TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

Machine Learning at TACC
Unsupervised Learning

July 2024

PRESENTED BY:
Sikan Li

Research Engineering, Scalable Computational Intelligence

sli@tacc.utexas.edu

Slides from Kelly Pierce

Outline

Unsupervised Learning Overview

Data Representation: PCA, Manifold Learning

Data Representation Hands-On

Data Structure: Clustering algorithms

Clustering Hands-On

Data Density: GMM, DBSCAN

Data Density Hands-On

Data Clusters

Overview

Learning data clusters

When should you think about data clustering?

- When you expect discrete groupings in continuous data, but don’t know them *a priori.*
- When you are looking for natural classifications within your data.

Data Clusters

K-Means Clustering

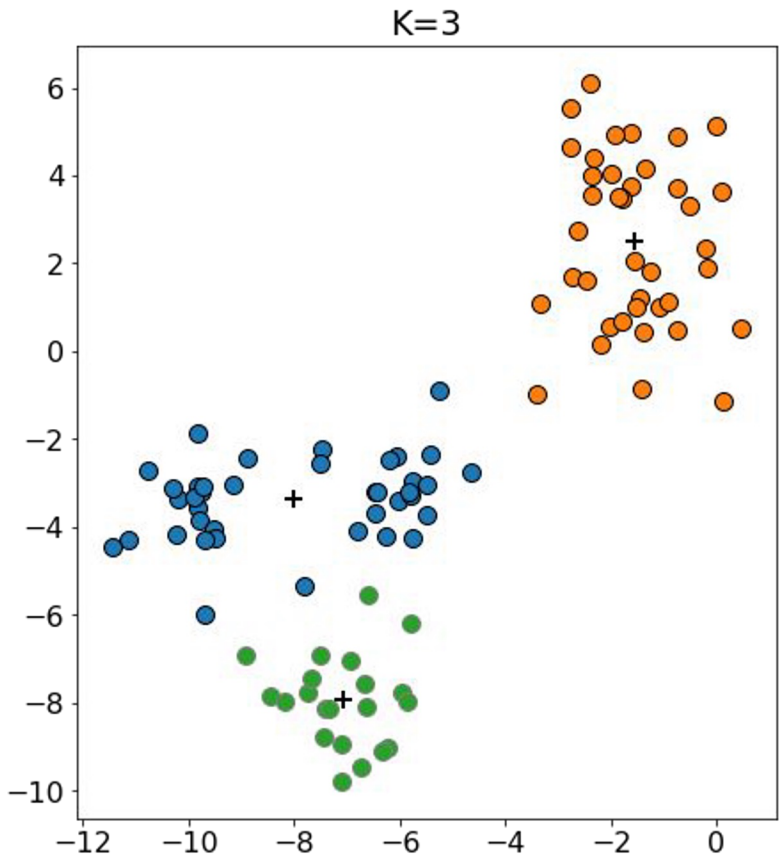
K-Means Clustering

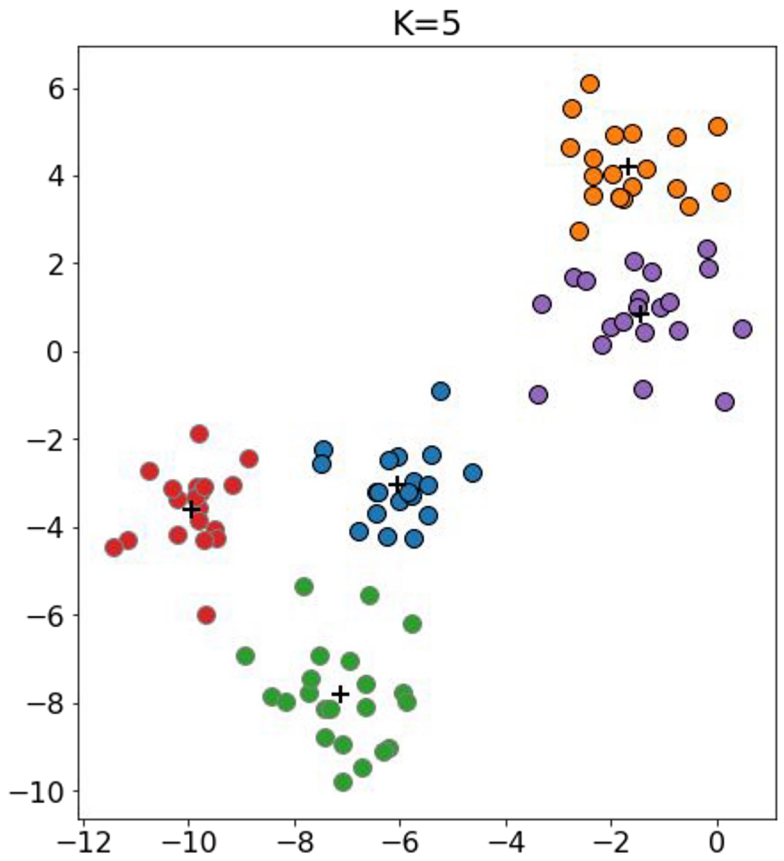
K-Means Clustering is a **partitional clustering** algorithm that seeks to identify cluster centroids in feature space.

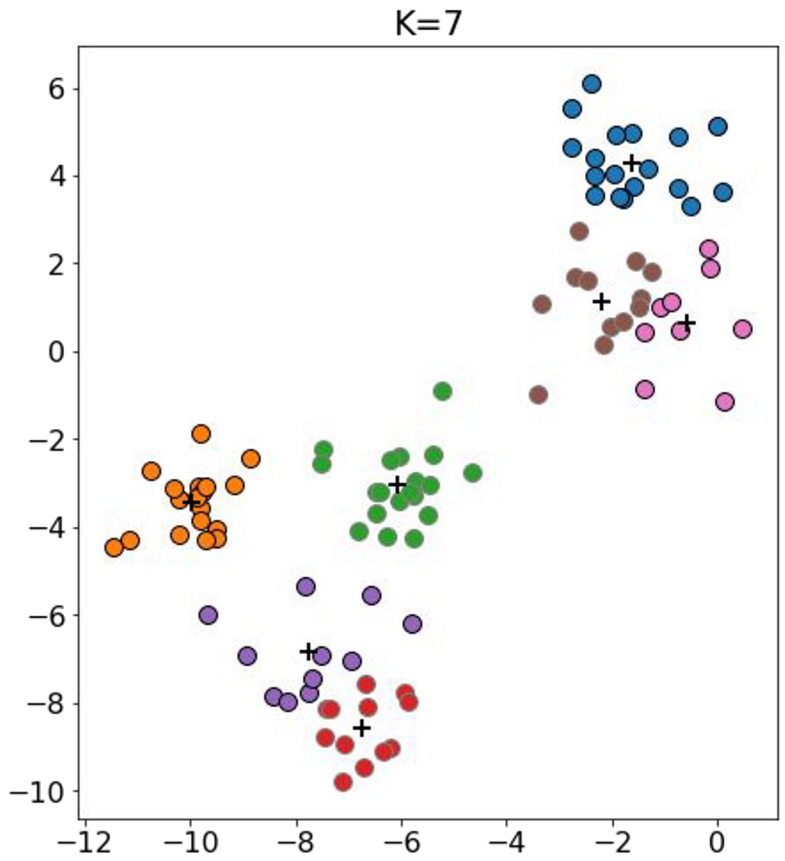
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

Visualization

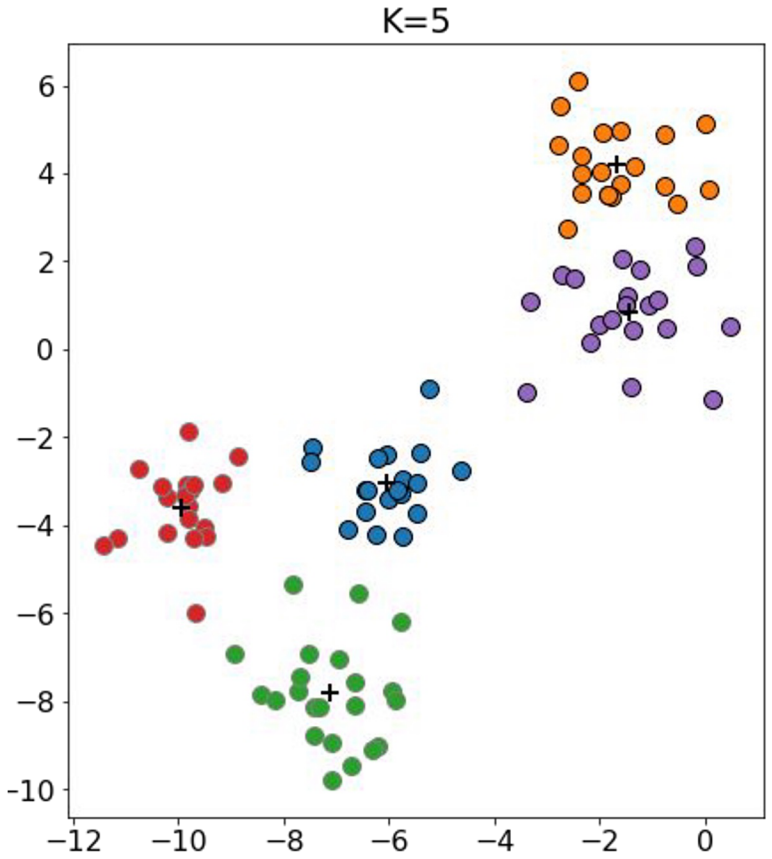
Selecting the best K







Selecting the best K



Using simulated data for **5 clusters** in two-dimensional data, we will consider the following strategies for selecting K:

1. Elbow method applied to mean squared error
2. Maximizing the Bayesian Information Criterion

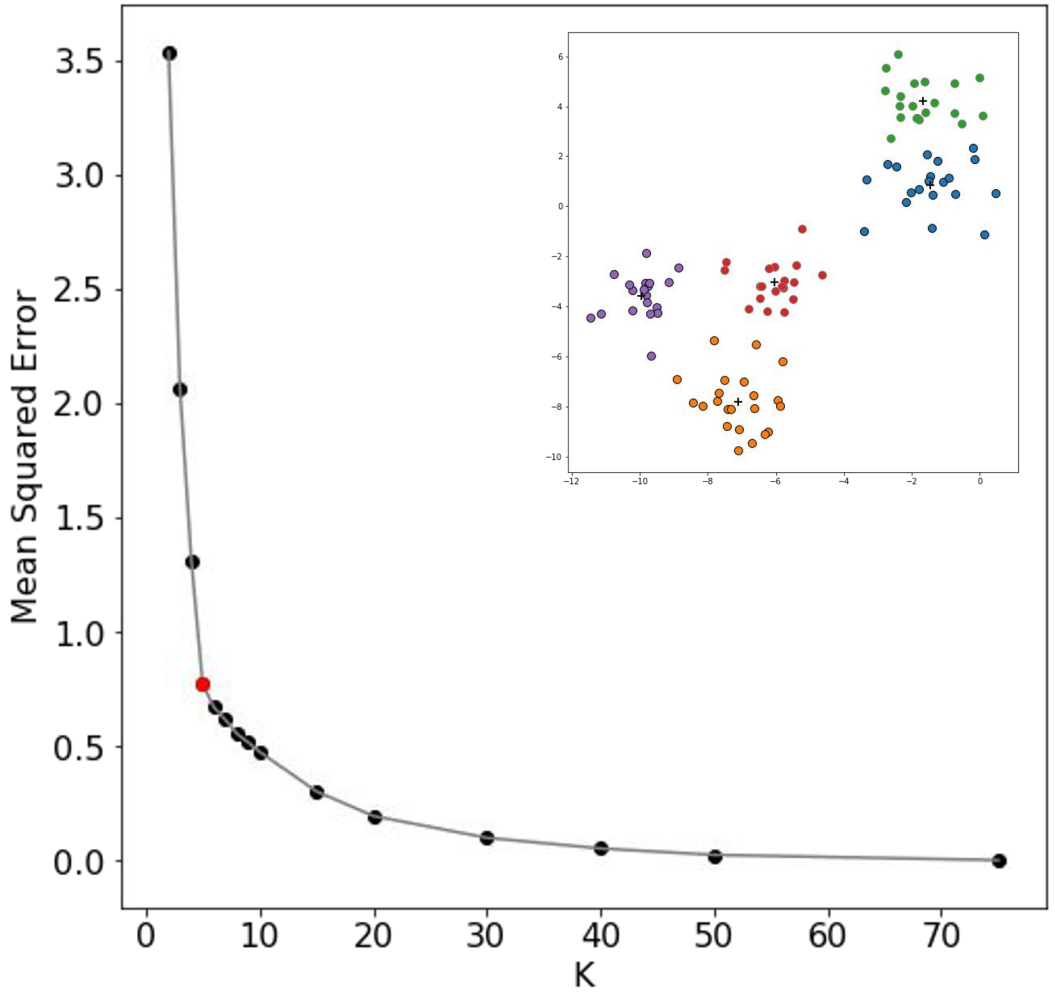
Elbow Method - Mean Squared Error

Fit model for a variety of different values of K.

Calculate mean squared error for each cluster

$$MSE = \frac{\Sigma(x_{ij} - \mu_j)^2}{n}$$

Where x_{ij} is the coordinate for point i, μ_j is the centroid for cluster j, and n is the total number of points.

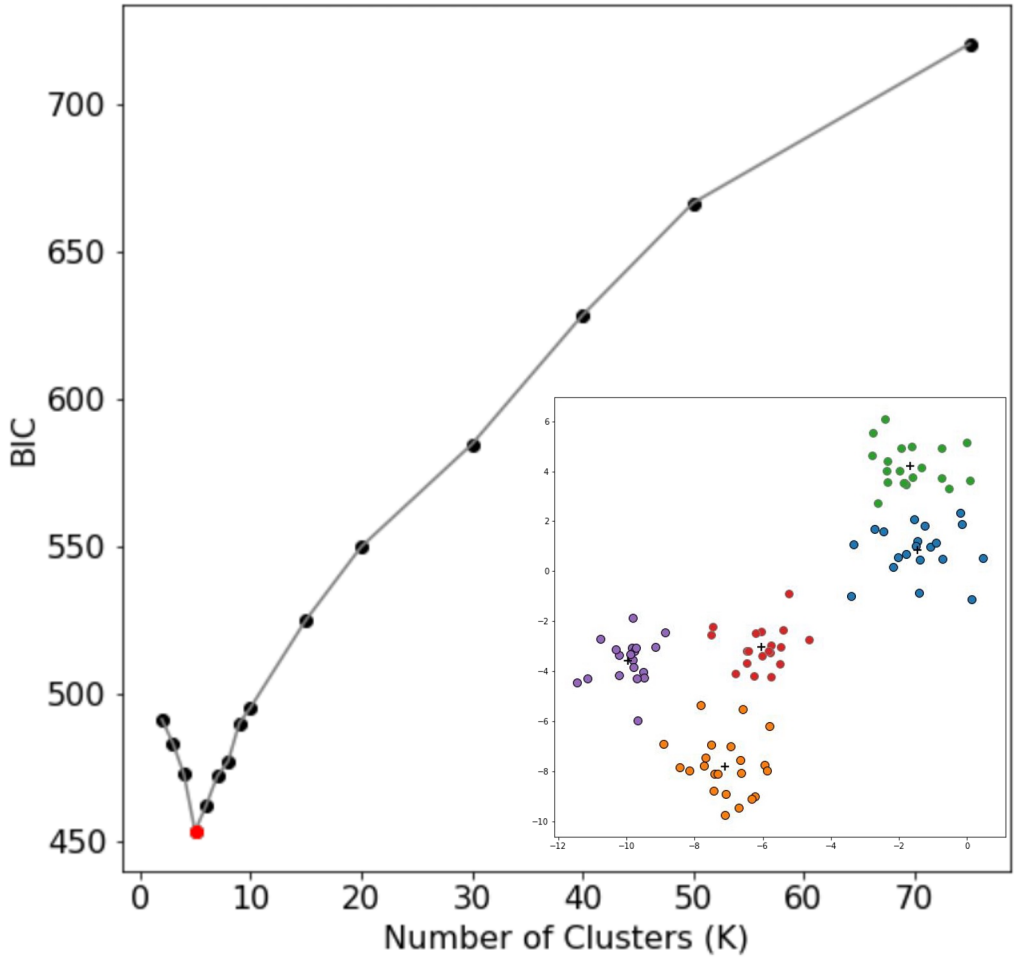


Maximizing Bayesian Information Criterion (BIC)

K means clustering assumes **spherical, clusters**. If we assume clusters are Gaussian, we can derive a likelihood function.

