# Proximity and Data Pre-processing

Huiping Cao

# Outline

- Types of data
- Data quality
- Measurement of proximity
- Data preprocess

# Similarity and Dissimilarity

- Dissimilarity (distance)
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Similarity
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Proximity refers to either a similarity or dissimilarity

# Similarity and Dissimilarity (cont.)

- Simple attributes, multiple attributes
  - E.g., Dense data: Euclidean distance
  - E.g., Sparse data: Jaccard and Cosine similarity

# Similarity/Dissimilarity for Simple Attributes

- $p$ and $q$ are the attribute values for two data objects.

| Attribute type | Dissimilarity | Similarity |
|---|---|---|
| Nominal | $d = \begin{cases} 0 \;\; if \;\; p = q \\ 1 \;\; if \;\; p \neq q \end{cases}$ | $s = \begin{cases} 1 \;\; if \;\; p = q \\ 0 \;\; if \;\; p \neq q \end{cases}$ |
| Ordinal | $d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n$-1, where $n$ is the number of values) | $s = 1 - \frac{\lvert p-q \rvert}{n-1}$ |
| Interval or Ratio | $d = \lvert p - q \rvert$ | $s = -d$, $s = \frac{1}{d+1}$ or $s = 1 - \frac{d - min_d}{max_d - min_d}$ |

where *min_d* and *max_d* are the minimum and maximum distances between every two values.

# Similarity and Dissimilarity – multiple attributes

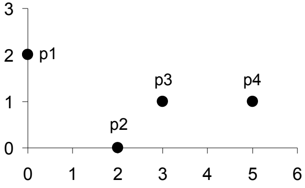
- Euclidean, Minkowski, DTW, etc.
- SMC, Jaccard, Cosine, Hamming, etc.

# Euclidean Distance

$$dist = \sqrt{\sum_{i=1}^{n} (q_i - q'_i)^2}$$

- $n$ is the number of dimensions (attributes)
- $q_i$ and $q'_i$: the $i$th attributes (components) of data objects $q$ and $q'$.
- Standardization is necessary, if scales differ.

# Euclidean Distance – Example




| 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 |

**Distance Matrix**

# Manhattan Distance

- Considered by Hermann Minkowski in 19th century Germany.

$$dist = (\sum_{i=1}^{n} |q_i - q_i'|)$$

- Taxi cab metric/distance, rectilinear distance, Minkowski's $L_1$ norm/distance, city block distance, or Manhattan length.
- Parameters
  - $n$ is the number of dimensions (attributes)
  - $q_i$ and $q_i'$ are, respectively, the $i$th attributes (components) of data objects $q$ and $q'$.
- Applications
  - In chess, the distance between squares on the chessboard for rooks is measured in Manhattan distance.
  - The length of the shortest path a taxi could take between two intersections equal to the intersections' distance in taxicab geometry.
  - Integrated circuits where wires only run parallel to the X or Y axis.

# Minkowski Distance

- The Minkowski distance is a metric on Euclidean space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance.

$$dist = (\sum_{i=1}^{n} |q_i - q_i'|^p)^{\frac{1}{p}}$$

- Parameters
  - $p$ is a parameter
  - $n$ is the number of dimensions (attributes)
  - $q_i$ and $q_i'$ are, respectively, the $i$th attributes (components) of data objects $q$ and $q'$.

# Minkowski Distance: Illustration

- $p = 1$. Manhattan distance.
- $p = 2$. Euclidean distance.
- $p \to \infty$. Supremum, Chebyshev ($L_{max}$ norm, $L_\infty$ norm) distance.

$$\lim_{p\to\infty}(\sum_{i=1}^{n}|q_i - q_i'|^p)^{\frac{1}{p}} = max_{i=1}^{n}|q_i - q_i'|$$

Kings and queens use Chebyshev distance in Chess.

