Similarity

Mining Massive Datasets

Carlos Castillo

Topic 03

upf. Universitat Pompeu Fabra Barcelona

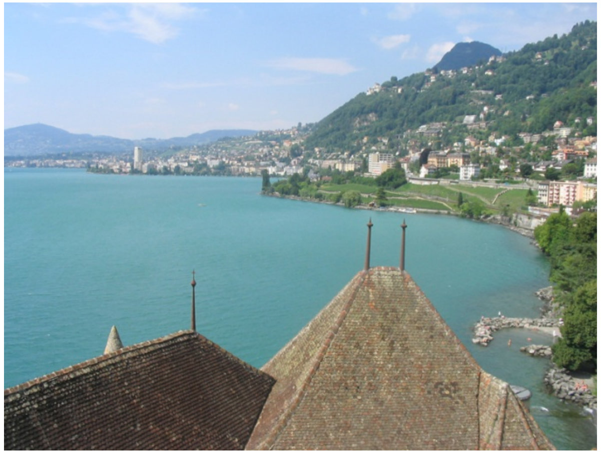
Main Sources

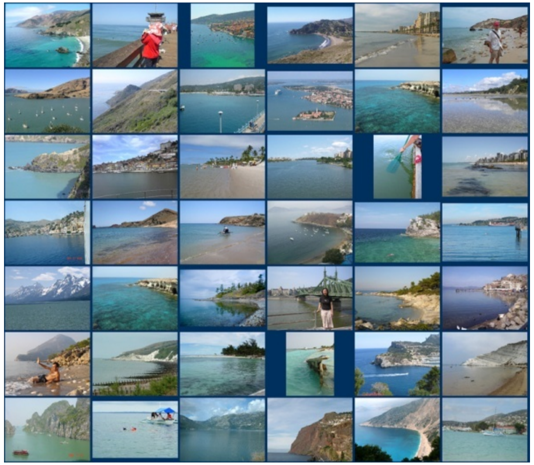
- Data Mining, The Textbook (2015) by Charu Aggarwal (Chapter 3) + slides by Lijun Zhang
- Data Mining Concepts and Techniques, 3rd edition (2011) by Han et al. (Section 2.4)
- Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 2)
- Mining of Massive Datasets 2nd edition (2014) by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

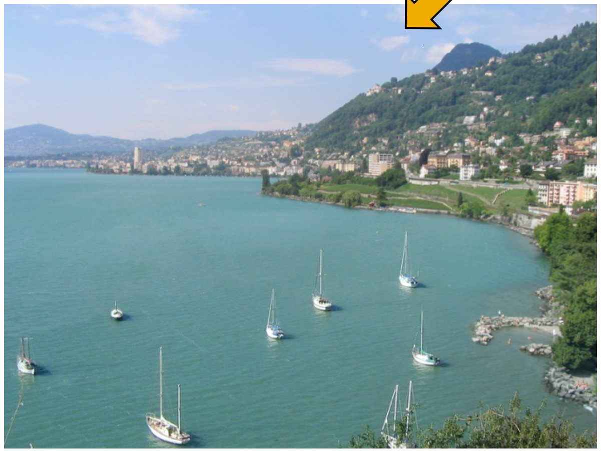
Example: scene completion

Mining of Massive Datasets 2nd edition (2014) by Leskovec et al. (Chapter 3)

