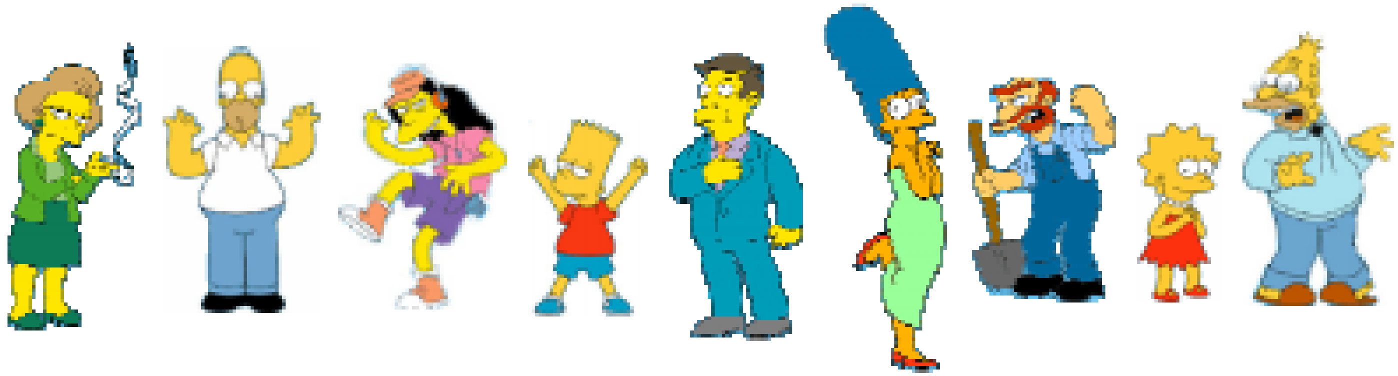
K-Means Clustering

COMP6065 – Artificial Intelligence

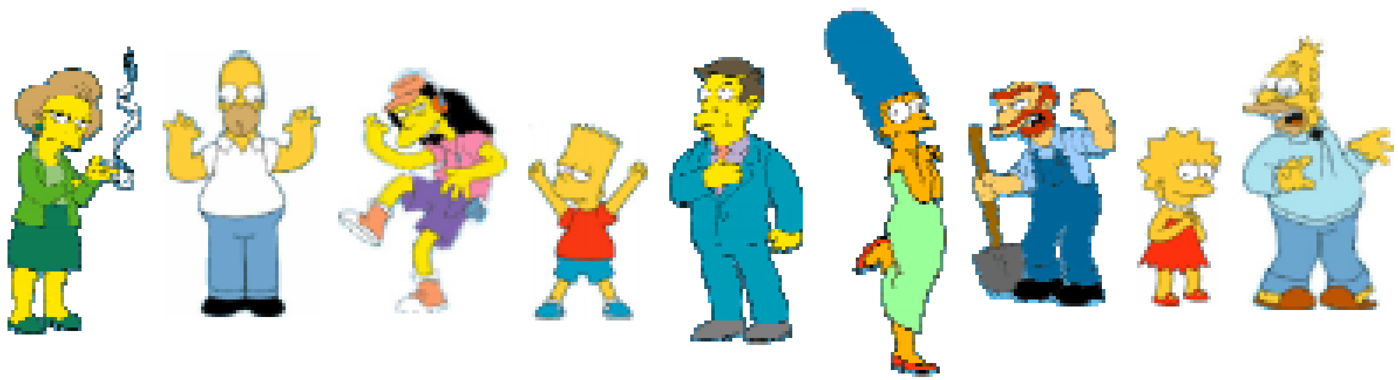
What is Clustering?

