2.3 Similarity and Distance Measures

MOTIVATION

- Pattern Mining:
  - Help in identifying frequent patterns within datasets
  - By understanding how similar or different data points are, algorithms can extract meaningful patterns that recur within the data
- Clustering:
  - Clustering involves grouping similar data points together
  - The concept of similarity or distance is fundamental to determine which data points belong to the same cluster
  - For instance, in k-means clustering, the distance measure helps in assigning data points to the nearest centroid.
- Outlier Detection:
  - Outlier detection aims to identify anomalies or rare items in the data.
  - By measuring how different a data point is from the rest (using distance measures), one can determine if it's an outlier.
  - Outliers often have a significantly higher or lower distance value compared to other data points.

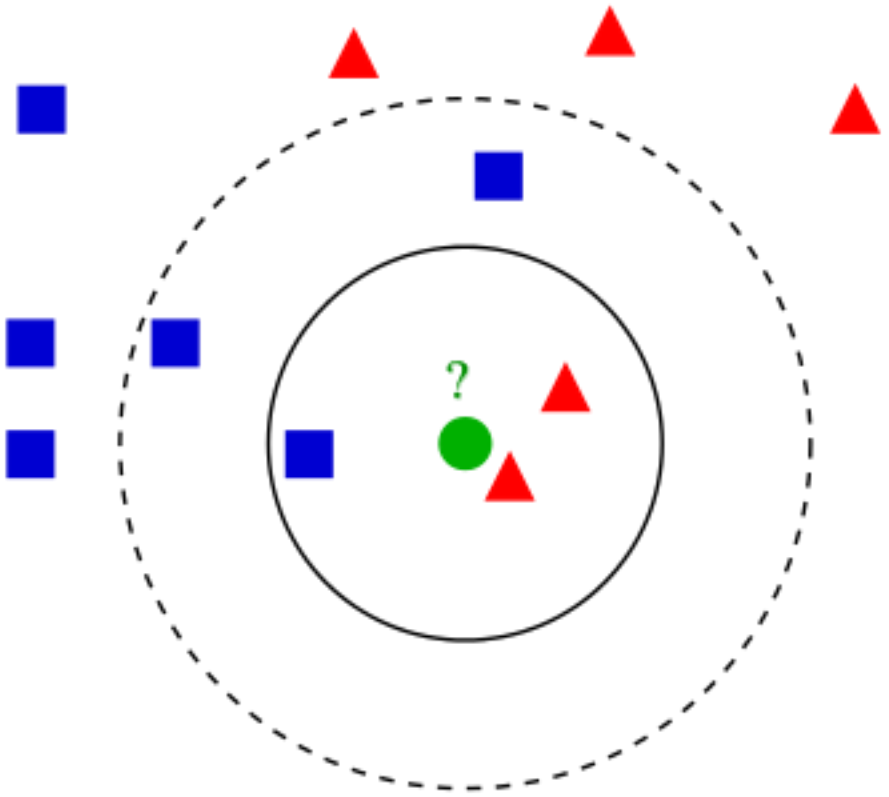


Image src

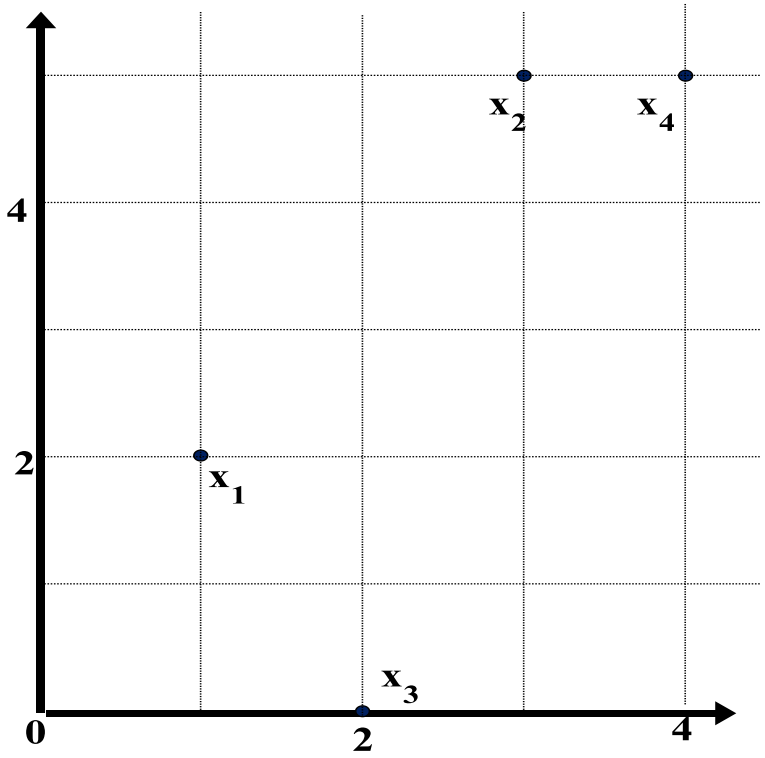
Similarity, Dissimilarity, and Proximity

- **Similarity measure** or **similarity function**
  - A real-valued function that quantifies the similarity between two objects
  - Measure how two data objects are alike: The higher value, the more alike
  - Often falls in the range [0,1]: 0: no similarity; 1: completely similar
- **Dissimilarity** (or **distance**) measure
  - Numerical measure of how different two data objects are
  - In some sense, the inverse of similarity: The lower, the more alike
  - Minimum dissimilarity is often 0 (i.e., completely similar)
  - Range [0, 1] or [0, ∞) , depending on the definition
- **Proximity** usually refers to either similarity or dissimilarity

Data Matrix

- Data matrix
  - A data matrix of *n* data points and *l* attributes

$$D = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1l} \\ x_{21} & x_{22} & \cdots & x_{2l} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ x_{n1} & x_{n2} & \cdots & x_{nl} \end{pmatrix}$$



