Cluster Analysis

Hardik Gupta

Machine Learning

Supervised Learning

(Predictive Analytics)

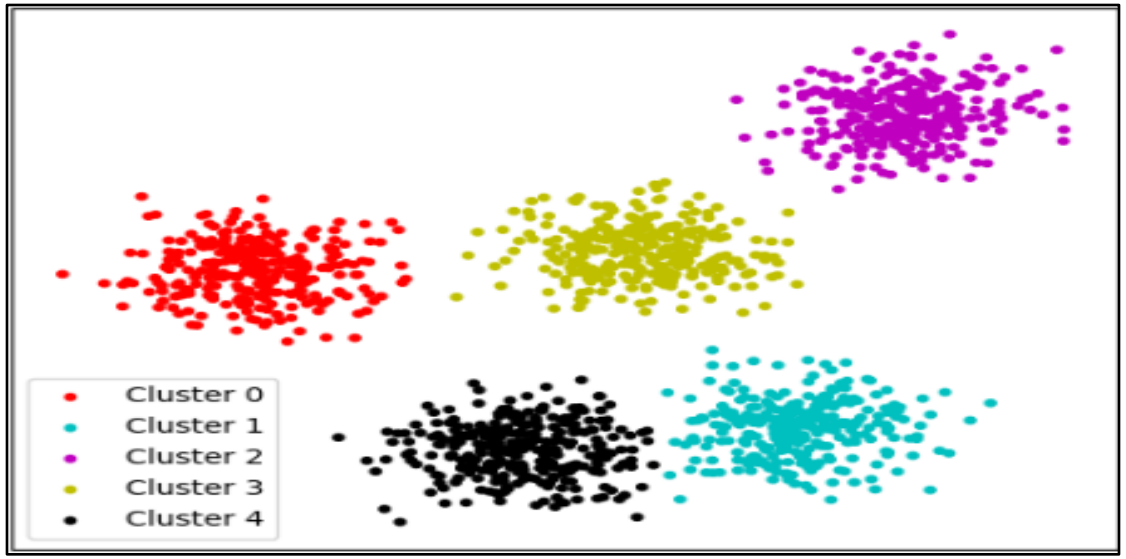
- Prediction (numerical Y)
- Classification (categorical Y)

Unsupervised Learning

- Segmentation/Clustering
- Dimension Reduction
- Relationship Mining
- Recommender System
- Network Analysis

Clustering

- Cluster Analysis (data segmentation) is an exploratory method for identifying homogenous groups (clusters) of records
- Similar records should belong to the same cluster
- Dissimilar records should belong to different clusters



Clustering

- Discover the underlying structure of the data
- What sub-population exist in the data
  - How many are there
  - What are their sizes
  - do elements in a sub-population have any common properties
  - Are sub-populations cohesive? Can they be further split up
  - Are there outliers

Motivating Example – Pizza Coupon

- Send targeted coupons to customers
- Different strategy for different segments of customer
  - Large Families
  - Small Families
  - Singles
  - College
- Segments created based on size of order, price and frequency

More Motivating Example

- **Industry Analysis:**

For a given industry, cluster firms based on growth rate, profitability, market size, product range, presence in various international markets, to understand industry structure (determine competitors)

- **Airlines – Frequent Flyers Program**

Identify clusters of similar passengers for the purpose of targeting different segments for different types of mileage offers. Passenger data include information on mileage history and on different ways they accrued or spent miles.

Types of Clustering Methods

- **Goal**
  - Monothetic: cluster members have some common property
    - Interpretable, related to decision trees/rule based clusters (temp > 100, visit doctor)
  - Polythetic: cluster members are similar to each other
    - Distance between members define membership
- **Overlap**
  - Hard clustering: clusters do not overlap
    - Elements either belongs to a cluster or not
  - Soft clustering: clusters may overlap
    - “strength of association” between element and cluster (e.g. probabilities)
- **Flat or Hierarchical**
  - Set of groups vs taxonomy

Agenda

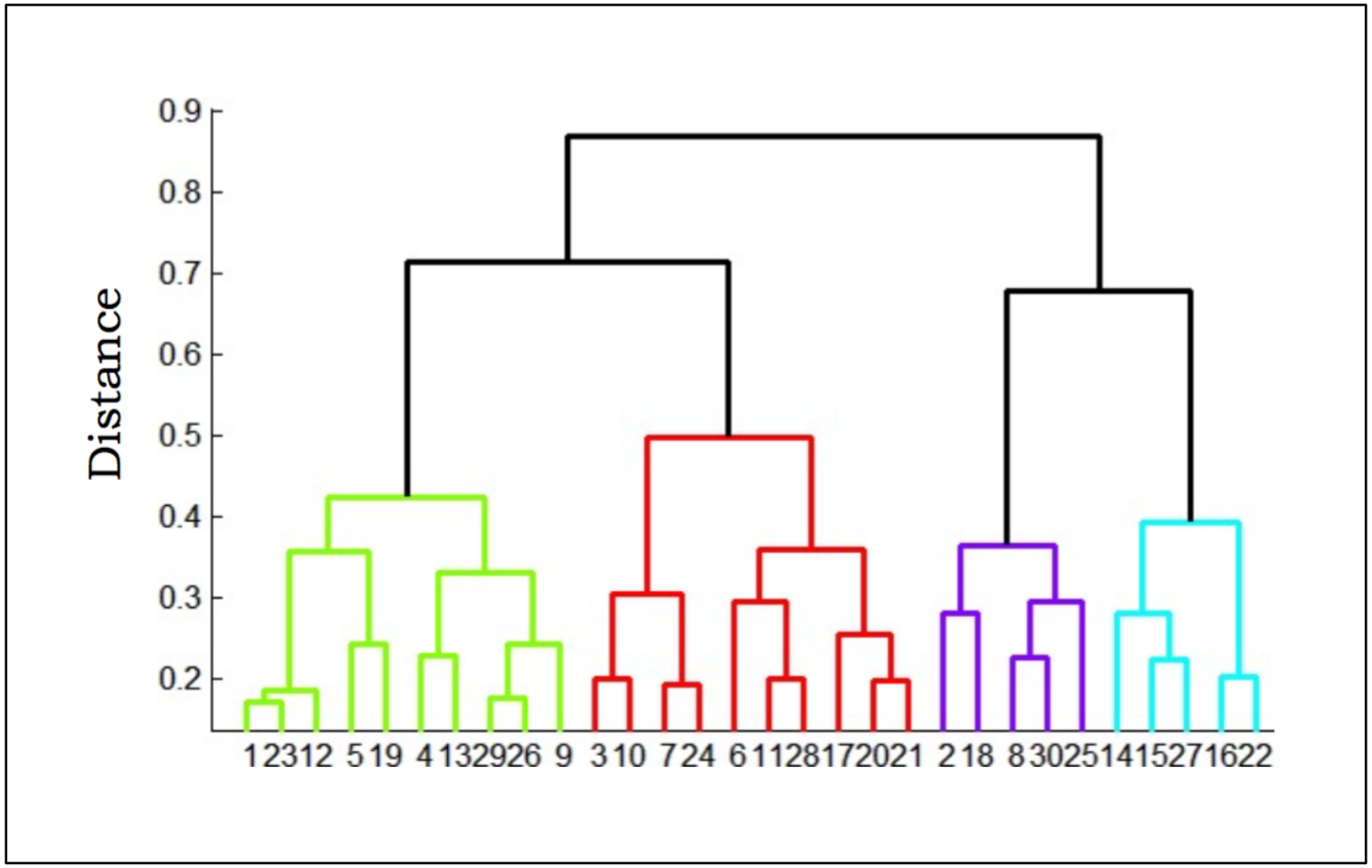
Hierarchical methods - Agglomerative:

- Sequentially merge group of records until all are put in one large group
  - Useful when goal is to arrange the clusters into a natural hierarchy
  - Requires specifying distance measure to find similarity.

