

# Similarity and Distances

---

Lijun Zhang

[zlj@nju.edu.cn](mailto:zlj@nju.edu.cn)

<http://cs.nju.edu.cn/zlj>





# Outline

---

- **Introduction**
- Multidimensional Data
- Text Similarity Measures
- Temporal Similarity Measures
- Graph Similarity Measures
- Supervised Similarity Functions
- Summary



# Introduction

---

## □ Motivation

*“Love is the power to see similarity in the dissimilar.”*—Theodor Adorno

## □ Definition

*Given two objects  $O_1$  and  $O_2$ , determine a value of the similarity  $Sim(O_1, O_2)$  (or distance  $Dist(O_1, O_2)$ ) between the two objects.*

- Distance functions for spatial data
- Similarity functions for text

## □ Representation

- Closed-form, such as Euclidean distance
- Defined algorithmically



# Outline

---

- Introduction
- **Multidimensional Data**
- Text Similarity Measures
- Temporal Similarity Measures
- Graph Similarity Measures
- Supervised Similarity Functions
- Summary



# Multidimensional Data (Vectors)

---

- **Quantitative Data**
- Categorical Data
- Mixed Quantitative and Categorical Data



# Quantitative Data (1)

## □ $L_p$ -Norm ( $p \geq 1$ )

■ Given  $\bar{X} = (x_1 \dots x_d)$  and  $\bar{Y} = (y_1 \dots y_d)$

$$Dist(\bar{X}, \bar{Y}) = \left( \sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$

Given a **vector space**  $V$  over a **subfield**  $F$  of the **complex numbers**, a **norm** on  $V$  is a **function**  $p: V \rightarrow \mathbf{R}$  with the following properties:<sup>[1]</sup>

For all  $a \in F$  and all  $\mathbf{u}, \mathbf{v} \in V$ ,

1.  $p(a\mathbf{v}) = |a| p(\mathbf{v})$ , (*absolute homogeneity* or *absolute scalability*).
2.  $p(\mathbf{u} + \mathbf{v}) \leq p(\mathbf{u}) + p(\mathbf{v})$  (*triangle inequality* or *subadditivity*).
3. If  $p(\mathbf{v}) = 0$  then  $\mathbf{v}$  is the **zero vector** (*separates points*).

[https://en.wikipedia.org/wiki/Norm\\_\(mathematics\)](https://en.wikipedia.org/wiki/Norm_(mathematics))



# Quantitative Data (2)

## □ $L_p$ -Norm ( $p \geq 1$ )

- Given  $\bar{X} = (x_1 \dots x_d)$  and  $\bar{Y} = (y_1 \dots y_d)$

$$Dist(\bar{X}, \bar{Y}) = \left( \sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$

- $p = 1$  : 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



# Quantitative Data (3)

---

## □ " $L_p$ -Norm" ( $p < 1$ )

- Given  $\bar{X} = (x_1 \dots x_d)$  and  $\bar{Y} = (y_1 \dots y_d)$

$$Dist(\bar{X}, \bar{Y}) = \left( \sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$

- $p = 0$  : Zero norm
  - ✓ Number of nonzero elements
  - ✓ Nonconvex
- $0 < p < 1$ : Fractional-norm
  - ✓ Nonconvex



# Impact of Domain-Specific Relevance



## □ Some Features are more Important

### ■ Credit-scoring

✓ Salary is more important than Gender

## □ Generalized $L_p$ -Norm

$$Dist(\bar{X}, \bar{Y}) = \left( \sum_{i=1}^d a_i \cdot |x_i - y_i|^p \right)^{1/p}.$$

- $a_1, \dots, a_d$  are nonnegative coefficients
- Generalized Minkowski distance



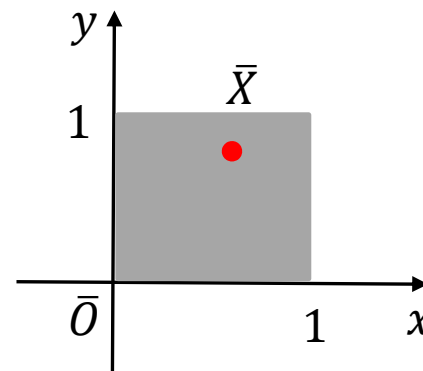
# Impact of High Dimensionality (1)

## □ Curse of Dimensionality

- Distance-based algorithms lose their effectiveness as the dimensionality increases

## □ An 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  $\bar{X}$





## Impact of High Dimensionality (2)

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

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

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

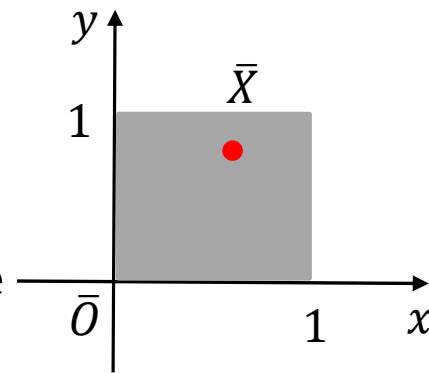
- $Dist(\bar{O}, \bar{X})$  is a random variable

- ✓ Since  $\bar{X}$  is a random variable

- ✓ Mean is  $\mu = d/2$

- ✓ Standard deviation  $\sigma = \sqrt{d/12}$

- With a probability at least 8/9



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



# Impact of High Dimensionality (2)

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

$$Dist(\bar{O}, \bar{X}) = \sum_{i=1}^d (x_i - o_i)$$

where  $\bar{X} =$

- $Dist(\bar{O}, \bar{X})$

✓ Since  $\bar{X}$

✓ Mean is

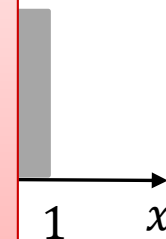
✓ Standard

- With a probability at least 8/9

## Chebyshev's inequality

Let  $X$  be a random variable with finite expected value  $\mu$  and finite non-zero variance  $\sigma^2$ . Then for any real number  $k > 0$ ,

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



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



## Impact of High Dimensionality (3)

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

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

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

- ✓ Mean is  $\mu = d/2$

- ✓ Standard deviation  $\sigma = \sqrt{d/12}$

- With a probability at least 8/9

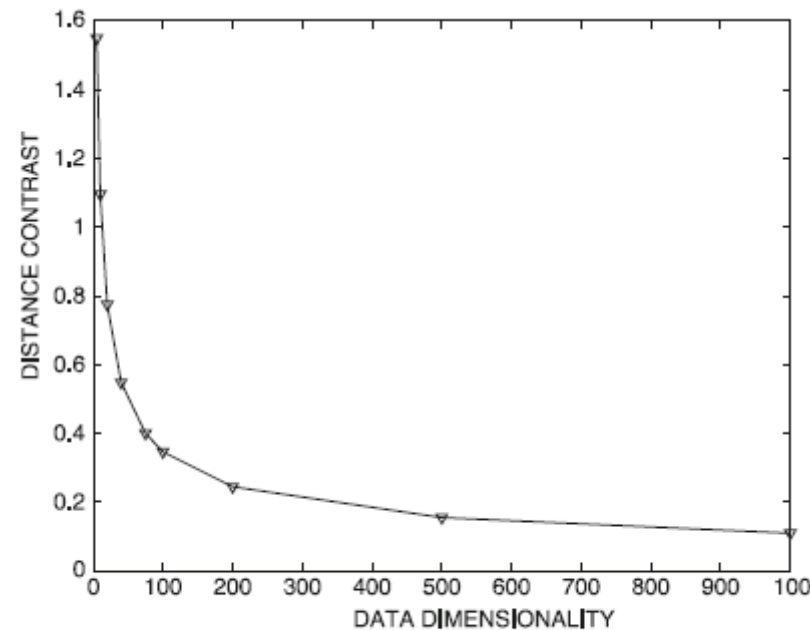
$$Dist(\bar{O}, \bar{X}) \in [\underbrace{\mu - 3\sigma}_{D_{min}}, \underbrace{\mu + 3\sigma}_{D_{max}}]$$

