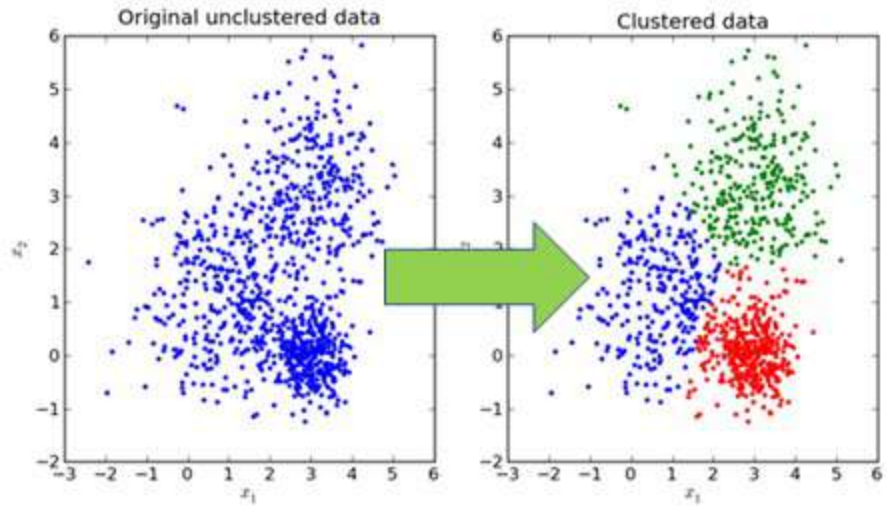
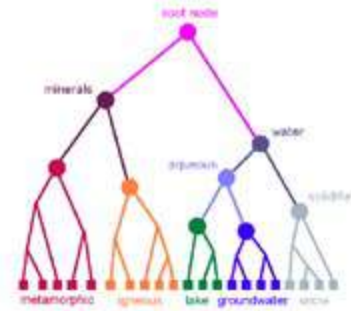
Master of Technology in Intelligent Systems

Problem Solving using Pattern Recognition

Module 4: Cluster Analysis

Charles Pang
Institute of Systems Science
National University of Singapore
Email: charlespang@nus.edu.sg





Topics

1. Introduction to Cluster Analysis
2. Application of Cluster Analysis
3. Types of Clustering Methods
4. How to Profile Clusters
5. How to Validate the Created Clusters
6. Limitations of Cluster Analysis
7. Demonstration of clustering using software tools (e.g. Python)

1. Introduction to Cluster Analysis

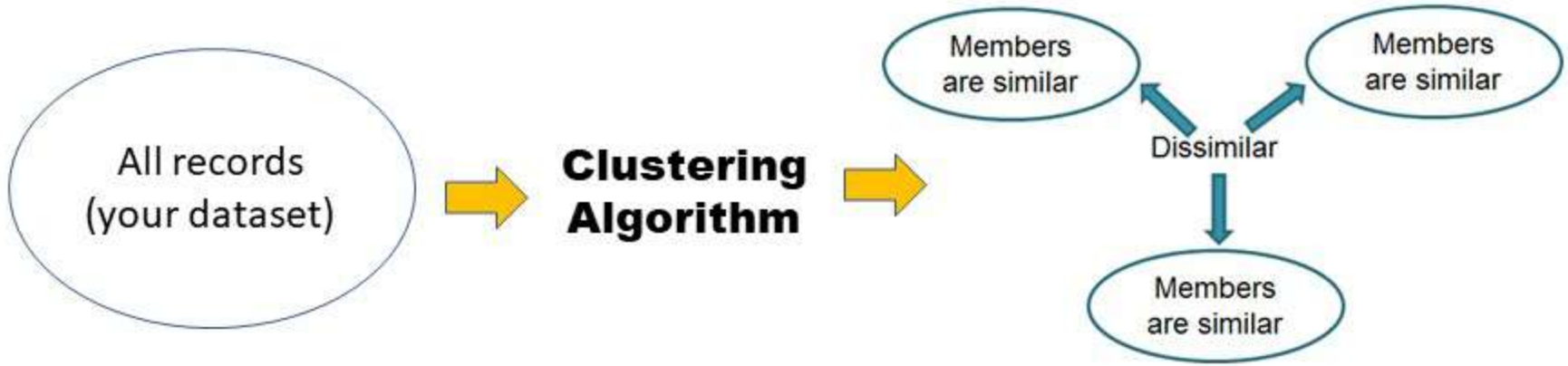
- Cluster Analysis (or simply "Clustering") is a **multivariate** data analysis technique
- It is an **unsupervised learning** method because there are no predefined classes or labels
- Clustering is useful for understanding complex business problems
  - Yields better understanding by uncovering **natural patterns** in data
  - **Reduces data dimensions** and make features more interpretable
  - **Divide and conquer** analysis of smaller, homogenous clusters may yield better results
  - Members of a cluster share the same characteristics and hence are likely to react in the same manner. E.g. "typical" buyers of products. We can **predict** new buyers by measuring their similarity to these clusters

Clustering vs Classification

- The terms “clustering”, “classification” are often used interchangeably.
- **Classification** is grouping observations into some known categories. The target category (e.g. fraud) is given upfront.
- **Clustering** is grouping observations into some unknown categories. The aim is to find homogenous subsets of observations and group them into clusters.
- **Segmentation** is sometimes used to refer to clustering techniques. But it can also refer to a more general process of dividing a dataset into partitions. E.g. divide the customers into Males and Females.

Goal of Clustering

- Clustering aims to draw partitions in your data to form clusters such that:
  - Observations **within** a cluster are **similar** to each other
  - Observations **between** clusters are **dissimilar** to each other
- Clustering algorithms are used to automatically draw such partitions:



- The assumption is that your data is **not uniformly distributed** but rather they comprise of natural clumps.

A Cluster Analysis Example

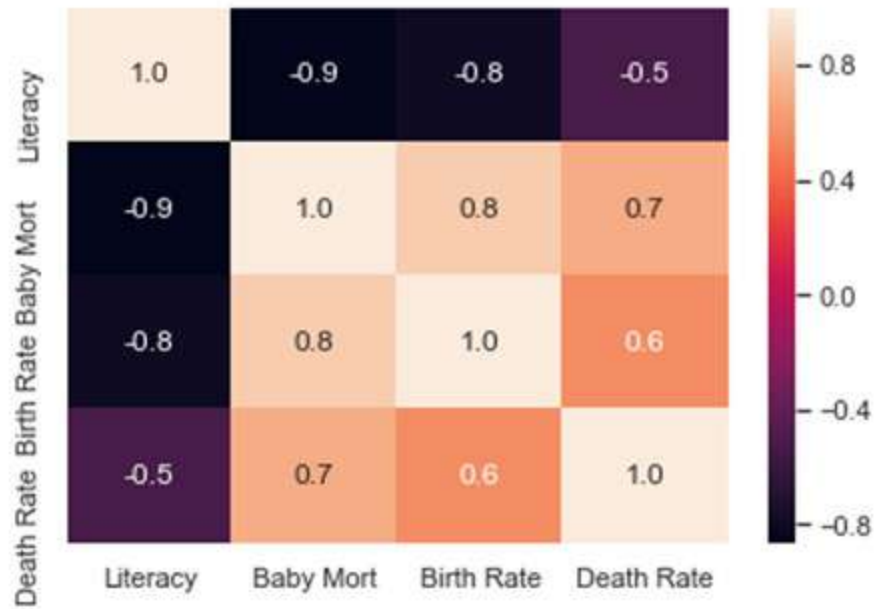
- We would like to find **meaningful groupings** of countries based on the following characteristics:
  - Literacy Rates
  - Baby Mortality
  - Birth Rates
  - Death Rates

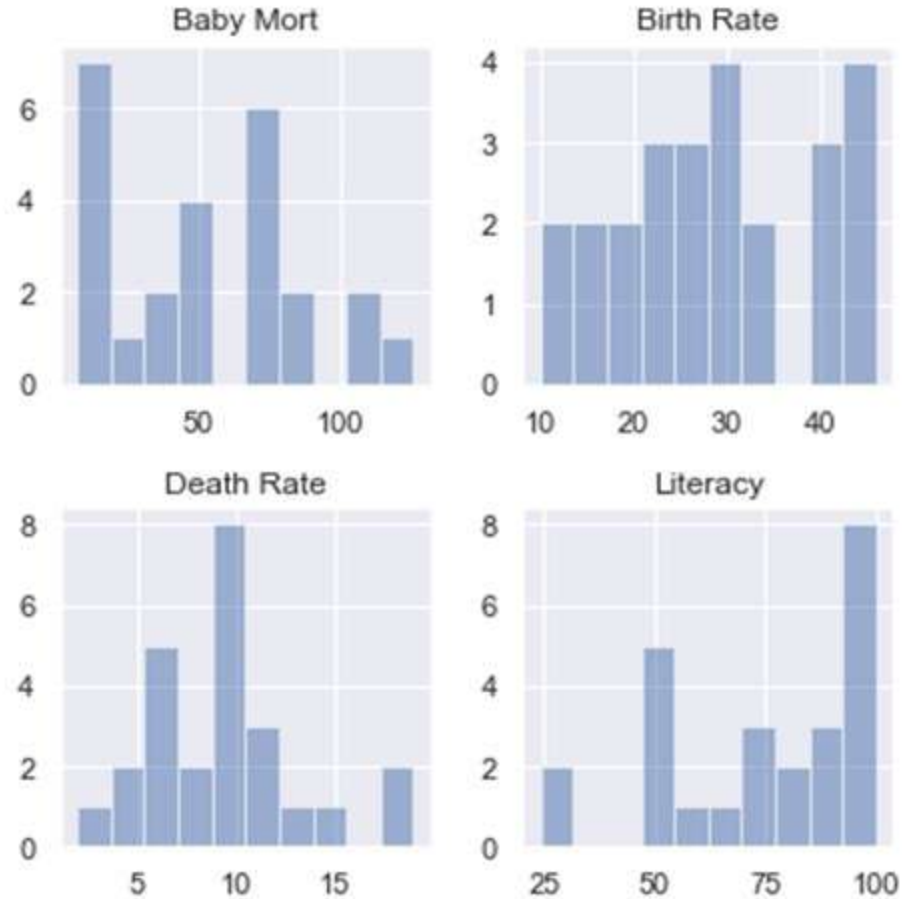
SN	Country	Literacy	Baby Mort	Birth Rate	Death Rate
1	Argentina	95	25.6	20	9
2	Australia	100	7.3	15	8
3	Bolivia	78	75	34	9
4	Cameroon	54	77	41	12
5	Chile	93	14.6	23	6
6	China	78	52	21	7
7	Costa Rica	93	11	26	4
8	Egypt	48	76.4	29	9
9	Ethiopia	24	110	45	14
10	Greece	93	8.2	10	10
11	Haiti	53	109	40	19
12	India	52	79	29	10

- A desirable output will be countries that share similar characteristics will be grouped together. Each grouping will be distinct from each other.

Summary Statistics

	Literacy	Baby Mort	Birth Rate	Death Rate
count	25.00	25.00	25.0	25.00
mean	73.48	52.91	29.0	9.36
std	22.47	35.71	11.1	3.90
min	24.00	7.30	10.0	2.00
25%	54.00	14.60	21.0	7.00
50%	78.00	52.00	28.0	9.00
75%	93.00	76.40	40.0	11.00
max	100.00	126.00	46.0	19.00



