Similarity:

Numerical Data

Mining Massive Datasets

Materials provided by Prof. Carlos Castillo — https://chato.cl/teach
Instructor: Dr. Teodora Sandra Buda — https://tbuda.github.io/

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

[Hays and Efros, SIGGRAPH 2007]

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

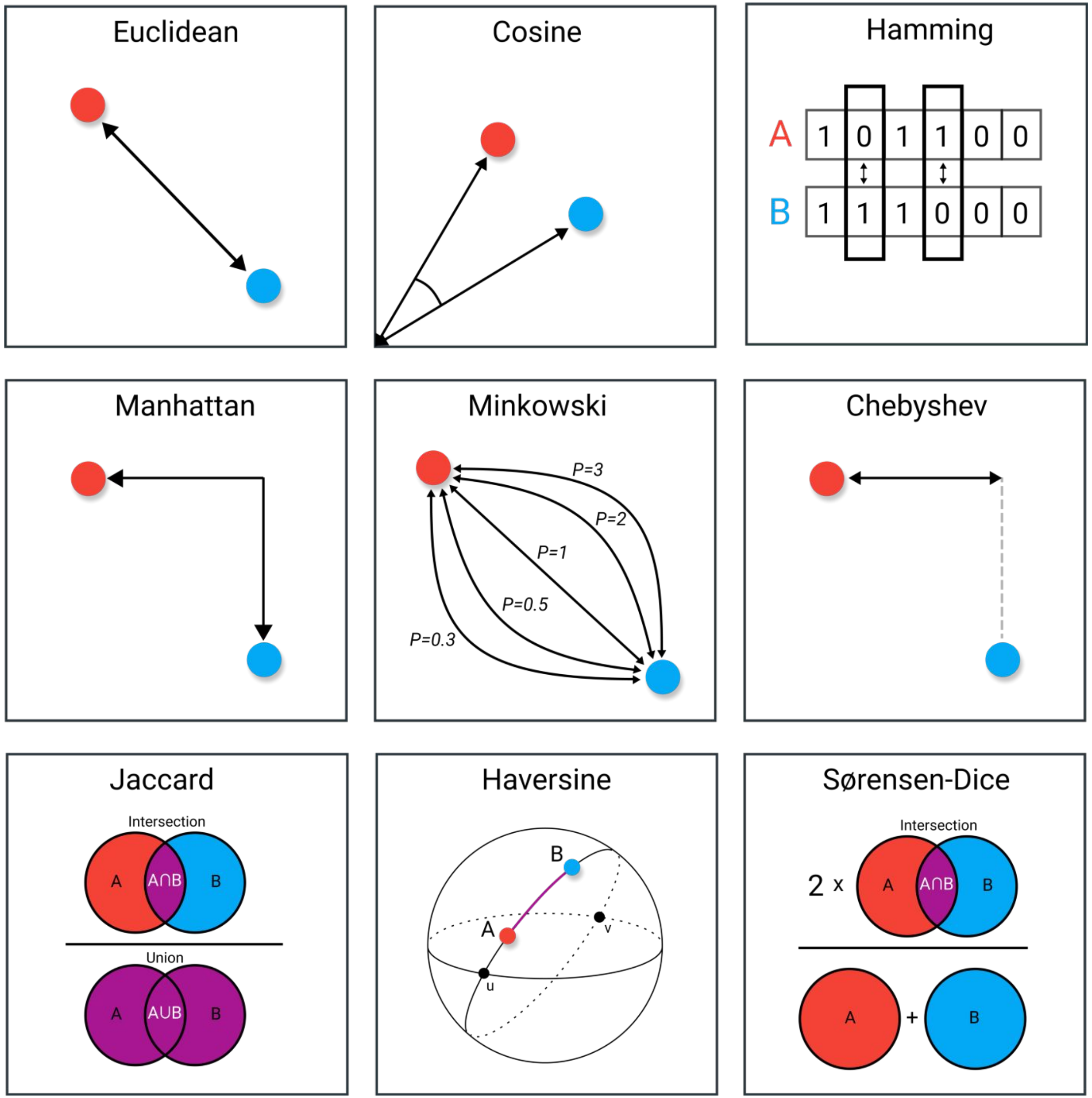
- Given two objects u and v, determine the value of:
  - similarity(u,v) and distance(u,v) *Often one is defined in terms of the other*
- **Similar** objects should have large similarity and small distance
- **Dissimilar** objects should have small similarity and large distance
- We can use closed-form 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	$d = \|x - y\|$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

