21. Clustering

Bruce E. Shapiro

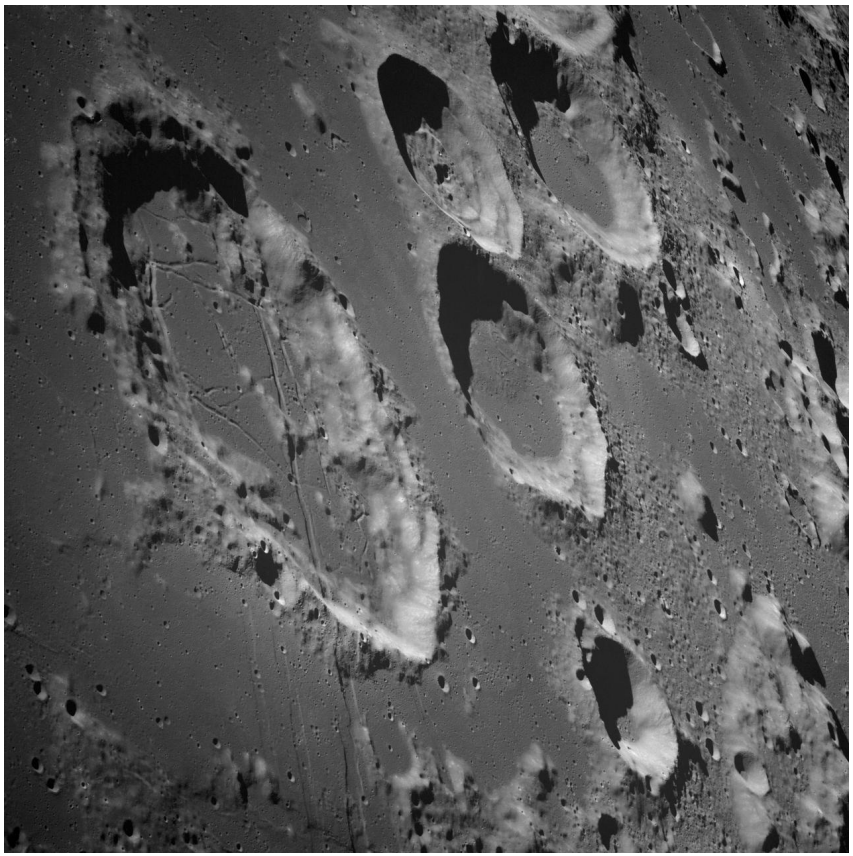
Getting Started in Machine Learning

Copyright (c) 2019. May not be distributed in any form without written permission.

Last revised: April 28, 2019

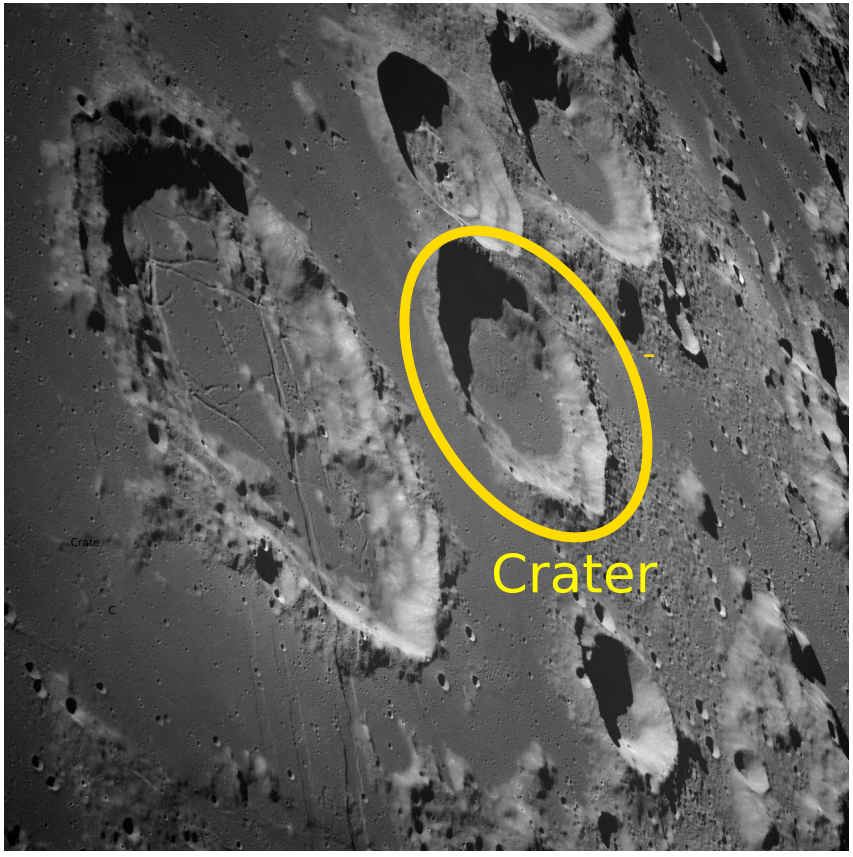
Clustering

- No exemplar ($\mathbf{Y}$) values to use for training
- Goal: find "hidden" patterns in the data



Clustering

- No exemplar ($\mathbf{Y}$) values to use for training
- Goal: find "hidden" patterns in the data



K-Means Method

- Designate number of desired clusters K
- Randomly place K points in the feature space. Designate these K points as the nominal cluster centroids.

K-Means Method

- Designate number of desired clusters K
- Randomly place K points in the feature space. Designate these K points as the nominal cluster centroids.
- Assign points to clusters based on nearest cluster centers.

K-Means Method

- Designate number of desired clusters K
- Randomly place K points in the feature space. Designate these K points as the nominal cluster centroids.
- Assign points to clusters based on nearest cluster centers.
- Calculate centroid of each cluster.

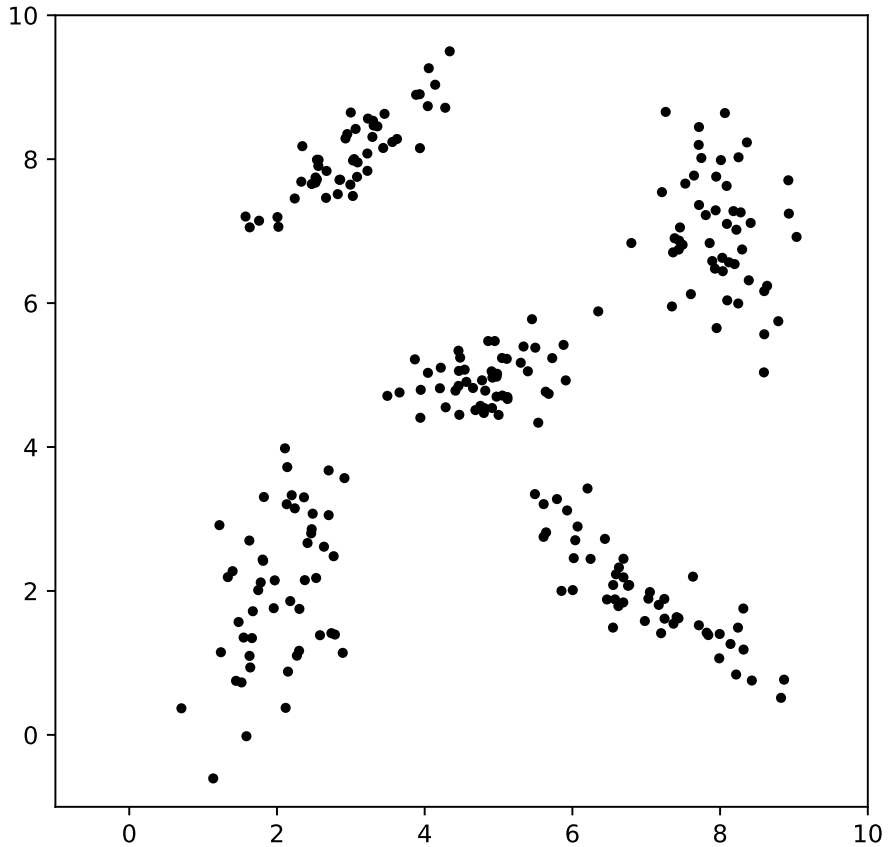
K-Means Method

- Designate number of desired clusters K
- Randomly place K points in the feature space. Designate these K points as the nominal cluster centroids.
- Assign points to clusters based on nearest cluster centers.
- Calculate centroid of each cluster.
- Repeat until centroid locations are converged.

