Clustering

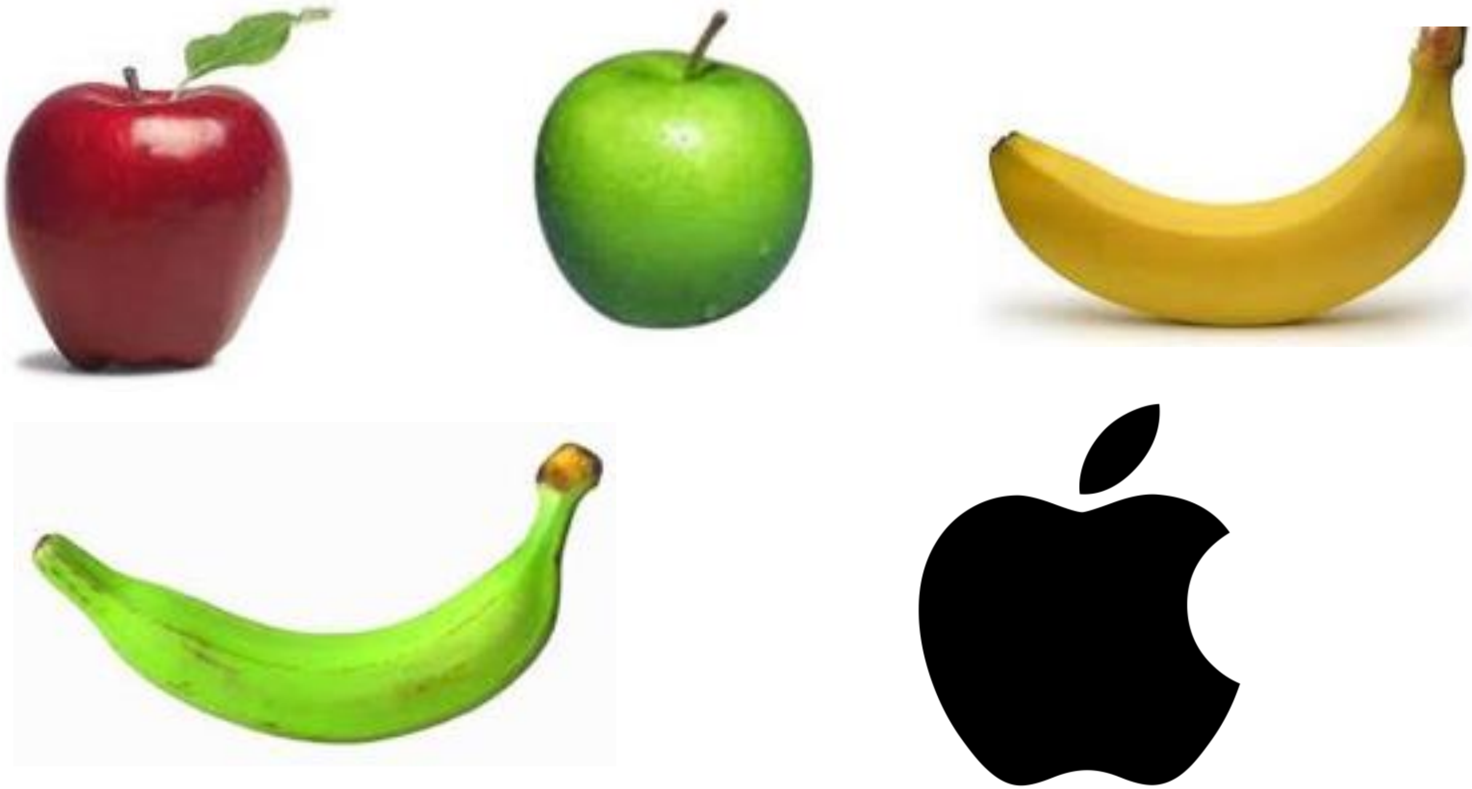
Data Intelligence and Learning (DIAL) Lab

Prof. Jongwuk Lee

Clustering Basics

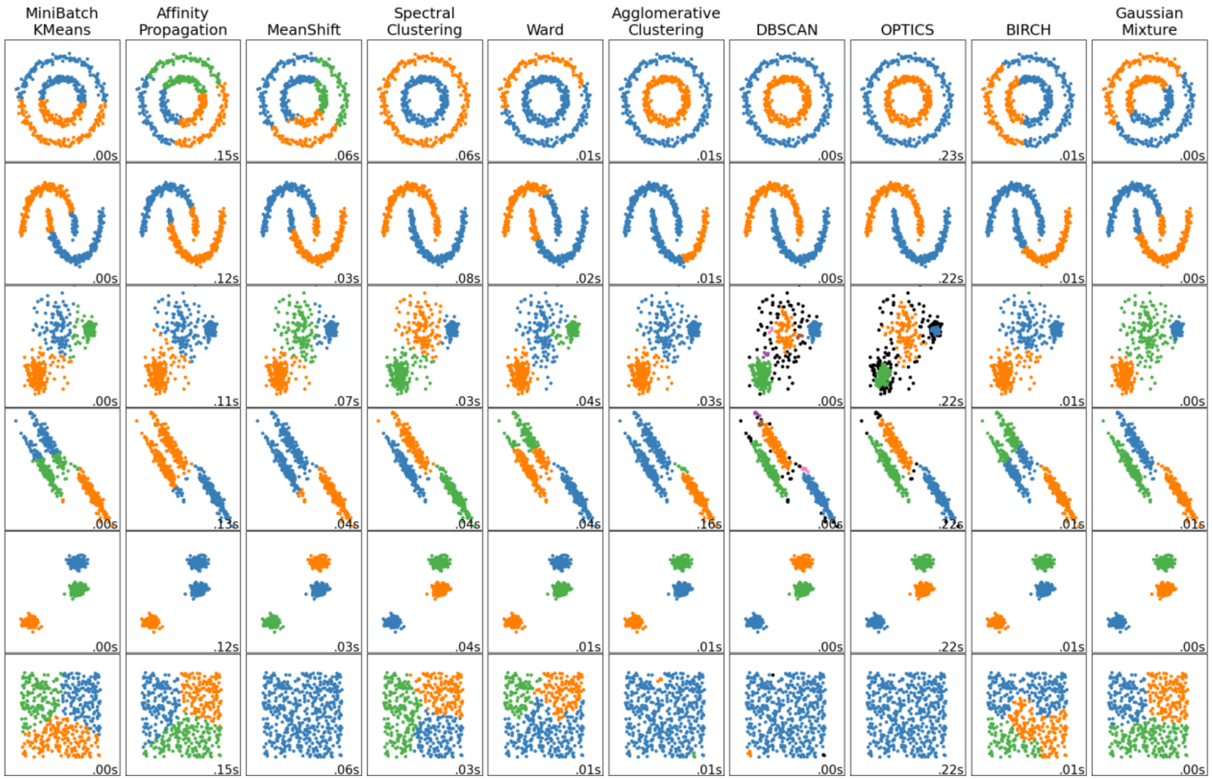
What is Clustering?

➢ **Given** **unlabeled data**, **how to partition into two groups?**



What is Clustering?

- **Goal: Partition unlabeled data into groups of similar samples.**
- **Q: Why is clustering useful?**
- **A: It helps automatically organize data.**

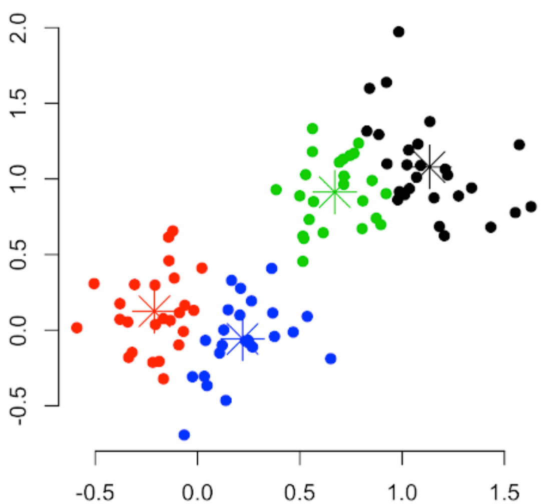


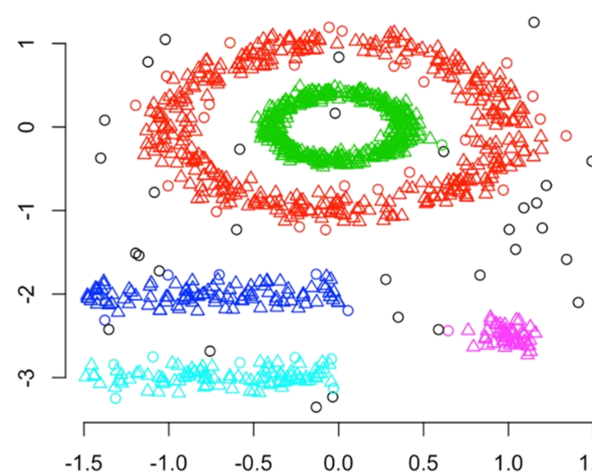
https://scikit-learn.org/stable/modules/clustering.html

What is Clustering?

- **Grouping similar samples into clusters**
  - **High intra-class similarity**: **cohesive** within clusters
  - **Low inter-class similarity**: **distinctive** between clusters

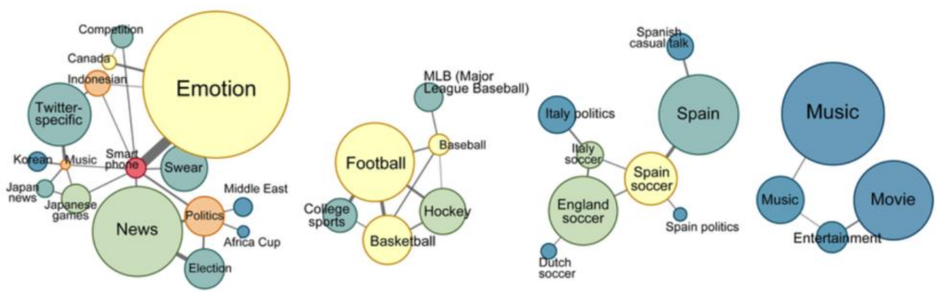
- **Clustering methods depend on the similarity measure.**
  - It is critical to define the **similarity** between samples according to **data characteristics**.



