

Similarity:

Numerical Data

Mining Massive Datasets

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



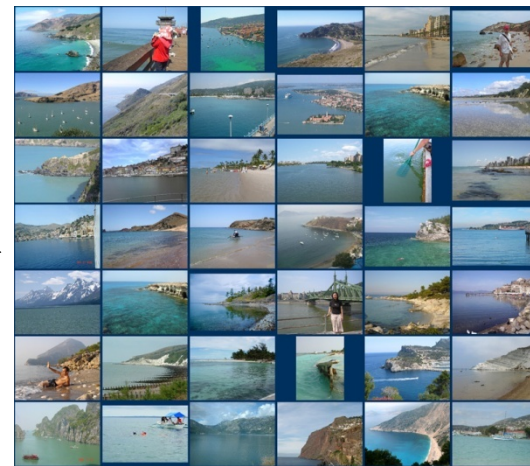
Universitat
Pompeu Fabra
Barcelona

Main Sources

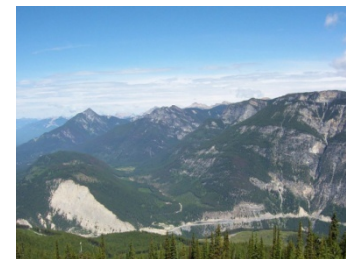
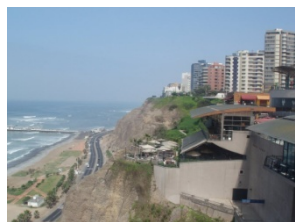
- Data Mining, The Textbook (2015) by Charu Aggarwal (Chapter 3) + [slides by Lijun Zhang](#)
- Data Mining Concepts and Techniques, 3rd edition (2011) by Han et al. (Section 2.4)
- Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 2)
- Mining of Massive Datasets 2nd edition (2014) by Leskovec et al. ([Chapter 3](#))

Example: scene completion

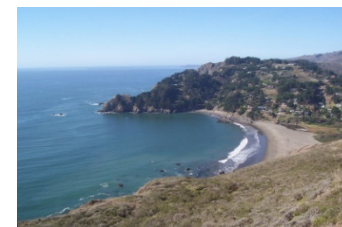
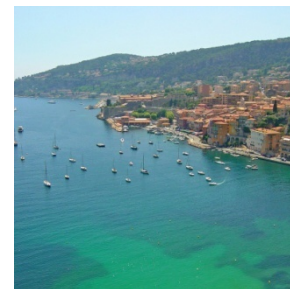
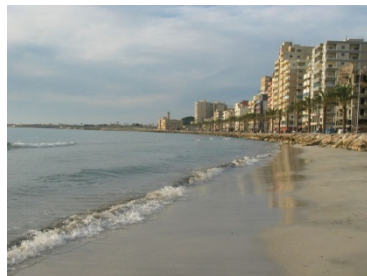
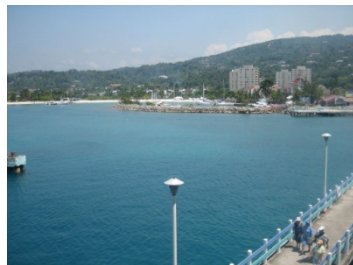
Scene completion problem



10 closest items in a collection of 20K images



10 closest items in a collection of 2M images



Computing similarity

Computing similarity is important

- **Many problems** can be expressed as finding “similar” sets:
 - Find near-neighbors in high-dimensional space
- Examples:
 - Pages with similar words, for duplicate detection or for classification by topic
 - Customers who purchased similar products, or products with similar customers
 - Images with similar features
 - Users who visited similar websites

