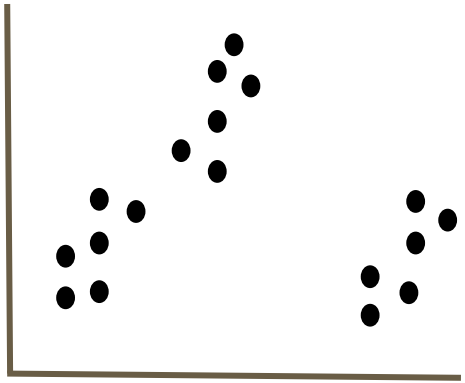
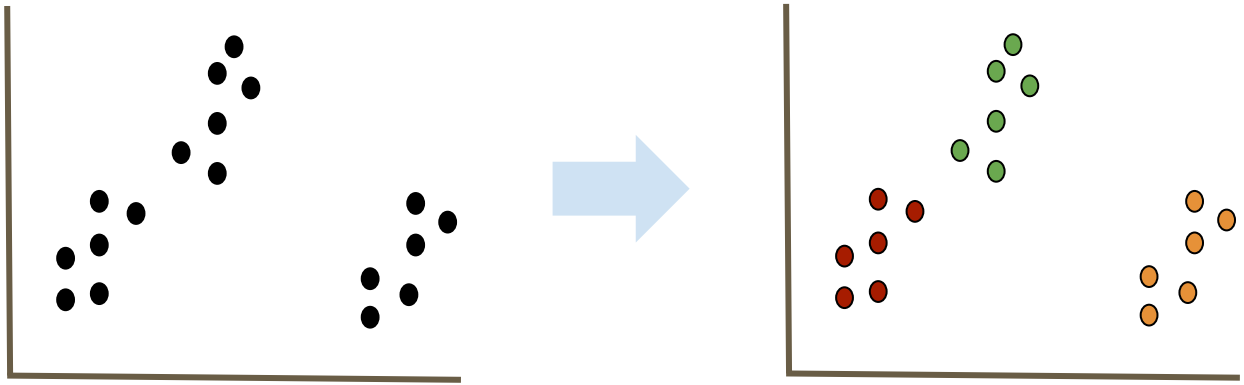
Clustering - Kmeans

Boston University CS 506 - Lance Galletti

What is a Clustering



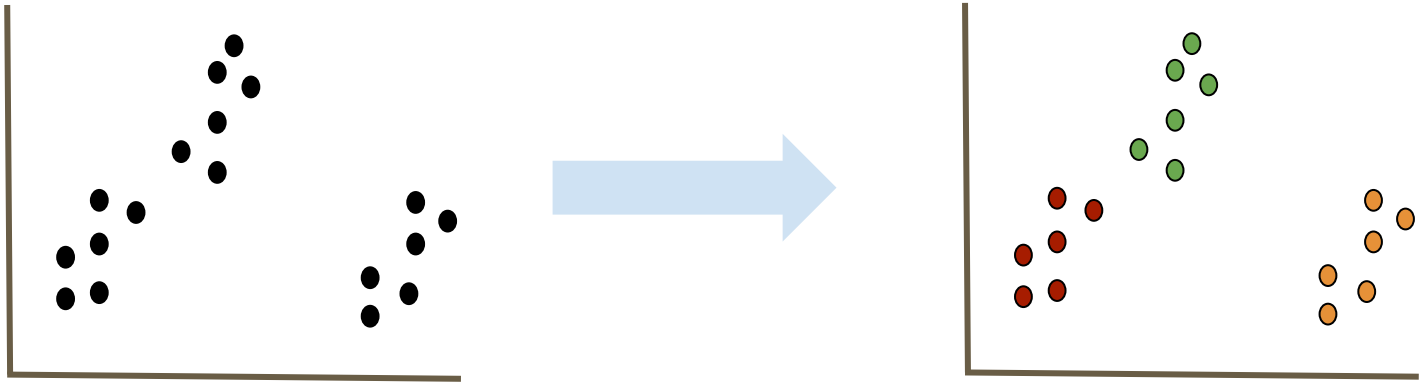
What is a Clustering



What is a Clustering

A clustering is a grouping / assignment of objects (data points) such that objects in the same group / cluster are:

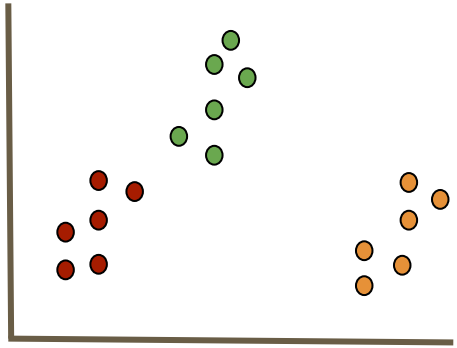
- similar to one another
- dissimilar to objects in other groups

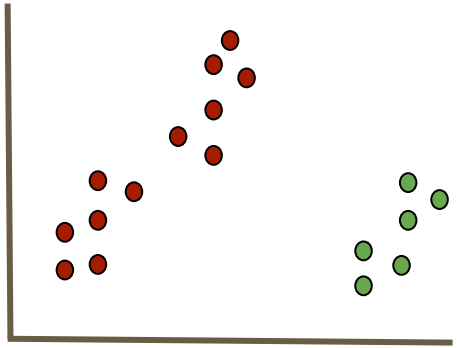


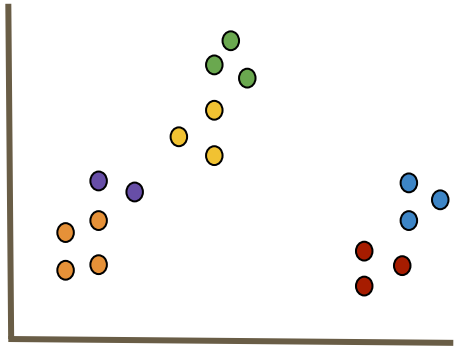
Applications

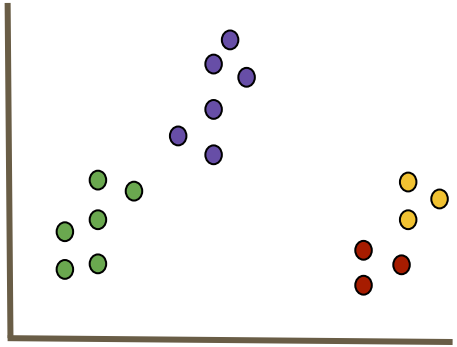
- Outlier detection / anomaly detection
  - Data Cleaning / Processing
  - Credit card fraud, spam filter etc.
- Feature Extraction
- Filling Gaps in your data
  - Using the same marketing strategy for similar people
  - Infer probable values for gaps in the data (similar users could have similar hobbies, likes / dislikes etc.)

Clusters can be Ambiguous









Types of Clusterings

Partitional **Making k partitions**

Each object belongs to exactly one cluster

Hierarchical

A set of nested clusters organized in a tree

Density-Based **Making a notion of closeness(density)**

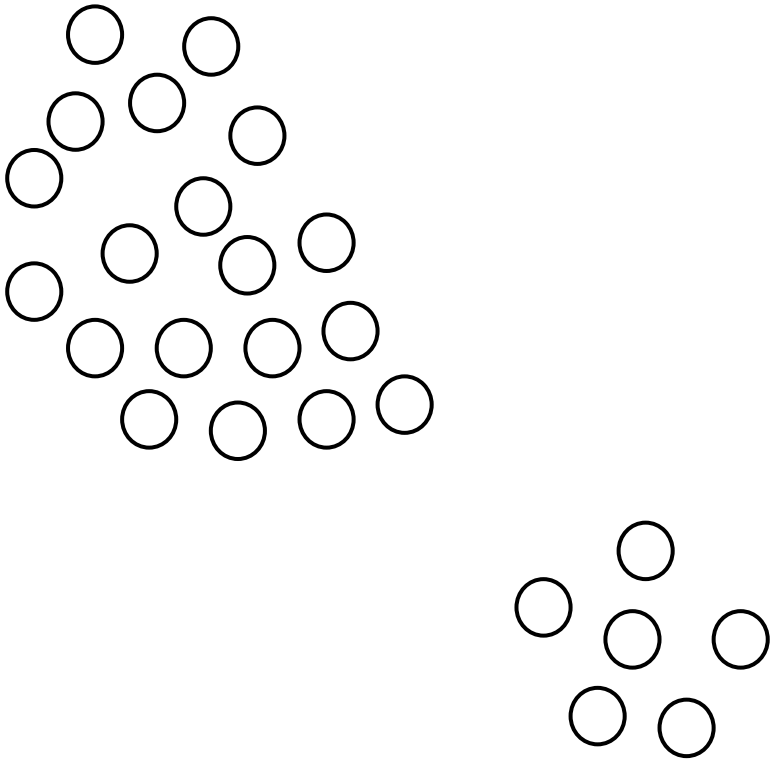
Defined based on the local density of points

Soft Clustering

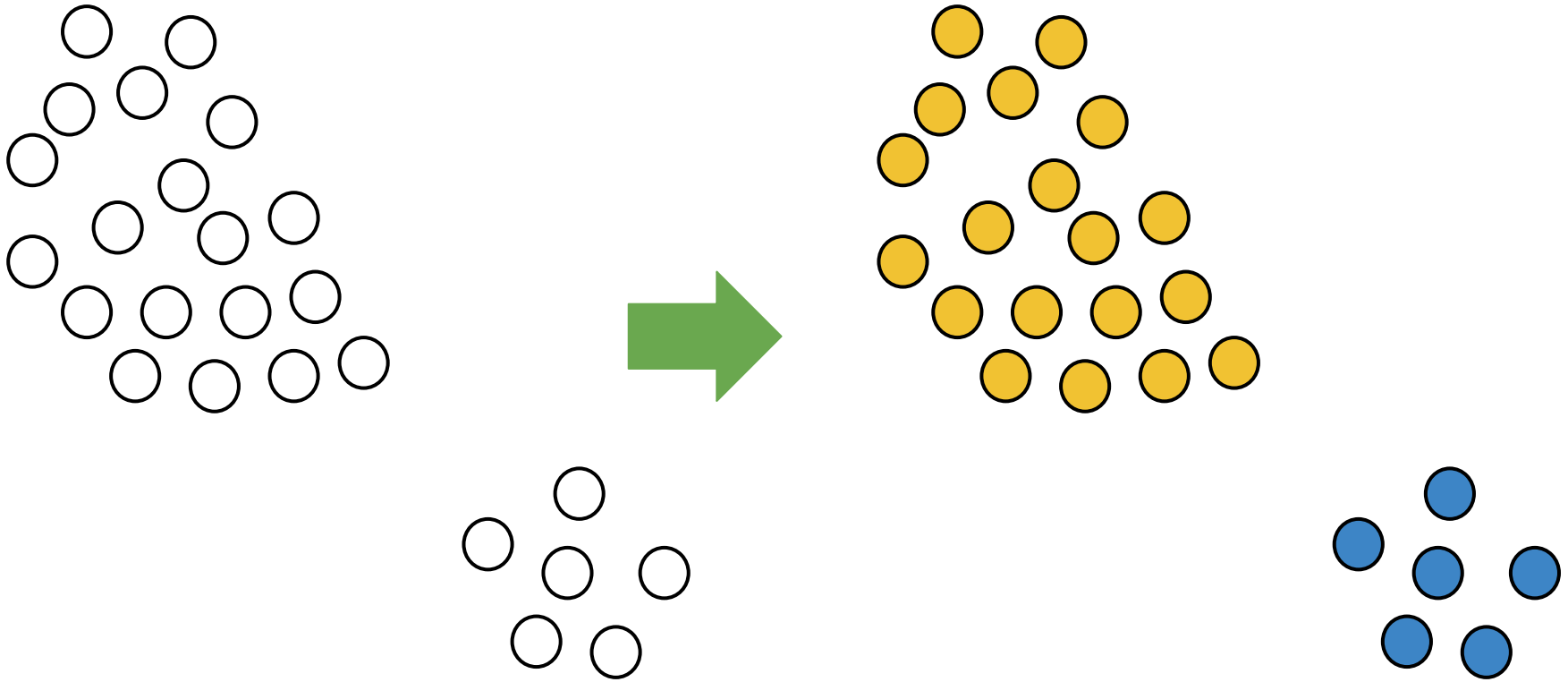
Each point is assigned to every cluster with a certain probability