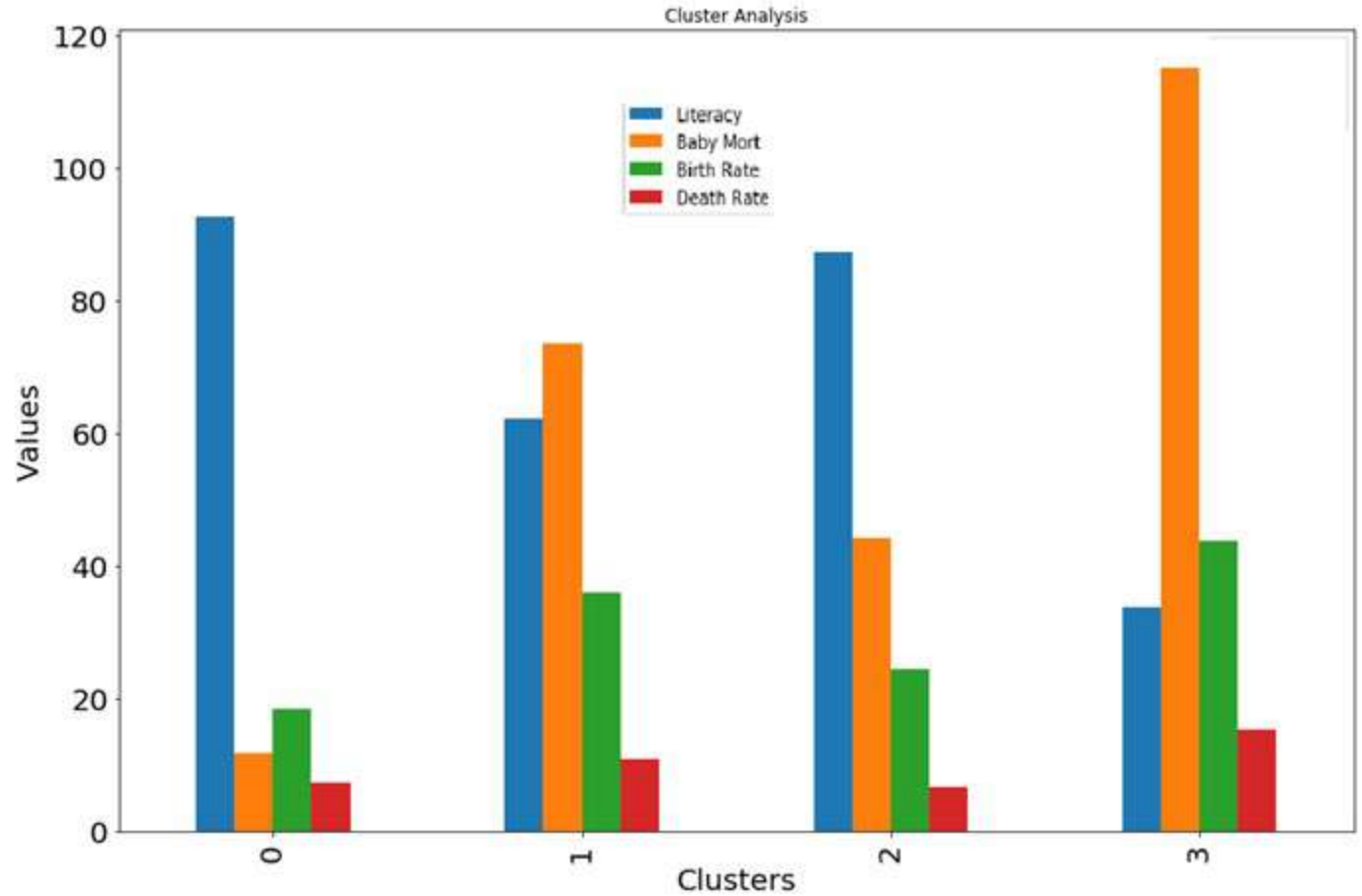


Cluster Analysis (output)

Results of the Cluster Analysis with 4 clusters

Cluster	0	1	2	3
Literacy	92.625	62.111	87.2	33.667
Baby Mort	11.862	73.544	44.2	115.000
Birth Rate	18.500	36.000	24.4	43.667
Death Rate	7.250	10.778	6.6	15.333
Count	8.000	9.000	5.0	3.000

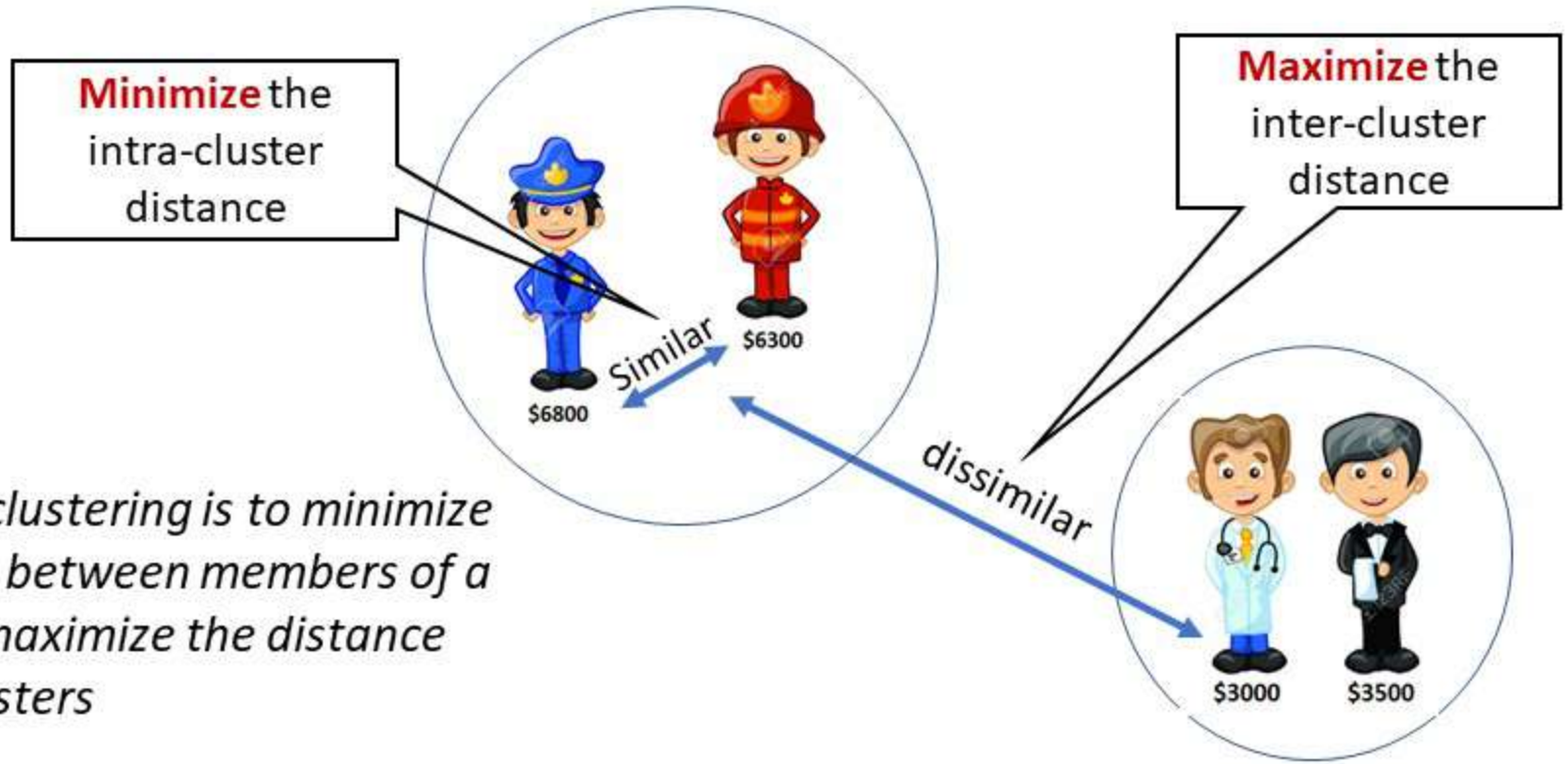
Country	Nigeria	Vietnam	Haiti
Argentina	Nicaragua	Mexico	Ethiopia
Australia	Kenya	China	Somalia
USA	Indonesia	Phillippines	
Chile	Egypt	Thailand	
Costa Rica	Cameroon		
Kuwait	Bolivia		
Greece	India		
Italy	Zambia		



Is there a meaningful pattern?

How do we measure similarity ?

- A **similarity measure** is used to decide whether data points are far apart or near to each other. Similarity measures are derived by using various mathematical formulas.
- With a single variable like income, it is easy to find their similarity:

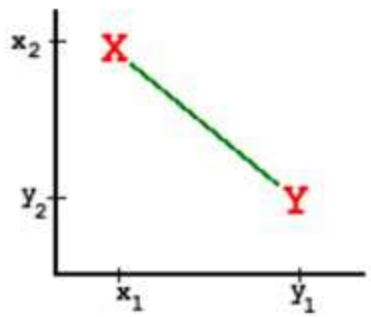


The goal of clustering is to minimize the distance between members of a cluster but maximize the distance between clusters

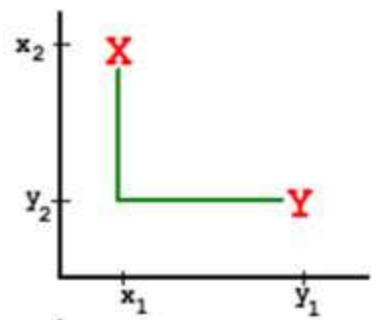
Calculating Similarities

- Mathematical formulas are used to calculate the "distance" between data observations
- If each data point is represented as numerical variables then it is relatively easy. We can use the following formulas:
  - Euclidean distance
  - City-block (Manhattan) distance

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$



$$\sum_{i=1}^{k}|x_i - y_i|$$



Euclidean Distance Measure

- Euclidean Distance is the most popular distance measure
- Given two cases *i*, and *j* (in the p-dimensional space) the distance is defined by

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}$$

ID	Age	Income
S1234567D	21	5600
S3456782X	56	4600
B1725353Y	39	7000

$\sqrt{(21-56)^2 + (5600-4600)^2}$

- Euclidean distance is **highly scale dependent**. Therefore, the data is usually normalized first.

Similarity Measure for Non-numerical data

Ordinal Data

- Customer Satisfaction: {*Pitiful, Poor, Satisfactory, Good, Excellent*}
- The data value used for clustering is the **index** of the ordering- i.e. treated as if it were continuous data

SN	Satisfaction	Job
1	Poor	lawyer
2	Excellent	doctor
3	Pitiful	lawyer

Dist(*Poor*, *Excellent*) =1-¾= 0.25
Dist(*Poor*, *Pitiful*) =1-¼= 0.75

Nominal Data

- Job: {Lawyer, Doctor, Teacher}
- The data value used for clustering is the **exact category name**.

SN	Satisfaction	Job
1	Poor	lawyer
2	Excellent	doctor
3	Pitiful	lawyer

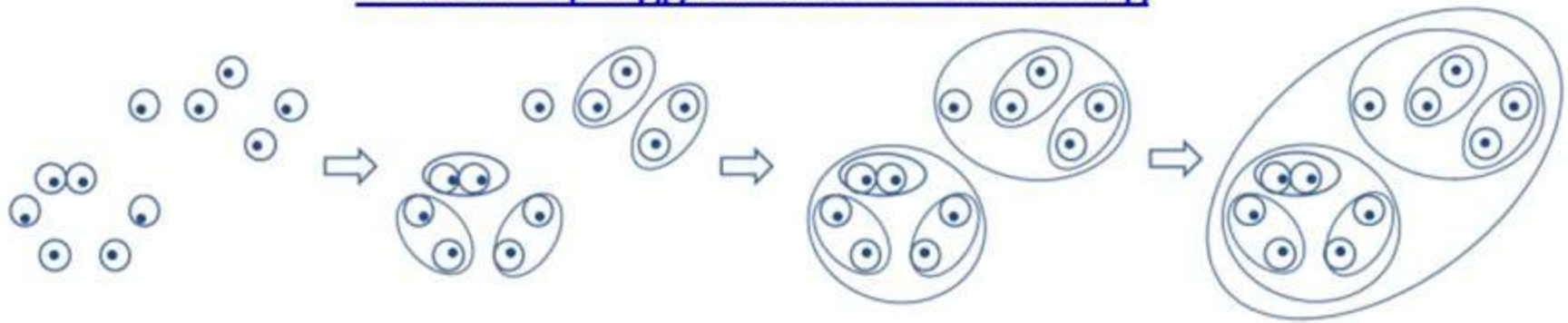
Dist(*Lawyer*, *doctor*) = 1
Dist(*lawyer*, *lawyer*) = 0

Types of Clustering methods

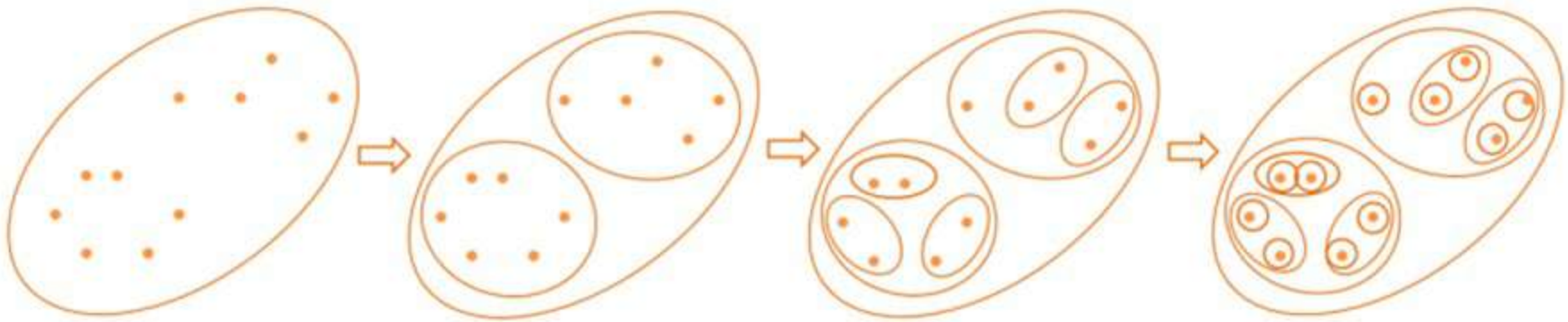
- Hierarchical Clustering
  - Agglomerative
  - Divisive
- Partition Clustering
  - K-Means Clustering
  - Density-based Clustering
- Fuzzy Clustering