- $L_{min}$ norm

$$\lim_{p\to-\infty}(\sum_{i=1}^{n}|q_i - q_i'|^p)^{\frac{1}{p}} = min_{i=1}^{n}|q_i - q_i'|$$

**Note**: Do not confuse $p$ with $n$, i.e., all these distances are defined for all numbers of dimensions.

# Minkowski Distance– Example

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

| L1 | p1 | p2 | p3 | p4 |
|---|---|---|---|---|
| p1 | 0 | 4 | 4 | 6 |
| p2 | 4 | 0 | 2 | 4 |
| p3 | 4 | 2 | 0 | 2 |
| p4 | 6 | 4 | 2 | 0 |

| L2 | 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 |

| $L_\infty$ | p1 | p2 | p3 | p4 |
|---|---|---|---|---|
| p1 | 0 | 2 | 3 | 5 |
| p2 | 2 | 0 | 1 | 3 |
| p3 | 3 | 1 | 0 | 2 |
| p4 | 5 | 3 | 2 | 0 |

**Distance Matrix**

# Issues to Consider in Calculating Distance

- Attributes have different scales
- Attributes are correlated
- Objects are composed of different types of attributes (Qualitative vs. quantitative)
- Attributes have different weights

# Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.
  - Positivity:
    $d(p, q) \geq 0$ for all $p$ and $q$;
    $d(p, q) = 0$ only if $p = q$.
  - Symmetry: $d(p, q) = d(q, p)$ for all $p$ and $q$.
  - Triangle Inequality: $d(p, r) \leq d(p, q) + d(q, r)$ for all points $p$, $q$, and $r$.
- A distance that satisfies these properties is a metric.

# Elastic distances

- Dynamic Time Warping (DTW)
- Edit distance based measure
    - Longest Common SubSequence (LCSS)
    - Edit Distance on Real Sequence (EDR)

# Elastic distances - DTW

- In time series analysis, Dynamic Time Warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed.
- DTW is a technique to find an optimal alignment[1] between time series if one time series can be warped non-linearly by stretching or shrinking it.
- Applications: similarities in walking patterns, video, audio, etc.
- Calculates an optimal match between two given sequences (e.g., time series) with certain restrictions

[1] M. Müller, Information Retrieval for Music and Motion, Springer, 2007, ISBN: 978-3-540-74047-6, http://www.springer.com/978-3-540-74047-6

# Elastic distances - DTW - Problem definition

- Given a reference sequence $s[1, \cdots, n]$ and a query sequence $t[1, \cdots, m]$ and local distance measure $dist(s[i], t[j])$
- Goal: find an alignment between $s$ and $t$ having minimal overall cost
- An $(n,m)$-warping path (or simply referred to as warping path if $n$ and $m$ are clear from the context) is a sequence $p{=}(p_1, \cdots, p_L)$ with $p_i = (n_i, m_i) \in [1 : n] \times [1 : m]$ for $i$ $(1 \leq i \leq L)$ satisfying the following three conditions
  - Boundary condition: $p_1{=}(1{,}1)$ and $p_L{=}(\mathrm{n{,}m})$
  - Monotonicity condition: $n_1 \leq n_2 \leq \cdots \leq n_L$ and $m_1 \leq m_2 \leq \cdots \leq m_L$
  - Step size condition: $p_{i+1} - p_i$ in $\{(1{,}0){,}(0{,}1){,}(1{,}1)\}$ for $i$ $(1 \leq i \leq L-1)$

# Elastic distances - DTW - Distance

- Total cost of a warping path $p=(p_1, \cdots, p_L)$ between $s$ and $t$ is

$$cost_p(s,t) = \sum_{i=1}^{L} dist(p_i) = \sum_{i=1}^{L} dist(s[n_i], t[m_i])$$

  where $p_i = (n_i, m_i)$.

- Let $p^*$ be an optimal warping path between $s$ and $t$

- The DTW distance DTW($s$,$t$) between $s$ and $t$ is defined as the total cost of $p^*$.

$$\begin{aligned} DTW(s,t) &= cost_{p^*}(s,t) \\ &= min\ \{cost_p(s,t) | p \text{ is an (n,m)-warping path}\} \end{aligned}$$

Check algorithm.

# DTW calculation using Python

Reference

```
import numpy as np
from fastdtw import fastdtw
from scipy.spatial.distance import euclidean

x = np.array([1, 2, 3, 3, 7])
y = np.array([1, 2, 2, 2, 2, 2, 2, 4])

distance, path = fastdtw(x, y, dist=euclidean)

print(distance)
print(path)

# 5.0
# [(0, 0), (1, 1), (1, 2), (1, 3), (1, 4), (2, 5), (3, 6), (4, 7)]
```

Another good library:

https://pyts.readthedocs.io/en/stable/generated/pyts.metrics.dtw.html

# Common Properties of a Similarity

- Similarities also have some well known properties.
  - Maximum similarity: $s(p, q) = 1$ only if $p = q$
  - Symmetry: $s(p, q) = s(q, p)$ for all $p$ and $q$
- $s(p, q)$ is the similarity between points (data objects) $p$ and $q$.