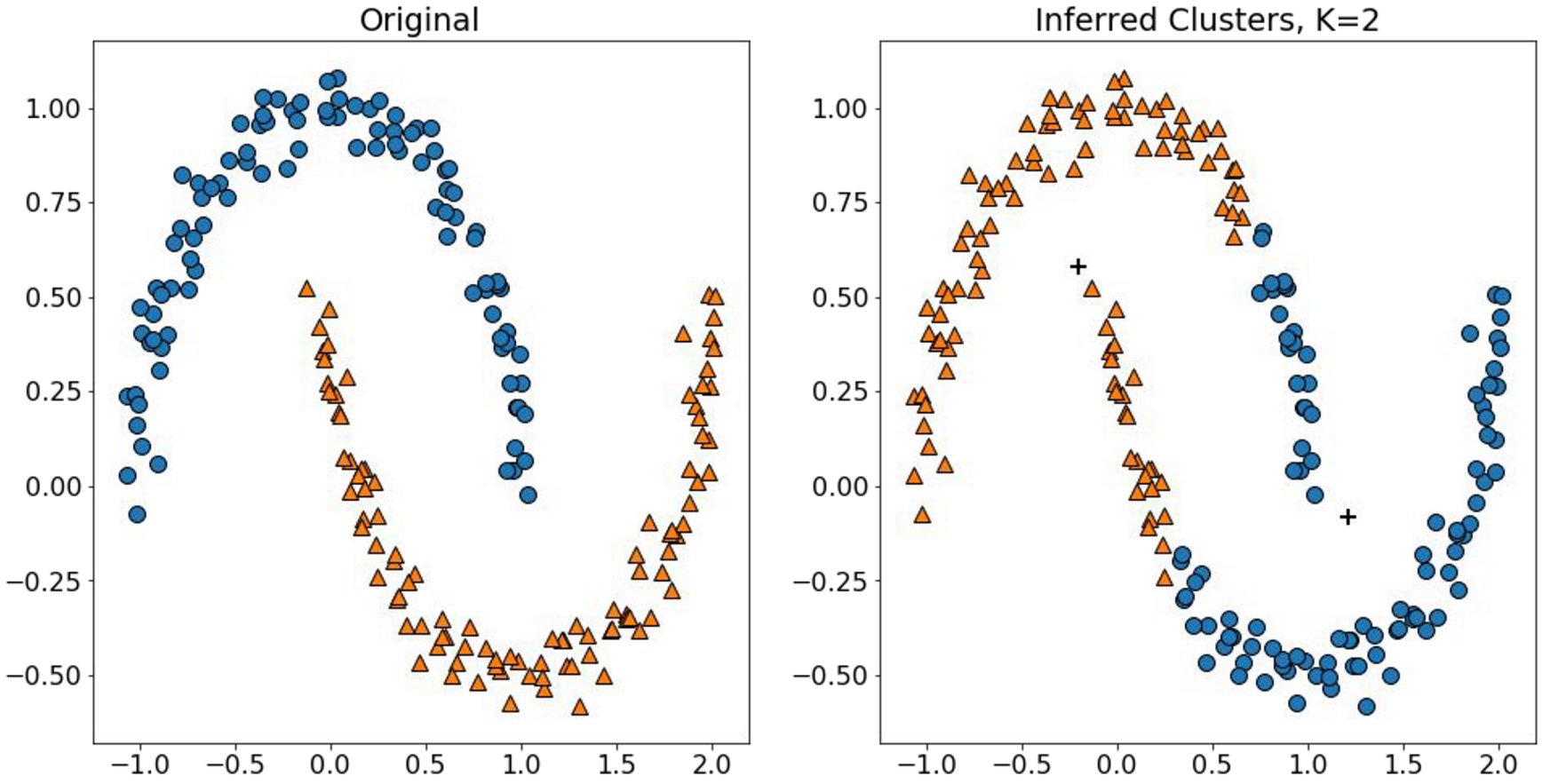
$$\mathrm{BIC} = \ln(n)k - 2\ln(\widehat{L})$$

BIC works best when you have many more observations than clusters.



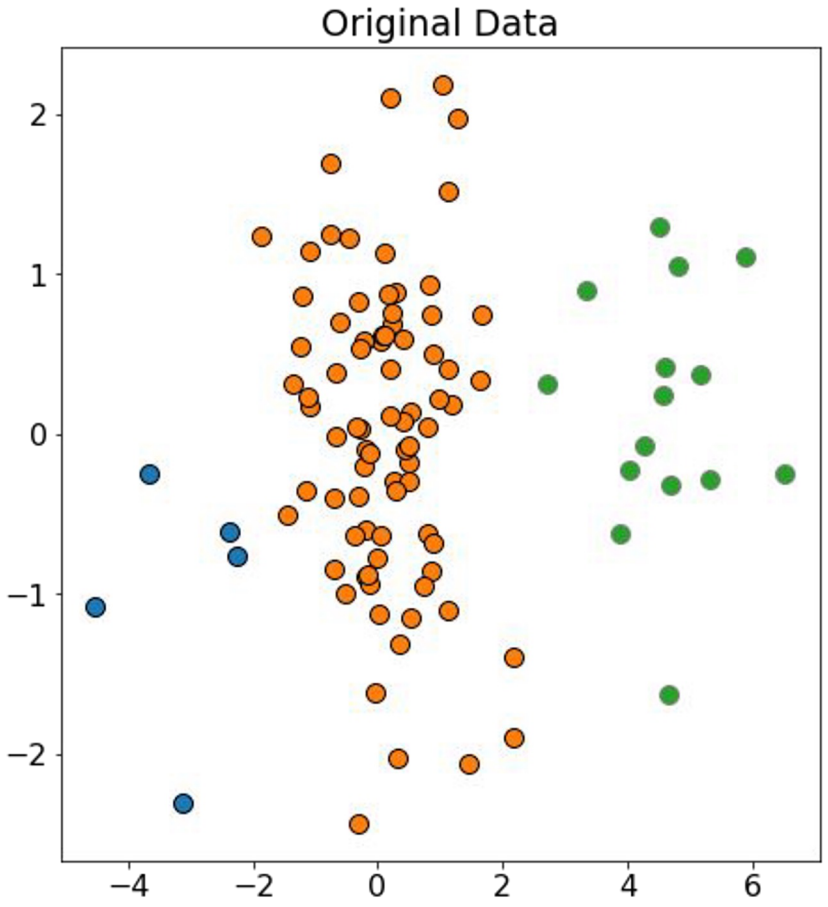
Impact of Cluster Shape

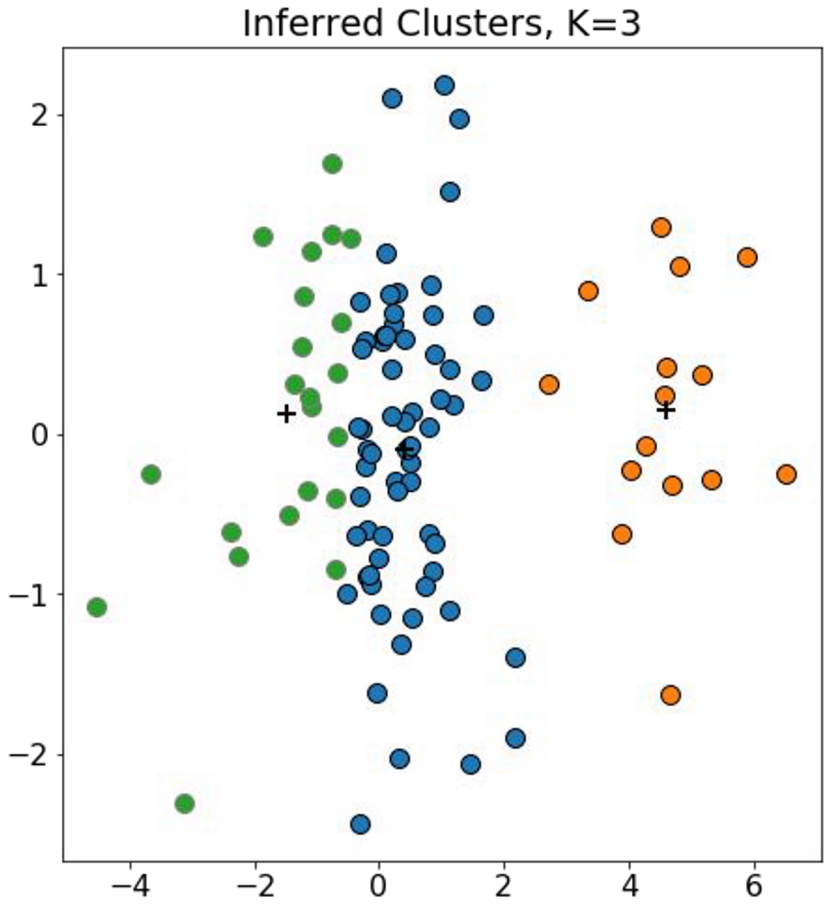
K-Means error function assumes spherical clusters and does not perform well on other shapes.



Impact of Cluster Size, Density

K-Means favors splitting large and/or sparse clusters to minimize error





Initial Centroid Selection

It is very unlikely that there will be one initial centroid selected within each cluster K

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K! n^K}{(Kn)^K} = \frac{K!}{K^K}$$

If K = 10, P = $10!/10^{10} = 0.00036$

It may take many iterations to converge on cluster centroids, and convergence is not guaranteed.

Expectation-Maximization (EM)

Generic algorithm applied broadly in machine learning Two iterative steps:

1. Expectation: Set expectation
2. Maximization: Maximize the fitness function

Application to K-Means:

1. Expectation: Assign points to nearest cluster center
2. Maximization: Update cluster centers

Data Clusters

Hierarchical K-Means Clustering

Hierarchical Clustering

- Agglomerative Clustering (bottom-up)
  - Start with points as individual clusters
  - At each step, merge the closest pair of clusters until only k clusters remain
- Divisive (top-down)
  - Start with a single, all-inclusive cluster
  - At each step, split clusters until there are k clusters

Data Clusters

Agglomerative Clustering

Agglomerative Clustering

Basic Algorithm

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** K clusters remain

Implementations differ in their computation of the proximity matrix

Methods for calculating inter-cluster similarity (proximity)

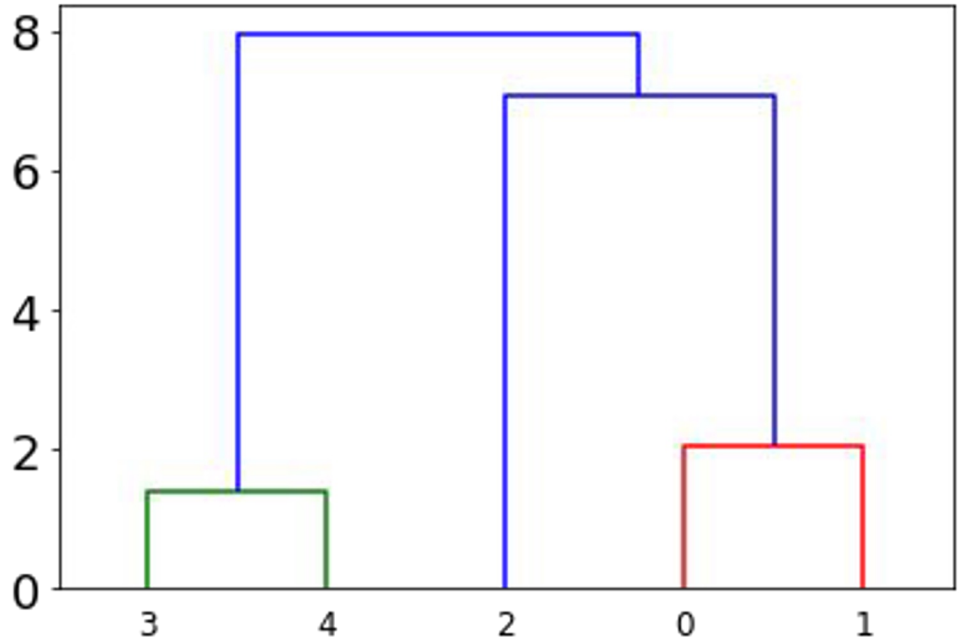
- MIN (Single Link)
- MAX (Complete Linkage)
- Group Average
- Ward’s Method

Cluster Similarity: MIN (Single Link)

The minimum distance (maximum proximity) between any two points in a pair of clusters defines cluster proximity.

Euclidean Distance Matrix

	0	1	2	3	4
0	0.0	2.0	7.1	8.0	9.3
1	2.0	0.0	9.1	8.4	9.8
2	7.1	9.1	0.0	9.9	10.9
3	8.0	8.4	9.9	0.0	1.4
4	9.3	9.8	10.9	1.4	0.0

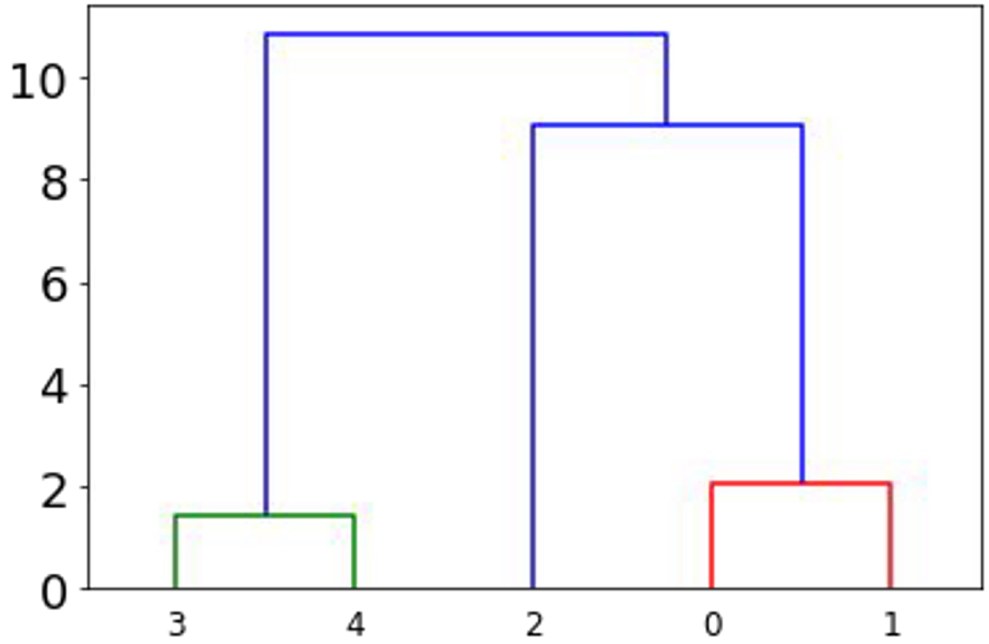


Cluster Similarity: MAX (Complete Linkage)

The maximum distance (minimum proximity) between any two points in a pair of clusters defines cluster proximity.