Two Variants of Hierarchical Clustering

Bottom Up Agglomerative Clustering



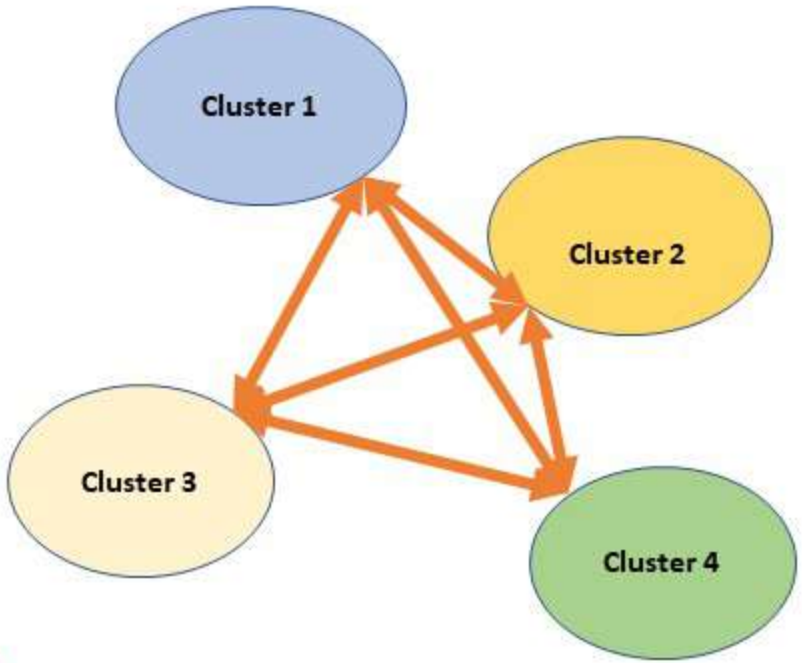
Top Down Divisive Clustering



Example

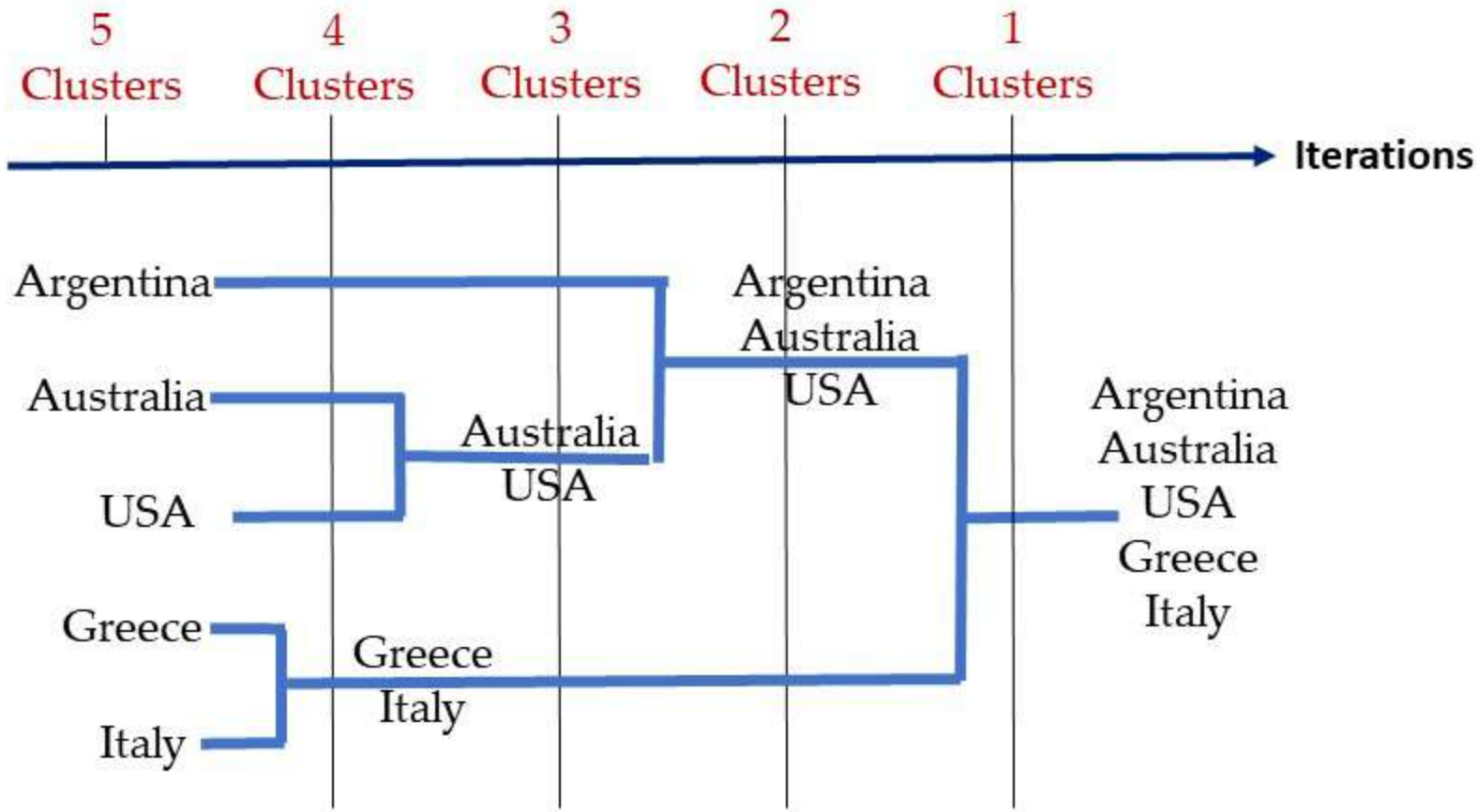
Objective is to find groupings of countries that share similar characteristics in terms of:

- Literacy
- Baby Mortality
- Births
- Deaths



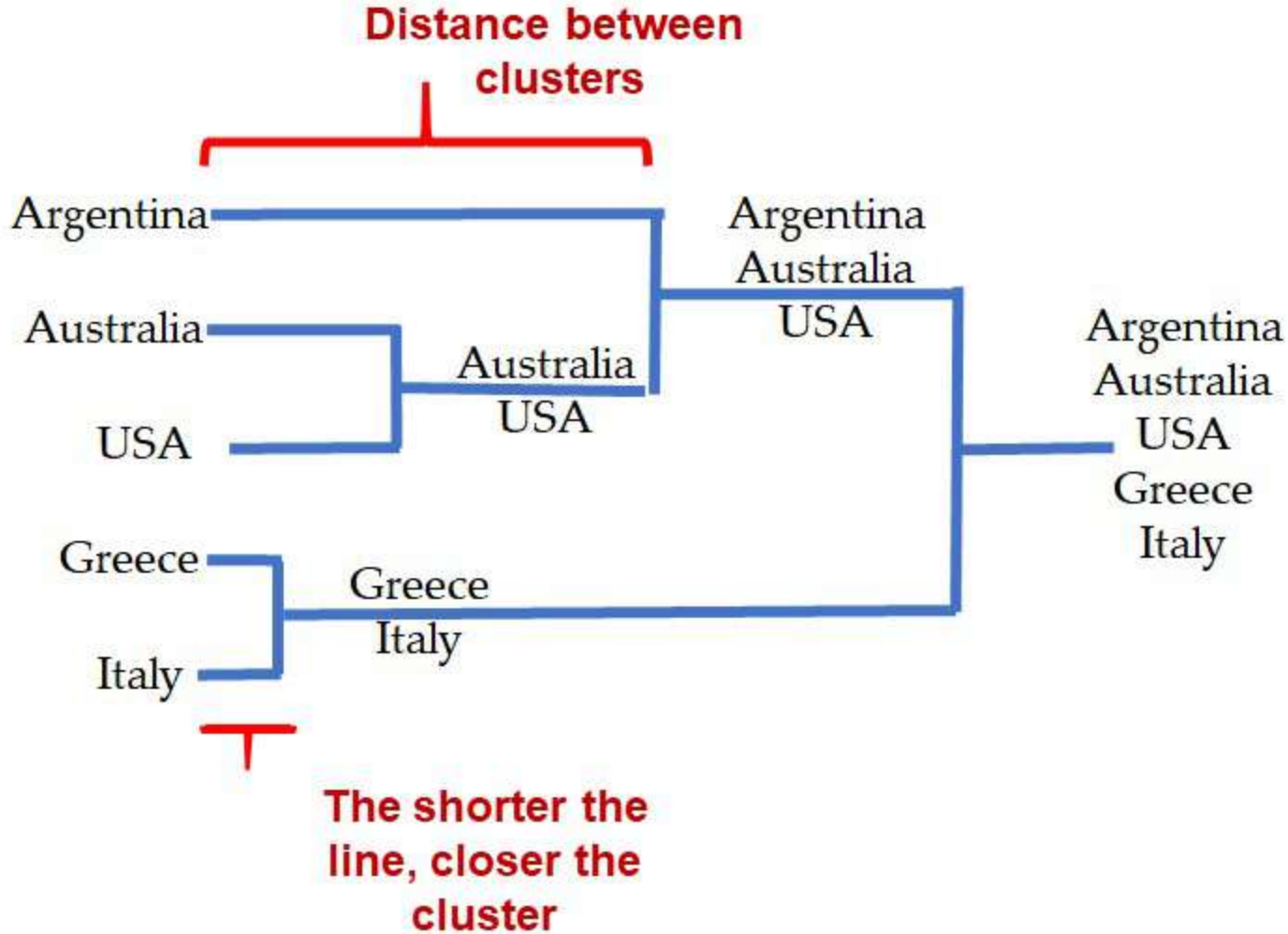
Country	Literacy	Baby Mort	Birth Rate	Death Rate
Argentina	95	25.6	20	9
Australia	100	7.3	15	8
Bolivia	78	75	34	9
Cameroon	54	77	41	12
Chile	93	14.6	23	6
China	78	52	21	7
Costa Rica	93	11	26	4
Egypt	48	76.4	29	9
Ethiopia	24	110	45	14
Greece	93	8.2	10	10
Haiti	53	109	40	19
India	52	79	29	10
Indonesia	77	68	24	9
Italy	97	7.6	11	10
Kenya	69	74	42	11
Kuwait	73	12.5	28	2
Mexico	87	35	28	5
Nicaragua	57	52.5	35	7
Nigeria	51	75	44	12
Phillippine	90	51	27	7
Somalia	24	126	46	13
Thailand	93	37	19	6
USA	97	8.1	15	9
Vietnam	88	46	27	8
Zambia	73	85	46	18

How does Agglomerative Clustering work?



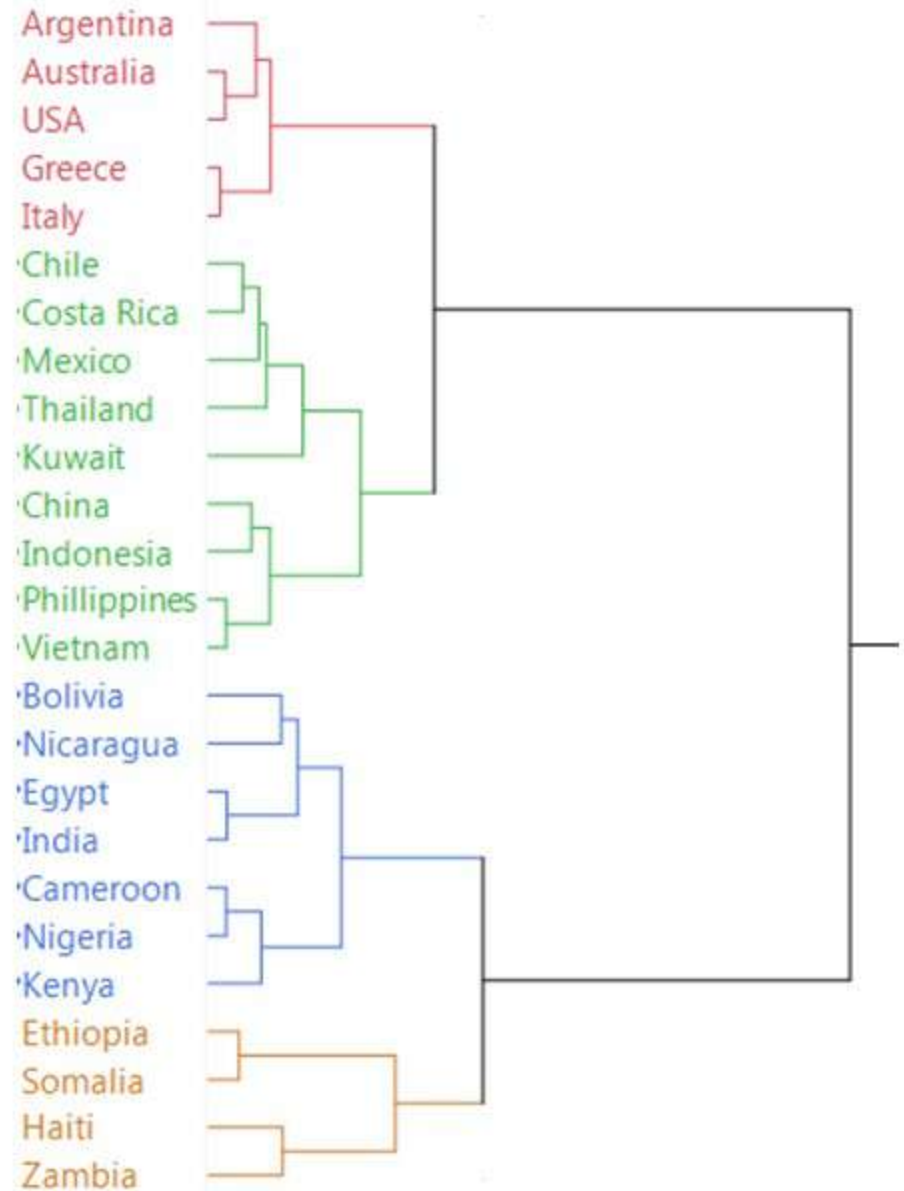
Iteration#1: Greece + Italy
Iteration#2: Australia + USA
Iteration#3: [Australia + USA] + Argentina