- Contrast

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

## Impact of High Dimensionality (4)

- Contrast  $\rightarrow 0$ , as  $d \rightarrow \infty$ 
  - As  $d$  increases, variation become neglectable



(a) Contrasts with dimensionality

# Impact of Locally Irrelevant Features

---



- Many features are likely to be irrelevant
  - Especially in high-dimensional data
- An Example
  - A cluster containing diabetic patients
    - ✓ Blood glucose level are more important
- $L_p$ -Norm
  - Suffer from the additive noise effects of the irrelevant features



# Impact of Different $L_p$ -Norms (1)

---

- Different  $L_p$ -Norms do not behave in a similar way
  - When the dimensionality is high
  - When there exist irrelevant features
- $L_\infty$ -Norms

$$\text{dist}(\bar{X}, \bar{Y}) = \max_i |x_i - y_i|$$

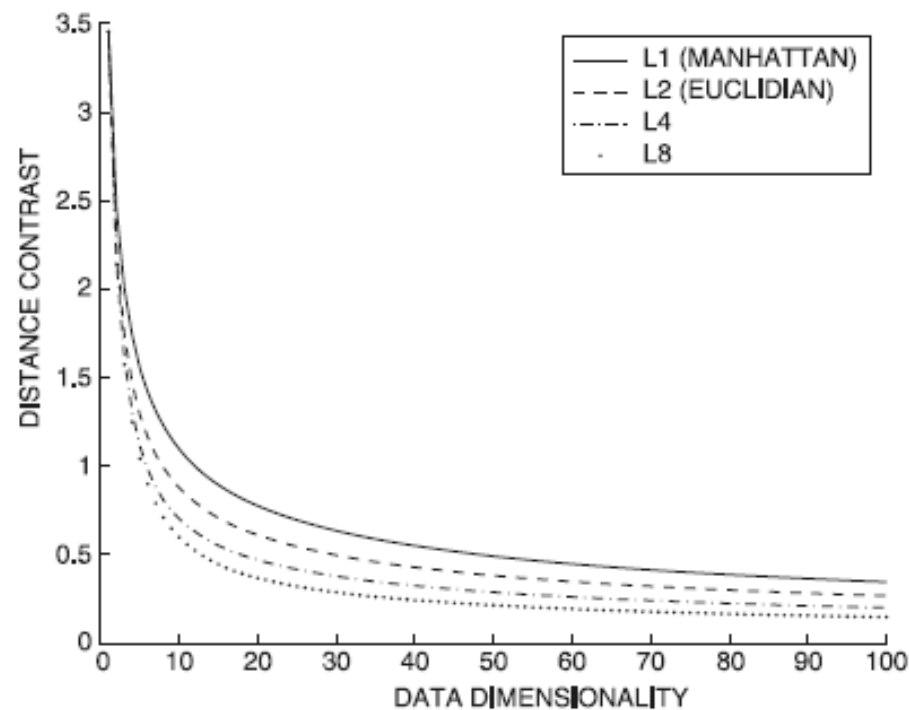
- Sensitive to noise
- Irrelevant attributes are emphasized for large values of  $p$





# Impact of Different $L_p$ -Norms (2)

- Distance contrasts are also poorer for large values of  $p$

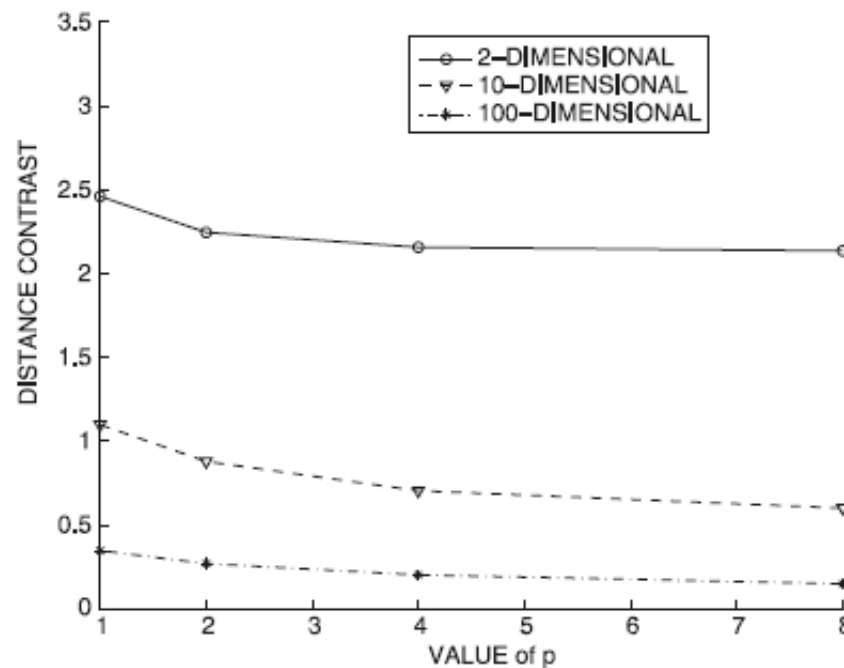


(b) Contrasts with norms



# Impact of Different $L_p$ -Norms (3)

- Distance contrasts are also poorer for large values of  $p$



(a) Contrast



# Match-Based Similarity Computation

## □ The Key Idea

- De-emphasize irrelevant features

## □ Proximity Thresholding

- Discretized each feature into  $k_d$  **equidepth** buckets

## □ Similarity Evaluation

$$\bar{X} = [x_1, x_2, \dots, x_d] \longrightarrow [1, 3, \dots, k_d]$$

$$\bar{Y} = [y_1, y_2, \dots, y_d] \longrightarrow [5, 3, \dots, k_d]$$

- $S(\bar{X}, \bar{Y}, k_d)$  is the set of features mapped to the same bucket



# Match-Based Similarity Computation

## □ The Key Idea

- De-emphasize irrelevant features

## □ Proximity Thresholding

- Discretized each feature into  $k_d$  **equidepth** buckets

## □ Similarity Evaluation

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

- $S(\bar{X}, \bar{Y}, k_d)$  is the set of features mapped to the same bucket



# Match-Based Similarity Computation

## □ The Key Idea

- De-emphasizes

## □ Proximity

- Discards  
equidistant

Picking  $k_d \propto d$  achieves a constant level of contrast in high dimensional space for certain data distributions.