Non-Hierarchical methods - K Means

  - Pre-specified number of clusters, assign records to each of the clusters in order to improve homogeneity within group.

Hierarchical Clustering



Hierarchical Clustering

- One of the most popular clustering technique for identifying groups in the dataset
- Not required to pre-specify the number of clusters to be generated as is required by the k-means approach.
- Hierarchical clustering has an added advantage over K-means clustering - results in an attractive tree-based representation of the observations called a dendrogram.

Hierarchical Clustering - Types

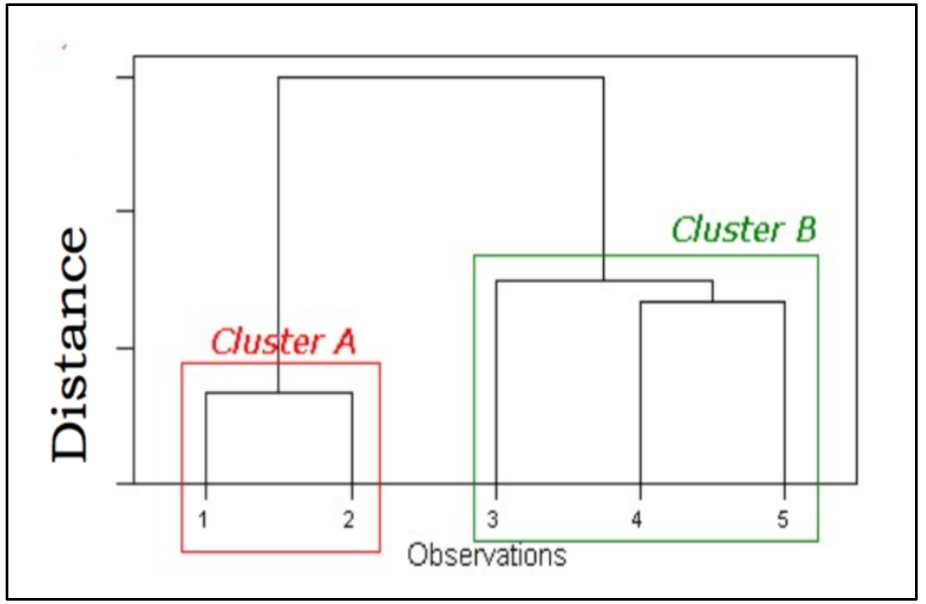
- Hierarchical clustering can be divided into two main types

- **Agglomerative clustering (Agglomerative Nesting)**
  - Works in a bottom-up manner.
  - Good at identifying small clusters

- **Divisive clustering**
  - Works in a top-down manner.
  - The algorithm is an inverse order of Agglomerative
  - Good at identifying large clusters

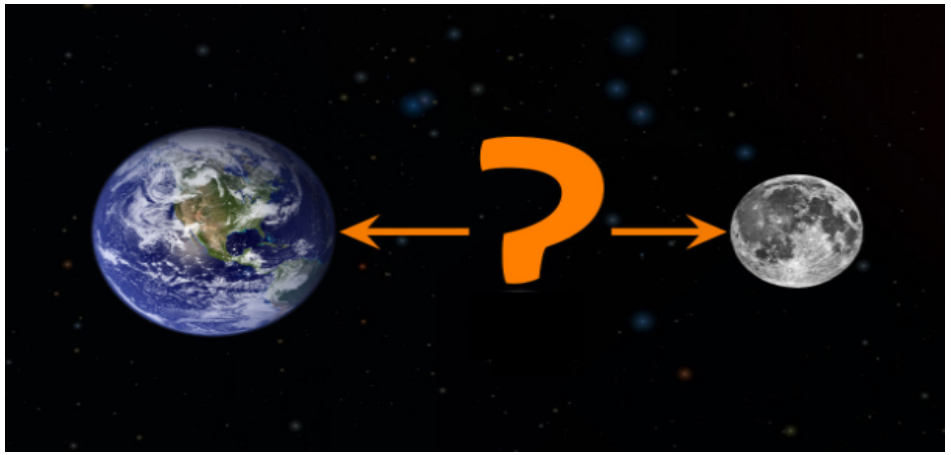
Agglomerative clustering

- Most popular hierarchical clustering technique
- Basic algorithm
  1. Compute the distance matrix between the input data points
  2. Let each data point be a cluster
  3. Repeat
     1. Merge the two closest clusters
     2. Update the distance matrix
  4. Until only a single cluster remains
- Dendrogram - tree like diagram that summarize the clustering process



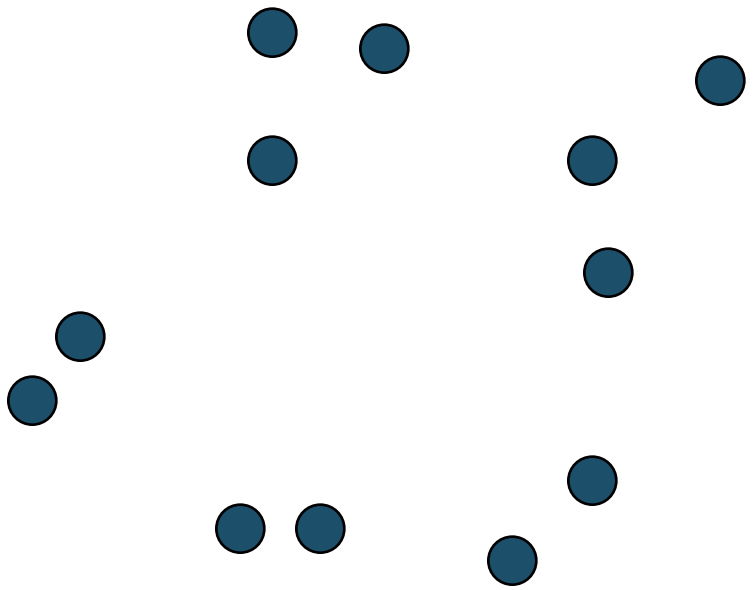
Similarity Measures

- Key operation is the computation of the distance between two records/clusters
  - Different definitions of the distance/similarity between records
  - Different definitions of the distance between clusters



Input/Initial setting

- Start with clusters of individual points and a distance/proximity matrix



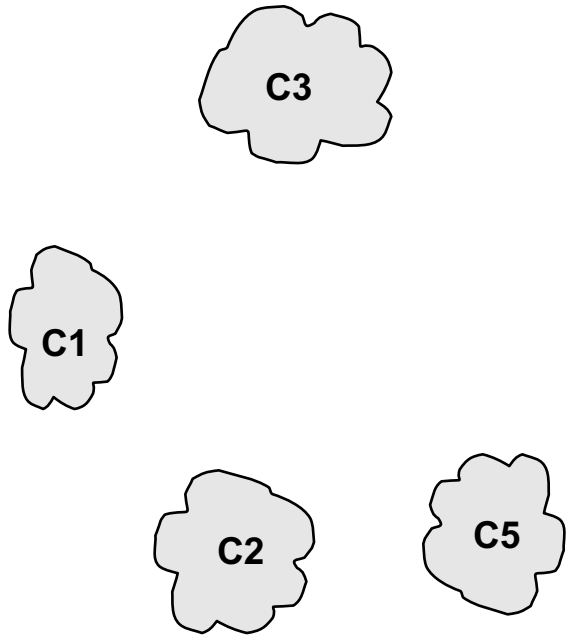
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						

Similarity metric e.g. Euclidean distance (L_2 norm)