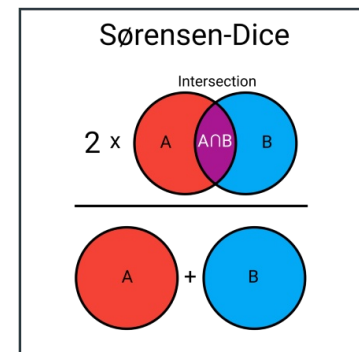
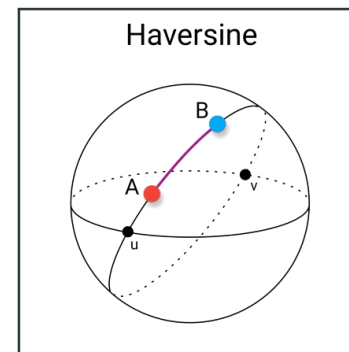
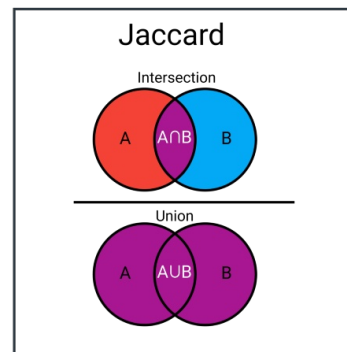
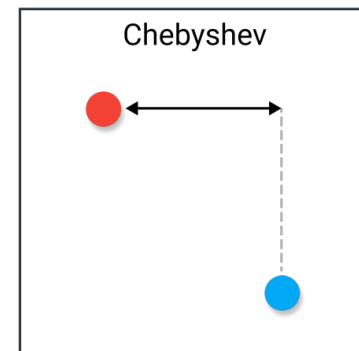
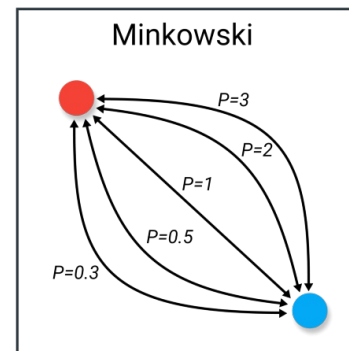
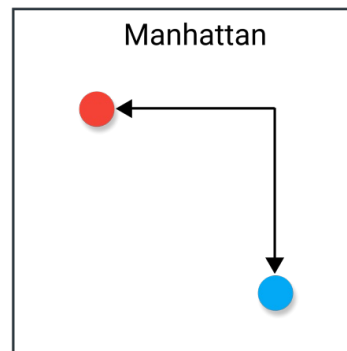
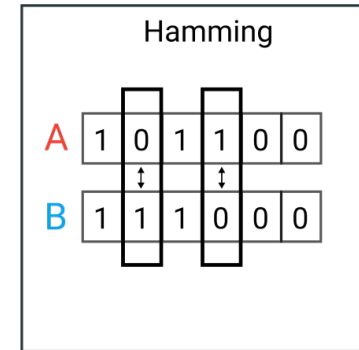
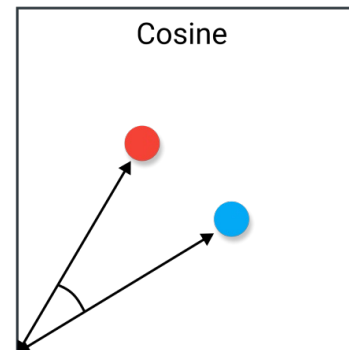
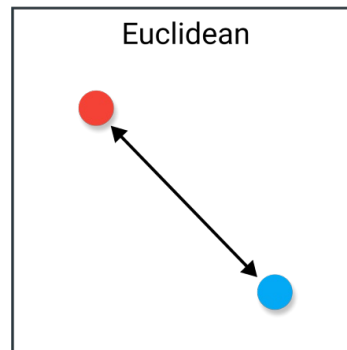
Similarity computation task

- Given two objects u and v , determine the value of:
 $\text{similarity}(u,v)$ and $\text{distance}(u,v)$ *Often one is defined in terms of the other*
- **Similar** objects should have
large similarity and small distance
- **Dissimilar** objects should have
small similarity and large distance
- We can use closed-form functions (e.g., euclidean distance) or an algorithm

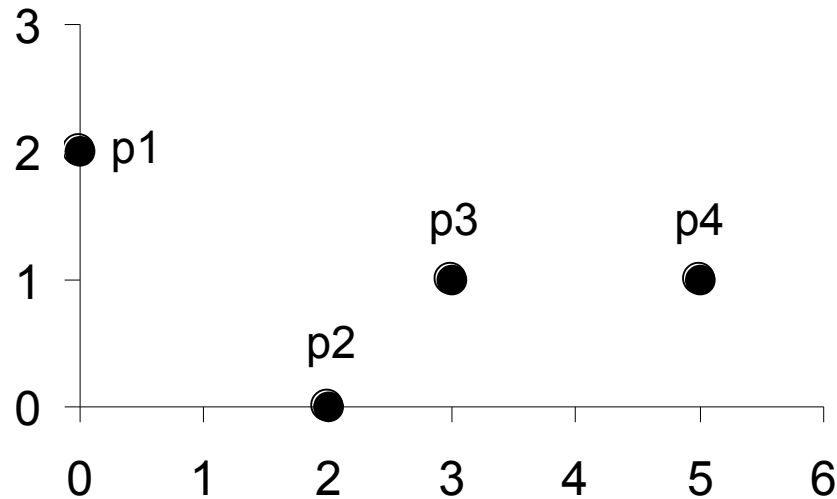
Simple single-attribute similarity

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = x - y / (n - 1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = x - y $	$s = -d, s = \frac{1}{1+d}, s = e^{-d},$ $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

Some distance measures



Euclidean distance: L_2 norm



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

	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

L_p norm, $p \geq 1$

- $p=1$: Manhattan norm
 - Sum of absolute values
- $p=2$: Euclidean norm
 - Square root of sum of squares
 - Rotation-invariant
- $p=\infty$: Infinity norm
 - Largest absolute value

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

Generalized L_p norm, $p \geq 1$

- Useful when some features are more important than others

$$L_p^{\text{GEN}} = \left(\sum_{i=1}^d a_i |x_i - y_i|^p \right)^{\frac{1}{p}}$$

- Coefficients a_i are domain-specific, typically non-negative

Exercise: compute L_p distance

- Given vectors

$$u = (22, 1, 42, 10)$$

$$v = (20, 0, 36, 8)$$

- Compute:

L_1 distance

L_2 distance

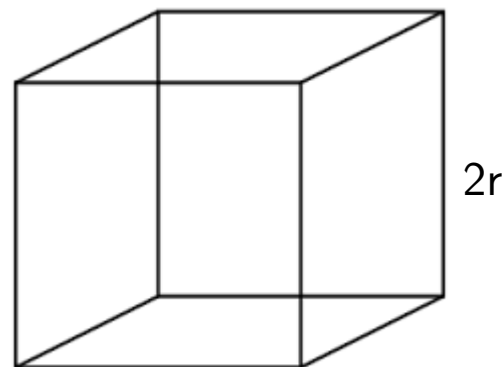
L_∞ distance

THE CURSE OF DIMENSIONALITY

THE CURSE OF DIMENSIONALITY

When the dimensionality is high, all points are similarly far from each other

Imagine a hypercube of side $2r$ in d dimensions. This hypercube has volume $(2r)^d$