## □ Similarity Evaluation

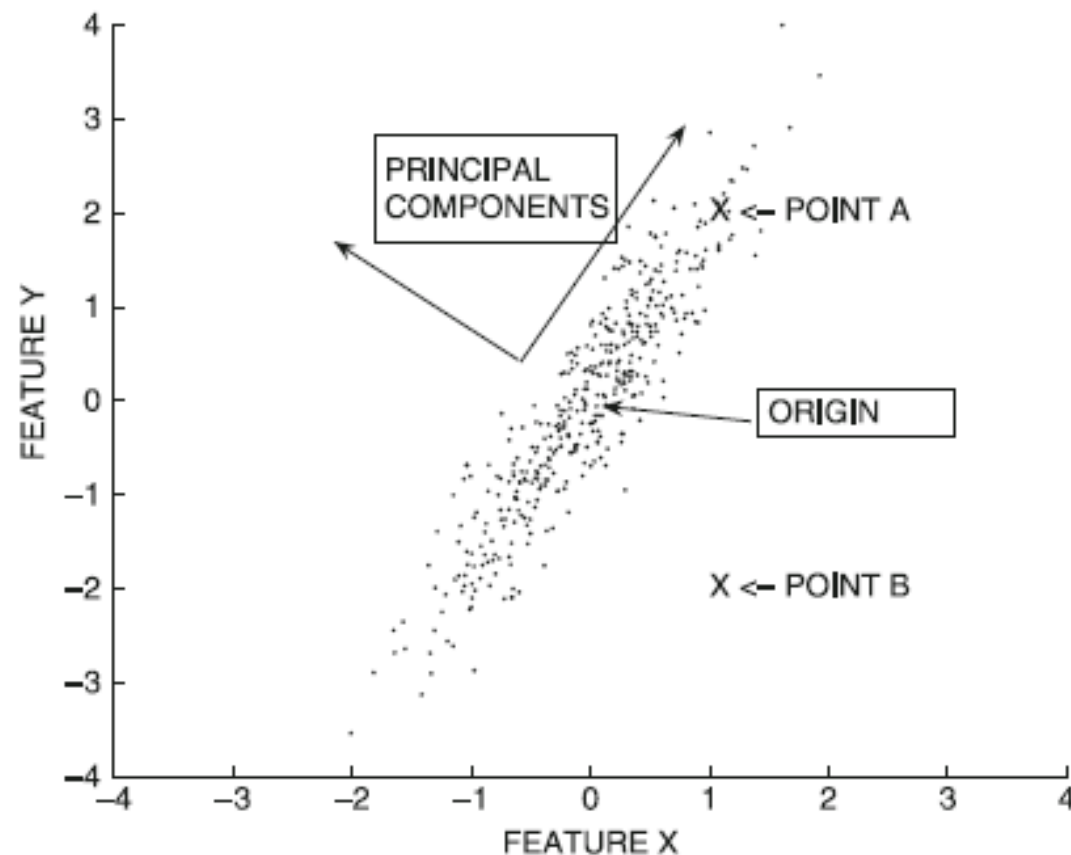
$$PSelect(\bar{X}, \bar{Y}, k_d) = \left[ \sum_{i \in S(\bar{X}, \bar{Y}, k_d)} \left( 1 - \frac{|x_i - y_i|}{m_i - n_i} \right)^p \right]^{1/p} \in [0, S(\bar{X}, \bar{Y}, k_d)]$$

- $S(\bar{X}, \bar{Y}, k_d)$  is the set of features mapped to the same bucket



# Impact of Data Distribution (1)

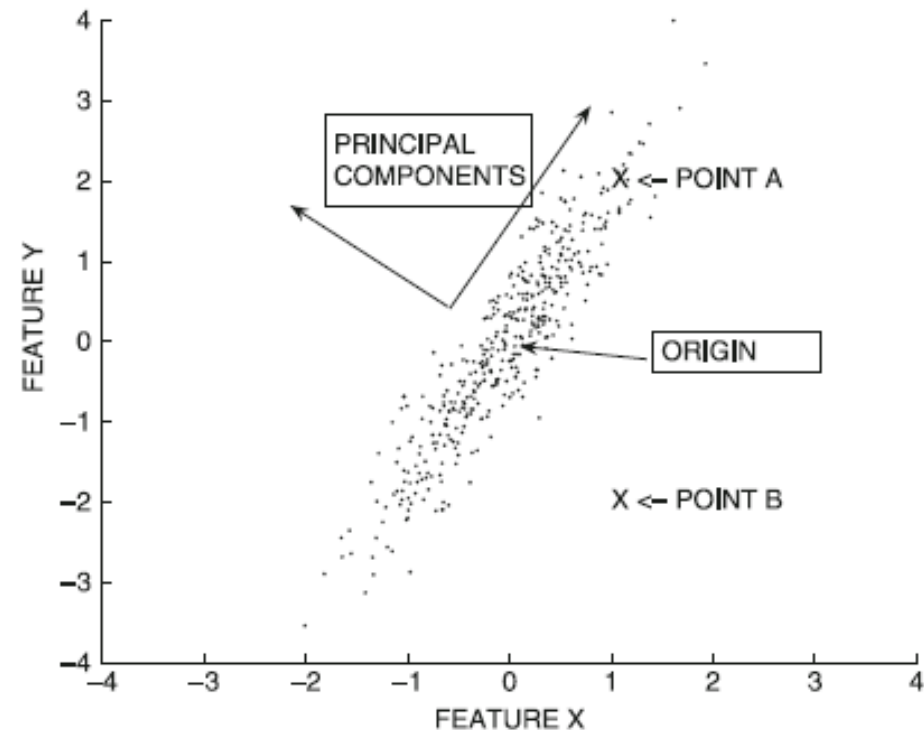
□  $A = (1, 2)$  and  $B = (1, -2)$





# Impact of Data Distribution (2)

- $A = (1, 2)$  and  $B = (1, -2)$ 
  - $O \rightarrow A$  is aligned with a **high-variance** direction
  - $O \rightarrow B$  is aligned with a **low-variance** direction
  - $O \rightarrow A$  ought to be less than  $O \rightarrow B$





# Impact of Data Distribution (3)

## □ The Mahalanobis distance

- Let  $\Sigma$  be the covariance matrix

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

- Projection+Normalization

- ✓ Let  $\Sigma = U\Lambda U^T = \sum_{i=1}^d \sigma_i \mathbf{u}_i \mathbf{u}_i^T$
- ✓ Then,  $\Sigma^{-1} = U\Lambda^{-1}U^T = \sum_{i=1}^d \sigma_i^{-1} \mathbf{u}_i \mathbf{u}_i^T$

