PYTHON PROGRAMMING AND MACHINE LEARNING

CLUSTERING

Yunghans Irawan (yirawan@nus.edu.sg)

Objectives

- Understand the application of clustering algorithms in machine learning
- Understand the following clustering machine learning algorithms:
  - K-Means (centroid based)
  - Aggomerative Clustering (connectivity based/ hierarchical)
  - DBSCAN (density based)

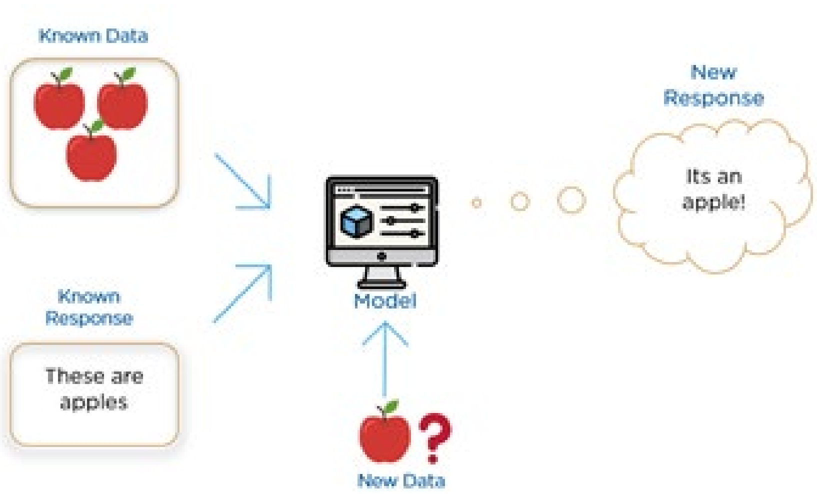
Clustering Objective

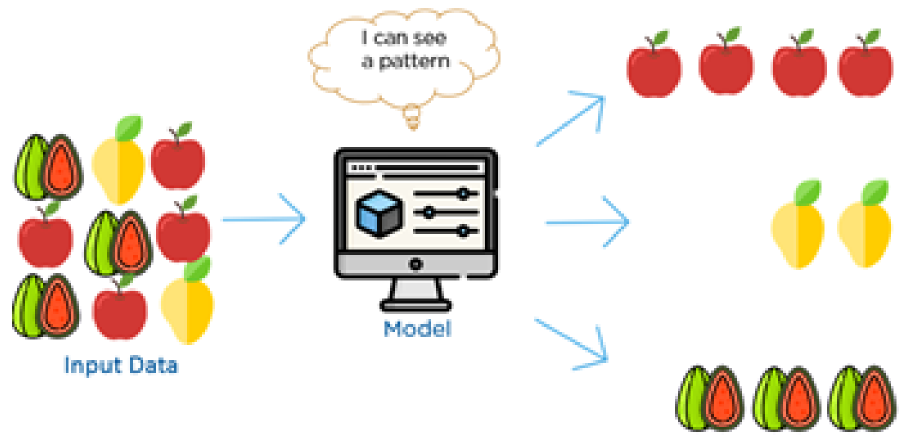
- Given a set of data:
  - Find possible clusters (groupings) of the data
  - The aim is that the data should be grouped in such a way that points within a single cluster are very similar and points in a different cluster are different
  - There are multiple possible answers

- Goal: Data exploration / discover hidden structure in data
  - Often used to convert a unsupervised learning problem into supervised learning

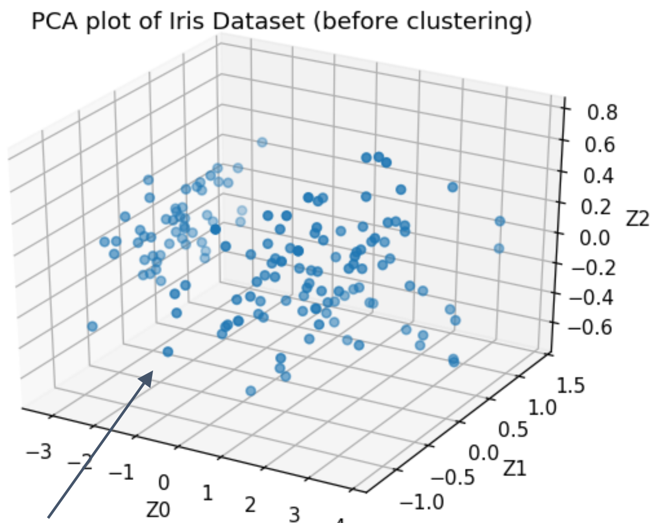
Supervised vs. Unsupervised

- Supervised Learning: find known answers
- Unsupervised Learning: find unknown patterns