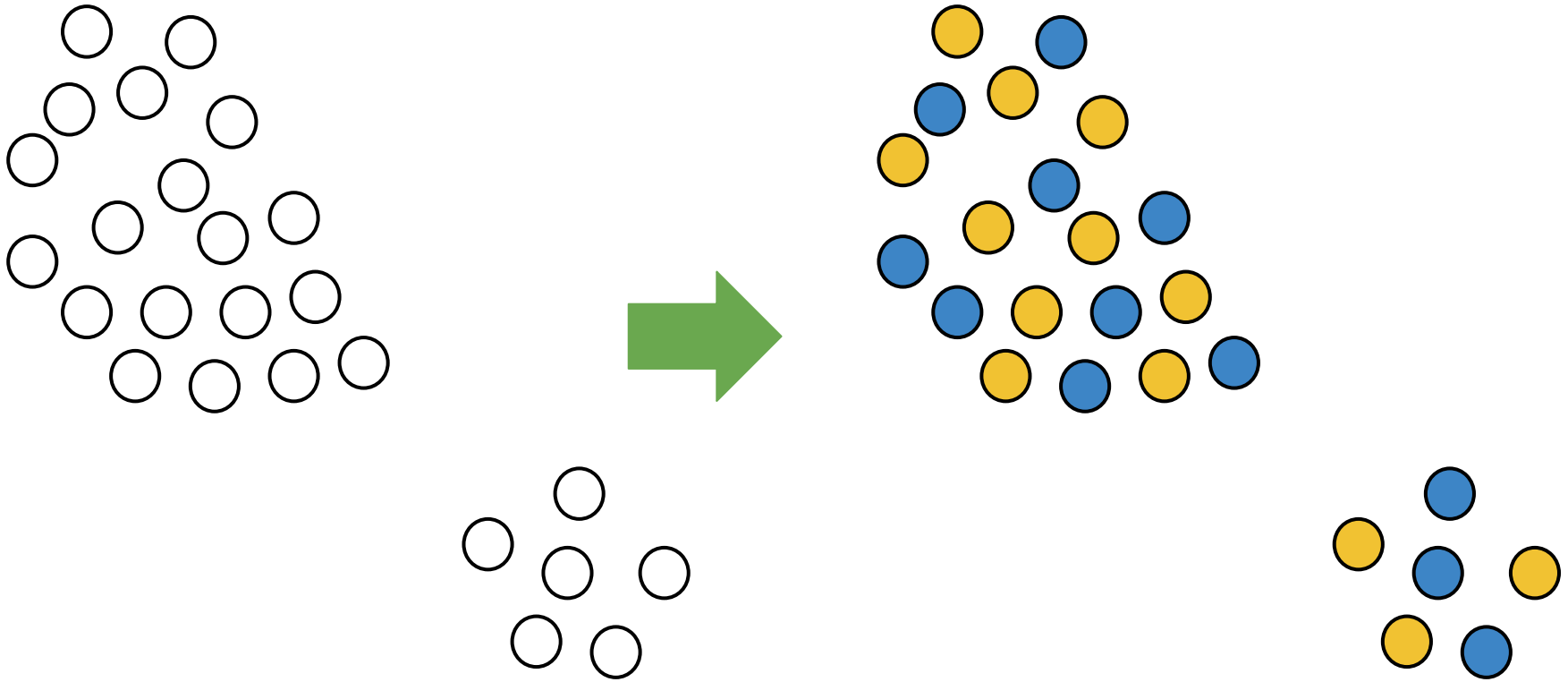
Partitional Clustering

Partitional Clustering

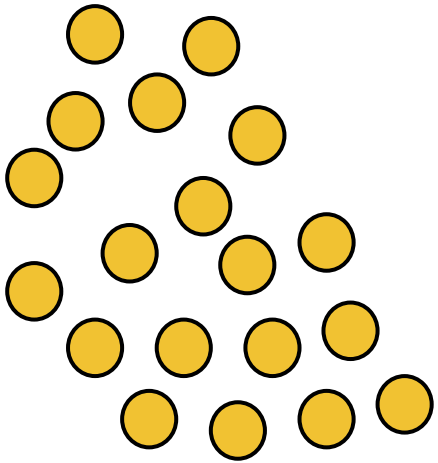


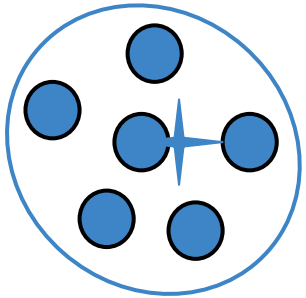
Goal: partition dataset into k partitions

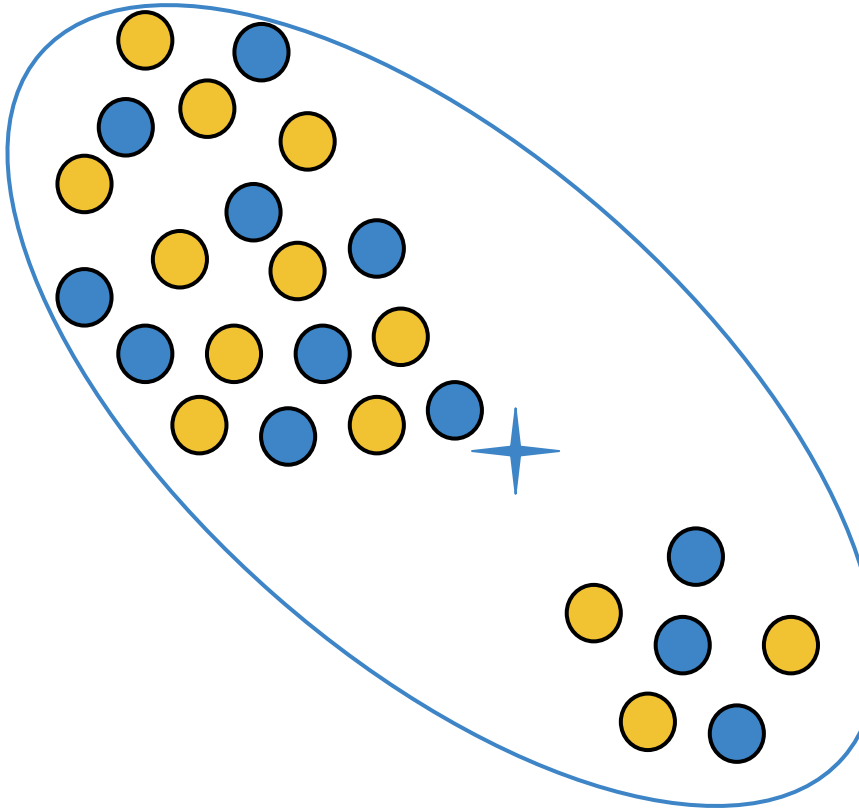




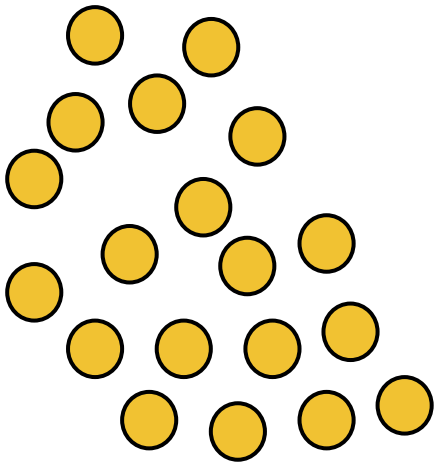
Similar things in the same cluster
Cluster with the smallest variance possible

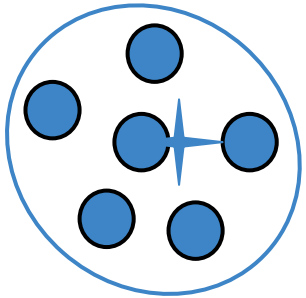


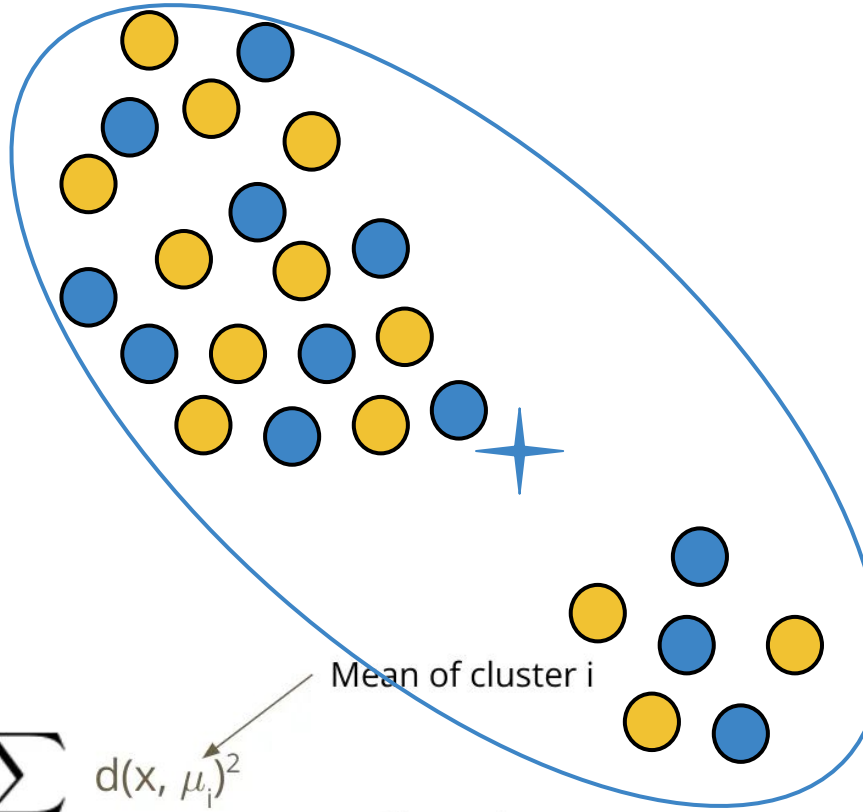




cost function: bigger value for the points that are spread







Mean of cluster i

$$\frac{1}{|C_i|}\sum_{x \in C_i} d(x, \mu_i)^2$$

Cluster i

Cost Function

take a distance between point and the mean and square it then, sum all up

$$\sum_{i}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

for every point in that cluster and sum all the value from each cluster

- Way to evaluate and compare solutions
- Hope: can find some algorithm that find solutions that make the cost small

K-means

Given **X = {x_1, … , x_n}** our dataset, **d** the euclidean distance, and **k**

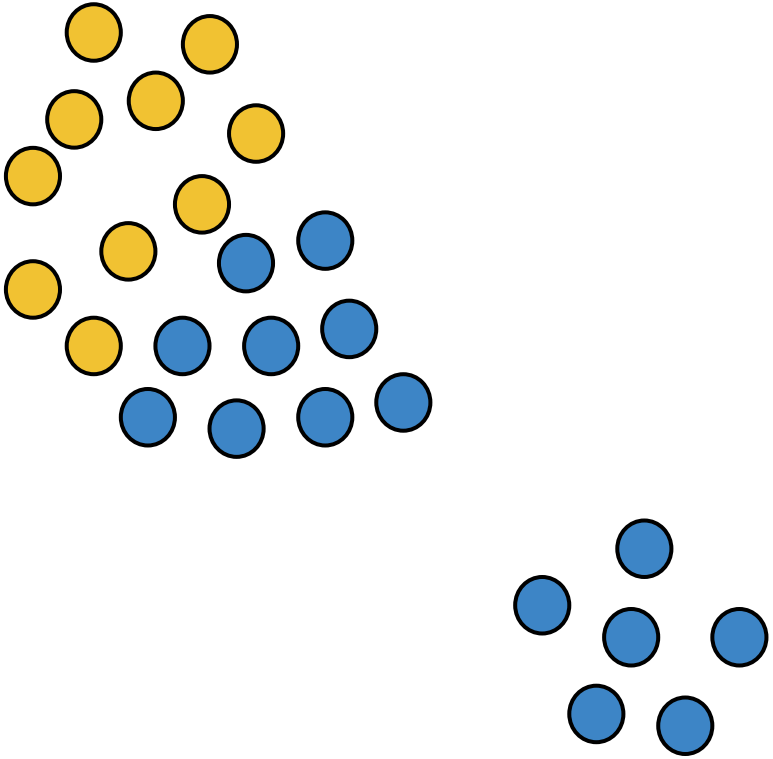
Find **k** centers **{μ_1, … , μ_k}** that minimize the **cost function**:

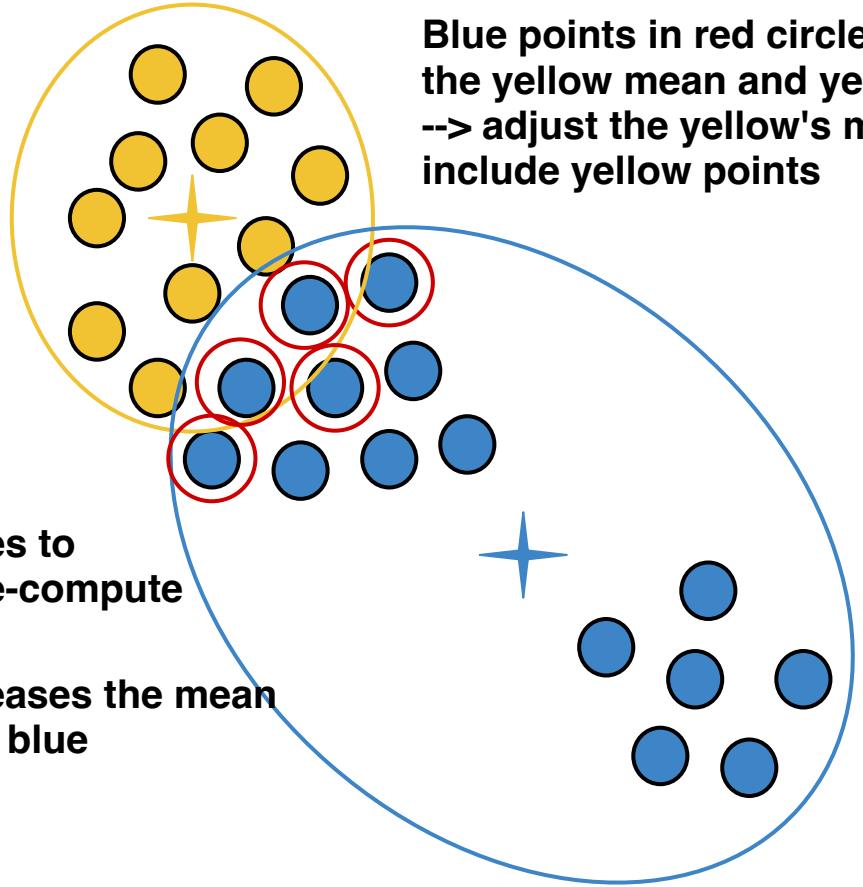
$$\sum_{i}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

When **k=1** and **k=n** this is easy. Why?

one cluster and separate cluster for every point

When **x_i** lives in more than 2 dimensions, this is a very difficult (**NP-hard**) problem



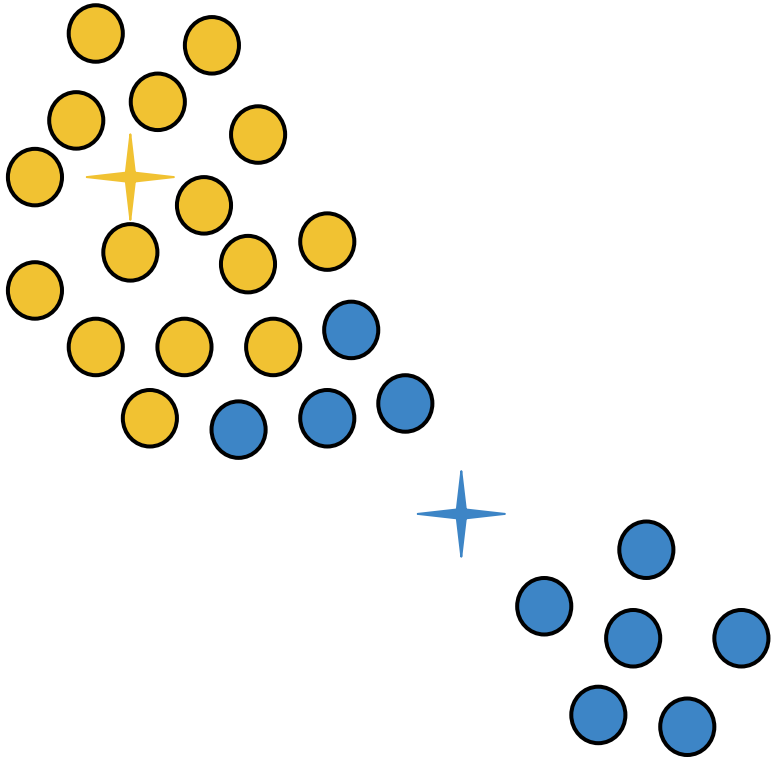


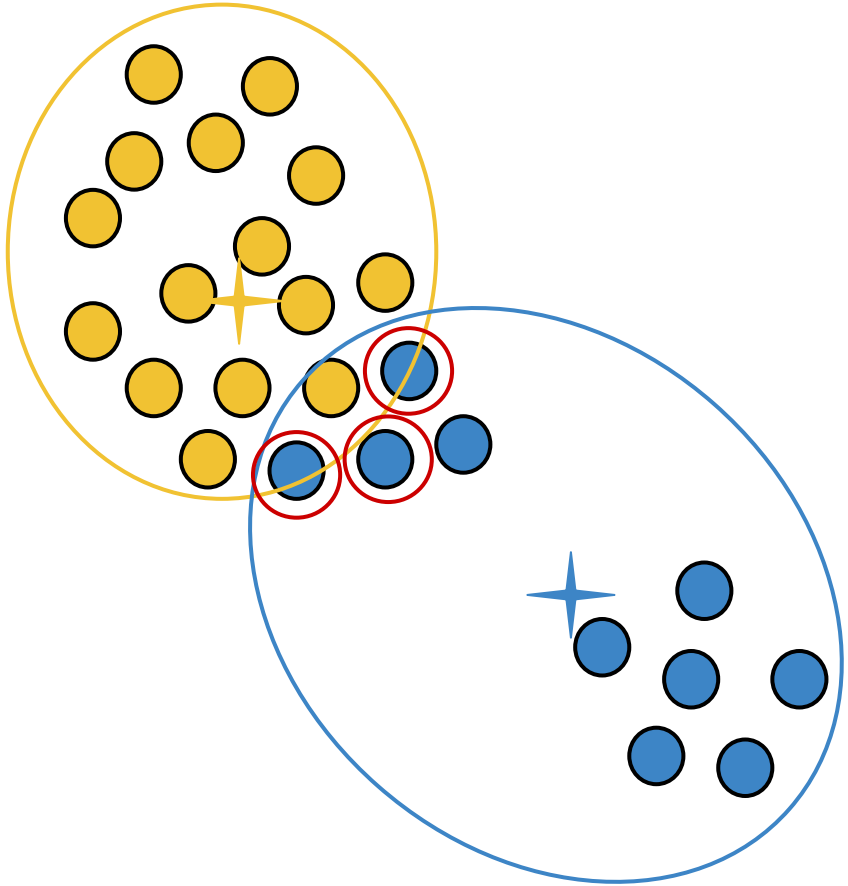
Blue points in red circles are closer to the yellow mean and yellow cluster --> adjust the yellow's mean to only include yellow points

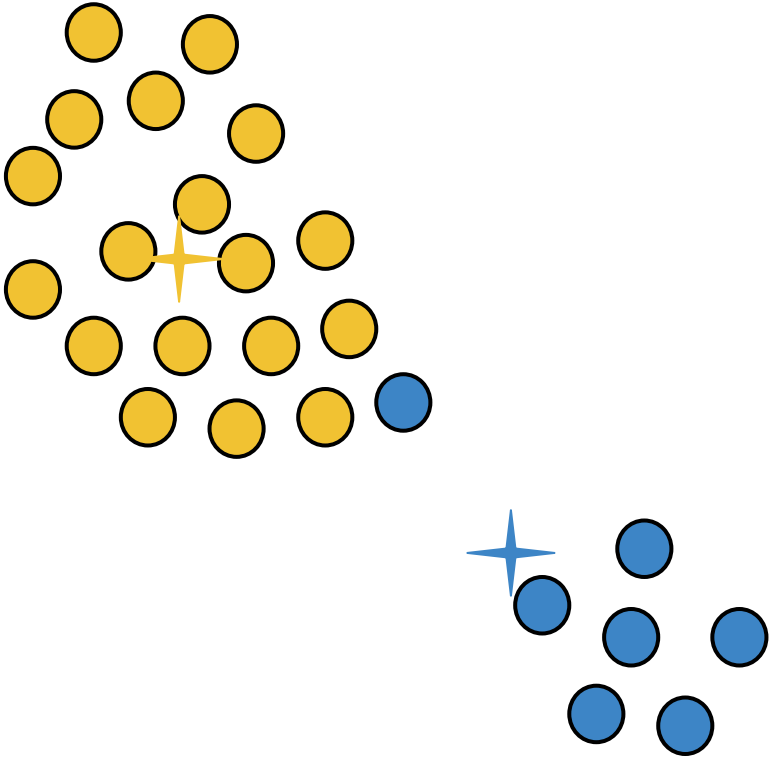
Move blue points in red circles to the yellow cluster and then re-compute the mean

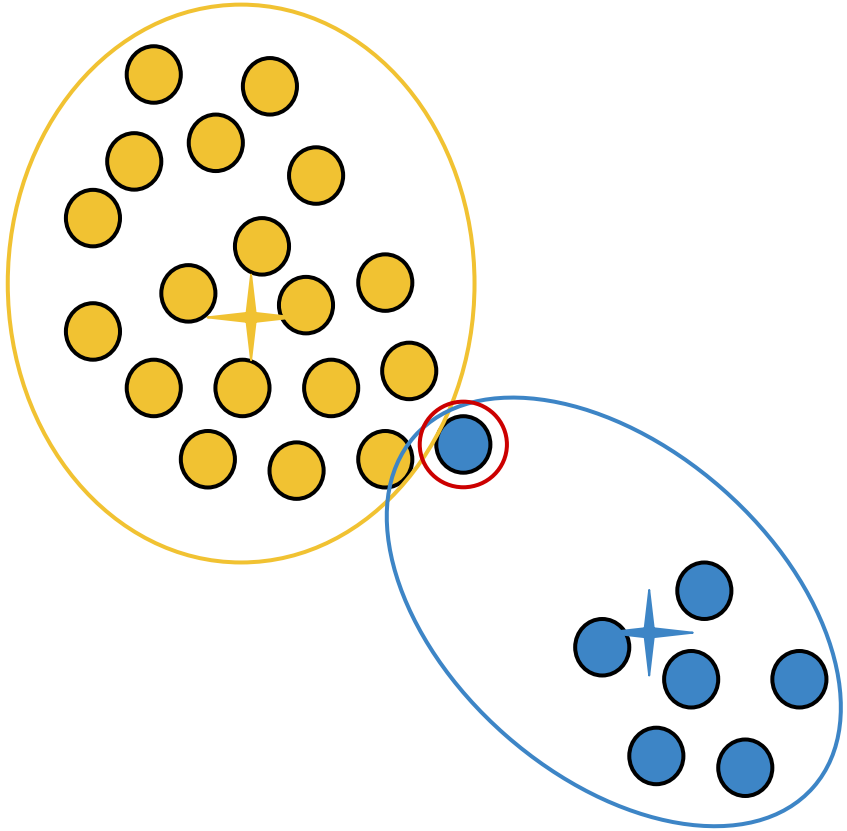
Recomputing the mean: increases the mean for yellow and decrease it for blue

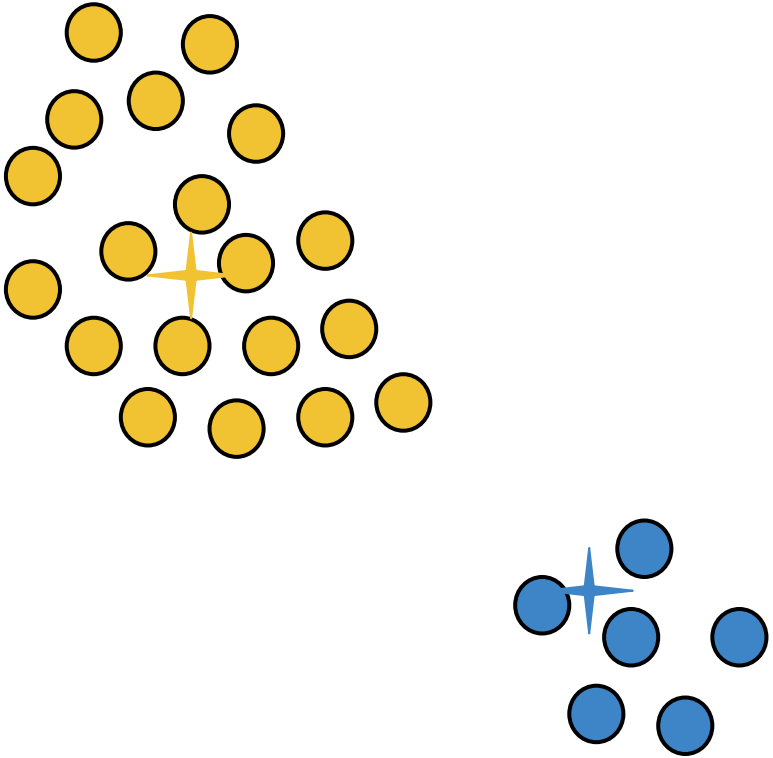
increase the mean for yellow and decrease it for blue