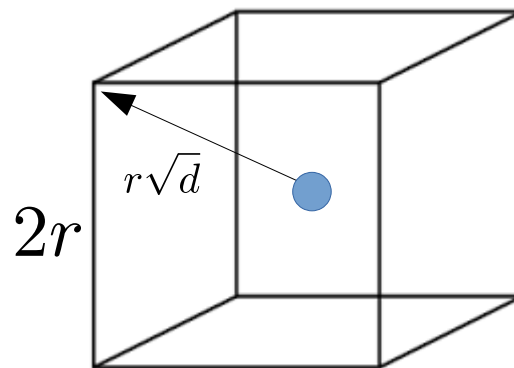
THE CURSE OF DIMENSIONALITY

When the dimensionality is high, all points are similarly far from each other

The corners are at distance $r\sqrt{d}$
from the center of the hypercube

That distance increases without bound as
the dimensionality increases!

Now, let us imagine a hypersphere of radius
 r inside the hypercube ...

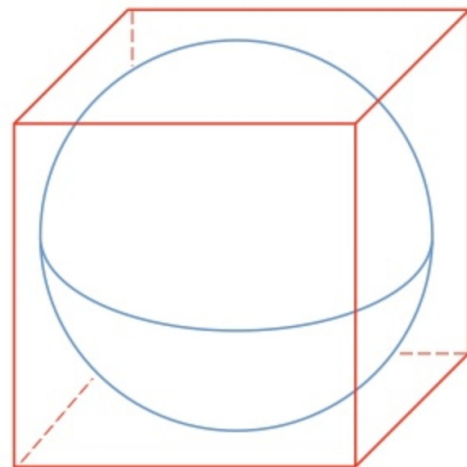
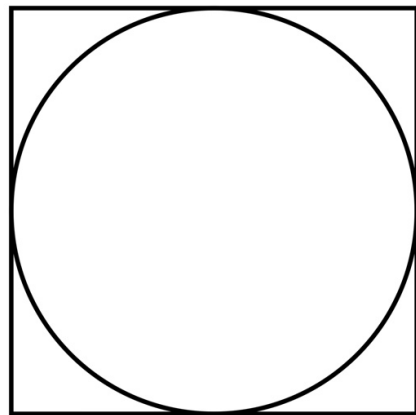


THE CURSE OF DIMENSIONALITY

When the dimensionality is high, all points are similarly far from each other

The corners are at distance $r\sqrt{d}$ from the center of the hypercube, which increases as the dimensionality increases

This means that a random point sampled from the hypercube is increasingly likely to be at distance larger than r from the center, i.e., outside of the hypersphere

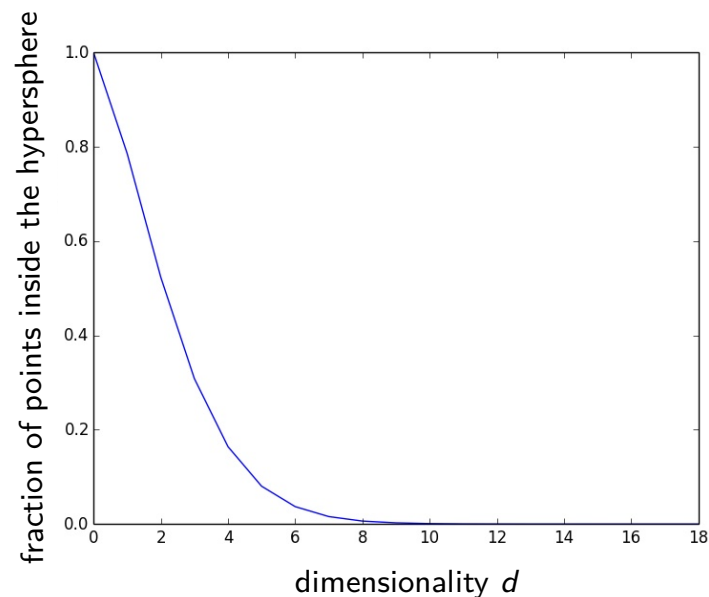
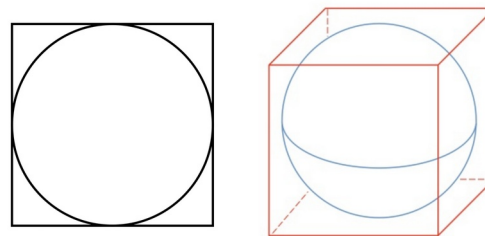


THE CURSE OF DIMENSIONALITY

When the dimensionality is high, all points are similarly far from each other

Indeed, most of the points will be neither inside the hypersphere (as we have seen) nor near the corners, but at distance

$$\sqrt{\frac{d}{3}} \pm \frac{2}{\sqrt{45d}}$$



Datawow, 2020

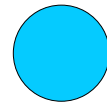
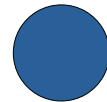
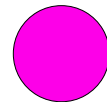
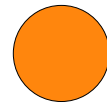
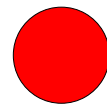
Wikipedia: Curse of Dimensionality

Often, less dimensions are better

THE CURSE OF DIMENSIONALITY

Often, less dimensions are better.

