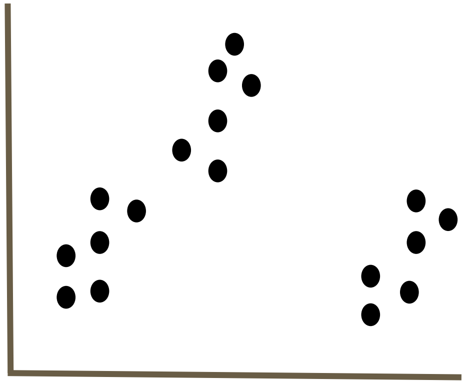
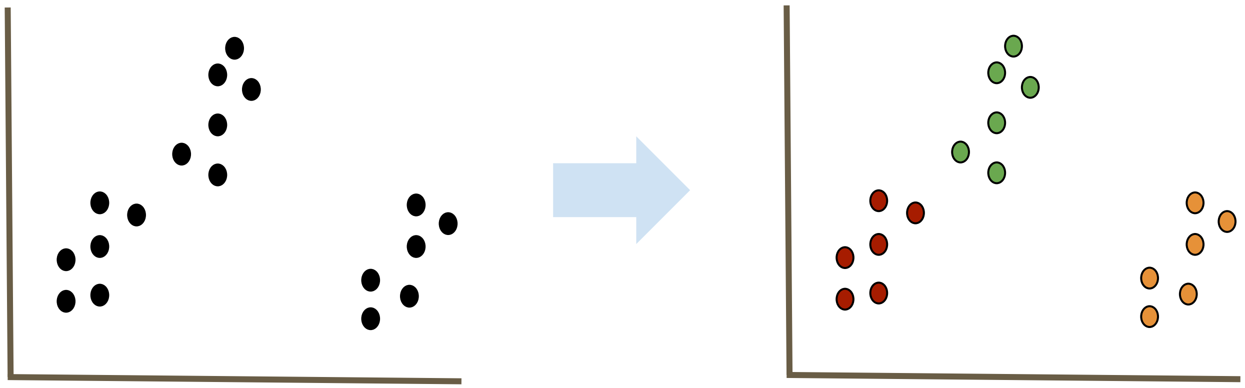
Clustering - Kmeans

Boston University CS 506 - Lance Galletti

What is a Clustering



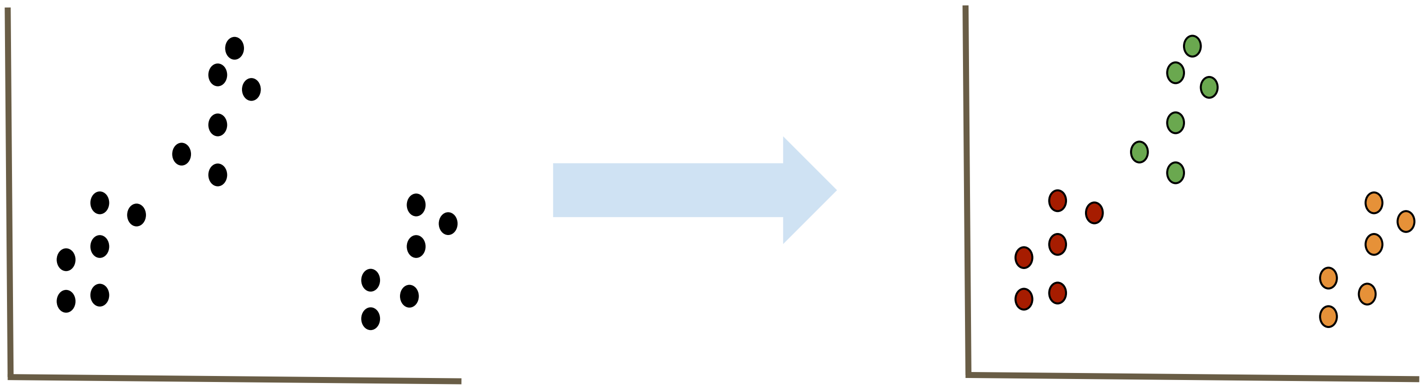
What is a Clustering



What is a Clustering

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

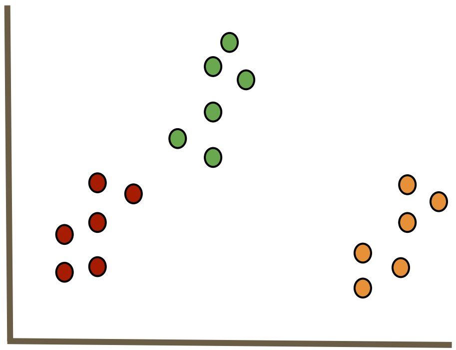
- similar to one another
- dissimilar to objects in other groups