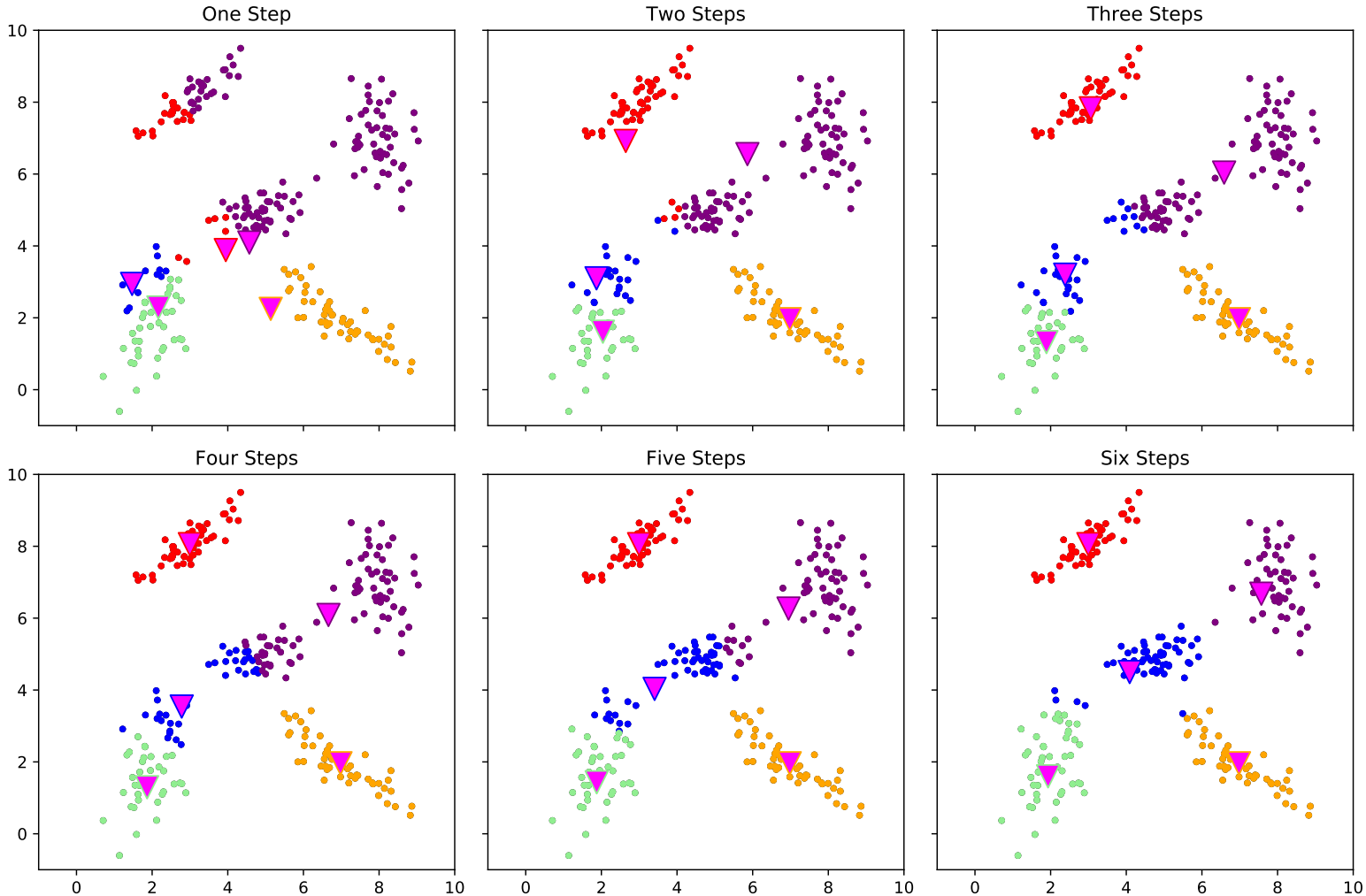
K-Means Method

- Designate number of desired clusters K
- Randomly place K points in the feature space. Designate these K points as the nominal cluster centroids.
- Assign points to clusters based on nearest cluster centers.
- Calculate centroid of each cluster.
- Repeat until centroid locations are converged.
- Problem: different initial guesses *may* lead to different results