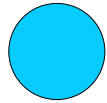
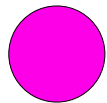
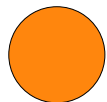
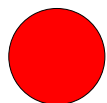
Suppose you have the following dataset of candy flavors represented in two dimensions. From this we can easily find **two clusters**, and learn that reddish candy are sweet and blueish candy are sour.



is_reddish	is_blueish	flavor
1	0	sweet
1	0	sweet
1	0	sweet
0	1	sour
0	1	sour
0	1	sour

THE CURSE OF DIMENSIONALITY

Now we add more dimensions ... but now **all points are equally far from each other**, there are basically **six clusters**, and we can just conclude that three candy are sweet and three candy are sour



is_red	is_orange	is_pink	is_navy	is_lblue	is_bblue	flavor
1	0	0	0	0	0	sweet
0	1	0	0	0	0	sweet
0	0	1	0	0	0	sweet
0	0	0	1	0	0	sour
0	0	0	0	1	0	sour
0	0	0	0	0	1	sour

Match-based similarity

Idea: to compute $\text{similarity}(u,v)$ ignore dimensions in which they are “too far apart”

- 1) Discretize each dimension into k_d equi-depth buckets
- 2) For two objects u, v , determine the dimensions in which they map to the same bucket
- 3) Compute L_p norm on those dimensions only

Match-based similarity (cont.)

$$PSelect(\overline{X}, \overline{Y}, k_d) = \left[\sum_{i \in S(\overline{X}, \overline{Y}, k_d)} \left(1 - \frac{|x_i - y_i|}{m_i - n_i} \right)^p \right]^{1/p}$$

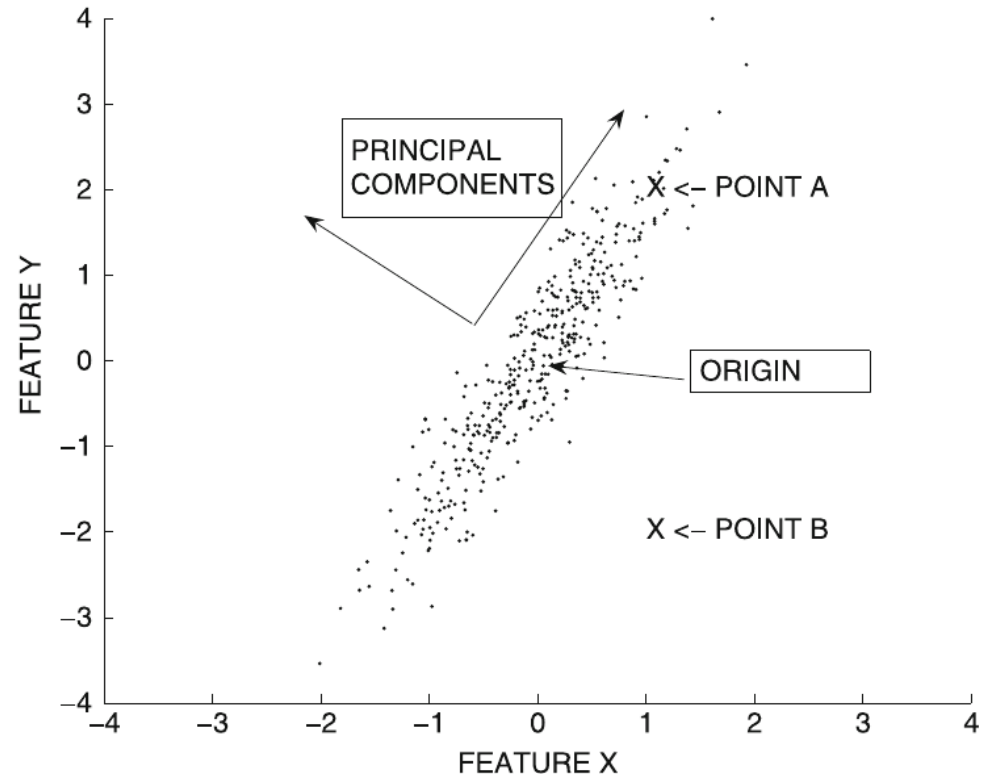
- $S(\overline{X}, \overline{Y}, k_d)$ is the set of features for which \overline{X} and \overline{Y} map to the same bucket
- m_i, n_i are the max and min value of that bucket
- $k_d \propto d$ achieves a constant level of contrast in high dimensions for certain data distributions

Distances and orientation

Useful distances, in general, depend on data distributions

Points A and B are equidistant from the origin

However, point A should be considered closer to the origin than point B (think of a perfectly circular cloud of points)