Interpreting Dendograms



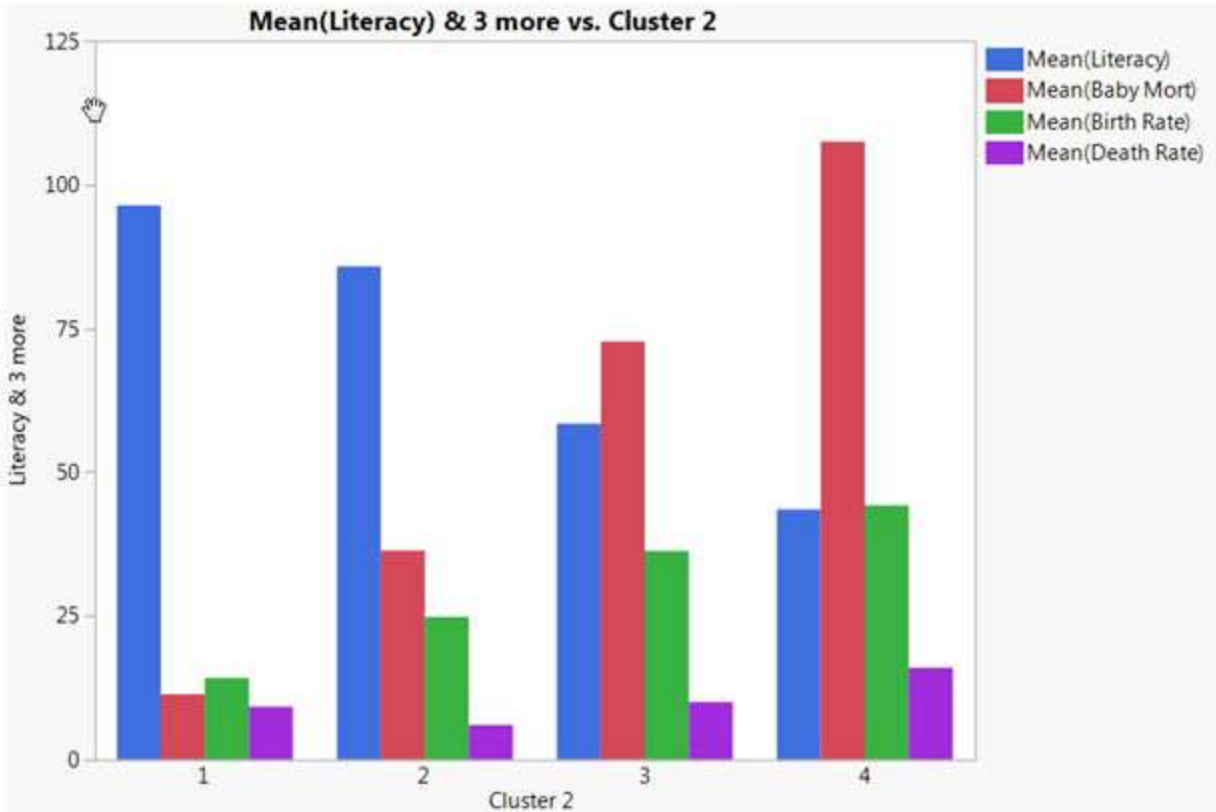
Output of Agglomerative Clustering

- A **dendrogram** is a tree diagram used to illustrate the arrangement of the clusters
- Read from left to right
  1. Greece + Italy
  2. Australia + USA
  3. Philippines + Vietnam
  4. Cameroon + Nigeria …
- This color display is for 4 clusters
  - C1: Argentina … Italy
  - C2: Chile … Vietnam
  - C3: Bolivia … Kenya
  - C4: Ethiopia … Zambia

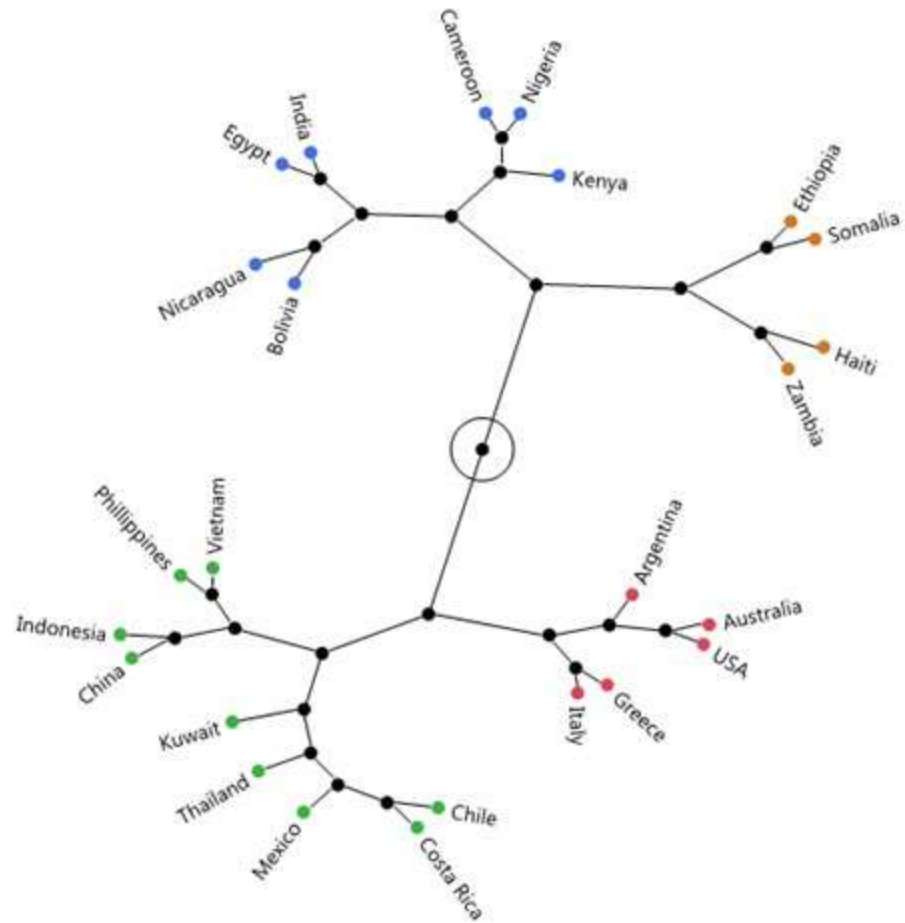


Cluster Visualization

Bar Chart



Constellation Plot



Agglomerative Clustering Algorithm

Start with ***N* records** in the dataset

- **Step 1**: Assign each record as its own cluster
- **Step 2**: Calculate the distances between each cluster
- **Step 3**: Find the closest pair of clusters and merge them into a single cluster
- **Step 4**: Repeat steps 2-4 until all records are clustered into a single cluster consisting of *N* members

Inter-cluster Distance Measure

Previously, we discussed how to compute the similarity between objects in a cluster. But how do we define **dissimilarities** between clusters?

The following are some of these methods:

1. **Single Linkage**

   Measure the distance between the closest points of two clusters.

2. **Complete Linkage**

   Measure the distance between the farthest points of two clusters.

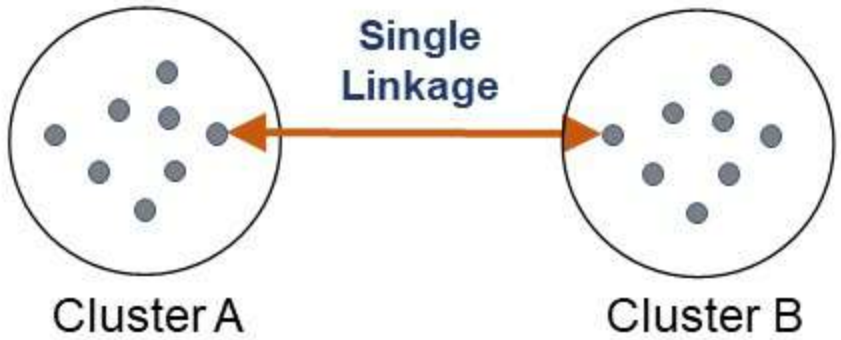
3. **Centroid**

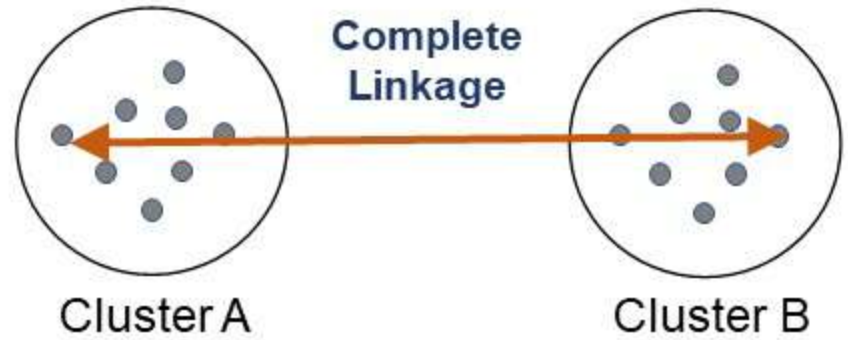
   Measure the distance between the centroids of two clusters.

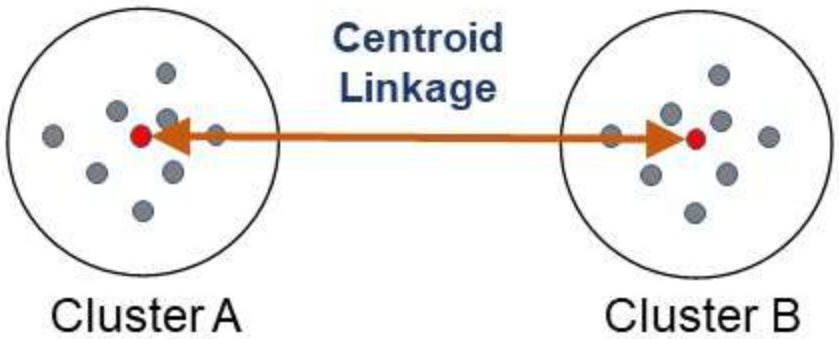
4. **Average**

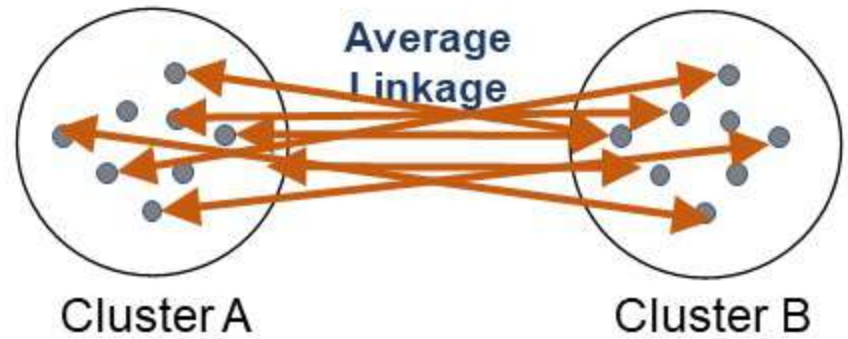
   Measure the distance between all possible combination of points between the two clusters and take the mean.