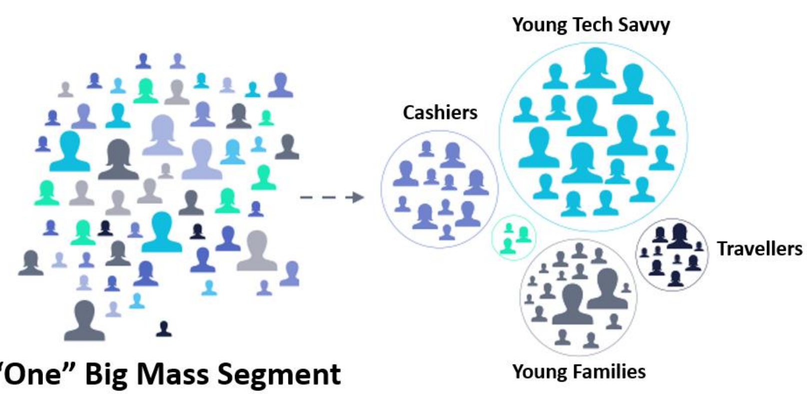


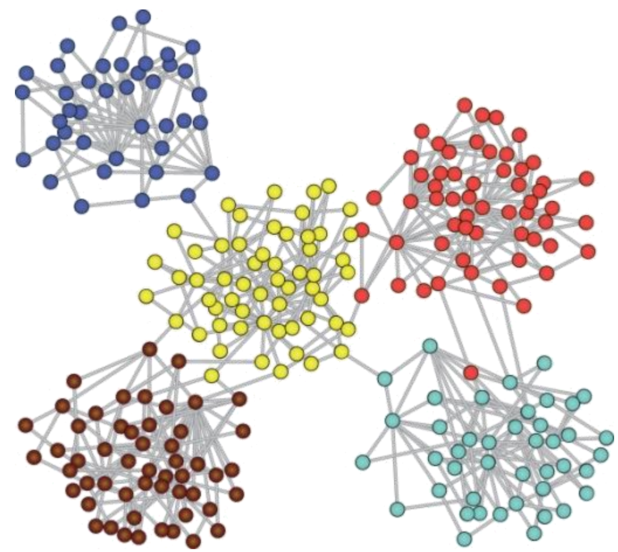
Popular Applications

- Categorize news articles or web pages by **topic**.
- Partition users in social networks by **interest**.
- Segment customers according to **purchase history**.



(a) News & Politics (b) Sports (c) Soccer (d) Entertainment





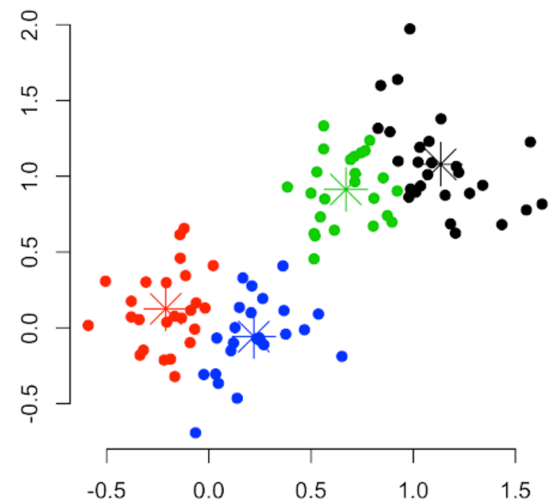
Various Clustering Approaches

- **Partitioning criteria**
  - **Single-level** vs. **hierarchical** partitioning
  - Multi-level hierarchical partitioning may be desirable.
- **Separation of clusters**
  - **Exclusive**: one tuple belongs to only one cluster.
  - **Non-exclusive**: one tuple may belong to one or more clusters.
- **Similarity measure**
  - **Distance-based**, e.g., Euclidian, road network, vector
  - **Connectivity-based**, e.g., density or contiguity
- **Clustering space**
  - **Full space** (often when low dimensional)
  - **Subspaces** (often in high-dimensional clustering)

Flat vs. Hierarchical Clustering

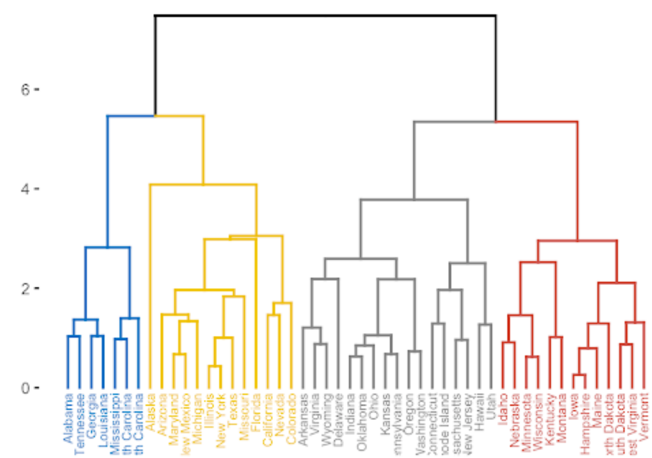
➢ Partitioning criteria

- **Single-level** vs. **hierarchical** partitioning



➢ Partitioning approach

- Constructing the clusters and evaluate them by some criterion.
- Minimizing the sum of square errors
- Methods: k-means, k-medoids



➢ Hierarchical approach

- Bottom-up or top-down models
- Methods: DIANA, AGNES, BIRCH, CAMELEON

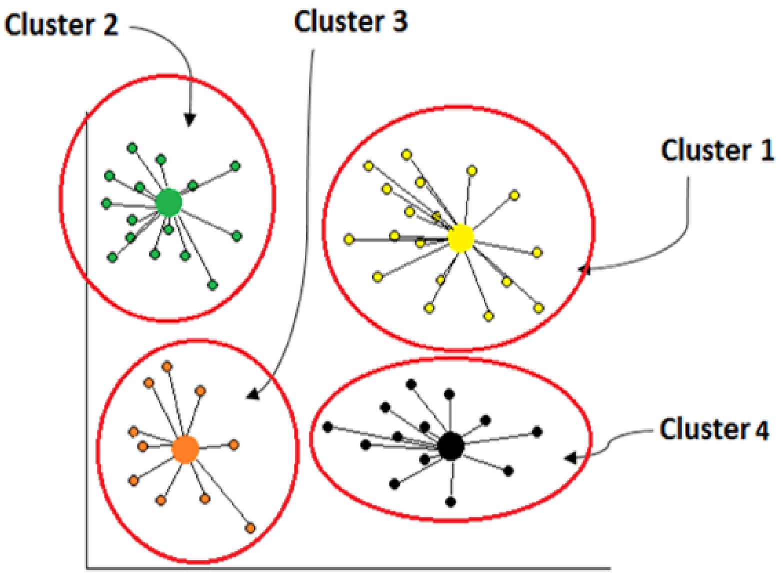
k-Means Clustering

Objective Function

- **Partition a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$ into k groups $\mathcal{C} = \{\mathbf{C}_1, \dots, \mathbf{C}_k\}$.**
  - The sum of squared distances is minimized.
  - Let $\boldsymbol{\mu}_j$ be the **centroid** of the cluster $\mathbf{C}_j$.

$$Error(\mathcal{D}) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in \mathrm{C}_i} \left\| \mathbf{x} - \boldsymbol{\mu}_j \right\|^2$$

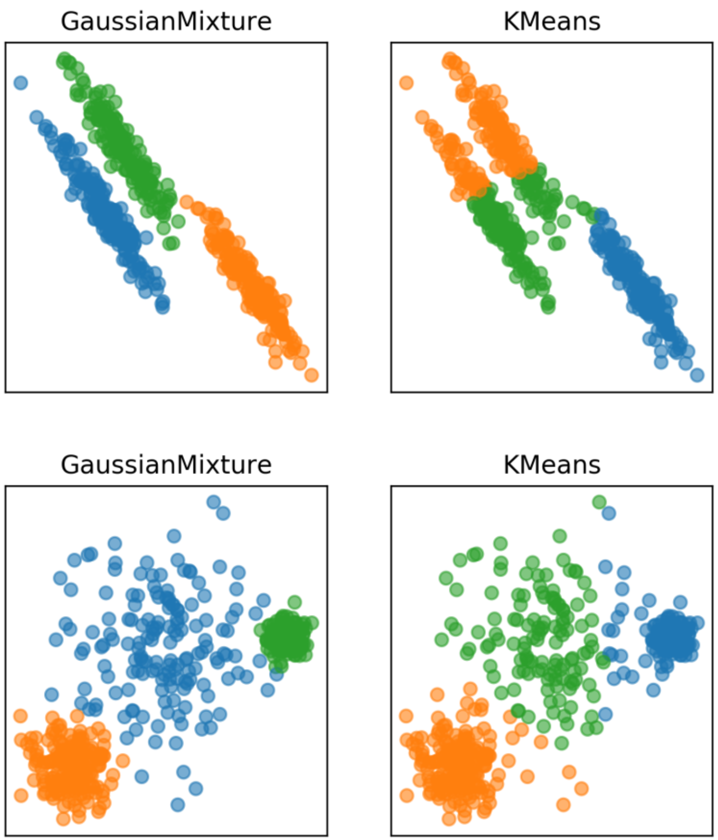
The distance between $\mathbf{x} \in \mathrm{C}_j$ and the centroid $\boldsymbol{\mu}_j$



Objective Function

- Finding an optimal solution is **NP-hard**.

- There are good heuristic methods that work well in practice.
  - K-means clustering
  - Gaussian mixture clustering



https://amueller.github.io/aml/03-unsupervised-learning/02-clustering-mixture-models.html

k-Means Clustering

- An iterative clustering algorithm
  - **Assigning samples** and **updating centroids**.
- Assuming Euclidean space, start by picking k centroids.

Initialize: Pick k **random samples** as **the centers of clusters**.