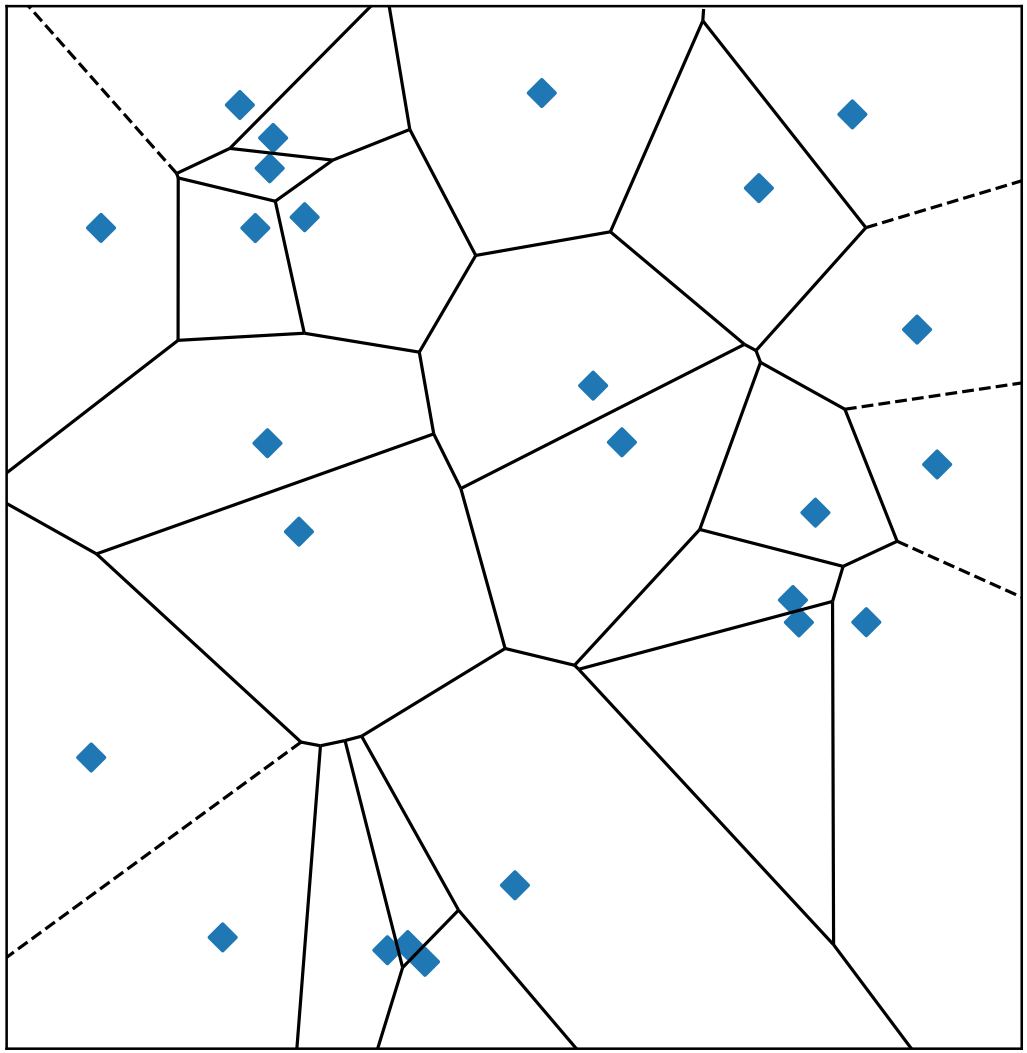
Example: Kmeans Data Set



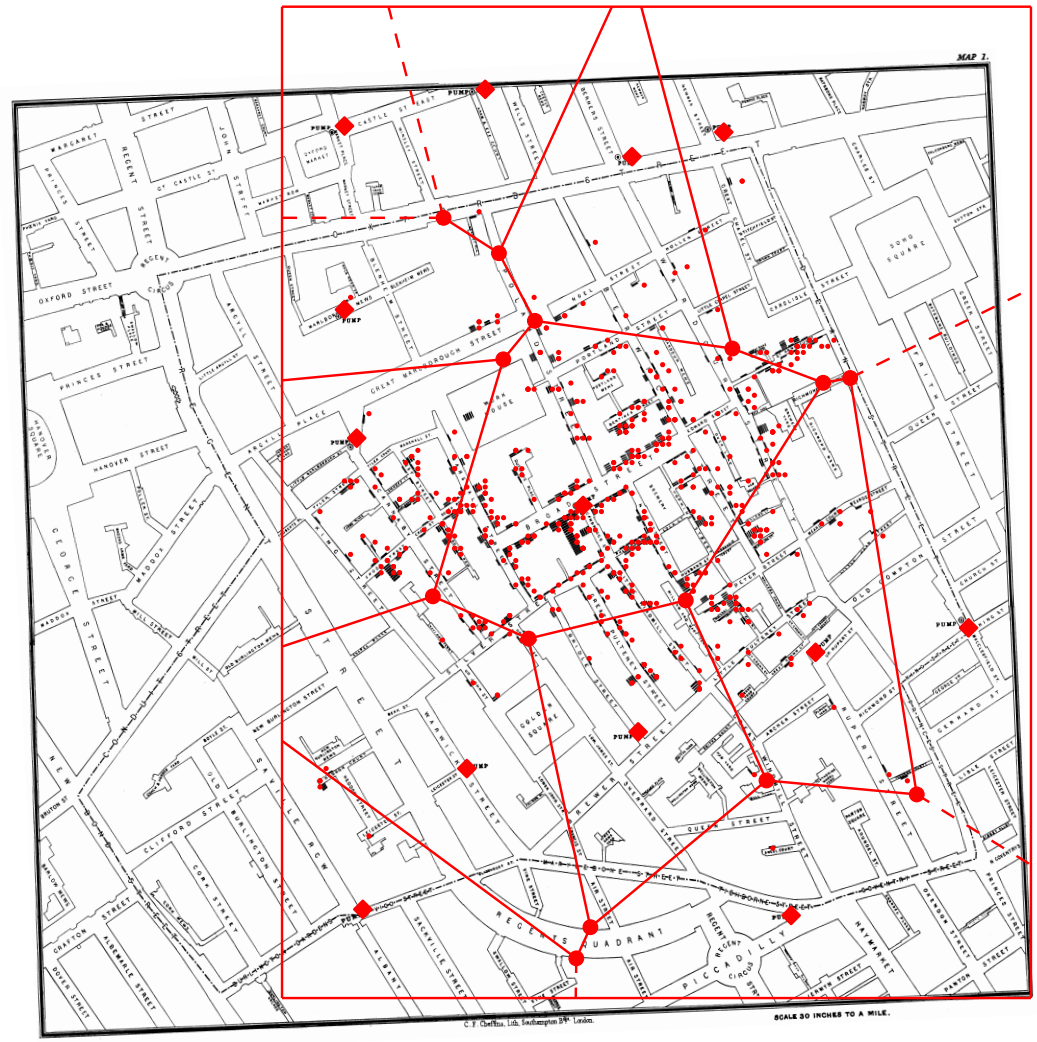
Example: Kmeans Iterations



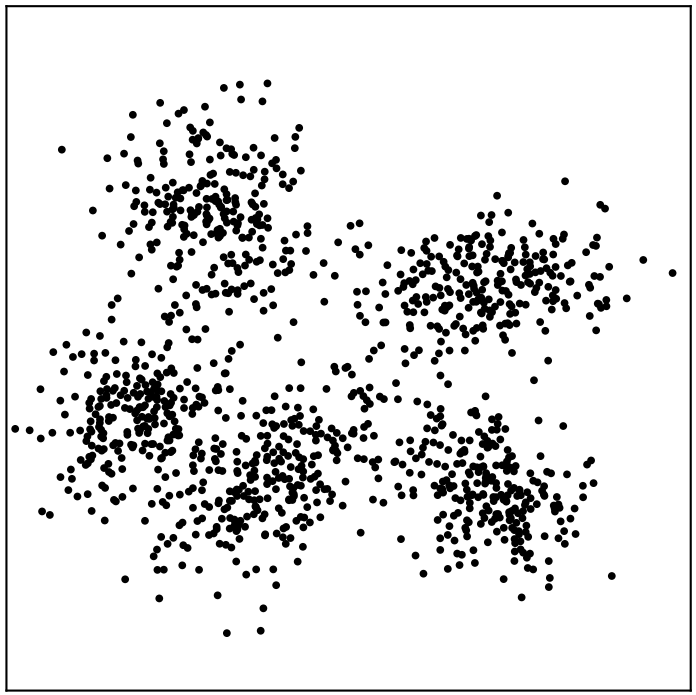
Clusters converge to Voronoi Cells

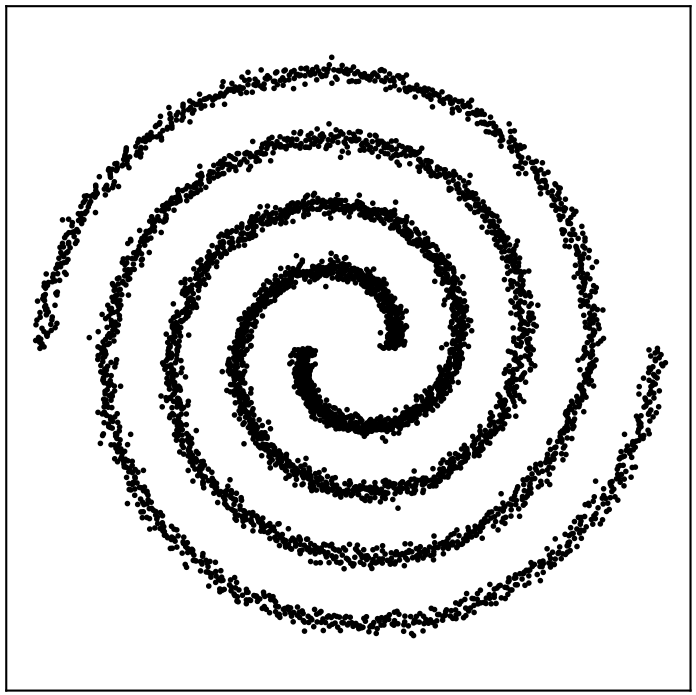


London Cholera Epidemic, 1854



K-Means in Python: Examples (Input)





