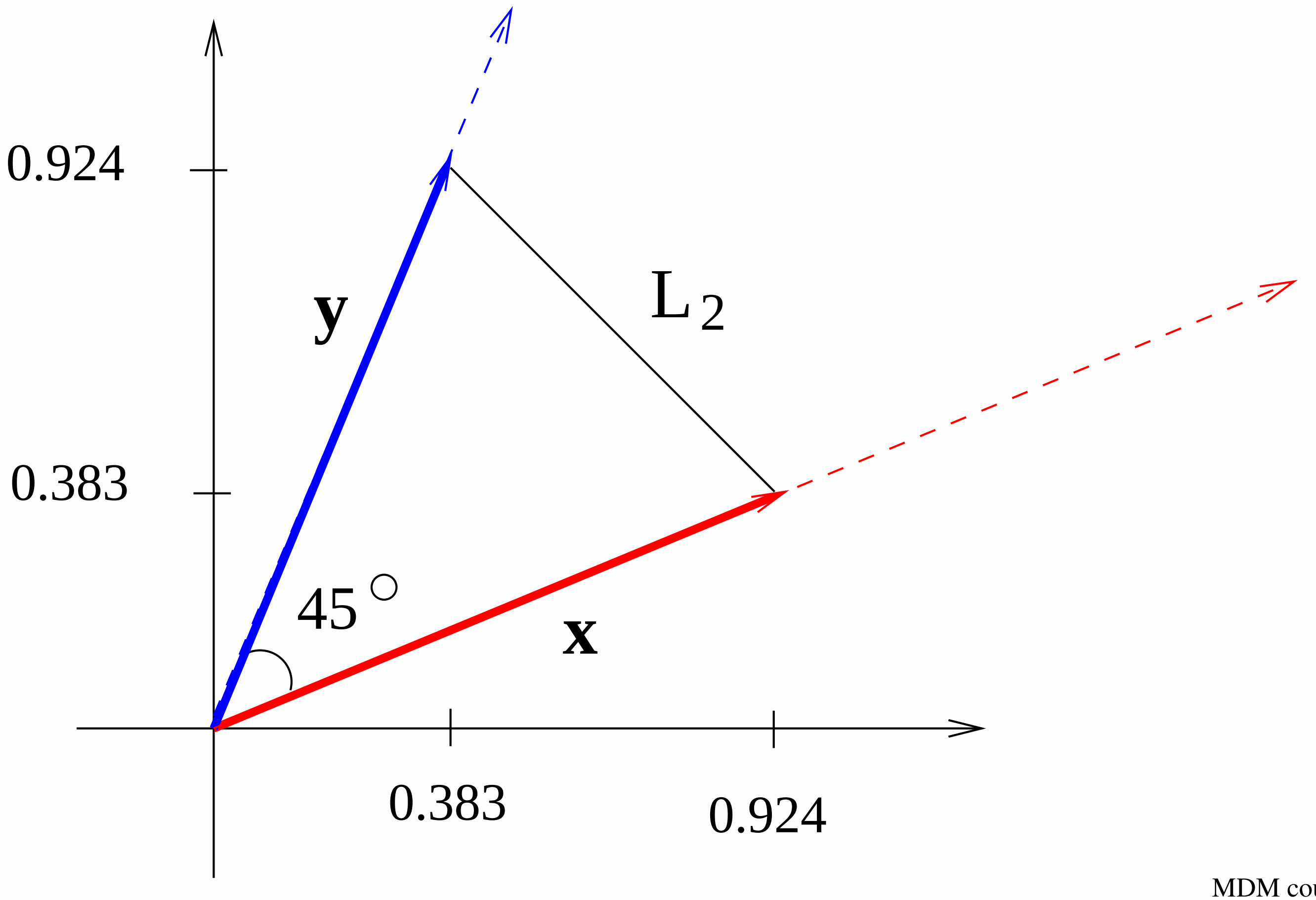
- suitable for numerical (continuous or integers) and binary data
- in $[-1, 1]$, most similar if $cos(\mathbf{x}, \mathbf{y}) = 1$
- popular for text documents (their numerical presentation)

Cosine distance: $1 - cos(\mathbf{x}, \mathbf{y})$

- $[0, 1]$ if all vector elements non-negative ($x_i \geq 0$)