Euclidean Distance Matrix

	0	1	2	3	4
0	0.0	2.0	7.1	8.0	9.3
1	2.0	0.0	9.1	8.4	9.8
2	7.1	9.1	0.0	9.9	10.9
3	8.0	8.4	9.9	0.0	1.4
4	9.3	9.8	10.9	1.4	0.0



Cluster Similarity: Group Average

Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

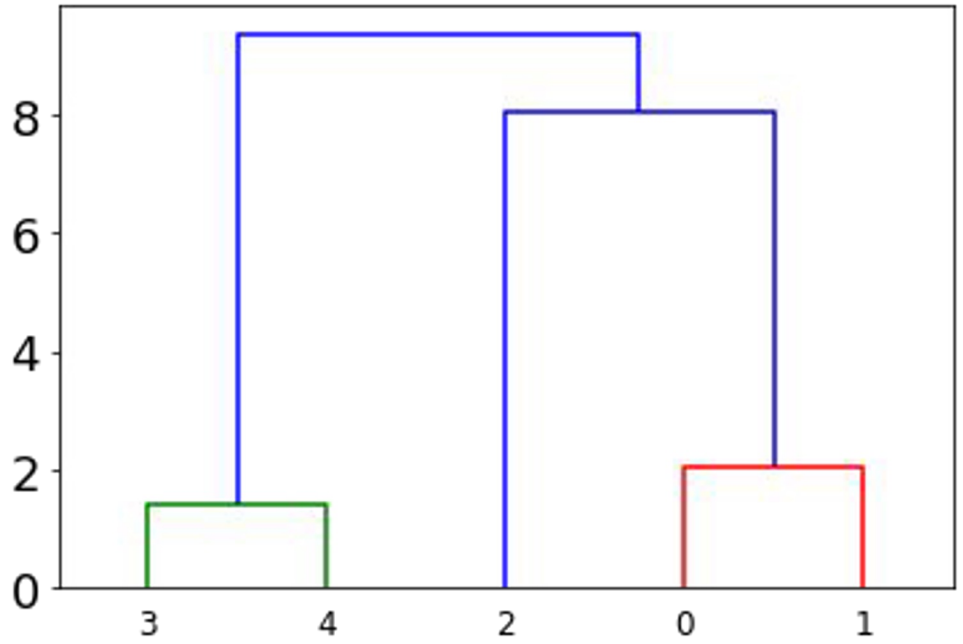
Cluster Similarity: Group Average

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

Initial

Euclidean Distance Matrix

	0	1	2	3	4
0	0.0	2.0	7.1	8.0	9.3
1	2.0	0.0	9.1	8.4	9.8
2	7.1	9.1	0.0	9.9	10.9
3	8.0	8.4	9.9	0.0	1.4
4	9.3	9.8	10.9	1.4	0.0



Cluster Similarity: Ward's Method

Similarity of two clusters is based on the increase in squared error when two clusters are merged.

$$\Delta(A,B) = \sum_{i \in A \cup B} \|\vec{x}_i - \vec{m}_{A \cup B}\|^2 - \sum_{i \in A} \|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i \in B} \|\vec{x}_i - \vec{m}_B\|^2$$

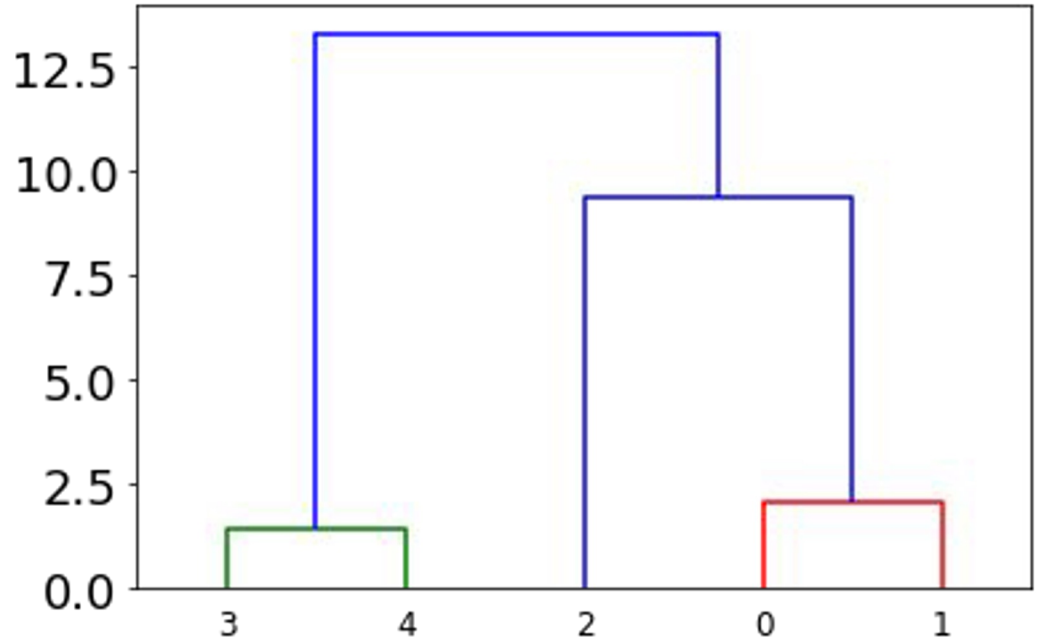
Hierarchical analog of K-Means clustering.

Similar to Group Average method where distance metric is Euclidean distance squared.

Cluster Similarity: Ward's Method

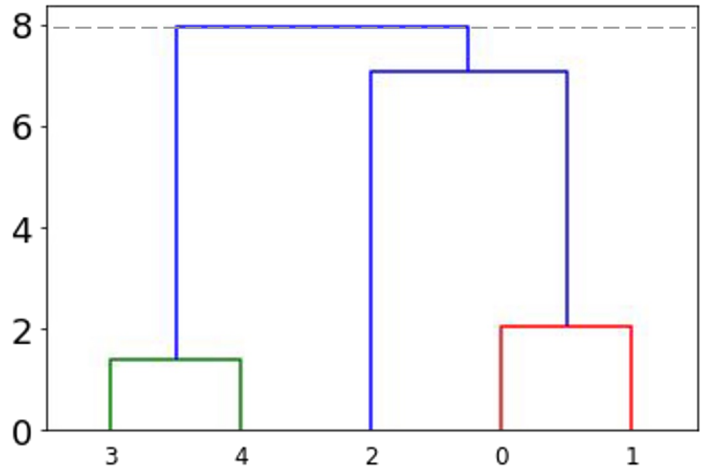
$$\Delta(A,B) = \sum_{i \in A \cup B} \|\vec{x}_i - \vec{m}_{A \cup B}\|^2 - \sum_{i \in A} \|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i \in B} \|\vec{x}_i - \vec{m}_B\|^2$$

	0	1	2	3	4
0	0.0	2.0	7.1	8.0	9.3
1	2.0	0.0	9.1	8.4	9.8
2	7.1	9.1	0.0	9.9	10.9
3	8.0	8.4	9.9	0.0	1.4
4	9.3	9.8	10.9	1.4	0.0

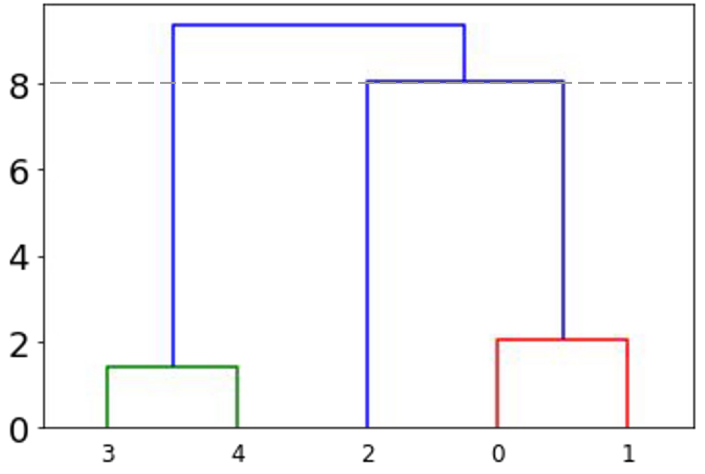


Hierarchical Clustering Summary

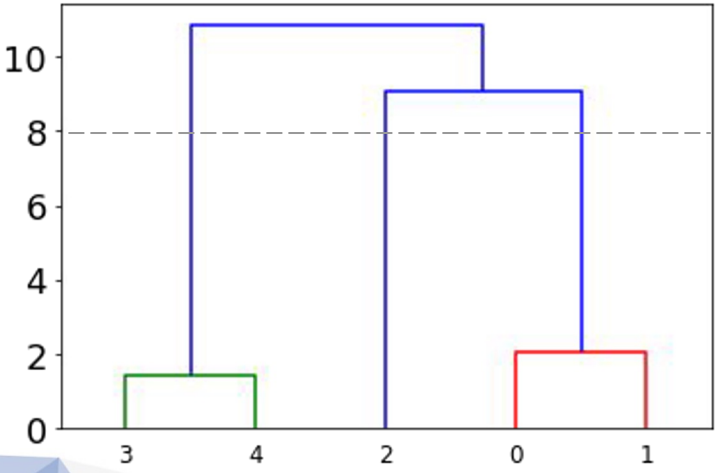
MIN



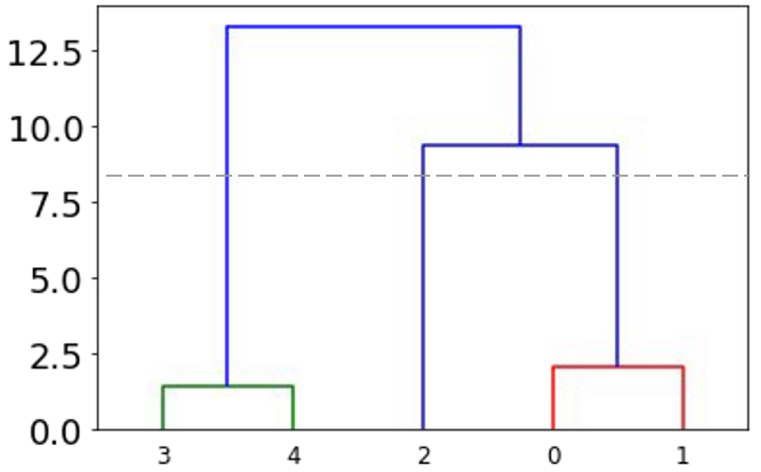
Group Average



MAX



Ward's Method



Limitations

All the hierarchical clustering methods discussed here are **deterministic**.

As the number of observations grows, the number of possible tree topologies increases dramatically. Consequently, these algorithms **scale poorly**.

Data Density

Overview

Learning data density

When should you think about data density?

- When you are looking for natural classifications, and want probabilistic rather than discrete classifications.
- When you would like to generate probability densities from
  which to sample/generate new data.

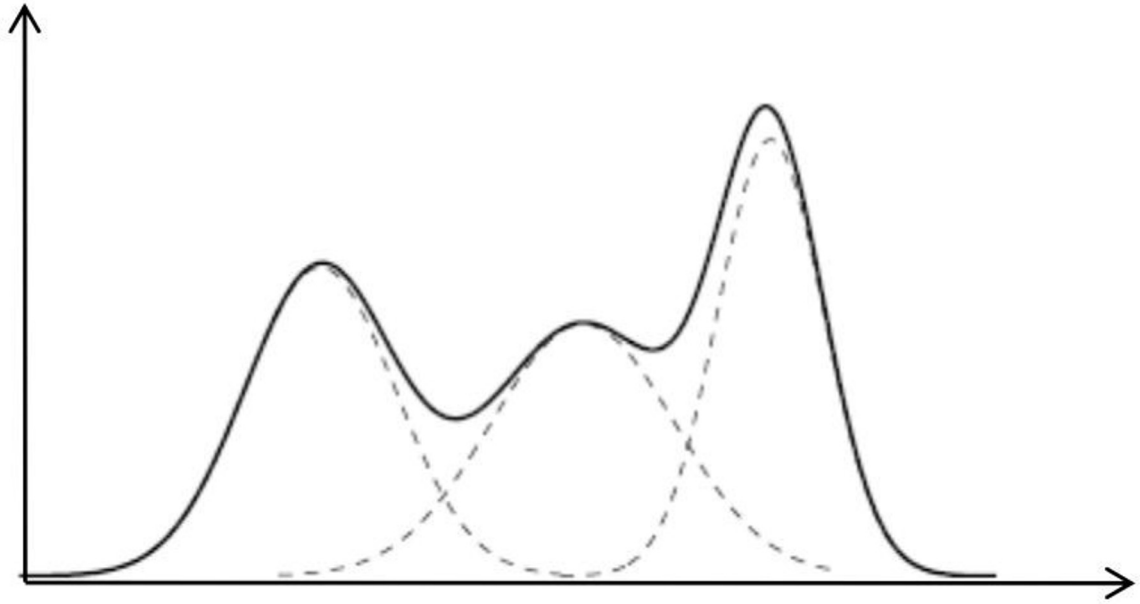
Data Density

Gaussian Mixture Models

Gaussian Mixture Models (GMM)

Consider that observations come from a mixture of K Gaussian distributions

- Given a set of input data, estimate distribution parameters that best describe that data
- Each cluster of the data contains observations drawn from the same Gaussian distribution



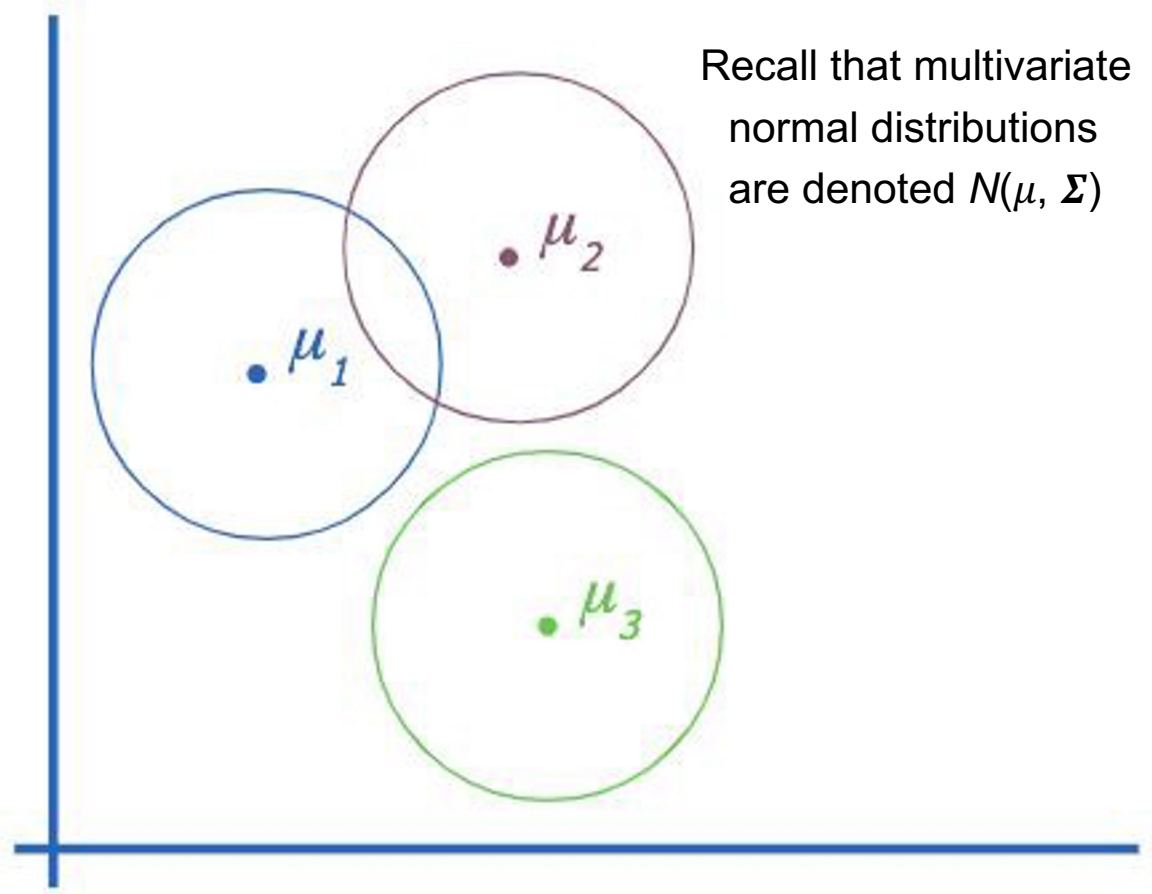
GMM Steps

Maximize log likelihood of data using Expectation Maximization:

1. Choose initial values for likelihood function parameters
2. Expectation: Calculate expected classes for each data point
3. Maximization: Update likelihood parameters (e.g. location and shape)