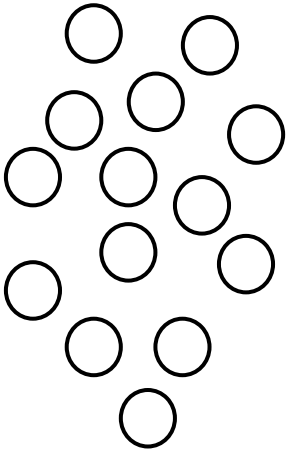


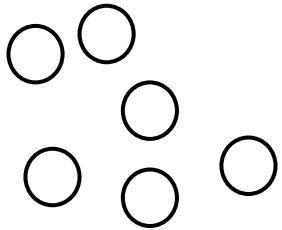


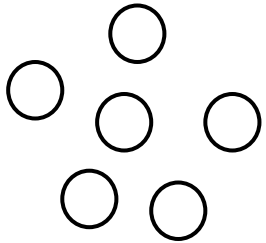


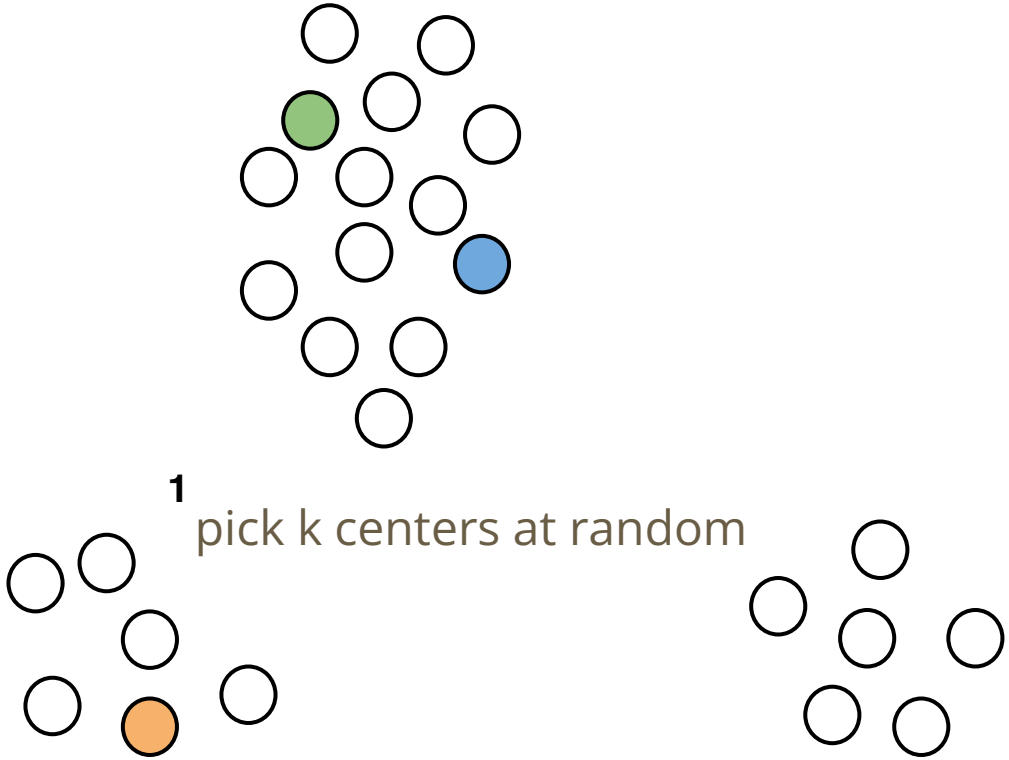
K-means - Lloyd's Algorithm

1. Randomly pick **k** centers $\{\mu_1, \dots, \mu_k\}$
2. Assign each point in the dataset to its closest center
3. Compute the new centers as the means of each cluster
4. Repeat 2 & 3 until convergence