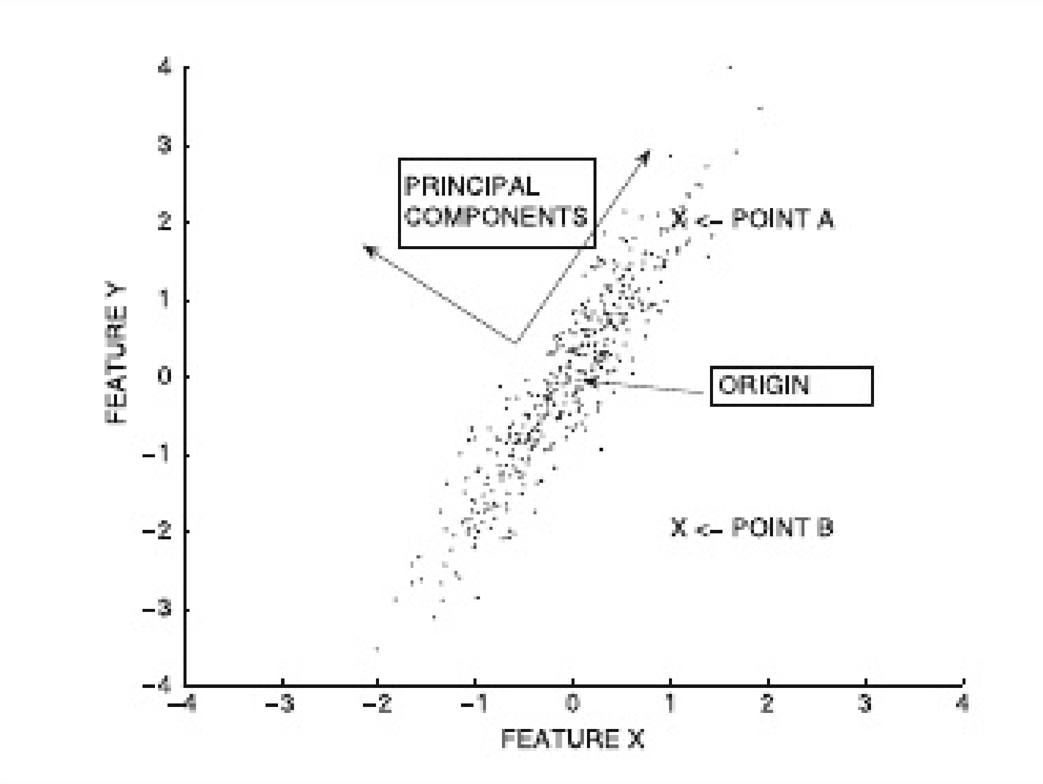
Cosine similarity and distance

Relationship to Euclidean distance L_2:
if vectors are normalized (length 1), $L_2^2(\mathbf{x}, \mathbf{y}) = 2(1 - cos(\mathbf{x}, \mathbf{y}))$



Should the distance reflect data distribution?

Should A and B be equally distant from the origin?



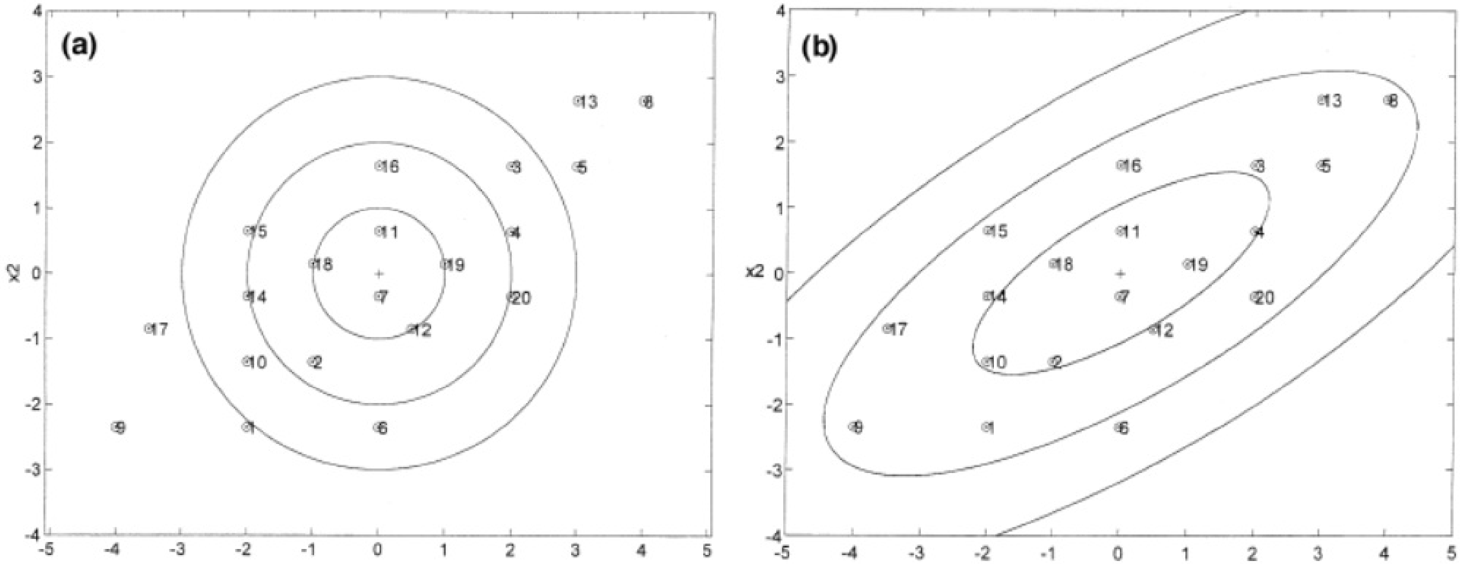
high variance direction $\Rightarrow$ more likely to be distant $\Rightarrow$
could consider A closer than B $\Rightarrow$
Mahalanobis distance

$$Maha(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\Sigma^{-1}(\mathbf{x} - \mathbf{y})^T}$$

(Σ = covariance matrix)

Read Aggarwal 3.2.1.6

Mahalanobis distance $Maha(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\Sigma^{-1}(\mathbf{x} - \mathbf{y})^T}$



Points on each circle have the same Euclidean distance to the origin.

Points on each ellipsis have the same Mahalanobis distance to the origin.

image source `https://queirozf.com/entries/similarity-measures-and-distances-basic-reference-for-data-science-practitioners`

Should the distance reflect data distribution?

Which pair of points are closest to one another?