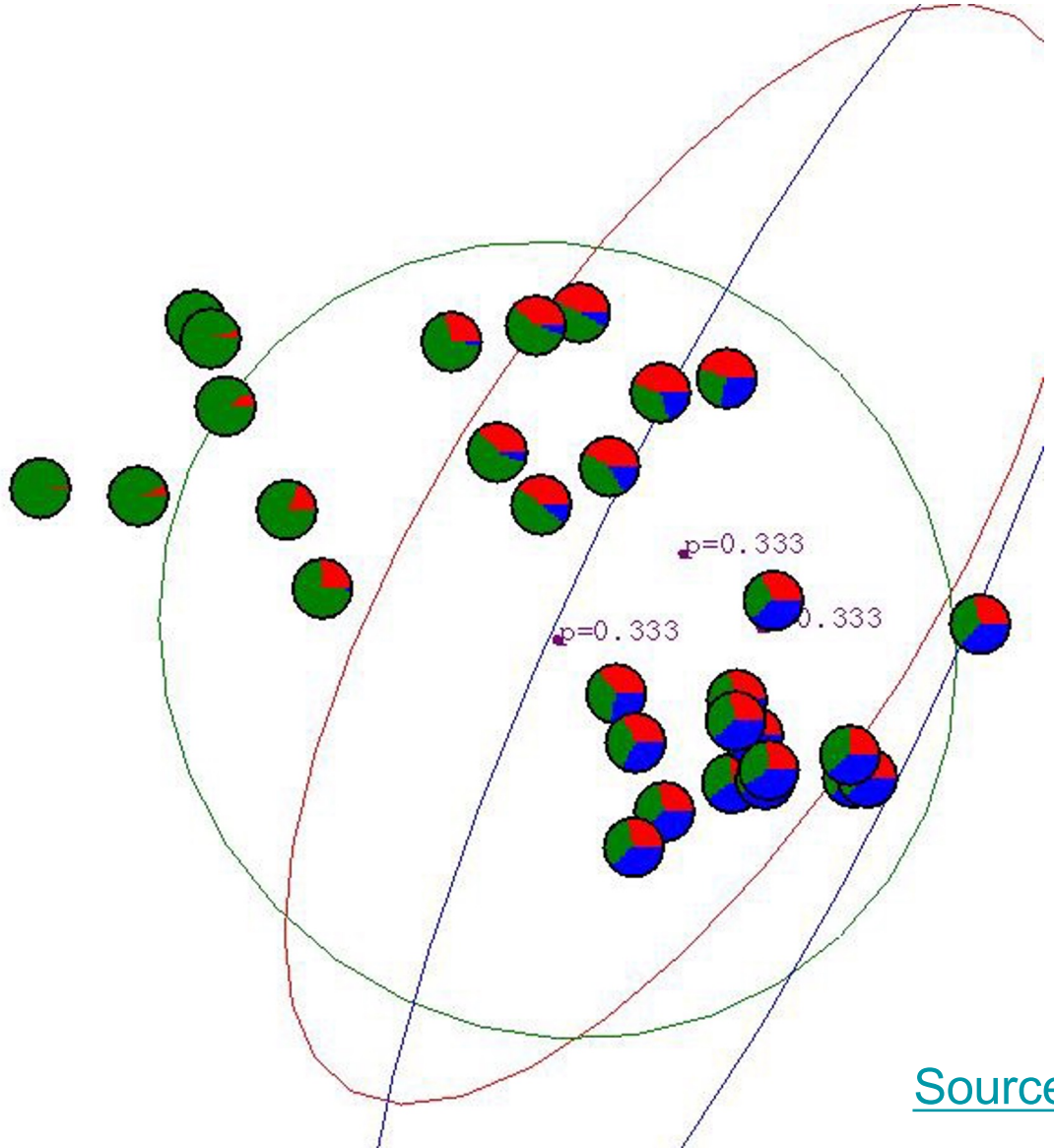
GMM Assumptions

1. There are k components, and the i^{th} component is ω_i
2. Component ω_i has an associated mean vector μ_i
3. Each component generates data from a Gaussian distribution with mean μ_i and covariance matrix $\sigma^2\boldsymbol{I}$



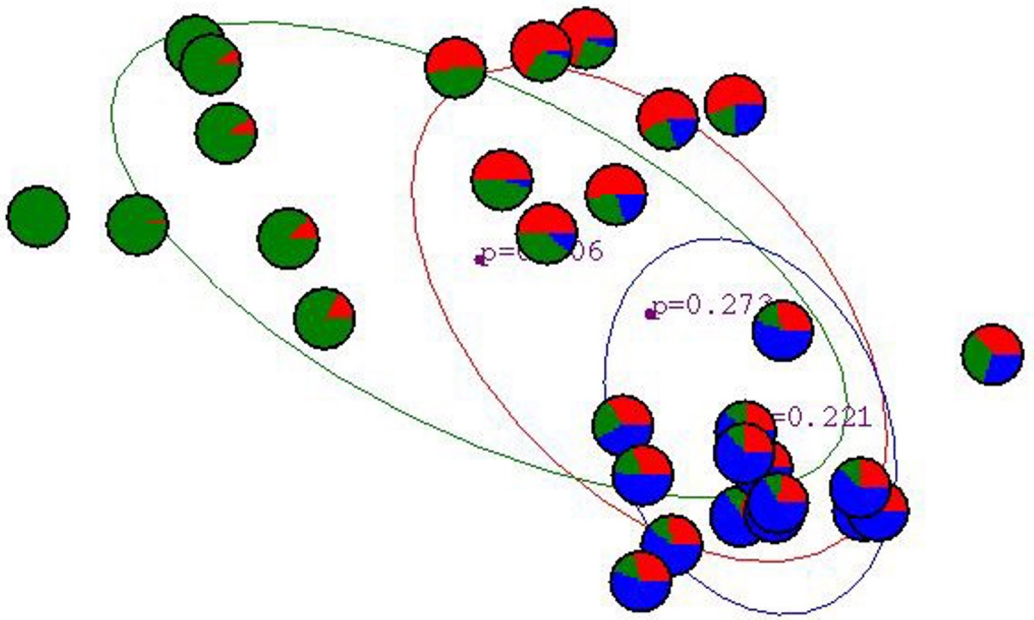
Recall that multivariate normal distributions are denoted $N(\mu, \boldsymbol{\Sigma})$