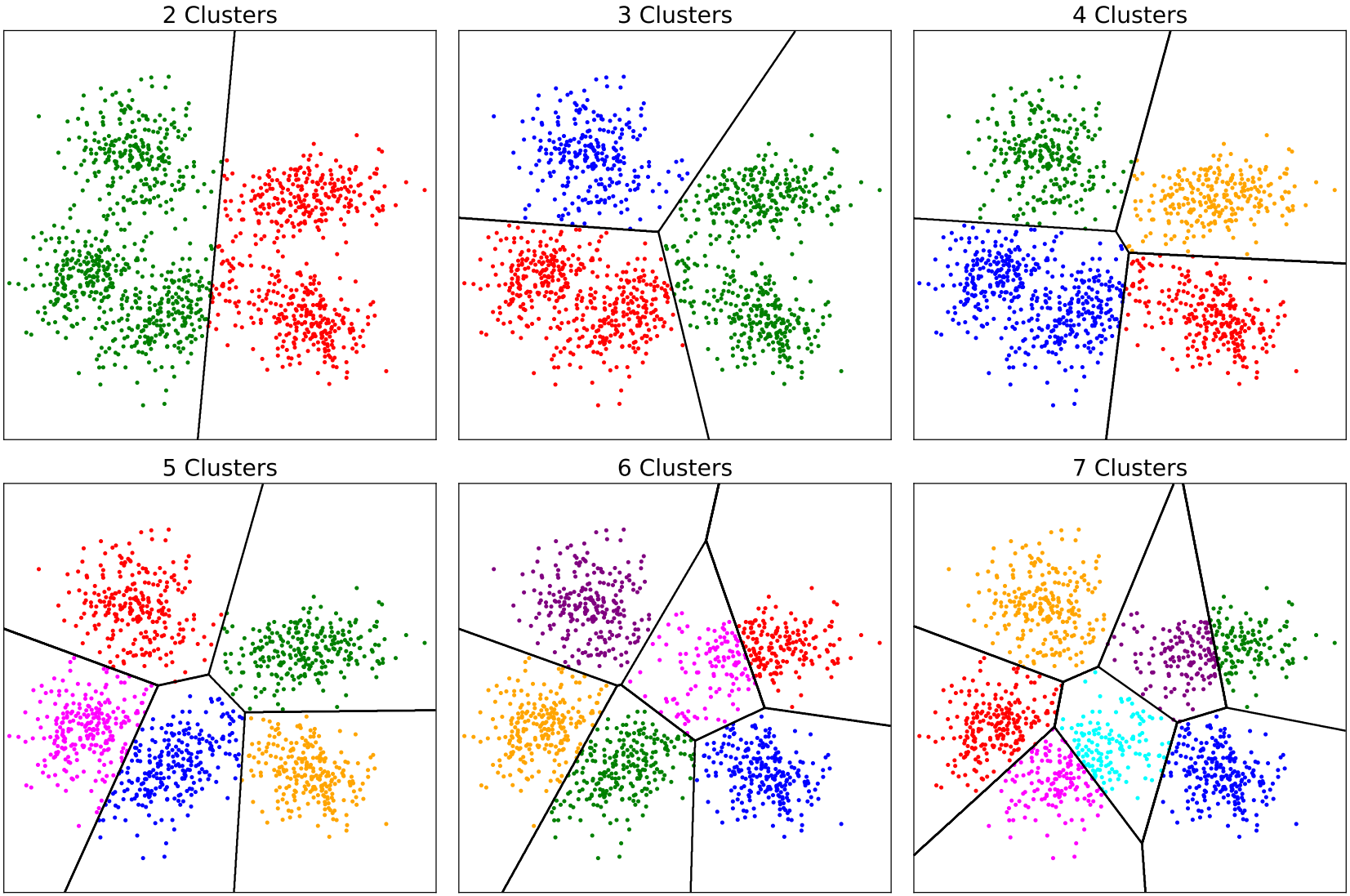
KMeans in Python

- **`X`** must be a **`numpy array`** containing the point coordinates
- **`K`** must be an integer set equal to the value of the number of clusters desired
- Code for finding clusters:

```
from sklearn.cluster import KMeans
model = KMeans(n_clusters=K)
model.fit(X)
```

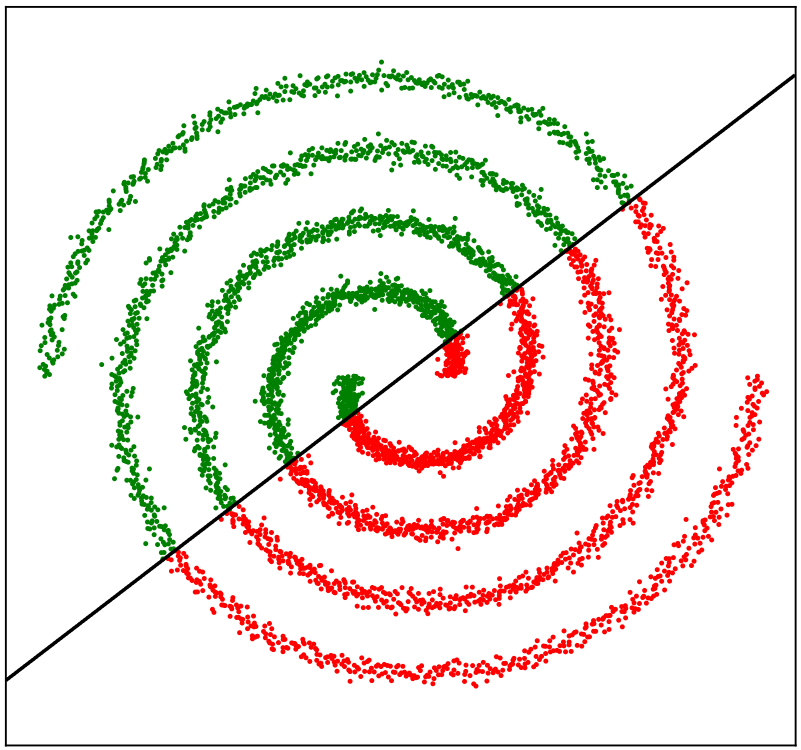
- Cluster labels are in **`model.labels_`**, an array of integers of the same length a **`X`** with each value between 0 and $K-1$