# Similarity Between Binary Vectors

- Compute similarities using the following quantities
  - $M_{01}$ = the number of attributes where $p$ was 0 and $q$ was 1
  - $M_{10}$ = the number of attributes where $p$ was 1 and $q$ was 0
  - $M_{00}$ = the number of attributes where $p$ was 0 and $q$ was 0
  - $M_{11}$ = the number of attributes where $p$ was 1 and $q$ was 1
- Similarity coefficient
- In the range of [0,1]

# Distance

- Simple Matching Coefficient (SMC)
- Jaccard Coefficient
- Cosine Similarity
- Hamming distance
- Correlation
- Mutual information (MI)

# Simple Matching Coefficient (SMC)

$$SMC = \frac{number\ of\ matches}{number\ of\ attributes} = \frac{M_{11} + M_{00}}{M_{01} + M_{10} + M_{11} + M_{00}}$$

Example:

- $p = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$
- $q = 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$
- $M_{01} = 2,\ M_{10} = 1,\ M_{00} = 7,\ M_{11} = 0$
- $SMC = \frac{7+0}{10} = 0.7$

# Jaccard Coefficient

$$J = \frac{number\ of\ 11\ matches}{number\ of\ attributes - 00\ attribute\ values} = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

- Handle asymmetric binary attributes
- Example:
  - $p = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$
  - $q = 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$
  - $M_{01} = 2,\ M_{10} = 1,\ M_{00} = 7,\ M_{11} = 0$
  - $J = \frac{0}{2+1+0} = 0$

# Cosine Similarity

- If d1 and d2 are two document vectors, then

$$cos(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1|| ||d_2||}$$

  - $\cdot$ indicates vector dot product
  - $||d||$ is the length of vector $d$

- $d_1 = 3\ 2\ 0\ 5\ 0\ 0\ 0\ 2\ 0\ 0$
- $d_2 = 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 2$
- $d_1 \cdot d_2 = 3 + 2 = 5$
- $||d_1|| = \sqrt{42} = 6.481$
- $||d_2|| = \sqrt{7} = 2.646$
- $cos(d1, d2) = \frac{5}{6.481 * 2.646} = 0.2916$

# Hamming distance

- Measure distance among vectors of equal length with nominal values.
- The number of positions at which the corresponding symbols are different.
- Example
  - "toned" and "roses" is 3.
  - 1011101 and 1001001 is 2.
  - 2173896 and 2233796 is 3.
- Named after *Richard Hamming*. It is used in telecommunication to count the number of flipped bits in a fixed-length binary word as an estimate of error, and therefore is sometimes called the signal distance.
- Not suitable for comparing strings of different lengths, or strings where not just substitutions but also insertions or deletions have to be expected

# Correlation

- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, $p$ and $q$, and then take their dot product

$$corr(p,q) = \frac{covariance(p,q)}{std(p) * std(q)} = \frac{s_{pq}}{s_p * s_q}$$

$$covariance(p,q) = s_{pq} = \frac{1}{n-1}\sum_{i=1}^{n}(p_i - \bar{p})(q_i - \bar{q})$$

$$std(p) = s_p = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(p_i - \bar{p})^2}$$

$$std(q) = s_q = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(q_i - \bar{q})^2}$$

$$\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i, \bar{q} = \frac{1}{n}\sum_{i=1}^{n} q_i$$

# Correlation - Example 1

```
x = (-3, 6, 0, 3, -6)
y = (1, -2, 0, -1, 2)
```

Compute the Pearsons correlation
$corr(x, y) = -1$, $x = -3y$

# Correlation - Example 2

```
x = (-3, -2, -1, 0, 1, 2, 3)
y = (9, 4, 1, 0, 1, 4, 9)
```

- Compute the Pearsons correlation $corr(x, y) = 0$, $y = x^2$
- No linear relationship, but nonlinear relationships can still exist.
- Pearson's correlation only measures linear correlation.