Scene completion problem

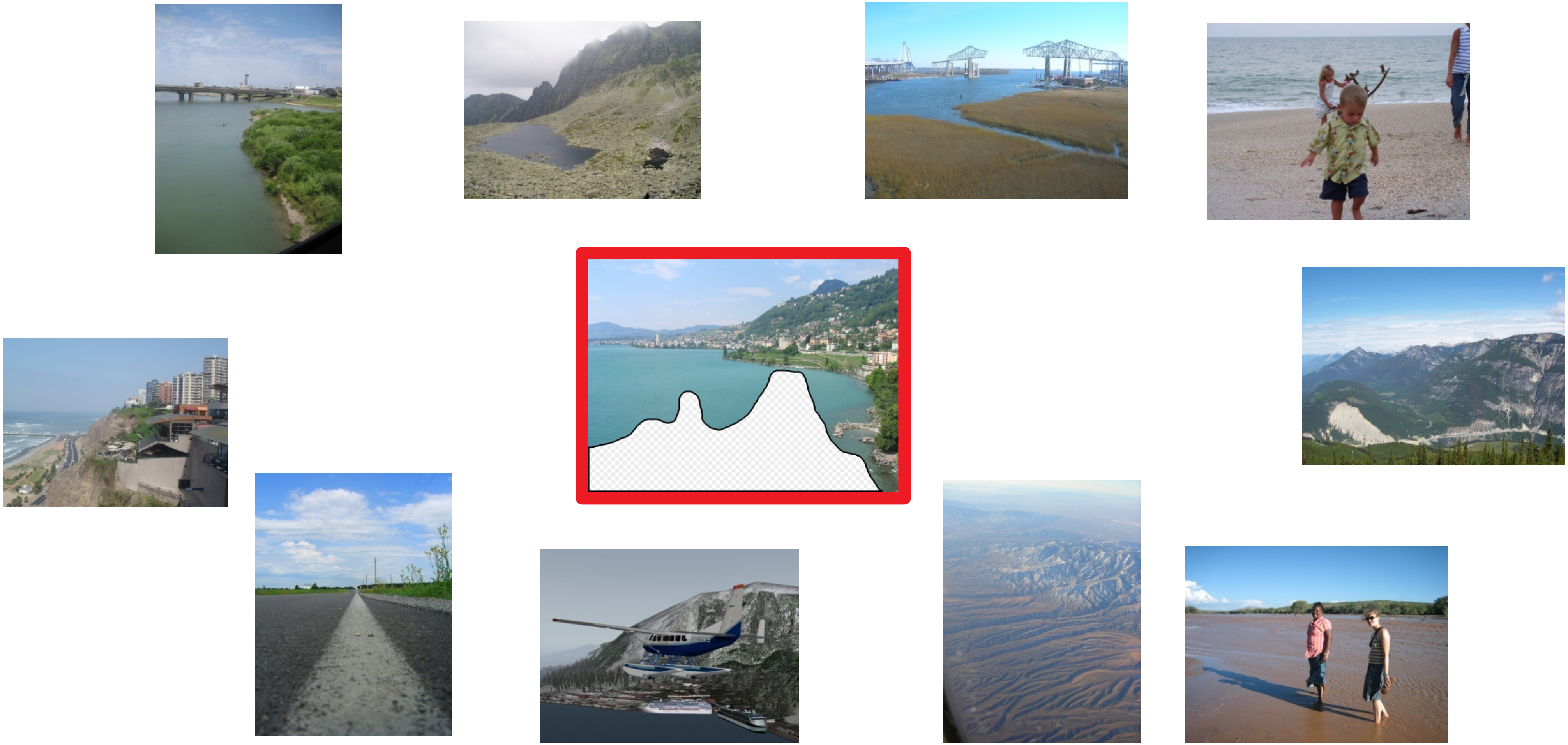




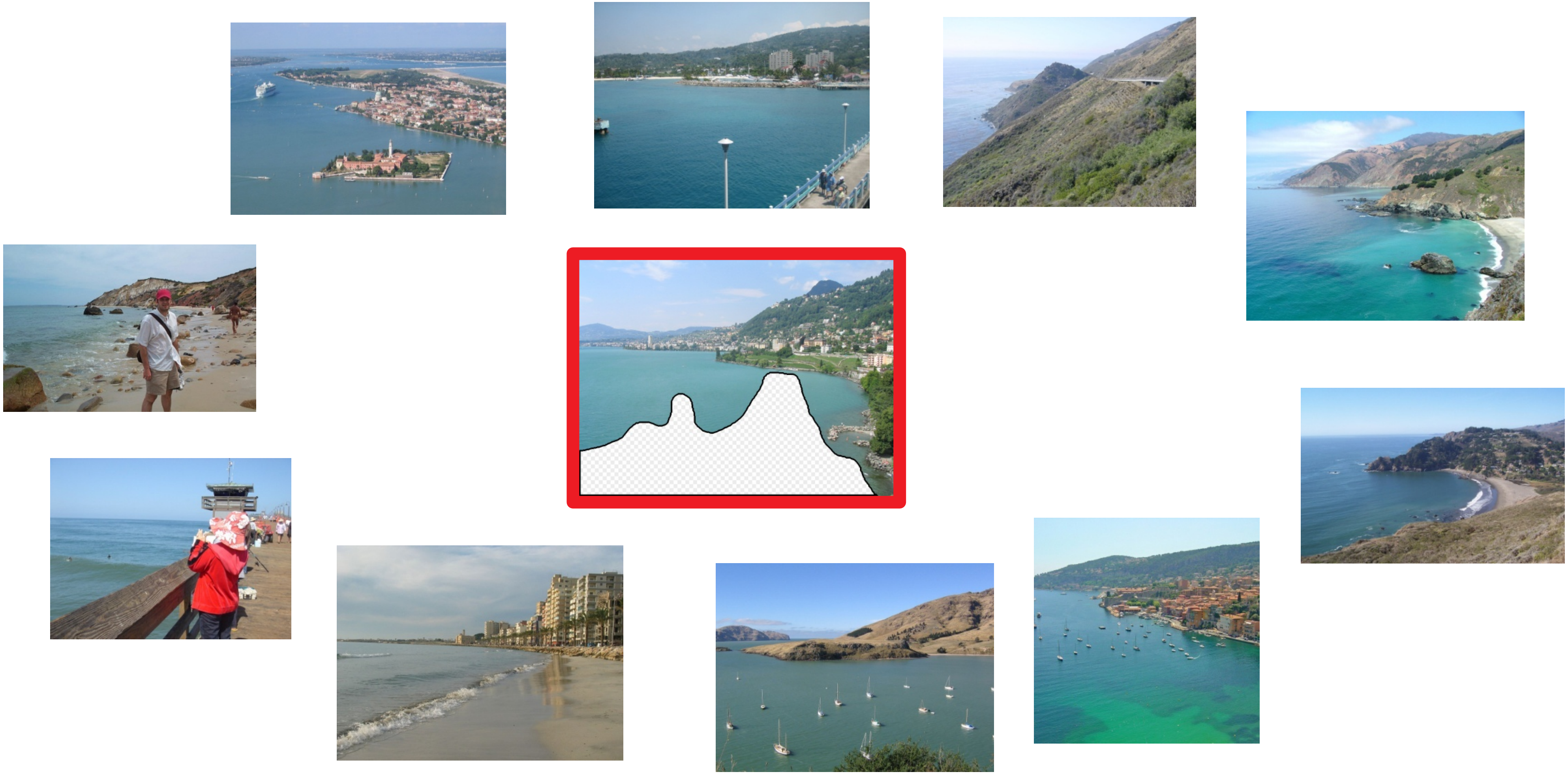


[Hays and Efros, SIGGRAPH 2007]

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
    - Products with similar customer sets
  - Images with similar features
  - Users who visited similar websites

Similarity computation task

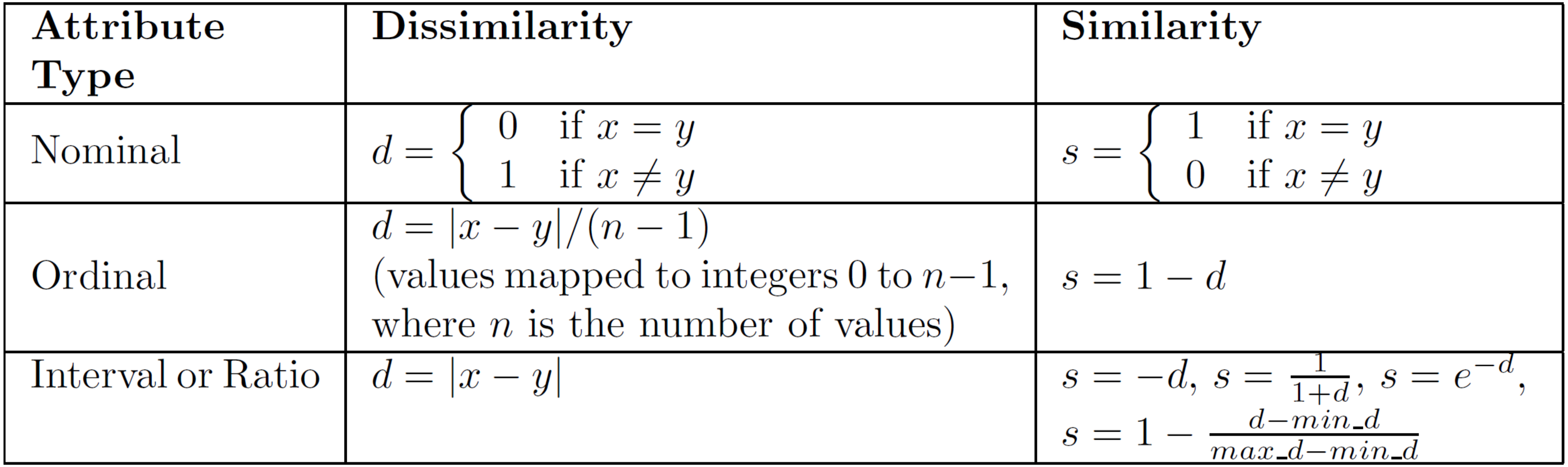
- Given two objects u and v, determine the value of:

  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
- Closed-form functions (e.g., euclidean distance) or algorithm

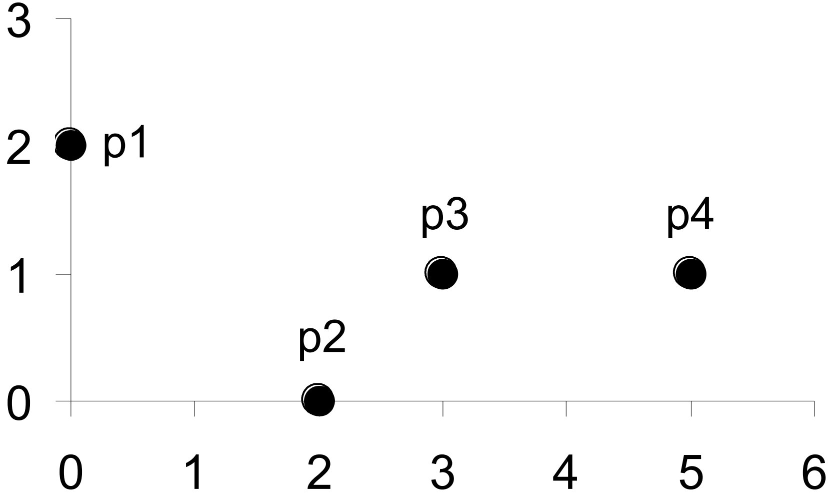
Simple single-attribute similarity



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	$d = \lvert x - y \rvert / (n - 1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

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

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)

The Curse of Dimensionality

L_p norm, p ≥ 1

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

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

Try it!

- Compute L_1, L_2, L_∞ norm between:

    (22, 1, 42, 10)

    (20, 0, 36, 8)

Generalized L_p norm, p ≥ 1

- Useful when some features are more important than others
- E.g., in credit scoring, salary is more important than gender
- a_i are domain-specific non-negative coefficients

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

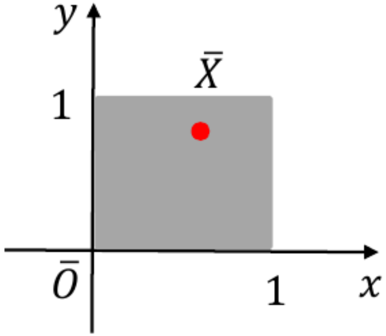
THE CURSE OF DIMENSIONALITY

- When the dimensionality is high, all points are at similar L_p distances from each other
- Example:

A unit cube of dimensionality d in the nonnegative quadrant

$\bar{X}$ is a random point in the cube

Manhattan distance between $\bar{O}$ and $\bar{X}$



THE CURSE OF DIMENSIONALITY

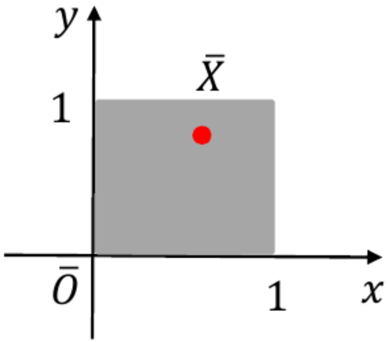
- Example (cont.):

