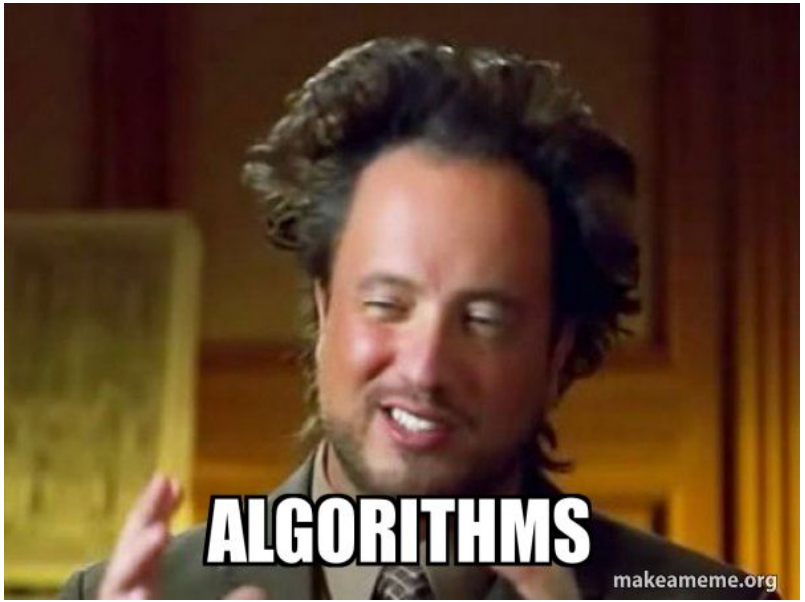
Algorithms Club

1.22.24

Why?

- Better understand how tools used in research are working
- Programming in different languages
- Learning from each other



Plan

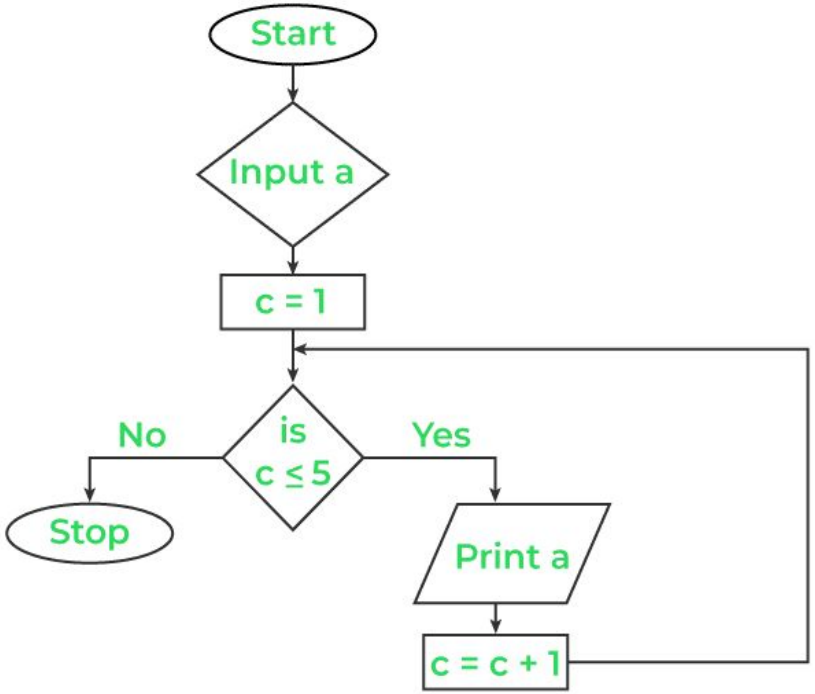
- Meet 6pm Mondays every 3 weeks in B118
- Session 1- Introduction to Algorithm
- Session 2- Talking about implementations

Session 1

- Volunteers to provide background of algorithm for first 20 - 30 minutes
- Discussion

Session 1

- Background of Problem
- Pseudocode/Implementation Strategies
- Mathematical Foundation
- Inputs/Outputs



