DEPAUL UNIVERSITY

DSC478: Programming Machine Learning Applications

Roselyne Tchoua

rtchoua@depaul.edu

School of Computing, CDM, DePaul University

Distance and Similarity Measures

- Many data mining tasks involve the comparison of objects and determining their **similarities** (or **dissimilarities**)
  - Clustering
  - Nearest-neighbor search, classification, and prediction
  - Correlation analysis
- Many of todays real-world applications rely on the computation similarities or distances among objects
  - Personalization (serving you content based on your past behavior)
  - Recommender systems (recommending items to you based on who you "look like"
  - Document categorization (what documents look alike➔share a topic)
  - Information retrieval (What document "look like" your query doc"?)

Similarity and Dissimilarity

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

Distance or Similarity Measures

Measuring Distance

- In order to group similar items, we need a way to measure the distance between objects (e.g., records)
- Often requires the representation of objects as "feature vectors"

An Employee DB

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

Feature vector corresponding to Employee 2: <M, 51, 64000.0>

Term Frequencies for Documents

	T1	T2	T3	T4	T5	T6
Doc1	0	4	0	0	0	2
Doc2	3	1	4	3	1	2
Doc3	3	0	0	0	3	0
Doc4	0	1	0	3	0	0
Doc5	2	2	2	3	1	4

Feature vector corresponding to Document 4: <0, 1, 0, 3, 0, 0>

Distance or Similarity Measures

- Properties of Distance Measures:
  - for all objects A and B, $dist(A, B) \geq 0$, and $dist(A, B) = dist(B, A)$
  - for any object A, $dist(A, A) = 0$
  - $dist(A, C) \leq dist(A, B) + dist(B, C)$
- Representation of objects as vectors:
  - Each data object (item) can be viewed as an n-dimensional vector, where the dimensions are the attributes (features) in the data
  - Example (employee DB): Emp. ID 2 = <M, 51, 64000>
  - Example (Documents): DOC2 = <3, 1, 4, 3, 1, 2>
  - The vector representation allows us to compute distance or similarity between pairs of items using standard vector operations, e.g., Cosine of the angle between vectors, Manhattan distance, Euclidean distance, Hamming distance

Distance Metrics

Minkowsky:

$$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^r \right)^{1/r}$$

Euclidean:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Manhattan / city-block:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

Camberra:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$$

Chebychev:

$$D(\boldsymbol{x},\boldsymbol{y}) = \max_{i=1}^{m} |x_i - y_i|$$

Quadratic:

$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T Q(\boldsymbol{x} - \boldsymbol{y}) = \sum_{j=1}^{m} \left(\sum_{i=1}^{m} (x_i - y_i) q_{ji} \right) (x_j - y_j)$$

Q is a problem-specific positive definite $m \times m$ weight matrix

Mahalanobis:

$$D(\boldsymbol{x},\boldsymbol{y}) = [\det V]^{1/m} (\boldsymbol{x} - \boldsymbol{y})^{\mathrm{T}} V^{-1} (\boldsymbol{x} - \boldsymbol{y})$$

V is the covariance matrix of $A_1..A_m$, and A_j is the vector of values for attribute j occuring in the training set instances $1..n$.

Correlation:

$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i=1}^{m} (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{m} (x_i - \overline{x_i})^2 \sum_{i=1}^{m} (y_i - \overline{y_i})^2}}$$

$\overline{x_i} = \overline{y_i}$ and is the average value for attribute i occuring in the training set.

Chi-square:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{1}{sum_i} \left(\frac{x_i}{size_{\boldsymbol{x}}} - \frac{y_i}{size_{\boldsymbol{y}}} \right)^2$$

sum_i is the sum of all values for attribute i occuring in the training set, and $size_{\boldsymbol{x}}$ is the sum of all values in the vector $\boldsymbol{x}$.

Kendall's Rank Correlation:

$$D(\boldsymbol{x},\boldsymbol{y}) = 1 - \frac{2}{n(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{i-1} \mathrm{sign}(x_i - x_j)\,\mathrm{sign}(y_i - y_j)$$

sign(x)=-1, 0 or 1 if $x < 0$, $x = 0$, or $x > 0$, respectively.

Figure 1. Equations of selected distance functions.
($\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of m attribute values).