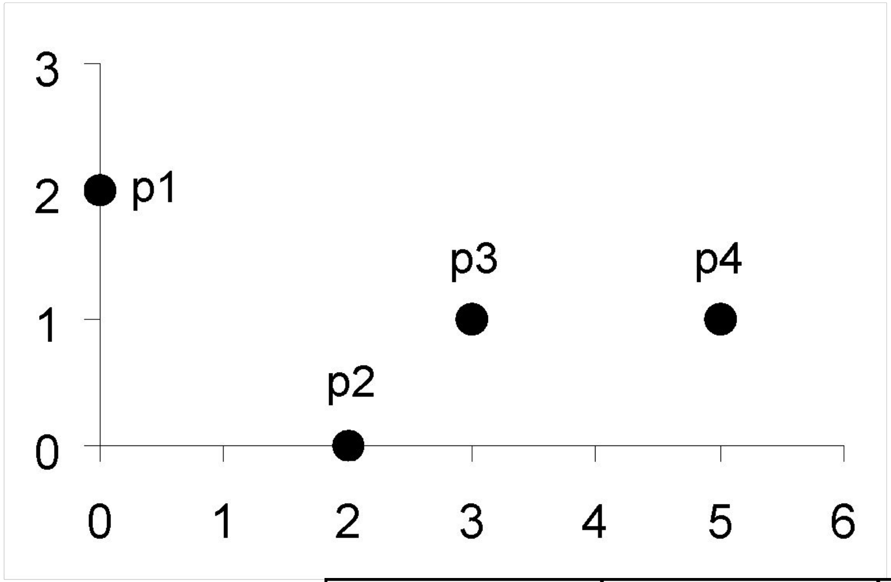
Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 2)

Some distance measures



https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa

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

Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 2)

L_p norm, $p \geq 1$

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

- $p=1$: 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

Generalized L_p norm, $p \geq 1$

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

- Useful when some features are more important than others
- 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

Answer

python

- Compute L_1, L_2, L_∞ norm between:
  - (22, 1, 42, 10)
  - (20, 0, 36, 8)

```
import numpy as np
```

```
x = [22, 1, 42, 10]
y = [20, 0, 36, 8]
```

```
np.linalg.norm(np.subtract(x,y), ord=1)
```

```
11.0
```

```
np.linalg.norm(np.subtract(x,y), ord=2)
```

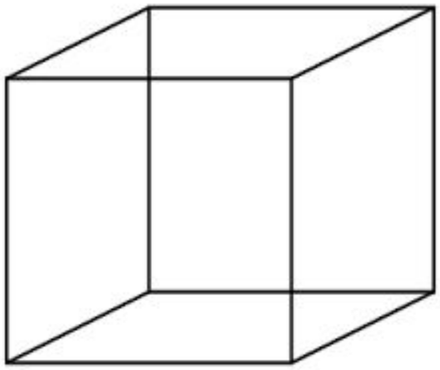
```
6.708203932499369
```

```
np.linalg.norm(np.subtract(x,y), ord=np.inf)
```

```
6.0
```

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 $(2r)^d$

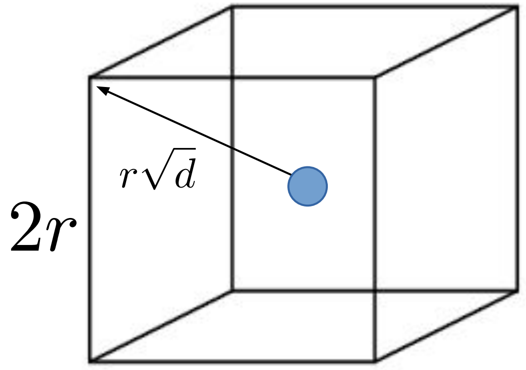


When the dimensionality is high, all points are similarly far from each other

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

That distance increases without bound as the dimensionality increases!

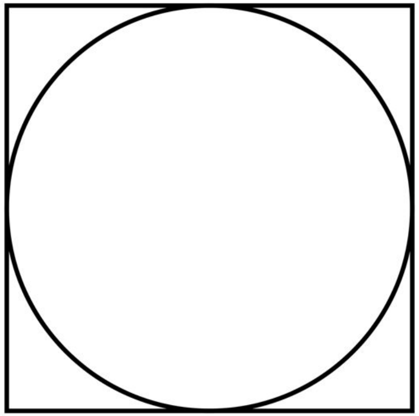
Now, let us imagine a hypersphere of radius *r* inside the hypercube ...

