Similarity and Distances

Data Mining: The Textbook, Chapter 3
Presented by Zachary Stine, 22 June 2018

Agenda

I. Introduction
II. Multidimensional Data
III. Text Similarity Measures
IV. Temporal Similarity Measures
V. Graph Similarity Measures
VI. Supervised Similarity Functions
VII. Summary

I. Introduction

I. Introduction

- Data mining requires a methodical way of quantifying similarity and dissimilarity.
- Distance and similarity serve the same purpose
  - High similarity implies small distance
  - Low similarity implies large distance
- Distance functions are highly sensitive to
  - The distribution of the data
  - The dimensionality of the data
  - The type of data
- A data mining algorithm is only as good as its distance function.
- *Never assume that Euclidean distance is the appropriate distance function to use!*

II. Multidimensional Data

A. *Quantitative Data*

L_p-norm

- Most common distance function for quantitative data.
- Special cases:
  - p = 2 → Euclidean distance
  - p = 1 → Manhattan distance
  - p = ∞ → absolute value of the dimension, *i,* for which X and Y are the most distant from each other.
- Euclidean and Manhattan distance are easily interpreted in spatial applications:
  - Euclidean: straight line between two points.
  - Manhattan: “City block” driving distance between two points on a grid.
- *Assumption: all features are equally important*

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

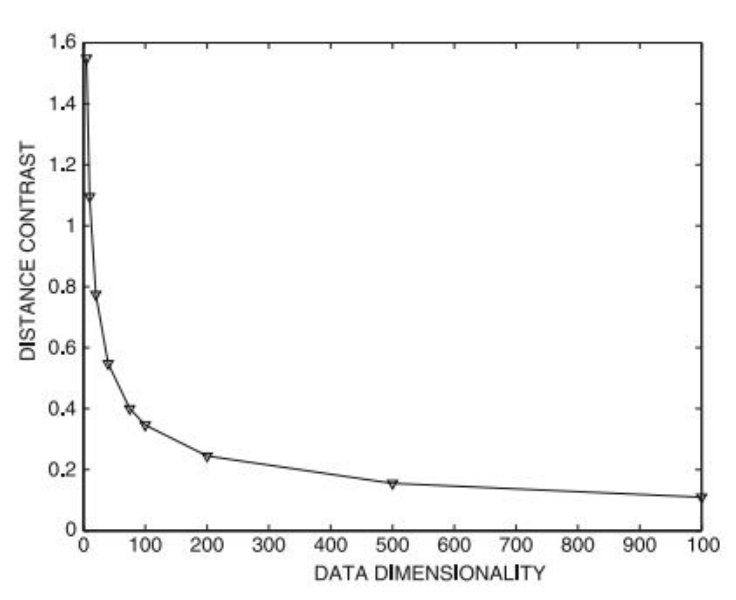
Impact of Domain-Specific Relevance

- Domain knowledge → are some features more important than others?
- If yes, assign a weight, a_i, to each dimension.
- *Generalized Minkowski distance*

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

weight

Impact of High Dimensionality

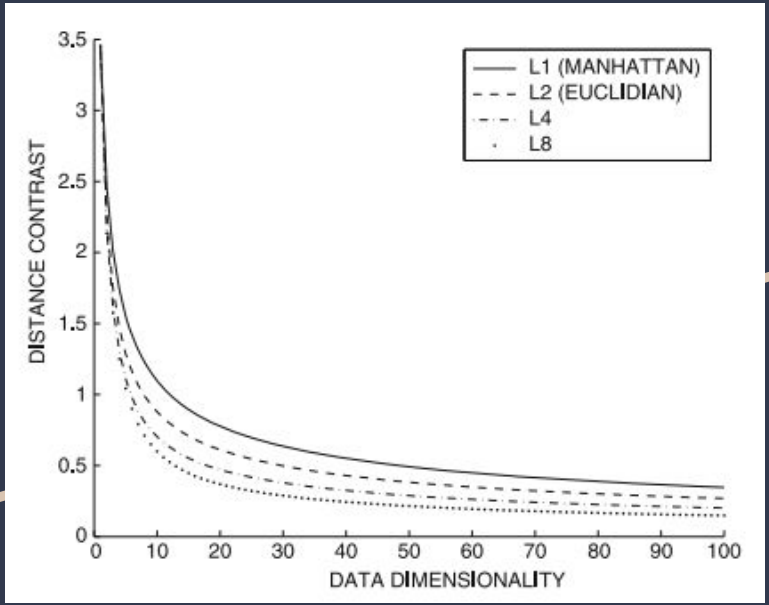


- As the # of dimensions increases, the usefulness of distance-based data mining applications decreases
- “Curse of Dimensionality”
- Example: distance-based clustering may group unrelated data points if the dimensionality is sufficiently large.
- *Contrast* → ratio of variation in the distances to the absolute values, i.e. how useful distance is at discriminating between objects that are different.