GMM Example: Initialization



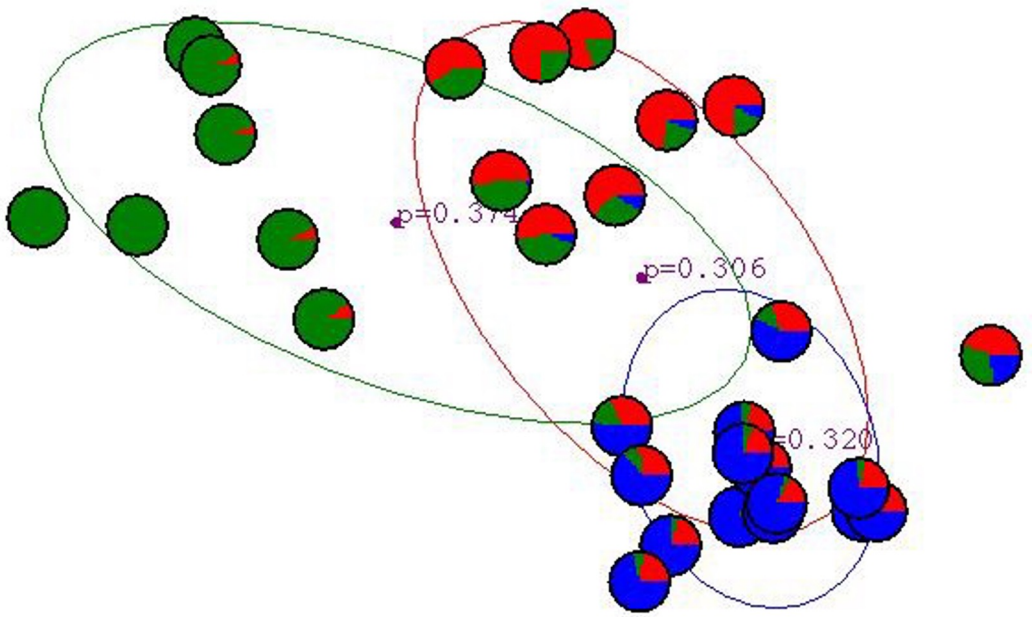
Source: Andrew Moore

GMM Example: After 1st Iteration



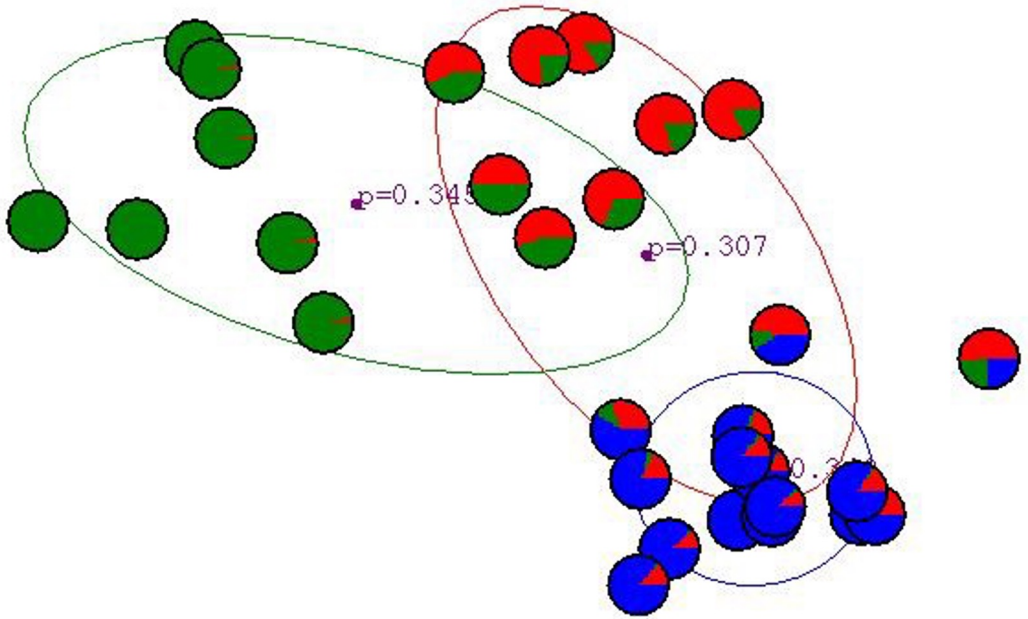
Source: Andrew Moore

GMM Example: After 2nd Iteration



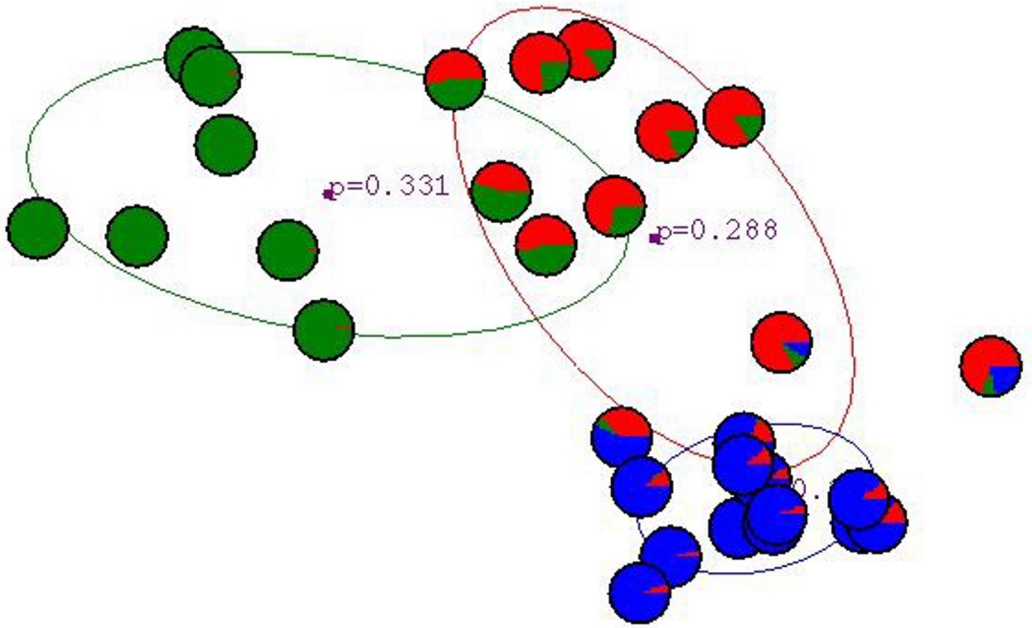
Source: Andrew Moore

GMM Example: After 3rd Iteration



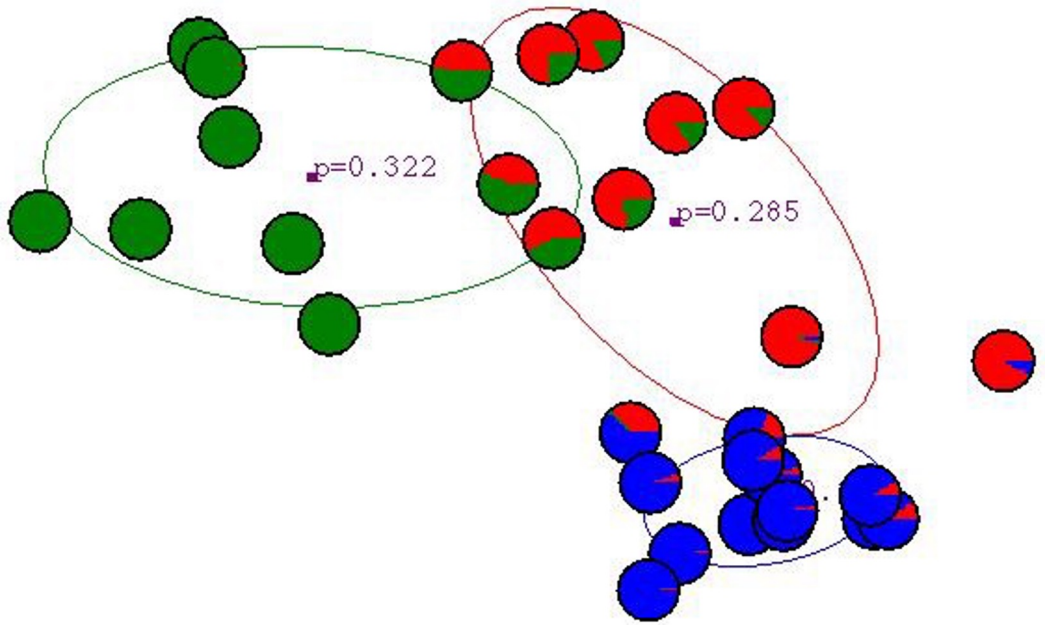
Source: Andrew Moore

GMM Example: After 4th Iteration



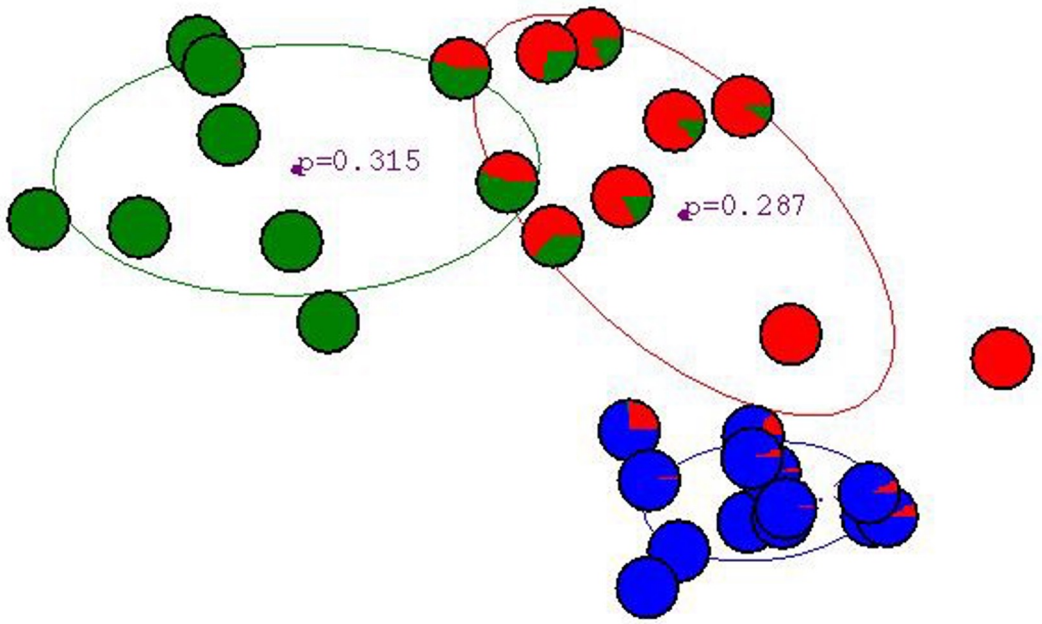
Source: Andrew Moore

GMM Example: After 5th Iteration



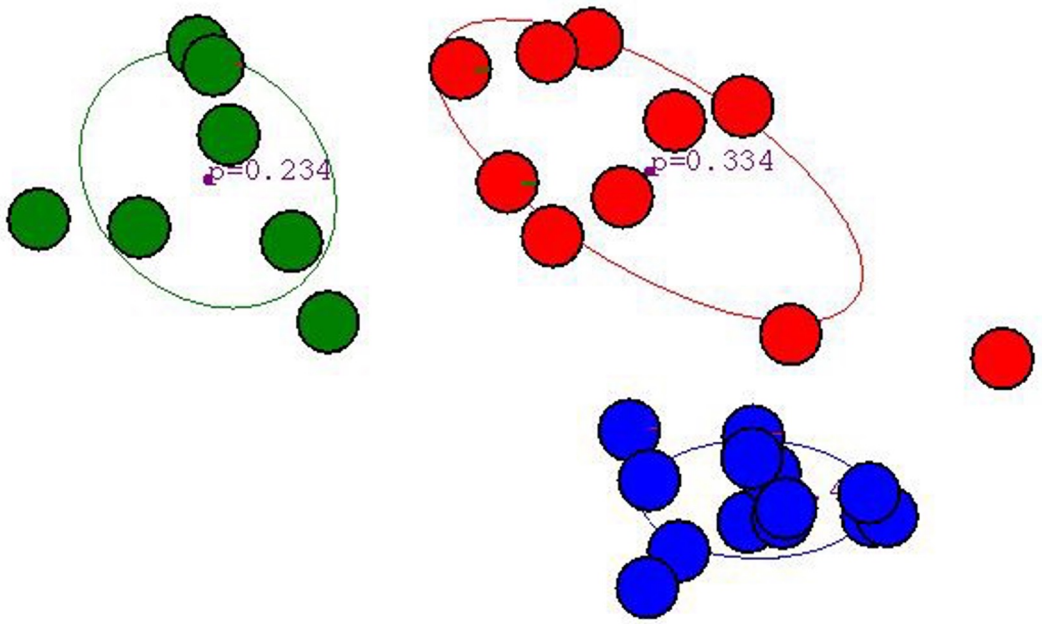
Source: Andrew Moore

GMM Example: After 6th Iteration



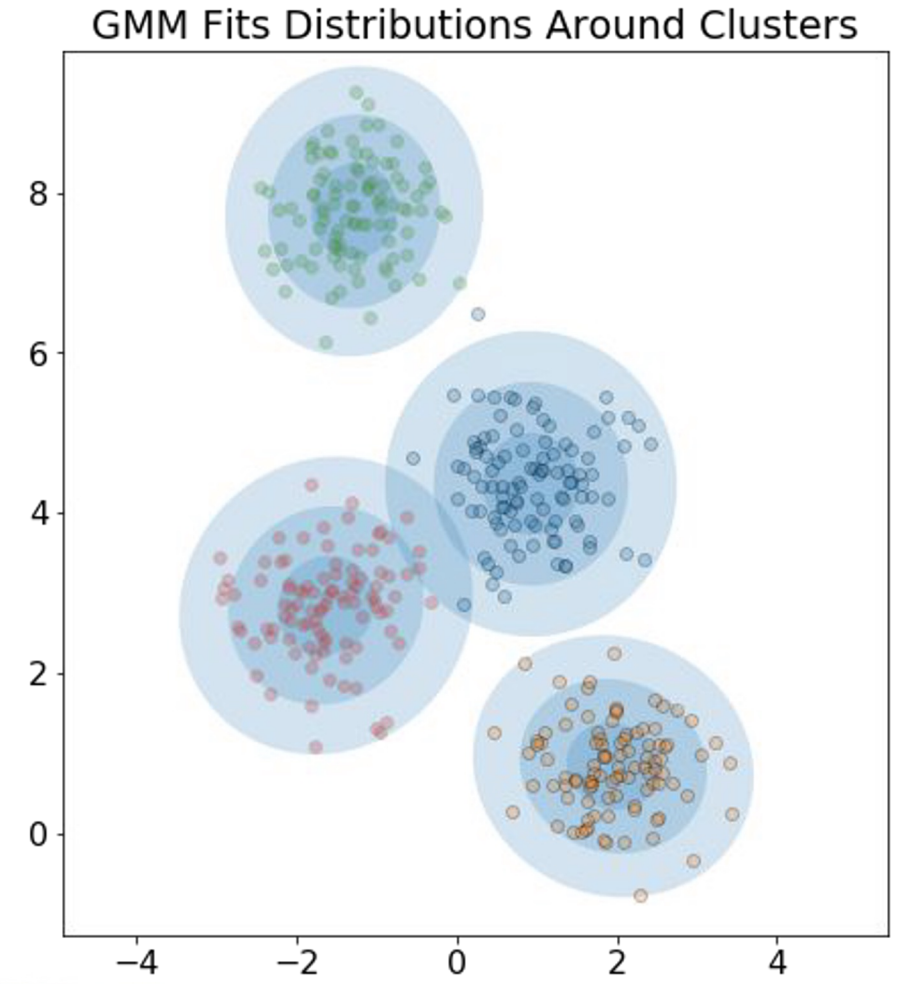
Source: Andrew Moore

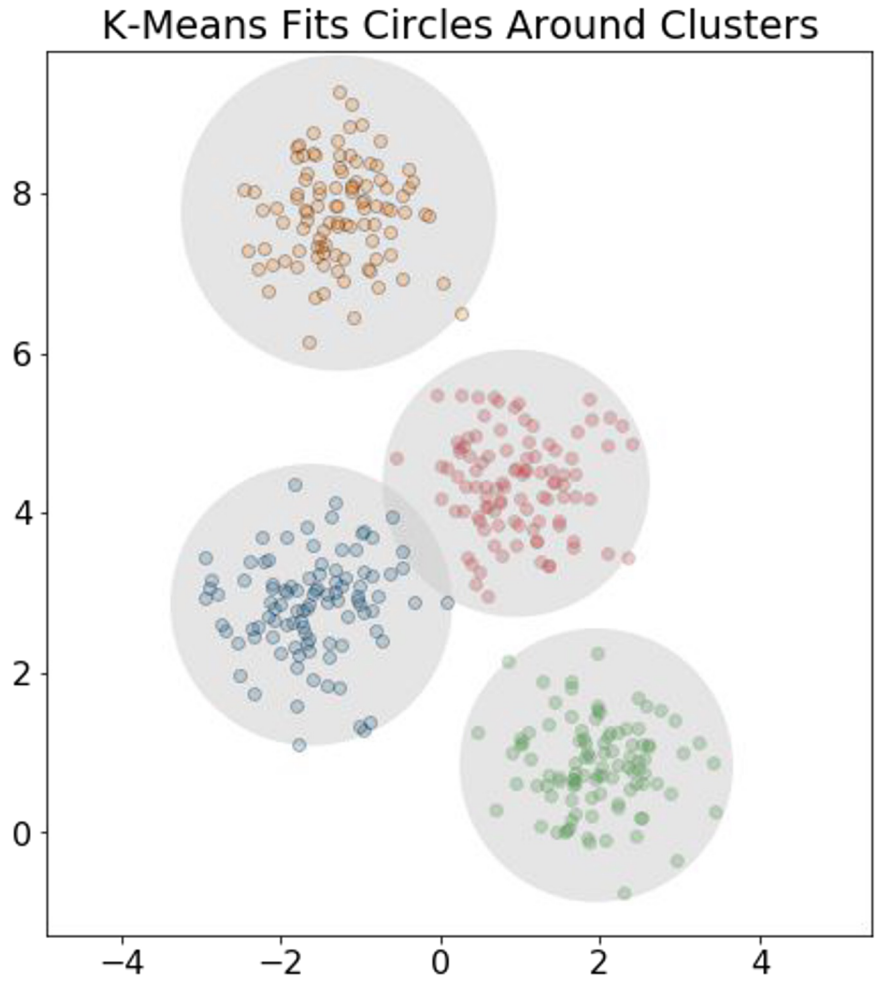
GMM Example: After 20th Iteration



Source: Andrew Moore

GMM vs K-Means for Clustering





GMM Covariance Matrix Influences Cluster Shape

Spherical Covariance Matrix

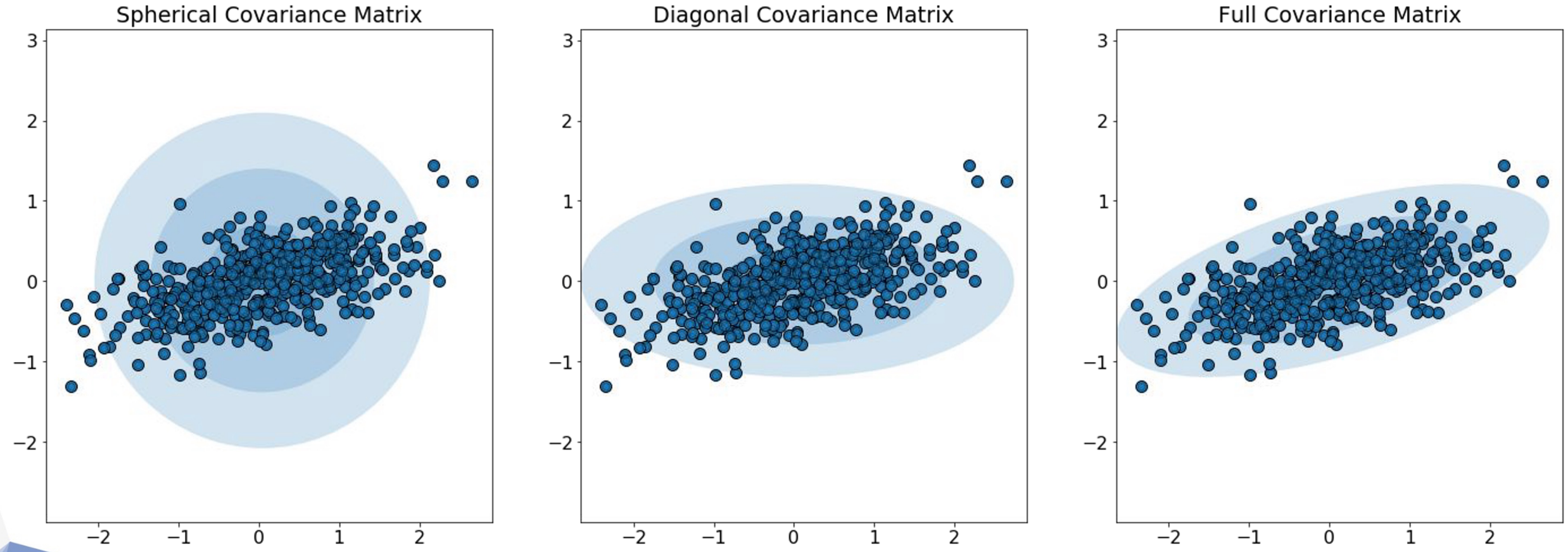
$$\Sigma_j = diag(\sigma_j^2, \sigma_j^2, \ldots, \sigma_j^2) = \sigma_j^2 I$$

Diagonal Covariance Matrix

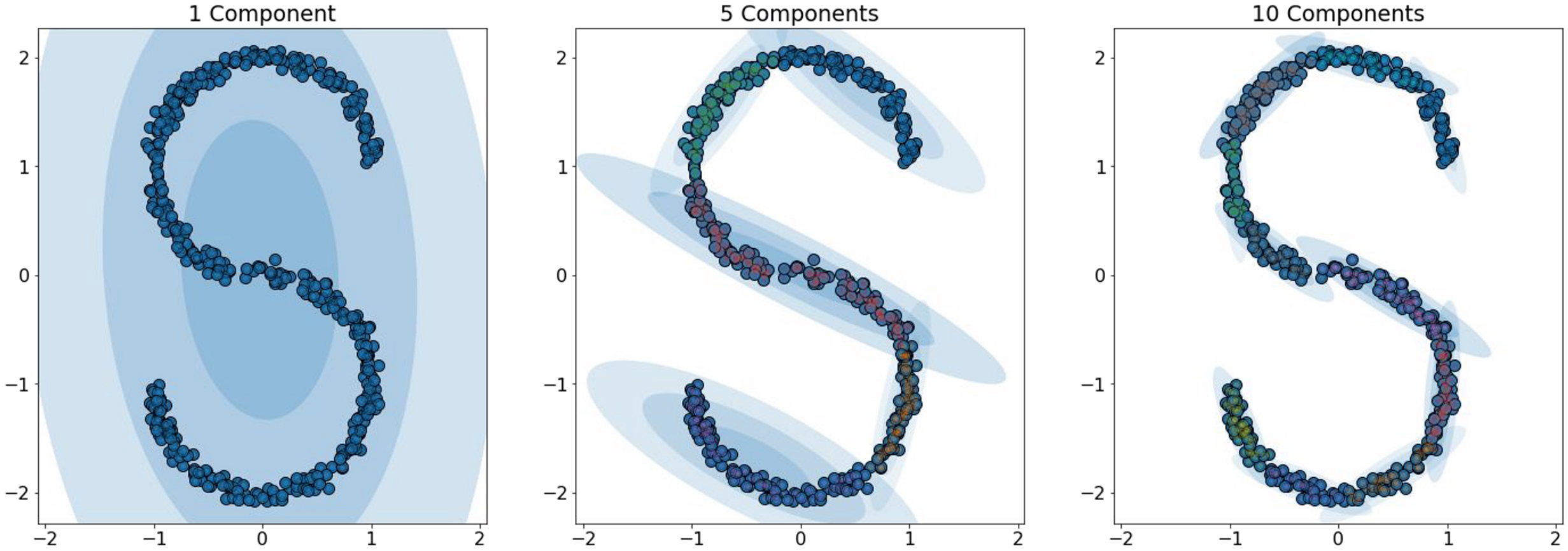
$$\Sigma_j = diag(\sigma_{j1}^2, \sigma_{j2}^2, \ldots, \sigma_{jd}^2)$$

Full Covariance Matrix

GMM Covariance Matrix Influences Cluster Shape



GMMs for Data Generation



Data Density

DBSCAN

DBSCAN: Density-Based Clustering

Clustering algorithm based on spatial relationships between observations, rather than relationship between observations and a cluster centroid.

Similar to GMM, DBSCAN can detect clusters of non-spherical shapes.

DBSCAN: Density-Based Clustering

ε is the radius that defines the neighborhood around a point ***p***

MinPts is the minimum number of neighbors ***p*** must have to be considered a core point.

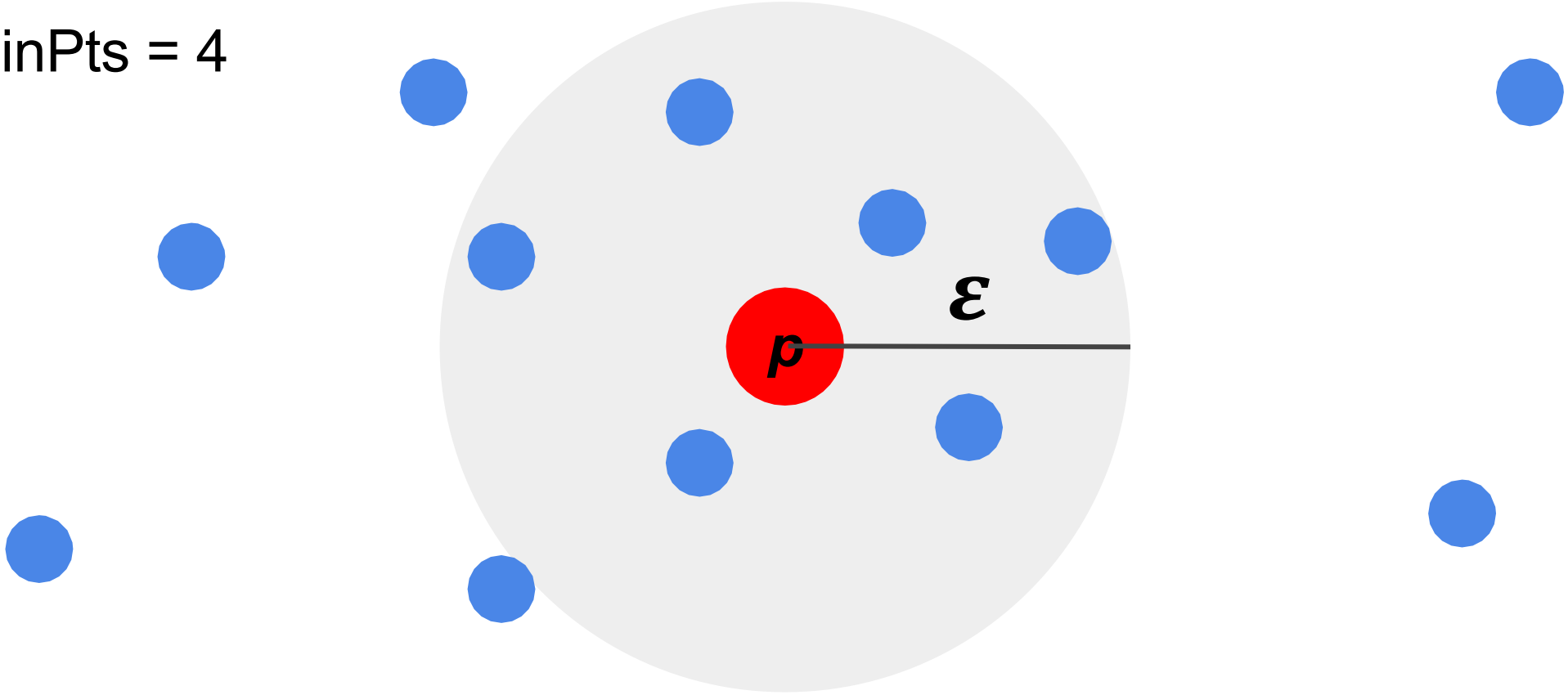
Core points are points that have >= **MinPts** neighbors within ε

Border points have < **MinPts** neighbors within ε but have a core point within ε

Noise points are all remaining points that are neither core nor border points.