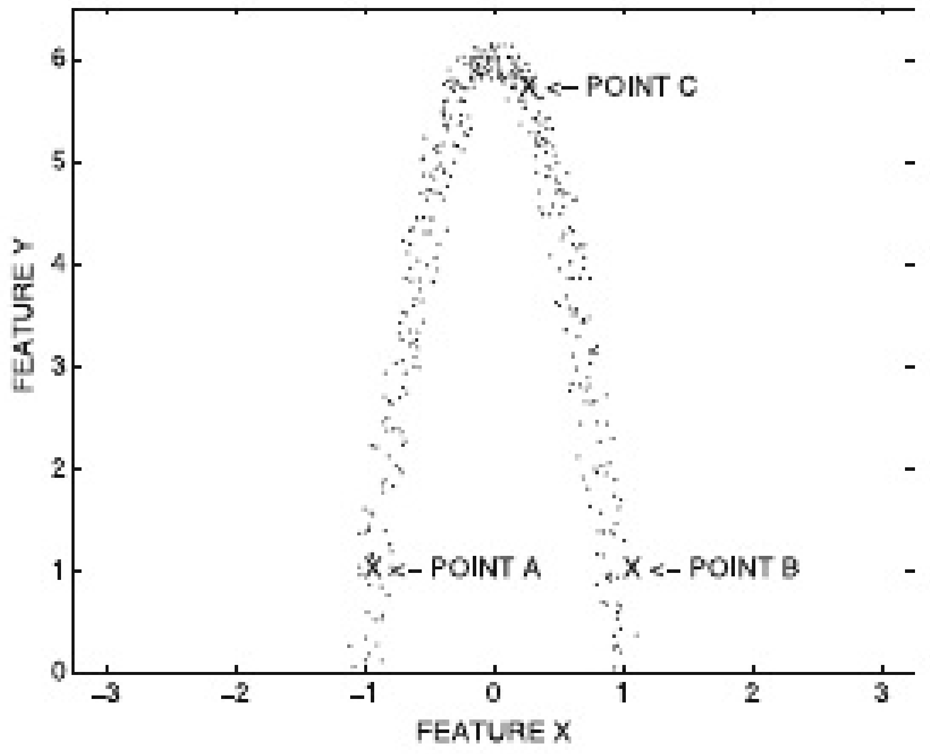


image source Aggarwal 3.2.1.7

Analogy: what is your walking distance to the other shore?

Idea: Measure distances along shortest paths in a nearest neighbour graph

ISOMAP method:

1. Create a nearest neighbour graph $G = (V, E)$ where each $v \in V$ in connected to K nearest neighbours and edge weights represent distances.
2. For any points $v_1, v_2 \in V$

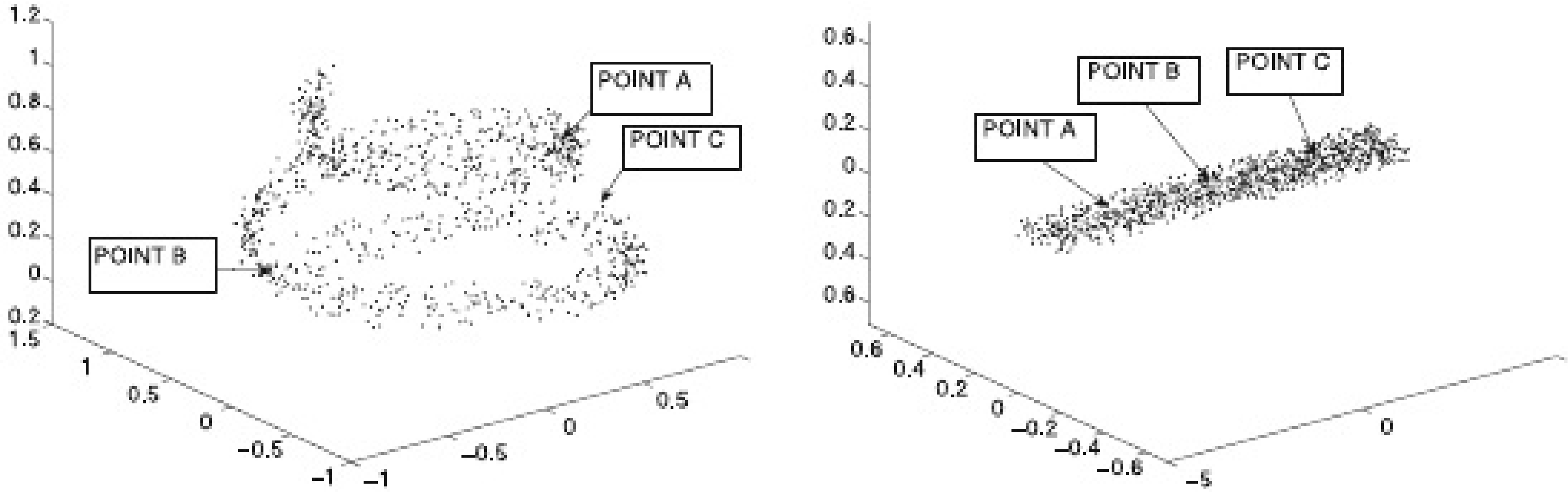
$$Dist(v_1, v_2) = |\text{shortest-path}(v_1, v_2)|$$

3. Optional step: embed the data into multidimensional space with multidimensional scaling $\rightarrow$ lower dimensional representation

Use either $Dist(v_1, v_2)$ or L_p distances in the new space

ISOMAP

The data shape becomes straightened out:



(a) A and C seem close (original data)

(b) A and C are actually far away (*ISOMAP* embedding)