Data Matrix and Distance Matrix

- Data matrix
  - Conceptual representation of a table
    - Cols = features; rows = data objects
  - *n* data points with *p* dimensions
  - Each row in the matrix is the vector representation of a data object

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Distance (or Similarity) Matrix
  - *n* data points, but indicates only the pairwise distance (or similarity)
  - A triangular matrix
  - Symmetric

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Issues with Distance Matrix

- Most distance measures were designed for linear/real-valued attributes
- Two important questions in the context of machine learning:
  - How best to handle nominal attributes?
  - What to do when attribute types are mixed?

Similarity for Nominal Attributes

- If object attributes are all nominal (categorical), then proximity measure are used to compare objects
- Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

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

- Method 2: Convert to Standard Spreadsheet format
  - For each attribute *A* create *M* binary attribute for the *M* nominal states of *A (dummy variables)*
  - Then use standard vector-based similarity or distance metrics

Proximity Measure for Binary Attributes

- A contingency table for binary data
  - q: number of attributes that equal 1 in both objects i and j
  - **r: number of attributes that equal 1 for object i but equal 0 for object j**
  - **s: number of attributes that equal 1 for object j but equal 0 for object i**
  - t: number of attributes that equal 0 in both objects i and j
- **Symmetric** binary attributes, each state is **equally valuable**
- Symmetric Binary Dissimilarity ➔

Object *j*

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

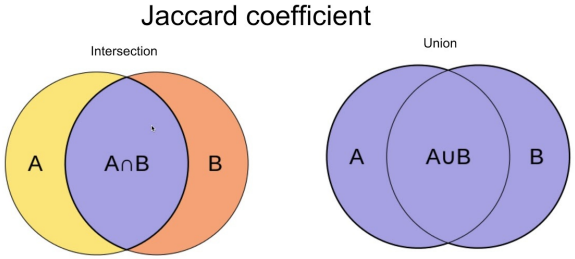
Distance, Difference

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

Proximity Measure for Binary Attributes

- Distance measure for asymmetric binary variables
  - e.g., 1 ➔ positive outcome of a test
  - 1 is then given more significance
  - Such attributes are "**monary**"
  - **t is ignored**

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



$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- Jaccard coefficient (similarity measure for asymmetric binary variables)
  - $sim_{Jaccard}(i,j) = 1 - d(i,j)$

https://www.geeksforgeeks.org/measures-of-distance-in-data-mining/

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

Normalizing or Standardizing Data before Computing Similarities

- Z-score: $z = \frac{x - \mu}{\sigma}$
  - x: raw value 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
  - negative when the value is below the mean, "+" when above
- Min-Max Normalization

$$x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}(new\max - new\min) + new\min$$

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

ID	Gender	Age	Salary
1	1	0.00	0.00
2	0	0.96	0.56
3	0	1.00	1.00
4	1	0.24	0.44
5	0	0.72	0.32

Common Distance for Numeric Data

- Consider two vectors
  - Rows in the data matrix

$$X = \langle x_1, x_2, \cdots, x_n \rangle$$

$$Y = \langle y_1, y_2, \cdots, y_n \rangle$$

- Common Distance Measures:
  - Manhattan distance:

$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

  - Euclidean distance:

$$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$

  - Distance can be defined as a dual of a similarity measure

$$dist(X,Y) = 1 - sim(X,Y)$$

Minkowski Distance

- Minkowski distance: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

  - where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and h is the order (the distance so defined is also called L-*h* norm)
- Note that Euclidean and Manhattan distances are special cases of Mikowski distance, L-1 and L-2 norms

Hamming Distance

- The Hamming distance d(000, 011) is 2 because:

$$000 \oplus 011 \text{ is } 011 \text{ (two 1s)}$$

- The Hamming distance d(10101, 11110) is 3 because:

$$10101 \oplus 11110 \text{ is } 01011 \text{ (three 1s)}$$

- Perform their XOR operation, ($a \oplus b$), and then count the total number of 1s in the resultant string
- e.g., Application: Used for error detection or error correction when data is transmitted over computer networks

Vector-Based Similarity

In some situations, distance measures provide a skewed view of data

- e.g., when the data is very sparse and 0's in the vectors are not significant (document topics)
- In such cases, typically vector-based similarity measures are used
- Most common measure: Cosine similarity
- Dot product of two vectors:

$$X = \langle x_1, x_2, \cdots, x_n \rangle \quad Y = \langle y_1, y_2, \cdots, y_n \rangle$$