- Organizing data into clusters such that there is
  - high intra-cluster similarity
  - low inter-cluster similarity
- Informally, finding natural groupings among objects.
- Clustering:
  - ➢ Unsupervised learning
  - ➢ Requires data, but no labels
  - ➢ Detect patterns, for example:
    - ➢ Group emails or search results
    - ➢ Customer shopping patterns
    - ➢ Regions of images

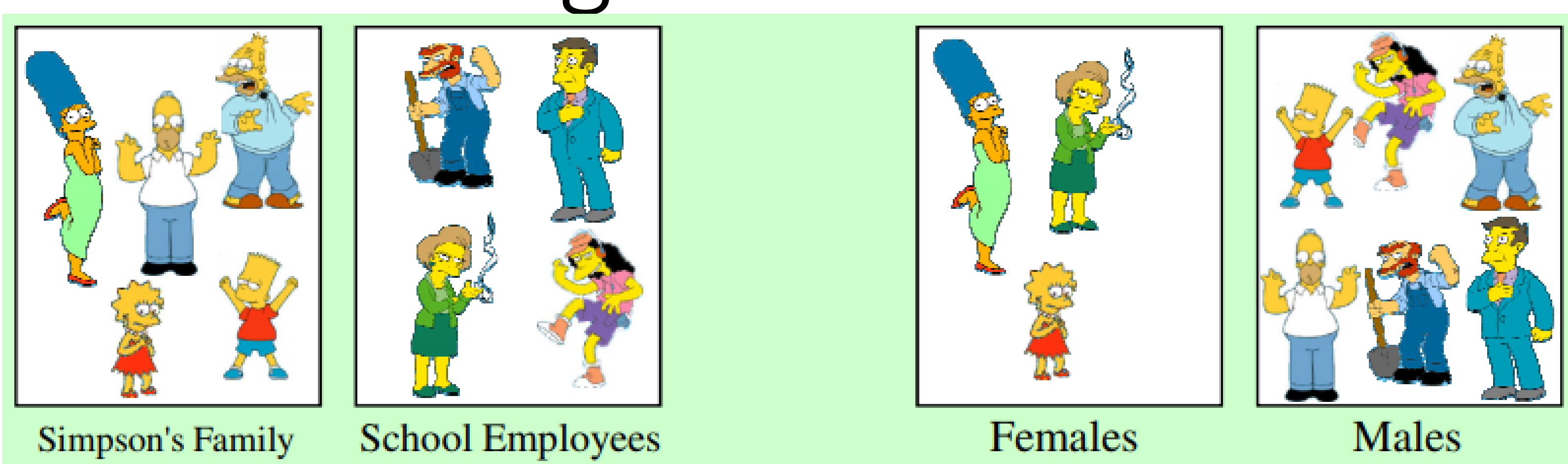
What is natural grouping among these objects?



What is natural grouping among these objects?



Clustering is **SUBJECTIVE**



What is similarity?

- According to Webster Dictionary, **similarity** is the quality or state of being similar; likeness; resemblance; as, a similarity of features.
- Similarity is hard to define, but… “we know it when we see it”

Common Distance Measures

- *Distance measure* will determine how the *similarity* of two elements is calculated and it will influence the shape of the clusters.
- The two most common distance measures:

1. The Euclidean distance (also called 2-norm distance) is given by:

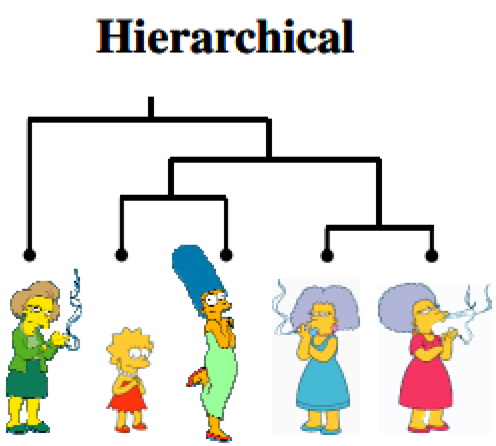
$$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

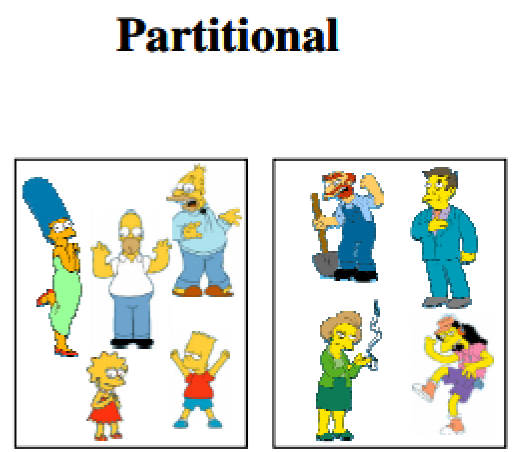
2. The Manhattan distance (also called taxicab norm or 1-norm) is given by:

$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$

Types of Clustering

- Partitional algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchical algorithms: Create a hierarchical decomposition of the set of objects using some criterion





K-Means Clustering

- K-Means Clustering is a type of **partitional clustering**
- The k-means algorithm is an algorithm to **cluster** n objects based on attributes into *k* **partitions**, where $k < n$.
- It assumes that the object attributes form a vector space.

K-Means Clustering

- An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion

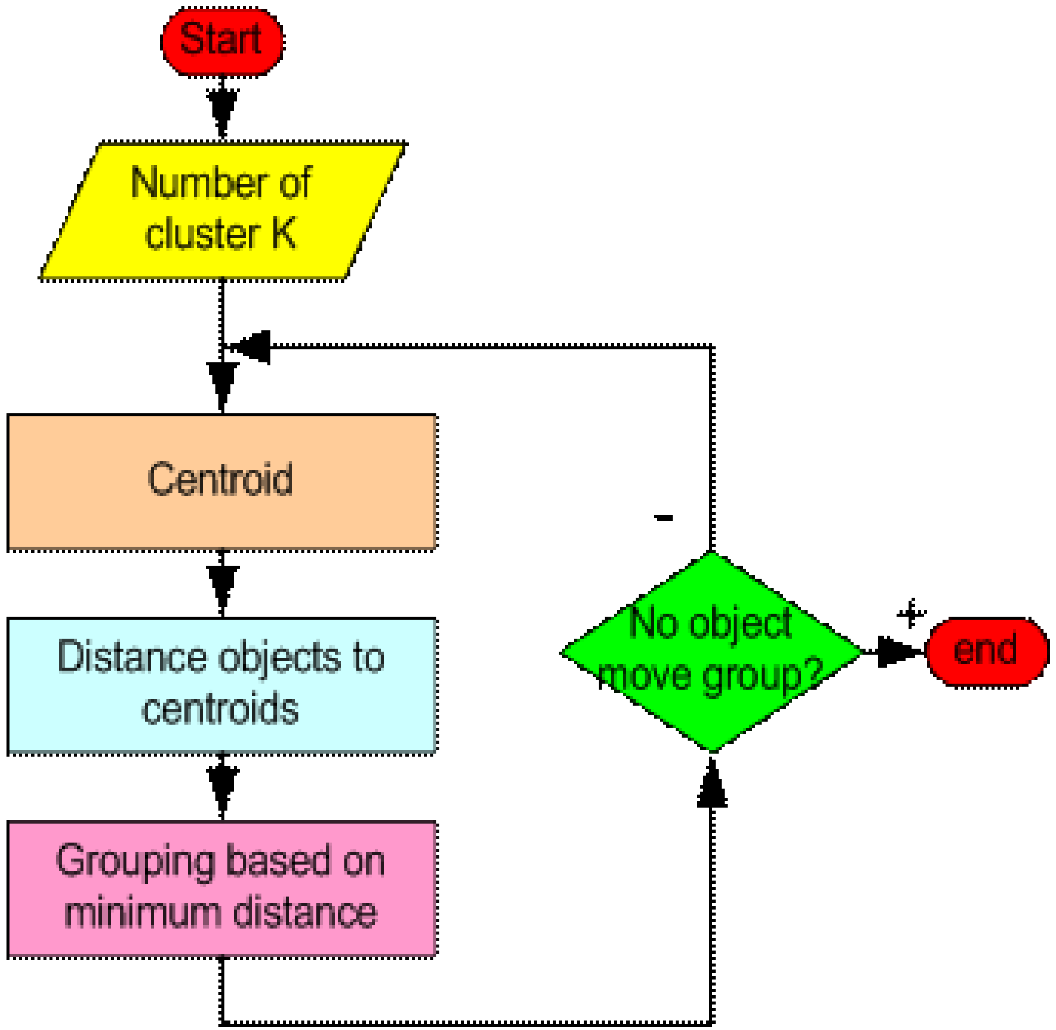
$$J = \sum_{j=1}^{K} \sum_{n \in s_j} \left| x_n - \mu_j \right|^2$$

where x_n is a vector representing the the n^{th} data point and u_j is the [geometric centroid]() of the data points in S_j.