Distance/Proximity Matrix

Intermediate State – Merging Records

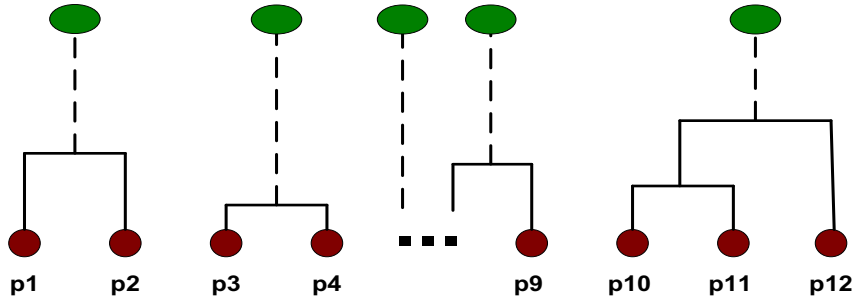
- After some merging steps, we have some clusters
- Cluster – collection of records



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

Distance based on type of Linkage e.g. single

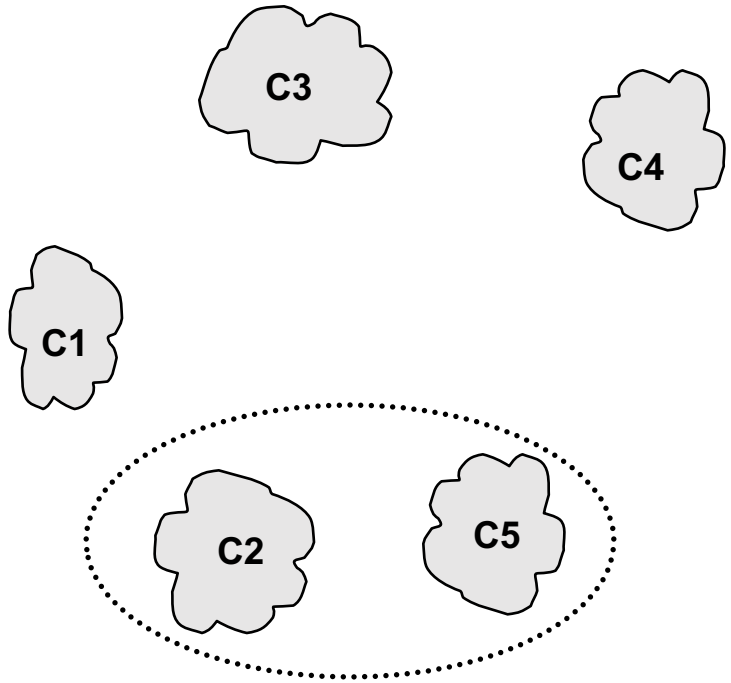


Intermediate State – Merging Records

- Merging of two individual records is achieved using similarity measure
- Famous similarity metrics
  - Euclidean
  - Normalized Euclidean
  - Mahalonobis
  - Manhattan
  - Cosine
- Let’s code and learn - Similarity_Distance.ipynb

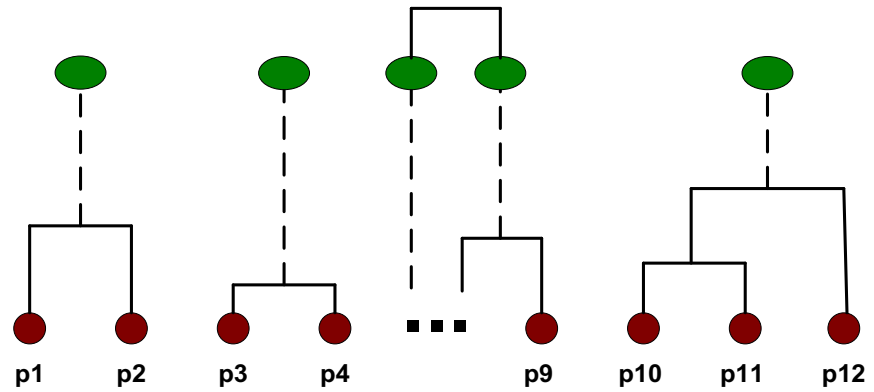
Intermediate State – Merging Cluster

- Merge the two closest clusters (C2 and C5) and update the distance matrix.



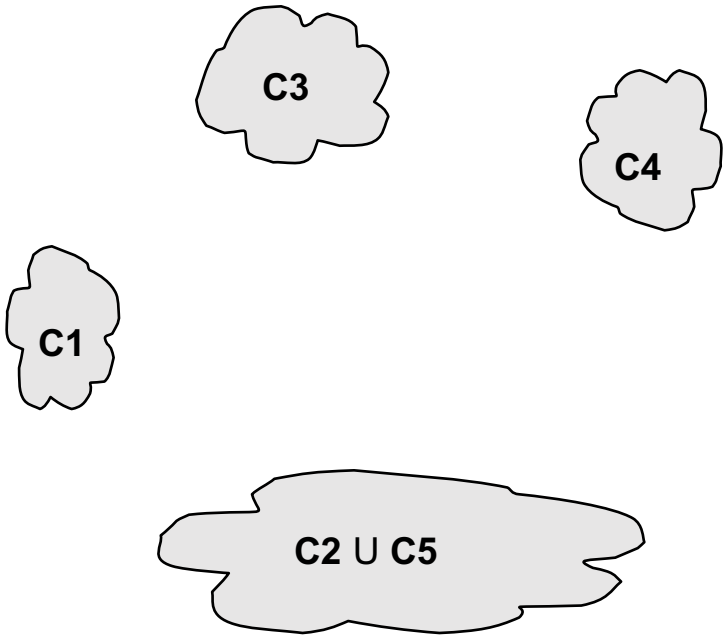
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

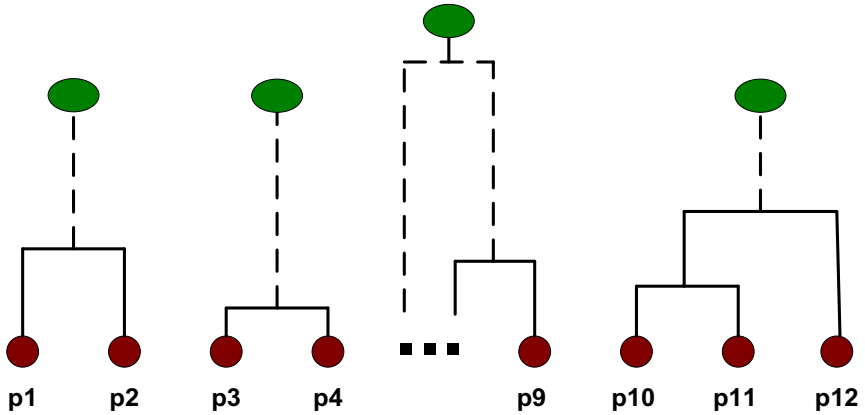


After Merging

- "How do we update the distance matrix?"