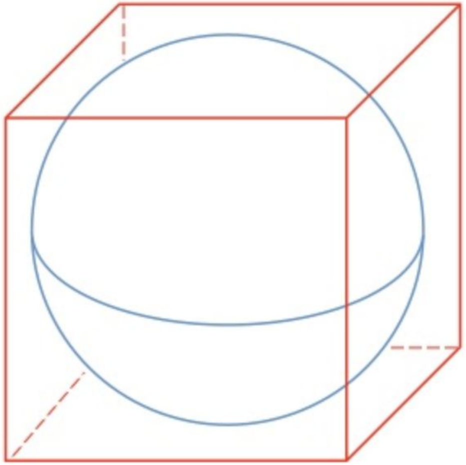


When the dimensionality is high, all points are similarly far from each other

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



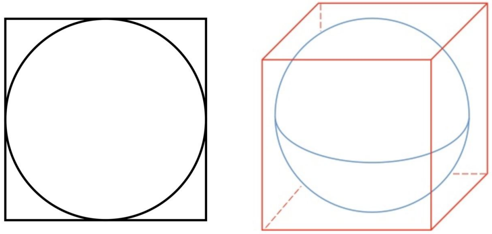


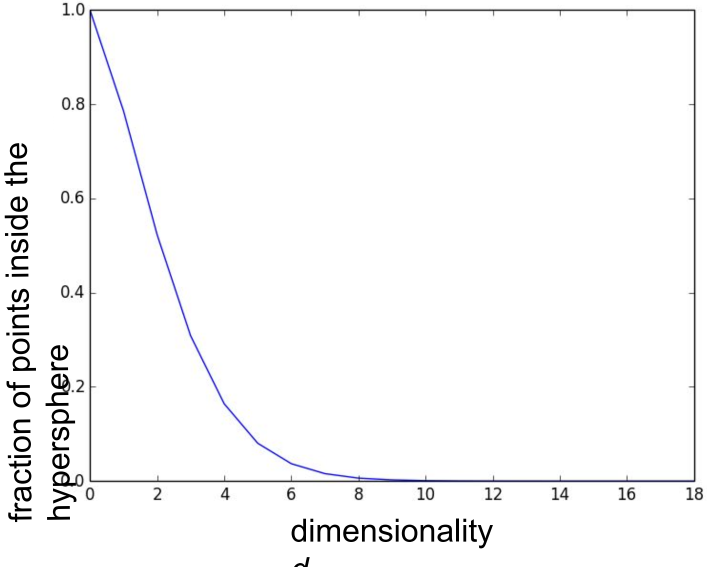
Datawow, 2020

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

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





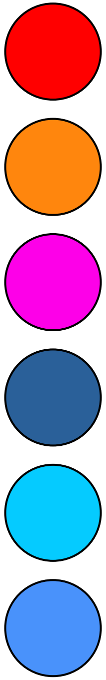
[Datawow, 2020]

Wikipedia: [Curse of Dimensionality]

Often, less dimensions are better

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

[Tony Yiu, 2019]

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_ora	is_pnk	is_nvy	is_lbl	is_blu	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

[Tony Yiu, 2019]

Match-based similarity

Idea: to compute 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.)

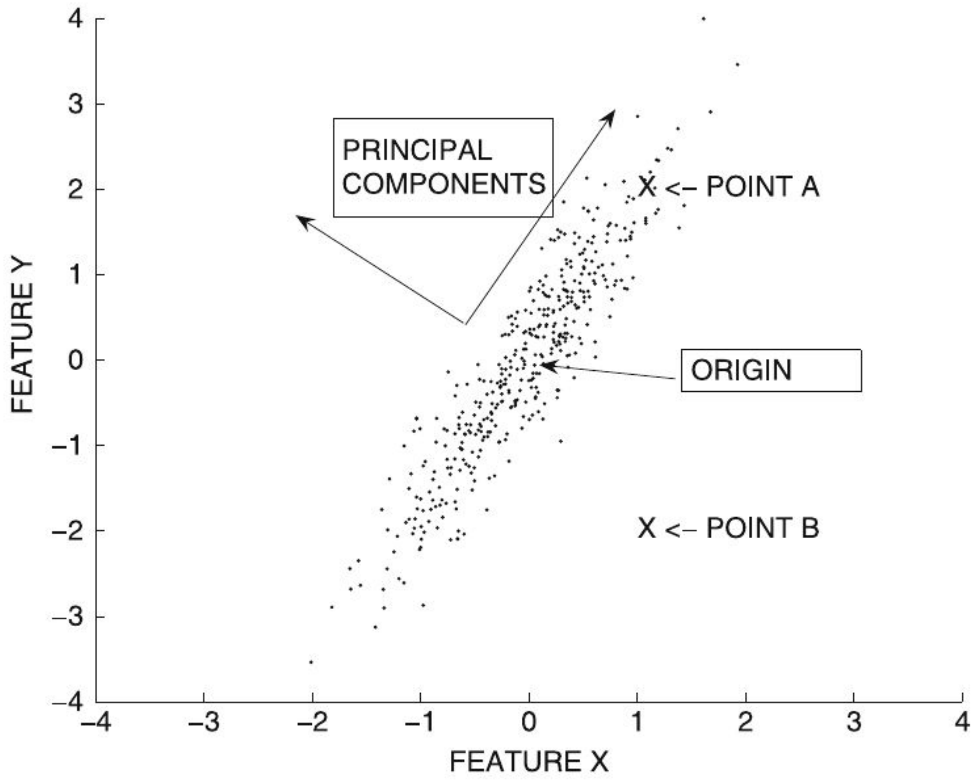
$$PSelect(\overline{X}, \overline{Y}, k_d) = \left[\sum_{i \in S(\overline{X}, \overline{Y}, k_d)} \left(1 - \frac{|x_i - y_i|}{m_i - n_i} \right)^p \right]^{1/p}$$