- Cosine Similarity = normalized dot product $sim(X,Y) = X \bullet Y = \sum_i x_i \times y_i$
- the norm of a vector X is: $\|X\| = \sqrt{\sum_i x_i^2}$
- the cosine similarity is:

$$sim(X,Y) = \frac{X \bullet Y}{\|X\| \times \|y\|} = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2} \times \sqrt{\sum_i y_i^2}}$$

Vector-Based Similarity

Why divide by the norm? $X = \langle x_1, x_2, \cdots, x_n \rangle$ $\|X\| = \sqrt{\sum_i x_i^2}$

- Example:
  - X = <2, 0, 3, 2, 1, 4>
  - $||X||$ = SQRT(4+0+9+4+1+16) = 5.83
  - $X^* = X / ||X||$ = <0.343, 0, 0.514, 0.343, 0.171, 0.686>
- Now, note that $||X^*|| = 1$
- So, dividing a vector by its norm, turns it into a *unit-length* vector
- Cosine similarity measures the angle between two-unit length vectors (i.e., the magnitude of the vectors are ignored).

Think of it as, it doesn't matter the length of the document, but just the essence of it… what is it about? No matter how short.

Example Application: Information Retrieval

- Documents are represented as “bags of words”
- Represented as vectors when used computationally
  - A vector is an array of floating point (or binary in case of bit maps)
  - Has direction and magnitude
  - Each vector has a place for every term in collection (most are sparse)

Document Ids

	nova	galaxy	heat	actor	film	role
A	1.0	0.5	0.3			
B	0.5	1.0				
C	1.0	0.8	0.7			
D	0.9	1.0	0.5			
E			1.0	1.0		
F			0.7			
G	0.5		0.7			0.9
H	0.6		1.0	0.3	0.2	
I		0.7	0.5	0.3		

a document vector

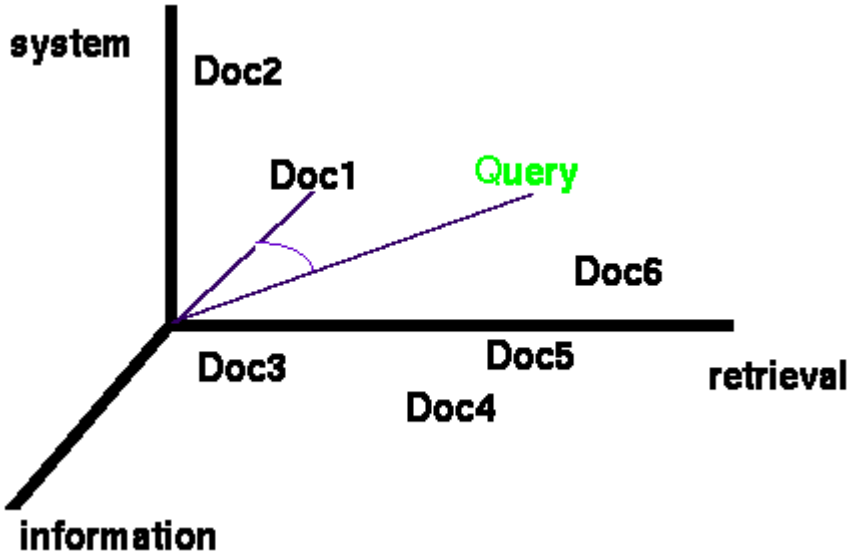
$$D_i = w_{d_{i1}}, w_{d_{i2}}, \ldots, w_{d_{it}}$$

$$Q = w_{q1}, w_{q2}, \ldots, w_{qt}$$

$$w = 0 \text{ if a term is absent}$$

Example Application: Information Retrieval

- Documents are represented as vectors in the term space
- Typically values in each dimension correspond to the frequency of the corresponding term in the document
- Queries represented as vectors in the same vector-space
- Cosine similarity between the query and documents is often used to rank retrieved documents