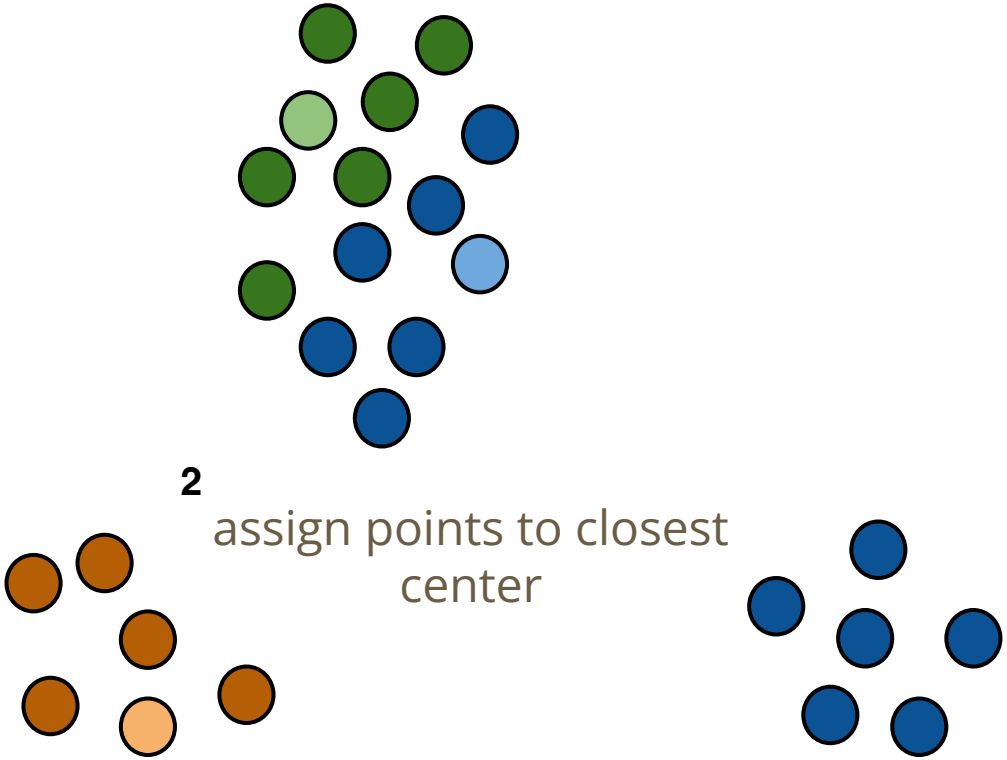


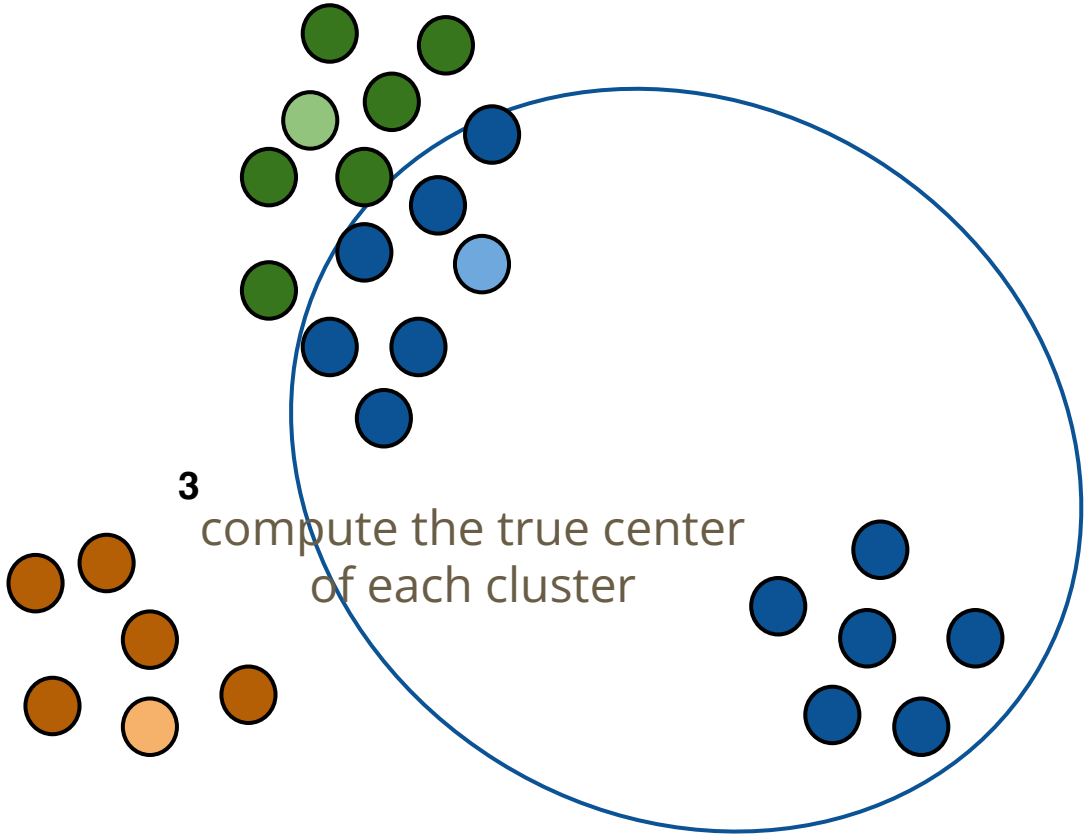


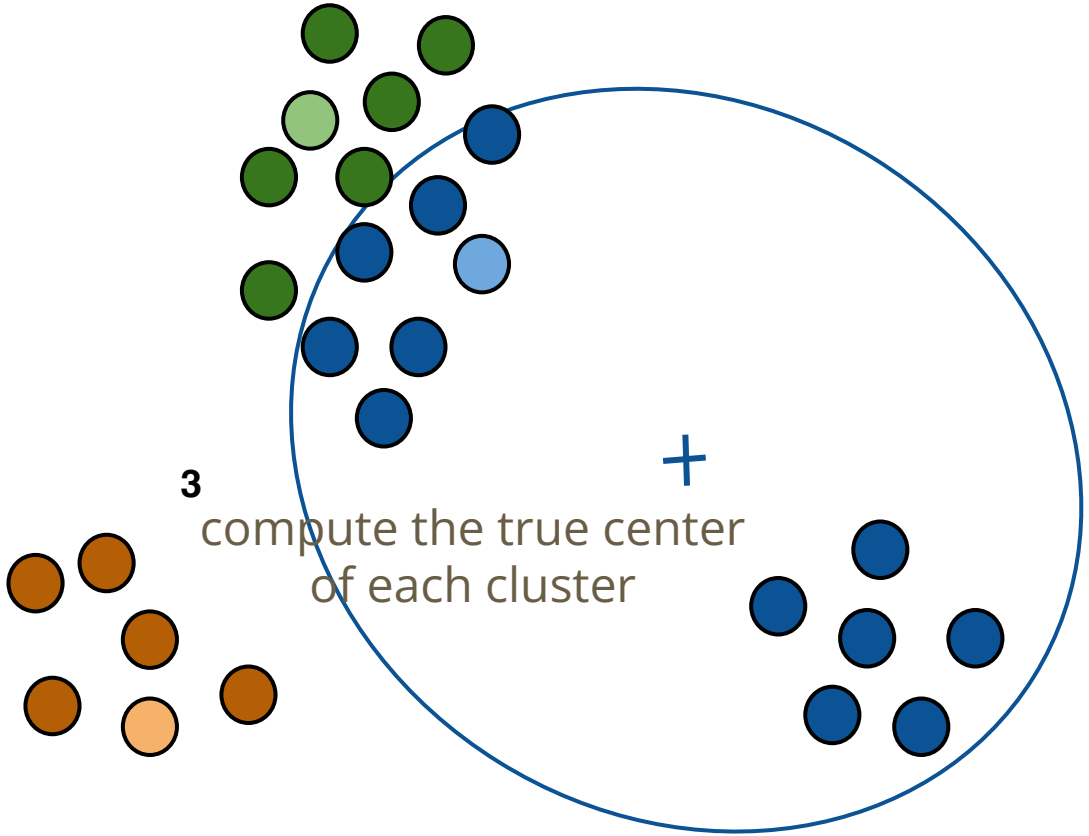




1 pick k centers at random

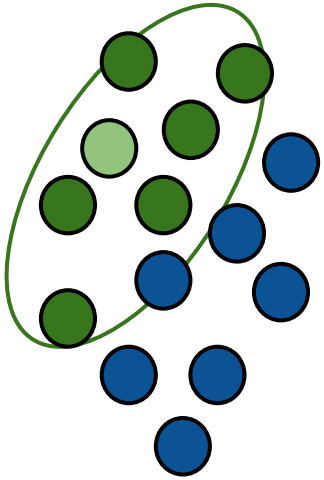

2
assign points to closest
center


3
compute the true center
of each cluster



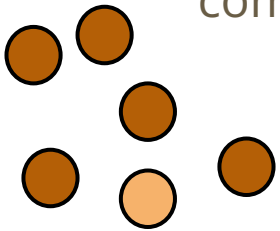
3

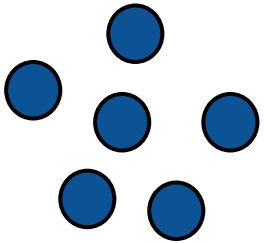
compute the true center
of each cluster

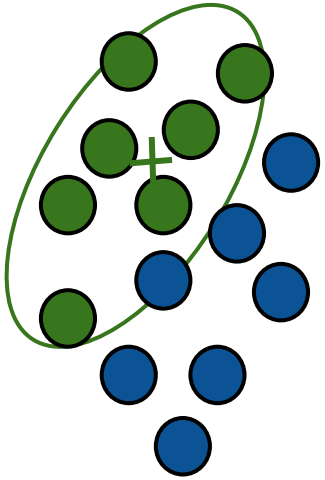


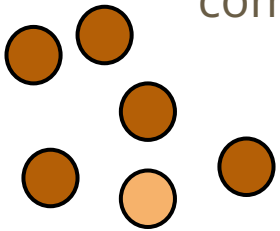
3

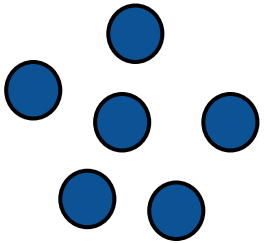
compute the true center
of each cluster





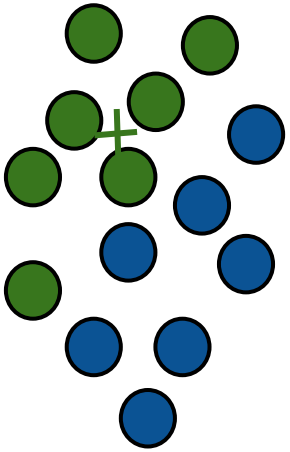


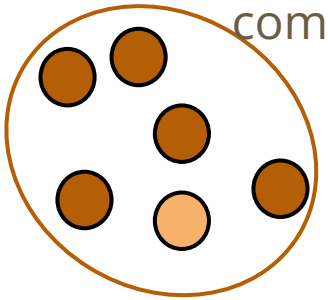




3

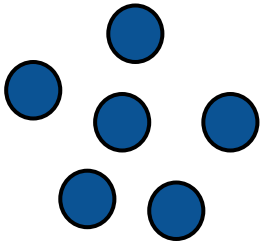
compute the true center
of each cluster

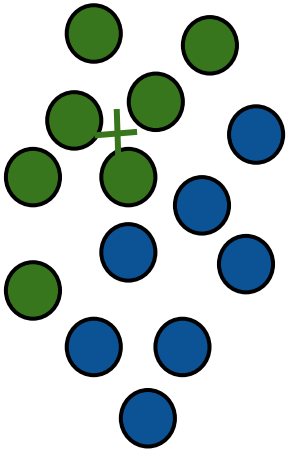




3

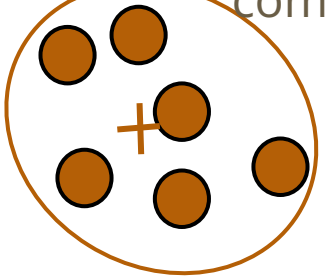
compute the true center
of each cluster

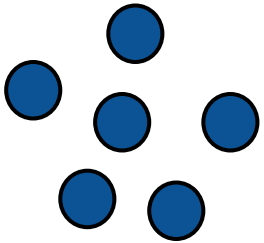


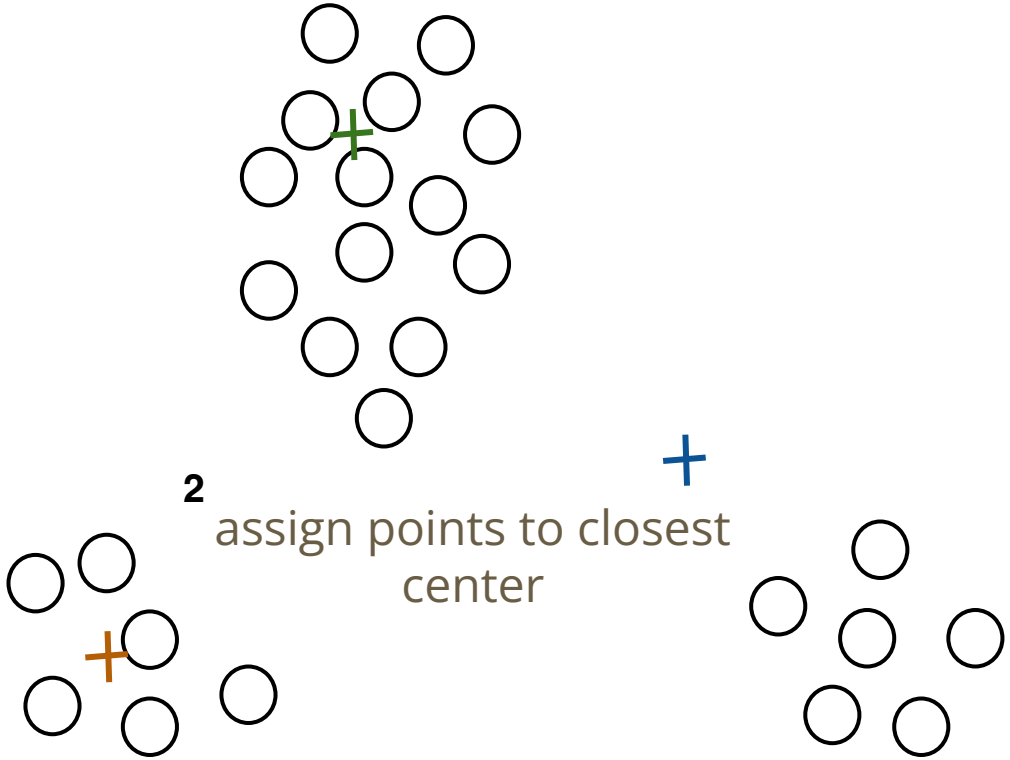