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		



Distance between two clusters

- Each cluster is a set of points
- How do we define distance between two sets of points
  - Linkage functions: takes two groups/clusters and returns a similarity score between them

Single Linkage (Nearest Neighbor)

- **Single-link distance** between clusters C_i and C_j is the ***minimum distance*** between any object in C_i and any object in C_j

$$D_{sl}(C_i, C_j) = \min_{x,y} \left\{ d(x, y) \middle| x \in C_i, y \in C_j \right\}$$

Complete Linkage (Farthest Neighbor)

- **Complete-link distance** between clusters C_i and C_j is the ***maximum distance*** between any object in C_i and any object in C_j

$$D_{cl}(C_i, C_j) = \max_{x,y} \left\{ d(x, y) \middle| x \in C_i, y \in C_j \right\}$$

Average Linkage

- **Average distance** between clusters C_i and C_j is the ***average distance*** between any object in C_i and any object in C_j

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

Centroid Linkage

- **Centroid distance** between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}(C_i, C_j) = d(r_i, r_j)$$

Ward's Distance

- **Ward's distance** between clusters C_i and C_j is the ***difference*** between the ***total within cluster sum of squares for the two clusters separately***, and the ***within cluster sum of squares resulting from merging the two clusters*** in cluster C_{ij}

$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

- r_i: centroid of C_i
- r_j: centroid of C_j
- r_{ij}: centroid of C_{ij}

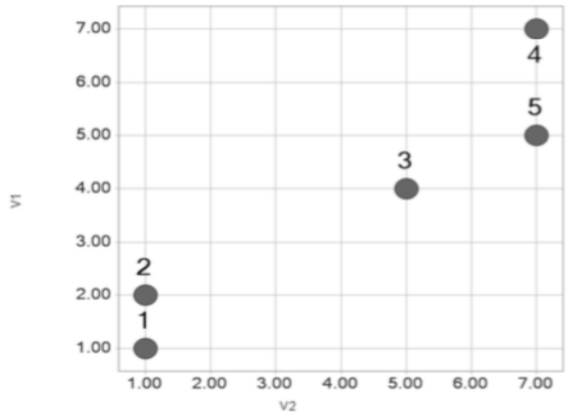
Pairwise distance between Clusters

- Single linkage (nearest neighbor): minimum distance between members of the two clusters
- Complete linkage (farthest neighbor): greatest distance between members of the two clusters
- Average linkage: average of all distances between members of the two clusters
- Centroid linkage: distance between their centroids (centers)
- Ward's method: merge clusters that provide smallest increase in the combined error sum of squares.

Agglomerative clustering Toy Example – Step by Step

Two variables, n=5 items:

item	v1	v2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7



Euclidean distance matrix

	1	2	3	4	5
1	0.0				
2	1.0	0.0			
3	5.0	4.5	0.0		
4	8.5	7.8	3.6	0.0	
5	7.2	6.7	2.2	2.0	0.0

Step 1

- Merge 1 & 2 into cluster A
- Use single linkage to compute distances from cluster A

	1	2	3	4	5
1	0.0				
2	1.0	0.0			
3	5.0	4.5	0.0		
4	8.5	7.8	3.6	0.0	
5	7.2	6.7	2.2	2.0	0.0

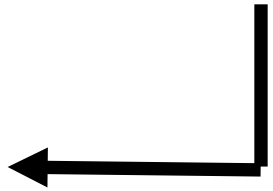
	A	3	4	5
A	0.0			
3	4.5	0.0		
4	7.8	3.6	0.0	
5	6.7	2.2	2.0	0.0