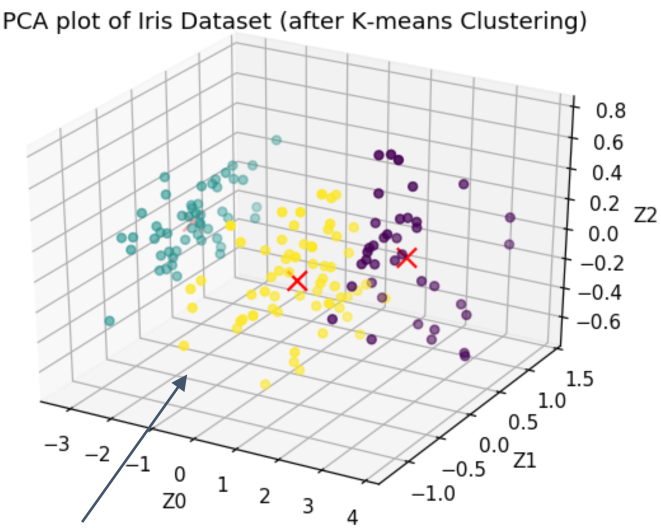




Clustering is unsupervised

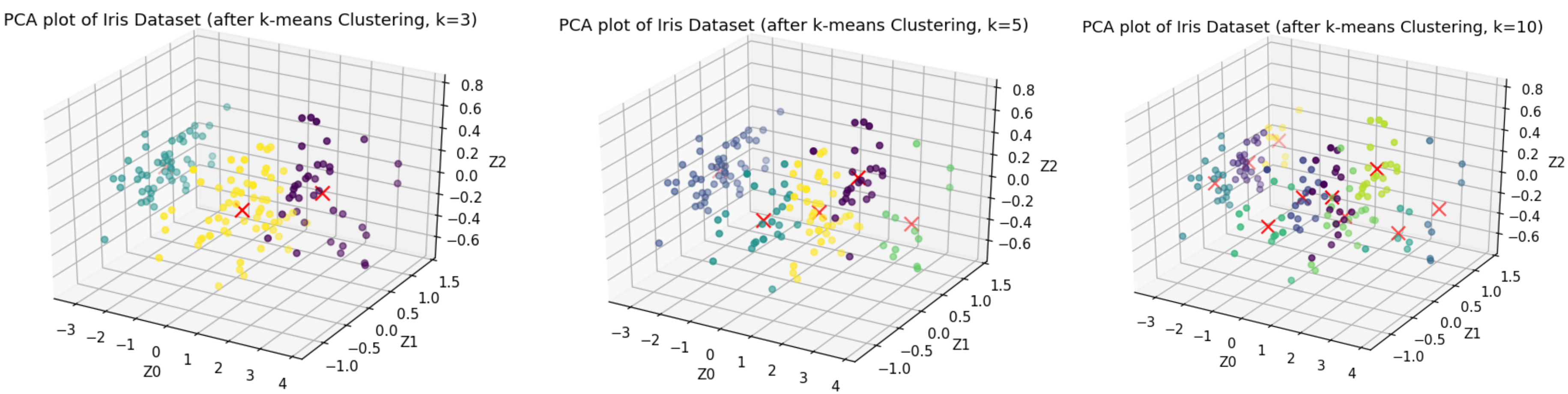


Start with
no labels



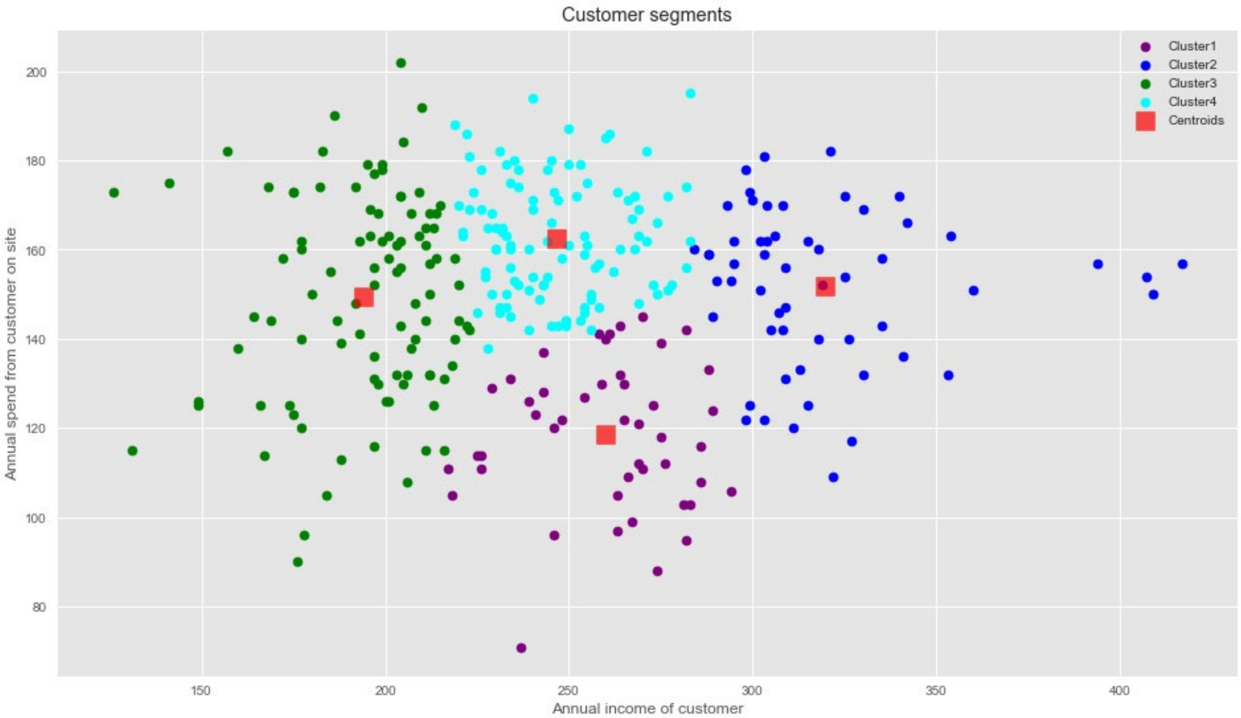
Labels added
based on
clusters found

Clustering can have multiple outcomes

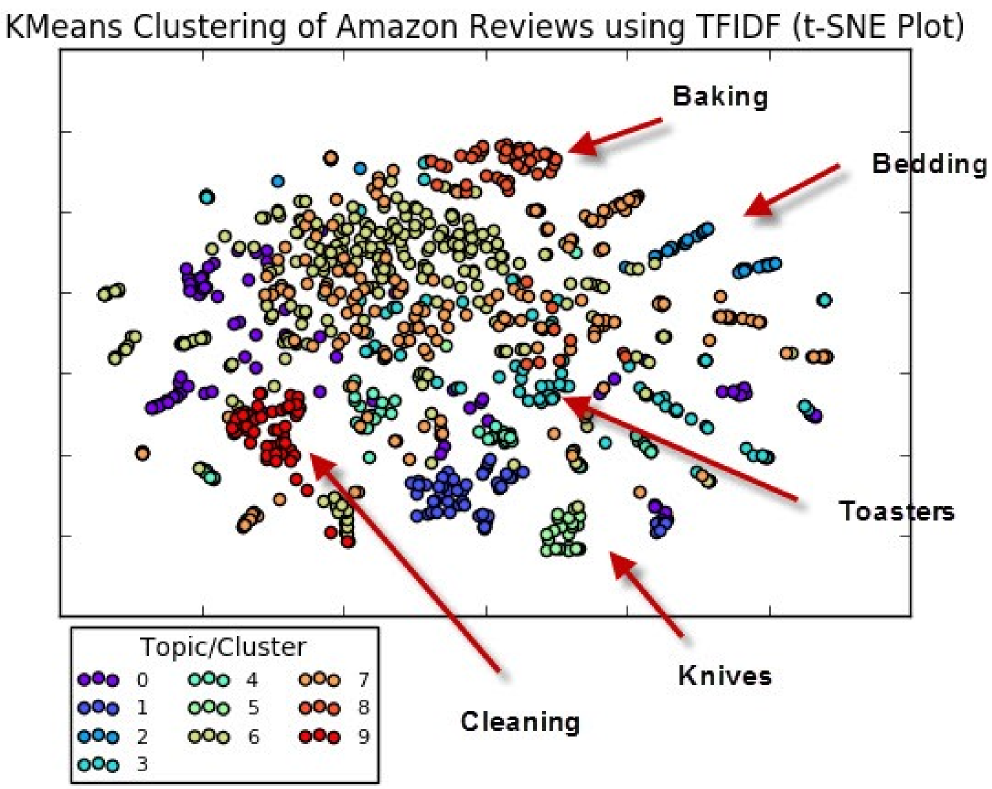


Compare metrics, manual inspection of clusters, ...

Finding Customer Segments



Finding Review Topics



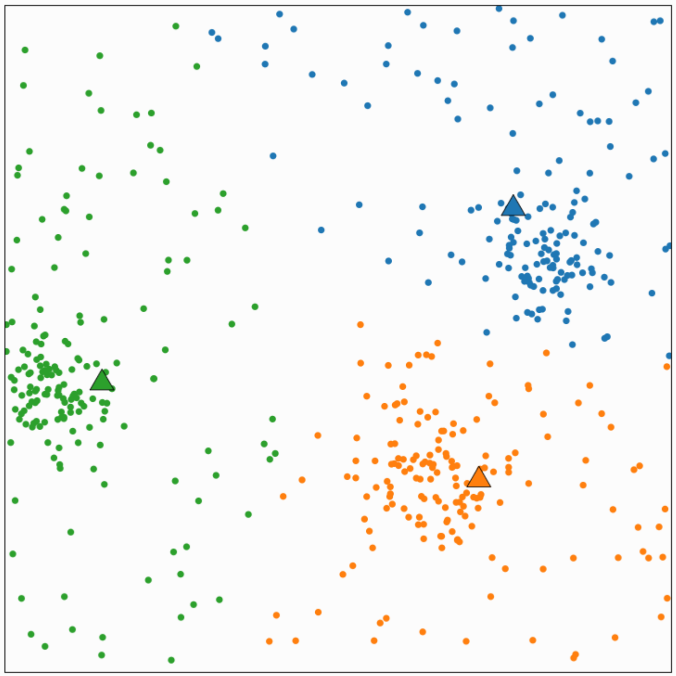
K-MEANS CLUSTERING

K-means Clustering

- Hyperparameter: k (number of clusters)
- Randomly initialize k centroids from samples
- For each sample
  - Compute distances from each centroid
  - Assign cluster from closest centroid
- Update centroids to the mean of the member samples
- Repeat 2 and 3 until centroids stop moving

Interactive Demo

Visualizing K-Means Clustering



Mean square point-centroid distance: 13119.49

The k-means algorithm is an iterative method for clustering a set of N points (vectors) into k groups or clusters of points.

Algorithm

Repeat until convergence:

Find closest centroid
Find the closest centroid to each point, and group points that share the same closest centroid.

Update centroid
Update each centroid to be the mean of the points in its group.