Session 2

- Show off implementations
- Discussion groups?
- Upload code to shared Github

Hierarchical clustering

Hierarchical clustering: A Method to build a hierarchy of cluster

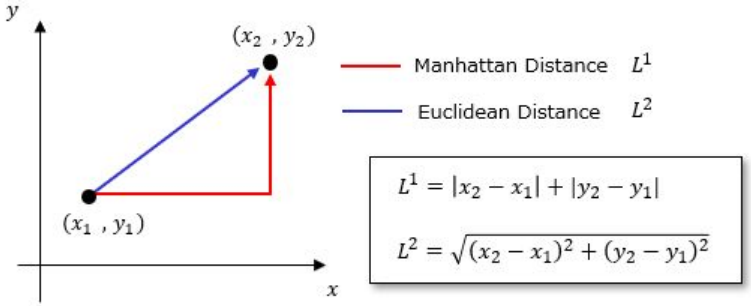
- Hierarchical clustering construct a clusters for all values of k (k =1 to n) either by **bottom-up** (agglomerative) or **top-down**(divisive) approach

- The only difference between k = r and k = r + 1 is that one of the r clusters splits up in order to obtain r + 1 clusters (or, to put it differently, two of the r + 1 clusters combine to yield r clusters)

Applications:

- analyze gene expression data by building clusters of genes with similar patterns of expression
- Phylogenetic analysis ….

Mathematics behind:



Similarity metric:

1) Manhattan Distance
2) Euclidean Distance

$$\begin{array}{l} \\ n \text{ objects} \end{array} \begin{array}{c} \overset{p \text{ variables}}{\overbrace{\phantom{\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \end{bmatrix}}}} \\ \begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \end{array}$$

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

A generalization of both the Euclidean and the Manhattan metric is the Minkowski distance

$$d(i, j) = \left(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q\right)^{1/q}$$

Weighted Euclidean distances:

$$d(i, j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \cdots + w_p(x_{ip} - x_{jp})^2}$$

Mathematical requirement of distance metric:

i) $d(i, j) \geq 0$
ii) $d(i, i) = 0$
iii) $d(i, j) = d(j, i)$
iv) $d(i, j) < d(i, h) + d(h, j)$

Need for data transformation in clustering

- To avoid the dependence on the choice of measurements
- To minimize the outliers effects (one may use the mean absolute deviation instead of standard deviation to minimize the outliers effect)

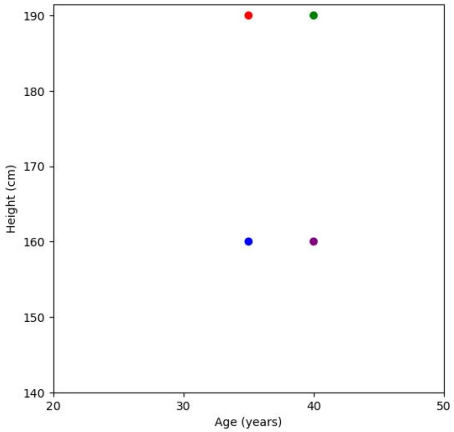
Techniques:

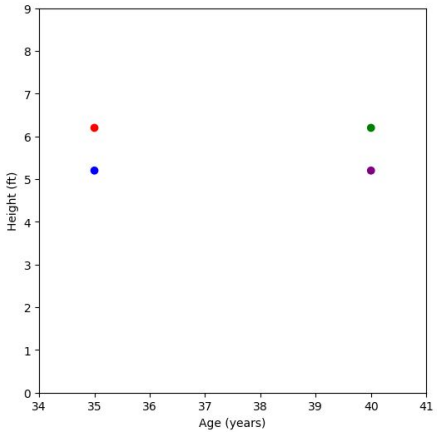
- ❏ Data standardization
- ❏ Normalization
- ❏ Minmax scaling

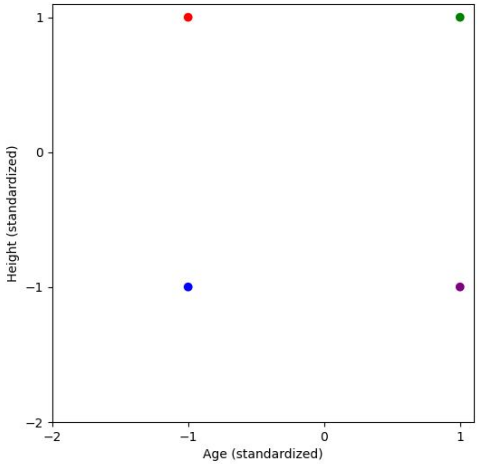
How the distribution in the data change with different data transformation?