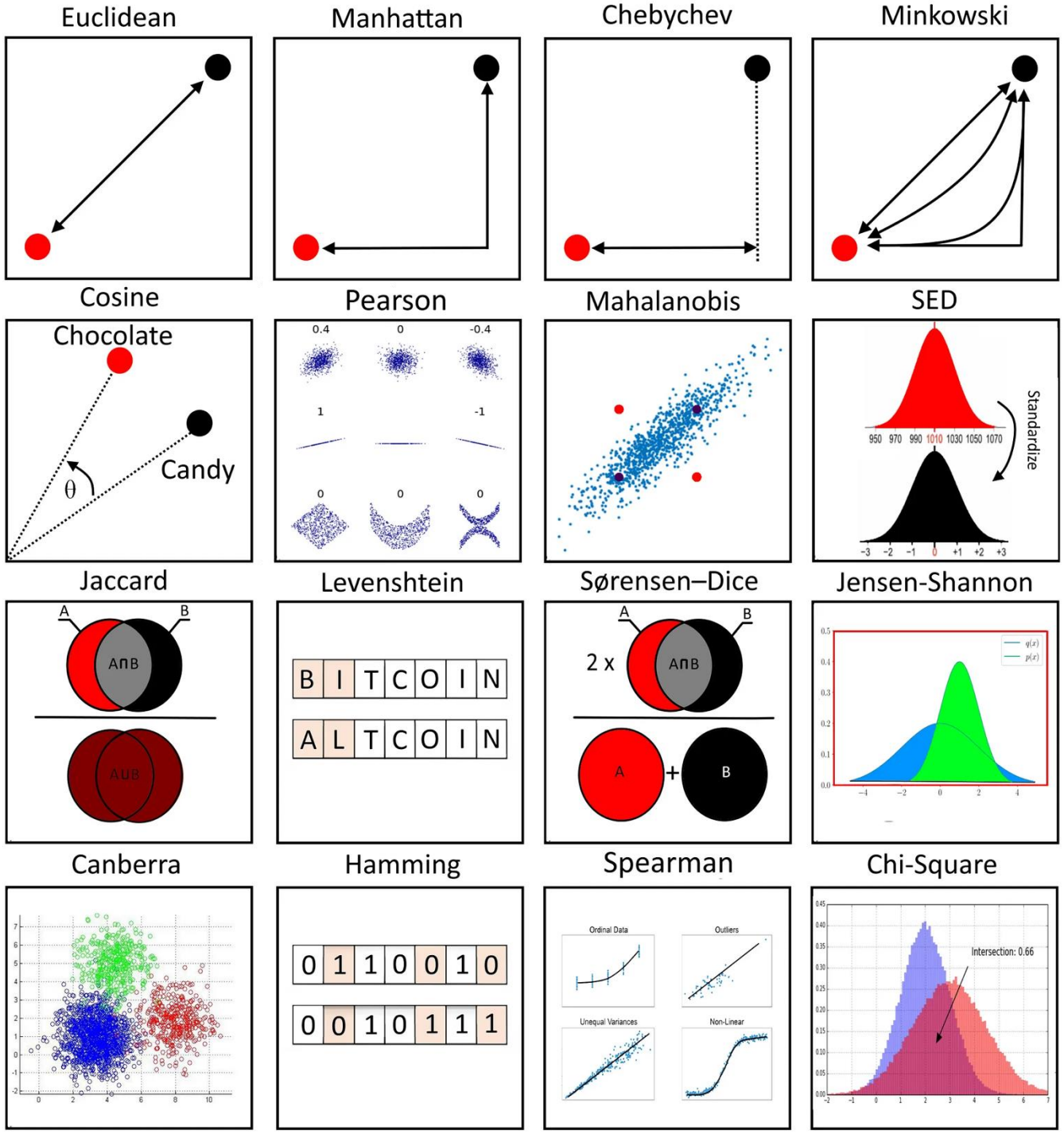
Example: Data Matrix

point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Dissimilarity or Distance Matrix

- Dissimilarity (distance) matrix
  - Matrix of distance measures; d(i, j) is the distance between data points i and j.
  - Usually a symmetric matrix with d(i, j) = d(j, i) and d(i, j) > 0.
  - Distance functions are usually different for real, boolean, categorical, ordinal, ratio, and vector variables
  - **Example:** Physical distance in km between latitude/longitude points.

$$\begin{pmatrix} 0 & & & \\ d(2,1) & 0 & & \\ \text{M} & \text{M} & \text{O} & \\ d(n,1) & d(n,2) & \ldots & 0 \end{pmatrix}$$



- There are a lot of different ways to calculate distance between two data points
- The choice depends on the data we are working with
- We will only focus on some of them in this chapter

src

Numeric Distance Metrics

- Manhattan (or city block) distance
  - E.g., the Hamming distance: the number of bits that are different between two binary vectors

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \mathrm{L} + |x_{il} - x_{jl}|$$