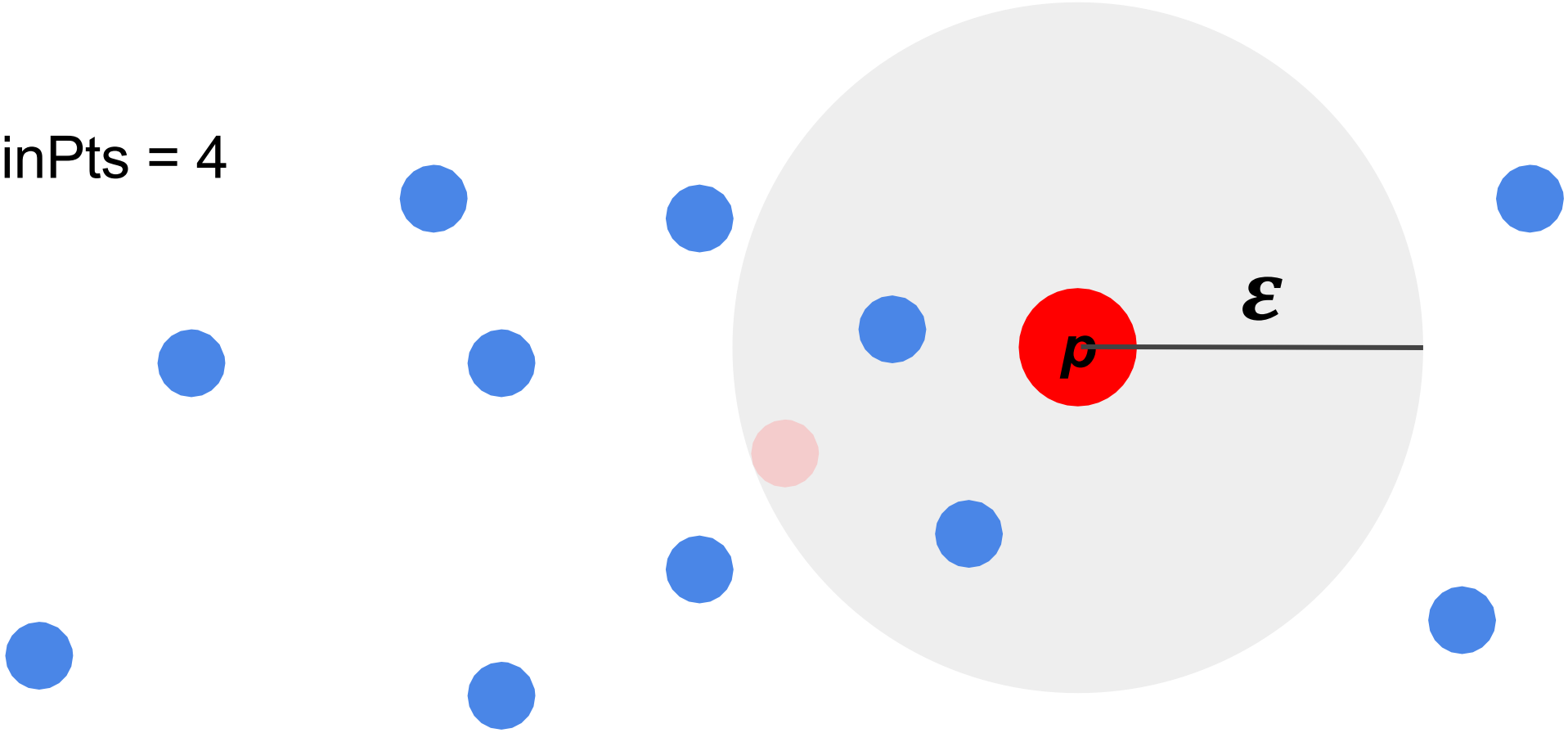
Core Point

minPts = 4



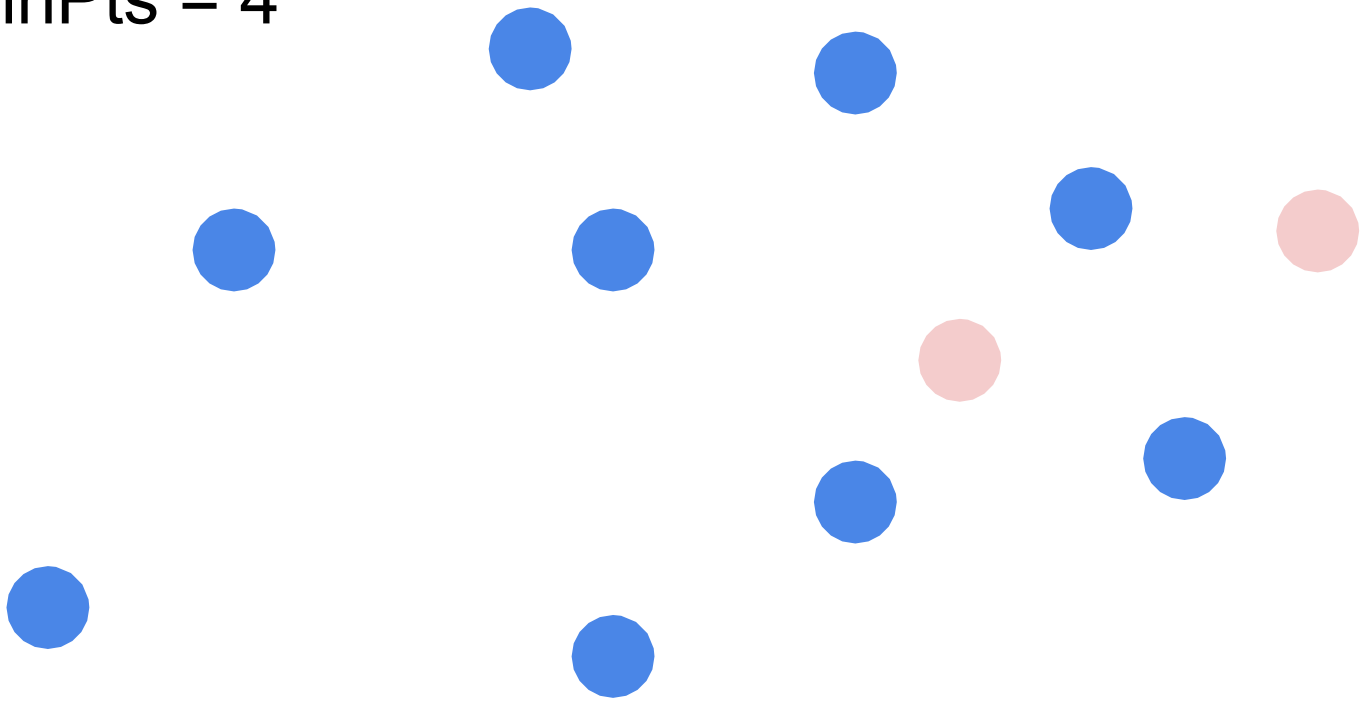
Border Point

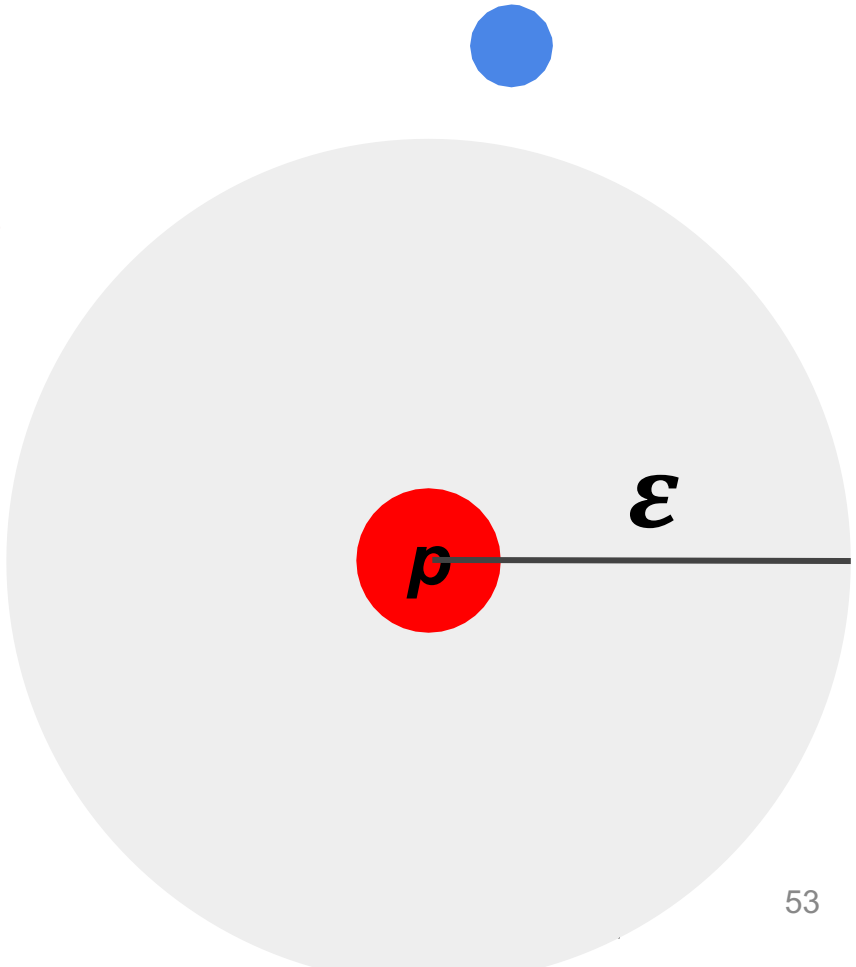
minPts = 4



Noise Point

minPts = 4

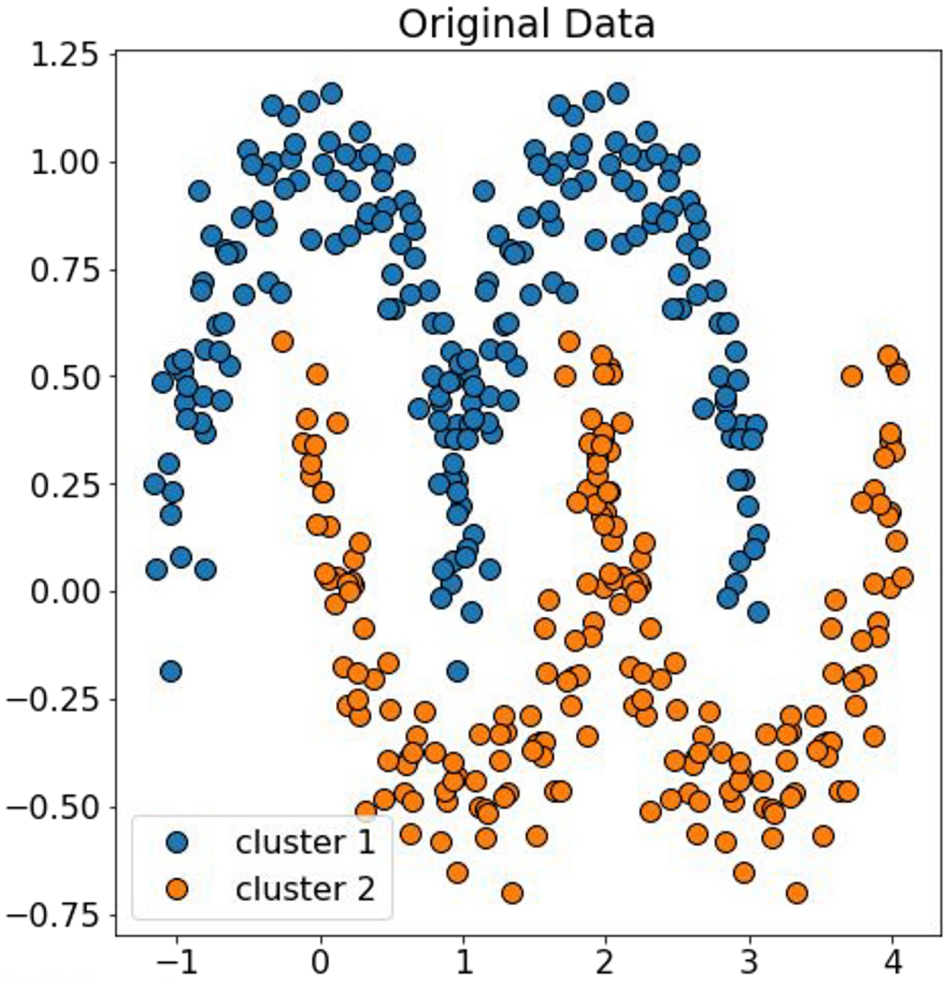


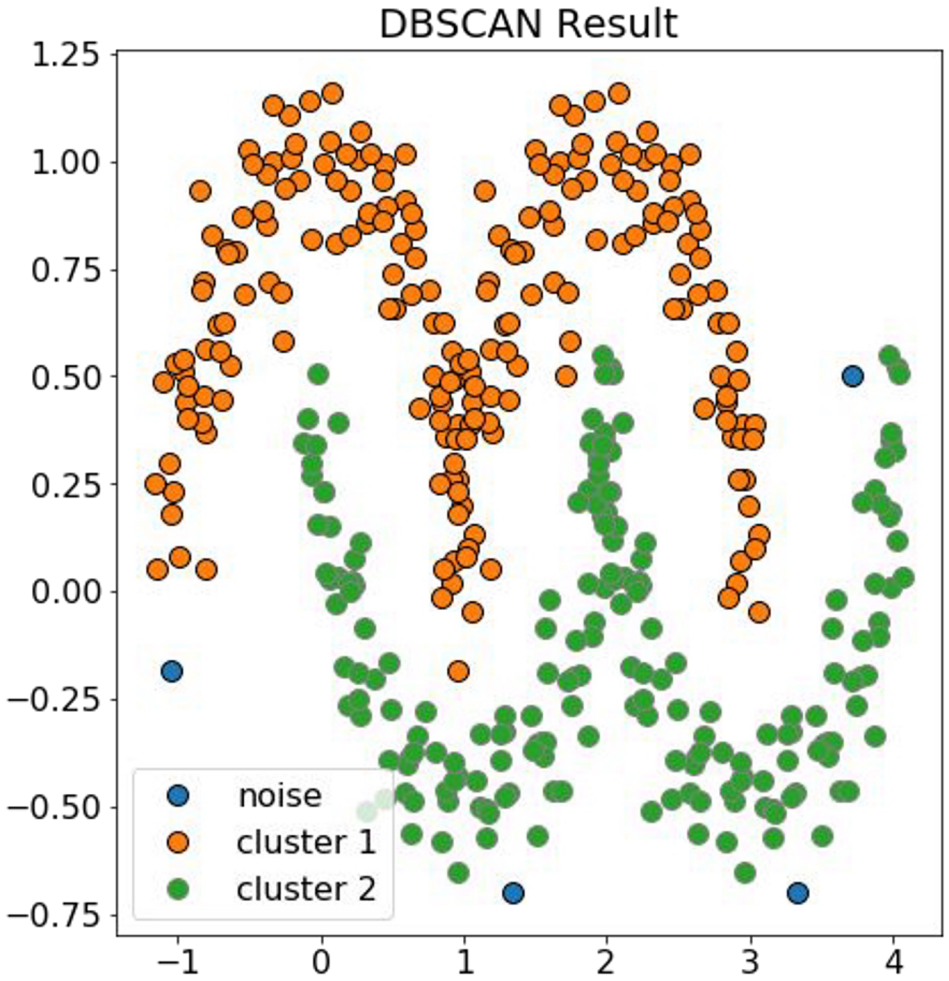


DBSCAN Algorithm

1. Eliminate noise points
2. *current_cluster_label* ← *1*
3. **For** all core points **do**
4. **If** core point has no cluster label **then**
5. *current_cluster_label* ← *current_cluster_label* + *1*
6. Label current core point with label *current_cluster_label*
7. **End if**
8. **For** all points in ε except the ith the point itself **do**
9. **If** the point does not have a cluster label **then**
10. Label the point with label *current_cluster_label*
11. **End if**
12. **End for**
13. **End for**