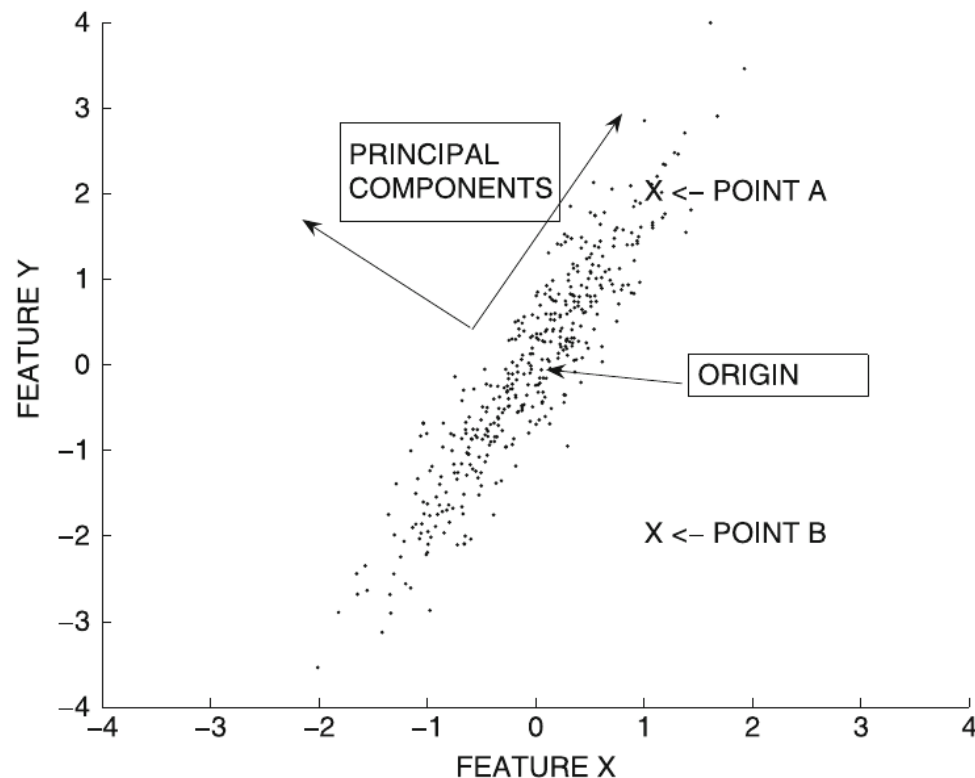


Useful distances, in general, depend on data distributions (cont.)

The Mahalanobis distance, with Σ covariance matrix

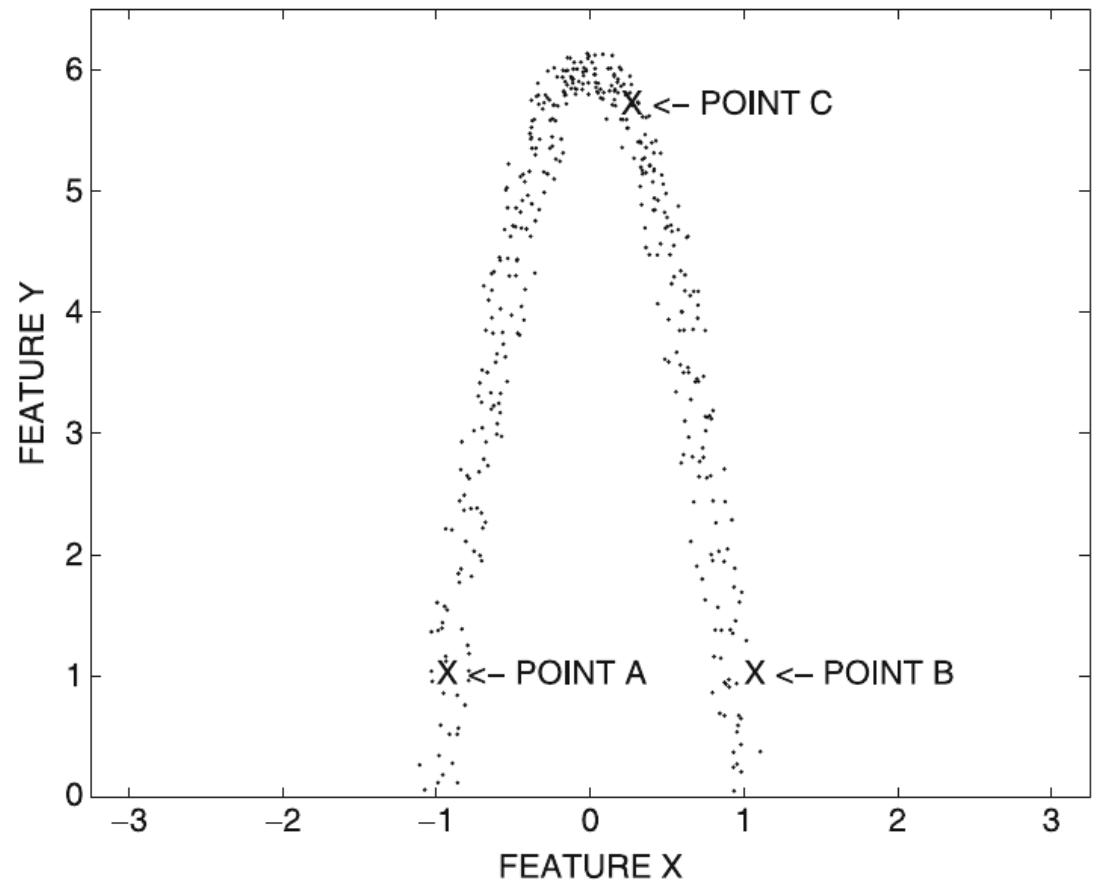
$$Maha(\bar{X}, \bar{Y}) = \sqrt{(\bar{X} - \bar{Y})\Sigma^{-1}(\bar{X} - \bar{Y})^T}.$$

is equivalent to applying PCA, dividing each coordinate by the standard deviation of that feature and computing Euclidean distance