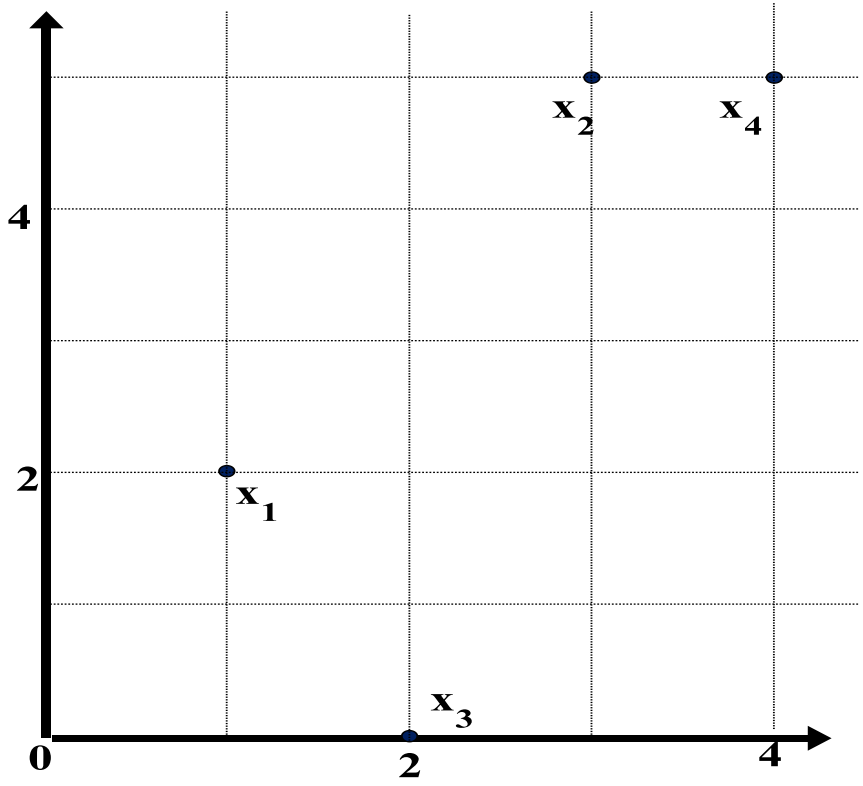
- Euclidean distance

$$d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \mathrm{L} + |x_{il} - x_{jl}|^2}$$

Notice the similar formulas here – these two metrics are both special cases of the Minkowski distance.

Example: Numeric Distance Metrics

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Manhattan

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Standardizing Numeric Data

- **Z-score:** $$z = \frac{x - \mu}{\sigma}$$
  - X: raw score to be standardized, μ: mean of the population, σ: standard deviation
  - the distance between the raw score and the population mean in units of the standard deviation
  - Perform on each feature to ensure that they are of similar magnitude
- **Why do we standardize data?**
  - For distance based clustering or prediction algorithms, the algorithm will weight features more heavily if they are of higher magnitude.
  - Performing standardization ensures that features will be considered equally in our predictions.
  - **Example:** Square feet vs. Number of bedrooms in a housing price prediction model

Proximity Measure for Binary Attributes

- A contingency table for binary data

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

- Distance measure for symmetric binary variables:

$$d(i, j) = \frac{r+s}{q+r+s+t}$$

  - (# items that don't match) / (total items)

- Distance measure for asymmetric binary variables:

$$d(i, j) = \frac{r+s}{q+r+s}$$

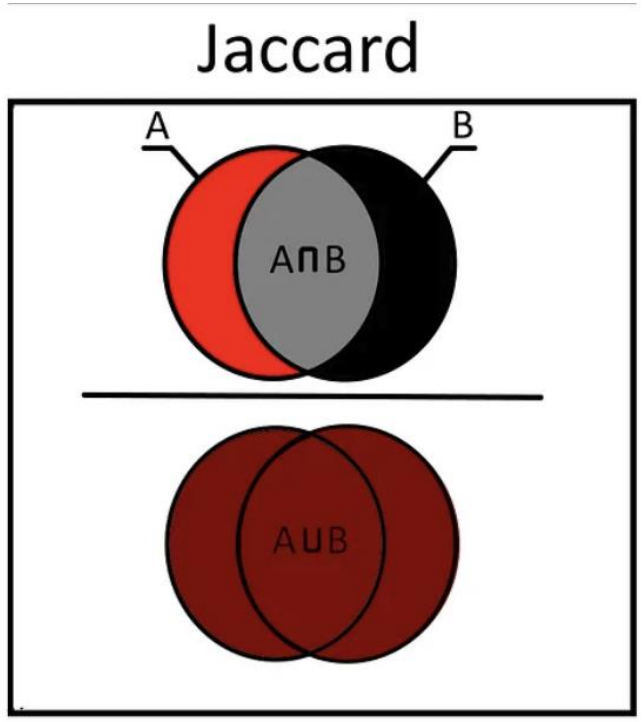
  - Here we exclude the (0,0) case from the total, assuming this is the over-represented case.