DBSCAN Example

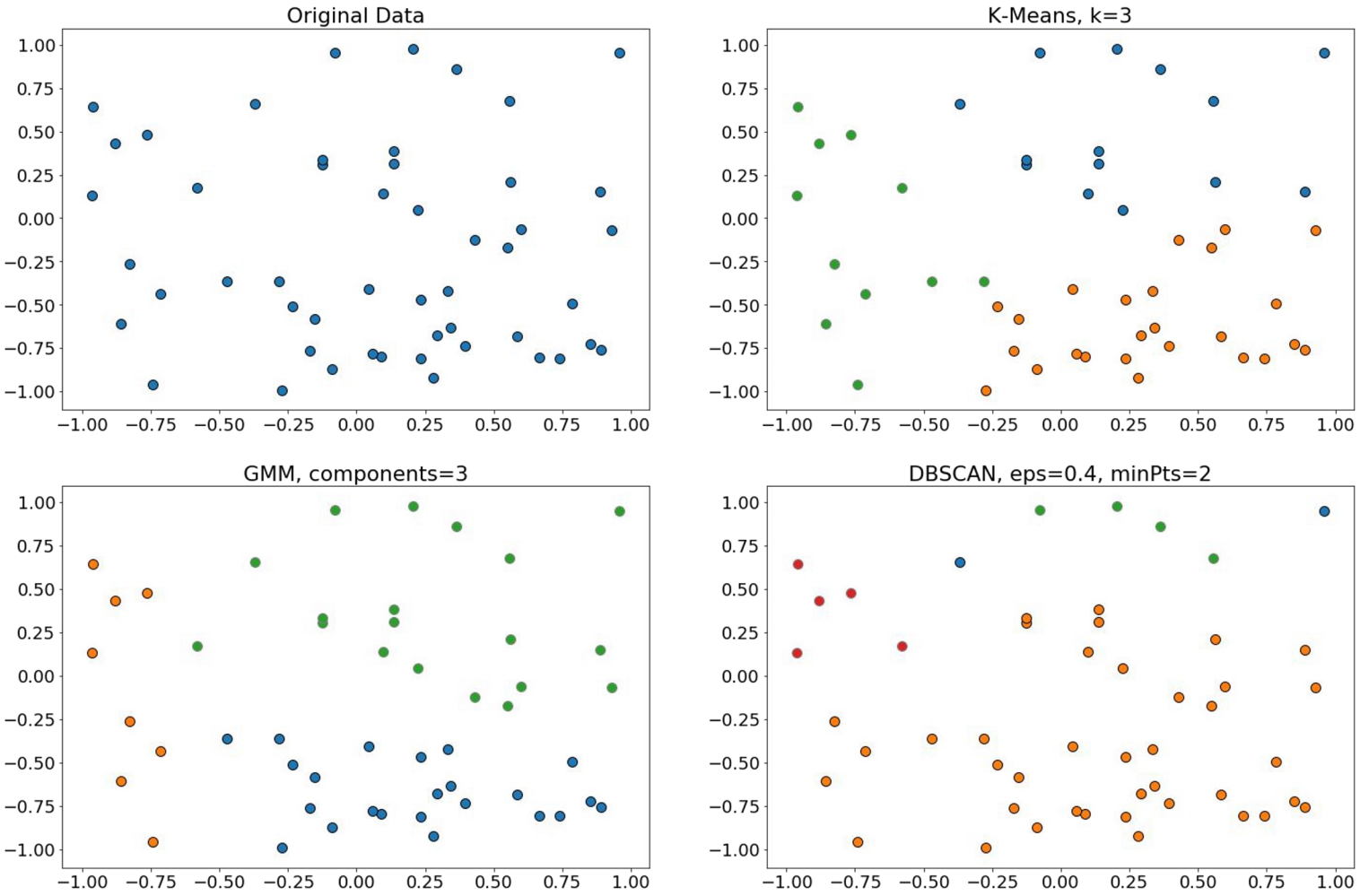




Unsupervised Learning

Summary

Detecting Structure in Randomness



Unsupervised Learning Summary

- Learn data **representation**, **clustering**, and/or **density**.
- Techniques discussed
  - Representation: Principal Components Analysis (PCA)
  - Clustering: K-Means, Agglomerative Hierarchical Clustering
  - Density: Gaussian Mixture Models, DBSCAN
- Caveats
  - Be wary of spurious patterns in unsupervised learning

More Resources

Interactive PCA demo at setosa.io

Comparison of Manifold Learning

Techniques Scikit-Learn Clustering

Examples Hierarchical Clustering Examples

Mathematical Intuition for Hierarchical

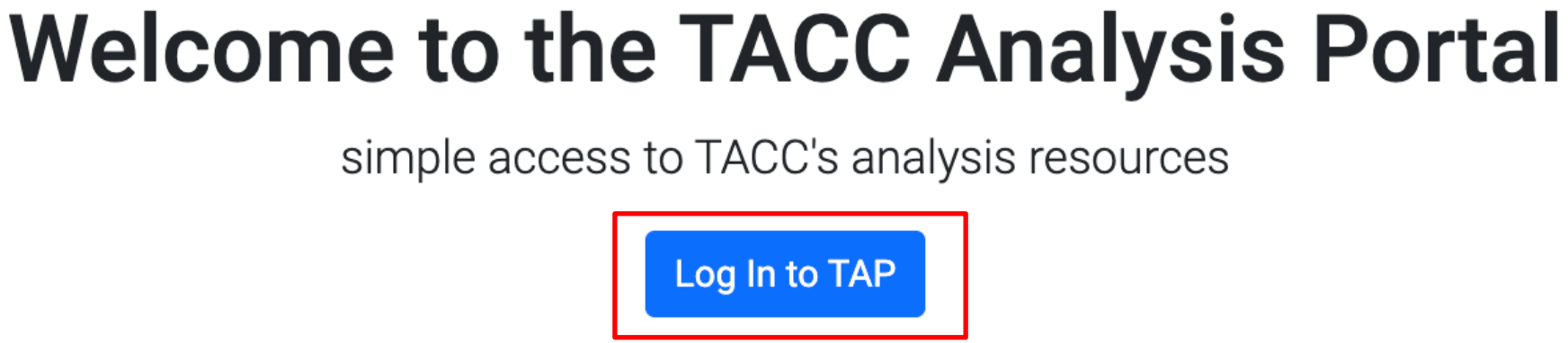
Clustering Gaussian Mixture Models in Python

Hands-On Exercise

Jupyter Set-Up and Data Representation

Start the TAP Session

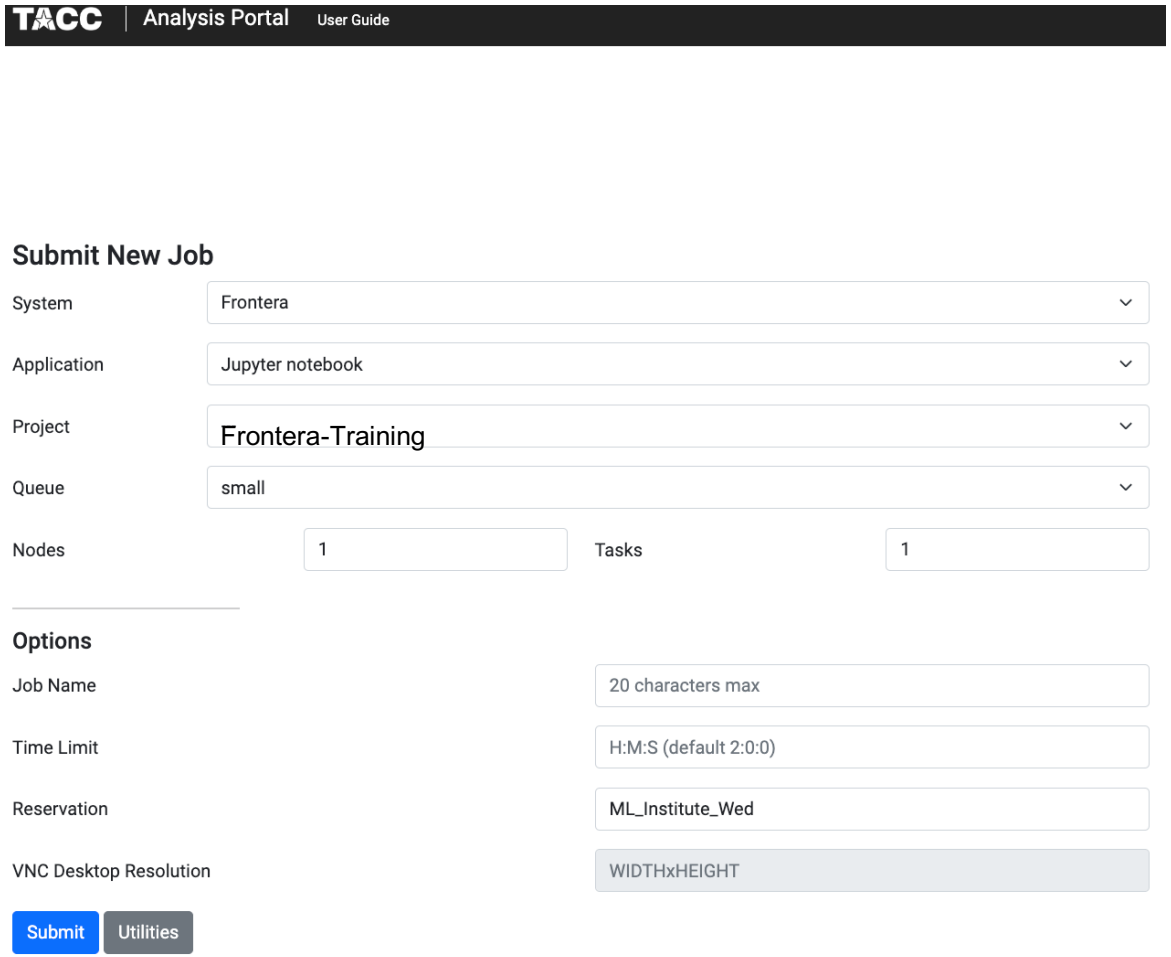
Log on to the Analysis Portal with your TACC user credentials
https://tap.tacc.utexas.edu/



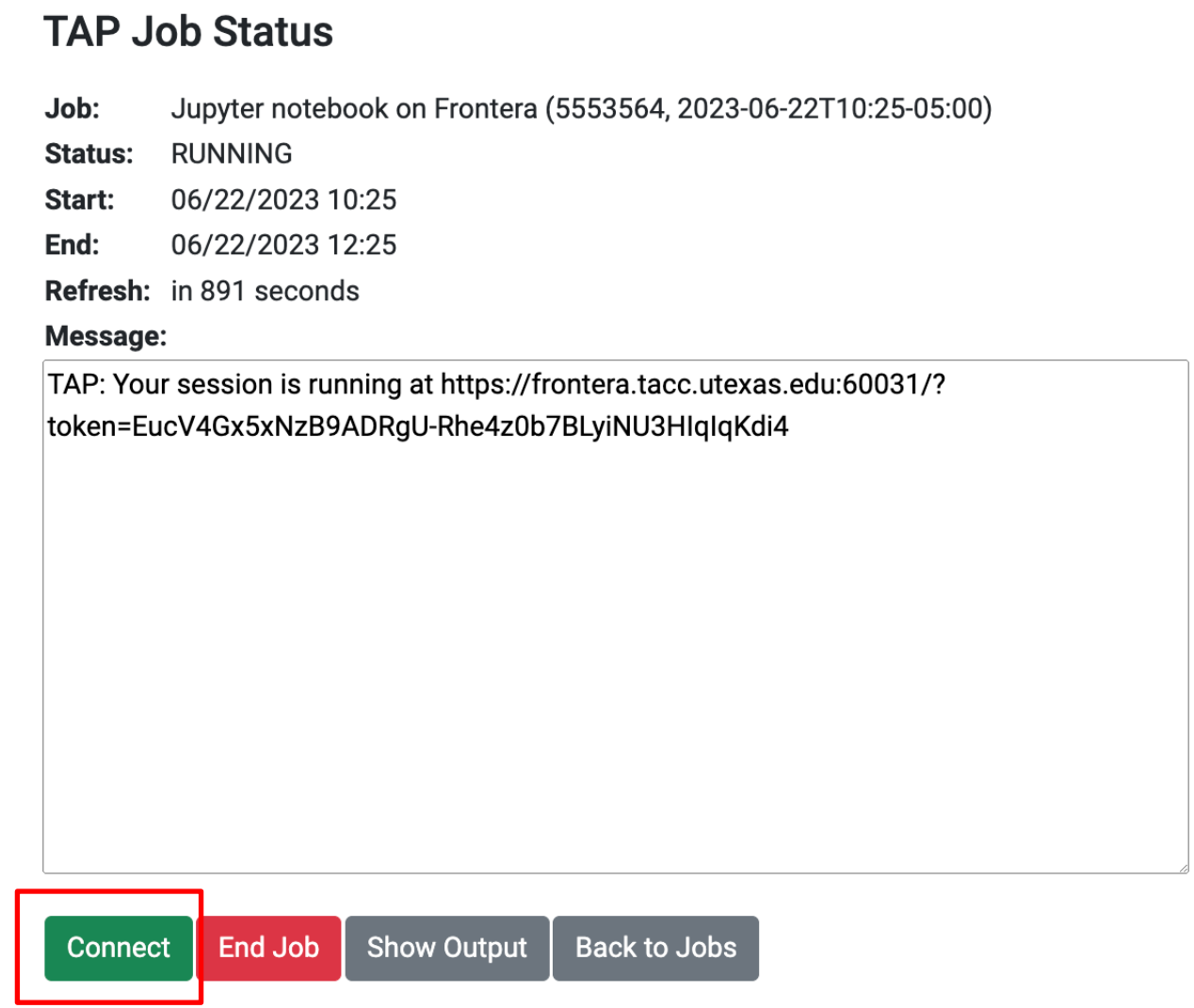
Start the Session

Launch a Jupyter Notebook

- Resource: Frontera
- Project: Frontera-Training
- Session Type: Jupyter Notebook
- Reservation ID: MLInst-Wed
- Queue: small
- Time Limit: 04:00:00