Alternate:

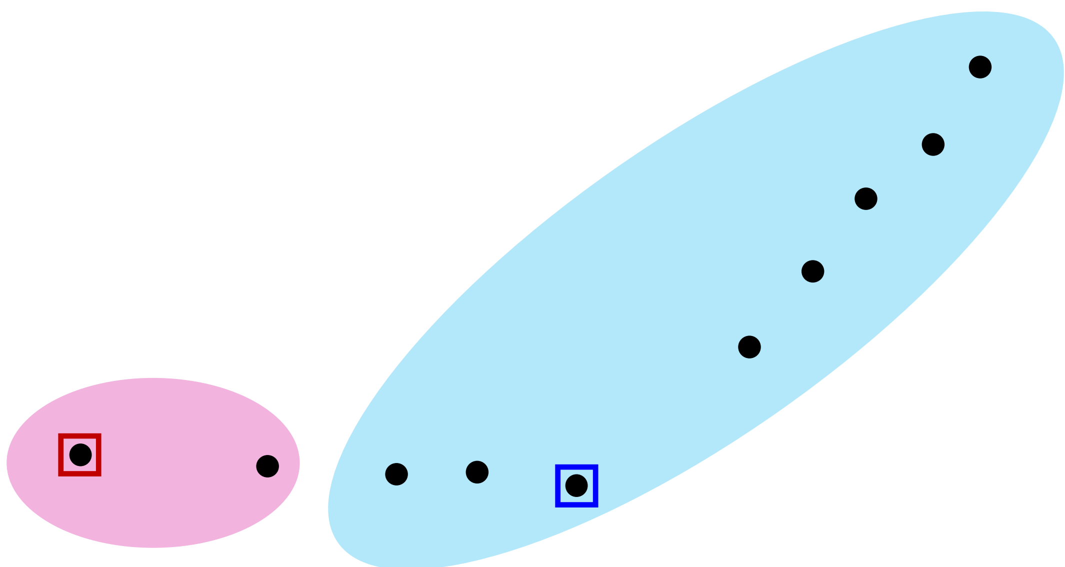
Assign samples to the **closest clustering center**.

Update the centroid as **the average of assigned samples**.

Stop when no sample assignments change.

Example

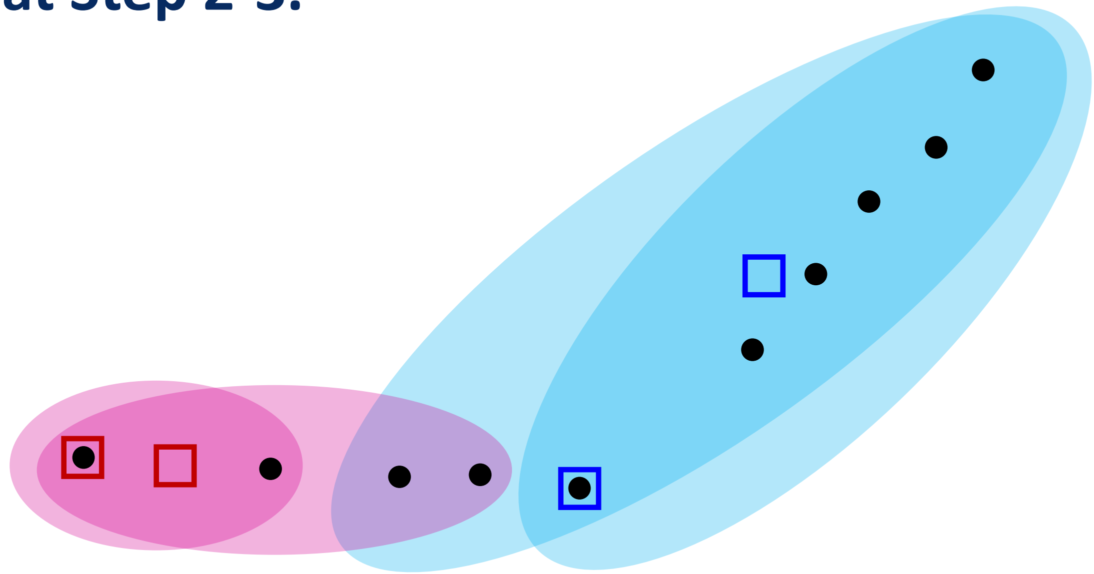
- Step 1: Initialize random samples as the centers of clusters.
- Step 2: Assign samples to the closest centroid.



● : sample
□ : centroid

Example

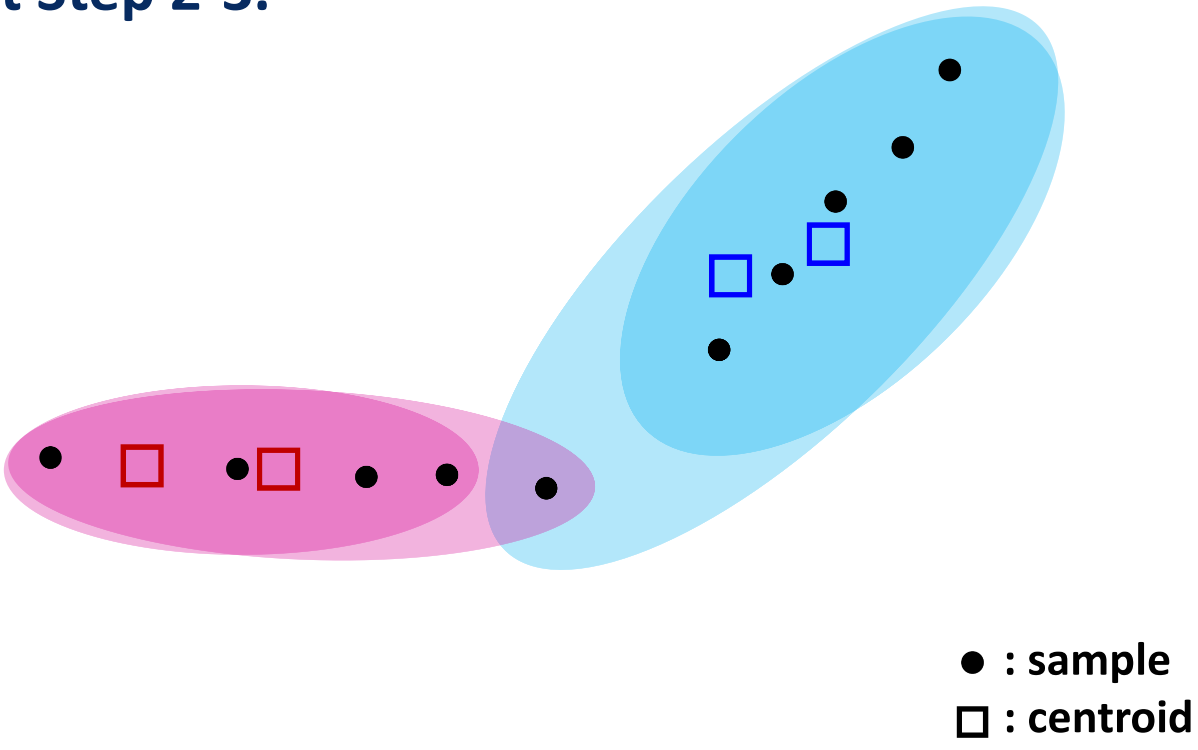
- Step 2: Assign samples to the closest centroid.
- Step 3: Update the centroid as the average of assigned samples.
- Repeat Step 2-3.



● : sample
□ : centroid

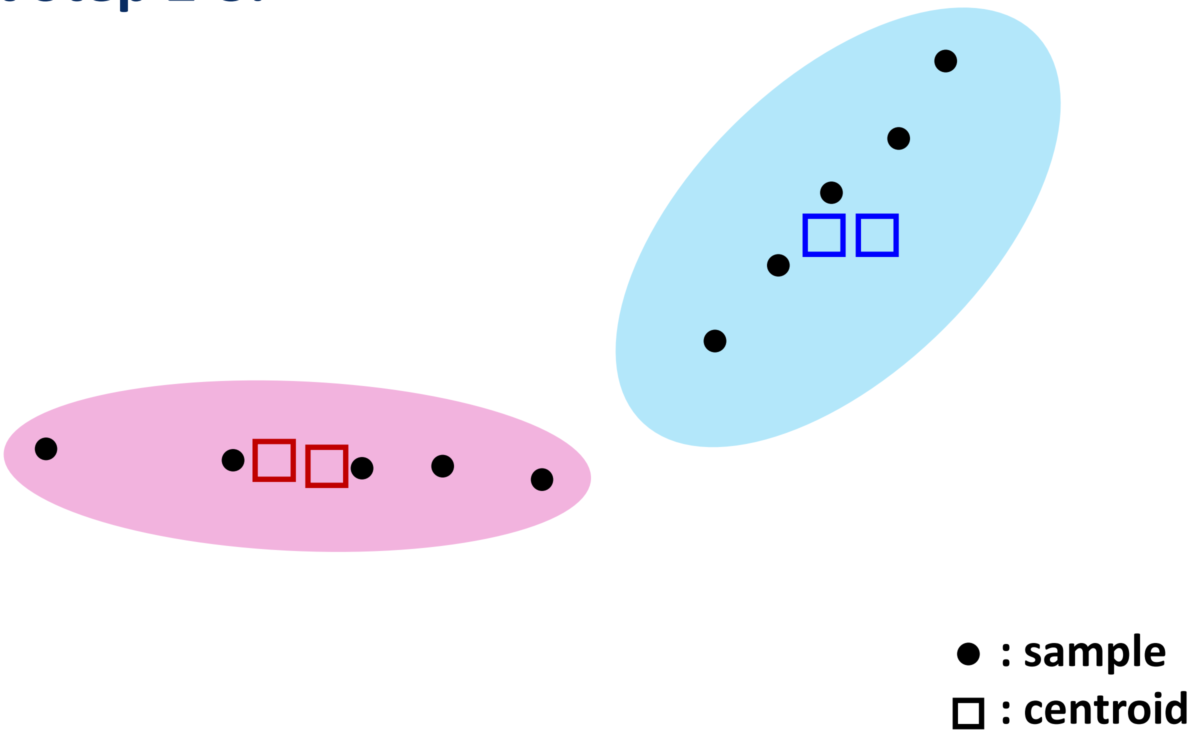
Example

- ➢ **Step 2: Assign samples to the closest centroid.**
- ➢ **Step 3: Update the centroid as the average of assigned samples.**
- ➢ **Repeat Step 2-3.**



Example

- Step 2: Assign samples to the closest centroid.
- Step 3: Update the centroid as the average of assigned samples.
- Repeat Step 2-3.