Update centroid

Data

Clustered points — Random

Number of clusters : 3

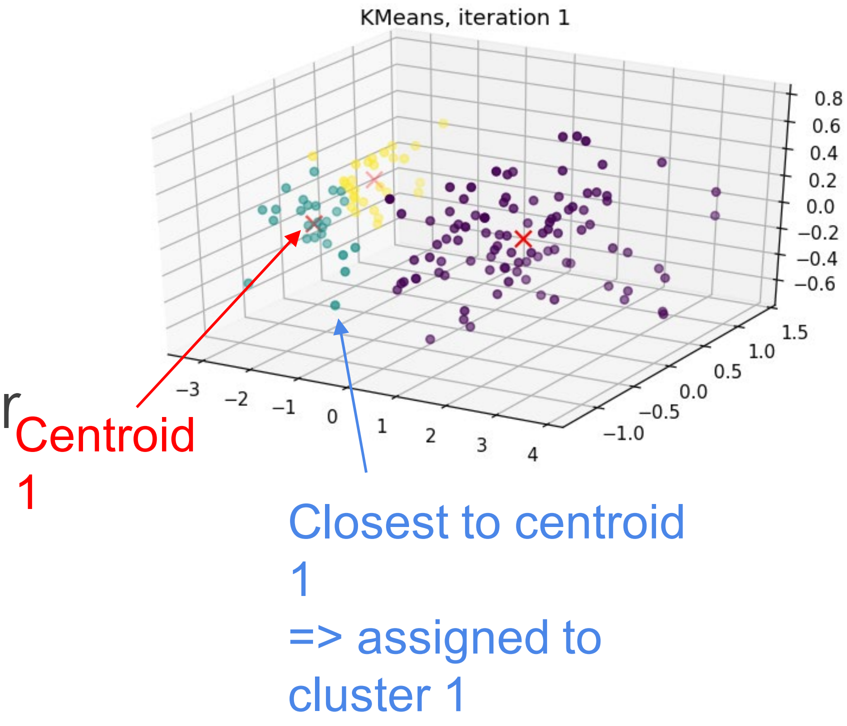
Number of centroids : 3

New points | New centroids

K-means Clustering

Hyperparameter: k (number of clusters)

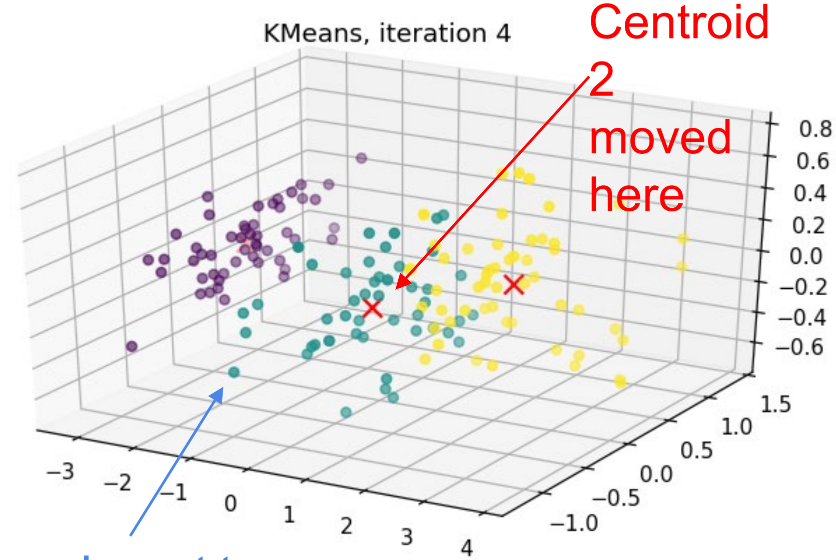
1. **Randomly initialize k centroids from samples**
2. **For each sample**
   a) **Compute distances from each centroid**
   b) **Assign cluster from closest centroid**
3. Update centroids using mean of member samples
4. Repeat 2 and 3 until centroids stop moving



K-means Clustering

Hyperparameter: k (number of clusters)

1. Randomly initialize k centroids from samples
2. For each sample
   a) Compute distances from each centroid
   b) Assign cluster from closest centroid
3. **Update centroids using mean of member samples**
4. **Repeat 2 and 3 until centroids stop moving**



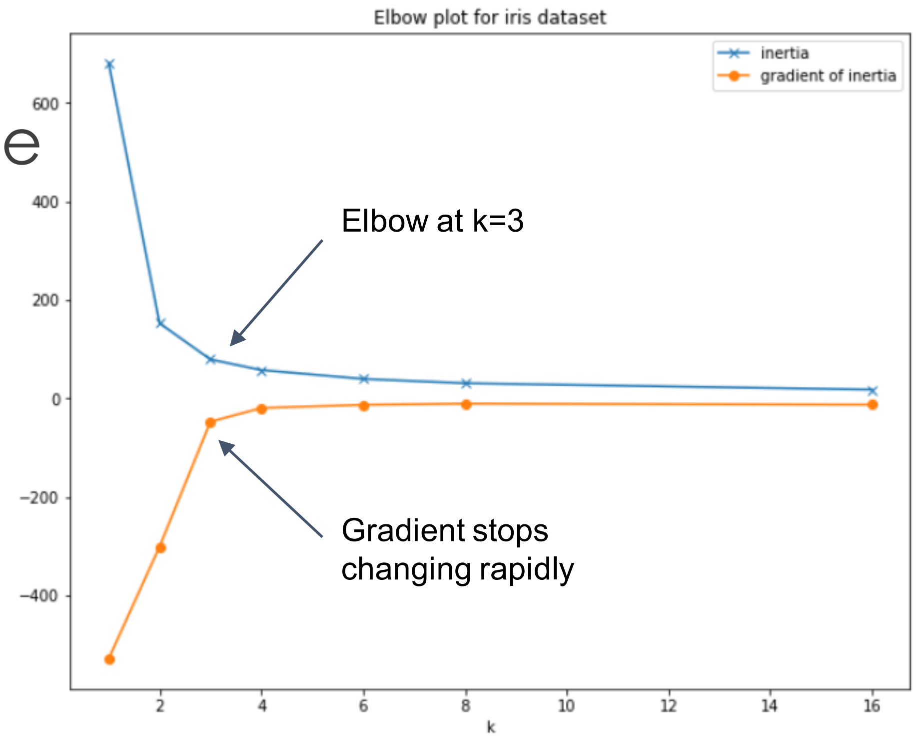
Centroid 2 moved here

Now closest to centroid 2 => updated to cluster 2

Selecting k for K-means

Empirical way

- Inertia: sum of squared distances of each sample to its closest centroid
- Inertia measures cluster compactness