Impact of Locally Irrelevant Features

- Some features may be generally relevant to a distance function
- However, the relevance of some features may change depending on the specific pair of objects being compared
- Global dimensionality reduction will not work in this case, since feature relevance is determined locally (i.e., between specific pairs of objects)

Impact of Different L_p-Norms



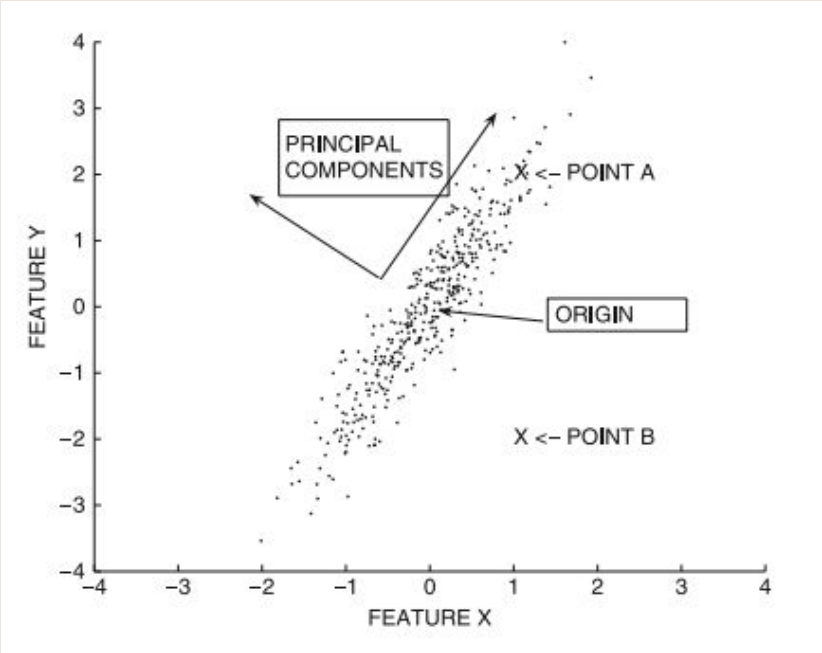
- Larger p will emphasize irrelevant features
- Example: X and Y have 1,000 dimensions and are equal on 999 dimensions, but very different on one.
  - Should probably be considered very similar
  - If p = ∞, considered completely different because of the one dimension where they differ.
- General rule: the larger the dimensionality, the smaller value of p to use
- Can also use fractional metrics where p is between 0 and 1
- Precise choice of p requires benchmarking in application-specific way

Match-Based Similarity Computation

- How to select locally relevant features for distance computations?
- Need to de-emphasize impact of noisy variation along individual attributes, while counting the cumulative match across dimensions
- Rationale: a pair of objects are unlikely to have similar values across many attributes just by chance, unless those attributes are locally relevant
- One way of doing this is proximity thresholding,
  - Divide each dimension into k_d equidepth buckets (each containing $1/k_d$ of the records)
  - For dimension i, if x_i and y_i are in the same bucket, they are in proximity on i
  - $S(X, Y, k_d)$ is the set of dimensions on which X and Y are in proximity
  - Formula is used which emphasizes these dimensions in computing distance

Should distances depend on the underlying data distribution of all data points?