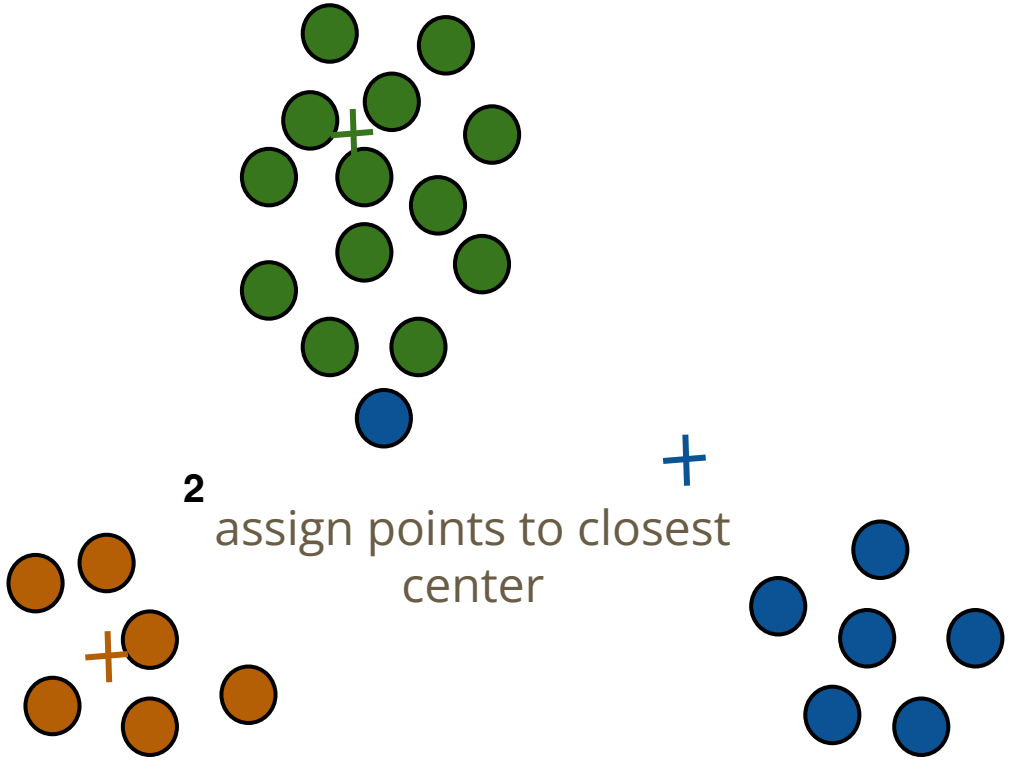
3

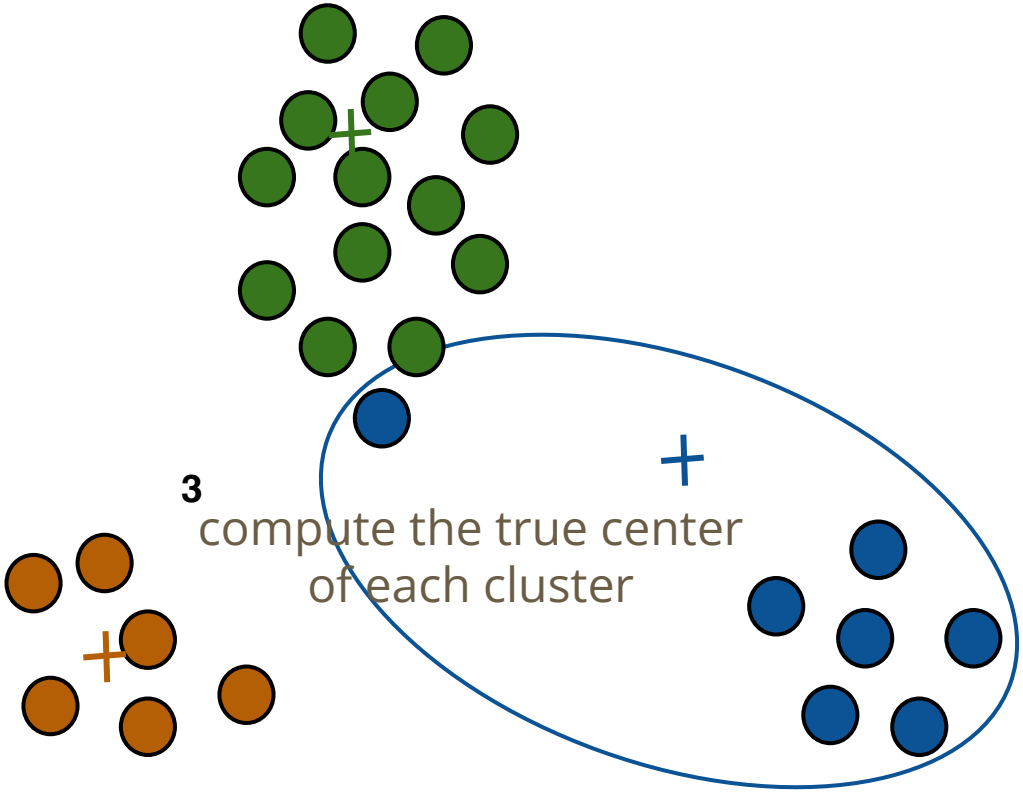
compute the true center
of each cluster

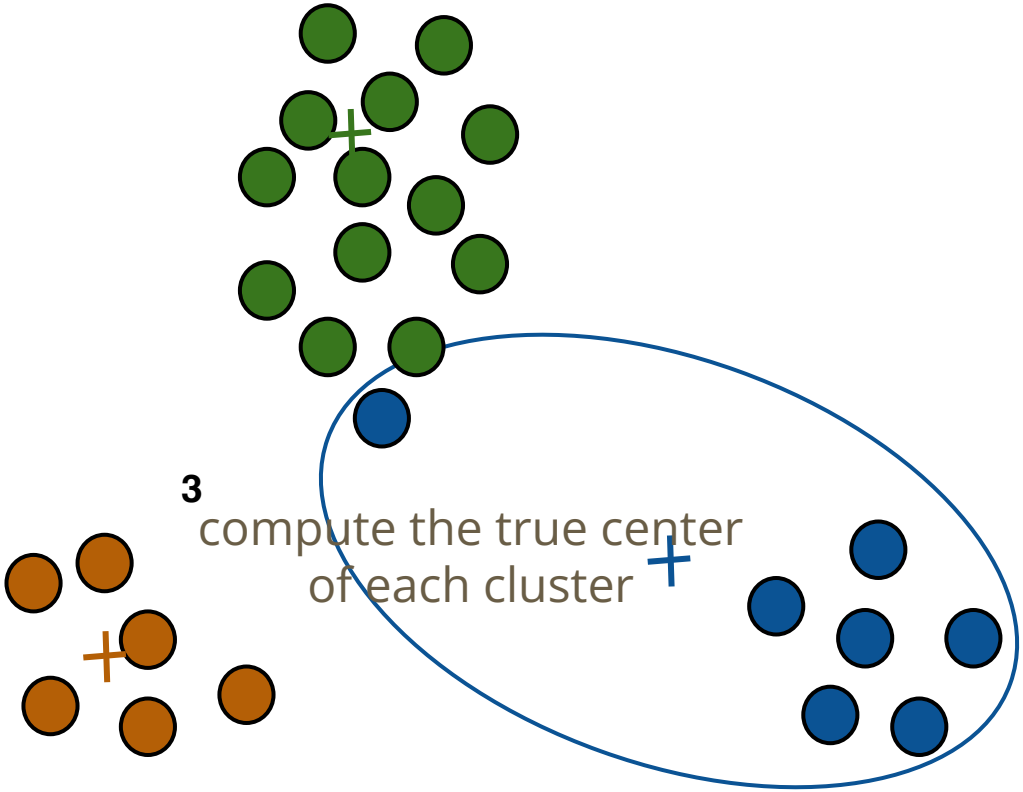


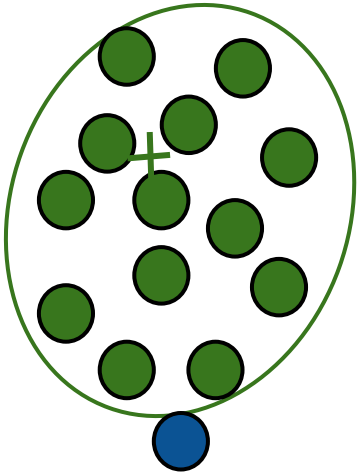



2
assign points to closest
center


2
assign points to closest
center

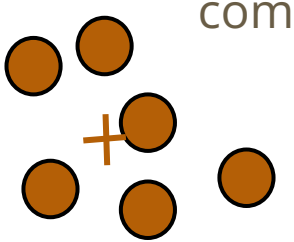

3
compute the true center
of each cluster

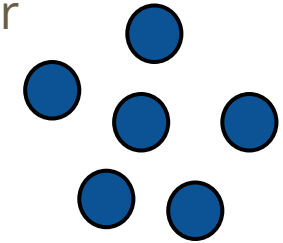

3
compute the true center
of each cluster

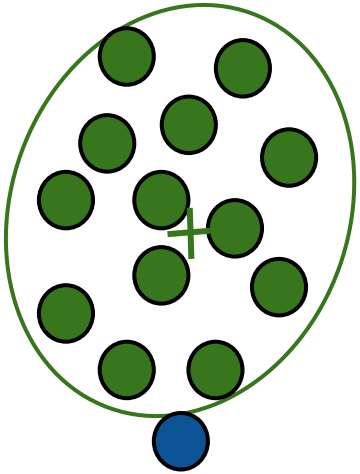


3

compute the true center of each cluster

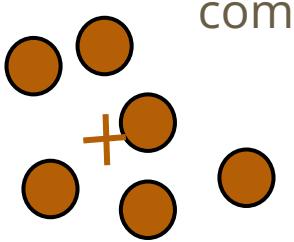


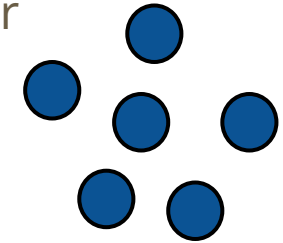


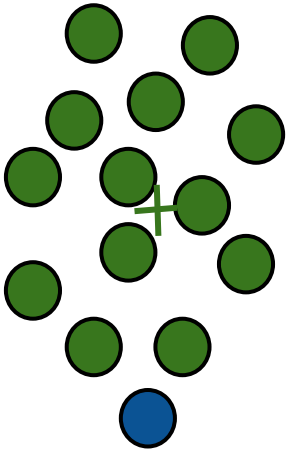


3

compute the true center of each cluster

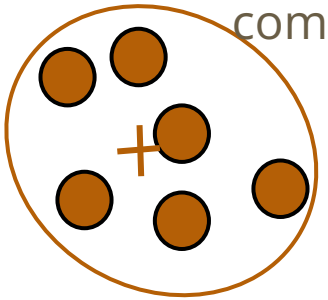


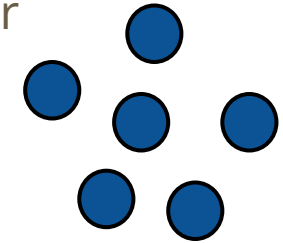


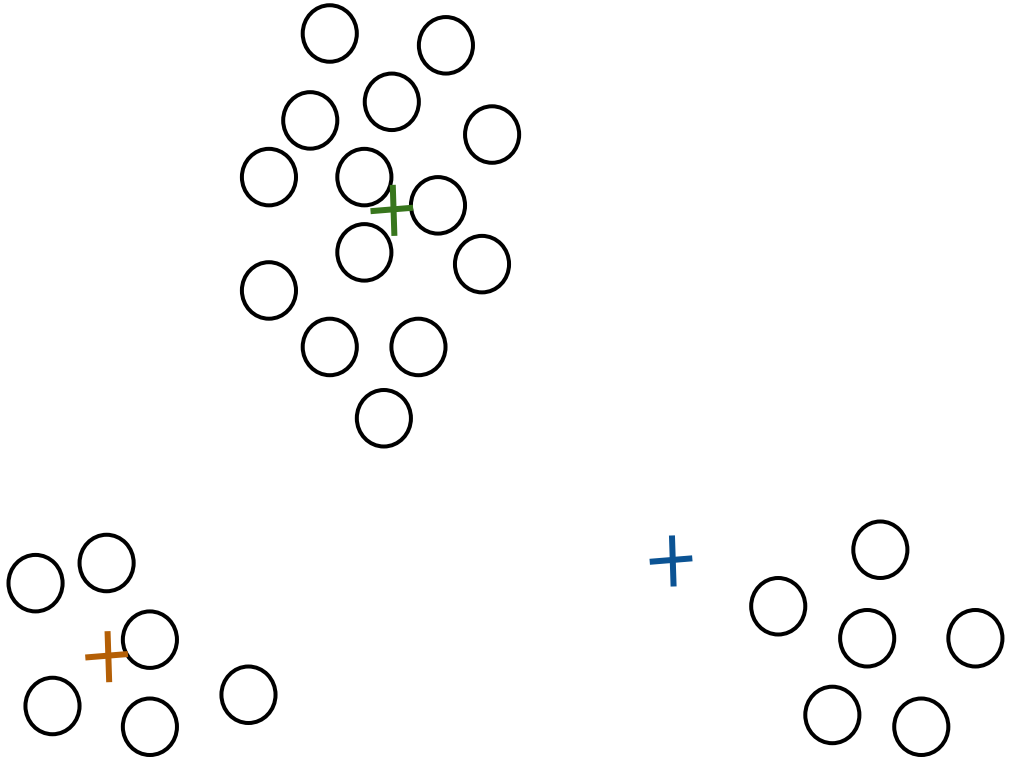


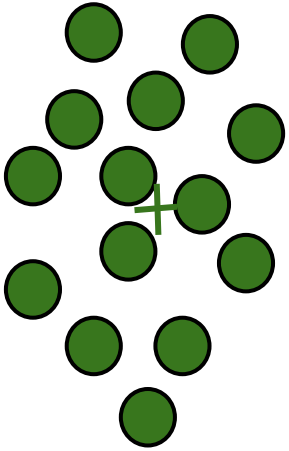
3

compute the true center of each cluster

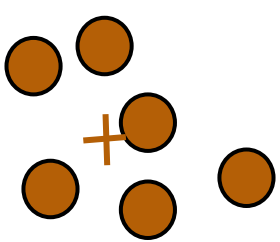


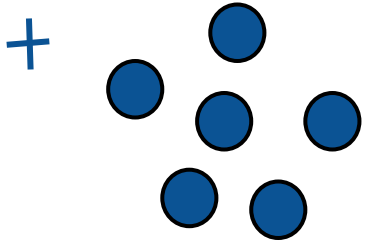