KMeans Results for Gaussian Clouds

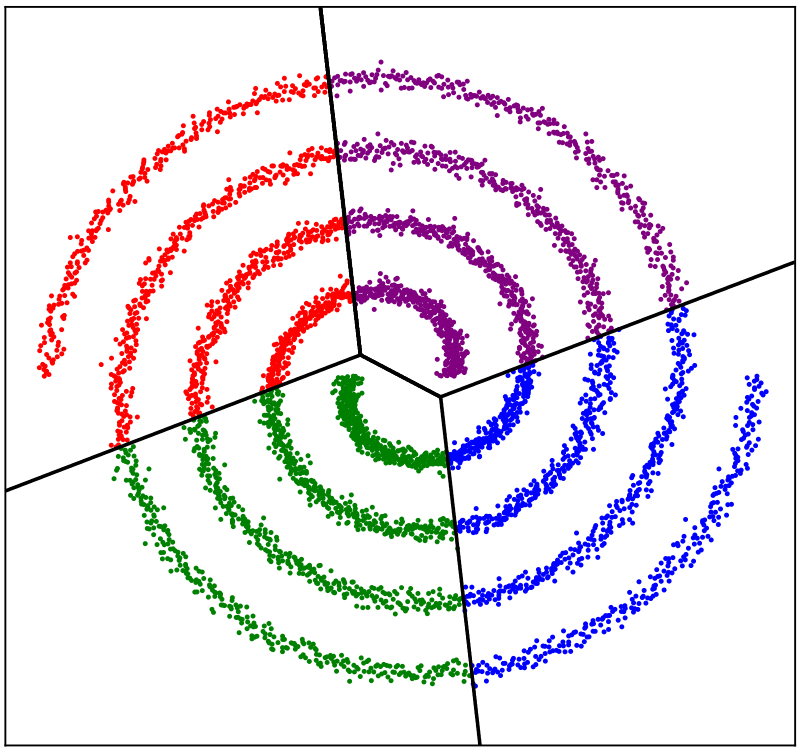


KMeans Results for Spirals

2 Clusters



4 Clusters



What is the Right Number of Clusters? Silhouette Index

- Definitions:

$$\mu_i(\mathbf{p}_i) = \text{mean distance from } \mathbf{p}_i \text{ to points } \mathbf{q}_j \neq \mathbf{p}_i, \forall \mathbf{q}_j \in C$$
$$\mu'_i(\mathbf{p}_i) = \text{mean distance from } \mathbf{p}_i \text{ to points } \mathbf{q}_j, \forall \mathbf{q}_j \notin C$$
$$M_i(\mathbf{p}_i) = \max_j(\mu_i, \mu'_i)$$

What is the Right Number of Clusters? Silhouette Index

- Definitions:

$$\mu_i(\mathbf{p}_i) = \text{mean distance from } \mathbf{p}_i \text{ to points } \mathbf{q}_j \neq \mathbf{p}_i, \forall \mathbf{q}_j \in C$$
$$\mu_i'(\mathbf{p}_i) = \text{mean distance from } \mathbf{p}_i \text{ to points } \mathbf{q}_j, \forall \mathbf{q}_j \notin C$$
$$M_i(\mathbf{p}_i) = \max_j(\mu_i, \mu_i')$$

- Silhouette Score for each point: $S_i = \dfrac{\mu_i'(\mathbf{p}_i) - \mu_i(\mathbf{p}_i)}{M_i(\mathbf{p}_i)}$

What is the Right Number of Clusters? Silhouette Index

- Definitions:

$$\mu_i(\mathbf{p}_i) = \text{mean distance from } \mathbf{p}_i \text{ to points } \mathbf{q}_j \neq \mathbf{p}_i, \forall \mathbf{q}_j \in C$$
$$\mu_i'(\mathbf{p}_i) = \text{mean distance from } \mathbf{p}_i \text{ to points } \mathbf{q}_j, \forall \mathbf{q}_j \notin C$$
$$M_i(\mathbf{p}_i) = \max_j(\mu_i, \mu_i')$$

- Silhouette Score for each point: $S_i = \dfrac{\mu_i'(\mathbf{p}_i) - \mu_i(\mathbf{p}_i)}{M_i(\mathbf{p}_i)}$
- Silhouette Index: $S = \dfrac{1}{n}\sum_{i=1}^{n} S_i$
- $0 < S < 1$, with $S \to 1$ for better defined clustering

What is the Right Number of Clusters? Calinski-Harabansz (CI) Index

- Recall the **Scatter Matrix** $\mathbf{S} = \mathbf{X}^\mathsf{T}\mathbf{X} = (m-1)\text{cov}\mathbf{X}$ where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 - \boldsymbol{\mu} \\ \mathbf{x}_1 - \boldsymbol{\mu} \\ \vdots \\ \mathbf{x}_{m-1} - \boldsymbol{\mu} \end{bmatrix}$$

  Here $\mathbf{x}_i$ are the feature vectors and $\boldsymbol{\mu}$ is a vector of means of the components of the features

- Let

$$\mathbf{S} = \text{in-cluster scatter matrix}$$

$$\mathbf{S}' = \text{between-cluster scatter matrix}$$

CH Index (continued)

- Define the CH Index as

$$\text{CH} = \left[\frac{n-k}{k-1}\right]\frac{\text{tr}\,\mathbf{S}'}{\text{tr}\,\mathbf{S}} = \ldots(\text{algebra omitted}) = \frac{\sigma'^2_1 + \sigma'^2_2 + \cdots + \sigma'^2_k}{\sigma^2_1 + \sigma^2_2 + \cdots + \sigma^2_n}$$

- This ratio is known as the **dispersion**, because

$$\sigma_i^2 = \text{in-cluster variance}$$
$$(\sigma')_i^2 = \text{between-cluster variance}$$

- Very high CI index means well defined clusters
- Unlike silhouette, CI is not limited to [0, 1]

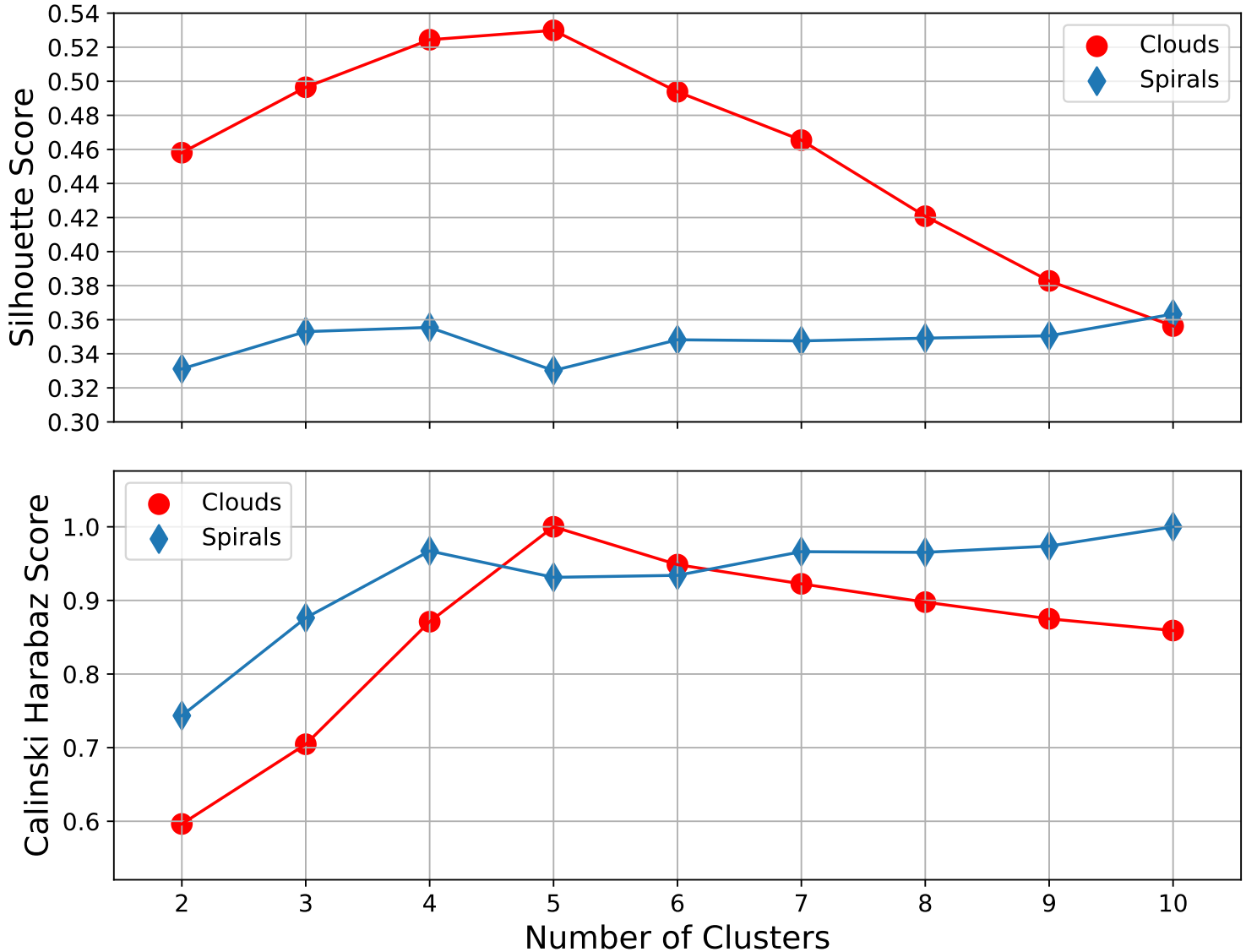
Silhouette and CH in Python

```
from sklearn.metrics import silhouette_score
from sklearn.metrics import calinski_harabaz_score as CH

from sklearn.clustering import KMeans
kmeans=KMeans(n_clusters=5)
kmeans.fit(X)

print(silhouette_score(X,kmeans.labels_))
print(CH(X,kemans.labels_))
```

```
0.514783493314483
1972.0630797994027
```