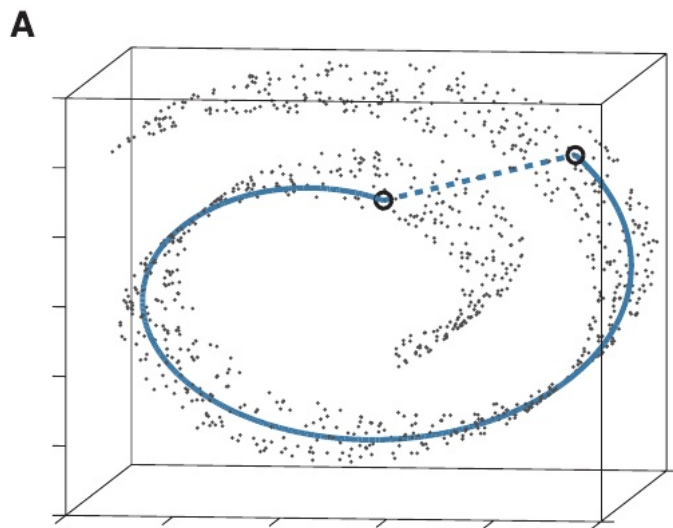
Non-linear distributions

Which point would you consider as closer to A?

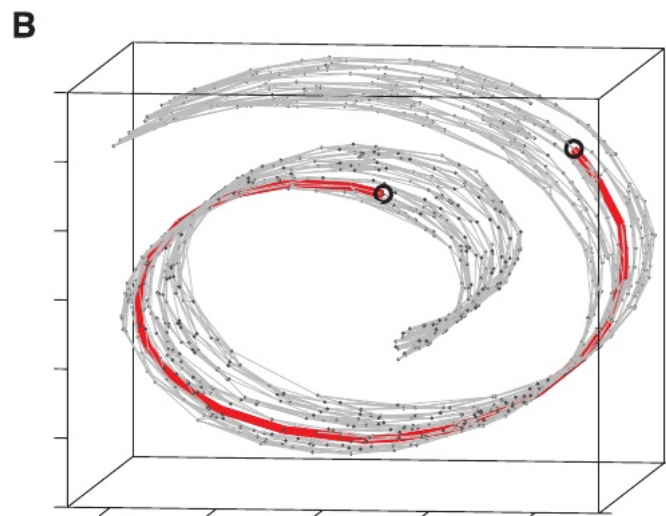


(Blackboard collaborate poll)

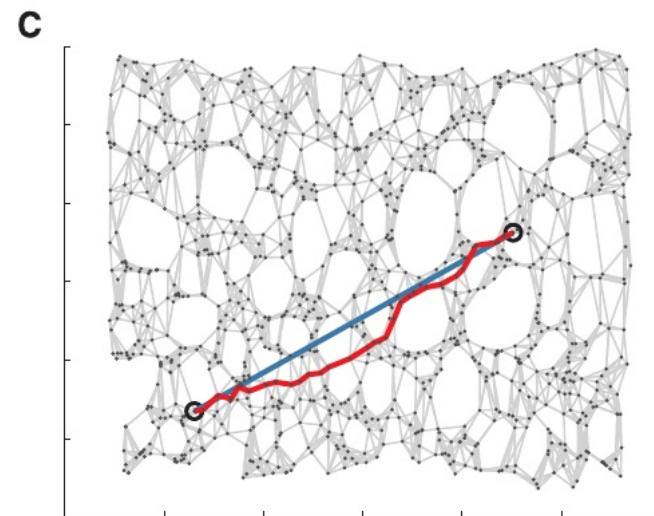
ISOMAP (general idea)