# Correlation - Python Example

```
from scipy.stats import pearsonr

p = np.array([1,3,5])
q = np.array([2,3,5])

corr, _ = pearsonr(p, q)
print('Pearsons correlation: %.3f' % corr)

#Pearsons correlation: 0.982
```

# Mutual information

- For measuring nonlinear correlation
- Come from information theory

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

where

- $H(X) = -\sum_{j=1}^{m} P(X = u_j) log_2 P(X = u_j)$, the entropy (average information) of $X$
- $H(Y) = -\sum_{k=1}^{n} P(Y = v_k, Y = v_k) log_2 P(Y = v_k)$, the entropy (average information) of $X$
- $H(X, Y) = -\sum_{j=1}^{m} \sum_{k=1}^{n} P(X = u_j, Y = v_k) log_2 P(X = u_j, Y = v_k)$, the joint entropy of $X$ and $Y$

- Minimum value is 0, representing that the value of one variable tells us nothing about the another.
- Maximum value (no fixed value), representing that one variable completely depend on another.

# Mutual information - Example

```
x = (-3, -2, -1, 0, 1, 2, 3)
y = (9, 4, 1, 0, 1, 4, 9)
```

- Compute the mutual information $I(X, Y) = 1.9502$

# General Approach for Combining Similarities

- Situation: attributes are of different types, an overall similarity is needed
- For the $i$th attribute, compute a similarity $s_i \in [0, 1]$
- Define $\delta_i$ for the $i$th attribute
  - $\delta_i = 0$ (i.e., do not count this attribute)
    if the $i$th attribute is an asymmetric attribute and both objects have a value of 0
    or if one of the objects has a missing value for the $i$th attribute
  - $\delta_i = 1$ otherwise
- Compute the overall similarity

$$similarity(p, q) = \frac{\sum_{i=1}^{n} \delta_i s_i}{\sum_{i=1}^{n} \delta_i}$$

# Using Weights to Combine Similarities

- May not want to treat all attributes the same.
- Use weights $w_i \in [0,1]$ and $\sum_{i=1}^{n} w_i = 1$.

$$similarity(p,q) = \frac{\sum_{i=1}^{n} w_i \delta_i s_i}{\sum_{i=1}^{n} \delta_i}$$

# Sampling

- It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.
- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.

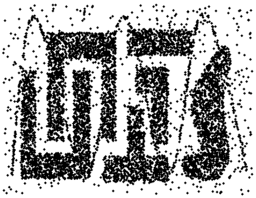
# Key Principle for Effective Sampling

- Using a sample will work almost as well as using the entire data sets, if the sample is representative
- A sample is representative if it has approximately the same property (of interest) as the original set of data

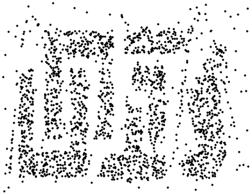
# Types of Sampling

- Simple Random Sampling: there is an equal probability of selecting any particular item
- Variations
  - Sampling without replacement: As each item is selected, it is removed from the population
  - Sampling with replacement: Objects are not removed from the population as they are selected for the sample.
    - In sampling with replacement, the same object can be picked up more than once
- Stratified sampling: Split the data into several partitions; then draw random samples from each partition

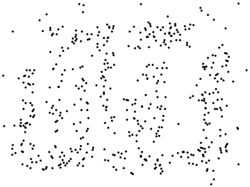
# Sample Size



**8000 points**



**2000 Points**



**500 Points**

# Curse of Dimensionality

- Data analysis becomes significantly harder as the dimensionality of the data increases.

- Data becomes increasingly sparse in the space that it occupies

# Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data mining algorithms
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise
- Techniques
  - Principle Component Analysis (PCA)
  - Singular Value Decomposition (SVD)
  - Others: supervised and non-linear techniques

# Feature Subset Selection

- Another way to reduce dimensionality of data
- Redundant features
  - Duplicate much or all of the information contained in one or more other attributes
  - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
  - Contain no information that is useful for the data mining task at hand
  - Example: students' ID is often irrelevant to the task of predicting students' GPA

# Feature Subset Selection–Techniques (S.S.)

- Brute-force approach:
  - Try all possible feature subsets as input to data mining algorithm
- Embedded approaches:
  - Feature selection occurs naturally as part of the data mining algorithm
- Filter approaches
  - Features are selected before data mining algorithm is run
- Wrapper approaches:
  - Use the data mining algorithm as a black box to find best subset of attributes

# Feature Creation (S.S.)

- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes
- Three general methodologies
  - Feature extraction: domain-specific
  - Mapping data to new space
    - Fourier transform
    - Wavelet transform
  - Feature construction: combining features. E.g., Density in cluster analysis.

- Chapter 2: Introduction to Data Mining (2nd Edition) by Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar
- https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics.pairwise