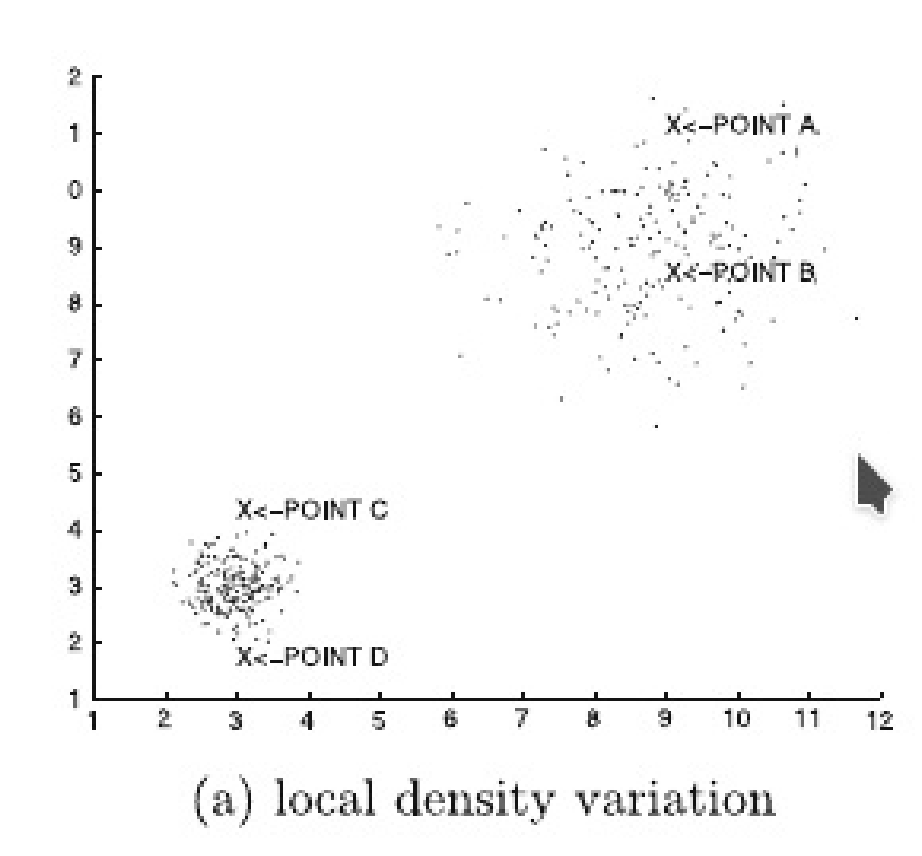
Figure 3.5: Impact of *ISOMAP* embedding on distances

---

image Aggarwal 3.2.1.7

Should the distance reflect data distribution?

Should $d(A, B) < d(C, D)$ or vice versa?

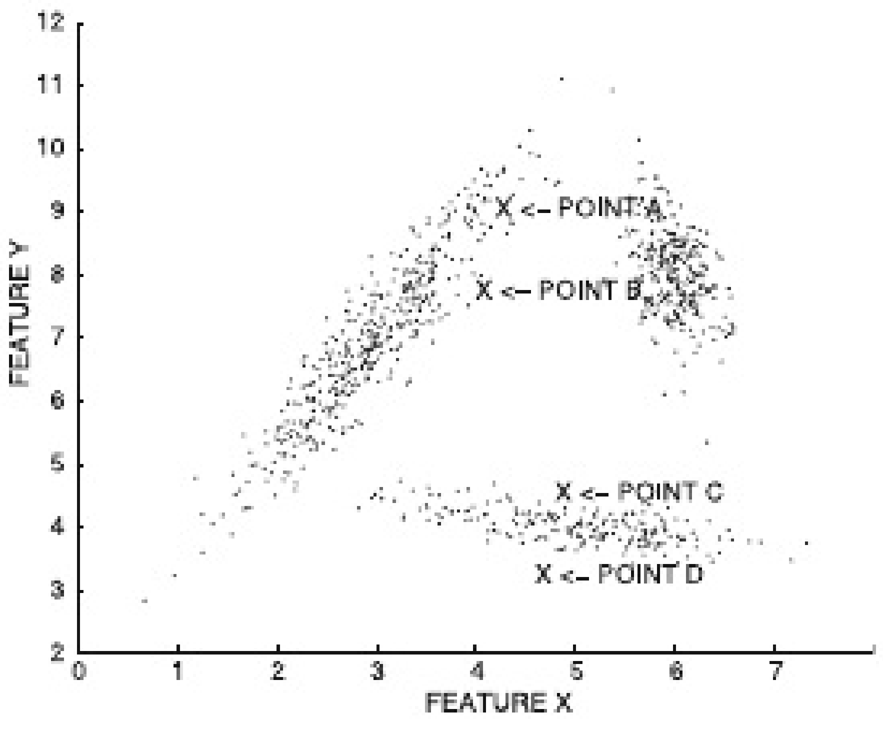


shared nearest-neighbour similarity = number of shared neighbours
$\Rightarrow$ similarity graph

Read Aggarwal 3.2.1.8

Should the distance reflect data distribution?

Should $d(A, B) < d(C, D)$ or vice versa?



(b) local orientation variation

- partition data and use local statistics to adjust distances (local Mahalanobis)
- but partitioning already requires distance measures!

Read Aggarwal 3.2.1.8

Categorical data: similarity

Generic function:

$$sim(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} w_i s(x_i, y_i)$$

- typically weight $w_i = \frac{1}{k}$ (k=number of features)
- many choices for s, e.g., in **overlap similarity** s is

$$s(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases} \qquad \Rightarrow$$

overlap similarity = fraction of dimensions where $\mathbf{x}$ and $\mathbf{y}$ have an equal value

Categorical data: similarity

Or take into account frequency of value:

$$p_i(x_i) = \frac{fr(A_i = x_i)}{n} \quad = \text{fraction of records having } A_i = x_i$$

Goodall measure (its one variant):

$$s(x_i, y_i) = \begin{cases} 1 - p_i^2(x_i) & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases}$$

Further reading Boriah et al. (2008): Similarity measures for categorical data: A comparative evaluation.