Original data

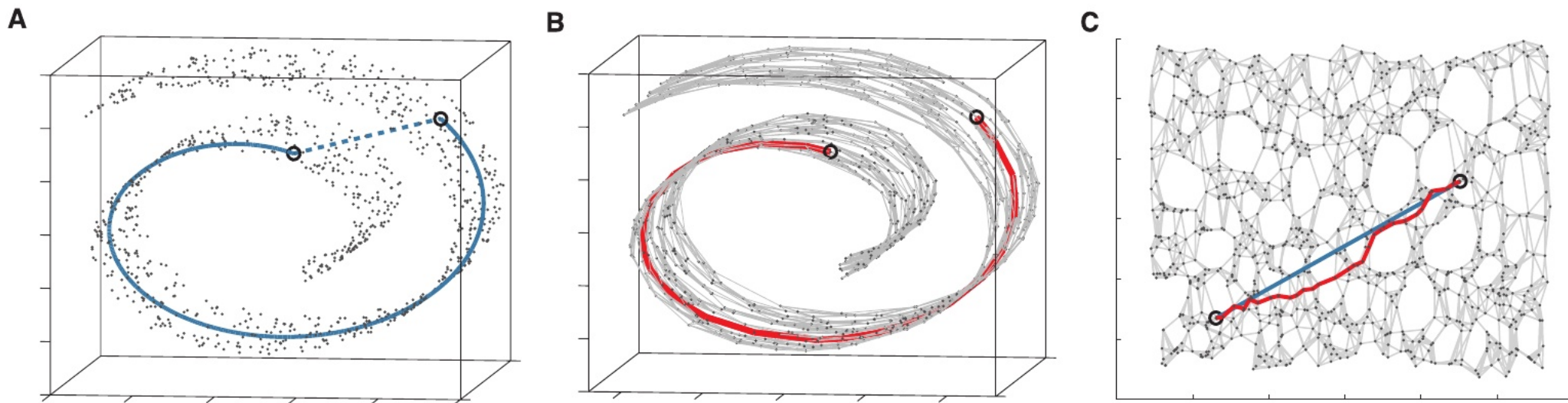


Nearest neighbors graph



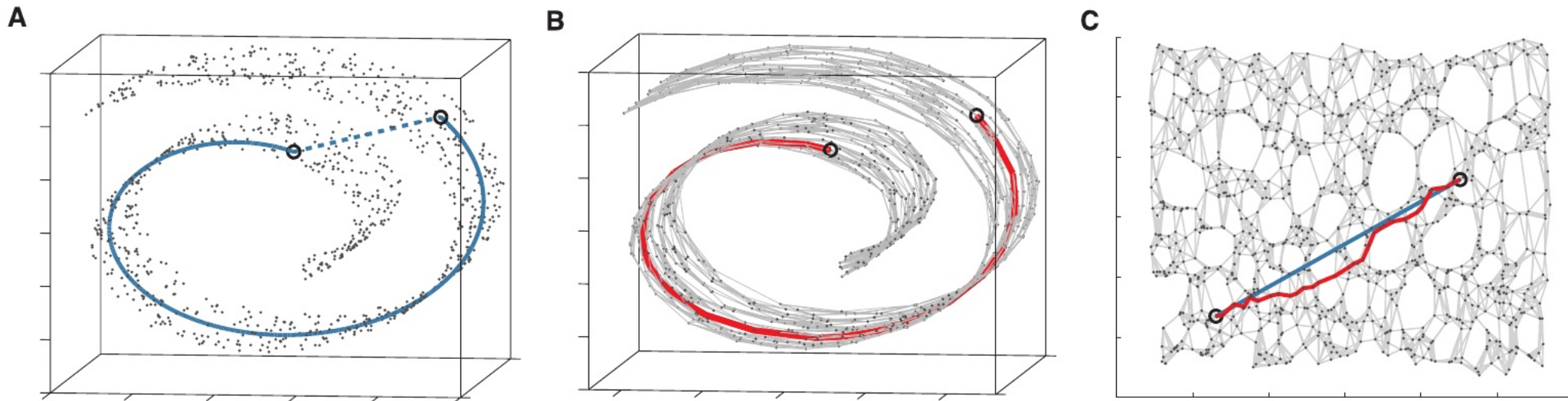
Graph projection

ISOMAP (1/3)



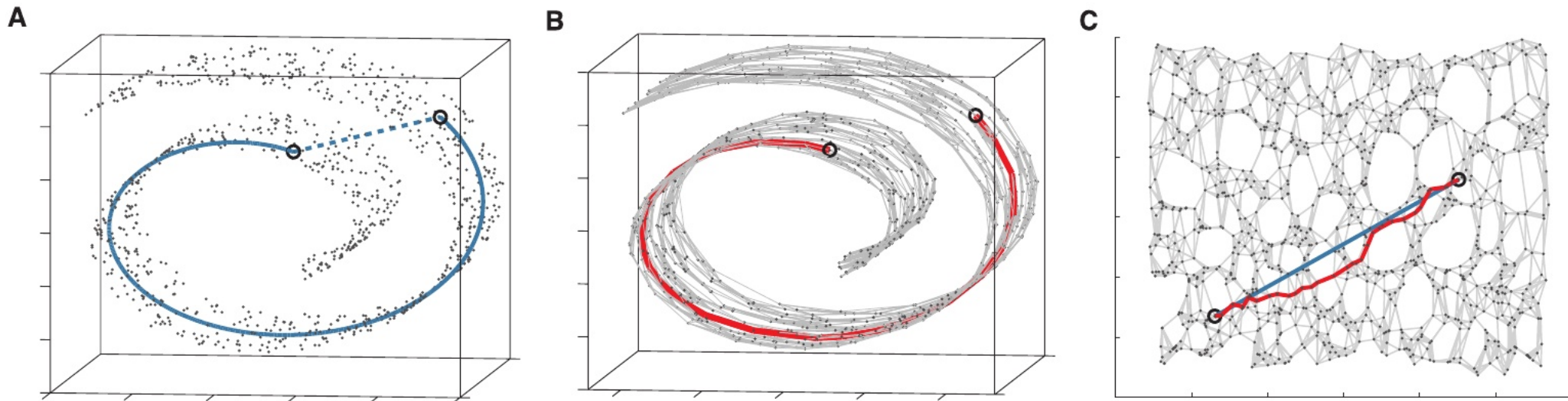
The first step is to connect each point to its k nearest neighbors (here $k=7$)

ISOMAP (2/3)



Now, shortest path or *geodesic* distances
can be computed on the graph
(red color)

ISOMAP (3/3)