YES

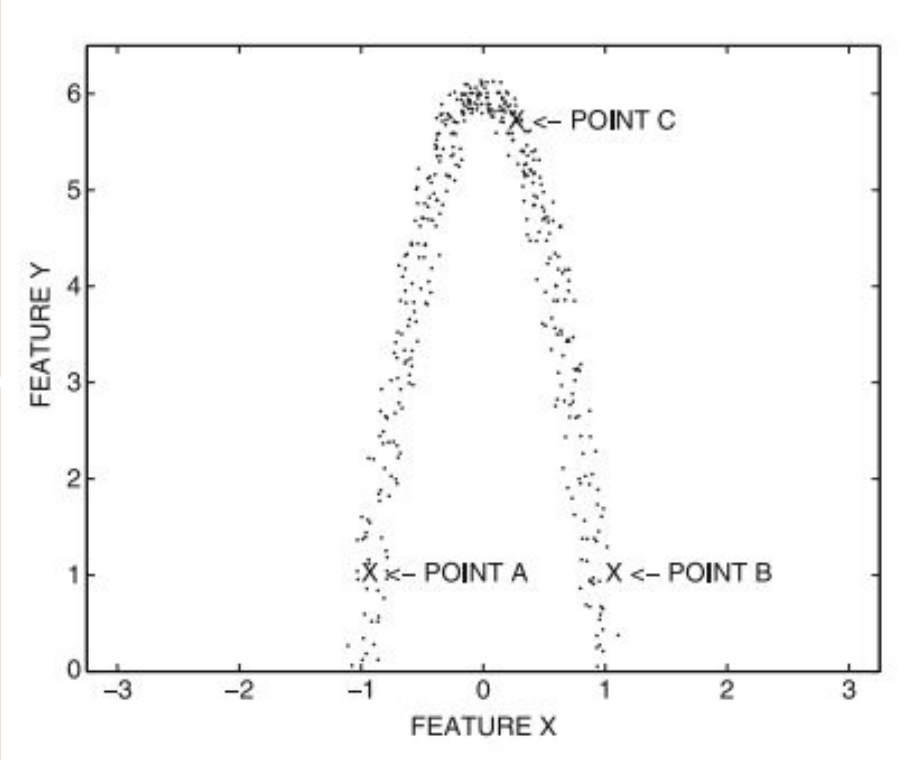

PRINCIPAL
COMPONENTS
X <– POINT A
ORIGIN
X <– POINT B
FEATURE Y
FEATURE X
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

Impact on Data Distribution

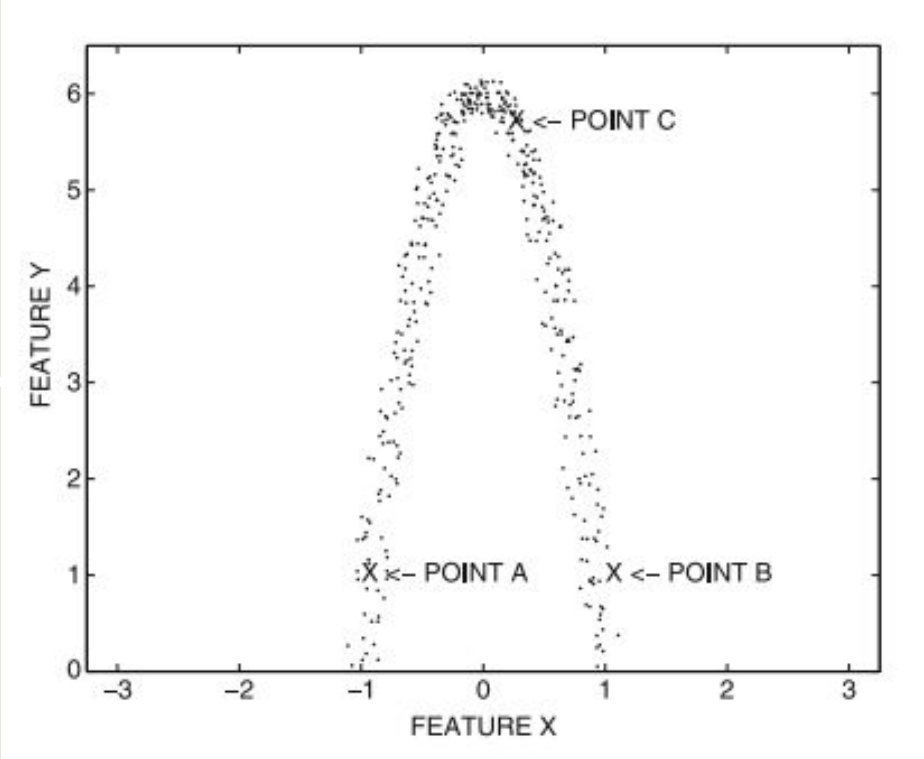
- L_p-norm only depends on the two points being compared; it completely ignores the global statistics of all the other data points
- Distances should depend on the underlying distribution of all data points
- Mahalanobis distance does this by including the covariance matrix of the data set
- Can think of as Euclidean distance that takes principal components into account

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

Which two points are the closest together?