- $S(X, Y, k_d)$ is the set of features for which X and 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

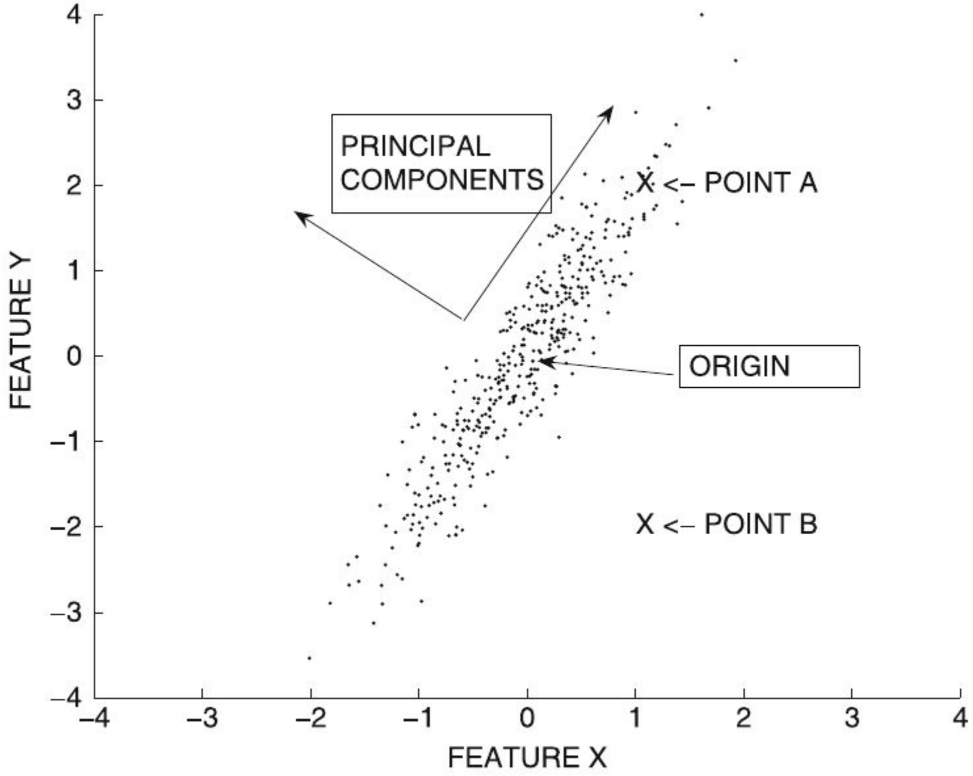
Points A and B are equidistant from the origin
However, point A should be considered closer to the origin than point B (think of a perfectly circular cloud of points)