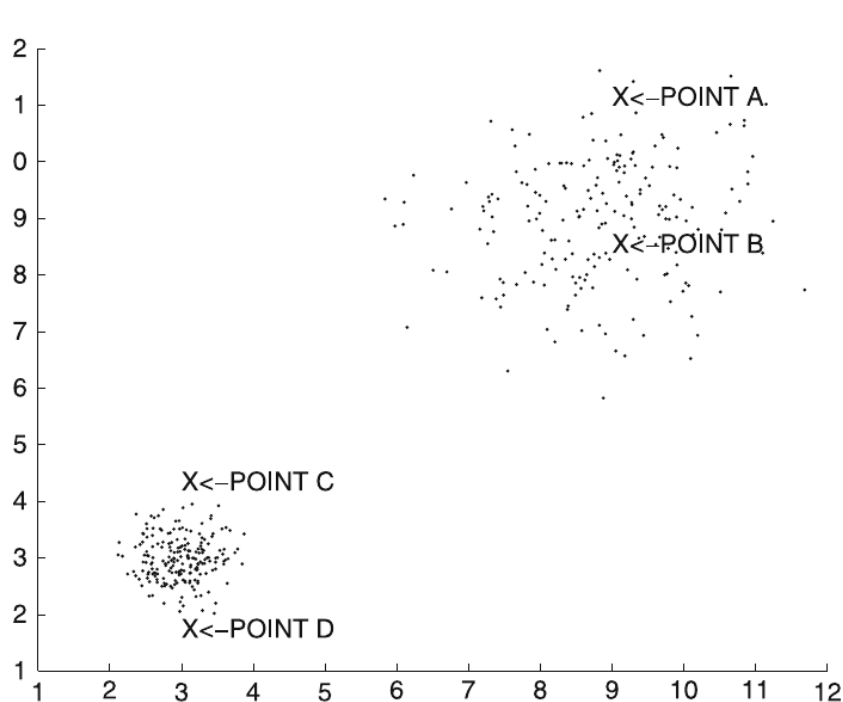


It is, however, more effective to project the graph and compute Euclidean distances in the projected graph (blue color)

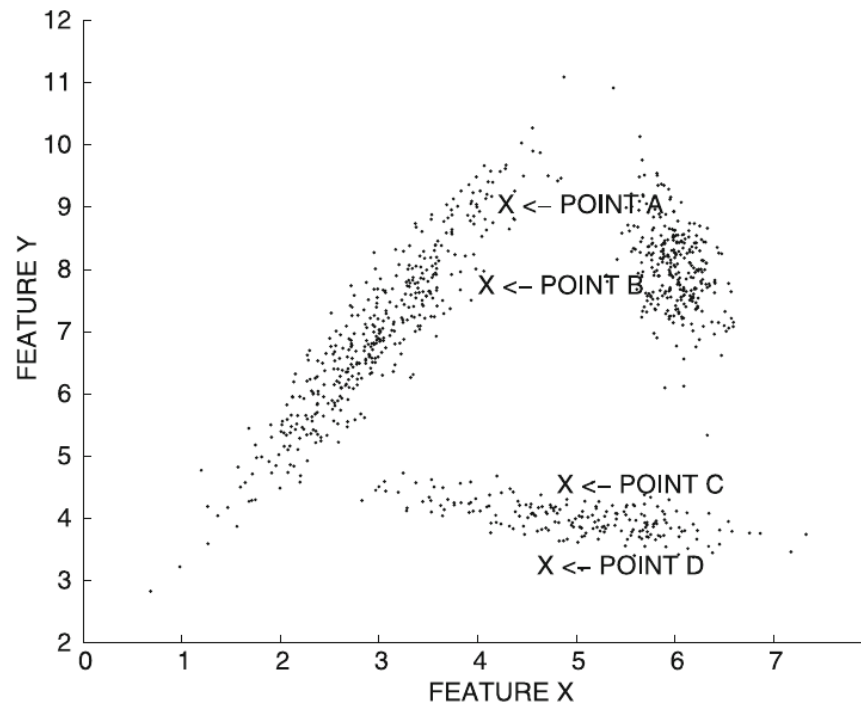
Local variations

Which distance should be larger? A-B or C-D?

Which distance should be larger? A-B or C-D?



(a) local density variation



(b) local orientation variation

Solution for local variations

- Partition the data into a set of local regions
 - (Nontrivial, which distance to use?)
- For any pair of objects, determine the most relevant region for the pair
- If they belong to the same region
 - Compute the pairwise distances using the local statistics of that region
 - E.g., local Mahalanobis distance
- If they belong to different regions
 - Global statistics or averaged statistics

Summary

Things to remember

- Distance/similarity is a key component of many data mining algorithms
- Sensitive to dimensionality
 - In many cases, having less dimensions is better
- Sensitive to local nature of data distribution

Exercises for TT06-TT07

- **Data Mining, The Textbook (2015) by Charu Aggarwal**
 - Exercises 3.9 on similarity measures
- **Introduction to Data Mining 2nd edition (2019) by Tan et al.**
 - Exercises 2.6 → 14-28
- **Mining of Massive Datasets 2nd edition (2014) by Leskovec et al.**
 - Exercises 3.5.7 on distance measures
- **Data Mining Concepts and Techniques, 3rd ed. (2011) by Han et al.**
 - Exercises 2.6 → 2.5-2.8