Proximity Measure for Binary Attributes

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

$$sim_{Jaccard}(i, j) = \frac{q}{q + r + s}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):
  - (# of (1,1) matches) / (total items excluding (0,0))
  - This tells us how many matching positives there are, out of all features with at least 1 positive represented.



Example: Dissimilarity between Asymmetric Binary Variables

Name	Sex	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- The 6 Covid-related attributes are asymmetric binary, we exclude sex for this exercise.
- Let the values Y and P be 1, and the value N be 0
- Distance:

$$d(i,j) = \frac{r+s}{q+r+s}$$

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Jack vs Mary

Jack \ Mary	1	0	Σ_{row}
1	2	0	2
0	1	3	4
Σ_{col}	3	3	6

Jack vs Jim

Jack \ Jim	1	0	Σ_{row}
1	1	1	2
0	1	3	4
Σ_{col}	2	4	6

Jim vs Mary

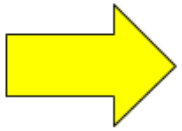
Jim \ Mary	1	0	Σ_{row}
1	1	1	2
0	2	2	4
Σ_{col}	3	3	6

Proximity Measure for Categorical Attributes

- Categorical data, also called nominal attributes
  - Example: Color (red, yellow, green), profession, etc.
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i, j) = \frac{p - m}{p}$$

Color
Red
Red
Yellow
Green
Yellow



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

src

- Method 2: Use a large number of binary attributes
  - Creating a new binary attribute for each of the *M* nominal states.

    These are often called dummy variables or one-hot encoding.

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank (e.g., freshman, sophomore, junior, senior)
- Can be treated like interval-scaled numeric attributes based on rank.
  - Replace *an ordinal variable value* by its rank: $r_{if} \in \{1, \ldots, M_f\}$
  - Map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

where M_f is the number of states, r_{if} is the rank of i-th object.

    - Example: freshman: 0; sophomore: 1/3; junior: 2/3; senior 1
      - Then distance: d(freshman, senior) = 1, d(junior, senior) = 1/3
  - Compute the dissimilarity using methods for interval-scaled variables

Attributes of Mixed Type

- A dataset may contain all attribute types
  - Nominal, symmetric binary, asymmetric binary, numeric, and ordinal

Table 2.4 A sample data table containing attributes of mixed types.

Object Identifier	Test-1 (nominal)	Test-2 (ordinal)	Test-3 (numeric)
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

Textbook
Page 45

Real databases contain various types of features

"So, how can we compute the dissimilarity between objects of mixed attribute types?" One approach is to group each type of attributes together, performing separate data mining (e.g., clustering) analysis for each type. This is feasible if these analyses derive compatible results. However, in real applications, it is unlikely that a separate analysis per attribute type will generate compatible results.

A better approach

A more preferable approach is to process all attribute types together, performing a single analysis. One such technique combines the different attributes into a single dissimilarity matrix, bringing all of the meaningful attributes onto a common scale of the interval [0.0, 1.0].

Textbook
Page 51

$$d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}},$$

Textbook
Page 51

Suppose that the data set contains p attributes of mixed types. The dissimilarity $d(i, j)$ between objects i and j is defined as

$$d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}, \tag{2.27}$$

where the indicator $\delta_{ij}^{(f)} = 0$ if either (1) x_{if} or x_{jf} is missing (i.e., there is no measurement of attribute f for object i or object j), or (2) $x_{if} = x_{jf} = 0$ and attribute f is asymmetric binary; otherwise, $\delta_{ij}^{(f)} = 1$. The contribution of attribute f to the dissimilarity between i and j (i.e., $d_{ij}^{(f)}$) is computed dependent on its type:

- If f is numeric: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_f - min_f}$, where max_f and min_f are the maximum and minimum values of attribute f, respectively;
- If f is nominal or binary: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise, $d_{ij}^{(f)} = 1$; and
- If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as numeric.