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

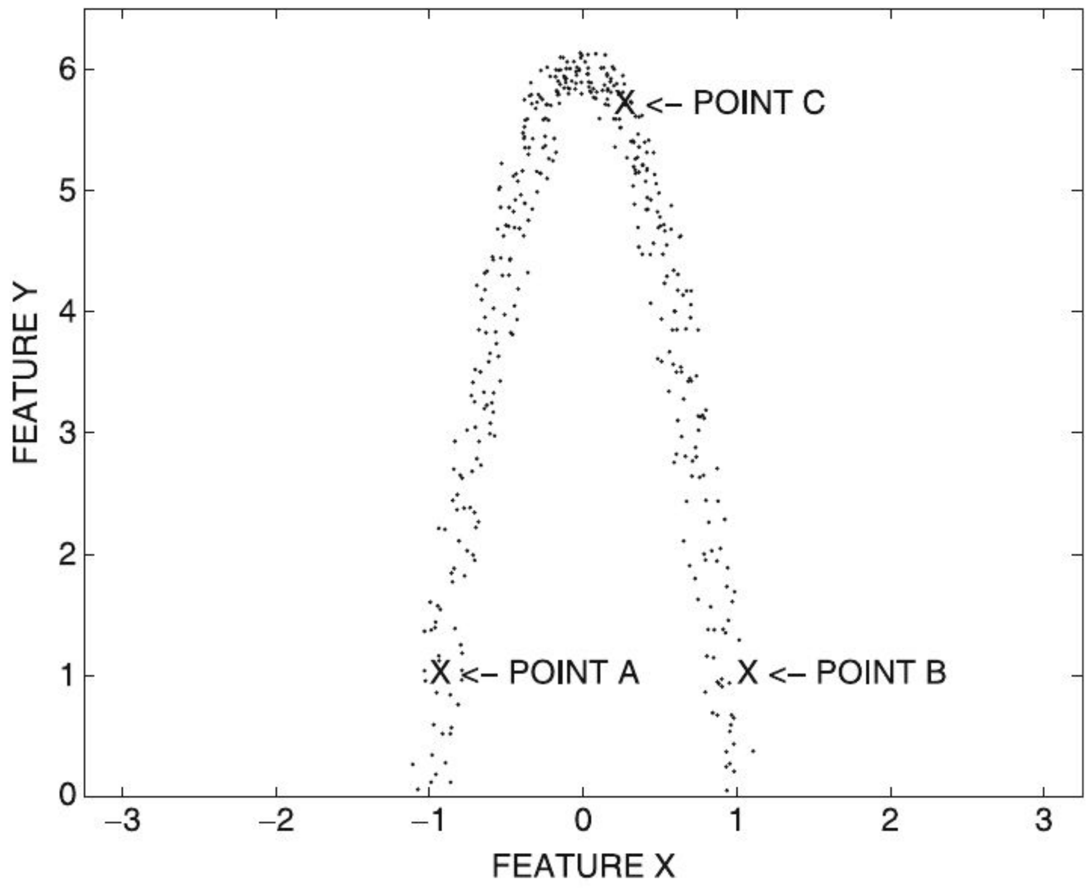
The Mahalanobis distance, with Σ covariance matrix

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



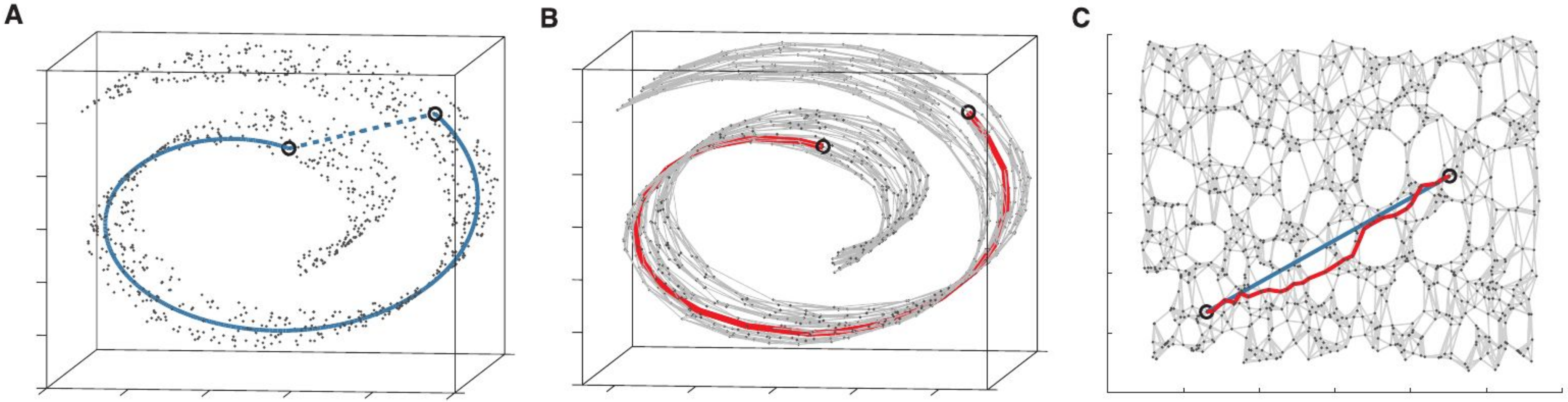
Non-linear distributions

Which point would you consider as closer to A?