Data standardization

Person	Age (yr)	Height (cm)	Height (ft)
A	35	190	6.2
B	40	190	6.2
C	35	160	5.2
D	40	160	5.2

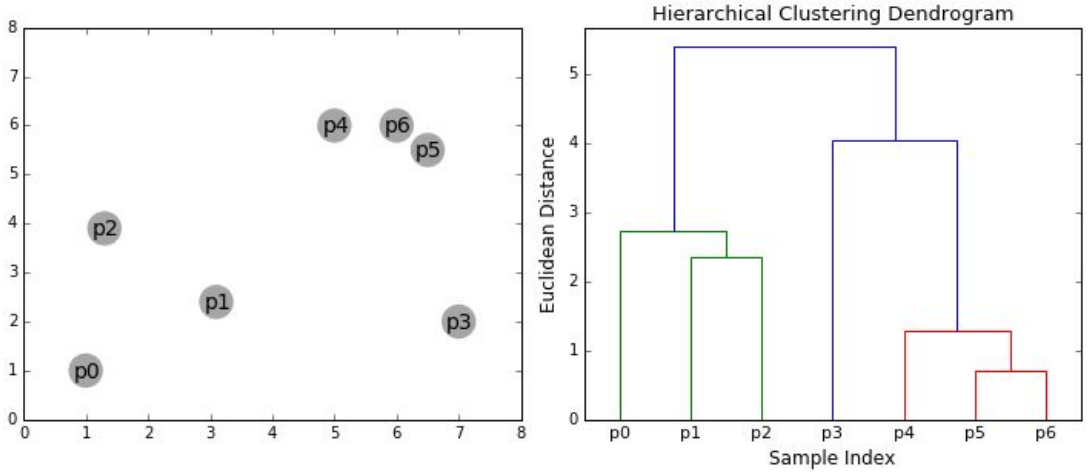






Bottom-up clustering (Agglomerative)

- To start with each data point belongs to their own cluster(i.e. at step 0 we have n clusters)
- Iteratively construct the similarity matrix and formed a cluster based on the optimum similarity



Pseudocode

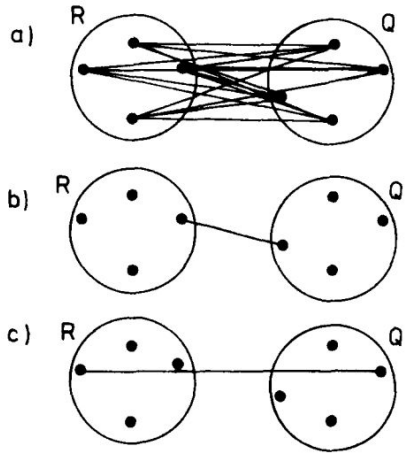
1) Start with all data points in their own cluster

2) Compute the similarity matrix, cluster those together which has optimum similarity score

   Case1: Similarity matrix between cluster and cluster
   Case2: Similarity matrix between cluster and data point

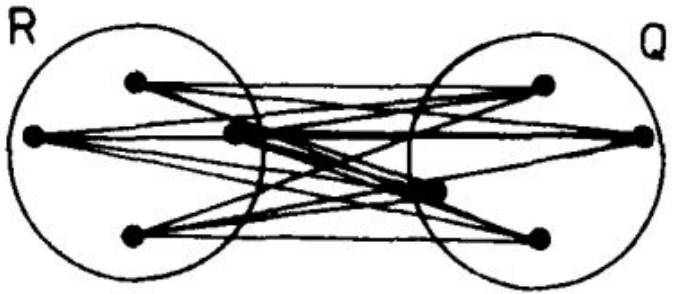
   Different techniques:

   a) Group average (UPGMA)
   b) Nearest neighbor
   c) Furthest neighbor



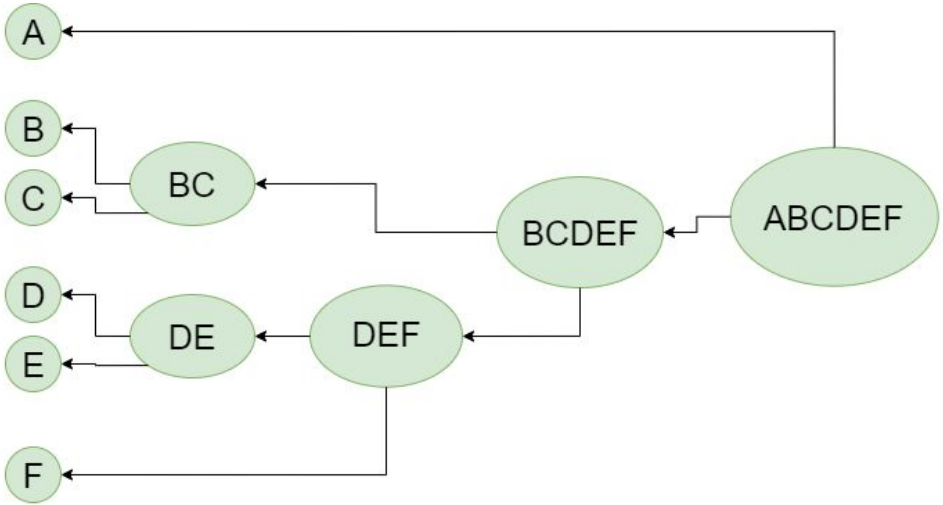
UPGMA - unweighted pair group method using arithmetic mean

$$d(R,Q) = \frac{1}{|R|\,|Q|} \sum_{\substack{i \in R \\ j \in Q}} d(i,j)$$



Divisive (Top-down) Clustering

- Everything Starts in one cluster
- Recursively split into smaller clusters



1. Compute dissimilarity matrix

	a	**b**	**c**	**d**	**e**
a	0.0	2.0	6.0	10.0	9.0
b	2.0	0.0	5.0	9.0	8.0
c	6.0	5.0	0.0	4.0	5.0
d	10.0	9.0	4.0	0.0	3.0
e	9.0	8.0	5.0	3.0	0.0

2. Find data point with highest dissimilarity

a- (2.0 + 6.0 + 10.0 + 9.0)/4 = 6.75
b- (2.0 + 5.0 + 9.0 + 8.0)/4 = 6.00
.
.
e- (9.0 + 8.0 + 5.0 + 3.0)/4 = 6.25

	a	**b**	**c**	**d**	**e**
a	0.0	2.0	6.0	10.0	9.0
b	2.0	0.0	5.0	9.0	8.0
c	6.0	5.0	0.0	4.0	5.0
d	10.0	9.0	4.0	0.0	3.0
e	9.0	8.0	5.0	3.0	0.0

3. Choose splinter group

a- (2.0 + 6.0 + 10.0 + 9.0)/4 = 6.75

b- (2.0 + 5.0 + 9.0 + 8.0)/4 = 6.00

.

.

e- (9.0 + 8.0 + 5.0 + 3.0)/4 = 6.25

	a	**b**	**c**	**d**	**e**
a	0.0	2.0	6.0	10.0	9.0
b	2.0	0.0	5.0	9.0	8.0
c	6.0	5.0	0.0	4.0	5.0
d	10.0	9.0	4.0	0.0	3.0
e	9.0	8.0	5.0	3.0	0.0

4. See what other point joins splinter

Dissimilarity
Remaining Objects

b- (5.0 + 9.0 + 8.0)/3 = 7.33
c = 4.67
d = 5.33
e- (8.0 + 5.0 + 3.0)/ 3 = 5.33

Dissimilarity
objects splinter

b = 2.0
c = 6.0
d = 10.0
e = 9.0

Difference

b = 7.33 - 2.0 = 5.33
c = 4.67 - 6.0 = -1.33
d = 5.33 - 10.0 = -4.67
e = 5.33 - 9.0 = -3.67

	a	b	c	d	e
a	0.0	2.0	6.0	10.0	9.0
b	2.0	0.0	5.0	9.0	8.0
c	6.0	5.0	0.0	4.0	5.0
d	10.0	9.0	4.0	0.0	3.0
e	9.0	8.0	5.0	3.0	0.0