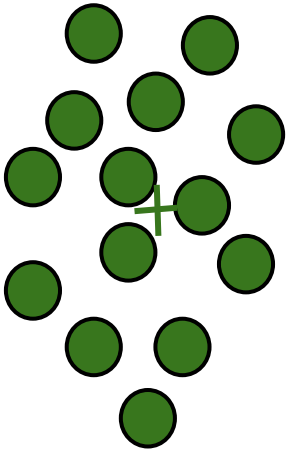




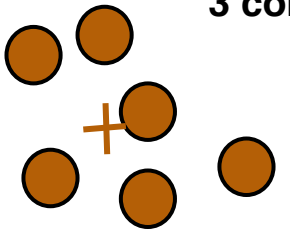
2 assign points to the closest center

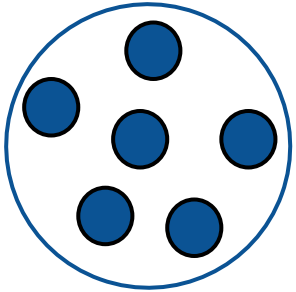


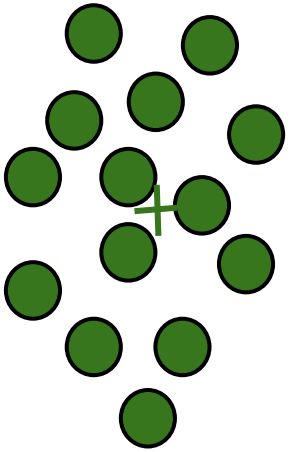




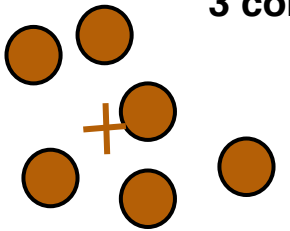
3 compute true center

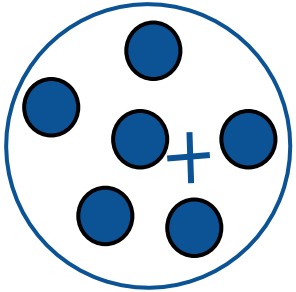


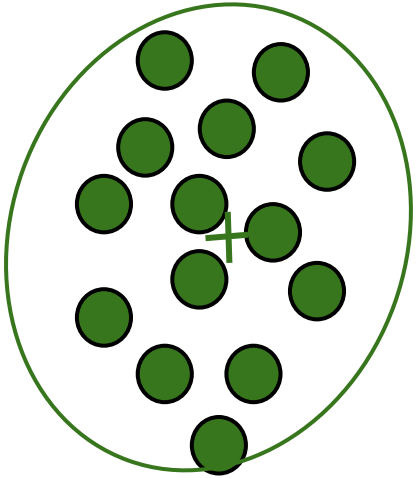




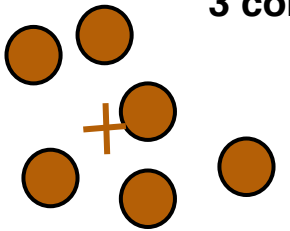
3 compute true center

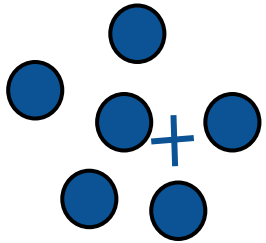


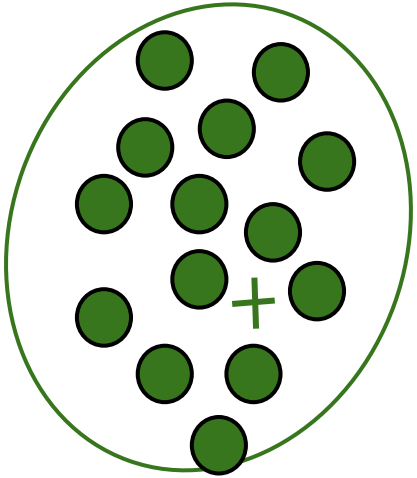




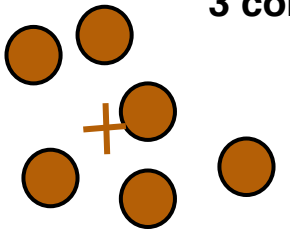
3 compute true center

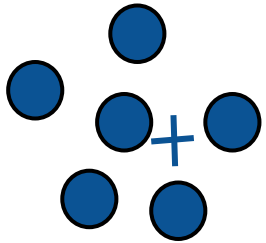


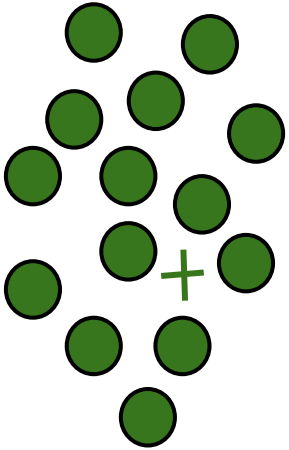


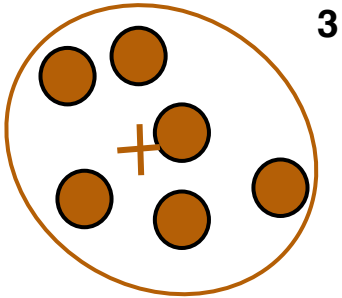


3 compute true center

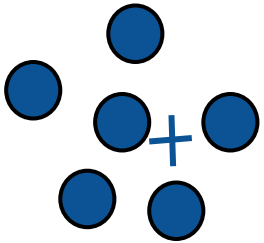


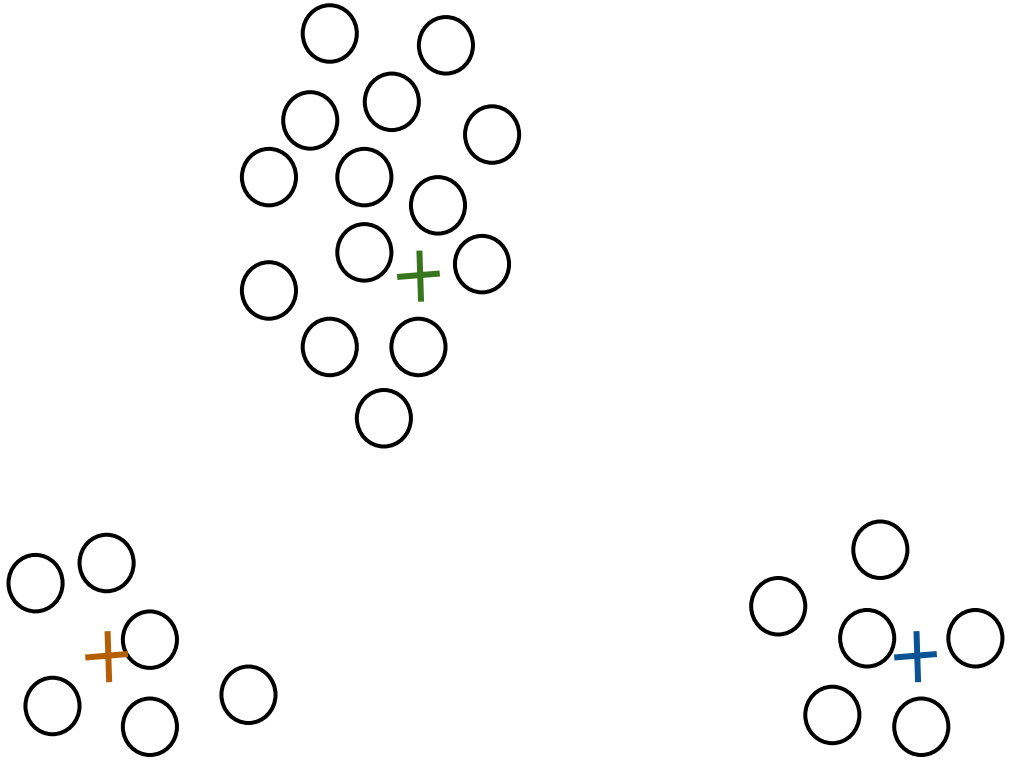


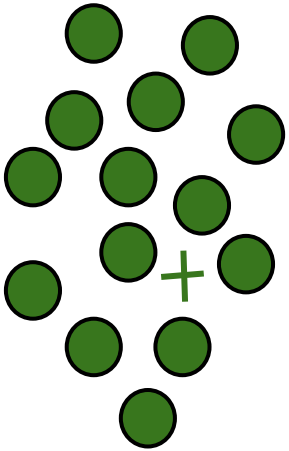




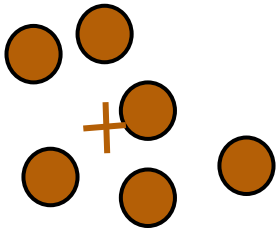
3 compute true center

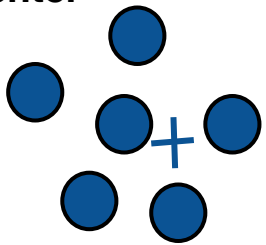