(Blackboard collaborate poll)

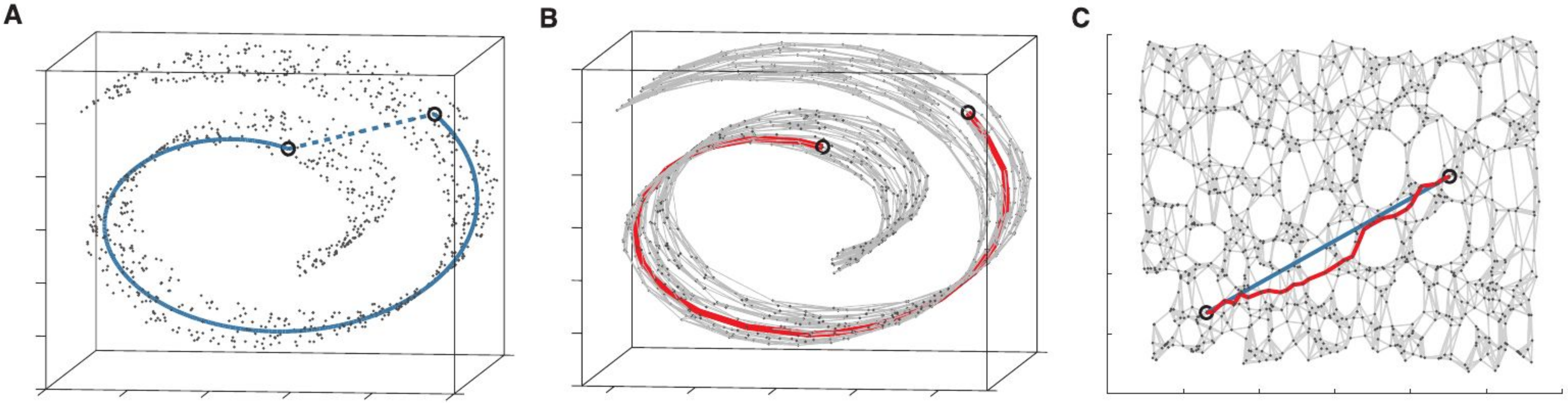
ISOMAP (general idea)



Original data | Nearest neighbors graph | Graph projection

Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." Science 2000.

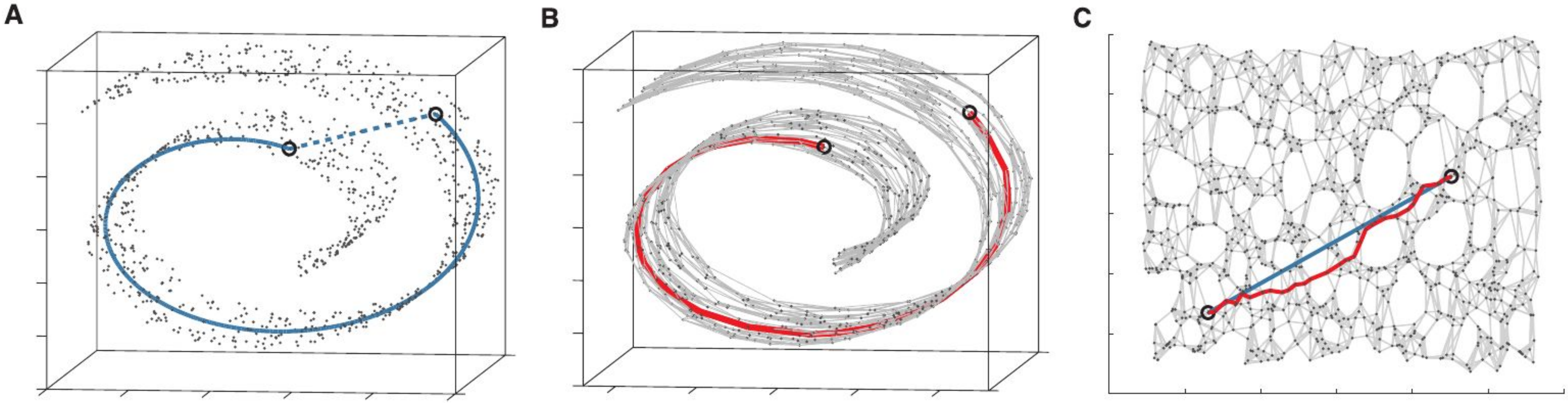
ISOMAP (1/3)



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

Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." Science 2000.