Manhattan distance between $\bar{O}$ and $\bar{X}$

$$Dist(\overline{O},\overline{X}) = \sum_{i=1}^{d}(Y_i - 0).$$

where $\bar{X} = [Y_1, \dots, Y_d]$

$Dist(\bar{O},\bar{X})$ is a random variable



- ✓ Since $\bar{X}$ is a random variable
- ✓ Mean is $\mu = d/2$
- ✓ Standard deviation $\sigma = \sqrt{d/12}$

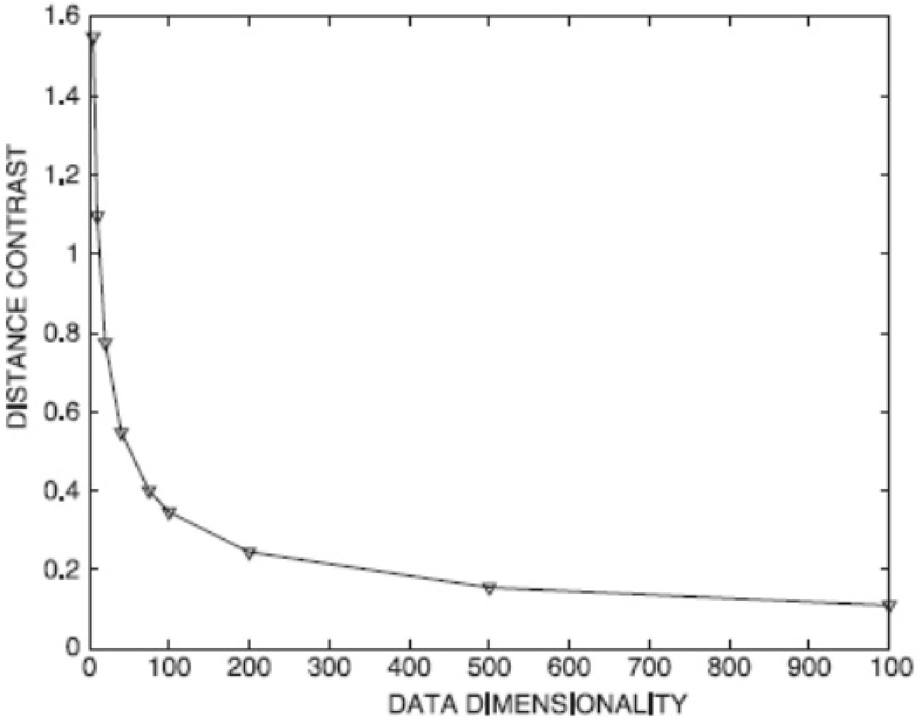
THE CURSE OF DIMENSIONALITY

Applying Chebyshev's inequality

$$\Pr(|X-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$

With a probability at least 8/9

$$Dist(\bar{O},\bar{X}) \in [\underbrace{\mu-3\sigma}_{D_{min}}, \underbrace{\mu+3\sigma}_{D_{max}}]$$

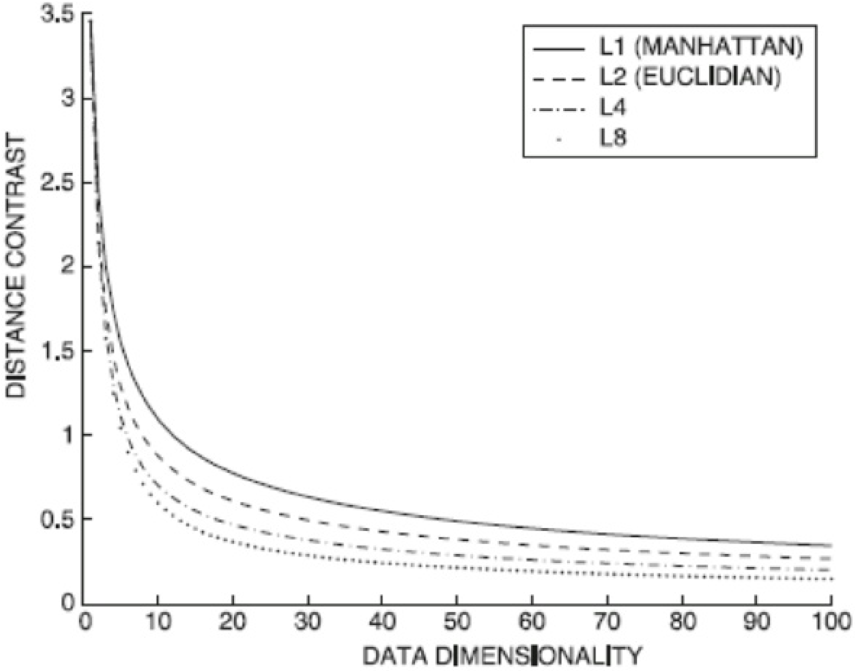


Contrast

$$\text{Contrast}(d) = \frac{D_{max} - D_{min}}{\mu} = \sqrt{12/d}.$$

Irrelevant features

- Many features are probably irrelevant for your purposes, specially in high-dimensional data
- L_p norm suffers from irrelevant features
- Contrast worsens for large p



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.)

$$PSelect(\overline{X},\overline{Y},k_d) = \left[\sum_{i\in S(\overline{X},\overline{Y},k_d)} \left(1 - \frac{|x_i - y_i|}{m_i - n_i}\right)^p\right]^{1/p} \in [0, S(\bar{X},\bar{Y},k_d)]$$

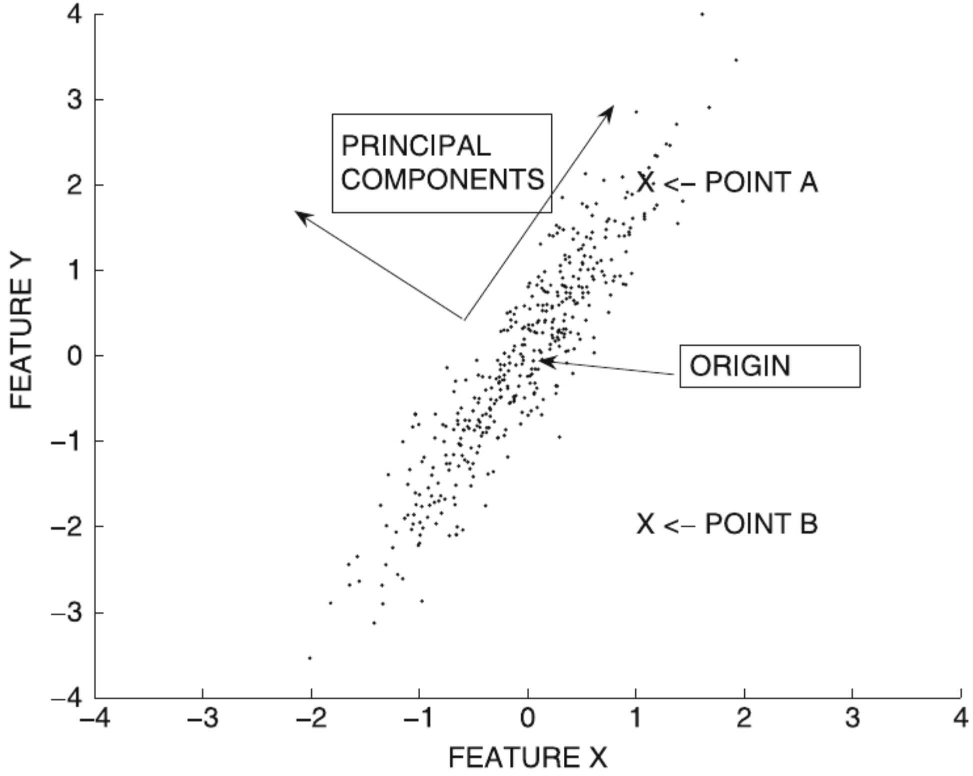
- $S(\overline{X}, \overline{Y}, k_d)$ is the set of features for which $\overline{X}$ and $\overline{Y}$ map to the same bucket
- m_i, n_i are the max and min value of that bucket
- $k_d \propto d$ achieves a constant level of contrast in high dimensions for certain data distributions