Task

Create a similarity graph using overlap similarity. Include only edges where similarity is $\geq 2/3$. Which foxes are most similar to Bella? What if you use the Goodall measure instead?

name	sex	colour	character
Bella	F	red	tame
Molly	F	red	shy
Teddy	M	red	tame
Ruby	F	red	brave
Coco	F	silver	cool
Max	M	silver	brave

$$\text{overlap} = \frac{\#(\text{overlapping feature values})}{\#\text{features}}$$

$$\text{Goodall} = \frac{\sum_{A_i \text{ shared}}(1-p_i^2(\text{shared value}))}{\#\text{features}}$$

Task

$$\text{overlap} = \frac{\#(\text{overlapping feature values})}{3}$$

pair	common	overlap	Goodall
Bella–Molly	F, red	2/3	
Bella–Teddy	red, tame	2/3	
Bella–Ruby	F, red	2/3	
Molly–Ruby	F, red	2/3	

Task

$$\text{Goodall} = \frac{\sum_{A_i \text{ shared}}(1 - p_i^2(\text{shared value}))}{\#\text{features}}$$

p_1(F)=2/3, p_1(M)=1/3, p_2(red)=2/3, p_2(silver)=1/3,
p_3(tame)=p_3(brave)=1/3, p_3(shy)=p_3(cool)=1/6
1-p_1^2(F)+1-p_2^2(red)=10/9
1-p_2^2(red)+1-p_3^2(tame)=13/9

pair	common	overlap	Goodall
Bella–Molly	F, red	2/3	10/27
Bella–Teddy	red, tame	2/3	13/27
Bella–Ruby	F, red	2/3	10/27
Molly–Ruby	F, red	2/3	10/27

Similarity in mixed data (without transformations)

Give weights to numerical and categorical components:

$$sim(\mathbf{x}, \mathbf{y}) = \lambda \cdot NumSim + (1 - \lambda) \cdot CatSim$$

- How to choose λ? ($\lambda \in [0, 1]$)
- e.g., fraction of numerical features in data
- $NumSim$ and $CatSim$ often in different scales $\Rightarrow$
    - calculate standard deviations (σ_N and σ_C) of pairwise similarities with $NumSim$ and $CatSim$

$$sim(\mathbf{x}, \mathbf{y}) = \lambda \cdot NumSim/\sigma_N + (1 - \lambda) \cdot CatSim/\sigma_C$$

Binary data: distance and similarity

Data points $\mathbf{x}$ and $\mathbf{y}$ are bit strings (length k)

Hamming distance = L_1 norm for binary data

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_i |x_i - y_i|$$

=number of positions where bits differ

x	0	1	0	0	1	0	0	1	0
y	1	0	0	0	0	1	0	1	1

Hamming distance 5

image source CS-E4600 fall 2019 slides

Set data can be presented as binary

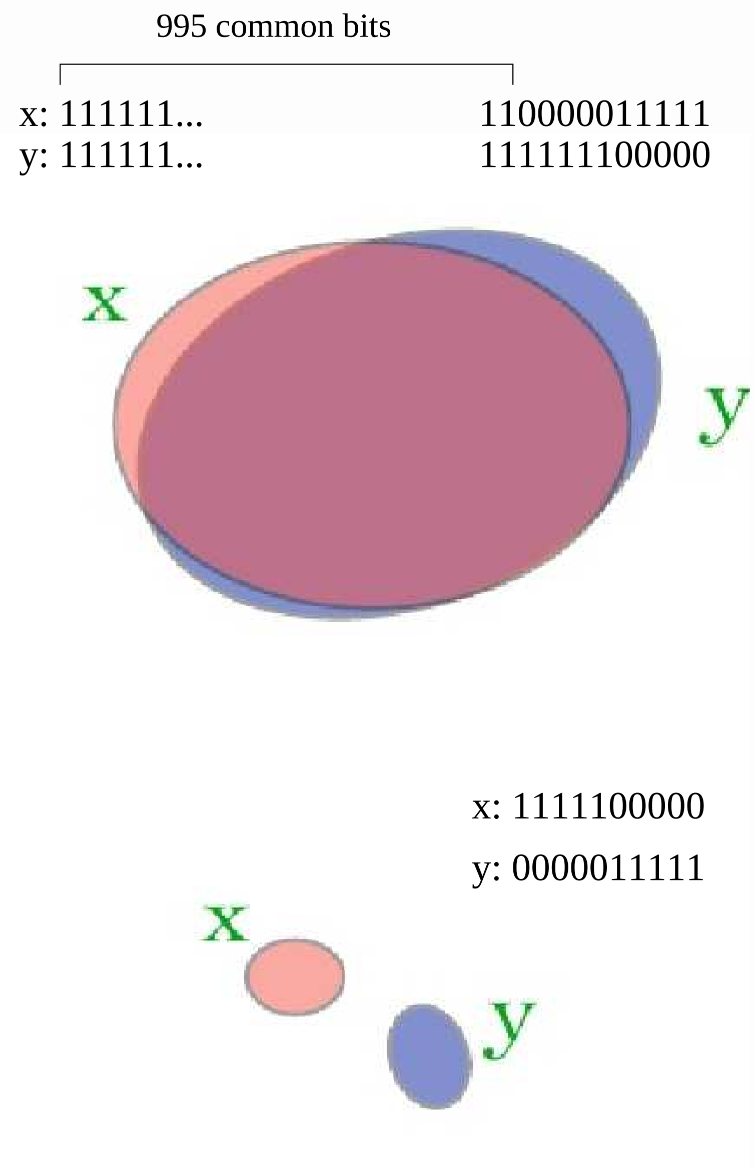
e.g., basket1: {white bread, cheese} ⇒ 0011000000000...

	low fat milk	apple juice	white bread	edam cheese	oranges	
basket1	0	0	1	1	0	
basket2	1	1	0	0	0	
basket3	0	1	0	1	0	...
basket4	1	0	1	0	1	
basket5			⋮			

- transactions (like market baskets)
- occurrence of words in documents
- over-expressed or under-expressed genes in samples

Set data often very **sparse** (= most values are 0s) ⇒ number of common elements more important

Hamming distance for transaction data?