Distance Measure Methods









Divisive Clustering Algorithm

- Start with **1 Cluster** consisting of *N* records
- **Step 1**: Apply a flat clustering algorithm (e.g. K-means) to split the cluster into 2 clusters
- **Step 2**: Add the 2 clusters to the pool of clusters
- **Step 3**: Extract 1 cluster from the pool that consist of > 1 member (data observation)
- **Step 4**: Repeat steps 1-4 until the pool consist only of clusters with a single member (data observation)

Limitations of Hierarchical Clustering

- You need to identify the point where the algorithm starts to group disjoint cases, & then decide on the number of clusters to retain
- Computationally heavy with a few thousand cases
- Dendrogram can be too large to read for thousands of data observations
- A single pass through the data may yield poorer results
- Appears to be outdated nowadays

K-Means Clustering

K-Means clustering

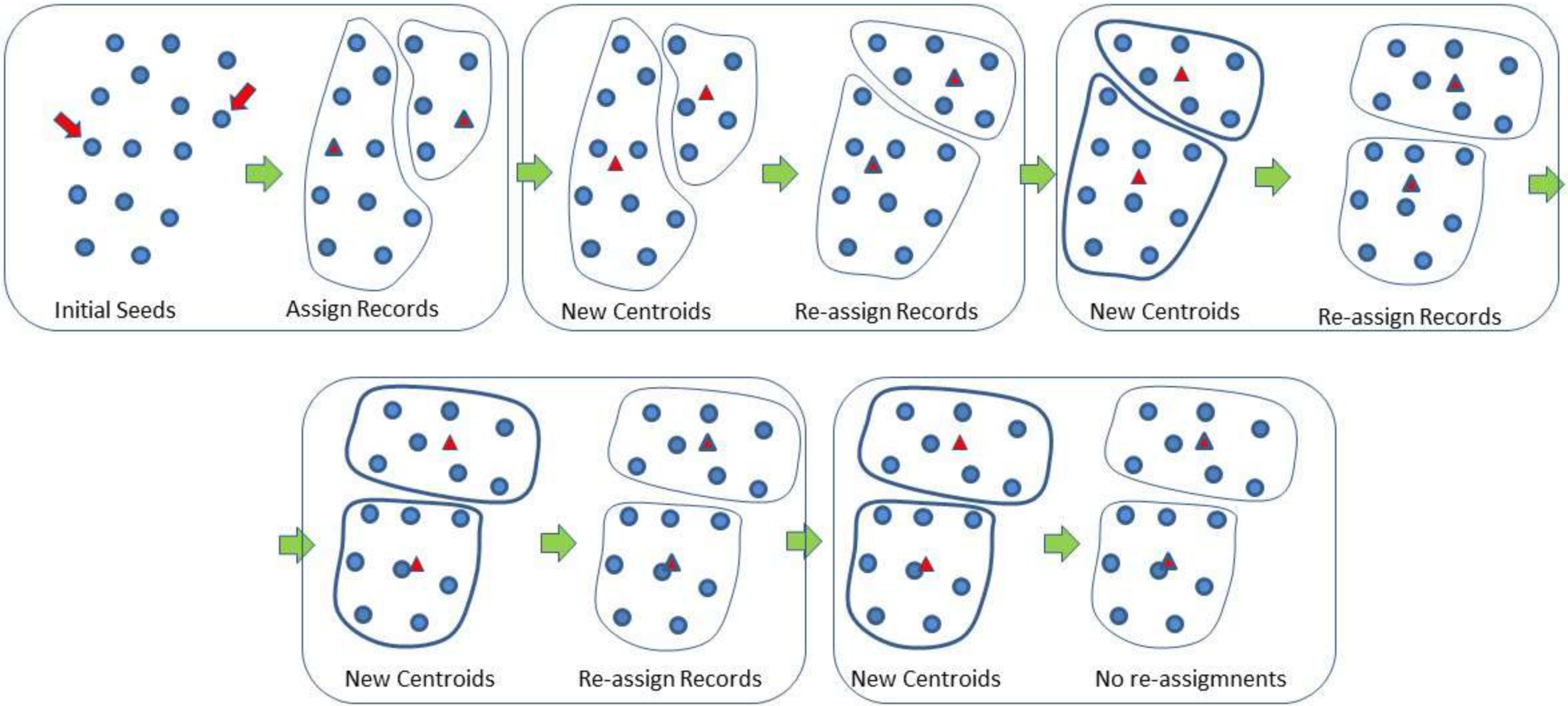
- K-means clustering is a type of top-down divisive clustering
- We start with a randomly **assigned K number of centroids**
- Data observations are then clustered based on its proximity to a nearest cluster centroid. Cluster centroids are dynamically calculated as cluster members migrate during each iteration of the algorithm.
- This iterative procedure is applied **recursively until it converges** and there are no more cluster membership migration or changes.
- It has the advantage of being **more efficient** since it does not generate a complete hierarchy all the way down to individual objects
- k-means is perhaps the **most widely used** clustering algorithm due to its simplicity and efficiency

https://nlp.stanford.edu/IR-book/

K-Means Clustering (cont'd)

- K-means requires the user to specify in advance the number of clusters to be formed (k clusters).
- If you have no intuition about the number clusters, then K-means requires a lot of **trial and error**, multiple runs, and evaluation of different solutions
- You can use Hierarchical clustering to get an indication of the number of clusters.
- It is distance based and unlike the hierarchical algorithms, it does not need to calculate the distance between all pairs of records
- K-means is computationally efficient and fast clustering algorithm that can handle both long (many records) and wide datasets (many input fields)

K-Means clustering method



K-means clustering algorithm

Specify *K* number of clusters

Step 1: Arbitrarily designate *k* data point as **seed points (initial centroids)**

Step 2: **Calculate the distances** between every data point and the centroid.

Step 3: **Re-assign** data points to the **nearest centroid** to form new clusters

Step 4: For each cluster that has lost or gained a data point, **re-compute** the new cluster centroids

Step 5: Repeat Step 2-4, until (a) no more re-assignment is possible or (b) the newly computed centroids do not deviate significantly from the original or (c) some maximum number of iteration is reached

Centroid Computation

- A **Centroid** represents the "centre" of all the data points:
  - A representative data point that is in the centre
  - A computed point such as the arithmetic mean
- The most common formula used is the Euclidean distance
- Other methods have been used include – median (i.e. K-Medoids),

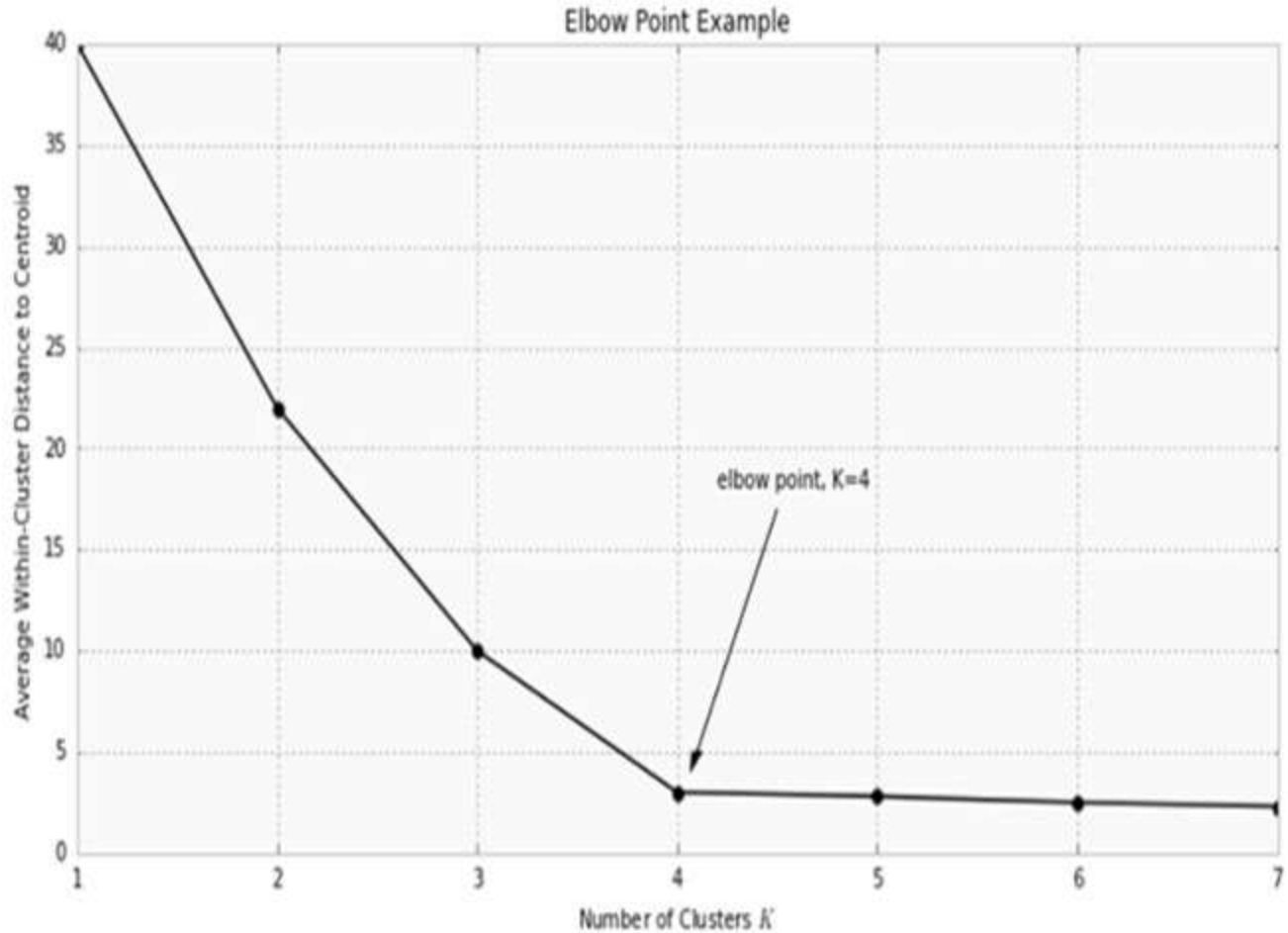
Deciding on the value of K

- The K-Means algorithm is somewhat **naïve** because it clusters the data into *k* clusters, even if *k* is not the right number of clusters to use.
- In general, there is no accurate method to find the *exact* value of *K*, but a **good estimate** can be obtained using these methods.
  - Using domain knowledge to ascertain the right number of cluster by running *K*-means several times and compare the results
  - Use the Sum of Squares "Elbow Method" (see next slide)
  - Others: cross-validation, information criteria, the information theoretic jump method, the silhouette method, and the G-means algorithm.

Some of these methods will be discussed later

The Elbow Method

- The Elbow method:
  1. Run k-means for a range of values e.g. $k = 2 \ldots 10$
  2. calculate the **sum of squared errors** (SSE) for each value of k
  3. Plot a line chart for SSE-vs-K
  4. Look for an **elbow** - that value of k is the best
- Note that the SSE will decrease towards 0 as we increase K.
- Our goal is to choose a small value of k just before (elbow) it starts to show diminishing returns.



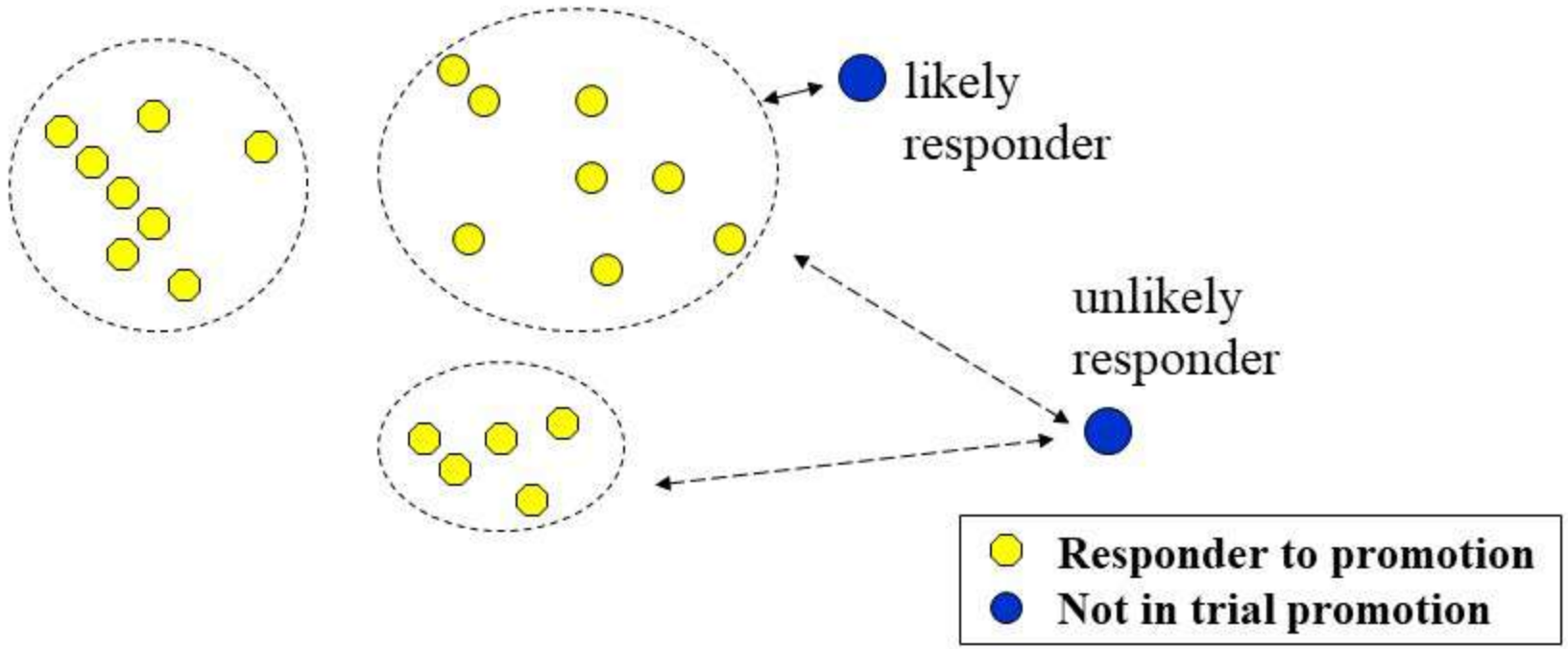
Alternative is to use the silhouette instead of the SSE