Reasoning

- Fewer clusters is better
- Elbow is when inertia stops decreasing dramatically

Performing K-Means Clustering (1)

Feature 1	Feature 2
1	1
2	2
3	3
4	4
5	5

K = 2

First we randomly choose 2 centroids from the sample.

Centroid 1	1	1
Centroid 2	2	2

We calculate distance between each samples to the centroids. We use the Euclidean distance formula:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Feature 1	Feature 2	Distance to Centroid 1	Distance to Centroid 2	Cluster
1	1	0	1.414213562	1
2	2	1.414213562	0	2
3	3	2.828427125	1.414213562	2
4	4	4.242640687	2.828427125	2
5	5	5.656854249	4.242640687	2

Performing K-Means Clustering (2)

Feature 1	Feature 2
1	1
2	2
3	3
4	4
5	5

We calculate the new centroids by taking the average from the members of the cluster.

Centroid 1	1	1
Centroid 2	3.5	3.5

We calculate distance between each samples to the centroids. We use the Euclidean distance formula:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Feature 1	Feature 2	Distance to Centroid 1	Distance to Centroid 2	Cluster
1	1	0	3.535533906	1
2	2	1.414213562	2.121320344	1
3	3	2.828427125	0.707106781	2
4	4	4.242640687	0.707106781	2
5	5	5.656854249	2.121320344	2

Performing K-Means Clustering (3)

Feature 1	Feature 2
1	1
2	2
3	3
4	4
5	5

We calculate the new centroids by taking the average from the members of the cluster.

Centroid 1	1.5	1.5
Centroid 2	4	4

We calculate distance between each samples to the centroids. We use the Euclidean distance formula:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Feature 1	Feature 2	Distance to Centroid 1	Distance to Centroid 2	Cluster
1	1	0.707106781	4.242640687	1
2	2	0.707106781	2.828427125	1
3	3	2.121320344	1.414213562	2
4	4	3.535533906	0	2
5	5	4.949747468	1.414213562	2

As the cluster assignment remains the same, we stop the iteration with the final centroids

Assigning cluster for a new sample

- If we need to assign a new sample to the cluster we just assign the new sample to the closest centroid.

Centroid 1	1.5	1.5
Centroid 2	4	4

- Example:
  - (0,0) will be assigned to cluster 1
  - (4,5) will be assigned to cluster 2
  - (1.75, 1.75) can be assigned to cluster 1 or 2 depending on how we write the program as the distance to centroid 1 and 2 are the same.

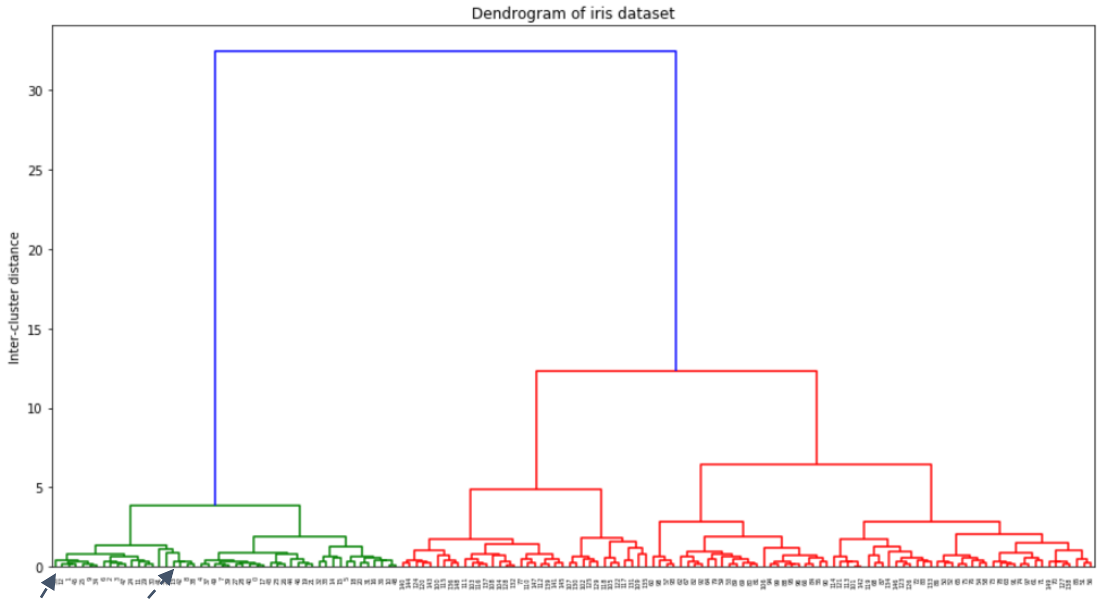
HIERARCHICAL CLUSTERING

Hierarchical Clustering

- Hierarchical clustering establish hierarchy between clusters.
- Two main approaches: top-down and bottom-up
- Top Down
  - Start with 1 cluster and split into more clusters
  - aka. Divisive Clustering
- Bottom Up
  - Start with N clusters (each node is a cluster) and merge into fewer clusters
  - aka. Agglomerative Clustering
  - “Agglomerative”: to collect or gather into a cluster or mass.