Computation Complexity

- **It converges in a finite number of iterations.** $\rightarrow O(tkn)$

- **Running time per iteration:**

Initialize: Pick k **random samples** as **the centers of clusters**.

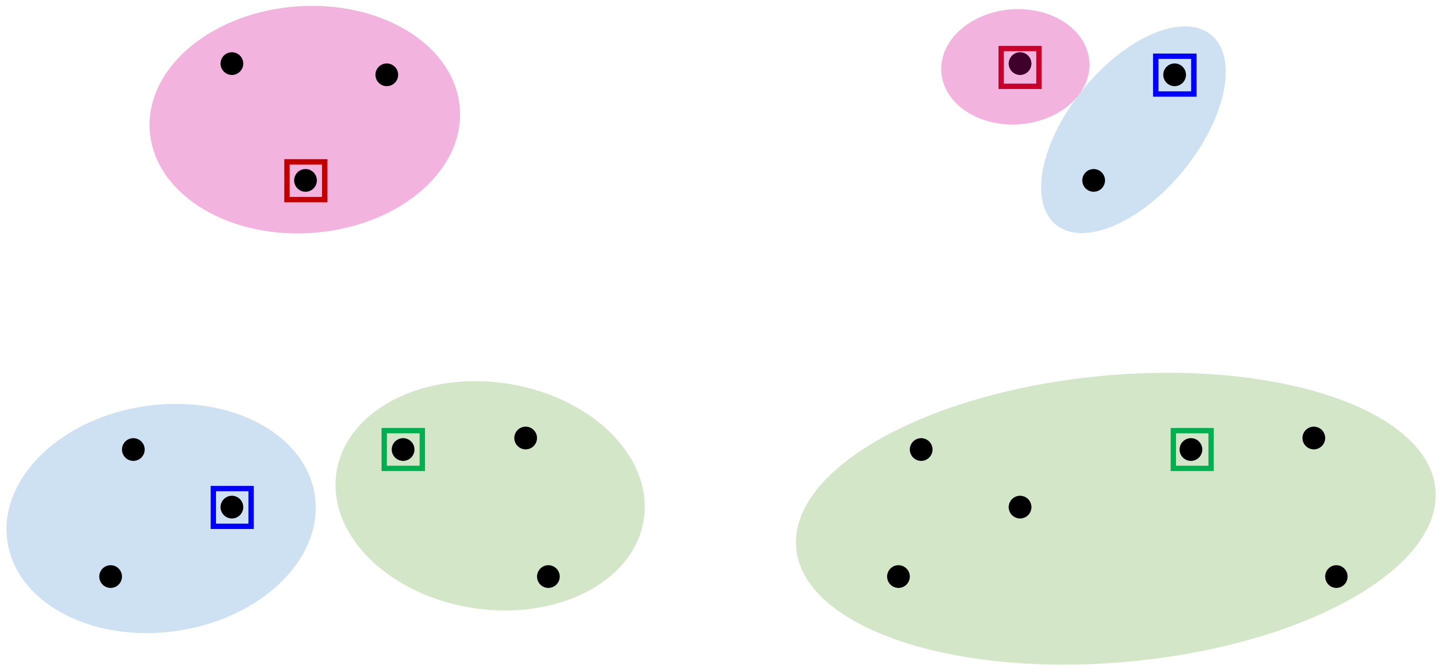
Alternate:

Assign samples to the **closest clustering center.** $\rightarrow O(kn)$
Update the centroid as **the average of assigned samples**.
$\rightarrow O(n)$

Stop when no sample assignments change.

Initialization of k-Means Clustering

➢ **The results can vary on random seed selection.**



Initialization of k-Means Clustering

- **Some seeds can result in a poor convergence rate or converge to sub-optimal clustering.**

- **k-means algorithm can get stuck easily in local minima.**
  - **Note: Try out multiple starting points. (very important!)**

- **Initialize with the results of another method.**
  - E.g., random farthest seed selection
  - k-means++ is supported by the scikit-learn library.

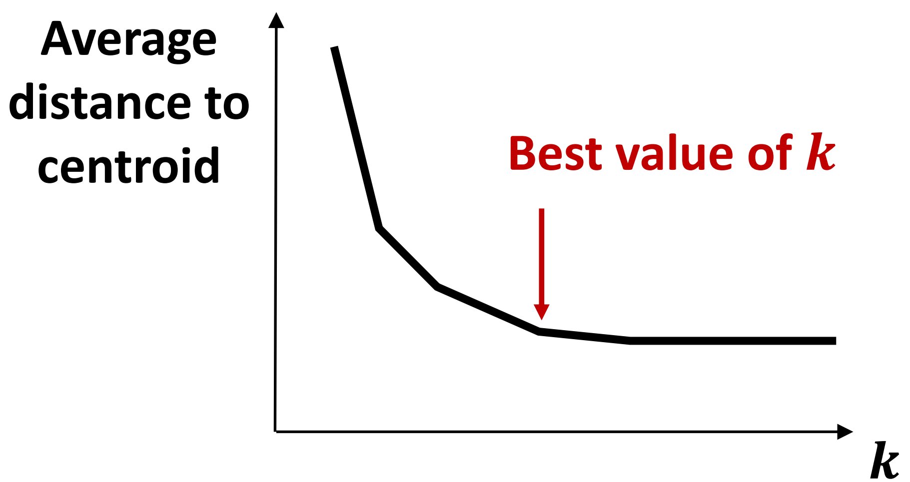
```
from sklearn.cluster import KMeans

model = KMeans(n_clusters=k,  init='k-means++')
```


https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

How to Choose k?

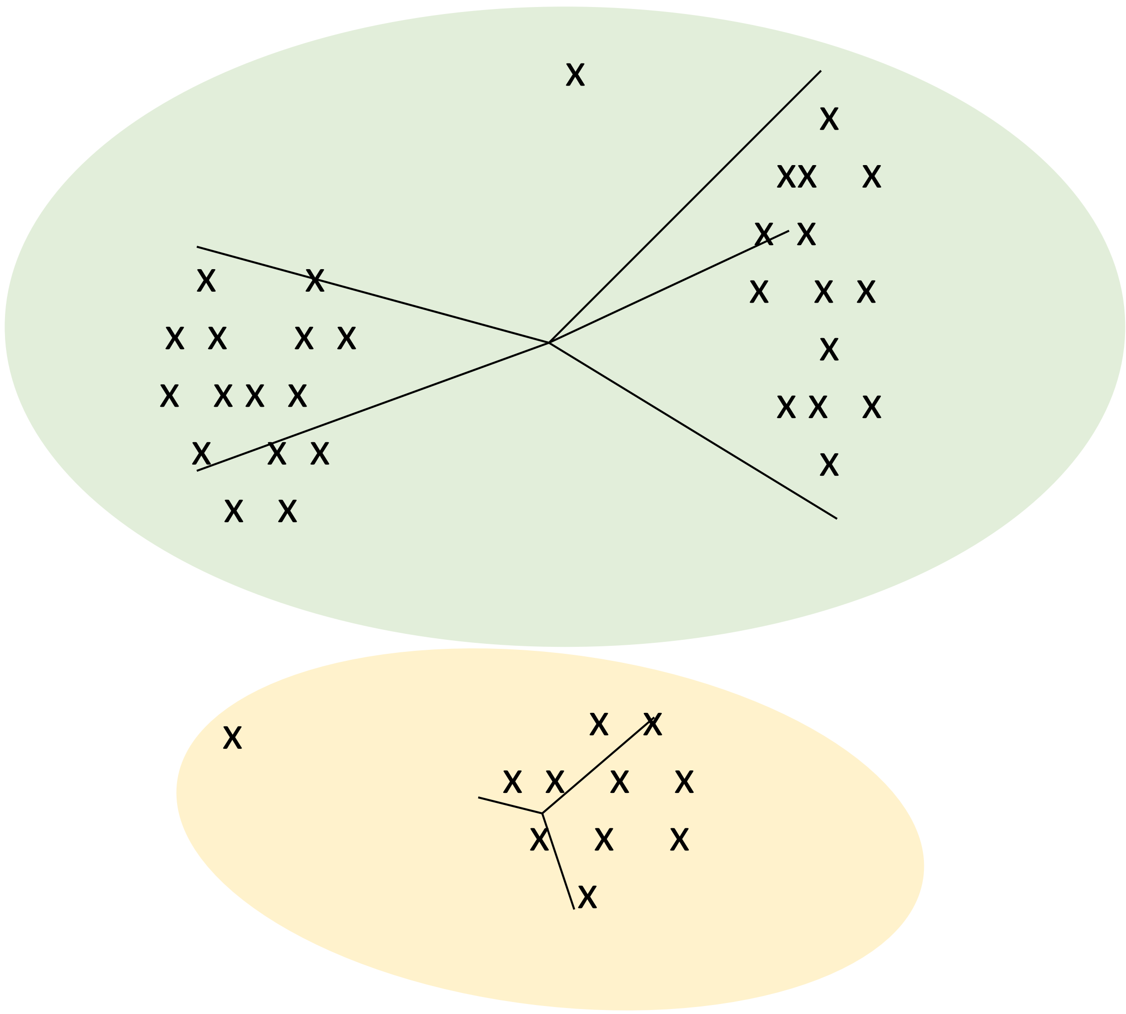
- Try different k, looking at the change in the average distance to centroid as k increases.

- Average falls rapidly until right k, then changes little.