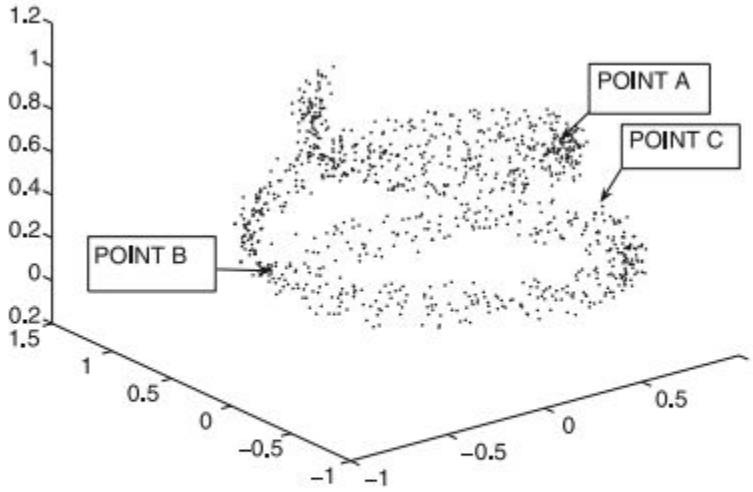


Euclidean: A and B
Geodesic: B and C

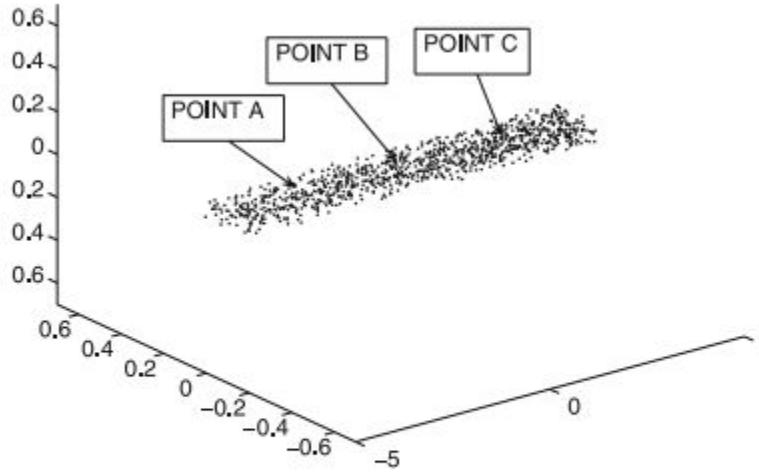


Nonlinear Distributions: Isomap

- Geodesic distances measure the shortest path length using point-to-point distance from each point to one of their k-nearest neighbors
- Point-to-point distance can be a standard metric like Euclidean distance.
- *Assumption: nonlinear distributions are locally Euclidean, but not globally Euclidean.*
- Isomap is a method for nonlinear dimensionality-reduction and embedding:
  - Step 1: Construct graph with edges connecting k-nearest neighbors weighted by distances
  - Step 2: For any pair of points, X and Y, distance is measured from the shortest path between the corresponding nodes in the graph
- Multidimensional scaling (MDS) can then be used to embed data in lower-dimensional space



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



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

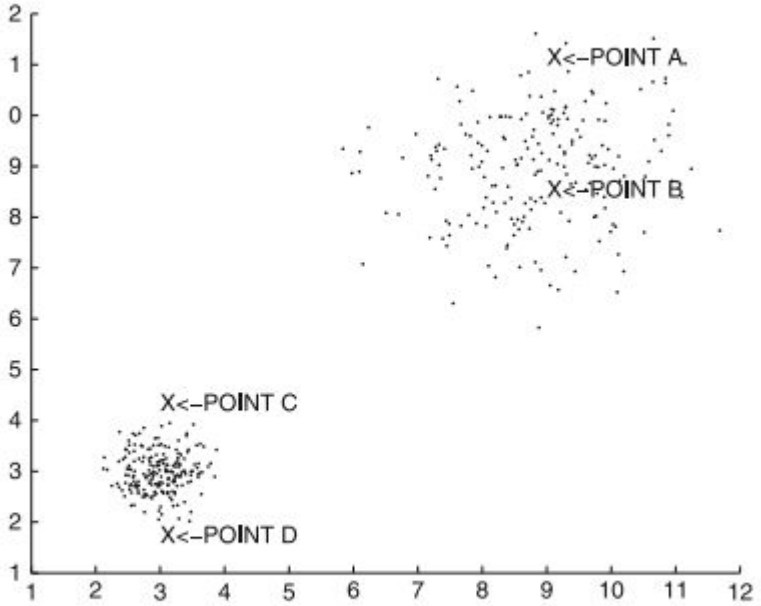
Example of nonlinear distribution being “flattened” via Isomap into lower-dimensional space.