Example of Document Similarity Calculations

Given the document-term matrix

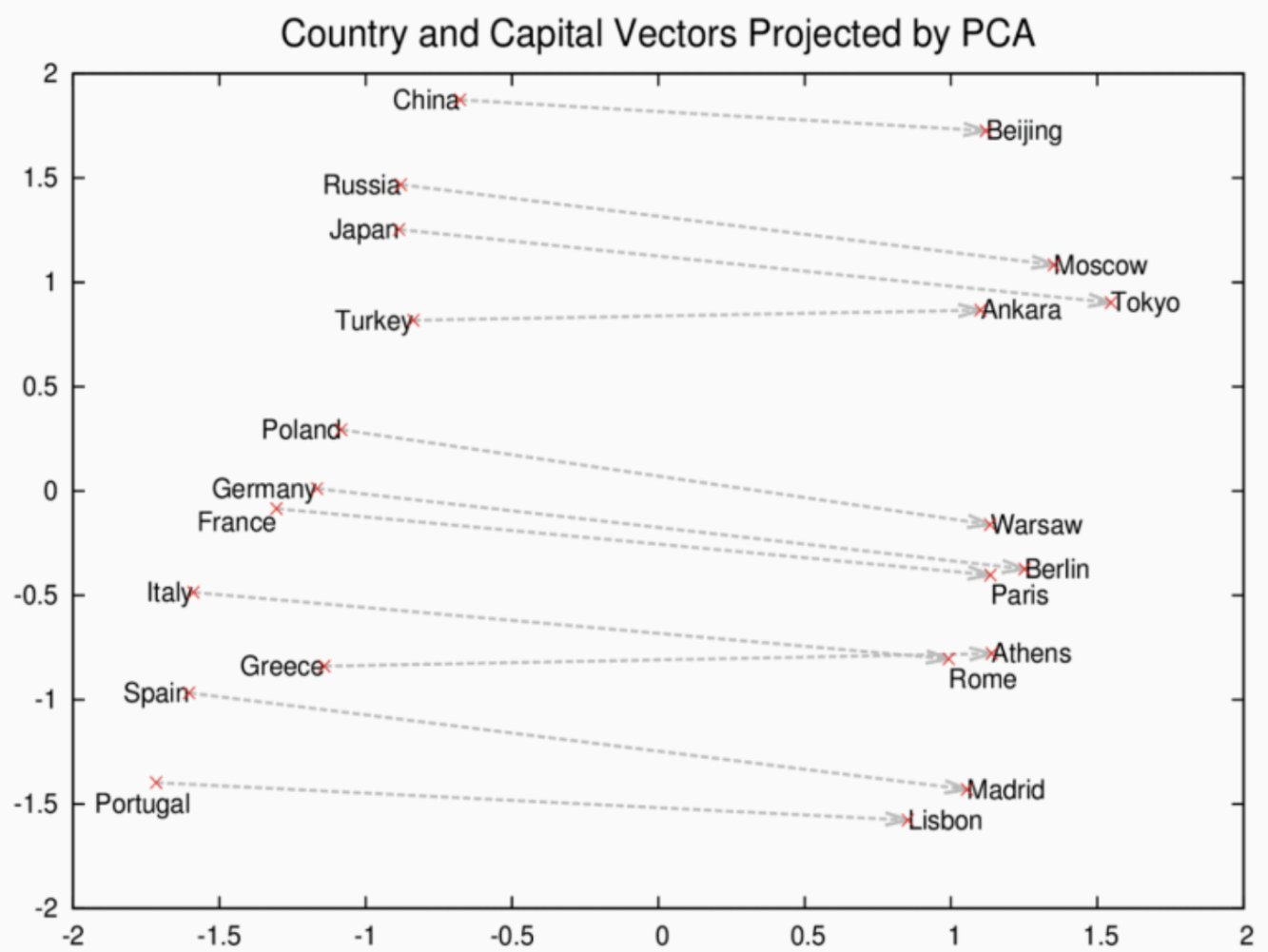
	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	0	4	0	0	0	2	1	3
Doc2	3	1	4	3	1	2	0	1
Doc3	3	0	0	0	3	0	3	0
Doc4	0	1	0	3	0	0	2	0
Doc5	2	2	2	3	1	4	0	2

Dot-Product(Doc2,Doc4) = <3,1,4,3,1,2,0,1> * <0,1,0,3,0,0,2,0>
0 + 1 + 0 + 9 + 0 + 0 + 0 + 0 = 10