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)

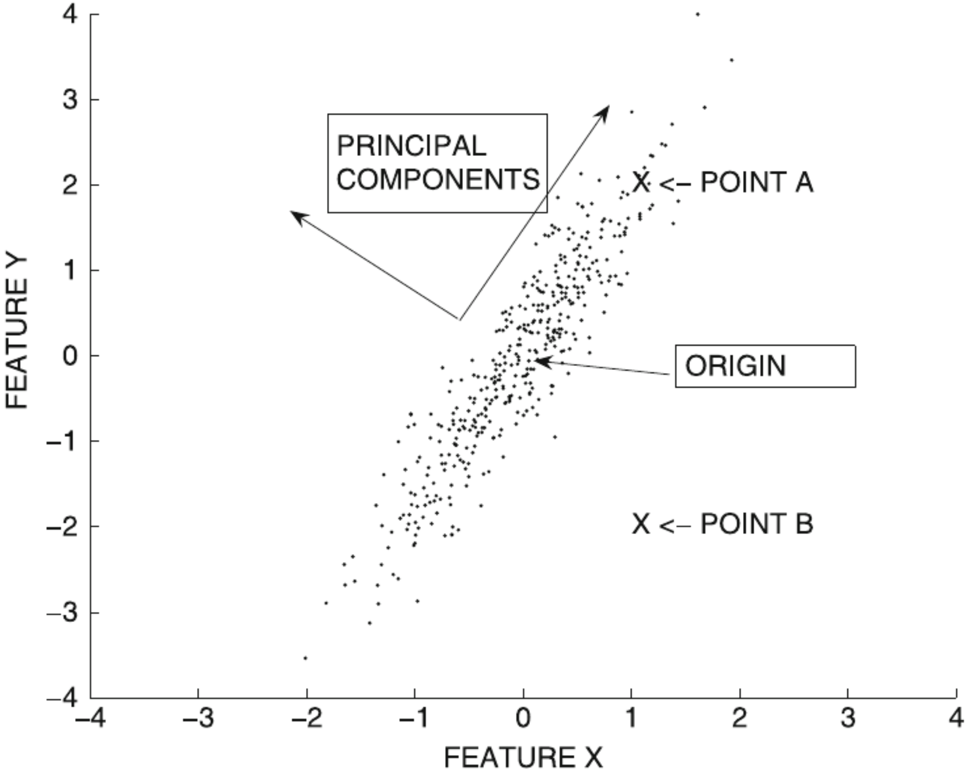


Useful distances, in general, depend on data distributions (cont.)

The Mahalanobis distance, with Σ covariance matrix

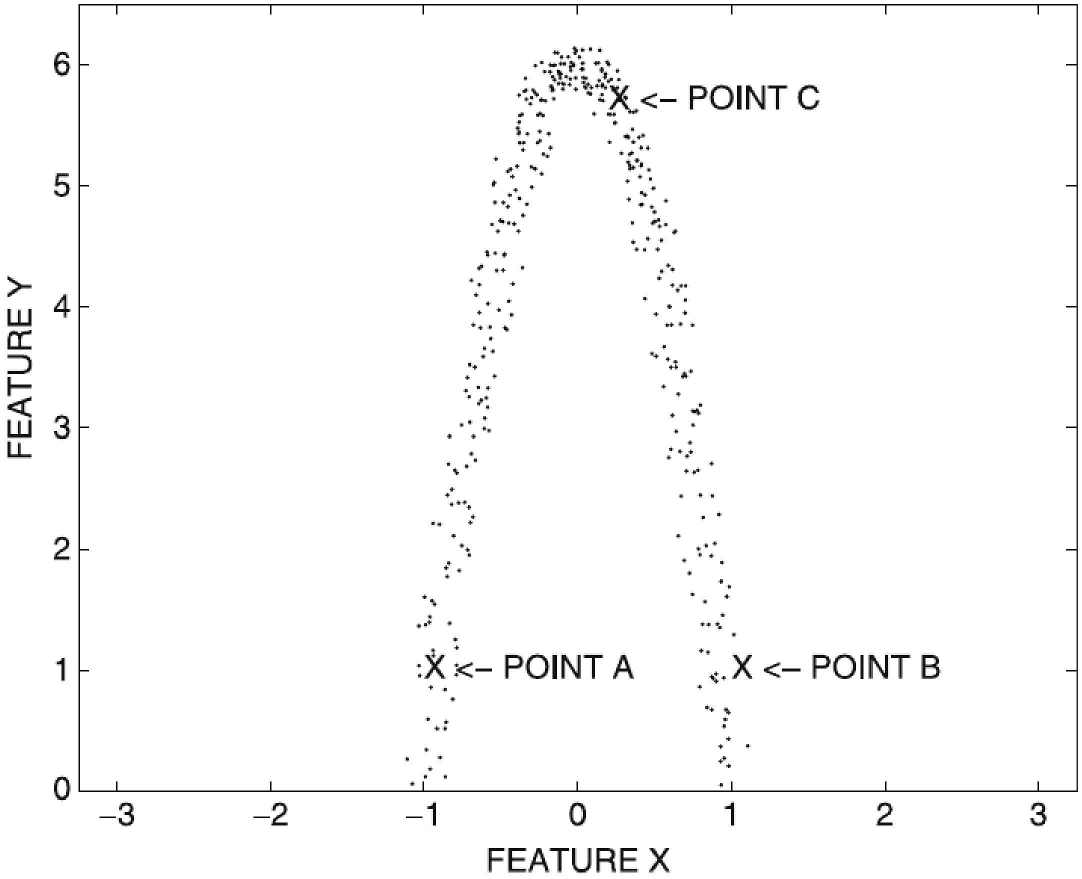
$$Maha(\overline{X}, \overline{Y}) = \sqrt{(\overline{X} - \overline{Y})\Sigma^{-1}(\overline{X} - \overline{Y})^T}.$$

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

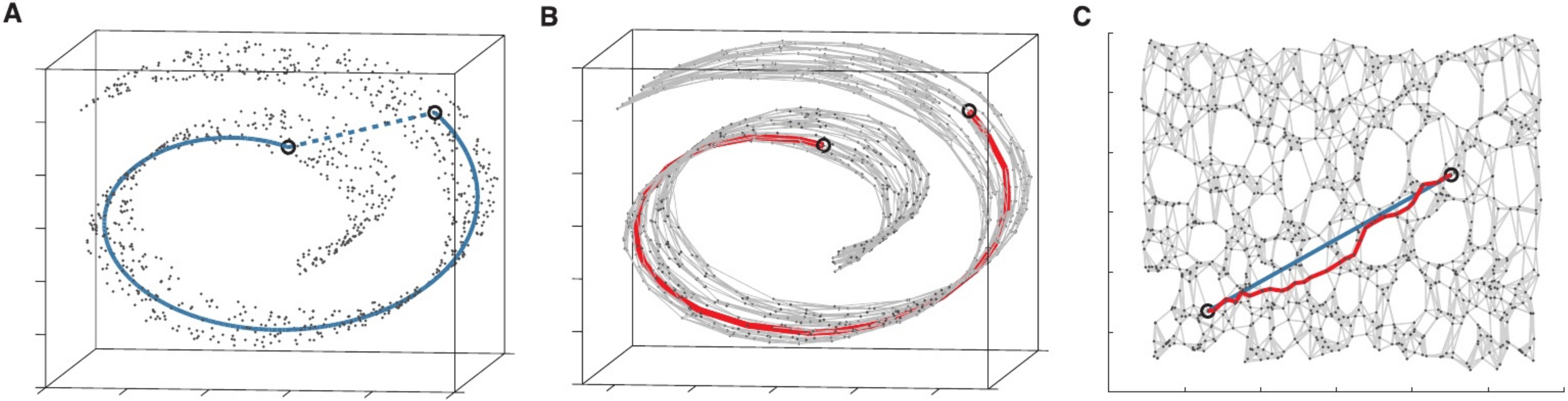


Non-linear distributions

Which point would you consider as closer to A?