1. Two sets with 1000 items and 995 common
2. Two sets with 5 items, but none common

Both have Hamming distance 10

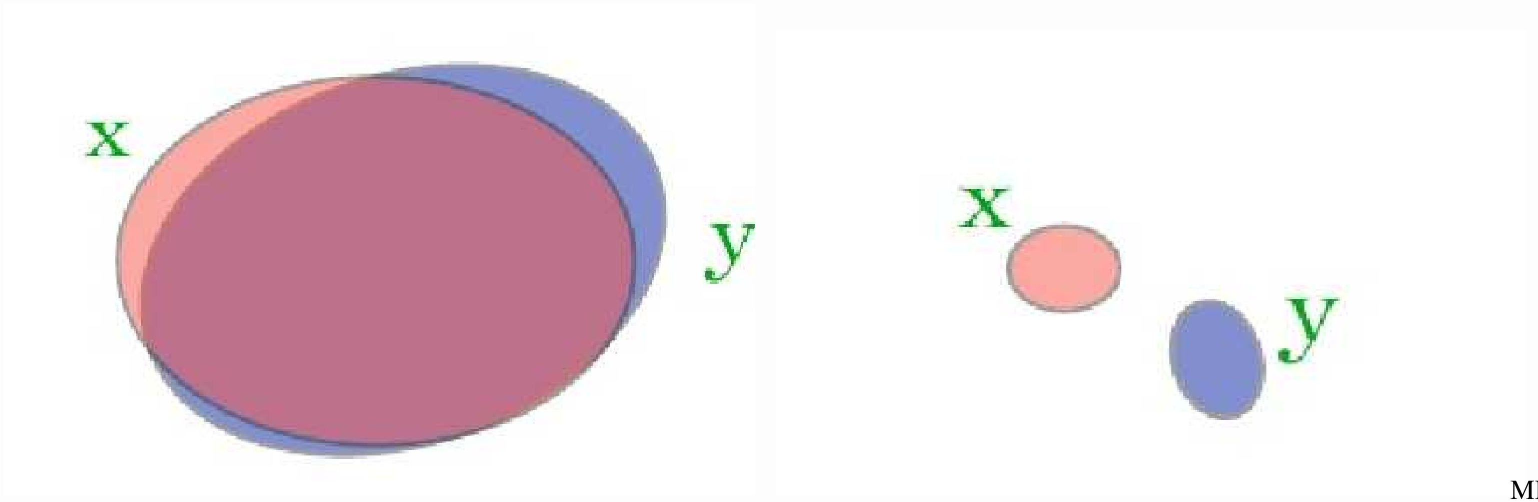
image source CS-E4600 fall 2019 slides/Aris Gionis

Jaccard coefficient for set similarity

Given sets $\mathbf{x}$ and $\mathbf{y}$

$$J(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x} \cap \mathbf{y}|}{|\mathbf{x} \cup \mathbf{y}|}$$

- treats 0s and 1s differently
- Previous example, case 1: $J = \frac{995}{1005} \approx 0.99$, case: 2 $J = 0$



String data: distance

Given strings $\mathbf{x}$ and $\mathbf{y}$ of the same length.
Modification of the Hamming distance