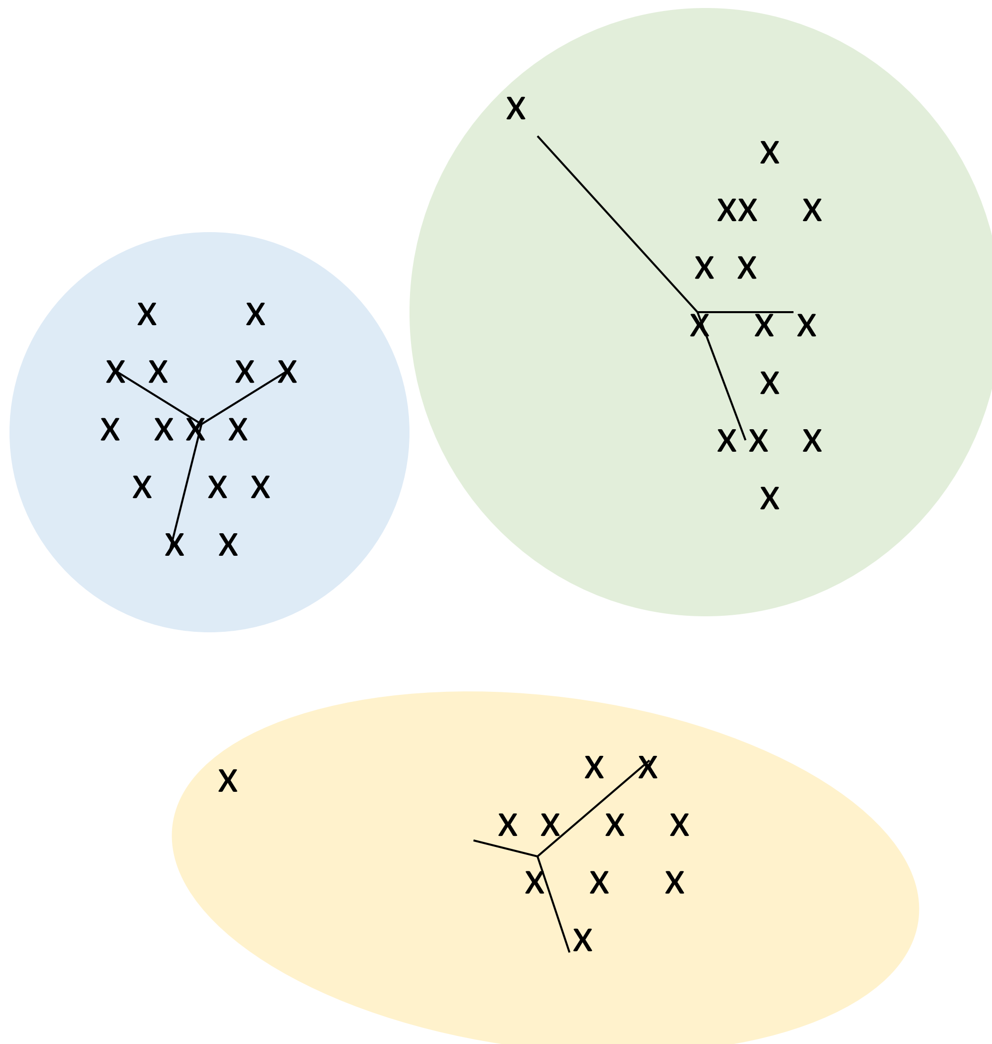
Example: Picking k

Too few;
many long distances to centroid



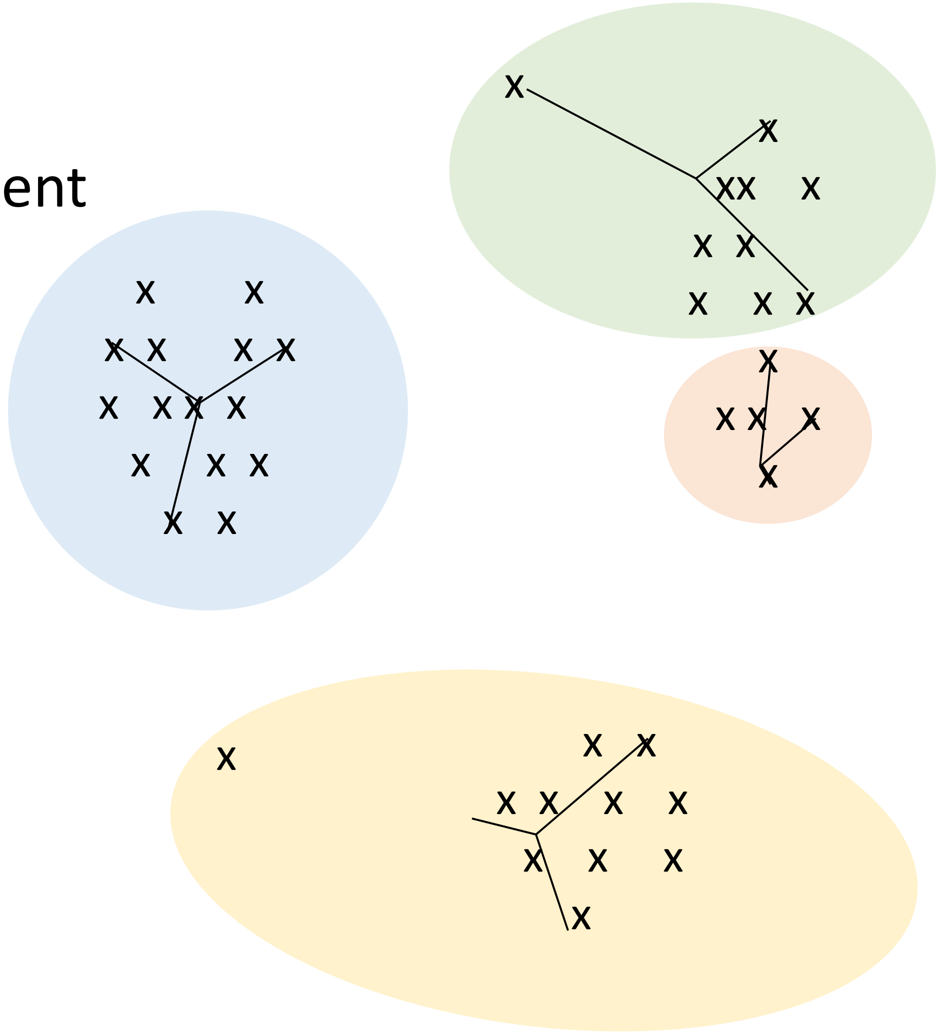
Example: Picking k

Just right;
distances
rather short



Example: Picking k

Too many;
little improvement
in average
distance



Summary of k-Means Clustering

- **Strength**
  - **Efficiency**: **It has linear complexity $O(tkn)$,** where n is # of instances, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.

- **Weakness**
  - It should specify **the number k of clusters in advance**.
  - It often terminates at **local optima**.
  - It is **sensitive to outliers**.

Image Segmentation

- The goal of image segmentation is to partition an image into several regions with a homogenous visual appearance.

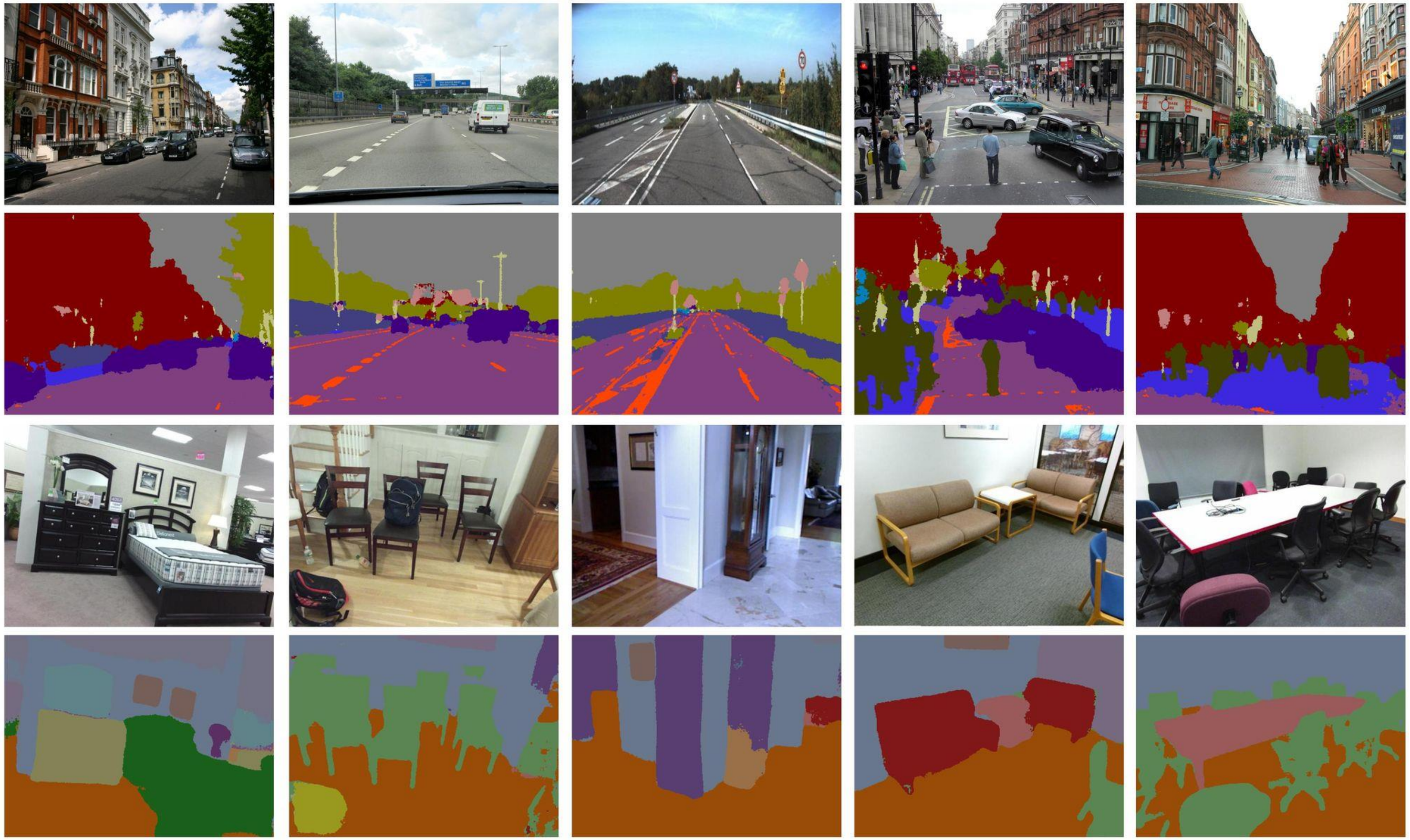
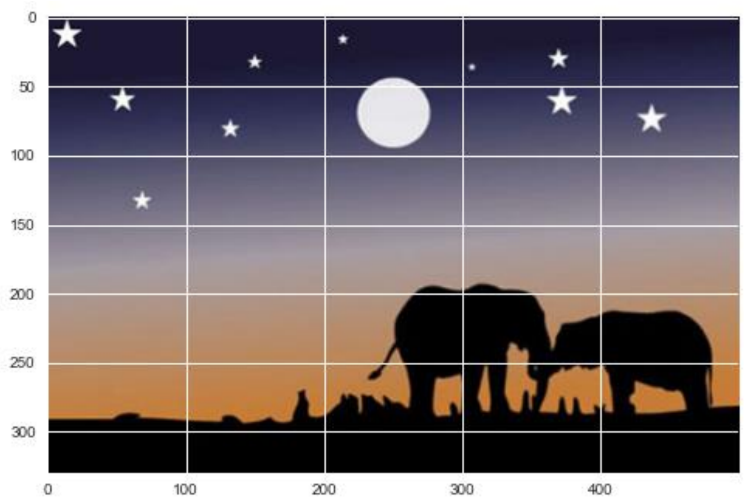