ISOMAP (general idea)



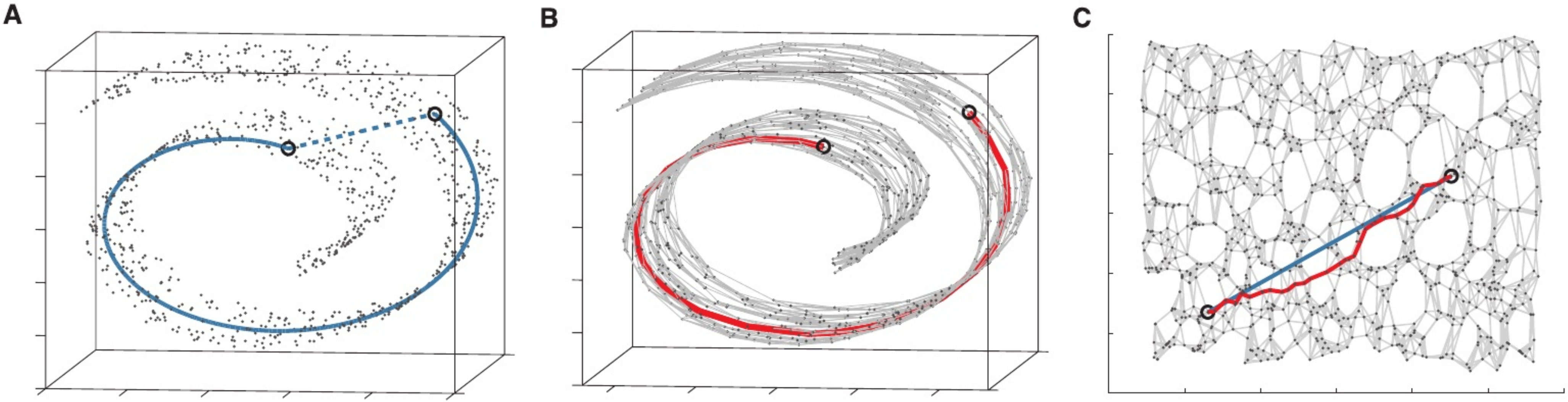
Original data

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.

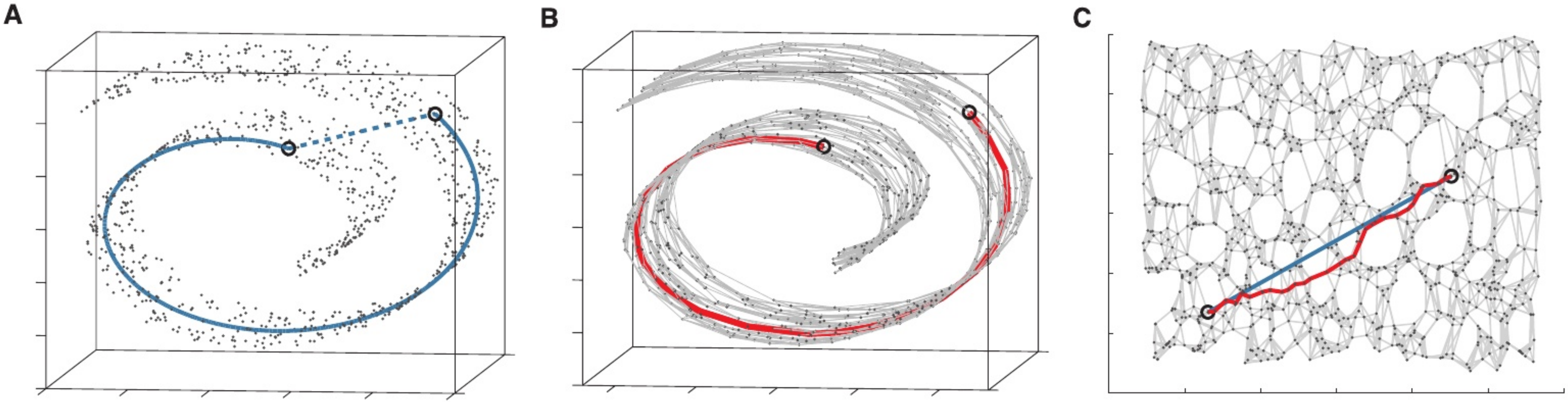
ISOMAP (1/3)



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.

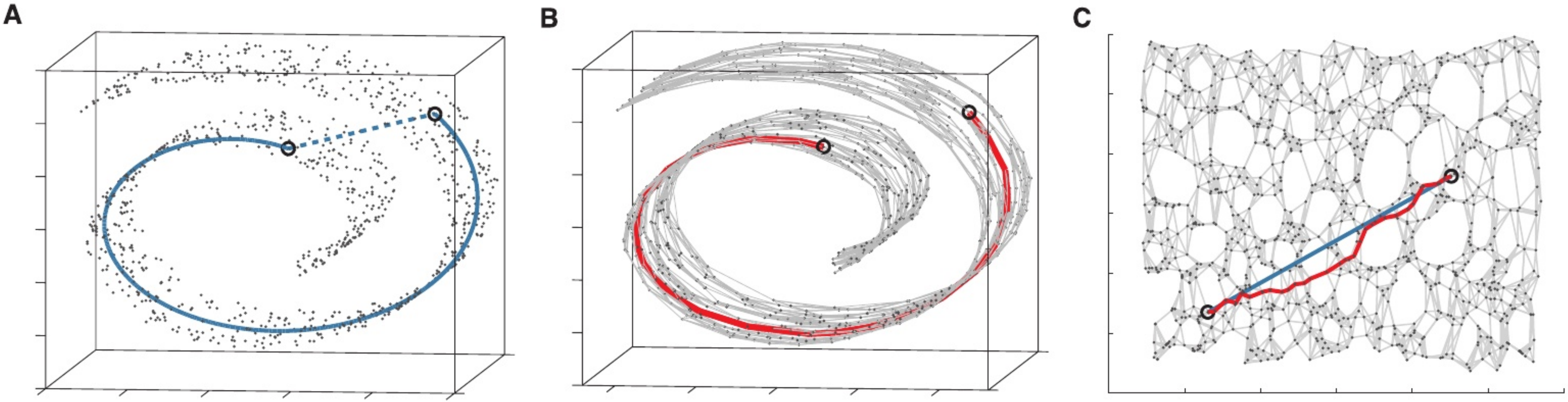
ISOMAP (2/3)



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)



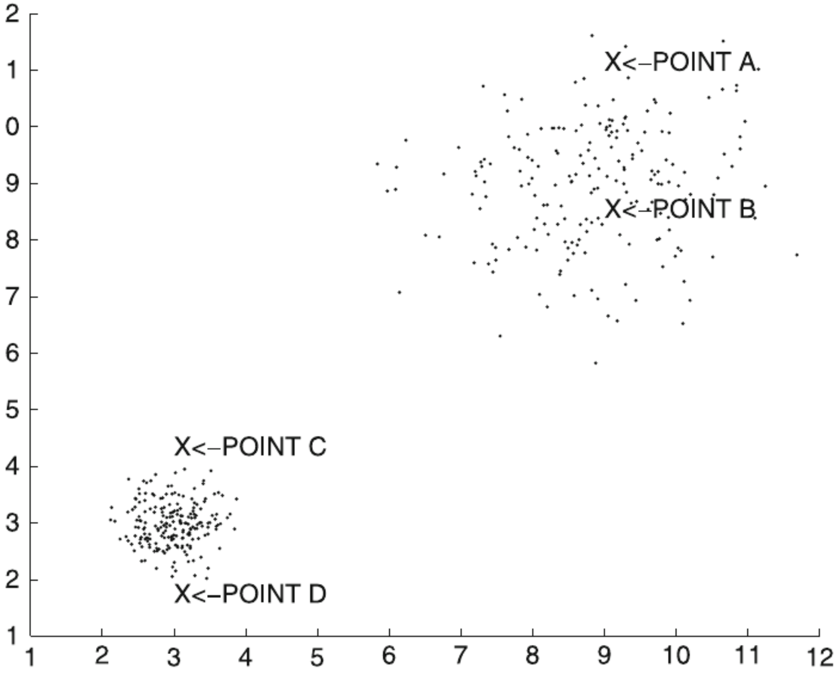
It is, however, more effective to project the graph and compute Euclidean distances in the projected graph (blue color)