Some observations about K-Means

- The K-Means algorithm is guaranteed to converge to a result. However, the result may be a **local optimum** (i.e. not necessarily the best possible).
- Since the K-means algorithm is trying to minimize the sum of squared differences *within* each cluster, the centroids naturally **gravitate towards larger, denser clusters**.
- Clustering is a **heuristic method** and therefore much experimentation is needed to achieve the desired outcome.
- K-Means can detect outliers and put them into a cluster.
- You can also choose to do an **anomaly analysis** before actual clustering.
- Pro: Fast, easy, General clustering for all kinds of business problems.
- Con: Sensitive to outlier/noise

Utilising Cluster Results

- Analyse clusters for knowledge discovery
- Use of clusters as predictive (or other) models
  - E.g. to generate a mailing list given a list of responders to a previous mailing campaign



Density-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise (**DBSCAN**)

Density-Based vs Centroid Based

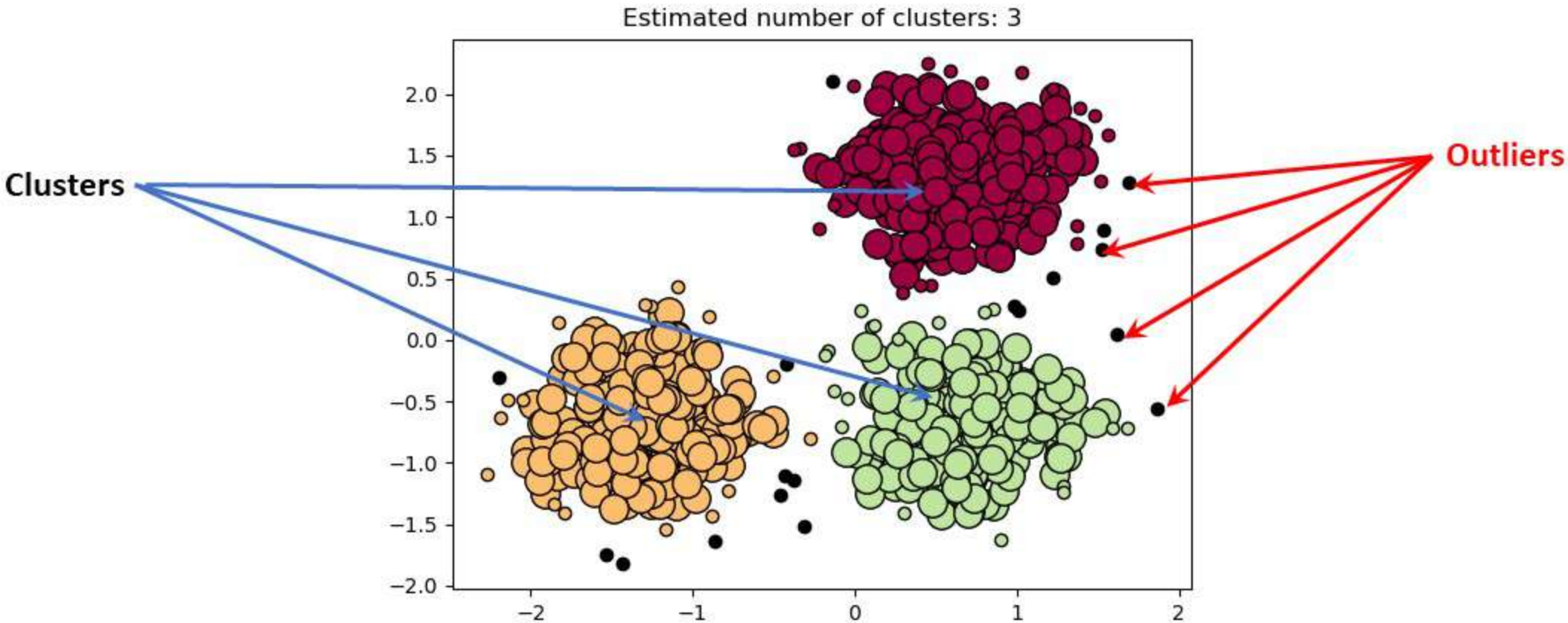
- K-Means is a **Centroid-based clustering** algorithm because data are assigned to the nearest centroid to form new clusters.
- One of the key strengths of K-Means is that it is easy to understand and efficiently used in practice.
- However, K-Means is highly sensitive to outliers, and can sometimes cause undesirable effects.
- It is possible that outliers are assigned to a nearest cluster where it is not supposed to.
- Outliers can also impact the centroid as it gets pulled towards these extreme values

The DBSCAN Method

- In DBSCAN, there are no centroids, and clusters are formed by **linking nearby points** to one another.
- There is the concept of **noise points** in the DBSCAN algorithm - points that do not belong to any clusters. These points are eliminated by the algorithm.
- In DBSCAN, there is no need to specify the number of clusters, but does require specifying two parameters that influence the decision of whether two nearby points should be linked into the same cluster.
  - A **distance threshold**, Eps (ε-epsilon), and
  - A **minimum number of points** ("MinPts")
- In DBSCAN, only a **single pass** through the data is needed, and once a point has been assigned to a particular cluster, it never changes.

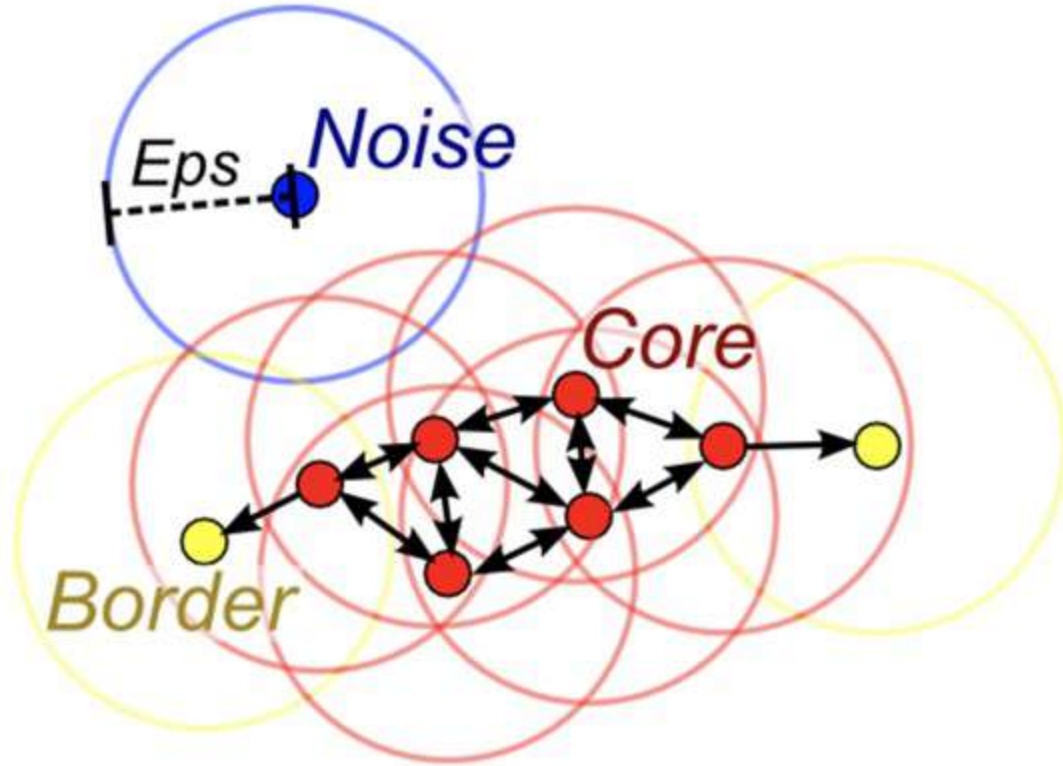
What is meant by Density-based?

- DBSCAN is a simple, but effective algorithm for finding **dense regions** of objects that are surrounded by low-density regions.

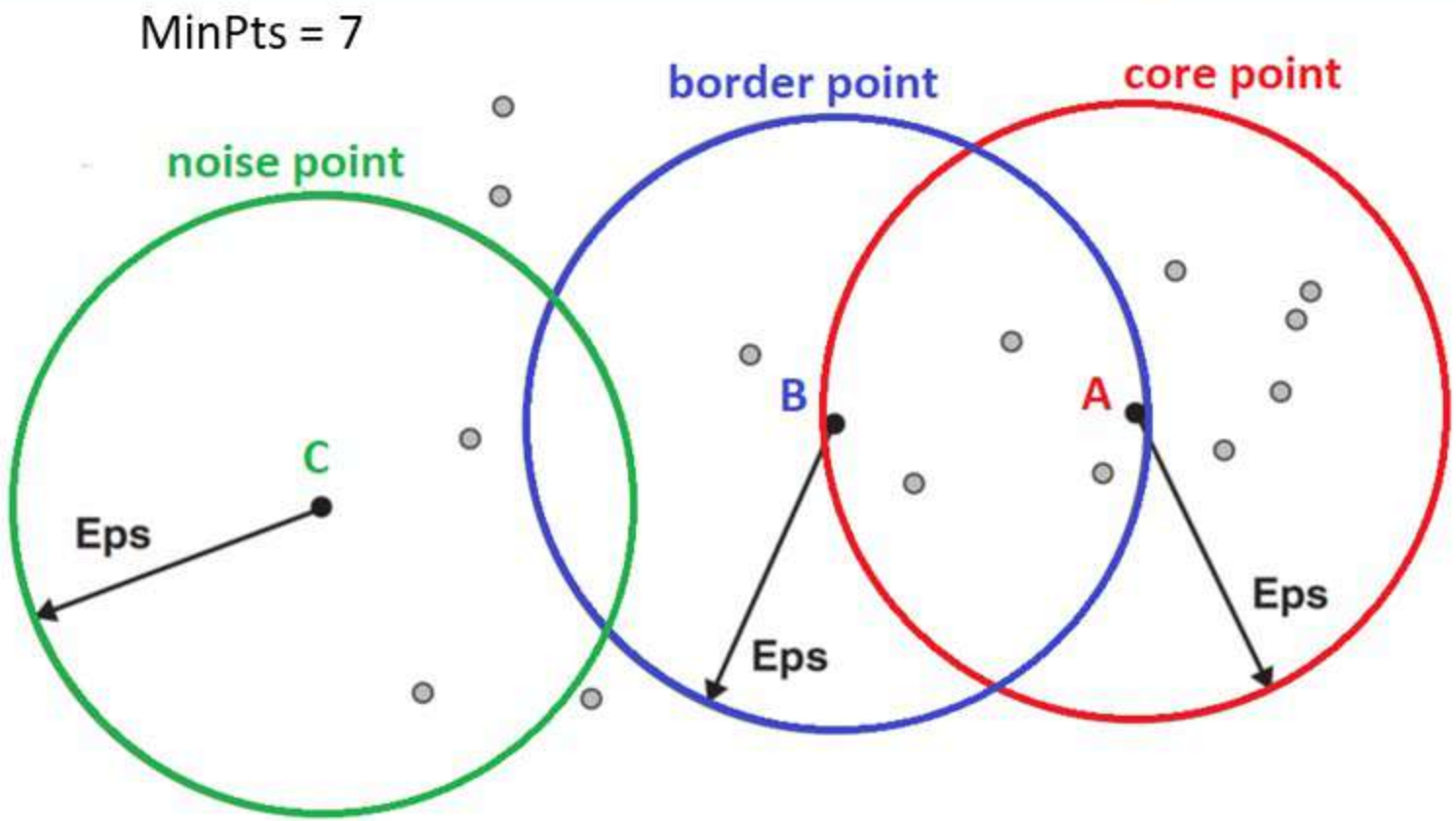


Basics of Density-based clustering

- **Density** at point *p* = number of points within a specified radius (*Eps*)
- A point is a **core point** if it has ≥ a specified number of points (*MinPts*) within *Eps*
  - These are points that are at the interior of a cluster
- A **border point** has fewer than *MinPts* within its *Eps* but is in the neighborhood of a core point
- A **noise point** is any point that is not a core point or a border point



Core, Border and Noise points



A is core because it has MinPts within its Eps neighborhood

B is a border point because it is within the Eps neighborhood of A but doesn't satisfy MinPts

C is a noise point because it is NOT within an Eps neighborhood and have no MinPts

DBSCAN Algorithm

Start with *N* data observations

Step 1: Randomly select a point, *P*

Step 2: Retrieve all points that satisfy the Eps and MinPts criteria w.r.t. to *P*

Step 3: If *P* is a core point, a cluster is formed

Step 4: Check if P is a border point, attach to core point

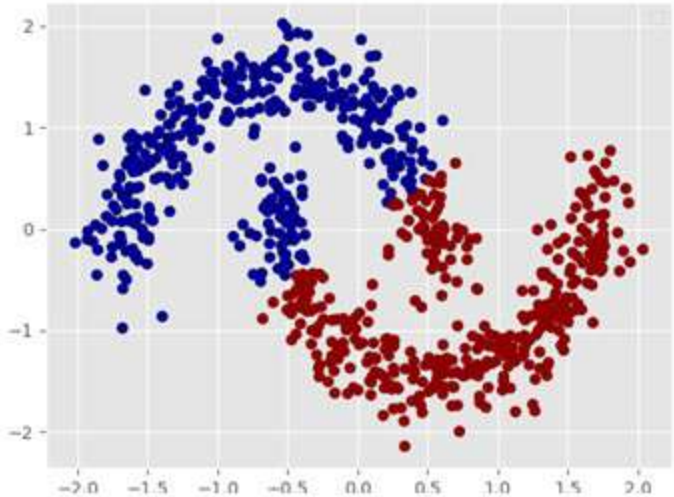
Step 5: Make each group of connected core points into a separate cluster.