Step 2 & 3

- Merge 4 & 5 – cluster B

	A	3	4	5
A	0.0			
3	4.5	0.0		
4	7.8	3.6	0.0	
5	6.7	2.2	2.0	0.0

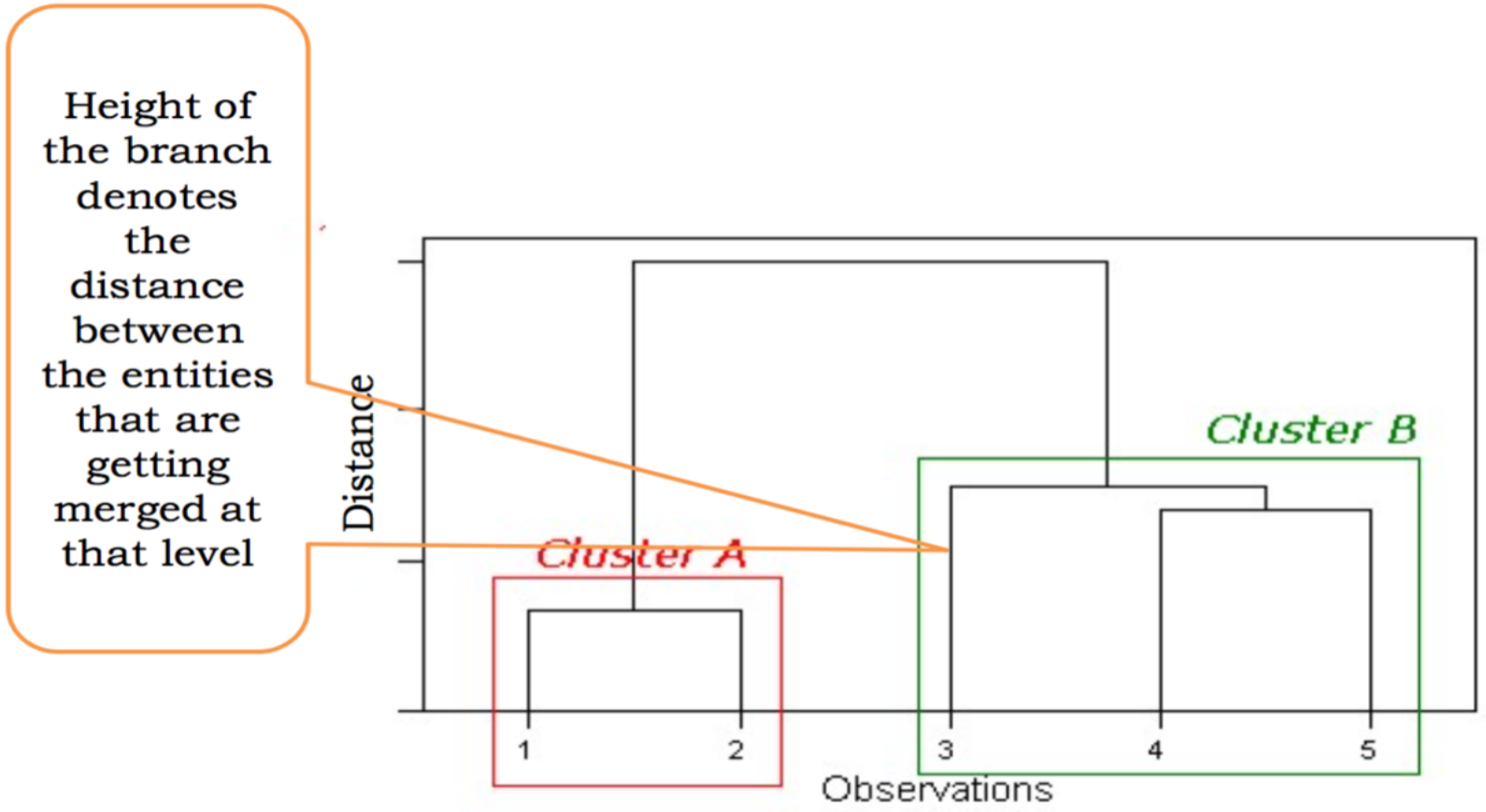
	A	3	B
A	0.0		
3	4.5	0.0	
B	6.7	2.2	0.0



Merge 3 & B

	A	B
A	0.0	
B	4.5	0.0

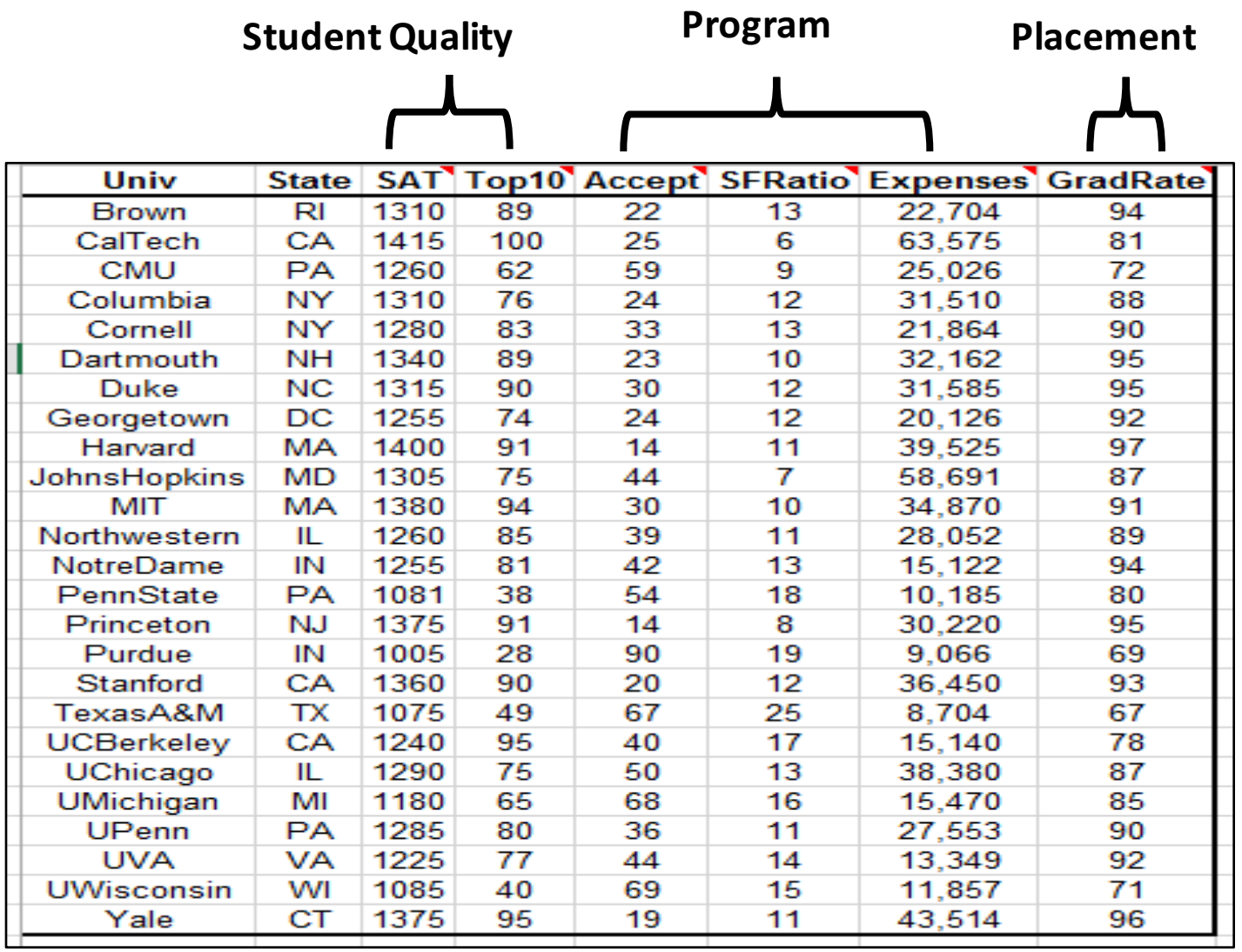
Hierarchical Clustering : Dendrogram



Case study - UG Business Programs

- Dataset - Universities_Clustering.csv
- Data for 25 undergraduate programs at business schools in US universities in 1995
- Data points:
  - Univ – University name
  - State – State
  - SAT – SAT score
  - Top10 - % new freshmen in top 10% of highschool class
  - Accept - % of applicants accepted
  - SFRatio – student to faculty ratio
  - Expenses – Estimated annual expenses
  - GradRate – Graduation rate %

Univ	State	SAT	Top10	Accept	SFRatio	Expenses	GradRate
Brown	RI	1310	89	22	13	22,704	94
CalTech	CA	1415	100	25	6	63,575	81
CMU	PA	1260	62	59	9	25,026	72
Columbia	NY	1310	76	24	12	31,510	88
Cornell	NY	1280	83	33	13	21,864	90
Dartmouth	NH	1340	89	23	10	32,162	95
Duke	NC	1315	90	30	12	31,585	95
Georgetown	DC	1255	74	24	12	20,126	92
Harvard	MA	1400	91	14	11	39,525	97
JohnsHopkins	MD	1305	75	44	7	58,691	87
MIT	MA	1380	94	30	10	34,870	91
Northwestern	IL	1260	85	39	11	28,052	89
NotreDame	IN	1255	81	42	13	15,122	94
PennState	PA	1081	38	54	18	10,185	80
Princeton	NJ	1375	91	14	8	30,220	95
Purdue	IN	1005	28	90	19	9,066	69
Stanford	CA	1360	90	20	12	36,450	93
TexasA&M	TX	1075	49	67	25	8,704	67
UCBerkeley	CA	1240	95	40	17	15,140	78
UChicago	IL	1290	75	50	13	38,380	87
UMichigan	MI	1180	65	68	16	15,470	85
UPenn	PA	1285	80	36	11	27,553	90
UVA	VA	1225	77	44	14	13,349	92
UWisconsin	WI	1085	40	69	15	11,857	71
Yale	CT	1375	95	19	11	43,514	96



Univ	State	SAT	Top10	Accept	SFRatio	Expenses	GradRate
Brown	RI	1310	89	22	13	22,704	94
CalTech	CA	1415	100	25	6	63,575	81
CMU	PA	1260	62	59	9	25,026	72
Columbia	NY	1310	76	24	12	31,510	88
Cornell	NY	1280	83	33	13	21,864	90
Dartmouth	NH	1340	89	23	10	32,162	95
Duke	NC	1315	90	30	12	31,585	95
Georgetown	DC	1255	74	24	12	20,126	92
Harvard	MA	1400	91	14	11	39,525	97
JohnsHopkins	MD	1305	75	44	7	58,691	87
MIT	MA	1380	94	30	10	34,870	91
Northwestern	IL	1260	85	39	11	28,052	89
NotreDame	IN	1255	81	42	13	15,122	94
PennState	PA	1081	38	54	18	10,185	80
Princeton	NJ	1375	91	14	8	30,220	95
Purdue	IN	1005	28	90	19	9,066	69
Stanford	CA	1360	90	20	12	36,450	93
TexasA&M	TX	1075	49	67	25	8,704	67
UCBerkeley	CA	1240	95	40	17	15,140	78
UChicago	IL	1290	75	50	13	38,380	87
UMichigan	MI	1180	65	68	16	15,470	85
UPenn	PA	1285	80	36	11	27,553	90
UVA	VA	1225	77	44	14	13,349	92
UWisconsin	WI	1085	40	69	15	11,857	71
Yale	CT	1375	95	19	11	43,514	96