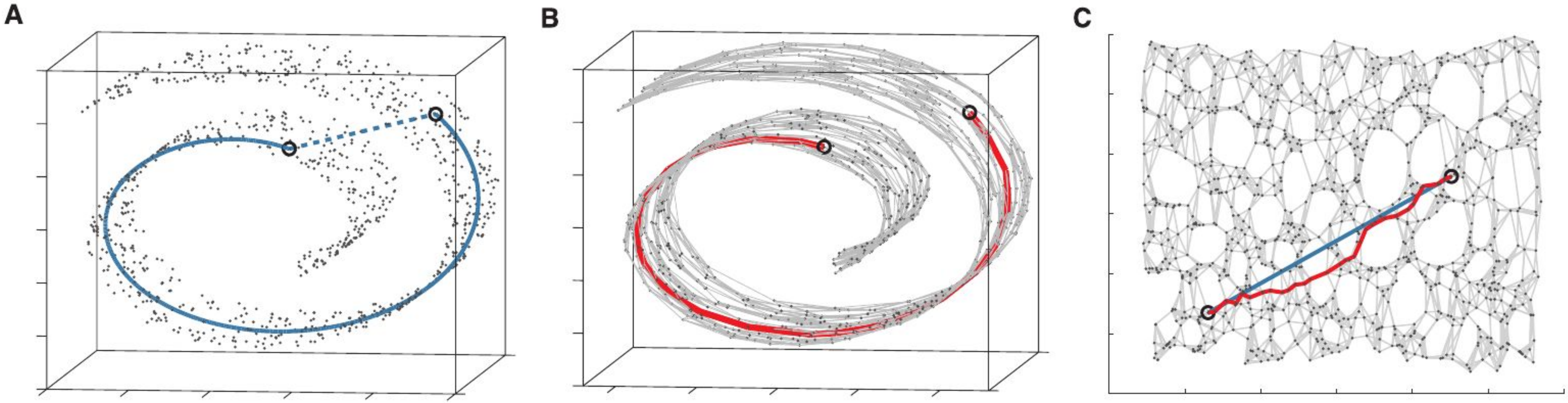
ISOMAP (2/3)



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

Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." Science 2000.

ISOMAP (3/3)



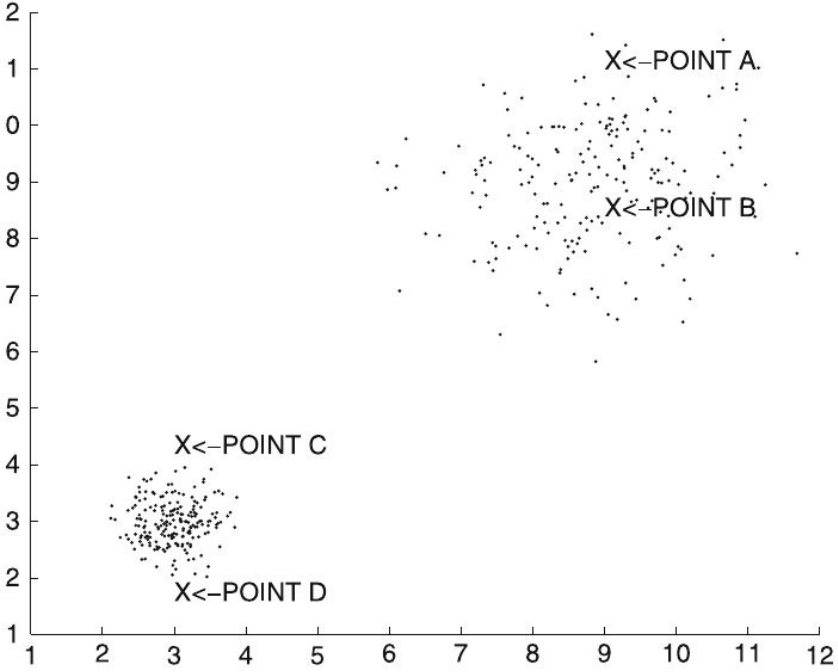
It is, however, more effective to project the graph and compute Euclidean distances in the projected graph (blue color)

Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." Science 2000.