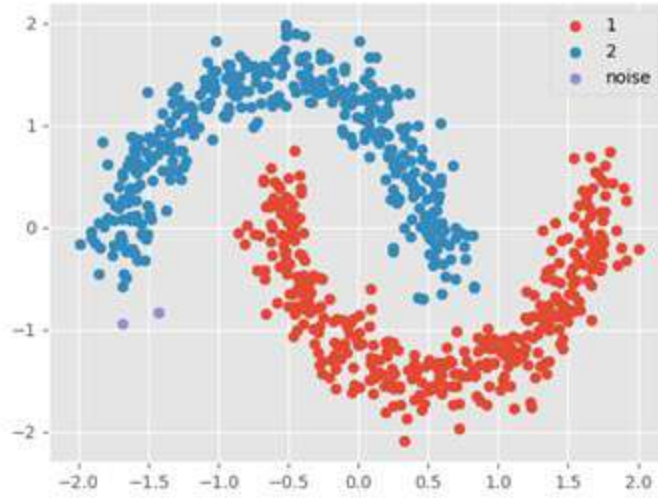
Step 6: Assign each border point to one of the clusters of its associated core points

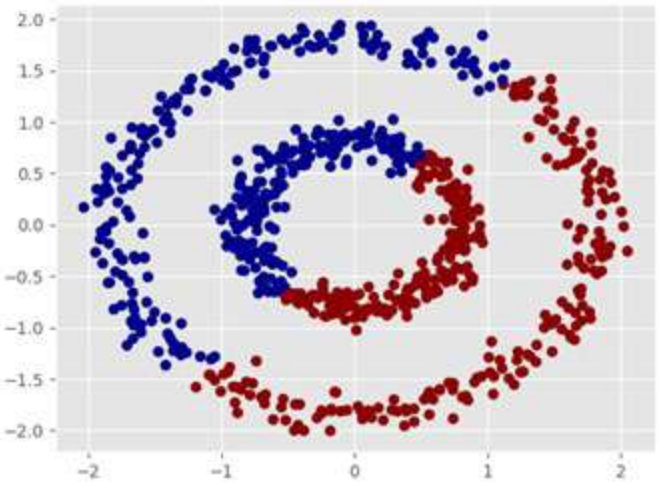
Some observations about DBSCAN

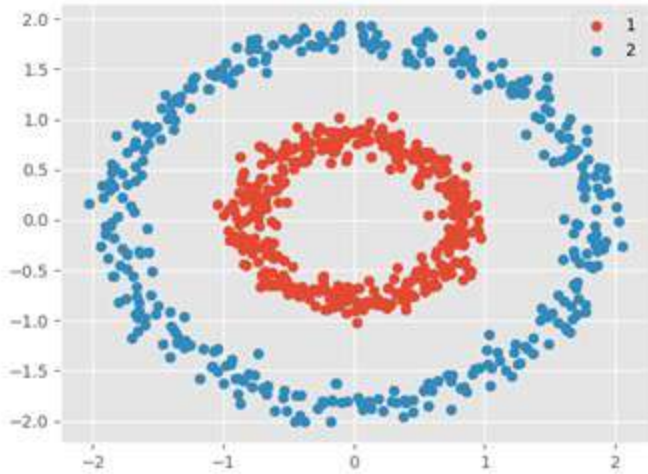
- Does not require you to specify the number of clusters
- Can find **arbitrarily shaped cluster** even if the cluster is completely surrounded by a different cluster
- Able to **detect noise** and is robust to outliers
- Pro: Arbitrary shaped data, robust to noise
- Con: require sufficient high-density areas, not popular in business

Kmeans vs DBSCAN









Fuzzy C-Means Clustering

Fuzzy Clustering

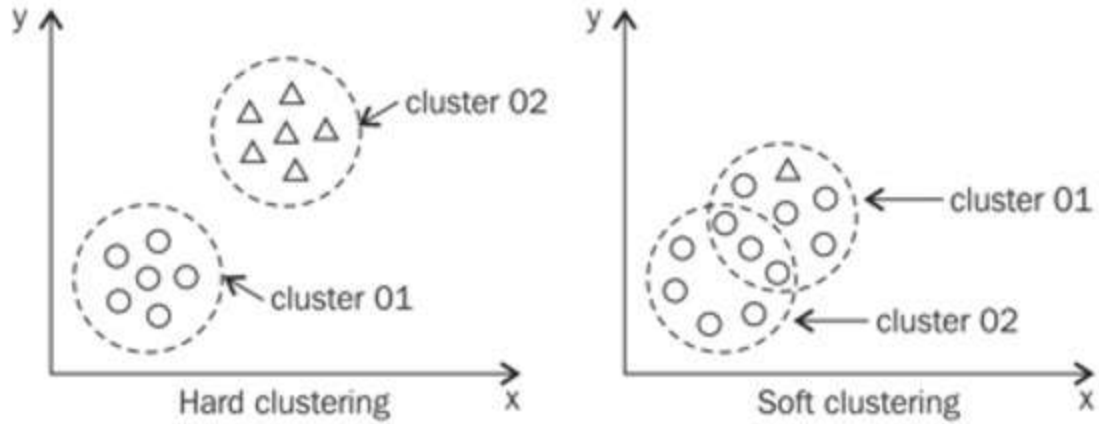
- Uses concepts from the field of **Fuzzy logic** and **Fuzzy Set Theory**
- Fuzzy clustering is a clustering method where each data observation has a probability of **belonging to every cluster**. In other words, each data observation has a set of membership coefficients corresponding to the degree of being in a given cluster.

```
Observation#n [cluster1=0.65, cluster2=0.05, cluster3=0.3]
```

- This type of clustering is known as **soft clustering** as opposed to hard-clustering such as K-means or DBSCAN.
- For instance, in k-means clustering, each object is assigned to exactly to one cluster. This is usually referred to as **hard clustering**.

Assigning observations to fuzzy clusters

- Data point that are closer to a particular cluster centroid are assigned a higher degree of membership than data points that are further away. The degree of membership is a numerical value varying from 0 to 1.



- The centroid of a cluster is calculated as the mean of all points, weighted by their degree of belonging to the cluster

Fuzzy C-means Algorithm

- **Step1**: Define number of cluster
- **Step2:** Randomly initialise the membership matrix
- **Step3:** Calculate the centroids
- **Step4**: Recalculate and update the membership matrix
- **Step5**: If the difference of centroid matrix between new and previous is less than ε the stop, otherwise go back to Step 3.

Evaluating Clustering Performance

Evaluating the Validity of Clustering Results

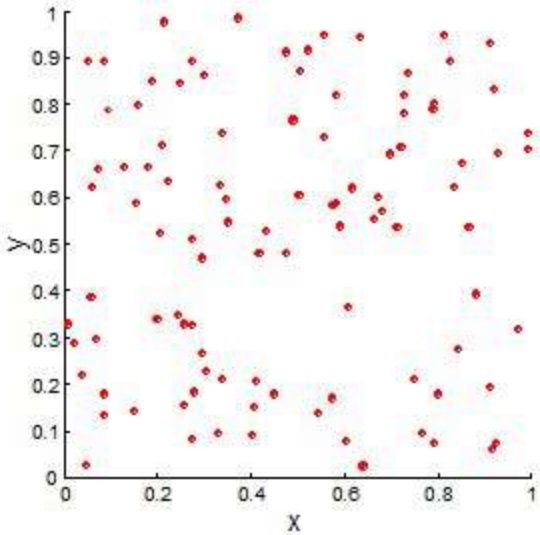
- For supervised classification we have a variety of measures to evaluate how good our model is Accuracy, Precision, Recall, Sensitivity, etc.
- For cluster analysis, how do we evaluate the "goodness" of the resulting clusters when it is not even a supervised algorithm?
- Moreover, the practical truth is that "clusters are in the eye of the beholder"!
- Clusters are meant to uncover patterns and provide insights for decision making, and therefore clusters must be valid in the eyes of the stakeholders – even when cluster statistics does not provide "strong" support

Interpreting the resulting Clusters

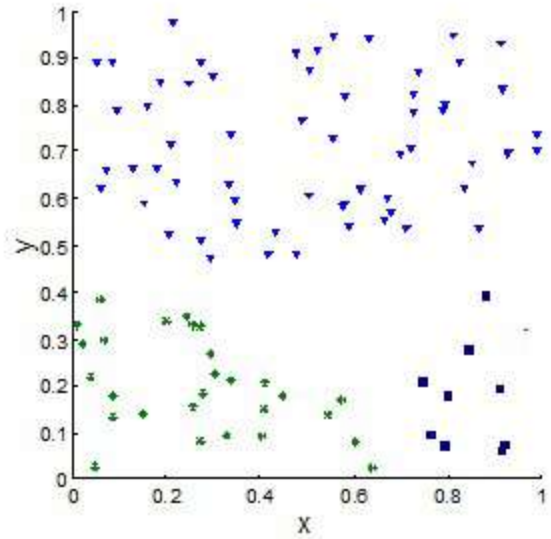
- Clusters are developed using mathematical algorithms without the contextual information.
- In general, you want to give meaningful labels to each cluster to highlight their distinct characteristics.
- Some points to consider:
  - Can the clustering be **explained** in practical terms?
  - Examine the value of each variable of the cluster **centroid**, (the cluster profile) of each cluster.
  - Look at the **distinguishing characteristics** of each cluster's profile and identify substantial differences between clusters
  - Cluster solutions failing to show substantial variation between clusters indicate that this may be a spurious clustering

Clusters found in random data

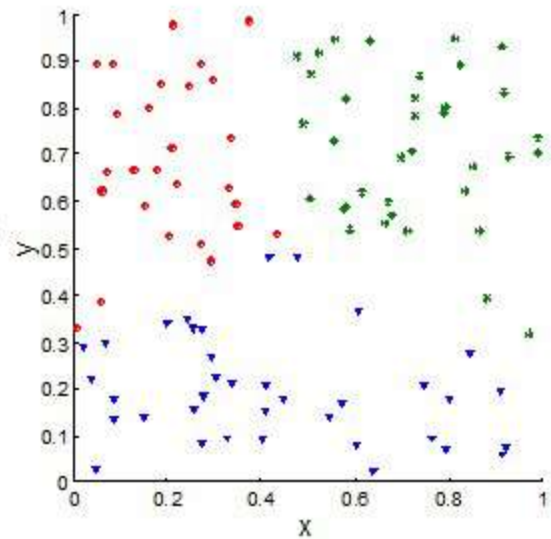
Random Points



DBSCAN

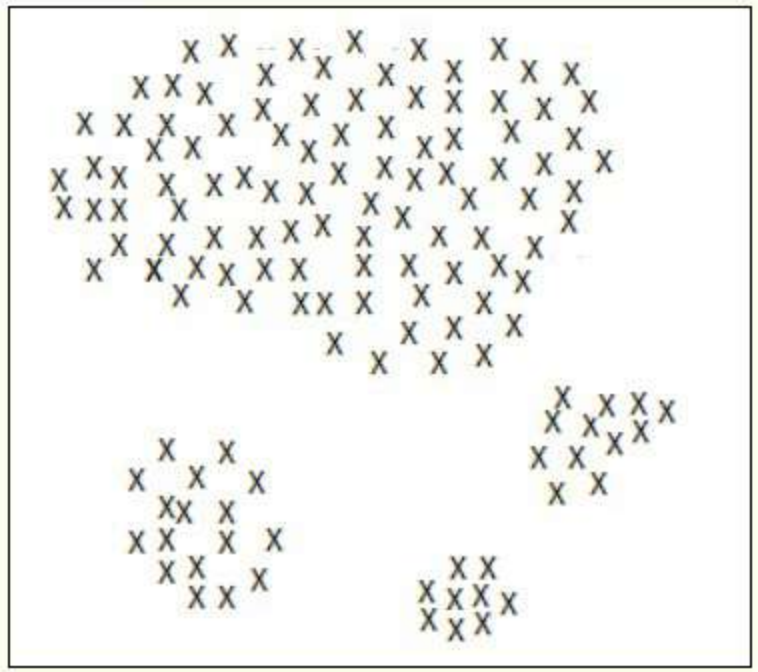


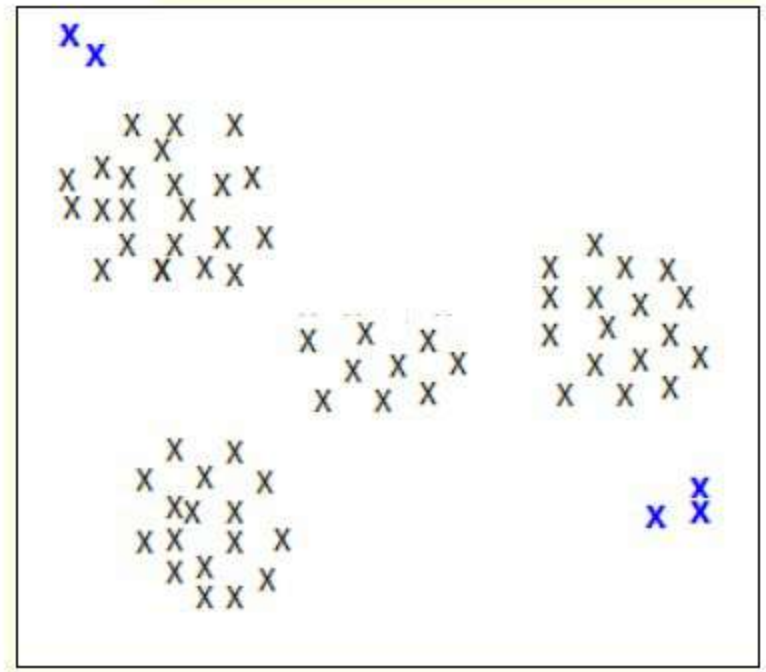
K-means



How to Evaluate the Cluster Solutions ?