Case study - UG Business Programs

Code Block - Clustering.ipynb

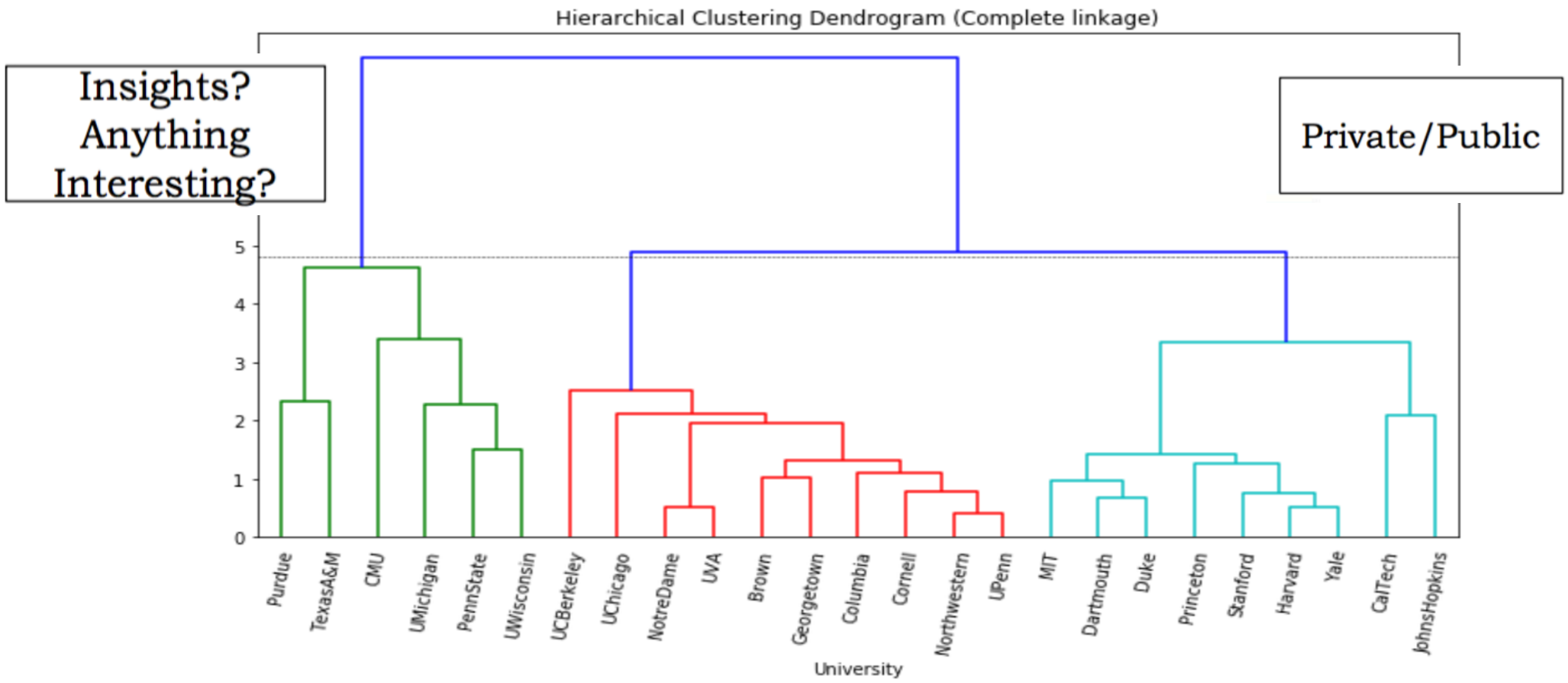
Why cluster universities?

- How can clustering help a prospective applicant?
- How can clustering help a business school dean?

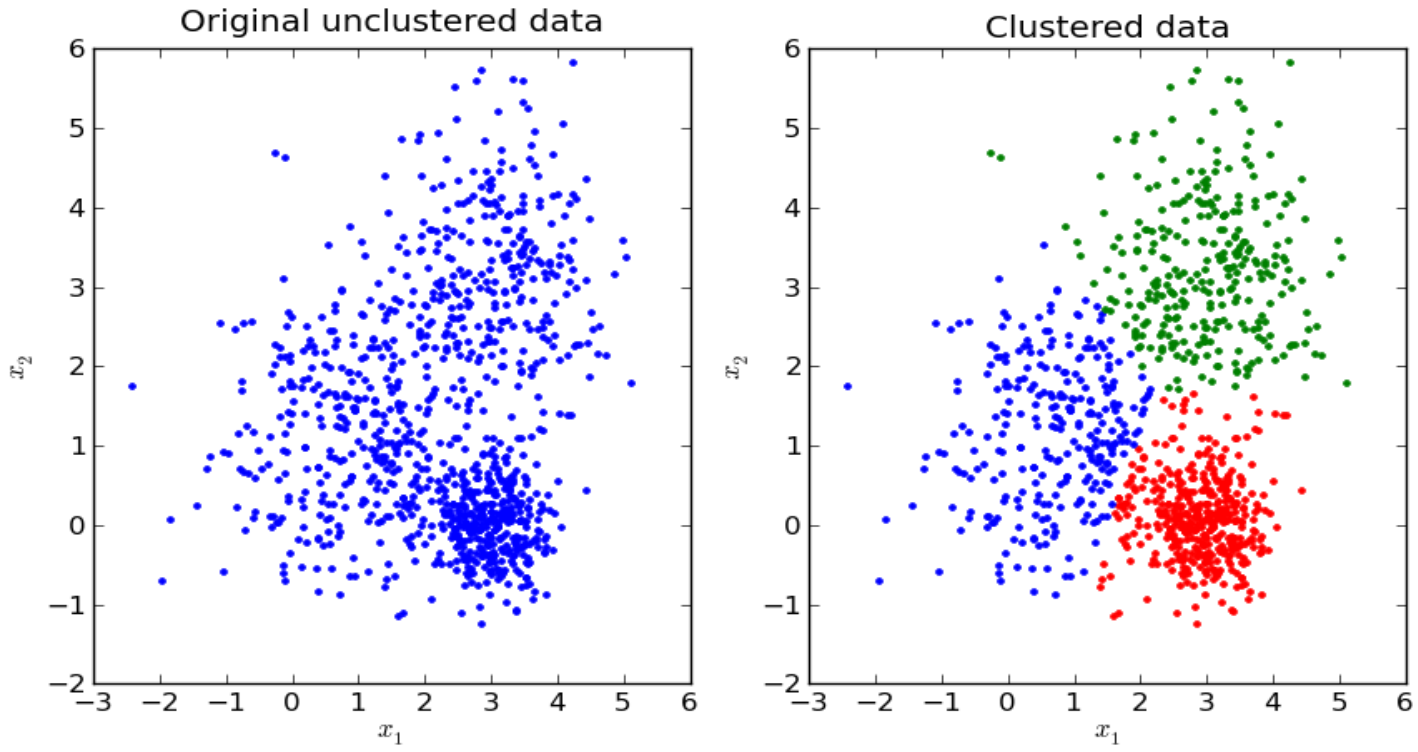
Evaluating Usefulness of Clustering

- What characterizes each cluster?
- Can we give a “name” to each cluster?
- Does this give us any insight?

Evaluating usefulness of clustering



K-Means Clustering



K-means clustering

- Predetermined number (K) of non-overlapping clusters
- Clusters are homogeneous yet dissimilar to other clusters
  - The "cluster center" is the arithmetic mean of all the points belonging to the cluster.
  - Each point is closer to its own cluster center than to other cluster centers.
- No hierarchy (no dendrogram)! End-product is final cluster memberships
- Useful for large datasets

K-means clustering - Algorithm

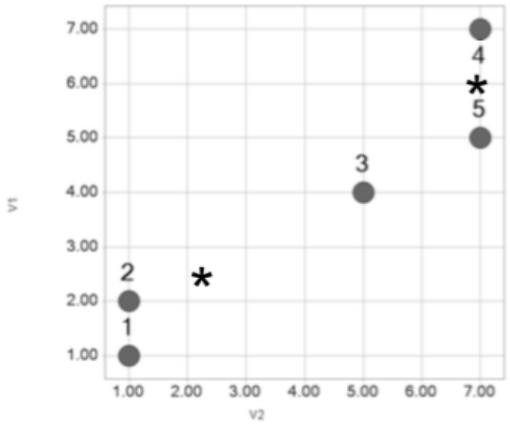
- Randomly select k objects from data set as initial cluster centers or means
- Assign each observation to their closest centroid, based on the Euclidean distance between the object and centroid
- For each of the k clusters
  - Update cluster centroid by calculating new mean values of all data points in cluster.
  - Centroid of k^{th} cluster is a *p* dimensional vector containing the means of all variables for the observations in the k^{th} cluster; *p* is the number of variables.
- Iteratively minimize the total within sum of square. Stop until cluster assignments stop changing or the maximum number of iterations is reached

$$tot.withiness = \sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

Toy Example – Step by Step

- Example K = 2

item	v1	v2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7



- Start with cluster A: 1,2,3 and cluster B: 4,5
- Compute cluster centroids

Toy Example – Cluster Centroid

What are the centroids of clusters A and B?