- add 1 for all positions that are different

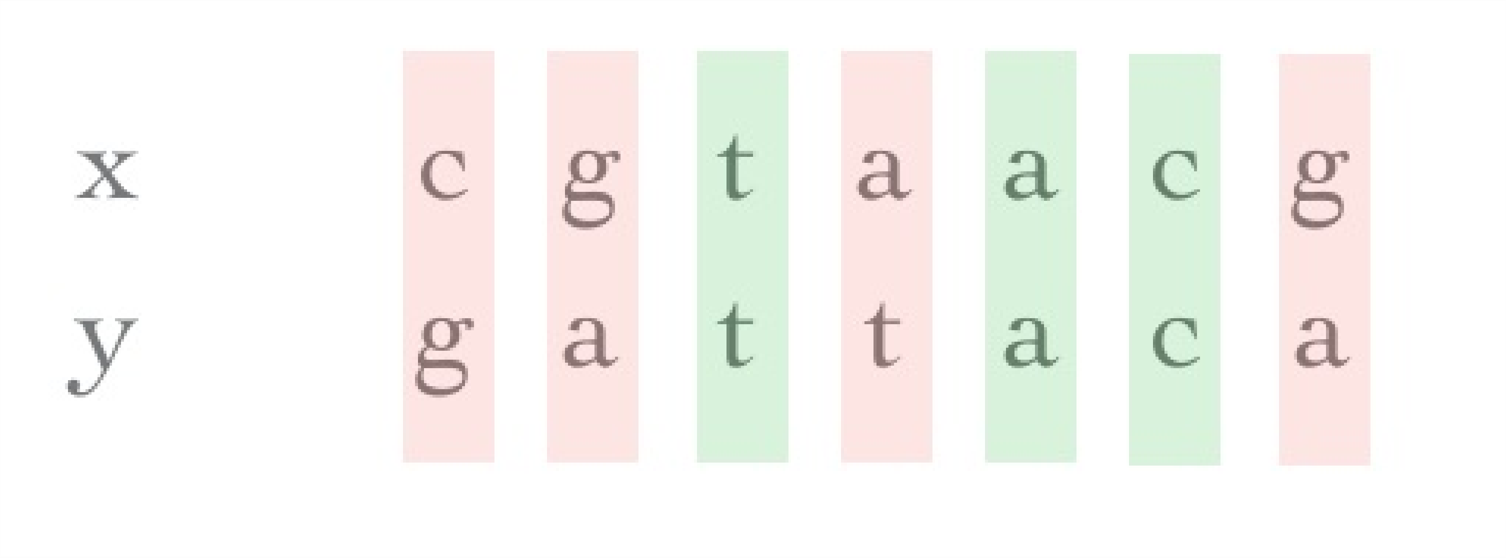
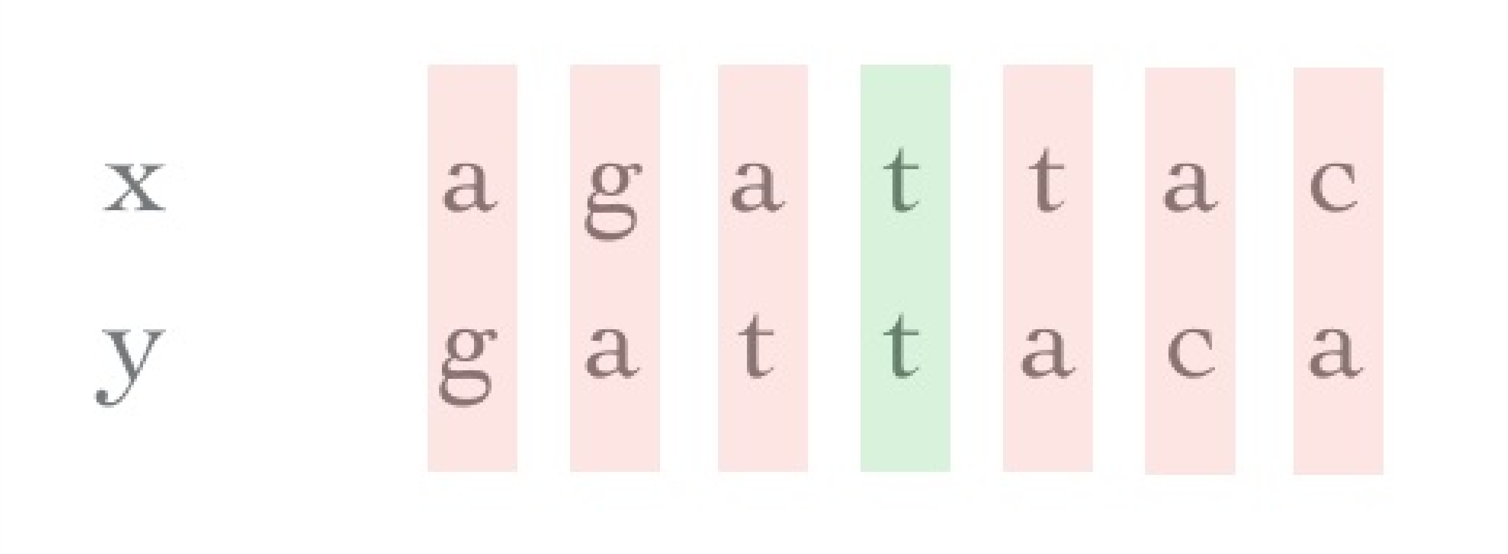


image source CS-E4600 fall 2019 slides/Aris Gionis

Is Hamming distance good for strings?

- Strings must have equal length
- Punishes a lot for small typos:



string Hamming distance = 6

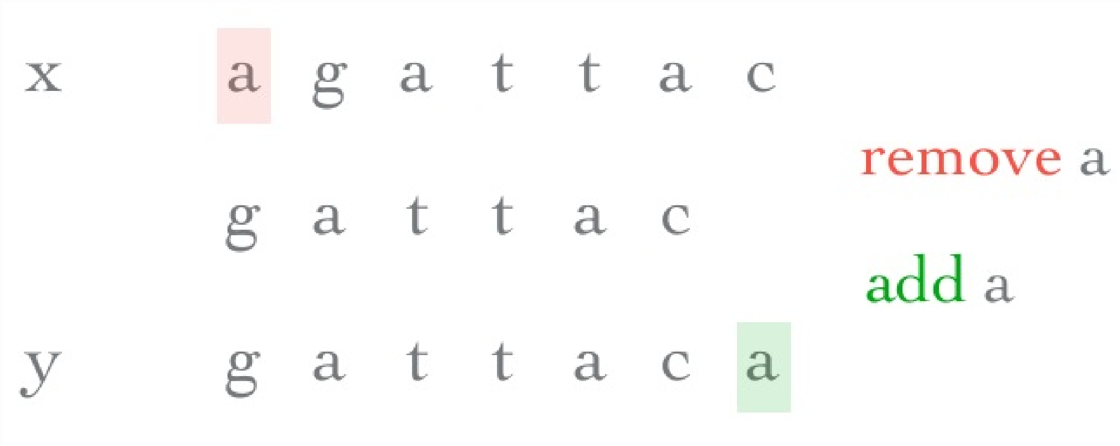
image source CS-E4600 fall 2019 slides/Aris Gionis

String edit distance

Given two strings $\mathbf{x}$ and $\mathbf{y}$, try to change one to another!