Note: CH scores are normalized to maximums; Silhouette scores are actual values.

Agglomerative Hierarchical Clustering

- Assign each point to a different cluster
- Group points together based on a **distance measure**
  - A distance measure measures the distance between individual points

Agglomerative Hierarchical Clustering

- Assign each point to a different cluster
- Group points together based on a **distance measure**
  - A distance measure measures the distance between individual points
- Group small clusters together into larger clusters using a **linkage measure**
  - A linkage measure measures the distance between clusters

Agglomerative Hierarchical Clustering

- Assign each point to a different cluster
- Group points together based on a **distance measure**
  - A distance measure measures the distance between individual points
- Group small clusters together into larger clusters using a **linkage measure**
  - A linkage measure measures the distance between clusters
- Repeat until all clusters are grouped together
- If you "connect the dots" as you are grouping you end up with a binary tree

Distance measures

- A distance measure measures the distance between individual points
- Euclidean distance $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum (p_i - q_i)^2} = \|\mathbf{p} - \mathbf{q}\|_2$.

Distance measures

- A distance measure measures the distance between individual points
- Euclidean distance $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum (p_i - q_i)^2} = \|\mathbf{p} - \mathbf{q}\|_2$.
- L_1 (Manhattan) norm $d(\mathbf{p}, \mathbf{q}) = \sum |p_i - q_i| = \|\mathbf{p} - \mathbf{q}\|_1$

Distance measures

- A distance measure measures the distance between individual points
- Euclidean distance $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum (p_i - q_i)^2} = \|\mathbf{p} - \mathbf{q}\|_2$.
- L_1 (Manhattan) norm $d(\mathbf{p}, \mathbf{q}) = \sum |p_i - q_i| = \|\mathbf{p} - \mathbf{q}\|_1$
- L_∞ (max) norm $d(\mathbf{p}, \mathbf{q}) = \max_i |p_i - q_i| = \|\mathbf{p} - \mathbf{q}\|_\infty$

Distance measures

- A distance measure measures the distance between individual points
- Euclidean distance $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum (p_i - q_i)^2} = \|\mathbf{p} - \mathbf{q}\|_2$.
- L_1 (Manhattan) norm $d(\mathbf{p}, \mathbf{q}) = \sum |p_i - q_i| = \|\mathbf{p} - \mathbf{q}\|_1$
- L_∞ (max) norm $d(\mathbf{p}, \mathbf{q}) = \max_i |p_i - q_i| = \|\mathbf{p} - \mathbf{q}\|_\infty$
- Mahalanobis distance $d(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^\mathsf{T} \mathbf{S}^{-1} (\mathbf{p} - \mathbf{q})}$

Distance measures

- A distance measure measures the distance between individual points
- Euclidean distance $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum (p_i - q_i)^2} = \|\mathbf{p} - \mathbf{q}\|_2$.
- L_1 (Manhattan) norm $d(\mathbf{p}, \mathbf{q}) = \sum |p_i - q_i| = \|\mathbf{p} - \mathbf{q}\|_1$
- L_∞ (max) norm $d(\mathbf{p}, \mathbf{q}) = \max_i |p_i - q_i| = \|\mathbf{p} - \mathbf{q}\|_\infty$
- Mahalanobis distance $d(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^\mathsf{T} \mathbf{S}^{-1} (\mathbf{p} - \mathbf{q})}$
- Hamming distance for string (count of bit difference)

Linkage Measures

Measures distance between clusters

- Single Link: $L(A, B) = \min\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$

Linkage Measures

Measures distance between clusters

- Single Link: $L(A, B) = \min\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$
- Complete Link: $L(A, B) = \max\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$

Linkage Measures

Measures distance between clusters

- Single Link: $L(A, B) = \min\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$
- Complete Link: $L(A, B) = \max\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$
- Group Average: $L(A, B) = \frac{1}{|A||B|} \sum_{\mathbf{p} \in A, \mathbf{q} \in B} d(\mathbf{p}, \mathbf{q})$

Linkage Measures

Measures distance between clusters

- Single Link: $L(A, B) = \min\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$
- Complete Link: $L(A, B) = \max\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$
- Group Average: $L(A, B) = \frac{1}{|A||B|} \sum_{\mathbf{p} \in A, \mathbf{q} \in B} d(\mathbf{p}, \mathbf{q})$
- Mean Distance: $L(A, B) = d\left(\frac{1}{|A|} \sum_{\mathbf{p} \in A} \mathbf{p}, \frac{1}{|B|} \sum_{\mathbf{p} \in B} \mathbf{p}\right)$.