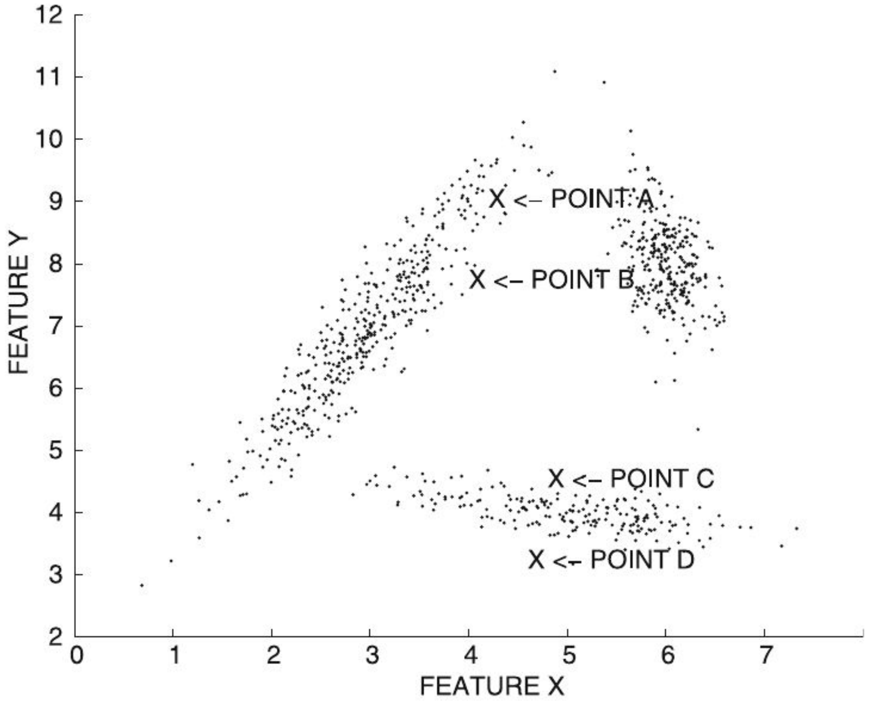
Local variations

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

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

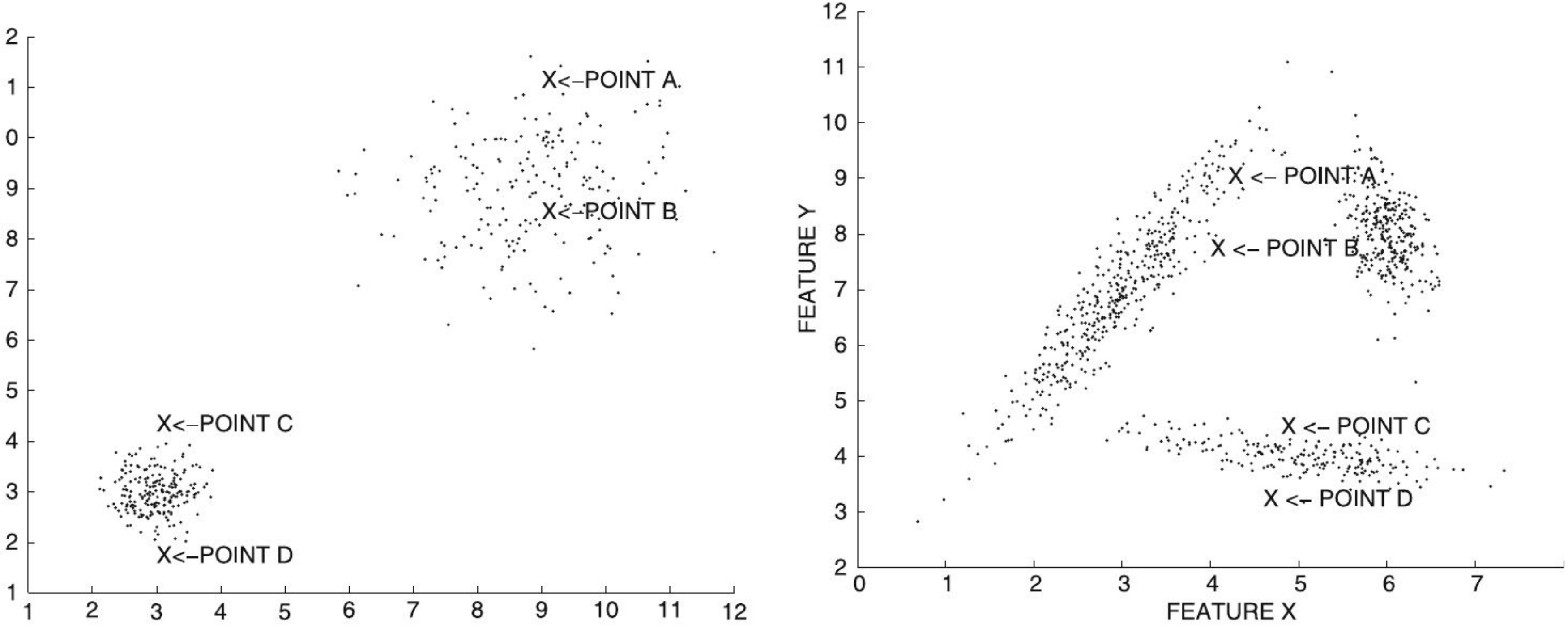


(a) local density variation



(b) local orientation variation

(Answer: in both cases C-D should be larger than A-B)



(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