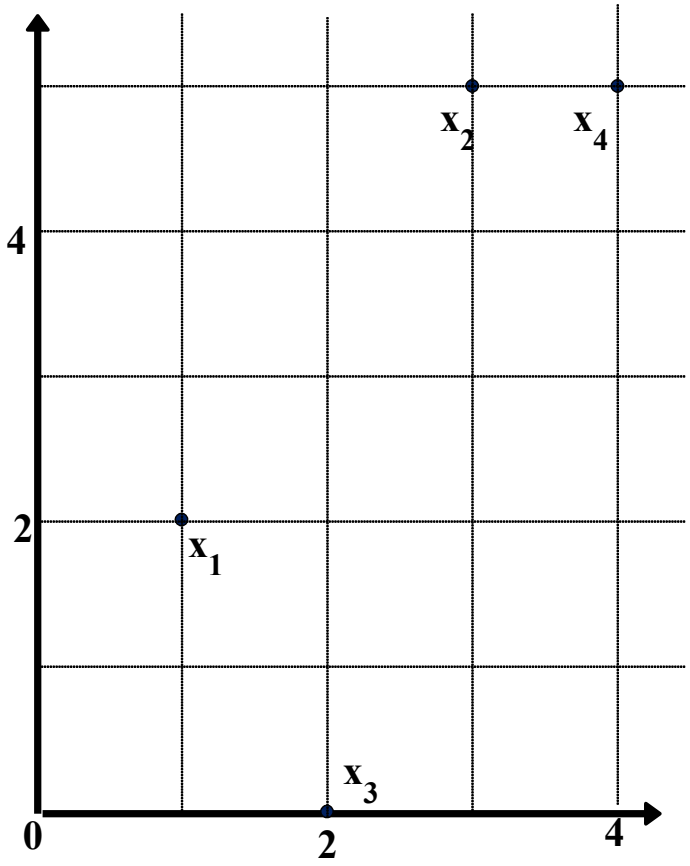
Norm (Doc2) = SQRT(9+1+16+9+1+4+0+1) = 6.4
Norm (Doc4) = SQRT(0+1+0+9+0+0+4+0) = 3.74

Cosine(Doc2, Doc4) = 10 / (6.4 * 3.74) = 0.42

Example: Word Relationships



Data Matrix vs. Distance Matrix



Data Matrix

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

Distance Matrix (Manhattan)

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

Distance Matrix (Euclidean)

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

Correlation as Similarity

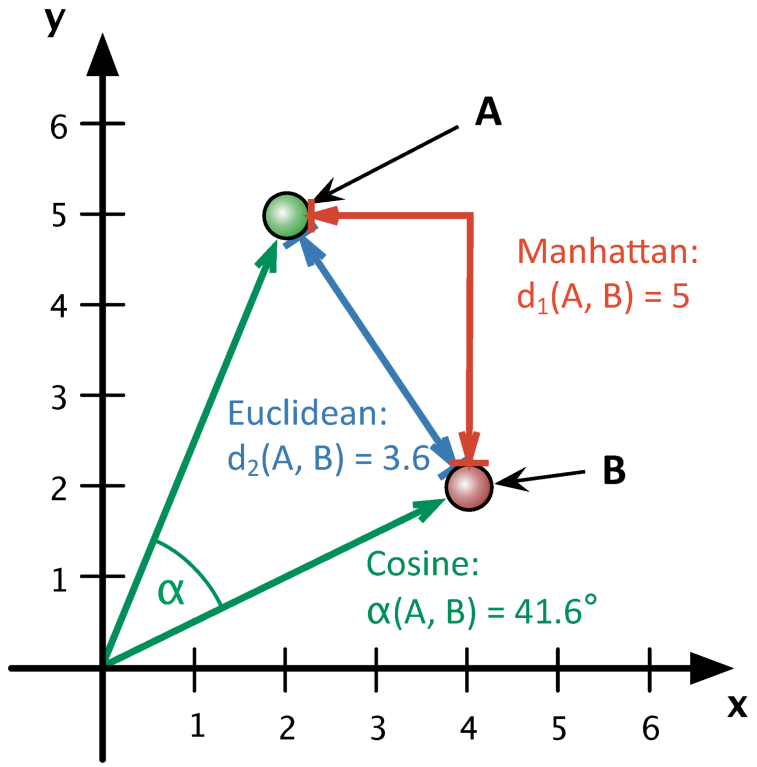
- In cases where there could be high mean variance across data objects (e.g., movie ratings), Pearson Correlation coefficient is the best option
- Pearson Correlation

$$corr(x,y) = \frac{\text{cov}(x,y)}{stdev(x) \cdot stdev(y)}$$

- Often used in recommender systems based on Collaborative Filtering

Summary: Different Distance Metrics

- Different problems/apps.
- Different data
  - Numeric: default is Euclidean
  - Sparse data? ➔ cosine
  - 0's and 1's ➔ sets ➔ jacquard or others
- Unfamiliar data: try multiple
- Mix types of attributes: define complex distances



https://medium.com/@prasoonthakur5/different-types-of-distances-used-in-machine-learning-7491128491b8