Hierarchical Clustering

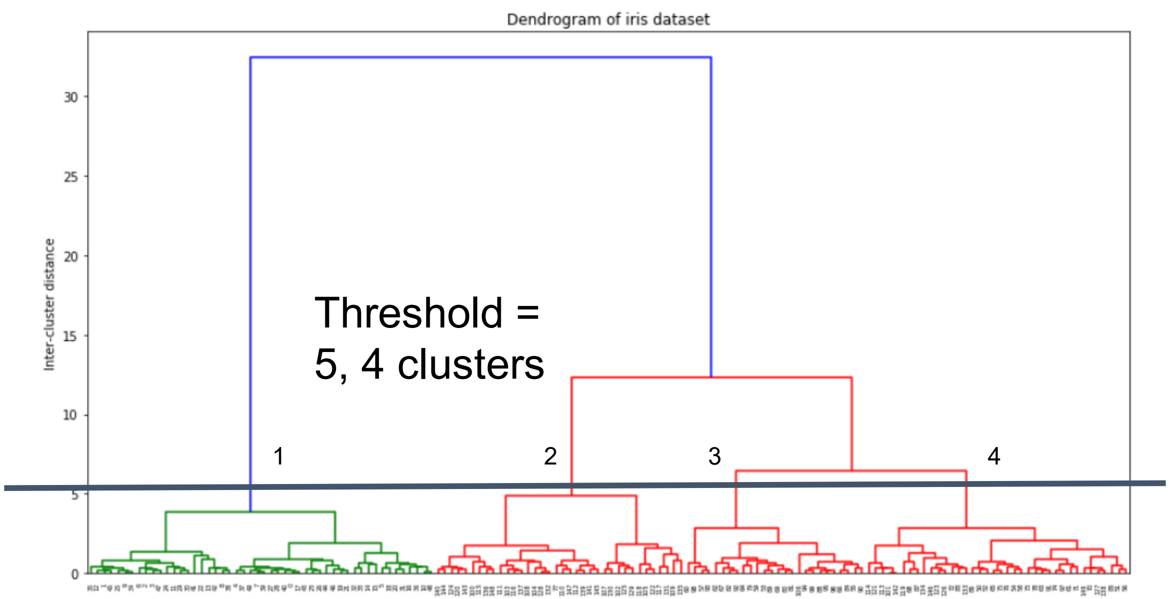
- Compute distances between samples
- Start with 1 sample in per cluster
    - Merge clusters by adjusting distance threshold



"per-sample" cluster (1 sample per cluster)

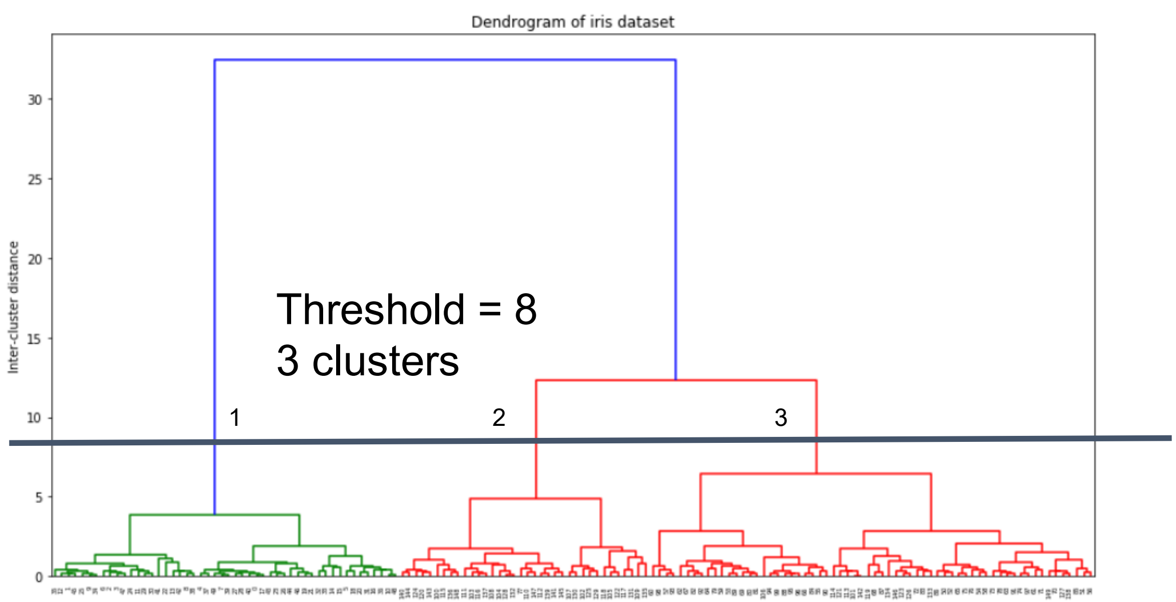
Hierarchical Clustering

- Compute distances between samples
- Start with each sample in its own cluster
  - Merge clusters by adjusting distance threshold