- Assess the **number** of clusters and their relative **sizes**
  - If a single cluster contains majority of the data (i.e. dominates), it could imply the need for further clustering. Bear in mind that it could also reflect the true profile of your customers
  - If there are very small clusters, it could mean that they represent outliers. This should warrant further investigation.





How to Evaluate the Cluster Solutions ?

- Assess the cluster **cohesion** and **separation**
- Cohesion can be calculated using **Sum of Square Error** (SSE)

$$SSE = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

- Or **pooled Standard deviation**

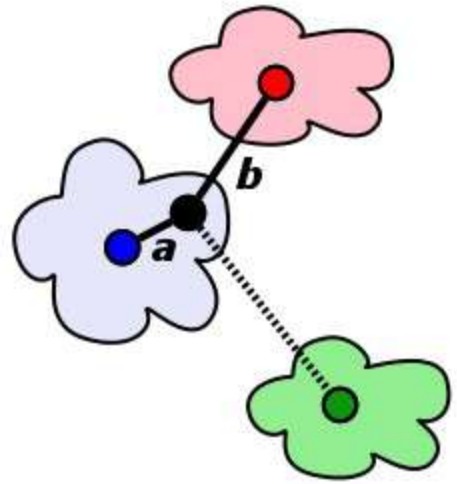
$$s_{pooled} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2 + \ldots + (n_k - 1)s_k^2}{n_1 + n_2 + \ldots + n_k - k}}$$

- Many tools combine the cohesion and separation measures into one statistic called the cluster **silhouette** score.
- The cluster silhouette is a combined measure of **internal cohesion** and **external separation** to gauge the quality of the cluster solution.

Cluster Silhouettes

- The silhouette coefficient for data, i

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



a(i) = mean intra-cluster distance (cohesion)

b(i) = mean nearest cluster distance (separation)

- The overall silhouette coefficient is the average of the data silhouettes

$$S = \frac{1}{n} \sum_{i=1}^{n} s(i)$$

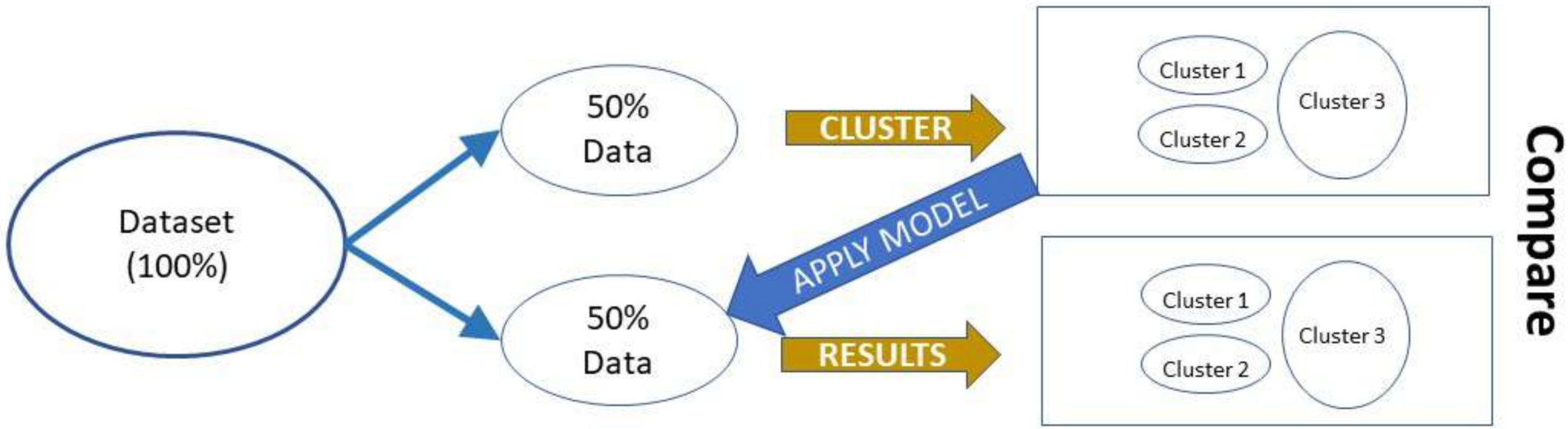
- The value range from -1 (extremely bad) to +1 (perfect clustering). Generally, we accept solutions with values >= 0.4

Validation of Cluster Analysis

- Model validation is particularly important for cluster analysis because it is an unsupervised learning technique.
- **Cross-validation** is a popularly used method
  - Randomly split the original data into **two groups**
  - Perform cluster analysis on each group
  - **Compare** the two cluster solutions by examining the similarities in
    - The number of clusters
    - The cluster profiles.

Validation of Cluster Analysis

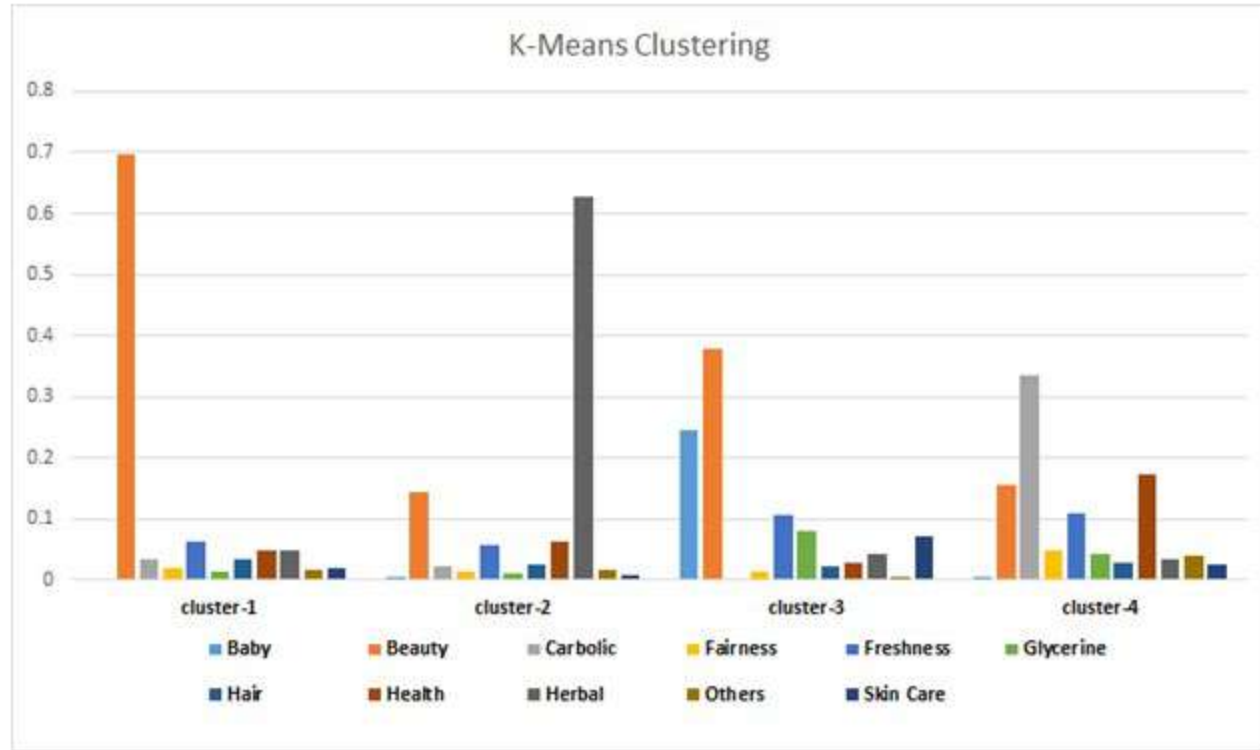
- Second method is to split randomly the clustering data set into two parts
- Development and Validation (50:50)
- Perform cluster analysis on development sample **only**
  - Create cluster profiles
  - Validate the results on the validation sample using development sample clustering algorithm (which becomes the implementation code)



Develop Profiles from each cluster

- Profiling of Clusters should be done with those variables where their pattern or distribution is different within each cluster
- e.g. Cluster 1 suggest “Beauty Conscious”; Cluster 2 : “Herbalist”; Cluster 3 : “Young Parents”; Cluster 4: “Healthnuts”

Product	cluster-1	cluster-2	cluster-3	cluster-4
Baby	0.003	0.006	0.246	0.006
Beauty	0.697	0.144	0.38	0.157
Carbolic	0.034	0.023	0	0.335
Fairness	0.021	0.014	0.013	0.048
Freshness	0.064	0.057	0.106	0.111
Glycerine	0.014	0.012	0.081	0.044
Hair	0.033	0.027	0.022	0.028
Health	0.048	0.064	0.029	0.173
Herbal	0.05	0.626	0.044	0.033
Others	0.017	0.018	0.006	0.041
Skin Care	0.019	0.008	0.072	0.025

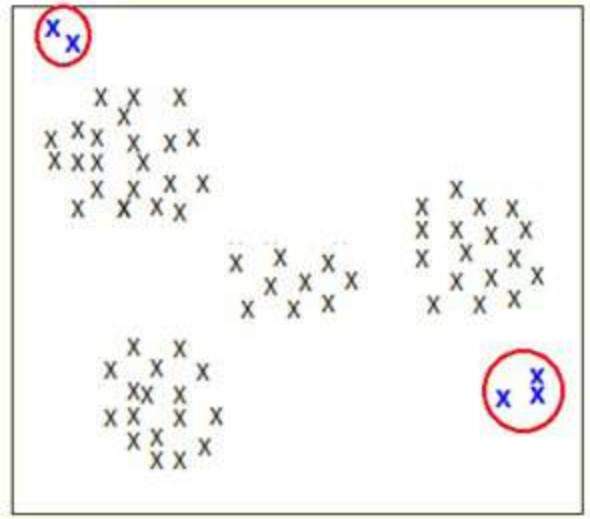


Making sense of Clusters

- Determine if different clusters exist and if there is a **"natural" grouping** for members within the data set.
- Identify those variables (or ranges of variables) that most **strongly define** the clusters.
- Determine the rationale underlying the clusters. The results of the clusters should make sense from a **business point** of view & should **generate business opportunities**.
- Identify anomalies or outliers; i.e. members who do not fit into any cluster
- Suggests a classifying scheme for new data members

What do we do with Outliers ?

- Outliers are data points that do not naturally fit into any cluster because of their unusually high or low value
- They distort your clustering by influencing your similarity measures
- It may be necessary to remove them to improve the clustering results
- But it may be good to investigate where the outlier came from
  - Was it faulty data collection or transcription?
  - Or was the outlier a representative of some different, new population?



Data Preparation

- Variable selection
  - Clustering with too many variables produces clusters that are hard to interpret
  - Use domain knowledge to guide the selection of “relevant” variables
- Adequate Sample Size
  - Sufficient size is needed to ensure representativeness of the population and its underlying structure, particularly small groups within the population.
- Standardization
  - Sometimes you can achieve better results from clustering with raw data
  - You will need to experiment a little
- Remove outliers
  - If it makes sense, remove observations that are known to be outliers
- Variable Type
  - Clustering works best with continuous or binary variable
  - Re-code categorical variable
- Check for Multi-collinearity
  - Not all collinearity is bad for clustering – but if too many variables are correlated it may cause redundancies.

Applications of Cluster Analysis

- Cluster Analysis is versatile and can be used in many business problems across many domains:
- Sales & Marketing: help marketers discover groups in their customer databases, and then use this insight to develop more targeted marketing campaigns
- Fraud Detection: Identify groups of customers whose transaction behavior is uncharacteristic
- Health & Bioinformatics: help physicians discover groups of patients with similar profiles and with a similar risk pattern, and use this insight to make predictions about diseases risks
- Insurance: Help identify groups of policy holders with high average claim cost

Demo, Workshop, Exercises

Thank You!