- only single-character edits are allowed
  - insert character
  - delete character
  - substitute character
- edit distance=minimum cost of such operations
- **Levenshtein distance**=minimum number of such operations (unit costs)
- edit operations can have different costs w_{ins}, w_{del}, w_{sub}
- **metric**, if positive costs and each operation has an inverse operation with the same cost

String edit distance examples



Levensteihn(kitten, sitting)=3:

1. **k**itten → **s**itten (substitute "s" for "k")
2. sitt**e**n → sitt**i**n (substitute "i" for "e")
3. sittin → sittin**g** (insert "g" at the end)

image source CS-E4600 fall 2019 slides

Text data: similarity between documents

Let's present text documents as **document-term matrices**.

- $\mathbf{x}$ and $\mathbf{y}$ are m -dimensional vectors (m = lexicon size)
- x_i = frequency of term i in the document $\mathbf{x}$
  - alternatively tf-idf value (tf-idf presentation) or binary value (Boolen model)
- then take **cosine similarity**:

$$cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$$

- in the Boolean model, Jaccard coefficient also possible

Task: Simplify the equation of cosine similarity when data is binary

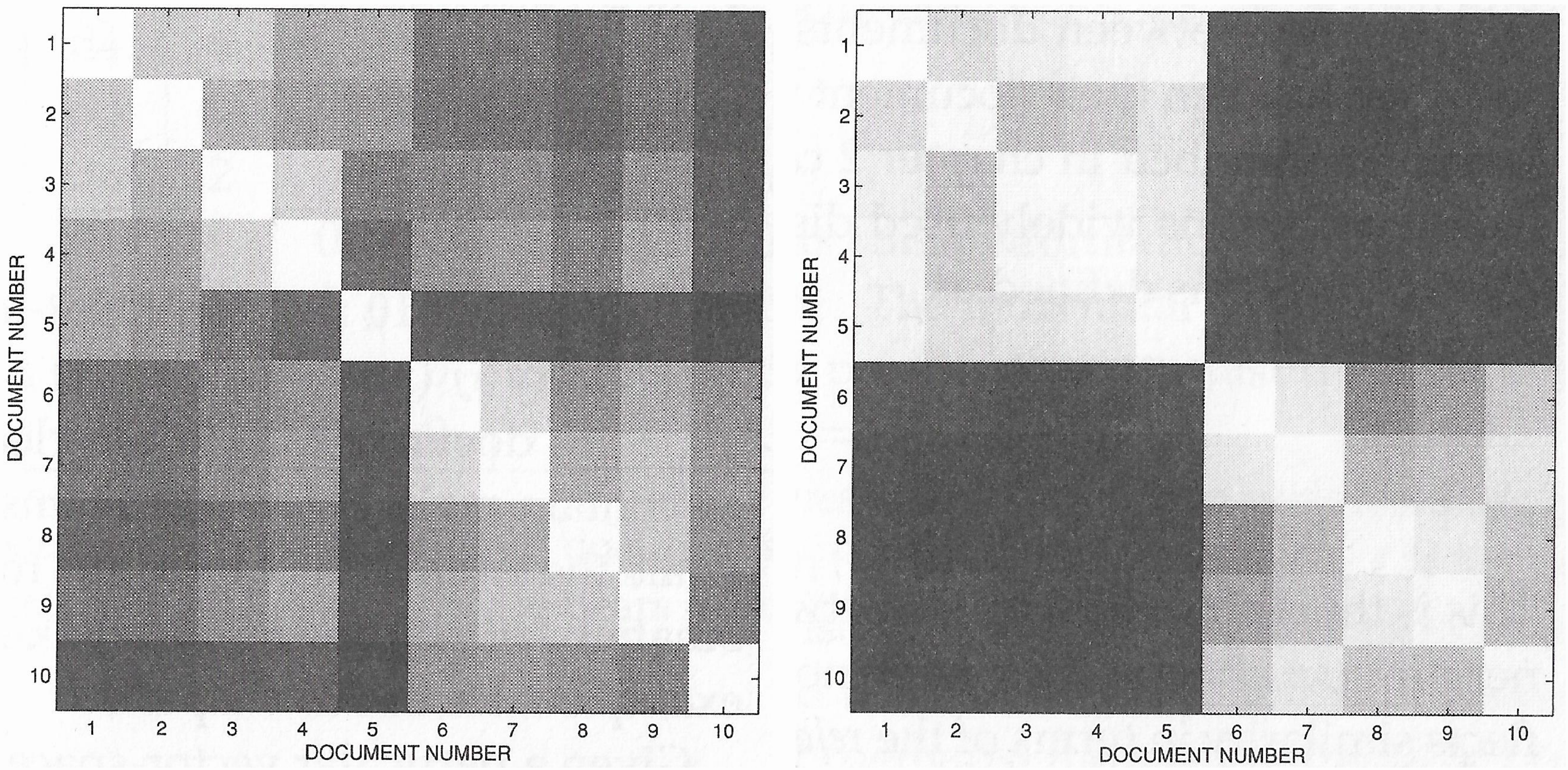
Text data: Example (Hand et al. 2001)

	t1	t2	t3	t4	t5	t6
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

source: Hand, Mannila, Smyth: Principles of data mining, 2001

Text data: Example (Hand et al. 2001)

Left: Euclidean distance (bright=small distance), right: cosine similarity (bright=large similarity)



Other data types

See the text book!

- time series: Ch 3.4
- graphs: Ch 3.5 and later in the course

Warning: There are many variants of the same measures and the names are not fixed! Give always the equation of the measure you use (+ a literature reference)!

Summary

- Choose distance and similarity measures carefully!
- Curse of dimensionality $\rightarrow$ for multidimensional data consider L_p with small p, cosine or match-based similarity
- If the distribution is very heterogenous, it is beneficial to adjust to local variations in distances (but costs!)
- Metric distances can speed similarity search, but non-metrics may perform better in high dimensions