K-Means Clustering

- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?



K-Means Clustering Algorithm

- **Step 1:** Begin with a decision on the value of k = number of clusters .
- **Step 2**: Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:

  1. Take the first k training sample as single-element clusters

  2. Assign each of the remaining (N-k) sample to the cluster with the nearest centroid. After each assignment, re-compute the centroid of the gaining cluster.

K-Means Clustering

- **Step 3:** Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- **Step 4 .** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

A Simple example showing the implementation of k-means algorithm
(using K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.
- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.50
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

- Step 4 :

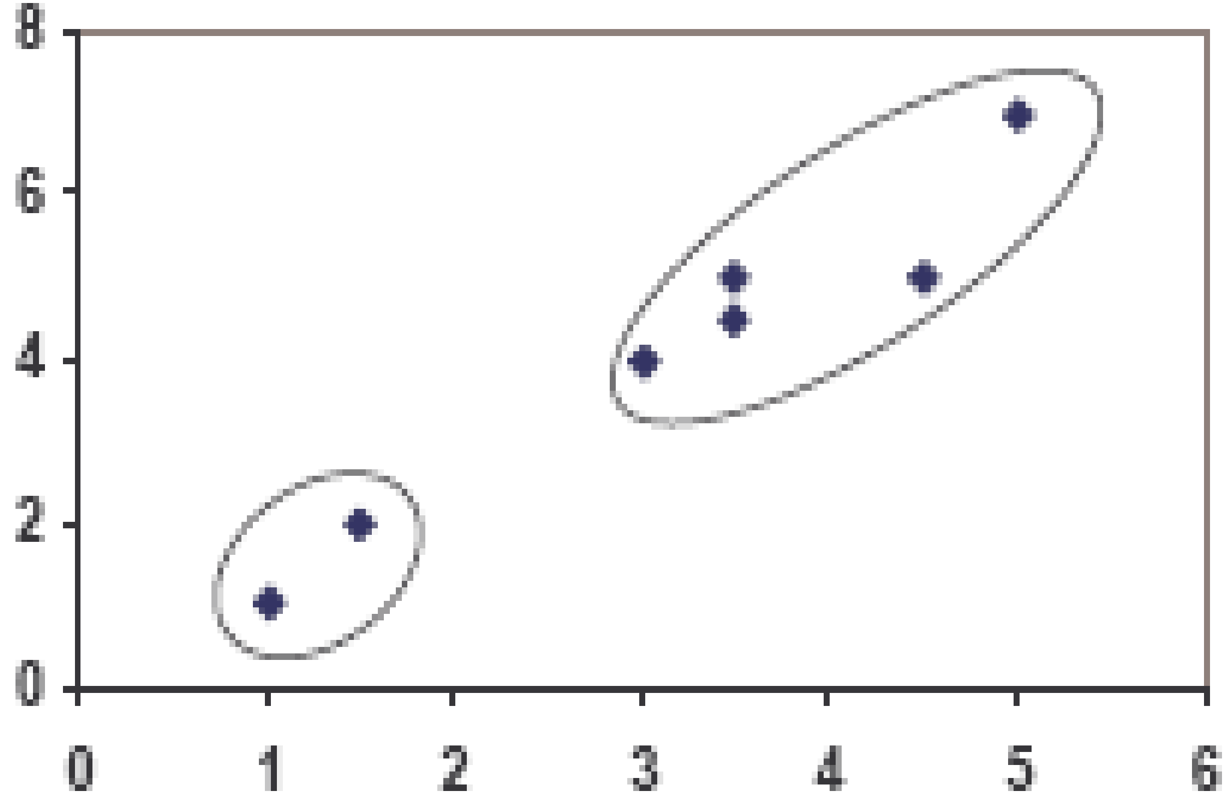
  The clusters obtained are:

  {1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0.56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



(with K=3)

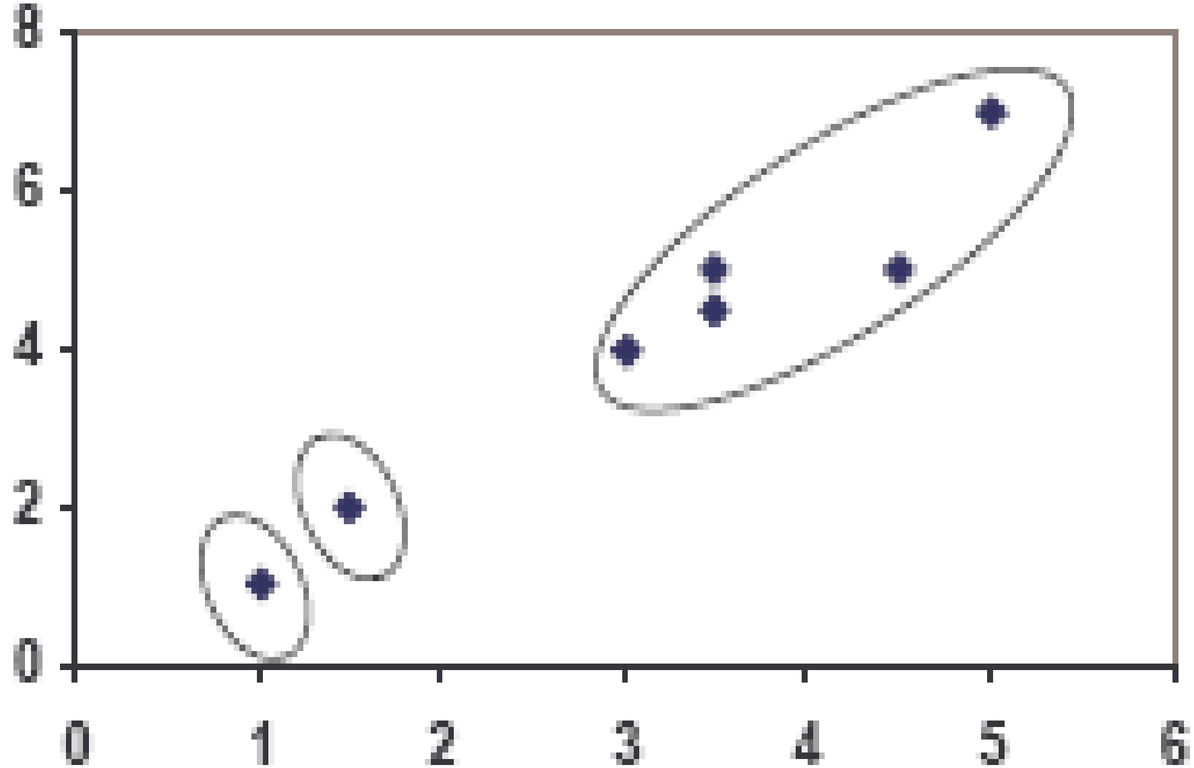
Individual	m1=1	m2=2	m3=3	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.8	3
7	4.30	3.20	0.71	3

Step 1

Individual	m1 (1.0, 1.0)	m2 (1.5, 2.0)	m3 (3.9, 5.1)	cluster
1	0	1.11	5.02	1
2	1.12	0	3.92	2
3	3.61	2.5	1.42	3
4	7.21	6.10	2.2	3
5	4.72	3.61	0.41	3
6	5.31	4.24	0.61	3
7	4.30	3.20	0.72	3

Step 2

PLOT



References

- *Artificial Intelligence: Foundations of Computational Agents*, second edition, Cambridge University Press 2017AA
- http://www.cs.cmu.edu/afs/andrew/course/15/381-f08/www/lectures/clustering.pdf
- https://kjambi.kau.edu.sa/GetFile.aspx?id=187901&Lng=AR&fn=k-mean-clustering.ppt