4. See what other point joins splinter

Dissimilarity
Remaining Objects

b- (5.0 + 9.0 + 8.0)/3 = 7.33
c = 4.67
d = 5.33
e- (8.0 + 5.0 + 3.0)/ 3 = 5.33

Dissimilarity
objects splinter

b = 2.0
c = 6.0
d = 10.0
e = 9.0

Difference

b = 7.33 - 2.0 = 5.33
c = 4.67 - 6.0 = -1.33
d = 5.33 - 10.0 = -4.67
e = 5.33 - 9.0 = -3.67

	a	b	c	d	e
a	0.0	2.0	6.0	10.0	9.0
b	2.0	0.0	5.0	9.0	8.0
c	6.0	5.0	0.0	4.0	5.0
d	10.0	9.0	4.0	0.0	3.0
e	9.0	8.0	5.0	3.0	0.0

Pseudocode

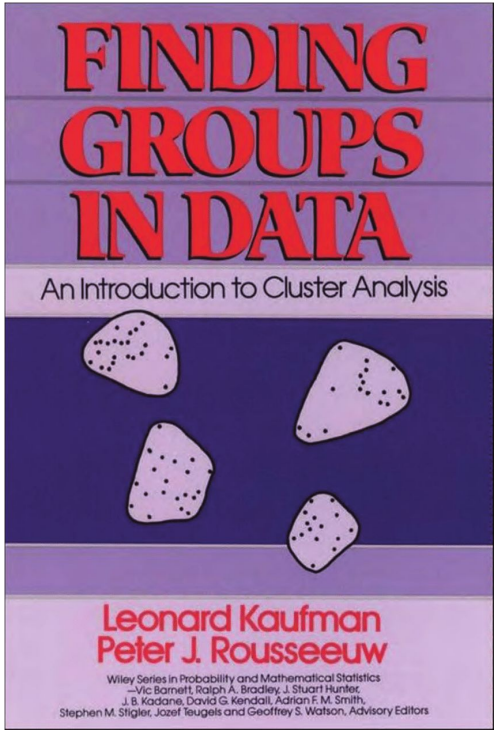
1. Compute dissimilarity/distance matrix for each data point
2. Find data point with highest average dissimilarity to other points
3. For each object of the larger group, compute average dissimilarity with the remaining objects
4. Calculate difference between average dissimilarity with objects of splinter group
5. Largest positive difference becomes new member of splinter group
    a. If all differences are negative, cycle stops
6. Next cluster to split is the one with the largest diameter. The cluster that has the largest difference between any two points
7. Repeat steps 2-7 until all leaves are singletons

Pseudocode 2

1. Start data in one cluster
2. Split data into two clusters using flat clustering method (K-means, DBSCAN)
3. Choose which cluster is best to split next
    a. Most heterogeneous
4. Repeat steps 2 & 3 until all points in one cluster

Comments

- Top-down more complex, **but**
  - More efficient if do not complete full hierarchy
  - More accurate because takes into account global distribution of data
- First Published in 1990 (DIANA)→



Discussion questions:

1) Does the **bottom-up** (agglomerative) and **top-down** (divisive) techniques produce the **same cluster** ?

2) **pros/cons** of **hierarchical clustering** over **K-means clustering** ?

3) How to handle the missing data points in the data set for clustering ?

4) What about using the **dissimilarity metric instead of similarity metric** in computing distance ? How does your cluster change with respect to that?

5) How would you construct the similarity matrix for phylogenetic tree ?
Or any other interesting data set that you work with in your research ?

References:

1) Finding Groups in Data: An Introduction to Cluster Analysis; Leonard Kaufman, Peter J. Rousseeuw; DOI:10.1002/9780470316801
https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316801