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

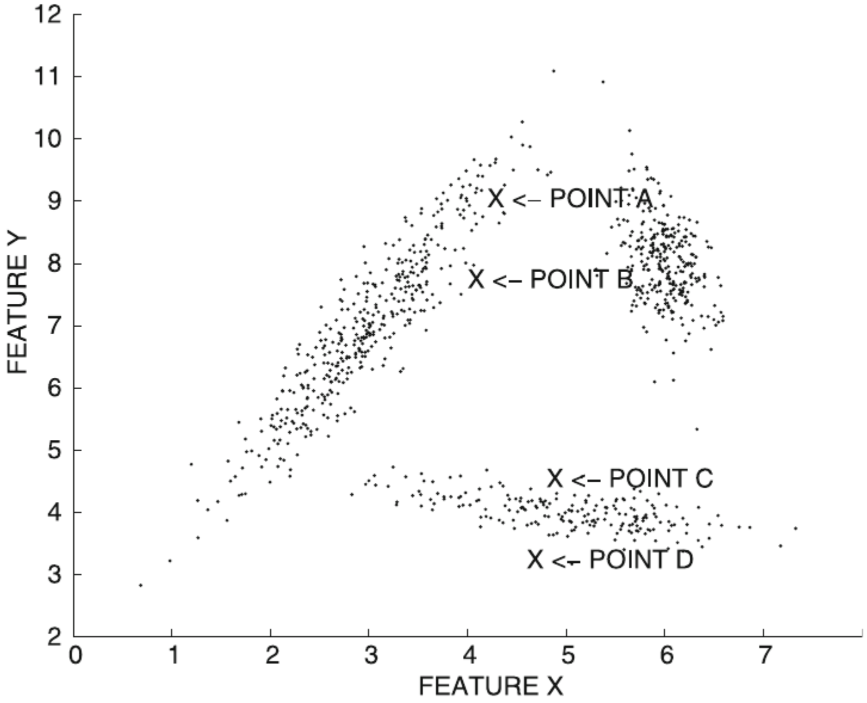
Local variations

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

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

Categorical and mixed data

Simple similarity for categorical data

- Given $\overline{X} = (x_1, \ldots, x_d); \overline{Y} = (x_1, \ldots, x_d)$
- Compute similarity as

$$\mathrm{sim}(\overline{X}, \overline{Y}) = \sum_{i=1}^{d} S(x_i, y_i)$$

- Simple coordinate-wise similarity

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

Weighing feature values by how rare they are

- Compute similarity as $\mathrm{sim}(\overline{X}, \overline{Y}) = \sum_{i=1}^{d} S(x_i, y_i)$
- Inverse occurrence frequency

  $p_i(z)$ is the probability that feature i takes value z

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

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

Goodall measure

Mixture of quantitative and categorical data

- Given $\overline{X} = (\overline{X_c}, \overline{X_n}); \overline{Y} = (\overline{Y_c}, \overline{Y_n});$
- Where c denotes the subset of categorical data and n the subset of numerical data

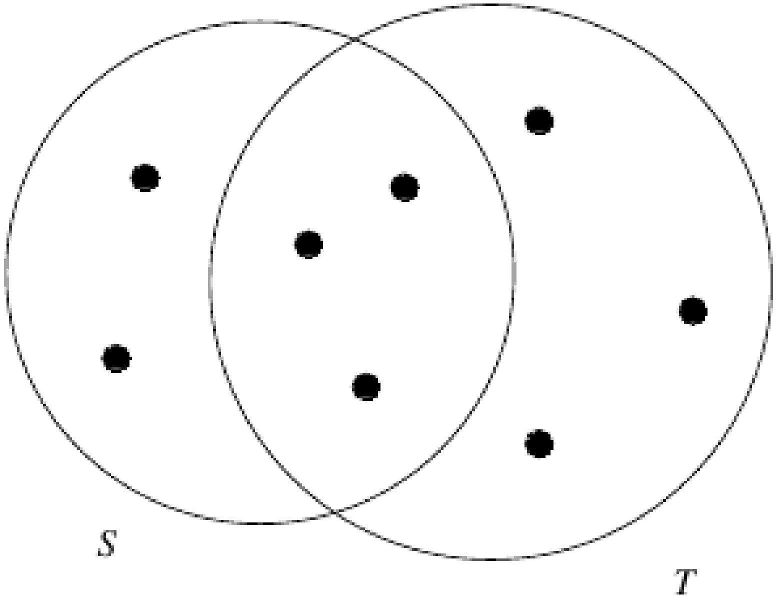
$$\operatorname{sim}(\overline{X}, \overline{Y}) = \lambda \operatorname{CatSim}(\overline{X_c}, \overline{Y_c}) + (1 - \lambda) \operatorname{NumSim}(\overline{X_n}, \overline{Y_n})$$

- In general λ is difficult to set, and additionally we should have variables with similar variances or normalize by variance

Binary and set data

Jaccard coefficient

Example: J(S,T) = 3/8



$$J(S,T) = \frac{|S \cap T|}{|S \cup T|}$$

Binary variables can be set as set inclusion variables

- If $\overline{X}=(x_1, \ldots, x_d)$ is such that $x_i = 1$, this can be seen as element $\overline{X}$ belonging to set i
- Alternatively, $\overline{X}$ can be seen as $S_{\overline{X}}$ the set of all variables i such that $x_i = 1$
- Extended Jaccard coefficient (Tanimoto distance)

$$J(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i \cdot y_i}$$

Try it!

- Compute Tanimoto and Jaccard* distance between:

  (0, 2, 1, 0, 3)

  (1, 2, 0, 0, 0)

  * For the Jaccard distance, binarize the vectors

$$J(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i \cdot y_i}$$

Text data

Text documents as vectors: L_p norms

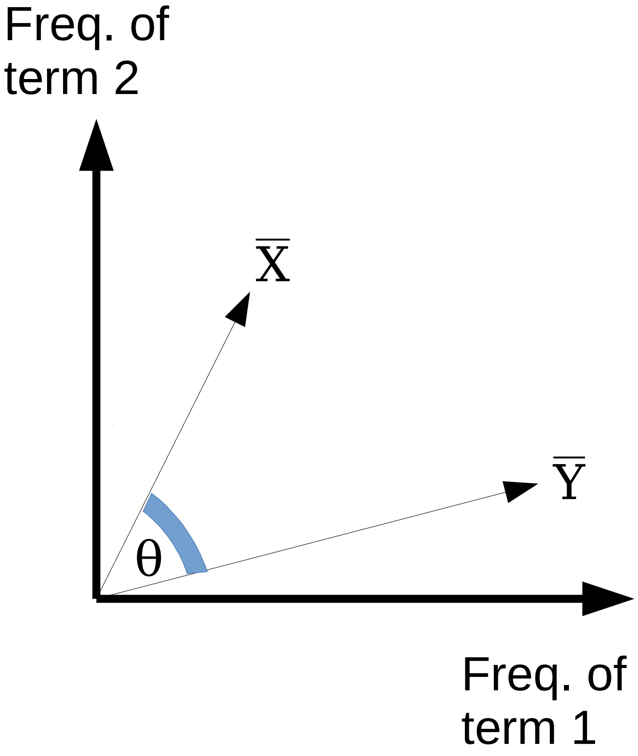
- As Quantitative Multidimensional Data
  - Bag of words model
  - They are very sparse
  - L_p norm does not work well
  - Long documents have long distance
- Dimensionality Reduction (A Possible Solution)
  - Latent Semantic Analysis (equivalent to SVD)
  - L_p norm in the new space

Text documents as vectors: angles

- What we care about is the relative frequency of terms

$$\mathrm{sim}(\overline{X},\overline{Y}) = \cos\theta$$

$$\mathrm{sim}(\overline{X},\overline{Y}) = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}$$



However, some terms are very common and others are very rare …

Text documents as vectors: tf-idf weighting (idf)

- $\mathrm{idf}(t) = \log \frac{n}{n_t}$
  - Global inverse document frequency of term t
  - Where n_t is the number of documents where term t appears, n is the total number of documents
- Typical variation (in Okapi BM25):

$$\mathrm{idf}(t) = \log \frac{n - n_t + 0.5}{n_t + 0.5}$$

Text documents as vectors: tf-idf weighting (tf)

- tf(x_i)
  - Frequency in a document of term x_i
  - Log frequency, square root of frequency, or similar to reduce the impact of terms of very high frequency

Text documents as vectors: tf-idf weighting (cont.)