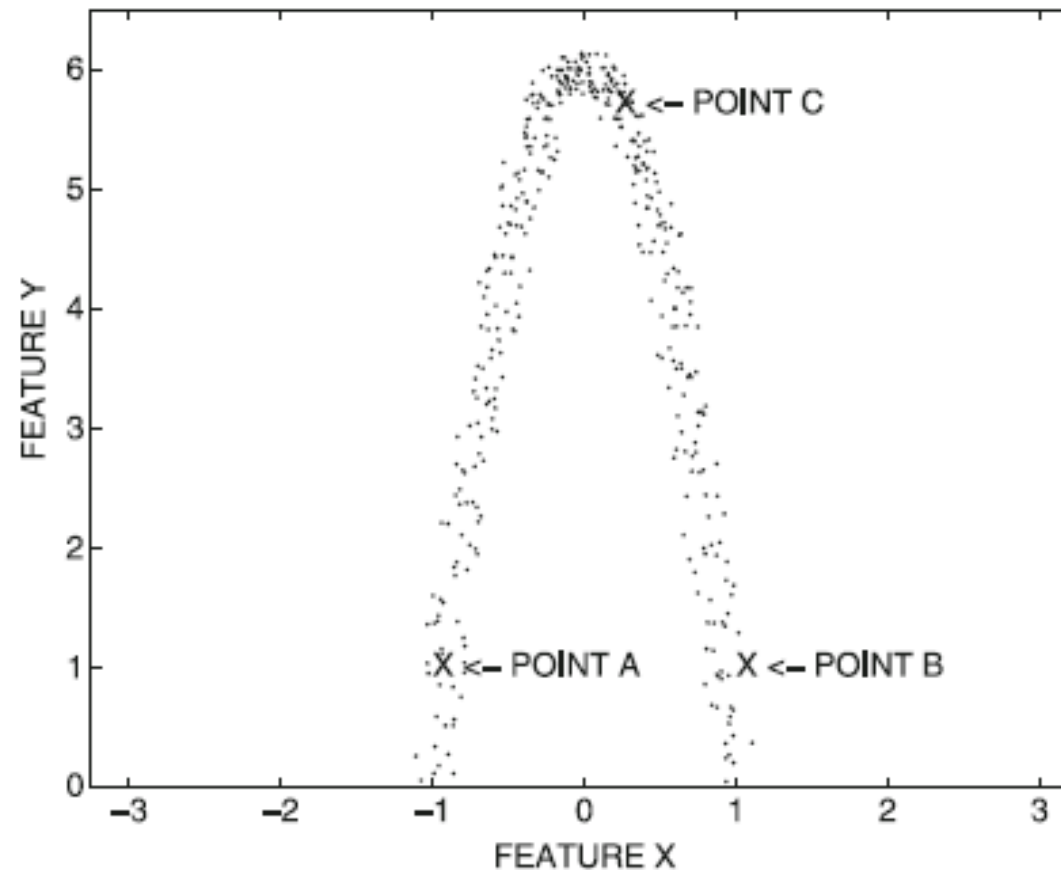
$$Maha(\bar{X}, \bar{Y}) = \sqrt{(\bar{X} - \bar{Y}) \left( \sum_{i=1}^d \sigma_i^{-1} \mathbf{u}_i \mathbf{u}_i^T \right) (\bar{X} - \bar{Y})^T} = \sqrt{\sum_{i=1}^d \frac{((\bar{X} - \bar{Y})\mathbf{u}_i)^2}{\sigma_i}}$$





# Nonlinear Distributions: ISOMAP (1)

□ Which one of  $B$  and  $C$  is closer to  $A$ ?

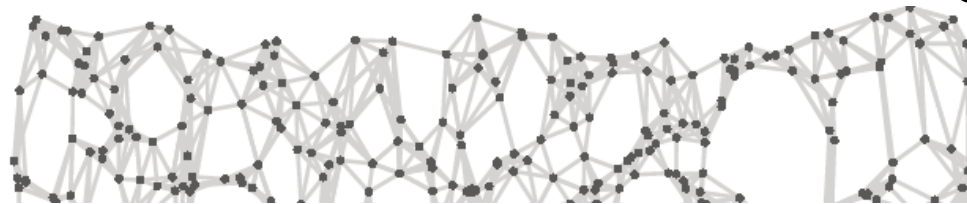


# Nonlinear Distributions: ISOMAP (2)

---

## □ Geodesic Distances

- Compute the  $k$ -nearest neighbors of each point
- Construct a weighted graph  $G$  with nodes representing data points, and edge weights representing (Euclidean) distance of these  $k$ -nearest neighbors



- $Dist(\bar{X}, \bar{Y})$  is the shortest path between  $\bar{X}$  and  $\bar{Y}$  in the graph



# Nonlinear Distributions: ISOMAP (3)

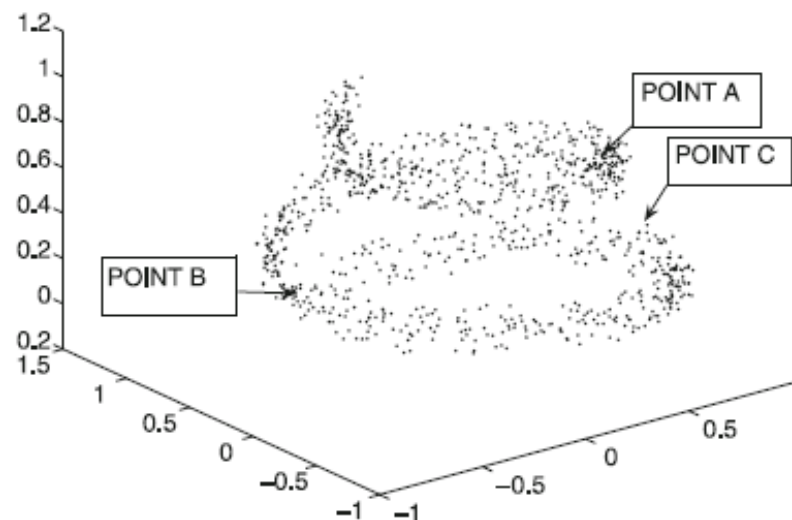
---

## □ Nonlinear Dimensionality Reduction by ISOMAP

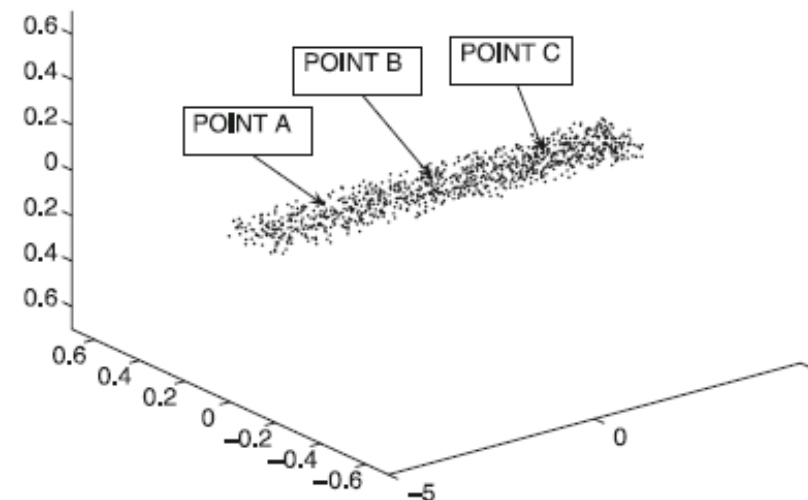
- Compute the  $k$ -nearest ...
- Construct a weighted graph  $G$  ...
- Compute the distances between **all pairs** of data points
  - ✓  $A$   $d \times d$  distance matrix
- Find vector representations by **multidimensional scaling (MDS)**
- $Dist(\bar{X}, \bar{Y})$  is the Euclidean distance of the new representations