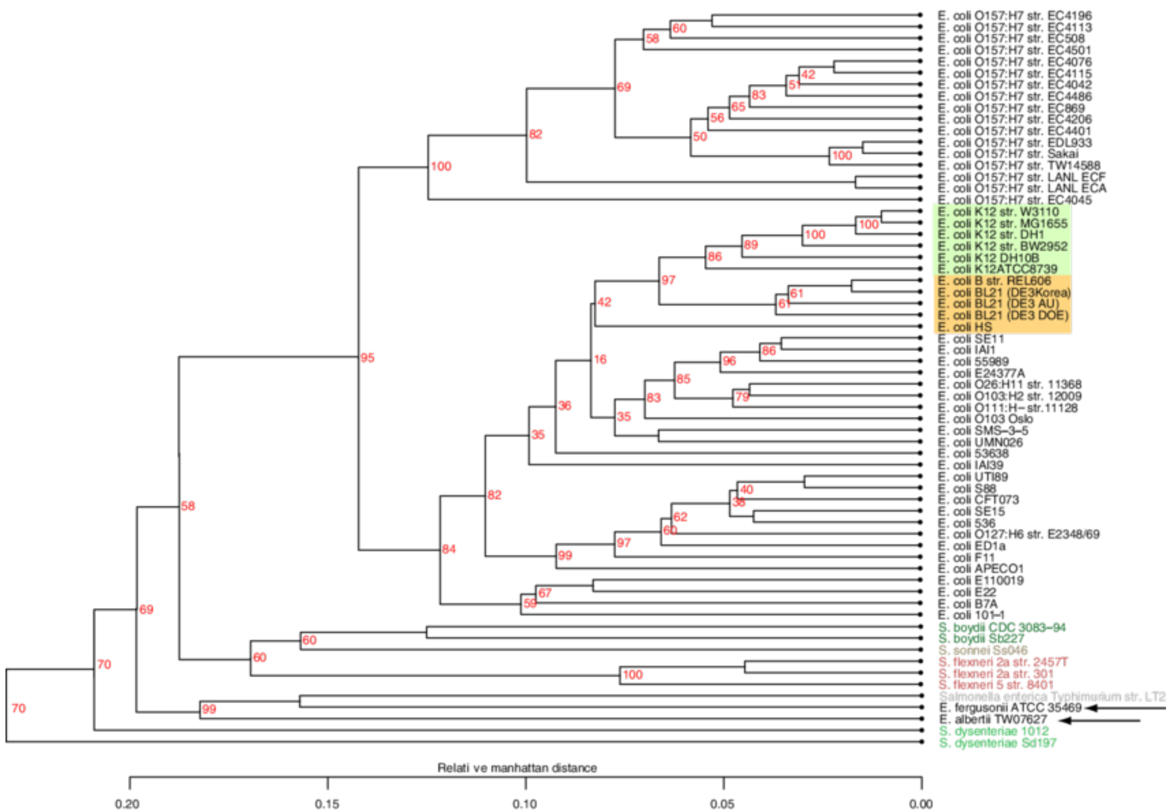


Hierarchical Clustering

- Compute distances between samples
- Start with each sample in its own cluster
  - Merge clusters by adjusting distance threshold



Clustering E. coli genomes



DBSCAN

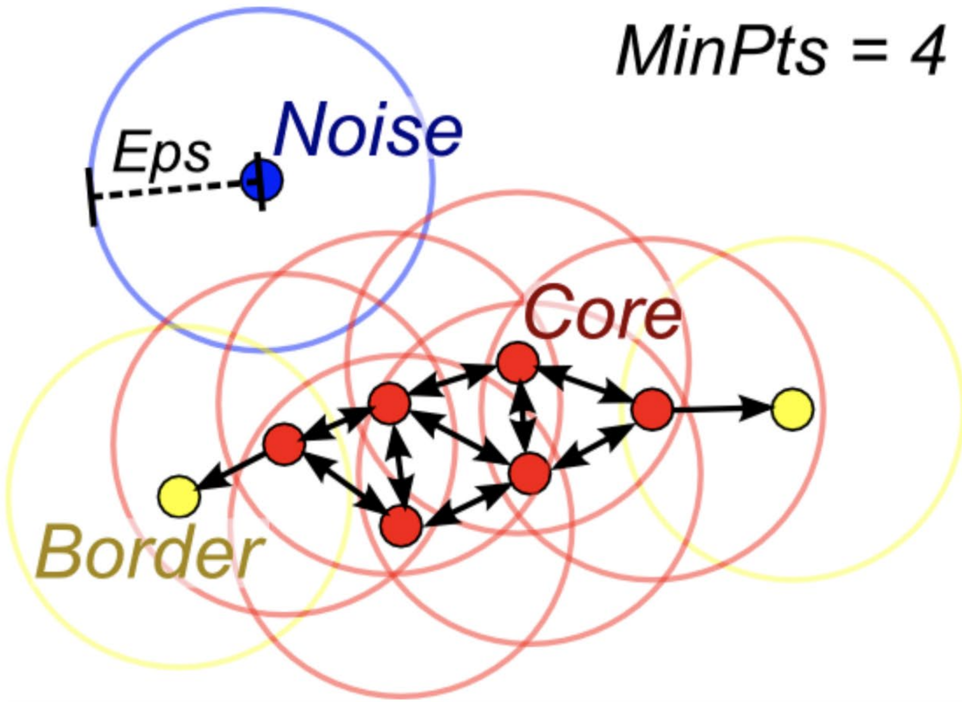
DBSCAN

- Stands for "Density Based Spatial Clustering of Applications with Noise”
- Advantages:
  - does not require the user to set the number of clusters beforehand
  - can capture clusters of complex shapes
  - can identify points that are not part of any cluster (very useful as outliers detector)
  - works by identifying points that are in crowded regions of the feature space, where many data points are close together (dense regions in feature space)
  - Points that are within a dense region are called core samples (or core points)

DBSCAN

- There are two parameters in DBSCAN: min_samples and eps
  - If there are at least min_samples many data points within a distance of eps to a given data point, that data point is classified as a core sample
  - core samples that are closer to each other than the distance eps are put into the same cluster by DBSCAN.