- $h(x_i) = tf(x_i) \times idf(x_i)$

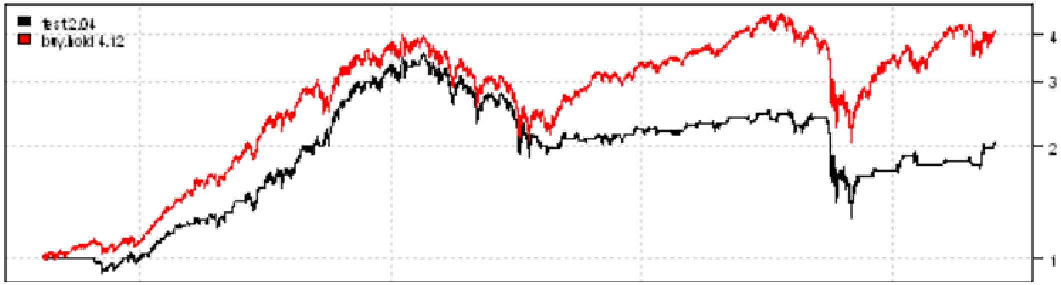
$$\mathrm{sim}(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} h(x_i) \cdot h(y_i)}{\sqrt{\sum_{i=1}^{d} h(x_i)^2} \cdot \sqrt{\sum_{i=1}^{d} h(y_i)^2}}$$

- Or Jaccard-like:

$$J(\overline{X}, Y) = \frac{\sum_{i=1}^{d} h(x_i) \cdot h(y_i)}{\sum_{i=1}^{d} h(x_i)^2 + \sum_{i=1}^{d} h(y_i)^2 - \sum_{i=1}^{d} h(x_i) \cdot h(y_i)}$$

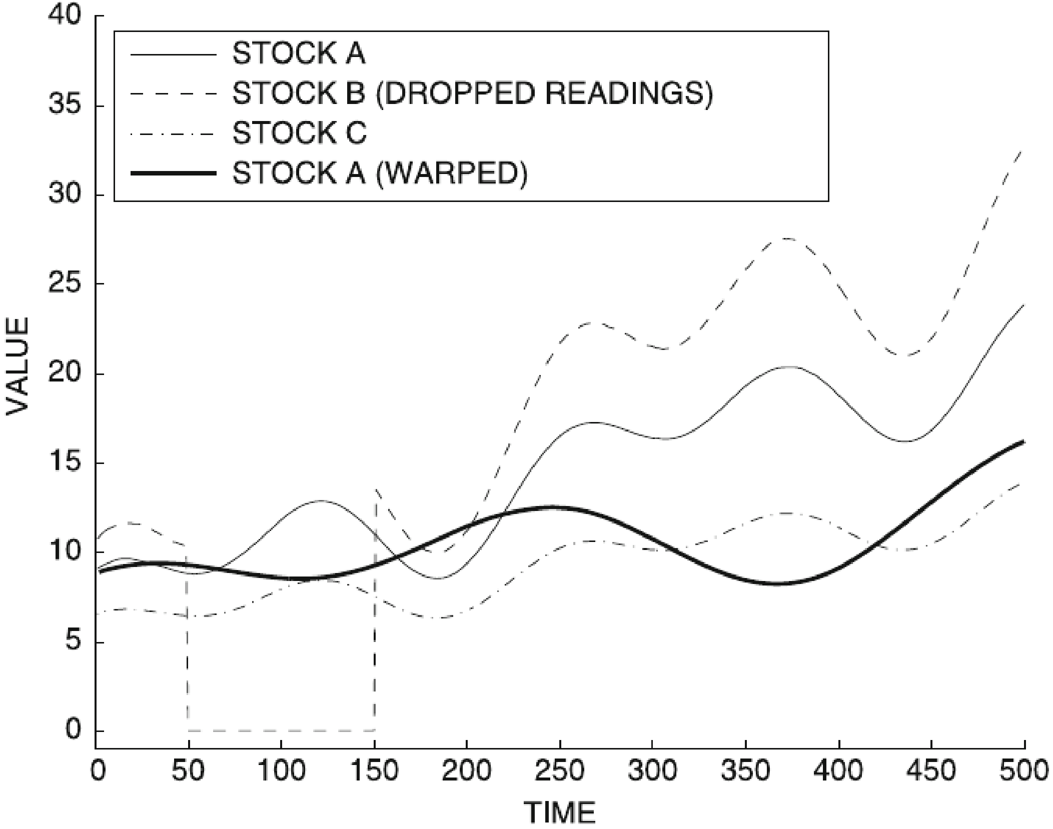
Continuous time series data

Misalignment between series



- Behavioral attributes
  - Scaling (range is larger or narrower)
  - Translation (series is shifted up or down)
- Contextual attribute (typically, time)
  - Scaling (time is stretched or compressed)
  - Translation or shift (starting time changes)
- Matches might not be contiguous (noisy segments)

Example of scaling, translation, noise



More on this
later in the course,
in the
sequence mining topic

Data Mining, The Textbook (2015) by Charu Aggarwal (Chapter 3).

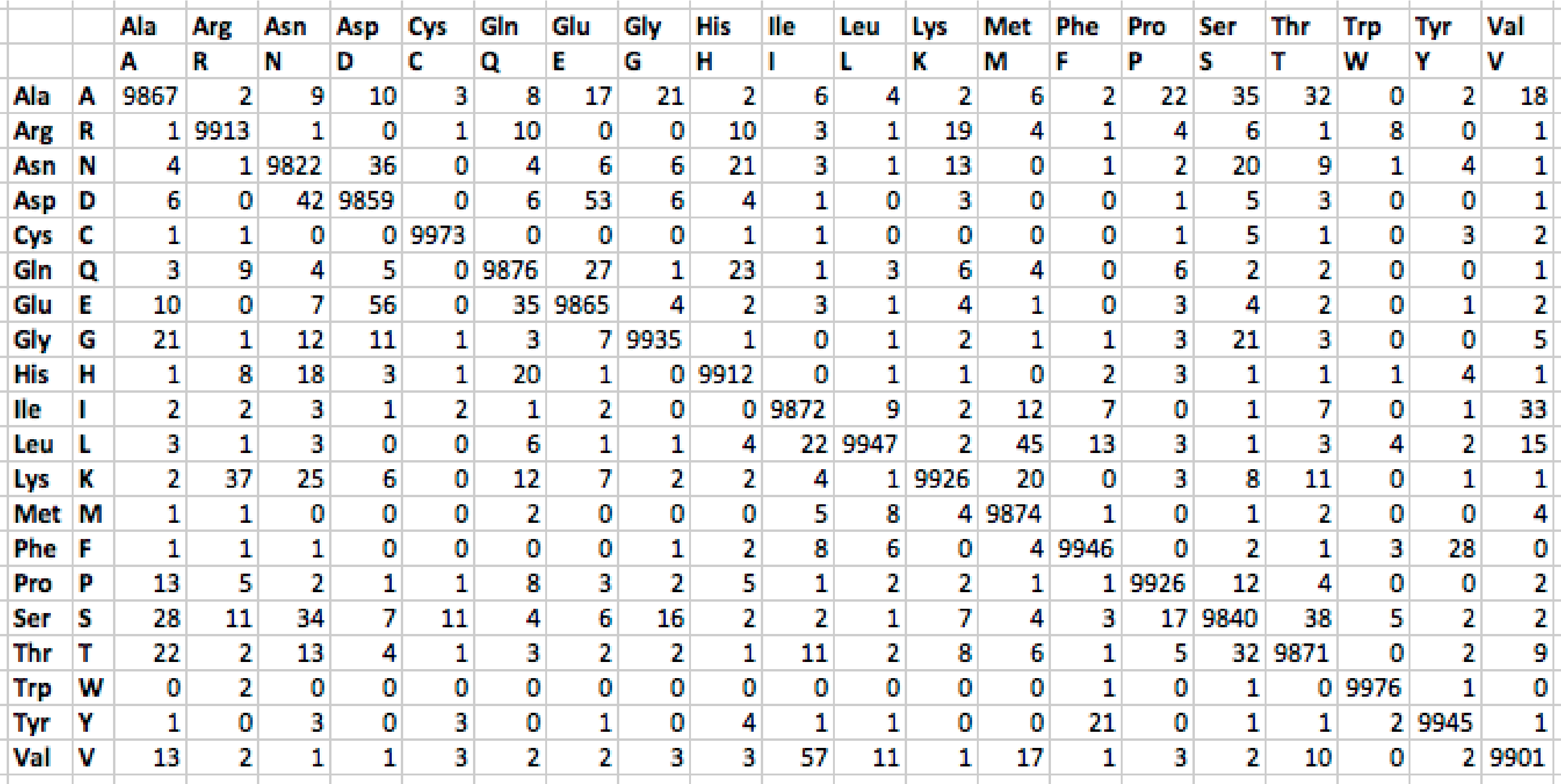
Warping means stretching or compressing time.