Image Segmentation using k-Means

- Groups similar pixels as homogenous visual objects.



Original image

Converted image

Dominant colors

https://medium.com/analytics-vidhya/image-segmentation-using-k-means-4bd79e801785

Image Segmentation using k-Means

$k = 2$ | $k = 3$ | $k = 10$ | Original image

Gaussian Mixture Model (GMM)

Density Estimation

- **Generative approach**
  - Assume that there is a latent parameter θ.
  - For all i, draw observed $\mathbf{x}^{(i)}$ given θ.

$$p\left(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)} \mid \theta\right) = \prod_{i=1}^{n} p\left(\mathbf{x}^{(i)} \mid \theta\right)$$

- **What if the model does not fit all data?**
  - Introduce different parameters $\theta_1, \cdots, \theta_k$ for different parts of data.

- **It is called partitioning algorithms or mixture modeling.**

Partitioning Algorithms

➢ K-means clustering

- **Hard assignment**: each sample belongs to only one cluster.

$$\theta_i \in \{\theta_1, \cdots, \theta_k\}$$

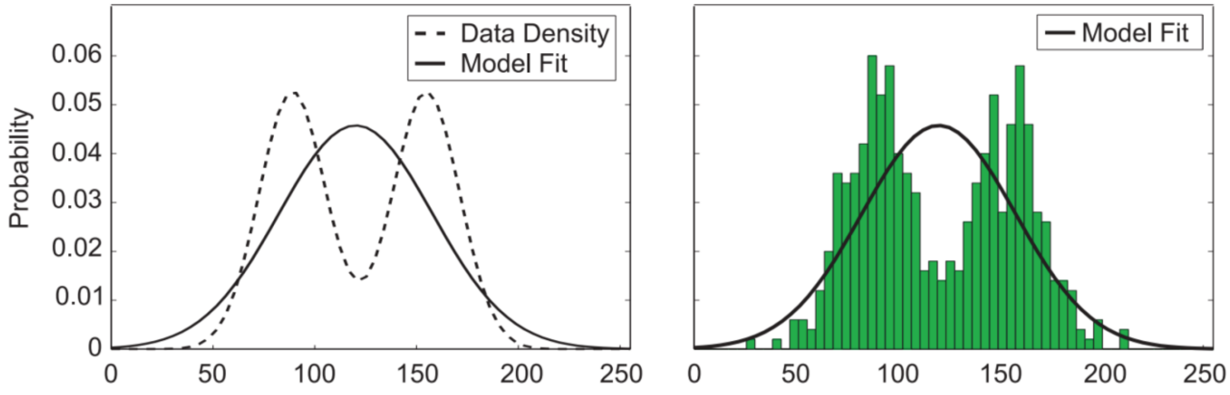
➢ Mixture modeling

- **Soft assignment**: a probability that a sample belongs to a cluster.

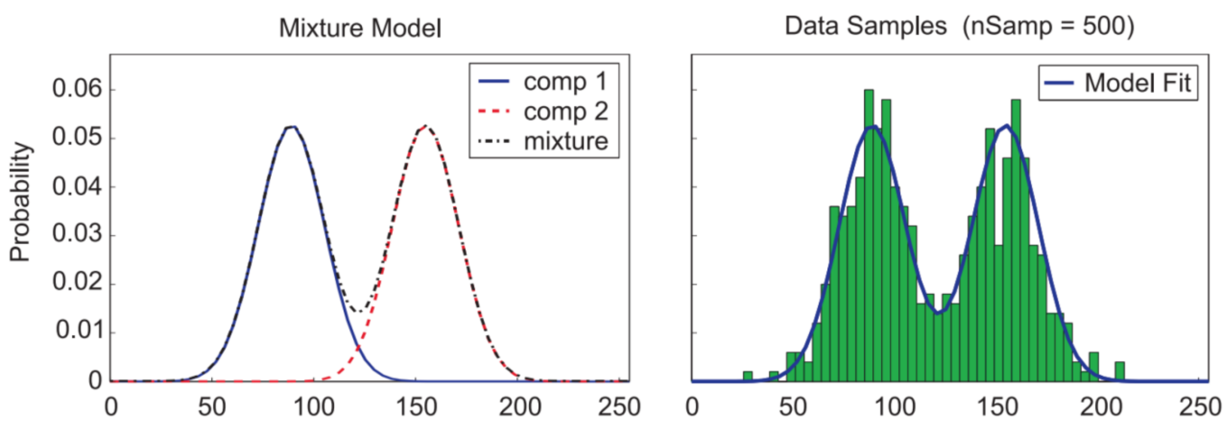
$$(\pi_1, \cdots, \pi_k), \pi_i \geq 0, \sum_{i=1}^{k} \pi_i = 1$$

Visualizing a Mixture of Gaussians (1D)

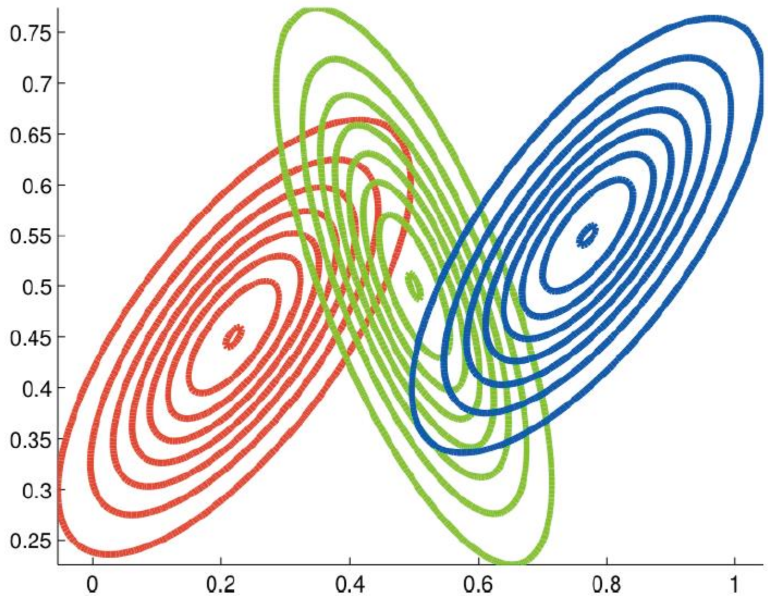
- If you fit a Gaussian to data,

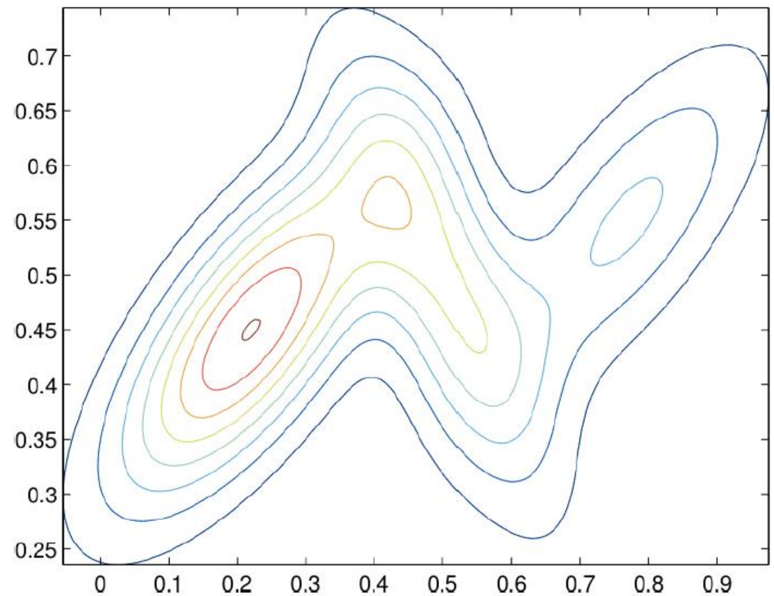


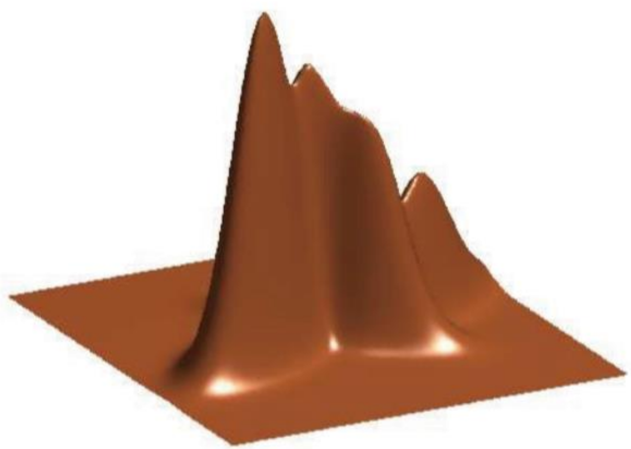
- We can try to fit a GMM with $k = 2$.