item	v1	v2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7

Centroid A = $\left(\frac{1+2+4}{3} = 2.33, \frac{1+1+5}{3} = 2.33\right)$

Centroid B = $\left(\frac{5+7}{2} = 6 , \frac{7+7}{2} = 7\right)$

Toy Example – Reassign Records

- Compute Euclidean distance of each record from each centroid, and re-assign to closest cluster

	Cluster A	Cluster B
Item 1	$\sqrt{(1-2.33)^2+(1-2.33)^2} = 1.89$	$\sqrt{(1-6)^2+(1-7)^2} = 7.81$
Item 2	1.37	7.21
Item 3	$\sqrt{(4-2.33)^2+(5-2.33)^2} = 3.14$	$\sqrt{(4-6)^2+(5-7)^2} = 2.83$
Item 4	6.60	1
Item 5	5.37	1

First Iteration results

- Cluster A: 1,2 Cluster B: 3,4,5
- Re-compute centroids
  - Centroid A = (1.5, 1), Centroid B = (5.33, 6.33)
- Re-compute distances of records to centroids

	Cluster A	Cluster B
Item 1	$\sqrt{(1-1.5)^2 + (1-1)^2} = 0.5$	$\sqrt{(1-5.33)^2+(1-6.33)^2} = 6.87$
Item 2	0.5	6.29
Item 3	$\sqrt{(4-1.5)^2+ (5-1)^2} = 4.72$	$\sqrt{(4-5.33)^2+ (5-6.33)^2} =1.89$
Item 4	8.14	1.80
Item 5	6.95	0.75

Case study - UG Business Programs

- Code block – Clustering.ipynb

Evaluating Usefulness of Clustering

- What characterizes each cluster?
- Can we give a “name” to each cluster?
- Does this give us any insight?

Determining Optimal Clusters

- Elbow Method
- Average Silhouette Method
- Calinkski-Harabasz Index

Elbow Method

- Objective of k-means clustering is to define clusters such that the total intra-cluster variation (known as total within-cluster variation or total within-cluster sum of square) is minimized

$$minimize\ (\sum_{k=1}^{k} W(Ck))$$

where C_k is the k^{th} cluster and $W(C_k)$ is the within-cluster variation.

- The total within-cluster sum of square (wss) measures the compactness of the clustering.

$$tot.withiness = \sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

Average Silhouette Method

- **Silhouette analysis** measures how well an observation is clustered and it estimates the **average distance between clusters**.
- Silhouette plot displays a measure of how close each point in one cluster is to points in neighboring clusters.
- For each observation ***i***, silhouette width s_i is calculated as:
  - ***a*** = average distance of ***i*** to the points in the same cluster
  - ***b*** = min (distance of ***i*** to points in another cluster)
- **Silhouette width** of the observation ***i*** is defined as: $\mathbf{Si} = \frac{b_i - ai}{max(ai,bi)}$
- Typically between 0 and 1.
- The closer to 1 the better.

Calinski-Harabasz Index

- Also known as variance ratio criterion
- Math formula is

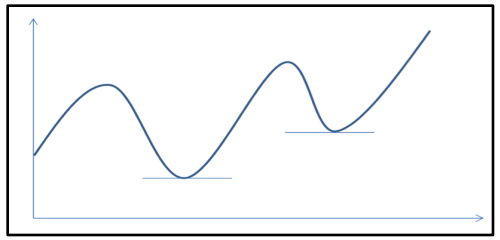
$$\frac{SS_B}{SS_W} \times \frac{N-k}{k-1}$$

- K = number of clusters
- N = total number of observations (data points)
- SS_W = overall within-cluster variance (overall within-cluster variance) – Calculated in Elbow Method
- SS_B = total sum of squares (Tss) minus SS_W
- *Tss* = squared distance of all data points from the dataset's centroid, this measure is independent with the number of cluster
- Optimal Clustering size has the largest CH index

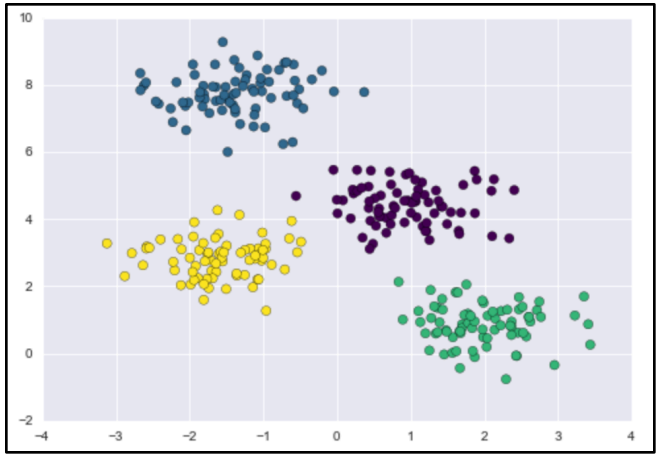
Issues with K-Means - 1

Global optimal result may not be achieved

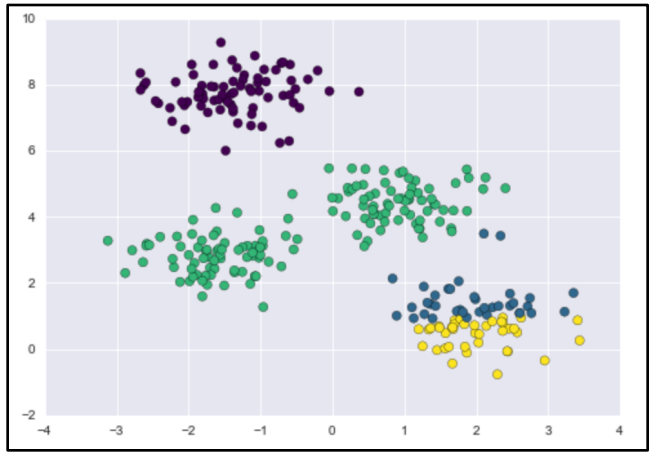
- K-means clustering is a minimization problem
- Existence of multiple local minima
- Use of different random seed can lead to poor results



Random seed 1



Random seed 2



Some alternatives to random initialization of the central points

- Multiple runs – Helps, but probability is not on your side (**n_init** attribute in scikit-learn's implementation of KMeans)

- Select original set of points by methods other than random.
  - k-means++: selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.