Discrete sequence data

Discrete sequences can be treated as strings

- Compute edit distance
- Compute longest common sub-sequence
- In genetic sequences, use
  PAM (*Point Accepted Mutation*) matrices
  - Indicate rarity (cost) of replacement

Example PAM matrix

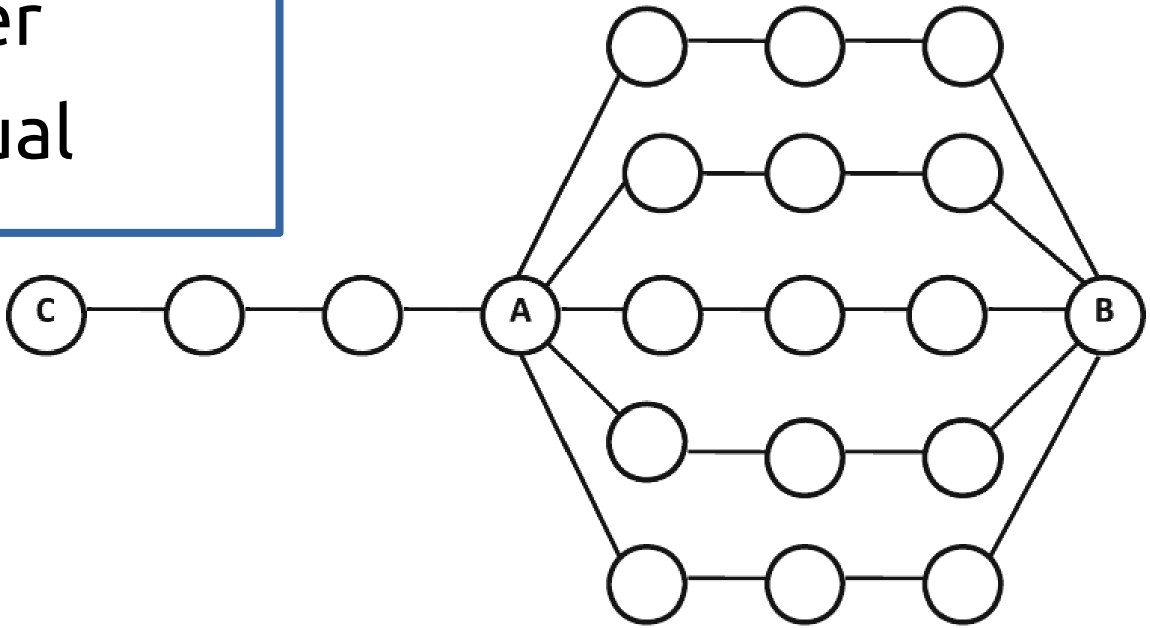


		Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Ala	A	9867	2	9	10	3	8	17	21	2	6	4	2	6	2	22	35	32	0	2	18
Arg	R	1	9913	1	0	1	10	0	0	10	3	1	19	4	1	4	6	1	8	0	1
Asn	N	4	1	9822	36	0	4	6	6	21	3	1	13	0	1	2	20	9	1	4	1
Asp	D	6	0	42	9859	0	6	53	6	4	1	0	3	0	0	1	5	3	0	0	1
Cys	C	1	1	0	0	9973	0	0	0	1	1	0	0	0	0	1	5	1	0	3	2
Gln	Q	3	9	4	5	0	9876	27	1	23	1	3	6	4	0	6	2	2	0	0	1
Glu	E	10	0	7	56	0	35	9865	4	2	3	1	4	1	0	3	4	2	0	1	2
Gly	G	21	1	12	11	1	3	7	9935	1	0	1	2	1	1	3	21	3	0	0	5
His	H	1	8	18	3	1	20	1	0	9912	0	1	1	0	2	3	1	1	1	4	1
Ile	I	2	2	3	1	2	1	2	0	0	9872	9	2	12	7	0	1	7	0	1	33
Leu	L	3	1	3	0	0	6	1	1	4	22	9947	2	45	13	3	1	3	4	2	15
Lys	K	2	37	25	6	0	12	7	2	2	4	1	9926	20	0	3	8	11	0	1	1
Met	M	1	1	0	0	0	2	0	0	0	5	8	4	9874	1	0	1	2	0	0	4
Phe	F	1	1	1	0	0	0	0	1	2	8	6	0	4	9946	0	2	1	3	28	0
Pro	P	13	5	2	1	1	8	3	2	5	1	2	2	1	1	9926	12	4	0	0	2
Ser	S	28	11	34	7	11	4	6	16	2	2	1	7	4	3	17	9840	38	5	2	2
Thr	T	22	2	13	4	1	3	2	2	1	11	2	8	6	1	5	32	9871	0	2	9
Trp	W	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	9976	1	0
Tyr	Y	1	0	3	0	3	0	1	0	4	1	1	0	0	21	0	1	1	2	9945	1
Val	V	13	2	1	1	3	2	2	3	3	57	11	1	17	1	3	2	10	0	2	9901

Graph data

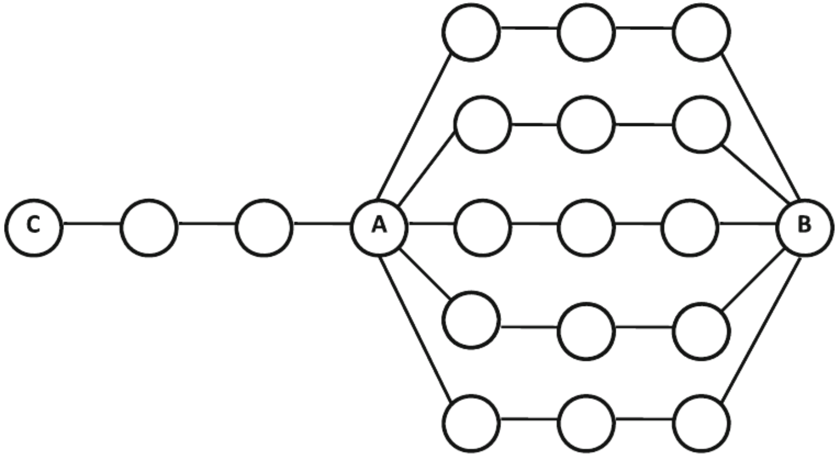
Distance/similarity in graph data

- Comparing A-B and A-C?
  - A-B should be closer
  - A-C should be closer
  - Both should be equal



Distance/similarity in graph data

- Distance-Based Measure
  - Shortest-path on the graph
  - Dijkstra algorithm
- Random Walk-Based Similarity
  - (e.g. personalized PageRank)
  - Accounts for multiplicity in paths during similarity computation



Under random walk similarity, A-B are closer than A-C

Supervised similarity functions

Learning a distance function through supervised ML

- Suppose you have data from experts, annotators, or user feedback:

$$\mathcal{S} = \{O_i, O_j : O_i \text{ is similar to } O_j\}$$

$$\mathcal{D} = \{O_i, O_j : O_i \text{ is dissimilar to } O_j\}$$

- Learn a distance *$f(O_i, O_j, \theta)$: UxU → [0,1]*

$$\min_{\theta} \sum_{(O_i,O_j)\in\mathcal{S}} (f(O_i,O_j,\theta)-0)^2 + \sum_{(O_i,O_j)\in\mathcal{D}} (f(O_i,O_j,\theta)-1)^2$$

Summary

Things to remember

- Distance/similarity is a key component of many data mining algorithms
- Sensitive to type, dimensionality, global/local nature of data distribution
  - Heterogeneous data may require local normalization
- Different solutions for different data types

Exercises for this topic

- **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