# Nonlinear Distributions: ISOMAP (4)

## □ An Example of ISOMAP



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



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

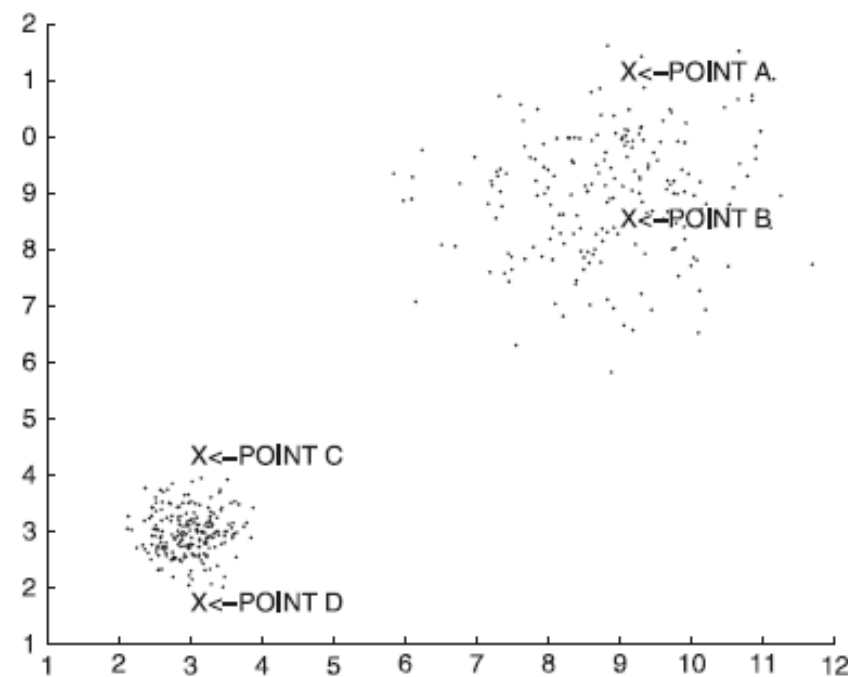
## ■ Manifold Learning (ISOMAP, LLE, LE)



# Impact of Local Data Distribution (1)

## □ Local Density Variation

- $C-D$  should be longer than  $A-B$



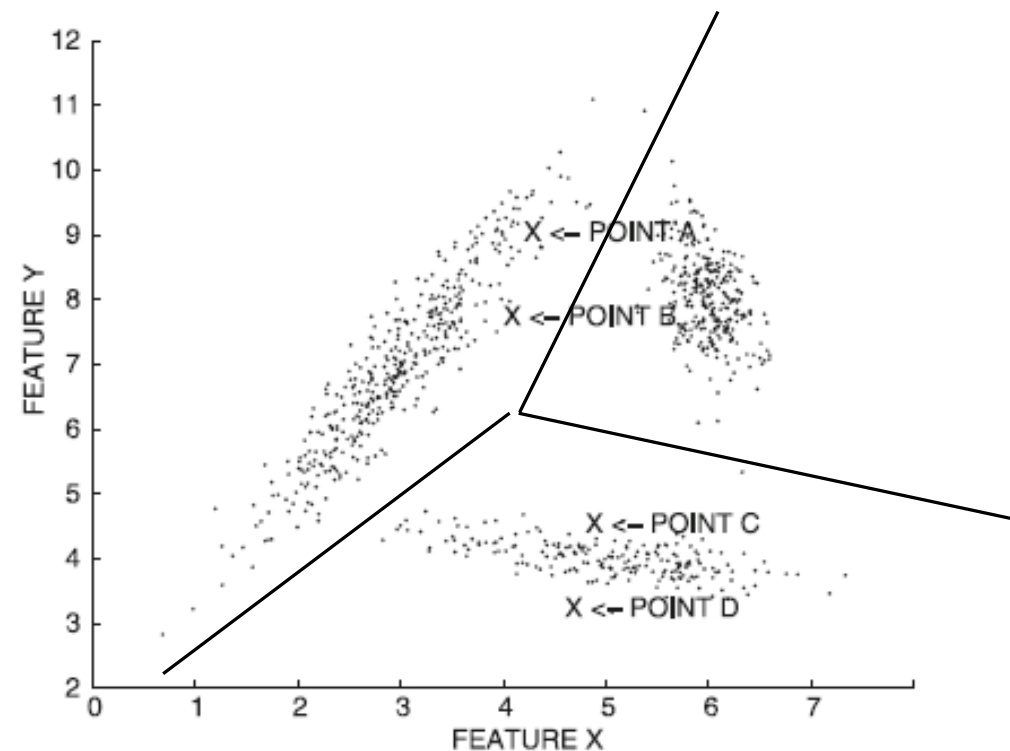
(a) local density variation



# Impact of Local Data Distribution (2)

## □ Local Density Variation

- $C-D$  should be longer than  $A-B$



(b) local orientation variation



# Impact of Local Data Distribution (3)

---

## □ Generic Methods

- Partition the data into a set of local regions (**Nontrivial**)
- 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
  - ✓ Local Mahalanobis distance
- If they belong to different regions
  - ✓ Global statistics or averaged statistics

## □ Shared Nearest-Neighbor Similarity



# Multidimensional Data (Vectors)

---

- Quantitative Data
- **Categorical Data**
- Mixed Quantitative and Categorical Data





# Categorical Data (1)

□ Given  $\bar{X} = (x_1 \dots x_d)$  and  $\bar{Y} = (y_1 \dots y_d)$

- Sum of similarities on the individual features

$$Sim(\bar{X}, \bar{Y}) = \sum_{i=1}^d S(x_i, y_i).$$

- The simplest  $S(x_i, y_i)$

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

- ✓ Ignore the relative frequencies
  - Two documents containing "Science" is less similar than two documents containing "Data Mining"



# Categorical Data (2)

□ Given  $\bar{X} = (x_1 \dots x_d)$  and  $\bar{Y} = (y_1 \dots y_d)$

- Sum of similarities on the individual features

$$Sim(\bar{X}, \bar{Y}) = \sum_{i=1}^d S(x_i, y_i).$$

- Inverse occurrence frequency

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

- ✓  $p_i(x_i)$  is the fraction of records in which the  $i$ -th feature takes on the value of  $x_i$



# Categorical Data (3)

□ Given  $\bar{X} = (x_1 \dots x_d)$  and  $\bar{Y} = (y_1 \dots y_d)$

- Sum of similarities on the individual features

$$Sim(\bar{X}, \bar{Y}) = \sum_{i=1}^d S(x_i, y_i).$$

- Goodall measure

$$S(x_i, y_i) = \begin{cases} 1 - p_i(x_i)^2, & x_i = y_i \\ 0, & x_i \neq y_i \end{cases}$$

- ✓  $p_i(x_i)$  is the fraction of records in which the  $i$ -th feature takes on the value of  $x_i$