Visualizing a Mixture of Gaussians (2D)

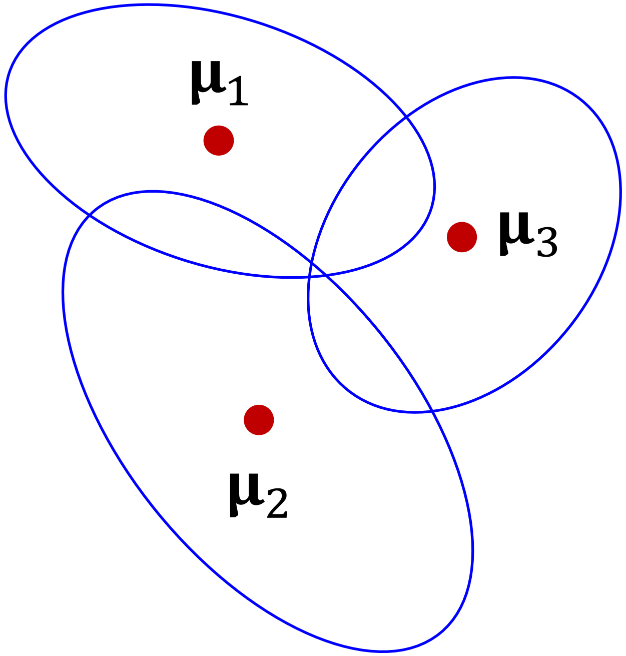






Gaussian Mixture Models (GMM)

- **It is represented by the mixture of k Gaussian distributions.**
- **Each component i has an associated mean vector $\boldsymbol{\mu}_i$.**
  - Each component generate data from $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.



- **Each data sample is generated using this process.**
  - Choose component i with the probability $\pi_i = p(y = i)$.
  - Data sample $x \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.

Gaussian Mixture Models (GMM)

- It is represented by **the mixture of k Gaussian distributions.**
- Since we have unlabeled data, y **is the hidden variable.**

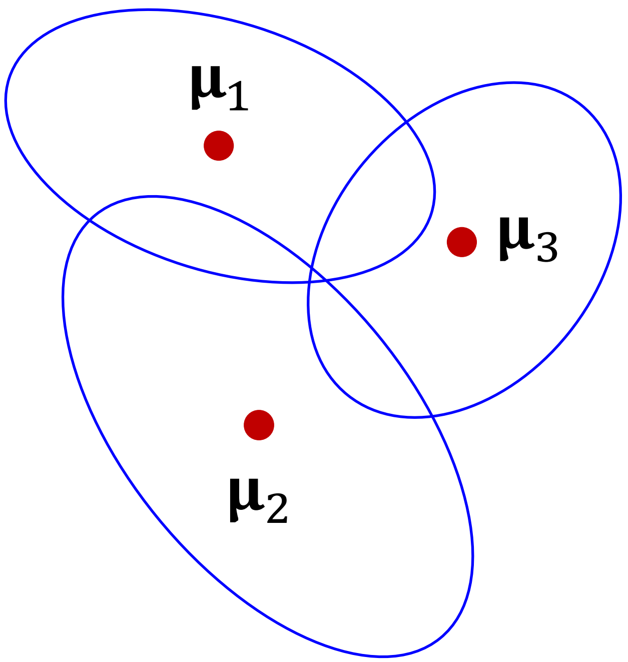
$$p(x \mid y = i) = \mathcal{N}(\mu_i, \Sigma_i)$$

$$p(x) = \sum_{i=1}^{k} p(x \mid y = i) p(y = i)$$

Observed data

Mixture component

Mixture proportion

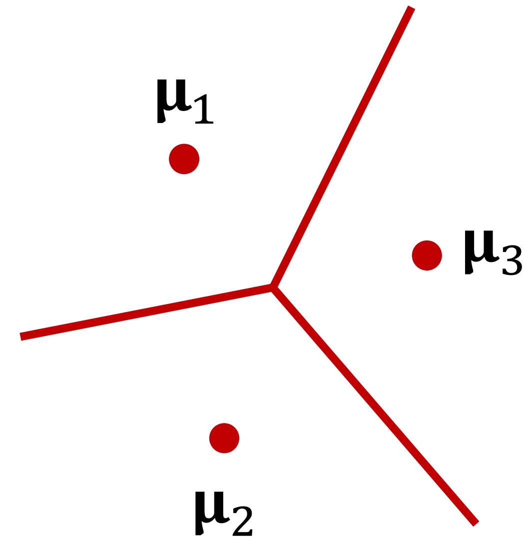


Simplifying Gaussian Mixture Models

- **Assume that**
  - For simplicity, $\Sigma_i = \sigma^2 \mathbf{I}$.

$$p(x \mid y = i) = \mathcal{N}(\mu_i, \sigma^2 \mathbf{I})$$

$$p(x \mid y = i) = \pi_i$$



- **Clustering x based on the posterior.**
  - It depends on $\mu_1, \cdots, \mu_k, \sigma^2, \pi_1, \cdots, \pi_k$.

$$\log \frac{p(y = i \mid x)}{p(y = j \mid x)} = \log \frac{p(x \mid y = i) p(y = i) / p(x)}{p(x \mid y = j) p(y = j) / p(x)} = \log \frac{p(x \mid y = i) p(y = i)}{p(x \mid y = j) p(y = j)}$$

$$= \log \frac{p(x \mid y = i) \pi_i}{p(x \mid y = j) \pi_j} = \log \frac{\exp\left(-\frac{1}{2\sigma^2} \|x - \mu_i\|^2\right) \pi_i}{\exp\left(-\frac{1}{2\sigma^2} \|x - \mu_i\|^2\right) \pi_j} = \mathbf{w}^{\mathrm{T}} x$$

MLE for GMM

- **Q: What if we do not know $\mu_1, \cdots, \mu_k, \sigma, \pi_1, \cdots, \pi_k$?**
- **A: Use Maximum Likelihood estimation (MLE).**
  - Let $\theta = \{\mu_1, \cdots, \mu_k, \sigma^2, \pi_1, \cdots, \pi_k\}$.

$$\operatorname*{argmax}_{\theta} \prod_{j=1}^{n} p\left(x^{(j)} \mid \theta\right) = \operatorname*{argmax}_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{k} p\left(y^{(j)} = i, x^{(j)} \mid \theta\right)$$

$$= \operatorname*{argmax}_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{k} p\left(y^{(j)} = i \mid \theta\right) p\left(x^{(i)} \mid y^{(j)} = i, \theta\right)$$

$$= \operatorname*{argmax}_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{k} \pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left\|x^{(j)} - \mu_i\right\|^2}{2\sigma^2}\right)$$

k-Means Clustering and GMM

- Assume that data come from a mixture of k Gaussian distributions with the same variance σ^2.
- Also, assuming **hard assignment**,

$$p(y=i) = \begin{cases} 1, & i = C(j) \\ 0, & \text{otherwise} \end{cases}$$

$$\underset{\theta}{\operatorname{argmax}} \prod_{j=1}^{n} p\left(x^{(j)} \mid \theta\right) = \underset{\theta}{\operatorname{argmax}} \prod_{j=1}^{n} \sum_{i=1}^{k} \pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left\|x^{(j)} - \mu_i\right\|^2}{2\sigma^2}\right)$$

$$= \underset{\theta}{\operatorname{argmax}} \prod_{j=1}^{n} \exp\left(-\left\|x^{(j)} - \mu_{C(j)}\right\|^2\right)$$

Note: $e^A e^B = e^{A+B}$

$$= \underset{\mu, C}{\operatorname{argmin}} \sum_{j=1}^{n} \left\|x^{(j)} - \mu_{C(j)}\right\|^2$$

Same as k-means clustering!

General GMM

- **Assume that**
  - $\theta = \{\mu_1, \cdots, \mu_k, \mathbf{\Sigma}_1, \cdots, \mathbf{\Sigma}_k, \pi_1, \cdots, \pi_k\}$ is known.

$$p(x \mid y = i) = \mathcal{N}(\mu_i, \mathbf{\Sigma}_\mathrm{i})$$

$$p(x \mid y = i) = \pi_i$$

$$\log \frac{p(y = i \mid x)}{p(y = j \mid x)} = \log \frac{p(x \mid y = i) p(y = i)/p(x)}{p(x \mid y = j) p(y = j)/p(x)} = \log \frac{p(x \mid y = i) p(y = i)}{p(x \mid y = j) p(y = j)}$$

$$= \log \frac{p(x \mid y = i)\pi_i}{p(x \mid y = j)\pi_j} = \log \frac{\frac{1}{\sqrt{2\pi|\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^\mathrm{T} \mathbf{\Sigma}_i^{-1} (\mathbf{x} - \mu_i)\right) \pi_i}{\frac{1}{\sqrt{2\pi|\Sigma_j|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_j)^\mathrm{T} \mathbf{\Sigma}_j^{-1} (\mathbf{x} - \mu_j)\right) \pi_j}$$

MLE for General GMM

- **Q: What if we do not know $\mu_1, \cdots, \mu_k, \boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_k, \pi_1, \cdots, \pi_k$?**

$$\operatorname*{argmax}_{\theta} \prod_{j=1}^{n} p\left(x^{(j)} \mid \theta\right) = \operatorname*{argmax}_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{k} p\left(y^{(j)} = i, x^{(j)} \mid \theta\right)$$