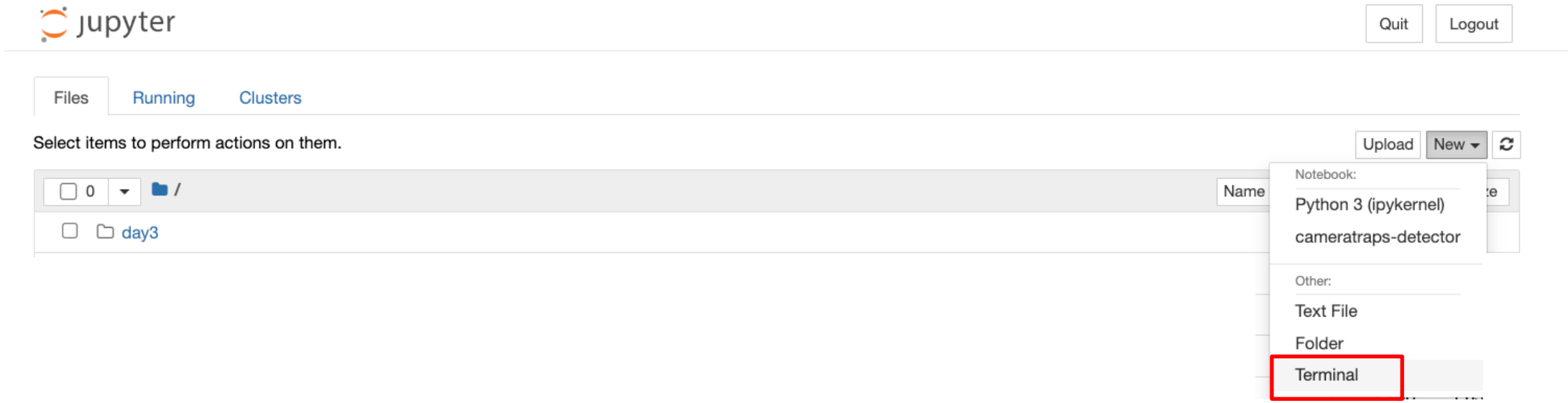


Starting Jupyter



Copy the Course Materials to Current Dir

1. Open a new terminal session



2. Copy the materials

```
cp -r /scratch1/01596/jrduncan/ML-Institute-Summer-2023/day3 .
```

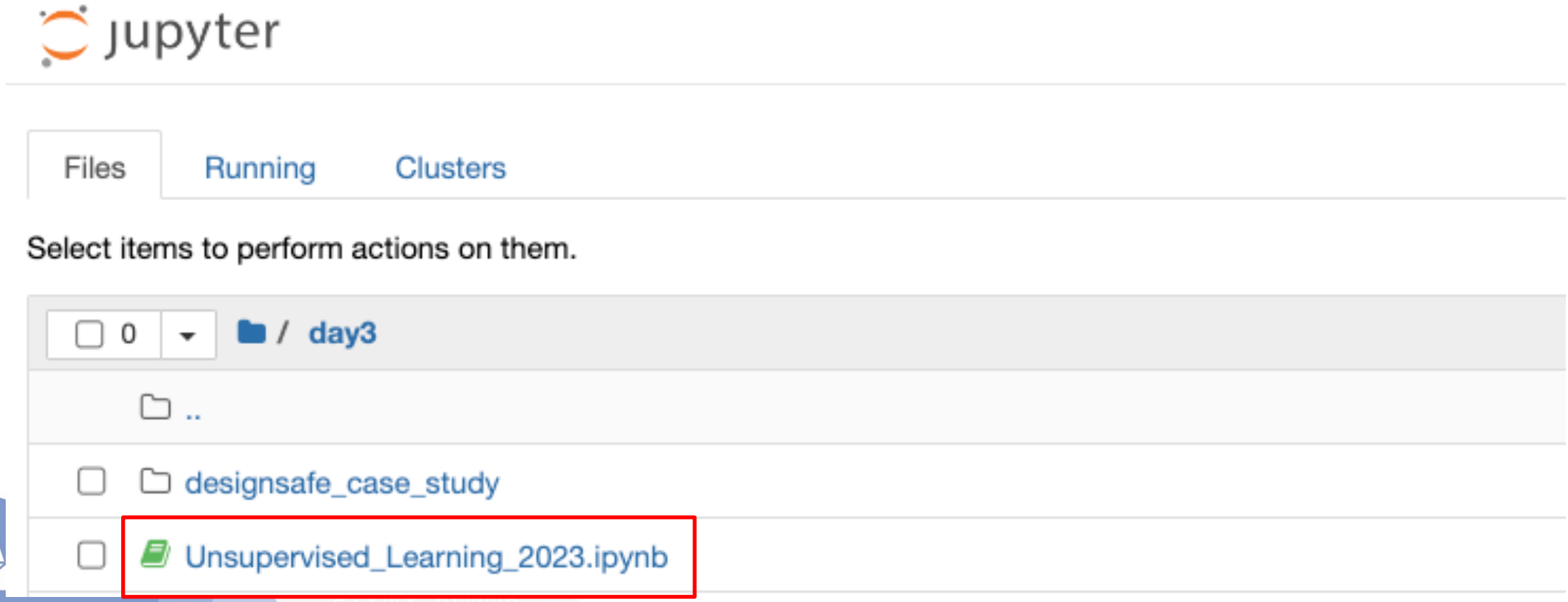
If you did not do hands-on in yesterday's training, and did not run this command
Run it in terminal

```
source /home1/00946/zzhang/ML-Institute-2023-PyTorch/day4-5/env.sh
```

- Restart Jupyter
  - Kill your current job
  - Submit a new one on TACC Analysis portal

Launch the Jupyter Notebook

Navigate into the **`day3`** directory and click on the notebook name to open it:



Hands-on break #1

Setup Materials

- Package Installation
- Plotting Utilities

Data Representation

- Principal Components Analysis (PCA) Example 1: Tumor Classification
  - Exercise 1
- PCA Example 2: Flower Identification
- PCA Example 3: Noise Reduction
- Multidimensional Scaling
- Locally Linear Embedding
- T-SNE

Hands-On Exercise

Clustering Algorithms

Hands-on break #2

Data Structure

- K-Means Clustering
  - Exercise 2
- Clustering Principal Components
  - Exercise 3
- Cluster Size
- Cluster Shape
- Agglomerative Clustering
  - Exercise 4

Hands-On Exercises

Data Density

DesignSafe Image Classification

Hands-on break #3

Data Density

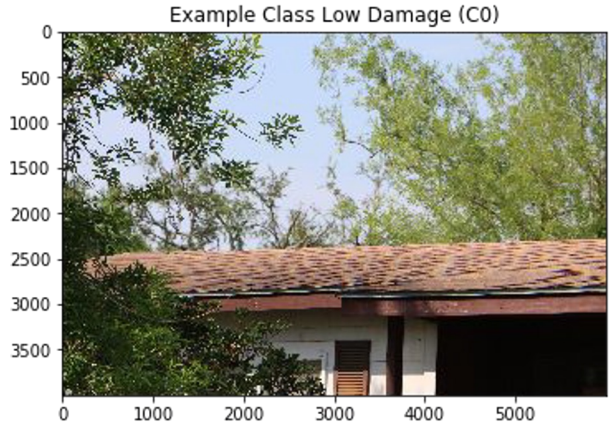
- Gaussian Mixture Models (GMM)
- Influence of Covariance Matrix on GMM
- GMM for Data Generation
- DBSCAN
    - Excercise 5

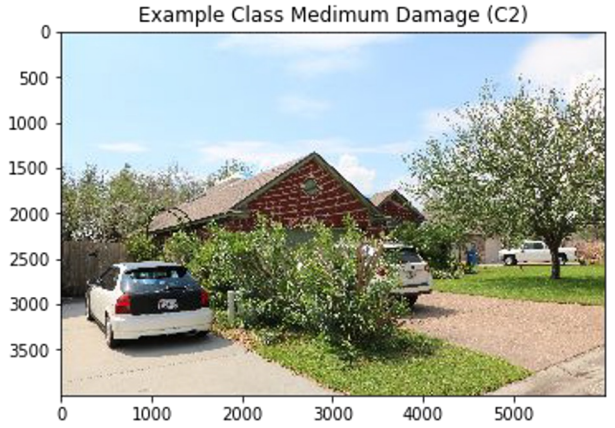
Structure in Random Data

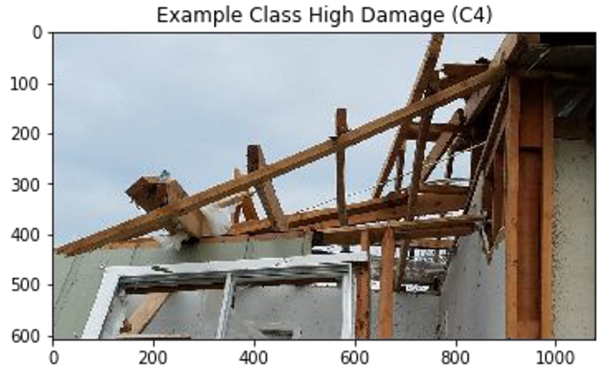
Exercise

DESIGNSAFE-CI
A NATURAL HAZARDS ENGINEERING COMMUNITY

Image Classification with Hurricane Harvey Dataset







TACC

Last Notebook: Supervised Learning

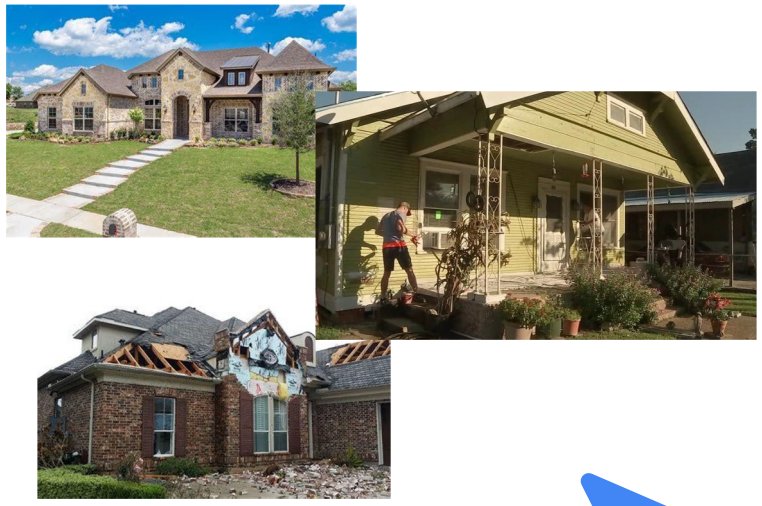


Image processing

Vectorized Data

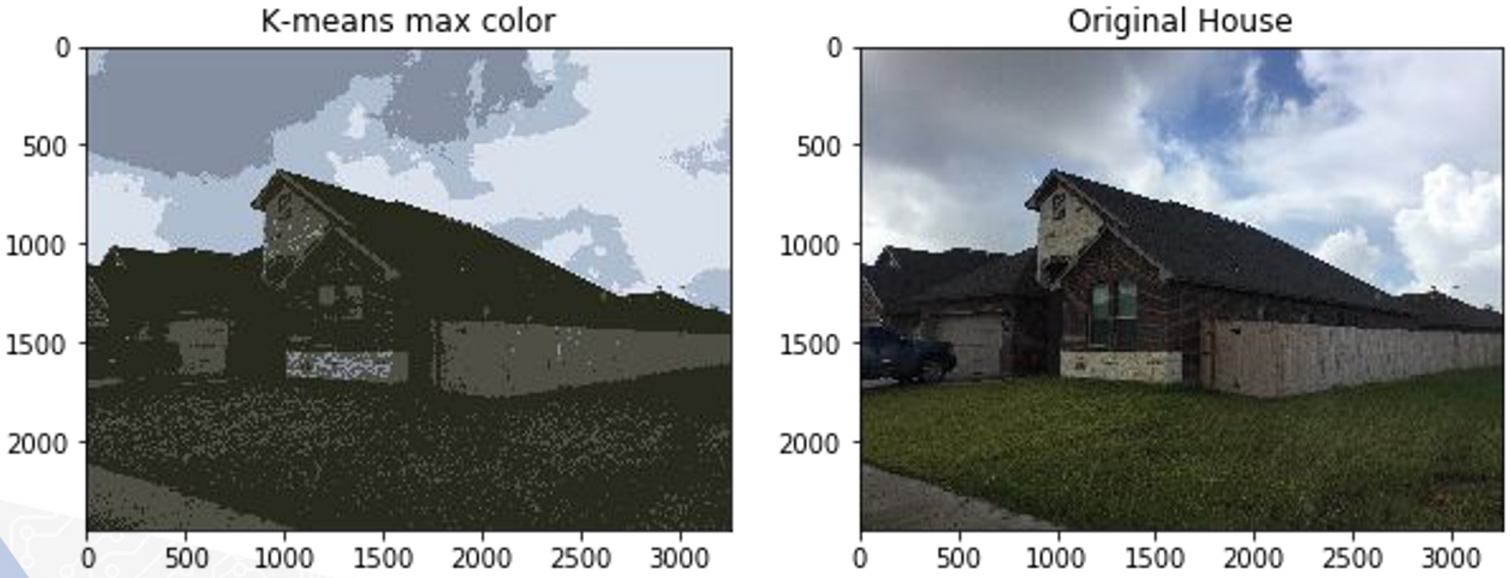
									low
		pixels							medium
									high

Train Model

Evaluate Model

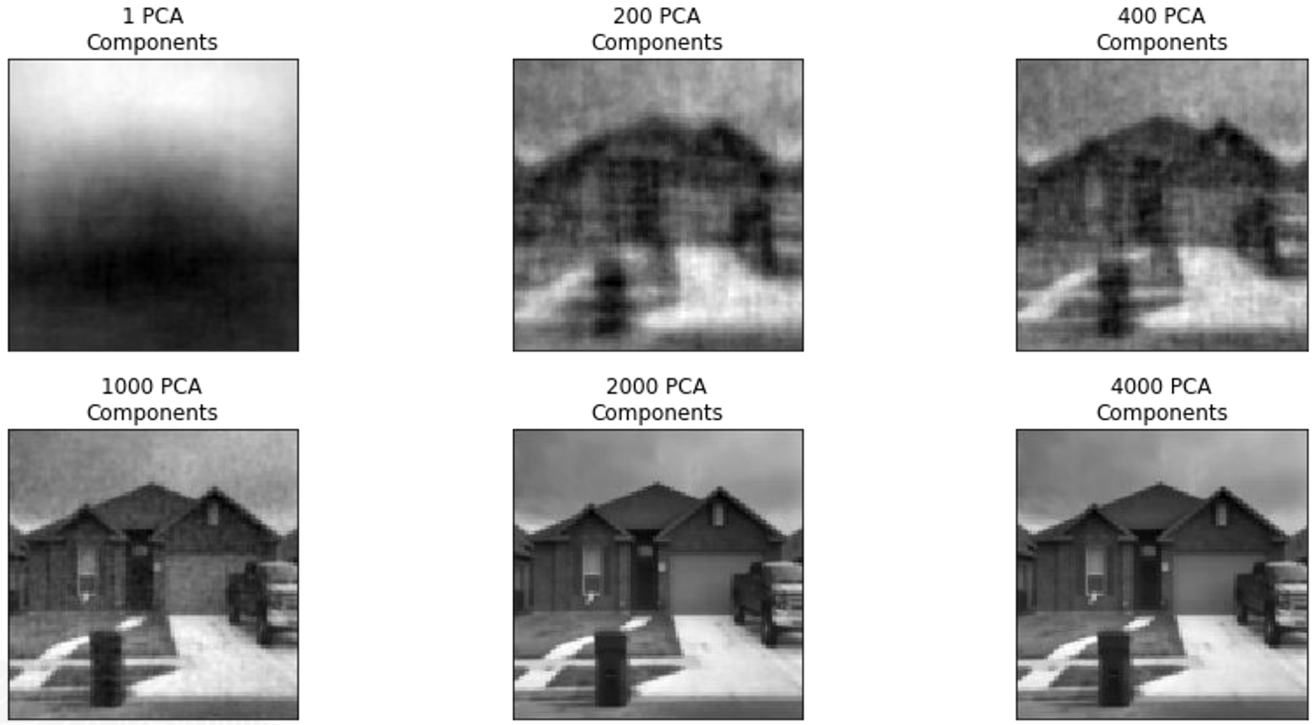
This Notebook: Image Compression with Unsupervised Learning

- Color Quantization with K-means
  - Reduce the total number of colors in an image
  - k-means on the colors in all pixels in an image
  - The number of colors is hyper parameter k
  - The color is the center of each cluster



This Notebook: Image Compression with Unsupervised Learning

- PCA for Image Compression
  - Run PCA on a dataset of images
  - Reconstruct images using principal components



TACC

This Week: Dimensionality Reduction with Unsupervised Learning

Vectorized

									low
				pixels					medium
									high

Dimensionality Reduction

			low
			medium
			high

Train Model

Evaluate Model

TACC

Instructions

In this exercise you will run one notebooks which explores applications of unsupervised learning in images

- Copy design safe data by running the cells in **copy-data.ipynb**
- Run the **designsafe_unsupervised_learning.ipynb**