Impact of Local Data Distribution

- Distribution of data may vary locally
- Two types of variation:
  - Local density variation
  - Local orientation variation (cluster shape)
- Methods that adjust for local variations often perform better.
- Local Outlier Factors (LOF) is an example of such a method.
- Two approaches
  - Shared nearest neighbor similarity
  - Generic methods

Shared Nearest Neighbor Similarity

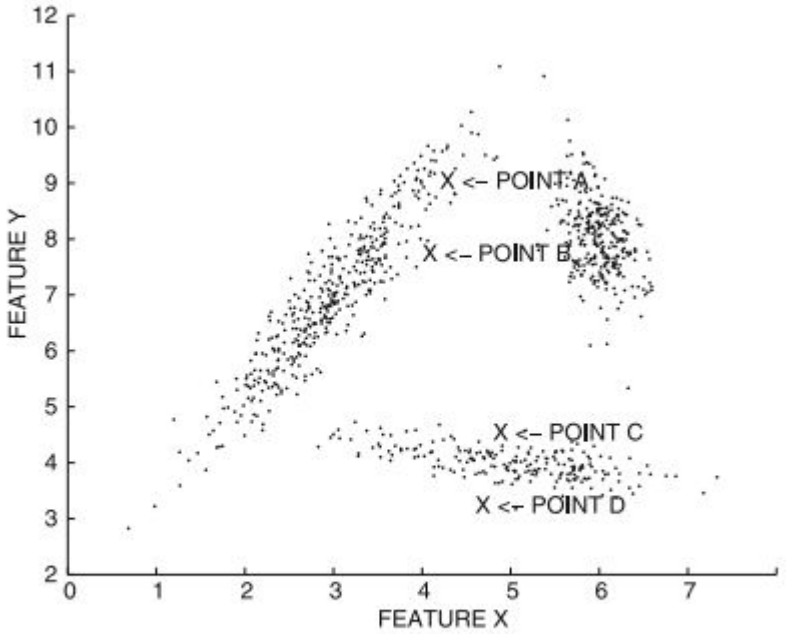
- k-nearest neighbors of each data point are computed.
- Similarity is equal to the number of common neighbors between two data points.
- Locally sensitive → does not care about absolute values of distance, simply shared neighbors.
- The more shared neighbors, the more similar the points are.
- C and D are less similar than A and B, despite having equivalent Euclidean distances.



(a) local density variation

Generic Methods

- Divide space into local regions
- Use the local statistics of each region to adjust distances.
- Have to determine the most relevant region to use for a given pair of points
- Extra steps needed if comparing points from different regions
- However, dividing the space into regions requires the use of a meaningful distance metric for clustering. Requires iterative solutions.



(b) local orientation variation

II. Multidimensional Data
B. *Categorical Data*

Categorical Data

- Since categorical data is not ordered, can't use the previous distance measures.
- Simplest categorical similarity is to count the number of elements in X and Y that are equal; the "overlap" measure.
- Overlap measure does not account for relative frequencies among different attributes.
  - Example: attribute can be 'Normal' or 'Diabetic'; 99% of the records have 'Normal' for this attribute.
  - If two records both have 'Normal', no significant information about the similarity between them
  - If two records both have 'Diabetic', then much better evidence for similarity
- Thus, need to account for $p_k(x) \rightarrow$ the fraction of records in which the k^{th} attribute is x.
  - Inverse occurrence frequency
  - Goodall measure

Using Aggregate Statistical Properties

- Instead of summing 1s and 0s (overlap method), need to incorporate $p_k(x)$: the fraction of records in which the k^{th} attribute is x.
- Inverse occurrence frequency

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

- Goodall measure

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

Mixed Quantitative and Categorical Data

- Need to weight the importance of the quantitative data relative to the categorical data with parameter λ.
- Let record $X = (X_n, X_c)$
  - X_n is the subset of attributes that are numerical
  - X_c is the subset of attributes that are categorical
- Need two similarity measures
  - NumSim(X, Y) for numerical similarity
  - CatSim(X, Y) for categorical similarity
- Then add them together, weighted by λ:

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

III. Text Similarity Measures

Text Data

- Text can be treated as quantitative data if using bag-of-words representations.
- Documents will be sparse; lots of 0s.
- All elements will be nonnegative.
- L_p-norm will not work well: measures size of documents more than similarity.
- Cosine measure does better because it is insensitive to length of the documents.