# Multidimensional Data (Vectors)

---

- Quantitative Data
- Categorical Data
- **Mixed Quantitative and Categorical Data**



# Mixed Quantitative and Categorical Data

- Given  $\overline{X} = (\overline{X}_n, \overline{X}_c)$  and  $\overline{Y} = (\overline{Y}_n, \overline{Y}_c)$ 
  - Where  $\overline{X}_n, \overline{Y}_n$  are the subsets of numerical attributes and  $\overline{X}_c, \overline{Y}_c$  are the subsets of categorical attributes
  - Weighted Average

$$Sim(\overline{X}, \overline{Y}) = \lambda \cdot NumSim(\overline{X}_n, \overline{Y}_n) + (1 - \lambda) \cdot CatSim(\overline{X}_c, \overline{Y}_c)$$

✓  $\lambda$  is difficult to decide

- Normalized Weighted Average

$$Sim(\overline{X}, \overline{Y}) = \lambda \cdot NumSim(\overline{X}_n, \overline{Y}_n) / \sigma_n + (1 - \lambda) \cdot CatSim(\overline{X}_c, \overline{Y}_c) / \sigma_c.$$



# Outline

---

- Introduction
- Multidimensional Data
- **Text Similarity Measures**
- Temporal Similarity Measures
- Graph Similarity Measures
- Supervised Similarity Functions
- Summary



# Text Similarity Measures (1)

---

## □ As Quantitative Multidimensional Data

- Bag of words model
- It is very **sparse**
- $L_p$ -norm does not work well
  - ✓ Long documents have long distance

## □ Dimensionality Reduction (A Possible Solution)

- Latent Semantic Analysis (SVD)
- $L_p$ -norm in the new space



# Text Similarity Measures (2)

## □ Cosine Similarity

- The angle between two documents

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

- Ignore the relative frequencies
  - ✓ Two documents containing "Science" is less similar than two documents containing "Data Mining"





# Text Similarity Measures (3)

## □ Cosine Similarity with TF-IDF

- Inverse document frequency

$$idf_i = \log(n/n_i).$$

where  $n_i$  is number of documents in which the  $i$ -th word occurs

- A damping function **may be** applied to term frequencies

$$f(x_i) = \sqrt{x_i}$$

$$f(x_i) = \log(x_i)$$

- ✓ The excessive presence of single word does not throw off the similarity measure



# Text Similarity Measures (4)

## □ Cosine Similarity with TF-IDF

- Normalized frequency for the  $i$ -th word

$$h(x_i) = f(x_i) \cdot id_i.$$

- Then, we define

$$\cos(\bar{X}, \bar{Y}) = \frac{\sum_{i=1}^d h(x_i) \cdot h(y_i)}{\sqrt{\sum_{i=1}^d h(x_i)^2} \cdot \sqrt{\sum_{i=1}^d h(y_i)^2}}.$$

## □ Jaccard coefficient

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



# Binary and Set Data

□ Given  $\bar{X} = (x_1, \dots, x_d)$  and  $\bar{Y} = (y_1, \dots, y_d)$   
with  $x_i, y_i \in (0,1)$

■ They can be treated as vector  
representations of two sets

$$S_X = \{i | x_i = 1\}$$

$$S_Y = \{i | y_i = 1\}$$

■ Jaccard coefficient

$$J(\bar{X}, \bar{Y}) = \frac{\sum_{i=1}^d x_i \cdot y_i}{\sum_{i=1}^d x_i^2 + \sum_{i=1}^d y_i^2 - \sum_{i=1}^d x_i \cdot y_i} = \frac{|S_X \cap S_Y|}{|S_X \cup S_Y|}.$$



# Outline

---

- Introduction
- Multidimensional Data
- Text Similarity Measures
- **Temporal Similarity Measures**
- Graph Similarity Measures
- Supervised Similarity Functions
- Summary



# Temporal Similarity Measures

---

- Temporal data
  - Continuous time series
  - Discrete sequences

# Time-Series Similarity Measures (1)

---



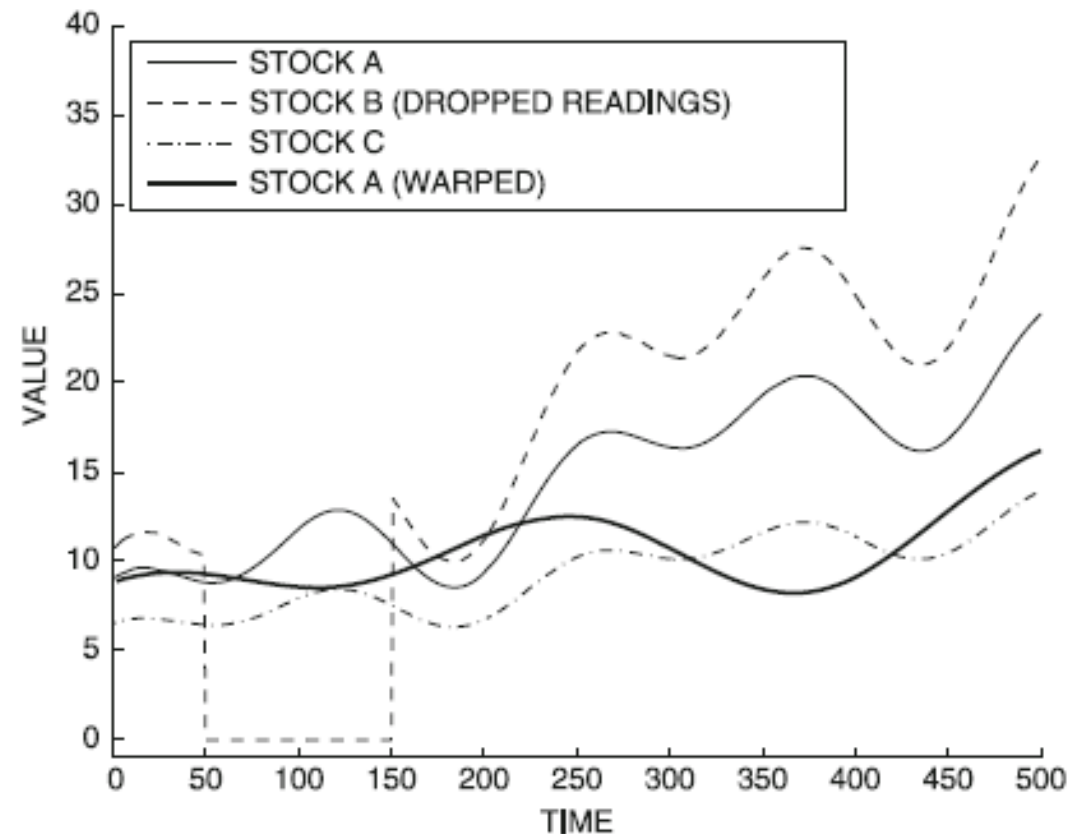
## □ Distortion Factors

- Behavioral attribute scaling and translation
- Temporal (contextual) attribute translation
- Temporal (contextual) attribute scaling
- Noncontiguity in matching