These steps are identical to what we have already seen for each of the individual attribute types. The only difference is for numeric attributes, where we normalize so that the values map to the interval [0.0, 1.0]. Thus the dissimilarity between objects can be computed even when the attributes describing the objects are of different types.

Textbook
Page 51

Table 2.4 A sample data table containing attributes of mixed types.

Object Identifier	Test-1 (nominal)	Test-2 (ordinal)	Test-3 (numeric)
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

The dissimilarity matrix for Test-1
(See example 2.18)

$$\begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

Dissimilarity matrix for Test-2.
(See example 2.22)

$$\begin{bmatrix} 0 & & & \\ 0.55 & 0 & & \\ 0.45 & 1.00 & 0 & \\ 0.40 & 0.14 & 0.86 & 0 \end{bmatrix}$$

Dissimilarity matrix for Test-3.
(See example 2.23)

Calculating the combined dissimilarity matrix using formula $d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$

Example, for d(3,1) $d(3, 1) = \frac{1(1)+1(0.50)+1(0.45)}{3}$

$$\begin{bmatrix} 0 & & & \\ 0.85 & 0 & & \\ 0.65 & 0.83 & 0 & \\ 0.13 & 0.71 & 0.79 & 0 \end{bmatrix}$$

(See Example 2.23)

Cosine Similarity of Two Vectors

- A **document** can be represented by a "bag of words" or a long vector, with each attribute recording the *frequency* of a particular term (such as word, keyword, or phrase) in the document

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- Applications: Information retrieval, gene feature mapping, very important in natural language processing (NLP).
- Cosine measure: If d_1 and d_2 are two vectors (e.g., term-frequency vectors), then

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

where $\bullet$ indicates vector dot product, $||d||$: the length (also called *norm*) of vector d

Example: Calculating Cosine Similarity

- Calculating Cosine Similarity: $$cos(d_1, d_2) = \frac{d_1 \bullet d_2}{\| d_1 \| \times \| d_2 \|}$$

  where • indicates vector dot product, $||d||$: the length of vector d

- Ex: Find the **similarity** between documents 1 and 2.

  d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0) d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

  - First, calculate vector dot product

    $d_1 \bullet d_2$ = 5 X 3 + 0 X 0 + 3 X 2 + 0 X 0 + 2 X 1 + 0 X 1 + 0 X 1 + 2 X 1 + 0 X 0 + 0 X 1 = 25

  - Then, calculate $||d_1||$ and $||d_2||$

$$\| d_1 \| = \sqrt{5\times5+0\times0+3\times3+0\times0+2\times2+0\times0+0\times0+2\times2+0\times0+0\times0} = 6.481$$

$$\| d_2 \| = \sqrt{3\times3+0\times0+2\times2+0\times0+1\times1+1\times1+0\times0+1\times1+0\times0+1\times1} = 4.12$$

  - Calculate cosine similarity: cos(d_1, d_2) = 25/ (6.481 X 4.12) = 0.94

Capturing Hidden Semantics in Similarity Measures

- The above similarity measures cannot capture hidden semantics
  - Which pairs are more similar: Geometry, algebra, music, politics?
- The same bags of words may express rather different meanings
  - "The cat bites a mouse" vs. "The mouse bites a cat"
  - This is beyond what a vector space model can handle
- Moreover, objects can be composed of rather complex structures and connections (e.g., graphs and networks)
- New similarity measures needed to handle complex semantics
  - Ex. Distributive representation and representation learning