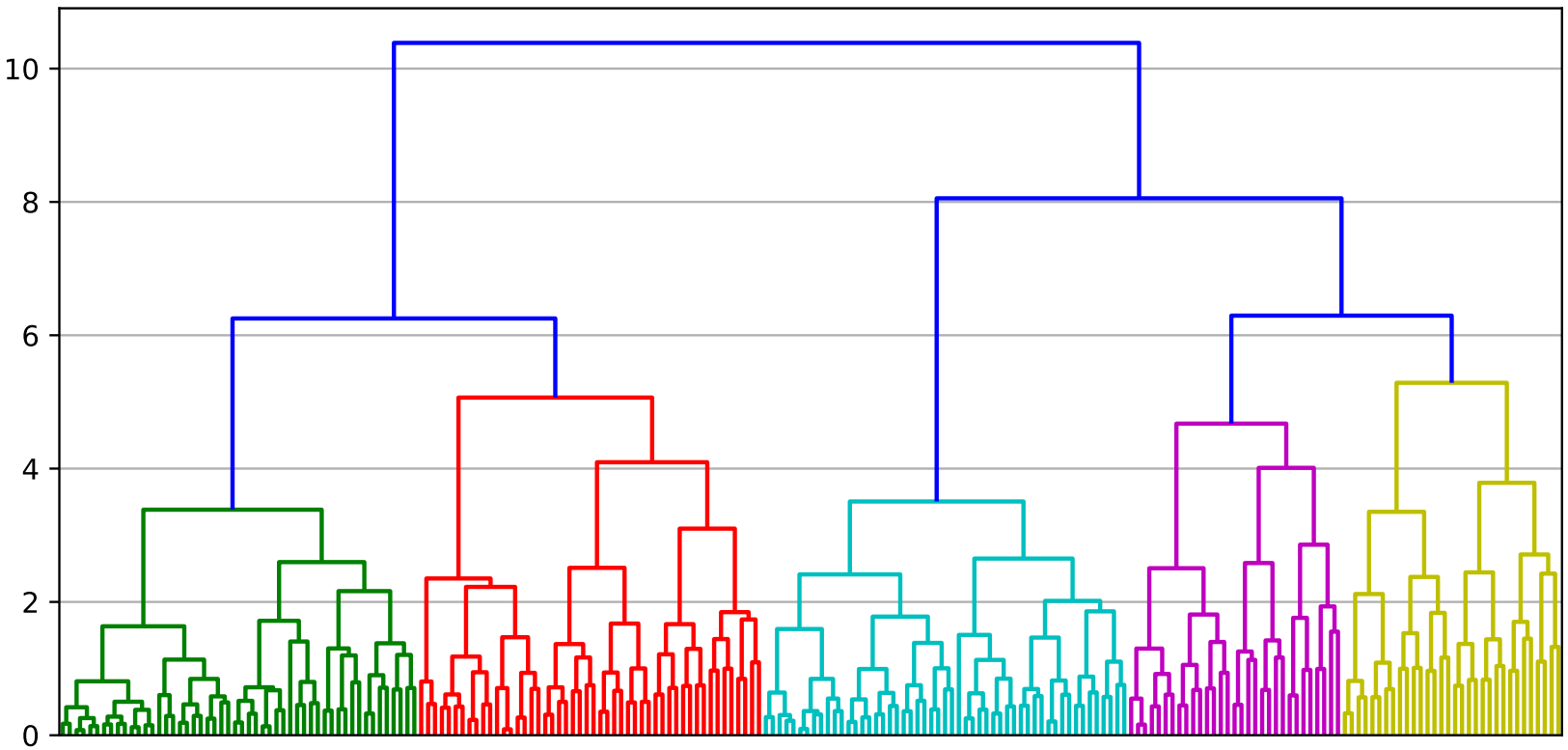
Linkage Measures

Measures distance between clusters

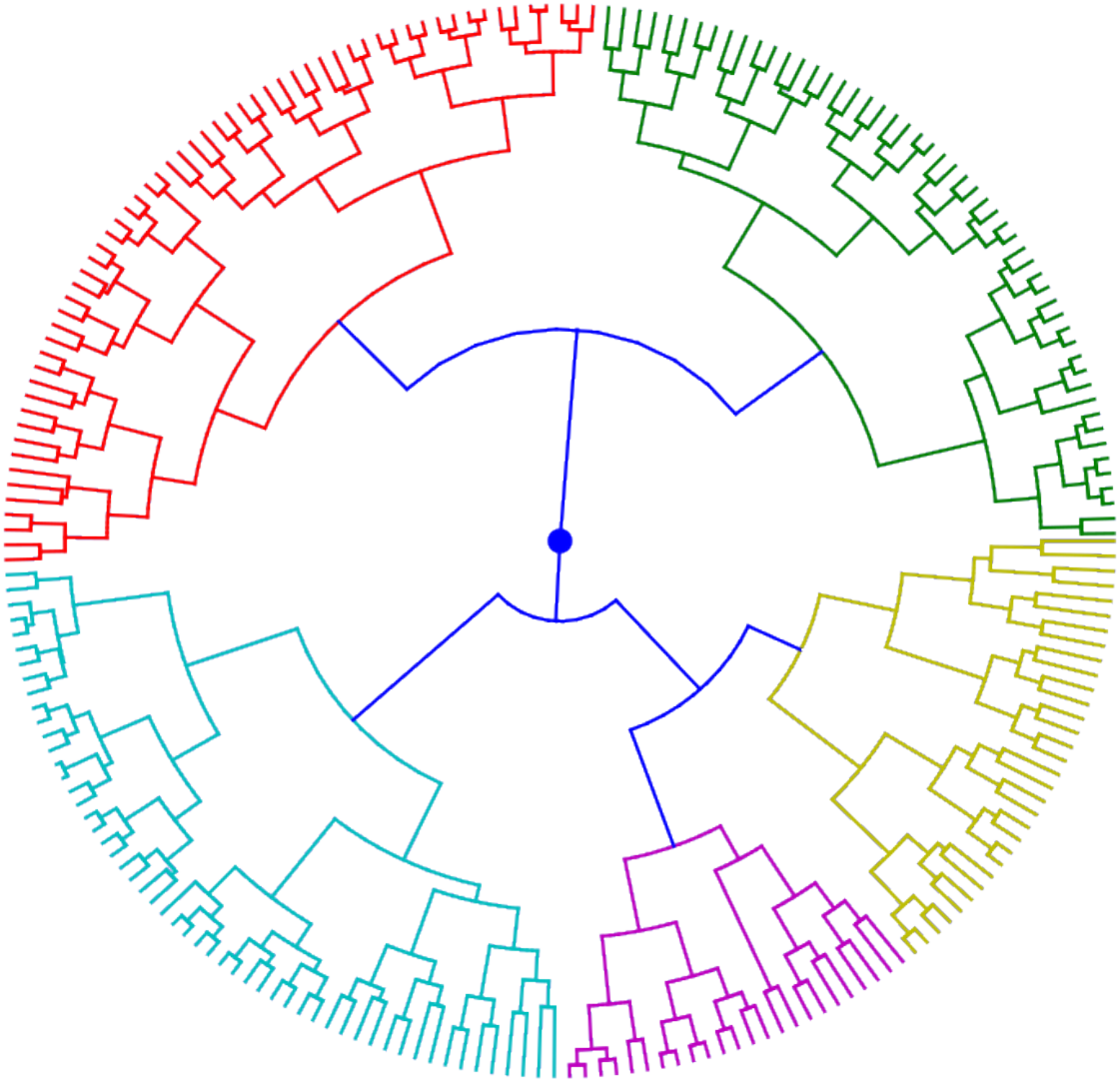
- Single Link: $L(A, B) = \min\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$
- Complete Link: $L(A, B) = \max\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{p} \in A, \mathbf{q} \in B\}$
- Group Average: $L(A, B) = \frac{1}{|A||B|} \sum_{\mathbf{p} \in A, \mathbf{q} \in B} d(\mathbf{p}, \mathbf{q})$
- Mean Distance: $L(A, B) = d\left(\frac{1}{|A|} \sum_{\mathbf{p} \in A} \mathbf{p}, \frac{1}{|B|} \sum_{\mathbf{p} \in B} \mathbf{p}\right)$.
- Ward's method (Minimum variance):

$$L(A, B) = \frac{|A||B|}{|A| + |B|} \left\| \left(\frac{1}{|A|} \sum_{\mathbf{p} \in A} \mathbf{p} - \frac{1}{|B|} \sum_{\mathbf{p} \in B} \mathbf{p} \right) \right\|_2^2$$