DBSCAN



Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

EVALUATION METRICS

Silhouette Coefficient

$$s = \frac{b-a}{max(a,b)}$$

$$-1 < s < 1$$

Bad clustering
(a >> b)

Good, dense clustering
(b >> a)

a: The mean distance between a sample and others in the **same cluster** (intra cluster distance)

b: The mean distance between a sample and all others in the **next nearest cluster** (inter cluster distance)

Homogeneity, Completeness, V-measure

- Use when labels are available
- Homogeneity: each cluster only contains members of 1 class
- Completeness: all members in 1 class are assigned to the same cluster
- V-measure:

$$2\frac{H.C}{H+C}$$

- Wait, aren't labels unavailable for an unsupervised learning problem?

Adjusted Random Index

- Given the knowledge of the ground truth class and our clustering algorithm assignments, the adjusted Rand index is a function that measures the **similarity** of the two assignments, ignoring permutations and with chance normalization.
- Compare the actual classes vs. cluster assignment
  - Measure how good is the clustering to serve as classification

Normalized Mutual Information

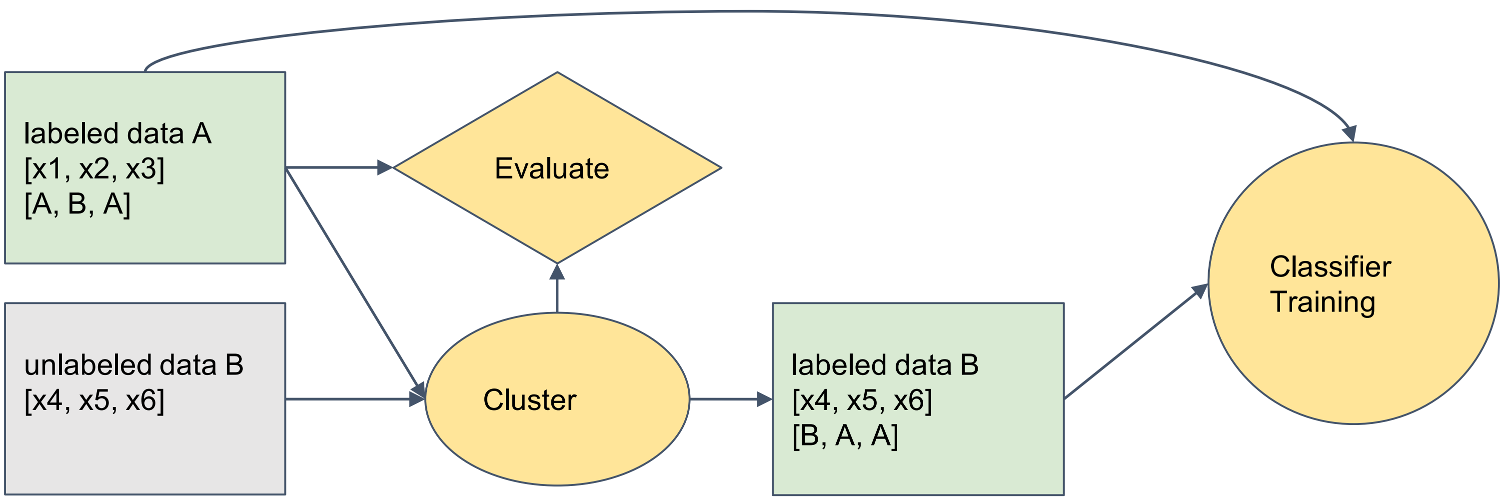
- Given the knowledge of the ground truth class and our clustering algorithm assignments, mutual information is a function that measures the **agreement** of the two assignments, ignoring permutations
- NMI (normalized mutual information) is more often used, AMI (adjusted mutual information) is more recent and normalized against chance.

ARI vs NMI

- Similarity vs. agreement?
  - They have the same objective, but measured using different theory and different mathematical formula
  - For those who is more mathematically inclined: http://jmlr.csail.mit.edu/papers/volume17/15-627/15-627 compares the two
- Conclusion
  - Both measures are correlated
  - Use **ARI** when the ground truth clustering has large equal sized clusters
  - Use **AMI/NMI** when the ground truth clustering is unbalanced and there exist small clusters

Semi-supervised Clustering

- When only a subset of training data is labeled
- Use clustering to predict labels for the unlabeled data
  - Evaluate clustering metrics using labeled data
- Train a classifier on the combined (labeled) dataset



HANDS ON: CLUSTERING

Applications of Clustering

- 10 interesting use cases of the K-means algorithm
- Entertainment: Song text mining and clustering
- Health: Clustering Medical Facilities
- Retail: Customer segmentation
- Manufacturing: Predicting failures in production lines
- Financial: Bank Marketing campaign analysis

Further study: Other clustering algorithms

- Kmeans++
- Self-organizing maps (SOMs)
- Cash Crops Clustering in Nepal
- Sequence clustering (Bioinformatics)
- Comparison of Sequence Clustering methods
- Biopython - toolkit for bioinformatics