Issues with K-Means - 2

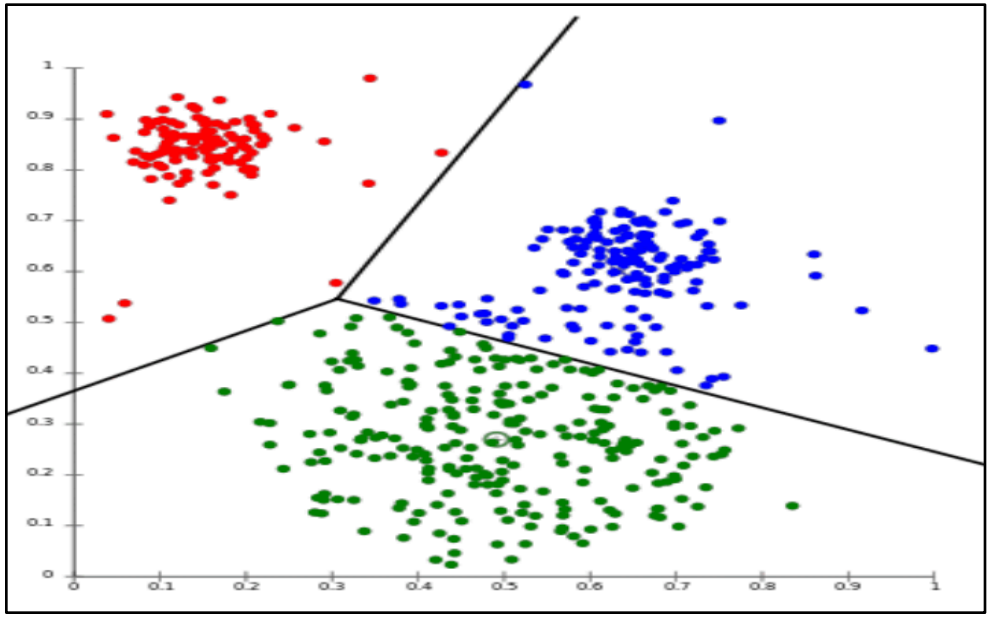
Number of clusters must be selected beforehand

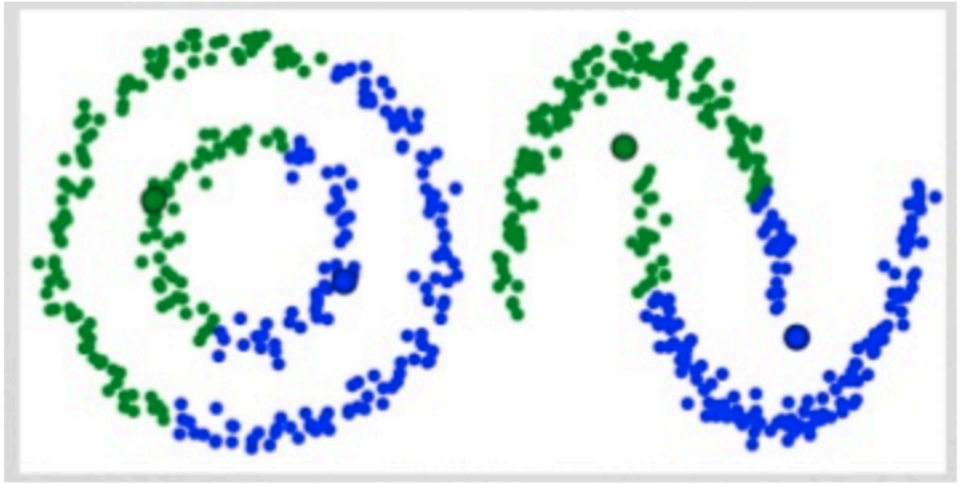
- Alternatively, complicated clustering algorithm which have better quantitative measure of the fitness per number of clusters can be used e.g., **Gaussian mixture models**
- Or, which *can* choose a suitable number of clusters e.g., **DBSCAN, mean-shift, or affinity propagation**

Issues with K-Means - 3

k-means is limited to linear cluster boundaries

- boundaries between *k*-means clusters will always be linear, will fail for more complicated boundaries



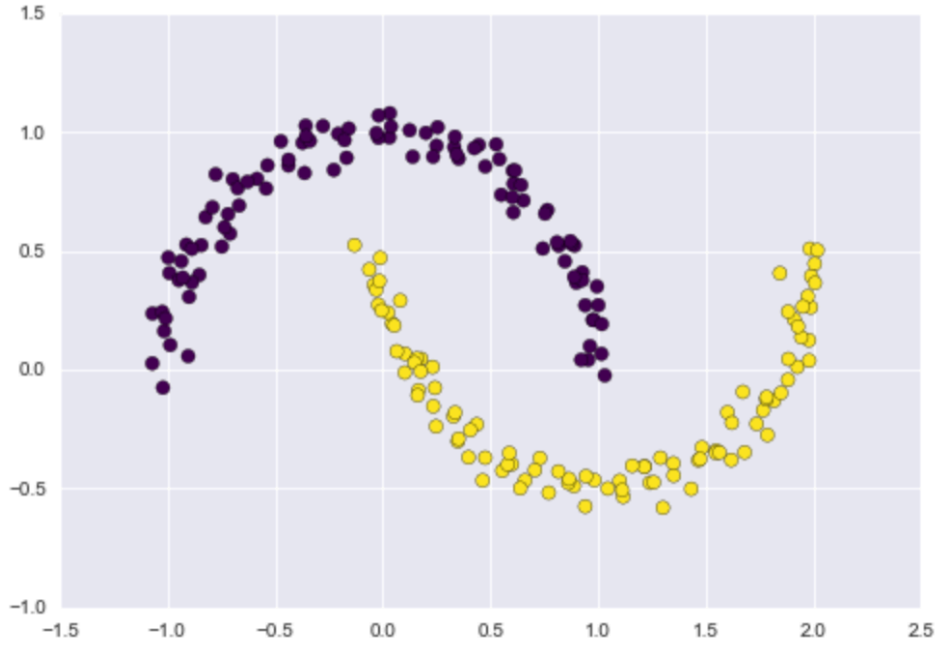


Two failure cases for K-Means

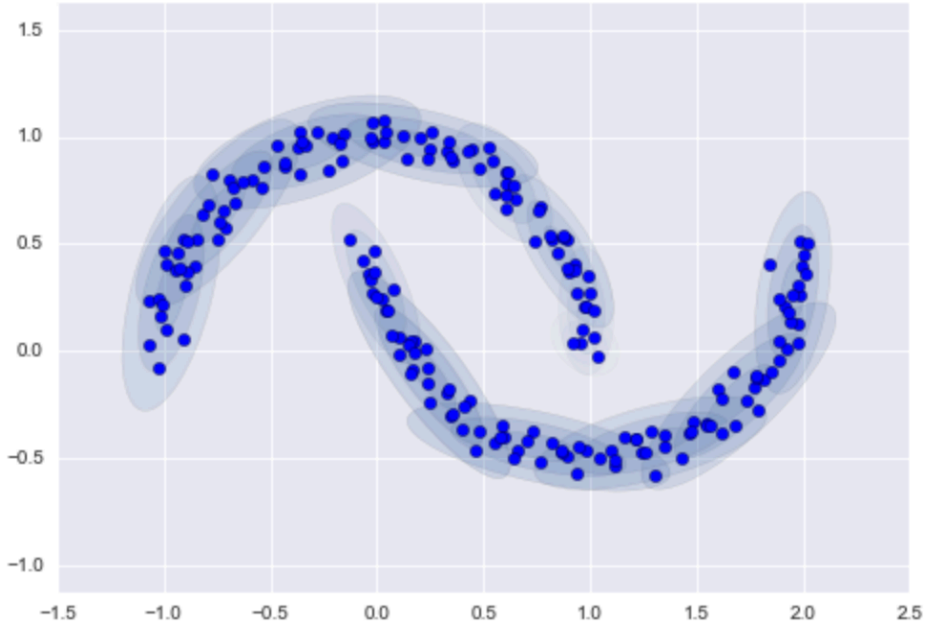
Issues with K-Means - 3

- Use of more sophiticated algorithm can solve the issue of linear boundaries – e.g **Spectral clustering, Gaussian mixture models**
-

Spectral clustering



GMM, 16 components



Issues with K-Means - 4

K-means can be slow for large numbers of samples

- K-means is NP-hard. For ***d*** dimensions, ***k*** clusters, and ***n*** observations, we will find a solution in ***O(ndk+1)*** time.
  - MiniBatch K-means - takes portions (batches) of data instead of fitting the whole dataset and then moves centroids by taking the average of the previous steps
  - https://scikit-learn.org/stable/modules/clustering.html#mini-batch-kmeans

Thank You