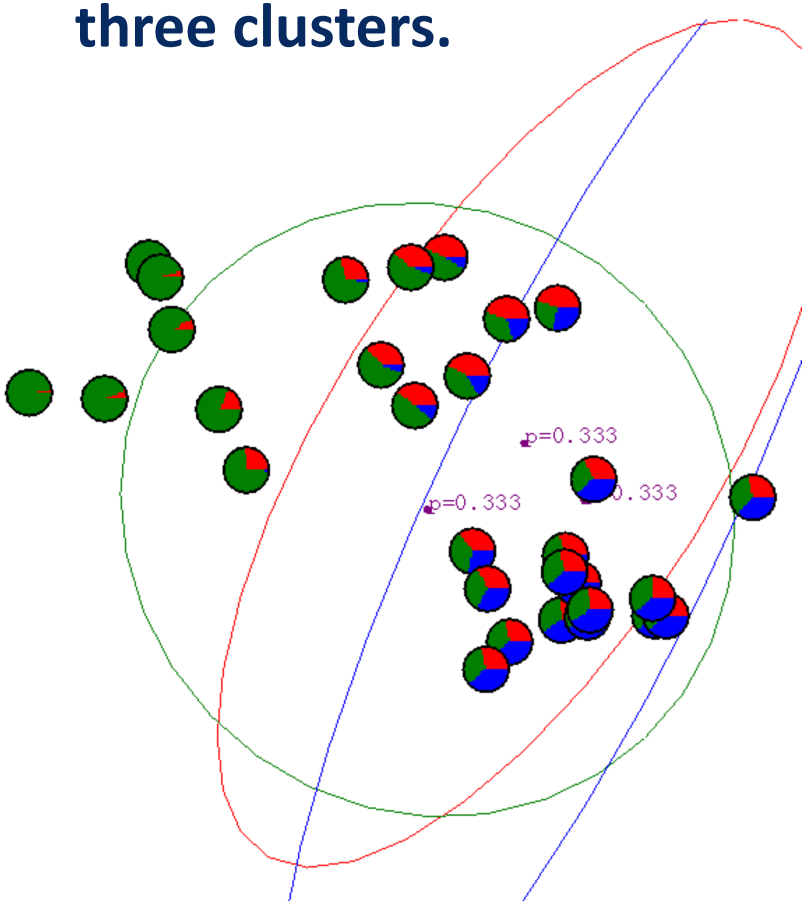
$$= \operatorname*{argmax}_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{k} p\left(y^{(j)} = i \mid \theta\right) p\left(x^{(i)} \mid y^{(j)} = i, \theta\right)$$

$$= \operatorname*{argmax}_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{k} \pi_i \frac{1}{\sqrt{2\pi |\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \mu_i)\right)$$

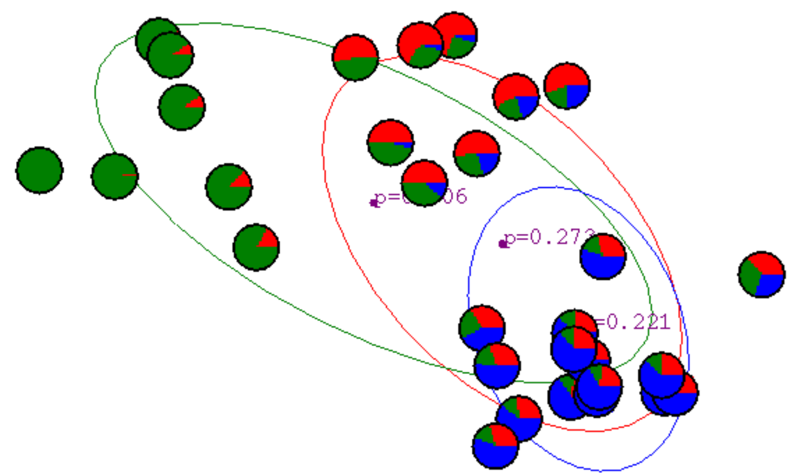
- **Although using the gradient descent method is possible, it is too slow to converge. ⇒ Use the EM algorithm.**

How does the EM Algorithm Work?

- At initialization, each sample has arbitrary weights for three clusters.



Iteration 1

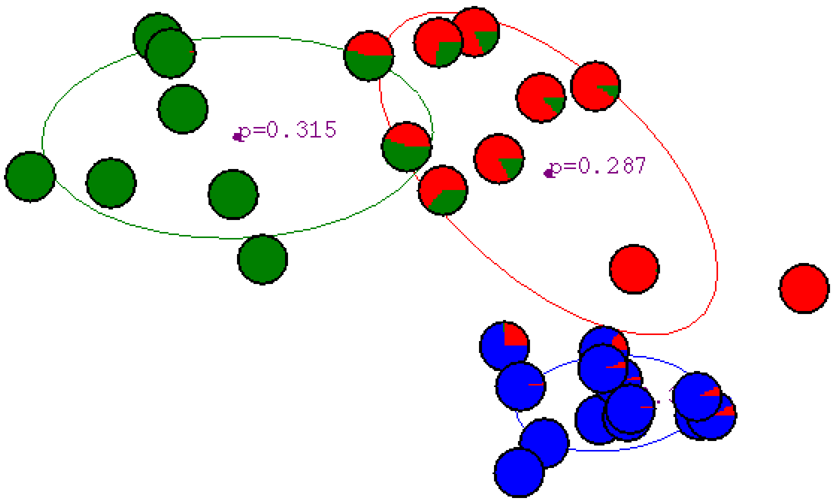


Iteration 2

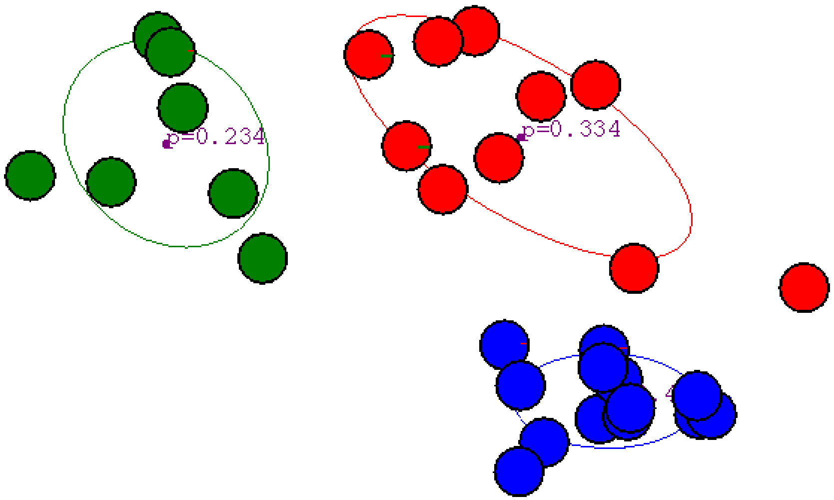
https://alliance.seas.upenn.edu/~cis520/dynamic/2022/wiki/index.php?n=Lectures.EM

How does the EM Algorithm Work?

- The weights of samples converge as the number of iterations increases.



Iteration 6



Iteration 20

Q&A

Alternating Optimization

Revisiting k-Means Clustering

- **Randomly initialize k centers.**

$$\boldsymbol{\mu}^{(0)} = \left(\boldsymbol{\mu}_1^{(0)}, \cdots, \boldsymbol{\mu}_k^{(0)}\right)$$

- **Assignment: At iteration t, assign each sample $j \in \{1, \ldots, n\}$ to the nearest center.**

$$C^{(t)}(j) \leftarrow \underset{i}{\operatorname{argmin}} \left\| \boldsymbol{\mu}_i^{(t)} - \mathbf{x}_j \right\|^2$$

Classify samples at iteration t

- **Recentering: update the center for each cluster.**

$$\boldsymbol{\mu}_i^{(t+1)} \leftarrow \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \sum_{j:C^{(t)}(j)=i} \left\| \boldsymbol{\mu} - \mathbf{x}_j \right\|^2$$

Reassign new centers at iteration t.

Optimization of k-Means Clustering

- **We define the potential function of centers $\boldsymbol{\mu}$ and point allocation C.**

$$F(\boldsymbol{\mu}, C) = \sum_{j=1}^{n} \left\| \boldsymbol{\mu}_{C(j)} - \mathbf{x}_j \right\|^2 = \sum_{i=1}^{k} \sum_{j:C(j)=i} \left\| \boldsymbol{\mu}_i - \mathbf{x}_j \right\|^2$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k), C = (C(1), \cdots, C(n))$

- **The optimal solution of the k-means clustering problem**

$$\min_{\boldsymbol{\mu}, C} F(\boldsymbol{\mu}, C)$$

Optimization of k-Means Clustering

Expectation step: Assign each sample to one of the centroids.

Fix $\boldsymbol{\mu}$, optimize C

$$\min_{C(1),\cdots,C(n)} \sum_{j=1}^{n} \left\|\boldsymbol{\mu}_{C(j)} - \mathbf{x}_j\right\|^2 = \sum_{j=1}^{n} \min_{C(j)} \left\|\boldsymbol{\mu}_{C(j)} - \mathbf{x}_j\right\|^2$$

Optimize C by fixing $\boldsymbol{\mu}$.

Maximization step: Update the centroid for each cluster.

Fix C, optimize $\boldsymbol{\mu}$

$$\min_{\boldsymbol{\mu}_1,\cdots,\boldsymbol{\mu}_k} \sum_{i=1}^{k} \sum_{j:C(j)=i} \left\|\boldsymbol{\mu}_i - \mathbf{x}_j\right\|^2 = \sum_{i=1}^{k} \min_{\boldsymbol{\mu}_i} \sum_{j:C(j)=i} \left\|\boldsymbol{\mu}_i - \mathbf{x}_j\right\|^2$$

Optimize $\boldsymbol{\mu}$ by fixing C.

Expectation-Maximization Algorithm

- The EM algorithm is a generalization of this approach.

- We can understand k-means clustering as the EM algorithm.
  - **Expectation**: Optimize C by fixing $\boldsymbol{\mu}$.
  - **Maximization**: Optimize $\boldsymbol{\mu}$ by fixing C.

- We investigate the **Gaussian Mixture model** using the EM algorithm.