# Time-Series Similarity Measures (2)



## □ Impact of scaling, translation, and noise





# Time-Series Similarity Measures (3)

---

## □ Impact of Behavioral Attribute Normalization

- Behavioral attribute translation
  - ✓ The behavioral attribute is mean centered
- Behavioral attribute scaling
  - ✓ The standard deviation is scaled to 1

## □ $L_p$ -Norm

$$Dist(\bar{X}, \bar{Y}) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

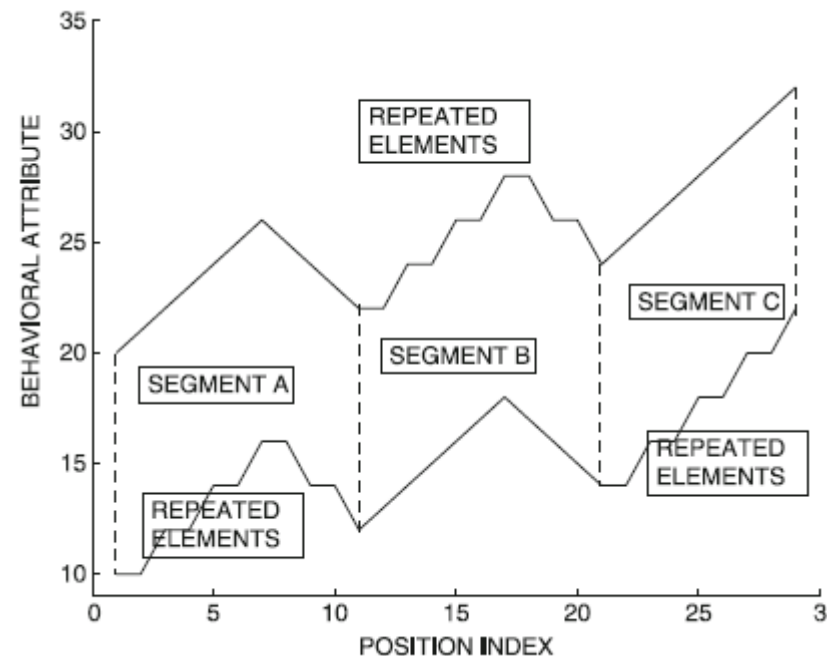
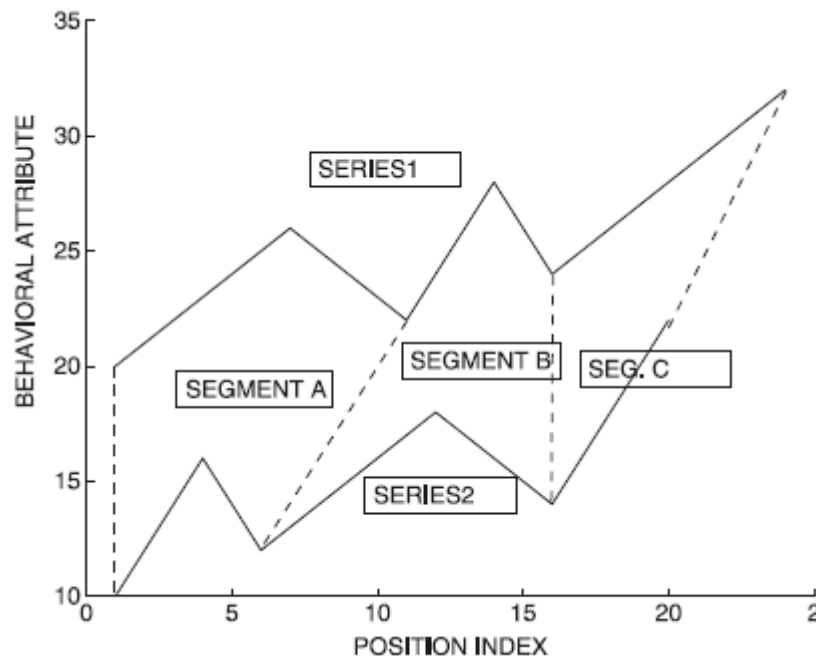
- Combined with wavelet transformations



# Dynamic Time Warping Distance (1)



## □ Address contextual attribute scaling



## □ Can be used either for time-series or sequence data



# Dynamic Time Warping Distance (2)

□ Given  $\bar{X} = (x_1 \dots x_m)$  and  $\bar{Y} = (y_1 \dots y_n)$

■ The two series have different lengths

□  $DTW(i, j)$

■ The distance between the first  $i$  elements of  $\bar{X}$  and the first  $j$  elements of  $\bar{Y}$

□ An Recursive Definition

$$DTW(i, j) = distance(x_i, y_j) + \min \begin{cases} DTW(i, j-1) & \text{repeat } x_i \\ DTW(i-1, j) & \text{repeat } y_j \\ DTW(i-1, j-1) & \text{repeat neither} \end{cases}$$

# Dynamic Time Warping Distance (3)



## □ Implementation

### ■ Recursive computer program

$$DTW(i, j) = distance(x_i, y_j) + \min \begin{cases} DTW(i, j-1) & \text{repeat } x_i \\ DTW(i-1, j) & \text{repeat } y_j \\ DTW(i-1, j-1) & \text{repeat neither} \end{cases}$$

### ■ Nested Loop

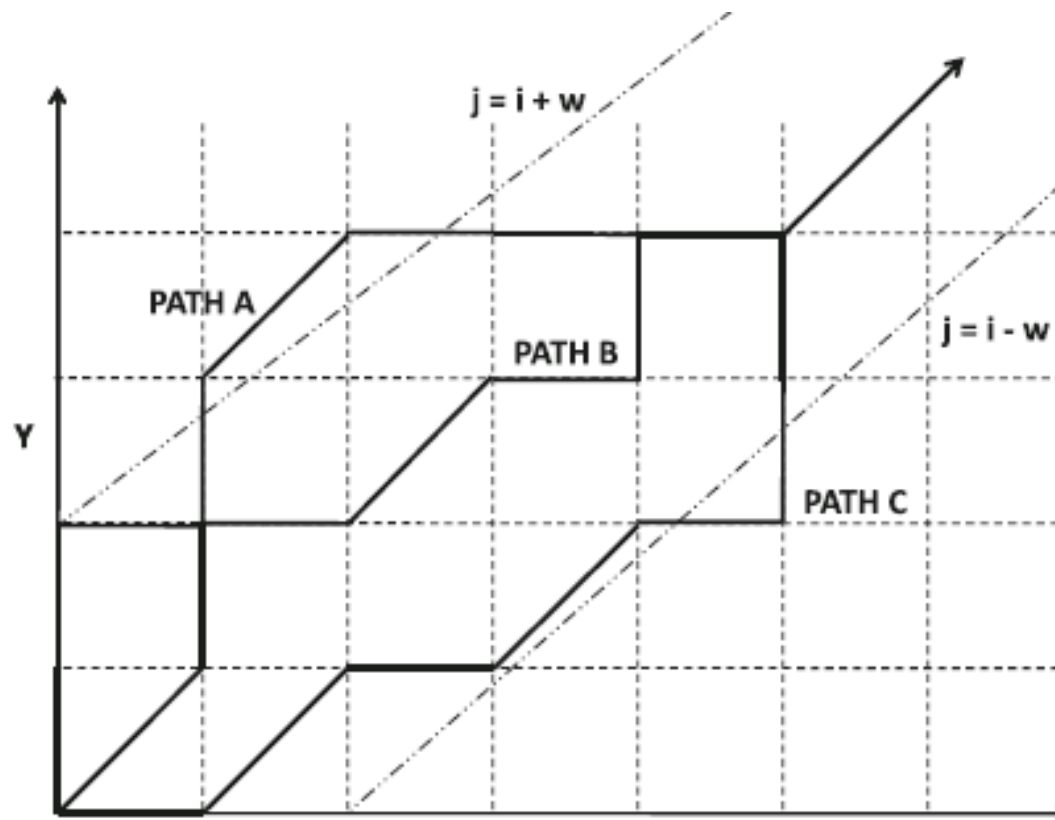
```
for  $i = 1$  to  $m$ 
  for  $j = 1$  to  $n$ 
    compute  $DTW(i, j)$  using Eq. 3.18
```