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

Using Aggregate Statistical Properties for Text

- If two documents match on an uncommon word, more likely to be similar.
- Can normalize raw word frequencies with inverse document frequency where n_i is the # of documents with the i^{th} word and n is the total # of documents.

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

- Can also use a damping function, $f(x_i)$, to make sure the excessive presence of a word does not throw off the measure
  - $f(x_i) = \sqrt{x_i}$
  - $f(x_i) = \log(x_i)$
- The normalized frequency, $h(x_i)$ and $h(y_i)$ can now be used to improve the cosine measure.

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

IV. Temporal Similarity Measures

Temporal Data

- Temporal data contains
  - Single contextual attribute representing time
  - One or more behavioral attributes measuring properties than change over time
- Time can be discrete or continuous
- Common distortion factors in time series data that metrics like Euclidean cannot account for:
  - Behavioral attribute scaling and translation
  - Temporal (contextual) attribute translation
  - Temporal (contextual) attribute scaling
  - Non-contiguity in matching

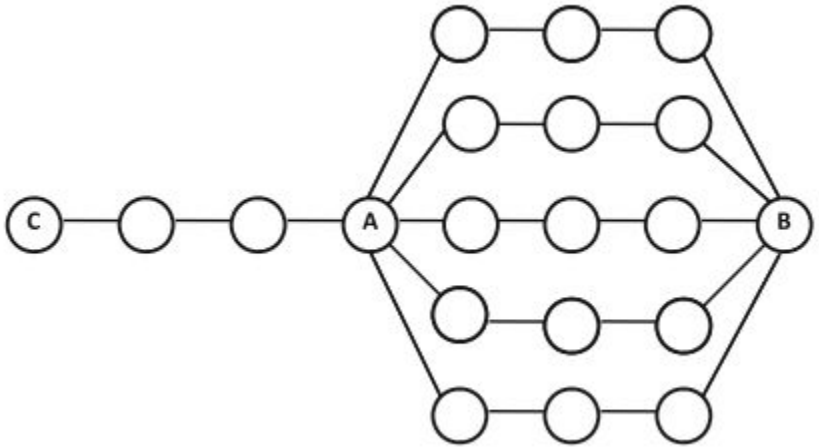
Methods for Dealing with Distortion Factors in Time Series

- Center behavioral attribute on the mean.
- Scale the standard deviation of the behavioral attribute to 1 unit.
- Time warping: stretch or compress the time series
- Dynamic time warping (DTW): time warping at specific intervals in the time series
- Time series of unequal length can be compared by using DTW to force them to the same length
- Window-based methods: divide each time series into segments and measure the similarity between segments of the two time series

V. Graph Similarity Measures

Similarity Between Two Nodes in One Graph

- Structural distance-based measure: shortest path between two nodes (Dijkstra's algorithm).
- Random walk-based measure: measures the multiplicity of paths between two nodes (homophily).
- A and C are closest if using shortest path.
- A and B are closest if using random walk.



Similarity Between Two Graphs

- Can be extremely challenging. Graph isomorphism problem is NP-hard.
- *Maximum common subgraph distance:* If two graphs contain a large, identical subgraph in common, likely to be similar.
- *Substructure-based similarity:* Count frequently occurring substructures.
- *Graph-edit distance:* Number of operations needed to transform one graph into the other.
- *Graph kernels*: see chapter 17.

VI. Supervised Similarity Functions

Supervised Similarity Functions

- Can incorporate feedback if one knows which pairs of objects are similar and dissimilar.

$$\mathcal{S} = \{(O_i, O_j) : O_i \text{ is similar to } O_j\}$$
$$\mathcal{D} = \{(O_i, O_j) : O_i \text{ is dissimilar to } O_j\}.$$

- Can use a learning algorithm to learn optimal weights for weighted L_p-norm, where Θ is the set of weights that will be learned.
- Distance function: $f(O_i, O_j, \Theta)$
- Goal is to learn Θ so that the following is met:

$$f(O_i, O_j, \Theta) = \begin{cases} 0 & \text{if } (O_i, O_j) \in \mathcal{S} \\ 1 & \text{if } (O_i, O_j) \in \mathcal{D} \end{cases}.$$

VII. Summary

Summary

- Distance and similarity measures have assumptions that are built into them and mean different things. If you don’t know what these assumptions are and what they mean, you probably don’t know what you are actually measuring.
- Distance measures are sensitive to
  - The type of data
  - The dimensionality of data
  - The global and local nature of data distribution
- *Distance and similarity measures should be selected with care.*

Thanks!