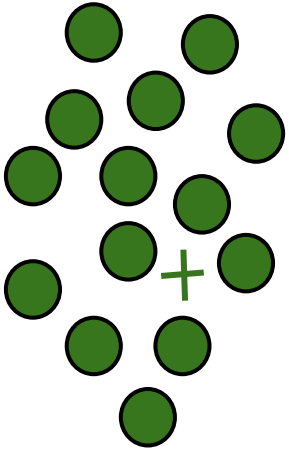




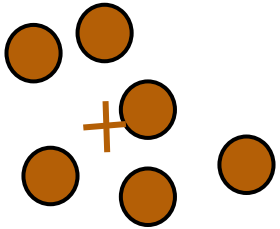
2 assign points to the closest center

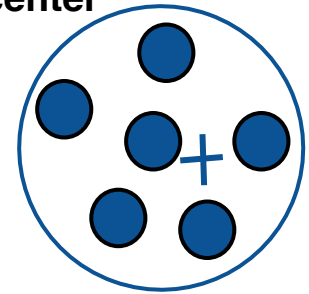


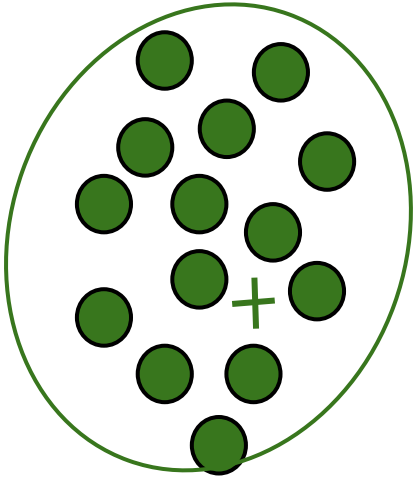




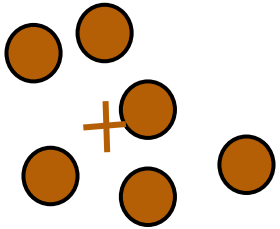
2 assign points to the closest center

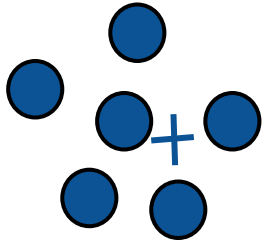


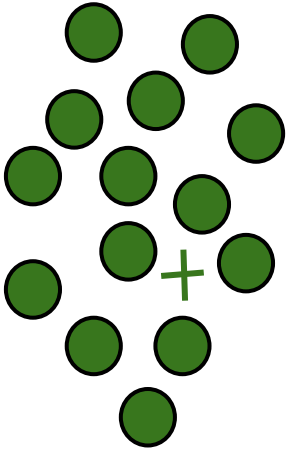




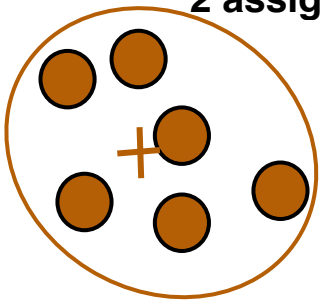
2 assign points to the closest center

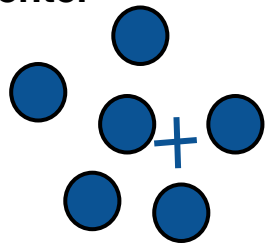


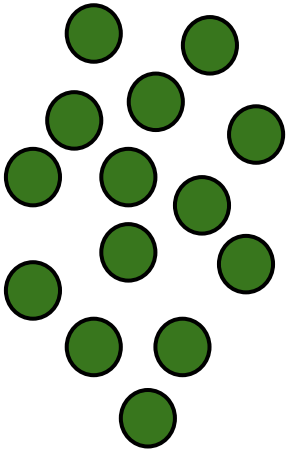




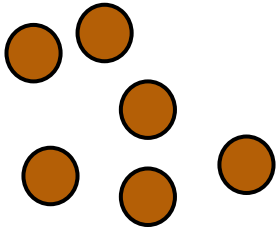
2 assign points to the closest center

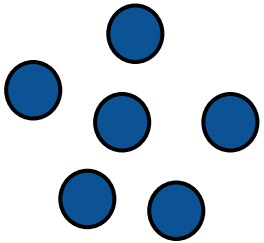




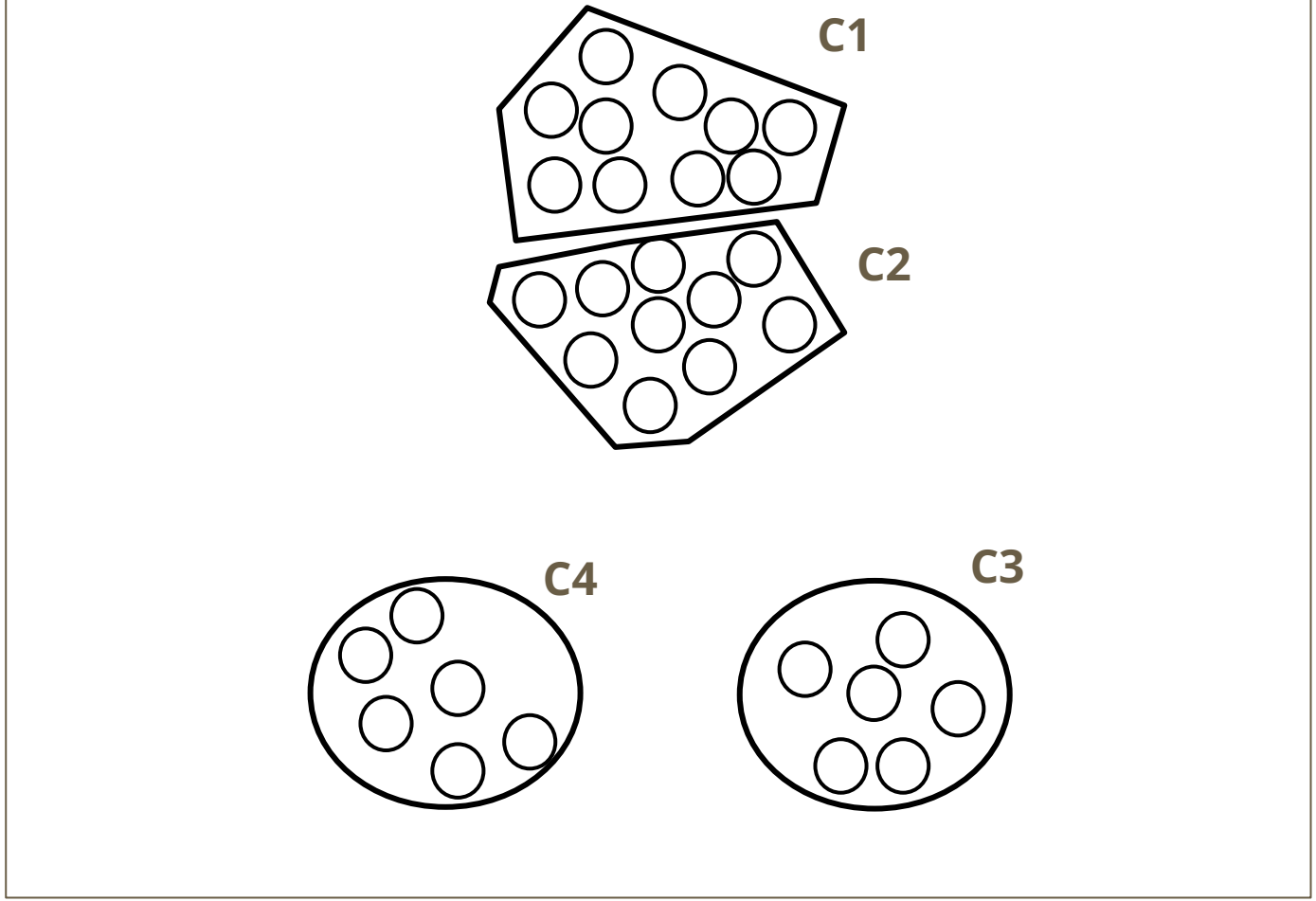


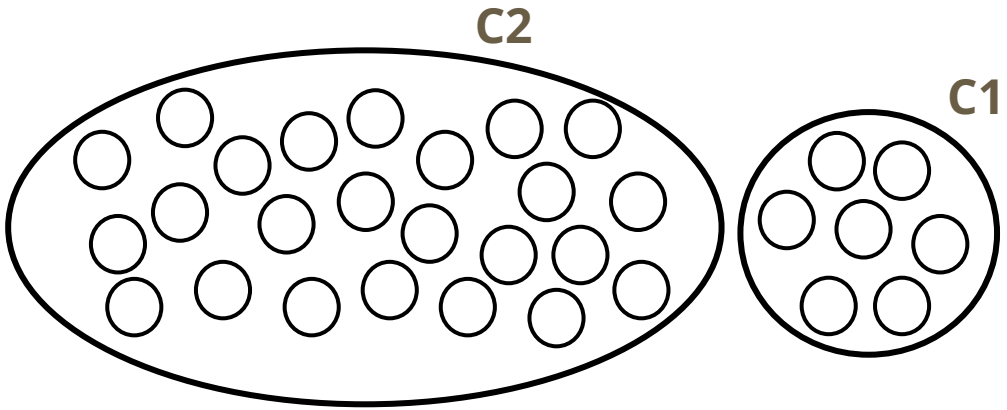
Converges

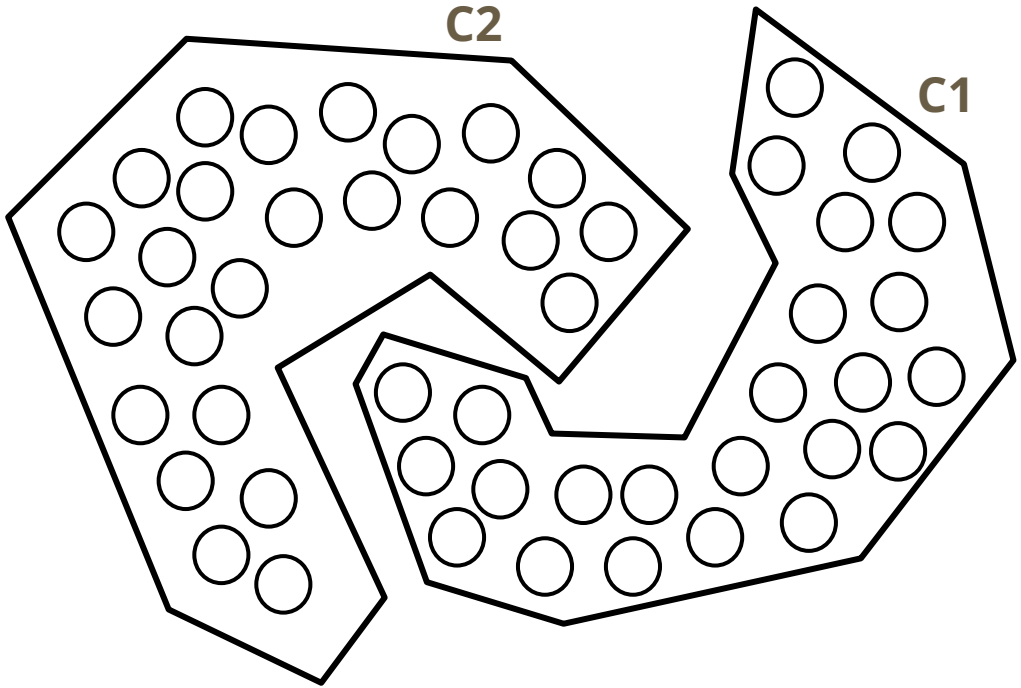


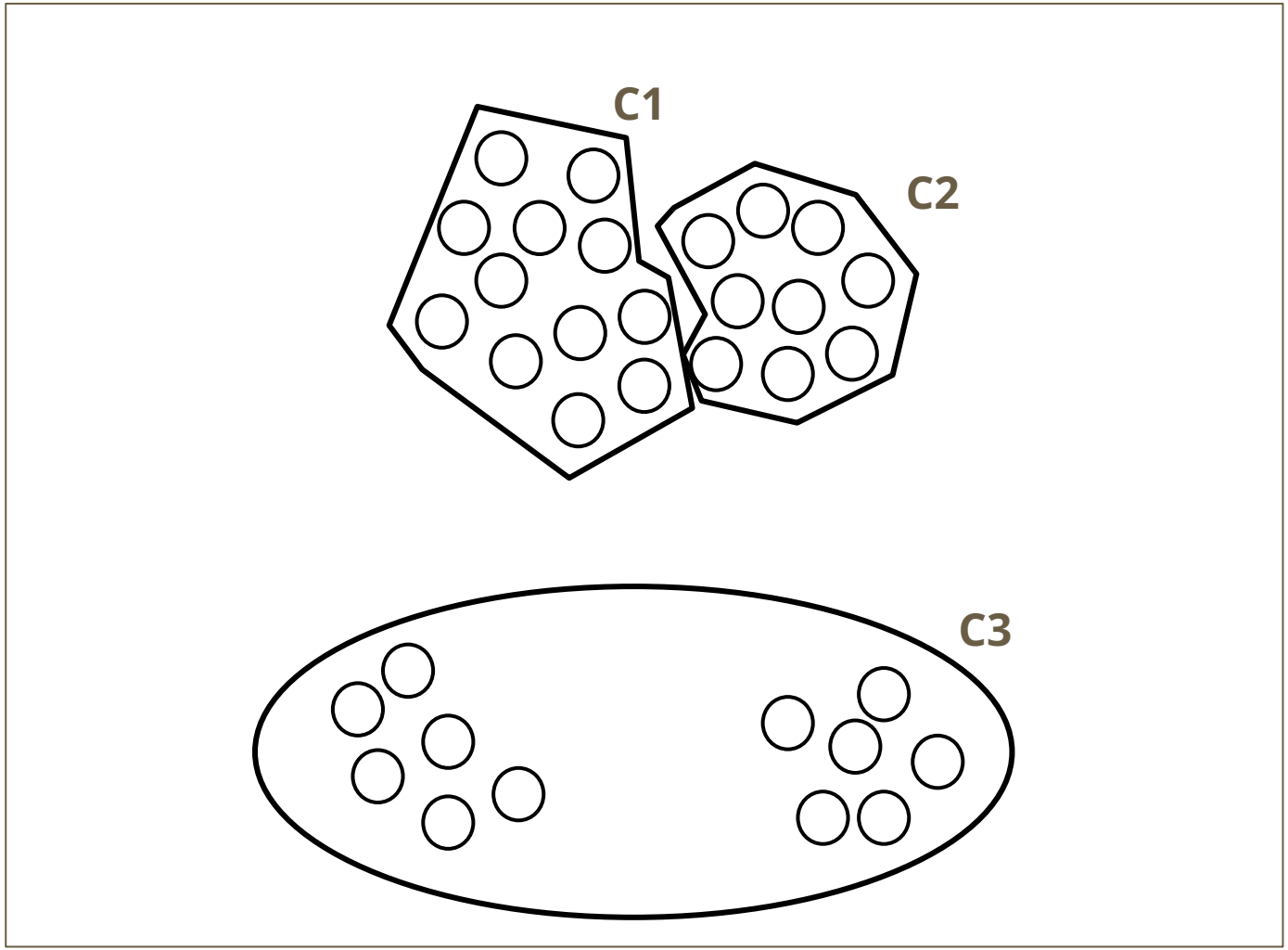


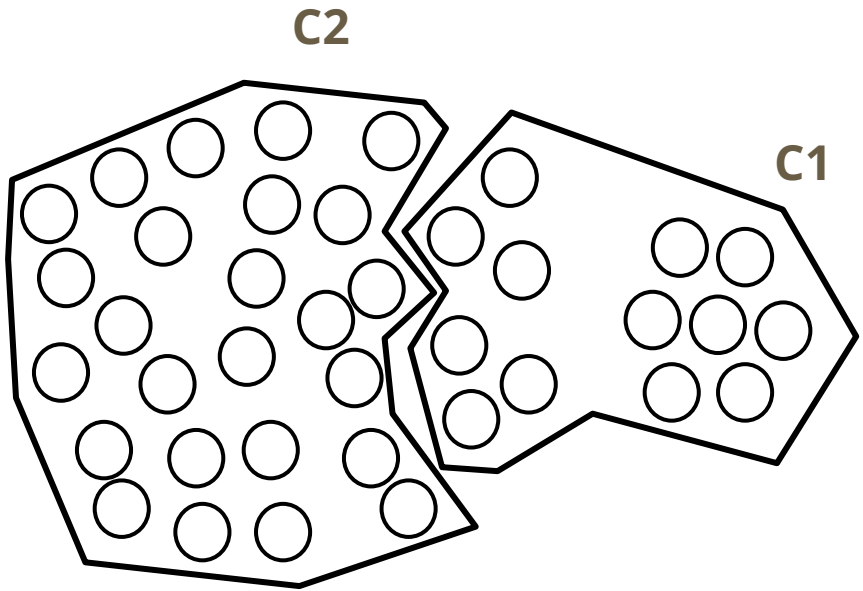
Questions

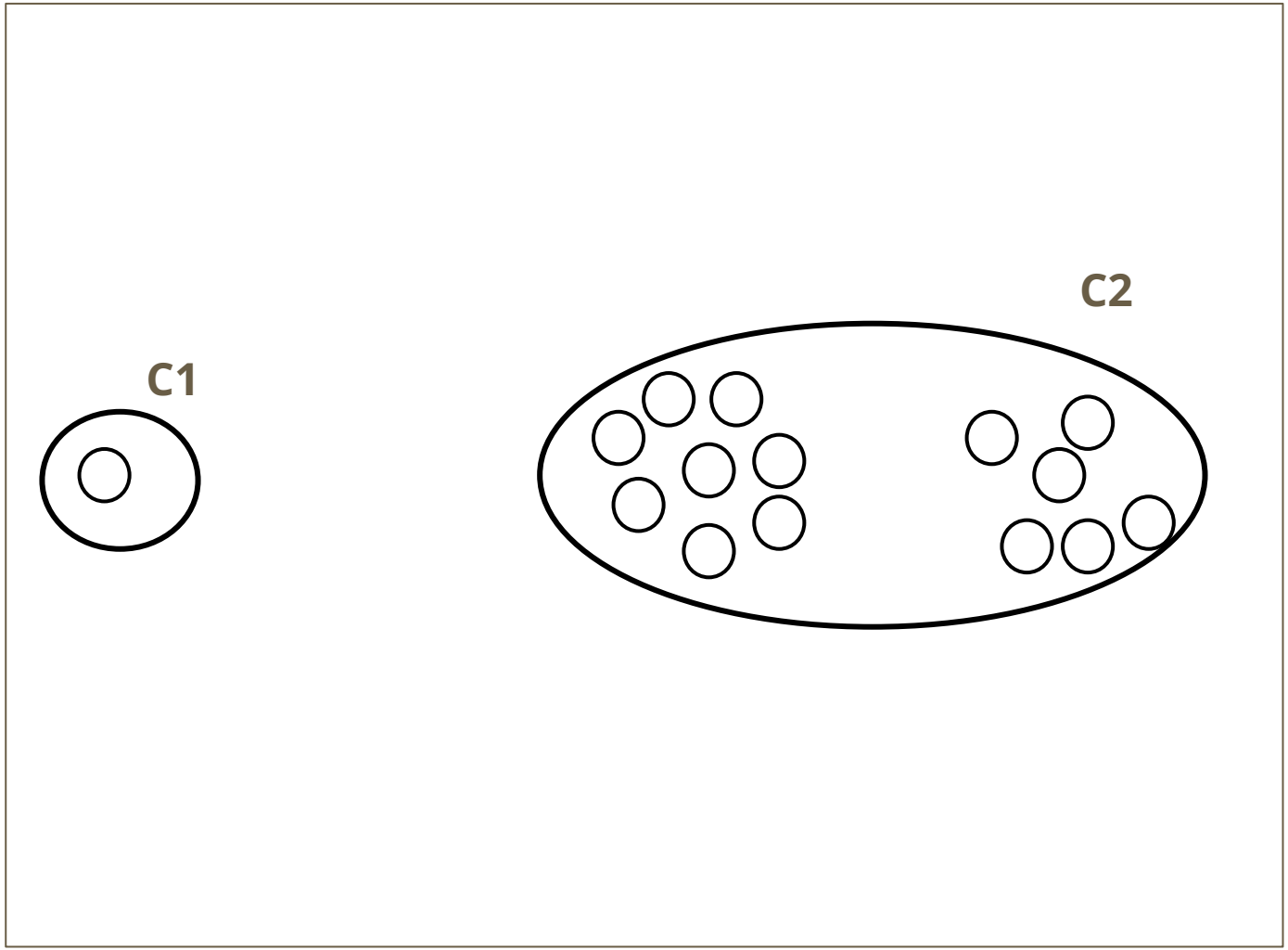

C1
C2
C4
C3


C2
C1


C2
C1


C1
C2
C3


C2
C1


C1
C2