Equation 3.18 yields a natural iterative approach. The approach starts by initializing  $DTW(0,0)$  to 0,  $DTW(0,j)$  to  $\infty$  for  $j \in \{1 \dots n\}$ , and  $DTW(i,0)$  to  $\infty$  for  $i \in \{1 \dots m\}$ . The algorithm computes  $DTW(i, j)$  by repeatedly executing Eq. 3.18 with increasing index values of  $i$  and  $j$ . This can be achieved by a simple nested loop in which the indices  $i$  and  $j$  increase from 1 to  $m$  and 1 to  $n$ , respectively

# Dynamic Time Warping Distance (3)



□ Optimal Warping = Optimal Path





# Temporal Similarity Measures

---

- Temporal data
  - Continuous time series
  - **Discrete sequences**



# Edit Distance (1)

---

## □ Edit Distance of Two Sequences

- The cost of “edits” to transfer the first one to the second one

## □ Edits

- Insertions
- Deletions
- Replacements

## □ Sequence abababab to babababa

- 8 Replacements
- 1 Deletion+1 Insertion



## Edit Distance (2)

□ Two Sequences  $\bar{X} = (x_1 \dots x_m)$  and  $\bar{Y} = (y_1 \dots y_n)$

■  $Edit(\bar{X}, \bar{Y})$  may not be the same as  $Edit(\bar{Y}, \bar{X})$

□  $Edit(i, j)$

■ The edit distance between the first  $i$  symbols of  $\bar{X}$  and the first  $j$  symbols of  $\bar{Y}$

□ An Recursive Definition

$$Edit(i, j) = \min \begin{cases} Edit(i-1, j) + \text{Deletion Cost} \\ Edit(i, j-1) + \text{Insertion Cost} \\ Edit(i-1, j-1) + I_{ij} \cdot (\text{Replacement Cost}) \end{cases}$$

Furthermore, the bottom of the recursion also needs to be set up. The value of  $Edit(i, 0)$  is equal to the cost of  $i$  deletions for any value of  $i$ , and that of  $Edit(0, j)$  is equal to the cost of  $j$  insertions for any value of  $j$ . This nicely sets up the dynamic programming approach.

# Longest Common Subsequence (LCSS)



□ LCSS of  $\bar{X} = (x_1 \dots x_m)$  and  $\bar{Y} = (y_1 \dots y_n)$ :

■ Length of the longest common  
subsequence

□  $LCSS(i, j)$

■ The LCSS between the first  $i$  symbols of  $\bar{X}$  and the first  $j$  symbols of  $\bar{Y}$

□ An Recursive Definition

$$LCSS(i, j) = \max \begin{cases} LCSS(i-1, j-1) + 1 & \text{only if } x_i = y_j \\ LCSS(i-1, j) & \text{otherwise (no match on } x_i) \\ LCSS(i, j-1) & \text{otherwise (no match on } y_j) \end{cases}$$

Furthermore, the boundary conditions need to be set up. The values of  $LCSS(i, 0)$  and  $LCSS(0, j)$  are always equal to 0 for any value of  $i$  and  $j$ . As in the case of the DTW and edit-distance computations, a nested loop can be set up to compute the final value.





# Outline

---

- Introduction
- Multidimensional Data
- Text Similarity Measures
- Temporal Similarity Measures
- **Graph Similarity Measures**
- Supervised Similarity Functions
- Summary

# Similarity between Two Nodes in a Single Graph

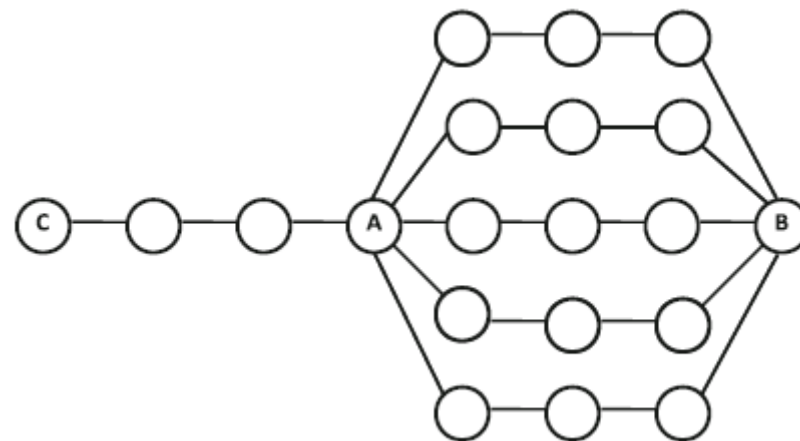


## □ Structural Distance-Based Measure

- Shortest-path on the graph
- Dijkstra algorithm

## □ Random Walk-Based Similarity

- Accounts for multiplicity in paths during similarity computation





# Similarity Between Two Graphs

---

## □ Extremely Challenging

- Even the graph isomorphism problem is NP-hard

## □ Possible Solutions

- Maximum common subgraph distance
- Substructure-based similarity
- Graph-edit distance
- Graph kernels



# Outline

---

- Introduction
- Multidimensional Data
- Text Similarity Measures
- Temporal Similarity Measures
- Graph Similarity Measures
- **Supervised Similarity Functions**
- Summary



# Supervised Similarity Functions

## □ 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 function that fits the feedback

### ■ Find parameter $\Theta$ to minimize

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

where  $f(O_i, O_j, \Theta)$  is a distance function with parameter  $\Theta$



# Outline

---

- Introduction
- Multidimensional Data
- Text Similarity Measures
- Temporal Similarity Measures
- Graph Similarity Measures
- Supervised Similarity Functions
- **Summary**



# Summary

---

- Multidimensional Data
  - $L_p$ -Norm, Generalized Minkowski distance
  - Match-Based Similarity Computation
  - Mahalanobis distance, Geodesic distances
  - Inverse Occurrence Frequency
- Text Similarity Measures
  - Cosine, TF-IDF
- Temporal Similarity Measures
  - Dynamic Time Warping
  - Edit Distance, Longest Common Subsequence
- Graph Similarity Measures
  - Shortest-path, Random Walk
- Supervised Similarity Functions