Dendrogram of Gaussian Clusters



Circularized Dendrogram



- Clustering in **sklearn**: labels clusters

```
from sklearn.cluster import AgglomerativeClustering as AC
clustering = AC(n_clusters=5).fit_predict(X)
```

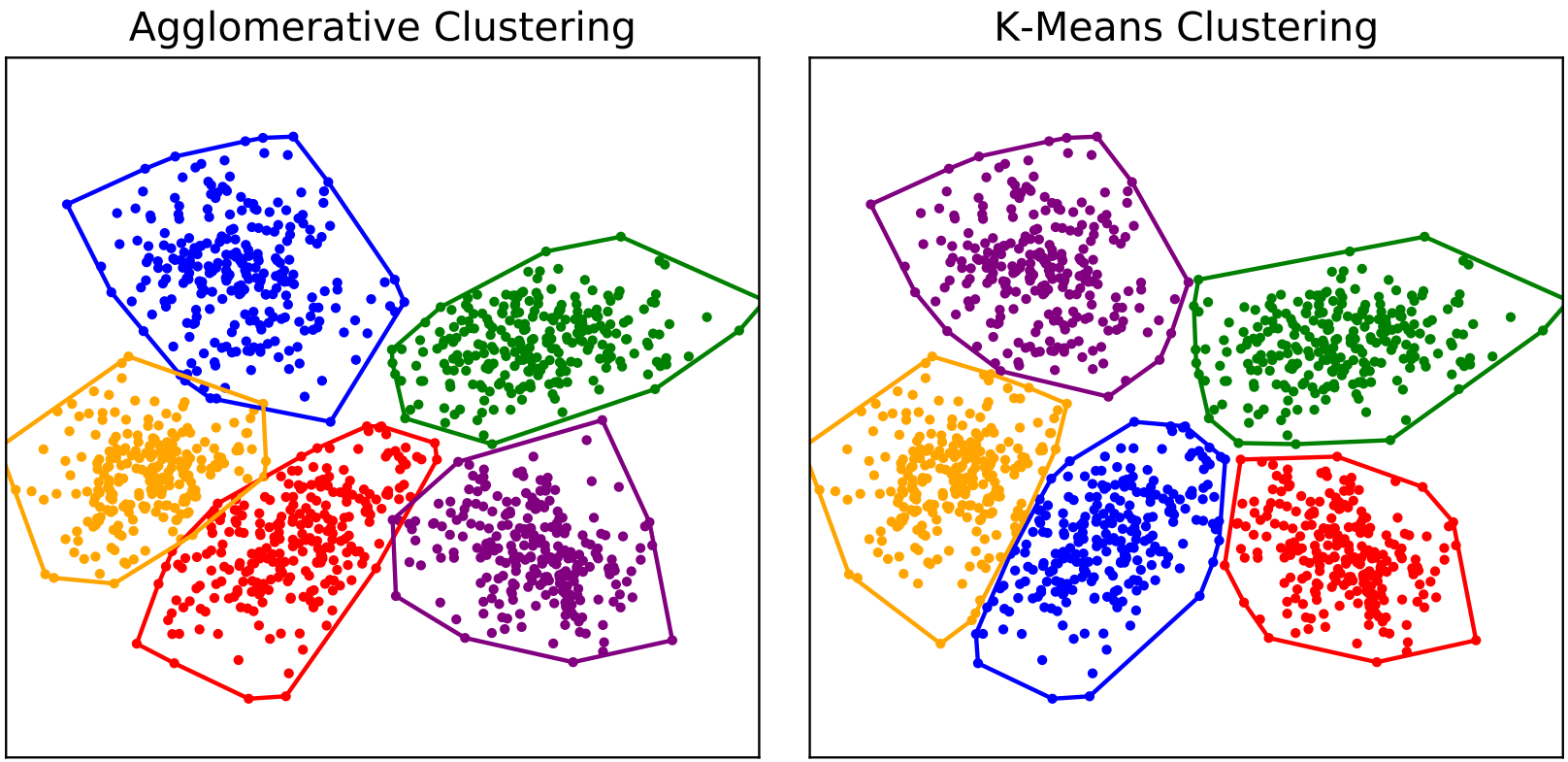
  the array **clustering** contains the cluster labels

- Clustering in **scipy**

```
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram,
    linkage
Z=linkage(X, method="ward")
dendrogram(Z,5,truncate_mode="level")
plt.show()
```

Comparing KMeans and Hierarchical

Display Convex Hull of each cluster



The **convex hull** of a set is the smallest convex set that includes every point in the set.

Topological Clustering: DBScan

- DBScan is a special case of a more general method: Spectral clustering
- Define an ϵ-**neighborhood** $N_\epsilon(\mathbf{p})$ of $\mathbf{p}$ as

$$N_\epsilon(\mathbf{p}) = \{\mathbf{q} | \|\mathbf{p} - \mathbf{q}\| \leq \epsilon\}$$

Topological Clustering: DBScan

- DBScan is a special case of a more general method: Spectral clustering
- Define an **ϵ-neighborhood** $N_\epsilon(\mathbf{p})$ of **p** as

$$N_\epsilon(\mathbf{p}) = \{\mathbf{q} | \|\mathbf{p} - \mathbf{q}\| \leq \epsilon\}$$

- We call **p** a **core point** if $|N_\epsilon(\mathbf{p})| \geq N$
  N is an integer parameter that defines the minimum number of points in a neighborhood to qualify as a core point.

Topological Clustering: DBScan

- DBScan is a special case of a more general method: Spectral clustering
- Define an **ϵ-neighborhood** $N_\epsilon(\mathbf{p})$ of **p** as

$$N_\epsilon(\mathbf{p}) = \{\mathbf{q} | \|\mathbf{p} - \mathbf{q}\| \leq \epsilon\}$$

- We call **p** a **core point** if $|N_\epsilon(\mathbf{p})| \geq N$
  N is an integer parameter that defines the minimum number of points in a neighborhood to qualify as a core point.
- We say that **p** is **directly density reachable from q** if
  - $\mathbf{p} \in N_\epsilon(q)$ and
  - $|N_\epsilon(\mathbf{q})| \geq N$

- We say that **p** is **density reachable** from **q** if there is a sequence of points $\mathbf{q} = \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n = \mathbf{q}$ such that each $\mathbf{p}_{i+1}$ is directly density reachable form $\mathbf{p}_i$.

- We say that **p** is **density reachable** from **q** if there is a sequence of points $\mathbf{q} = \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n = \mathbf{q}$ such that each $\mathbf{p}_{i+1}$ is directly density reachable form $\mathbf{p}_i$.
- We say that **p** is **density connected** to **q** if there is a point **x** such that both **p** and **q** are density reachable from **x**.

- We say that **p** is **density reachable** from **q** if there is a sequence of points $\mathbf{q} = \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n = \mathbf{q}$ such that each $\mathbf{p}_{i+1}$ is directly density reachable form $\mathbf{p}_i$.
- We say that **p** is **density connected** to **q** if there is a point **x** such that both **p** and **q** are density reachable from **x**.
- We say that the C is a **cluster of points** in the set S if
  - if $\mathbf{p} \in C$ and **q** is density reachable from **p**, then $\mathbf{q} \in C$
  - **p** is density connected to **q**

```
from sklearn.cluster import DBSCAN
clustering = DBSCAN(eps=.5, min_samples=5).fit(XSPIRAL)
clusteringlabels=list(set(clustering.labels_))
```