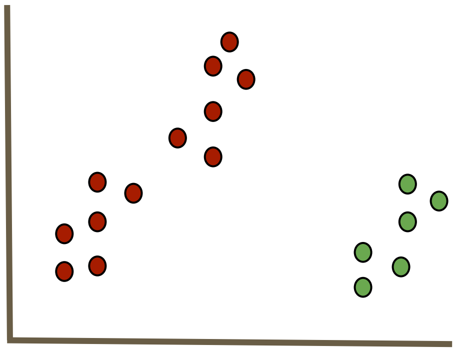


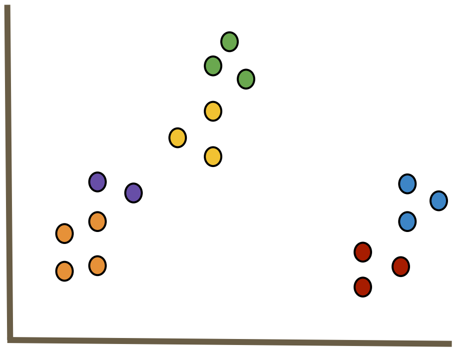
Applications

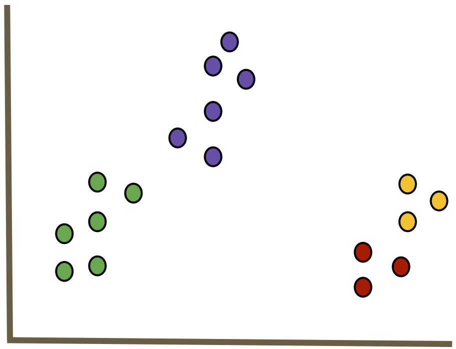
- Outlier detection / anomaly detection
  - Data Cleaning / Processing
  - Credit card fraud, spam filter etc.
- Feature Extraction
- Filling Gaps in your data
  - Using the same marketing strategy for similar people
  - Infer probable values for gaps in the data (similar users could have similar hobbies,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

Clusters can be Ambiguous









Types of Clusterings

Partitional

Each object belongs to exactly one cluster (k partitions)

Hierarchical (high level hierarchy of data points)

A set of nested clusters organized in a tree

Density-Based (create data clusters)

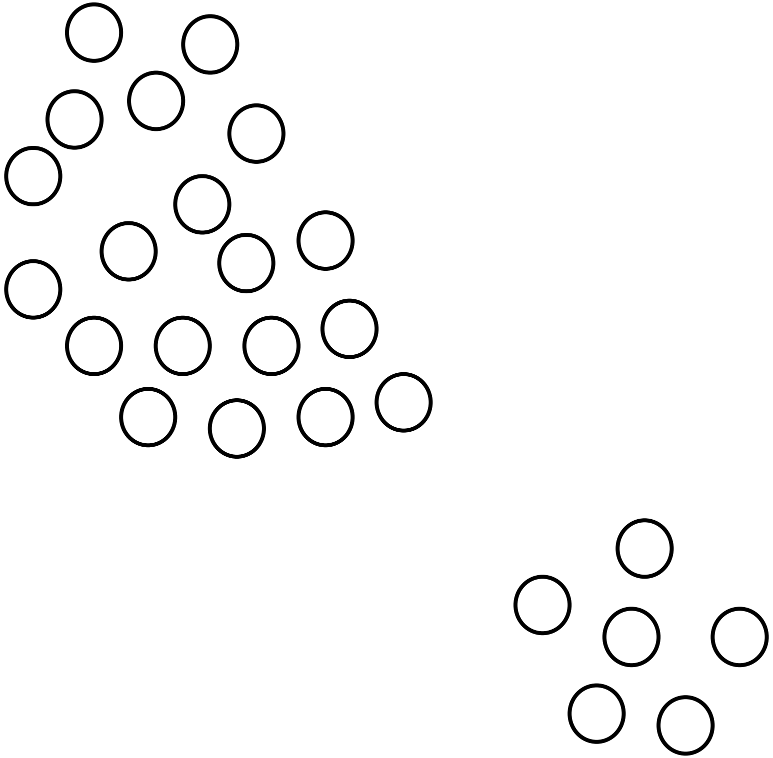
Defined based on the local density of points

Soft Clustering (“probabilistic” clustering)

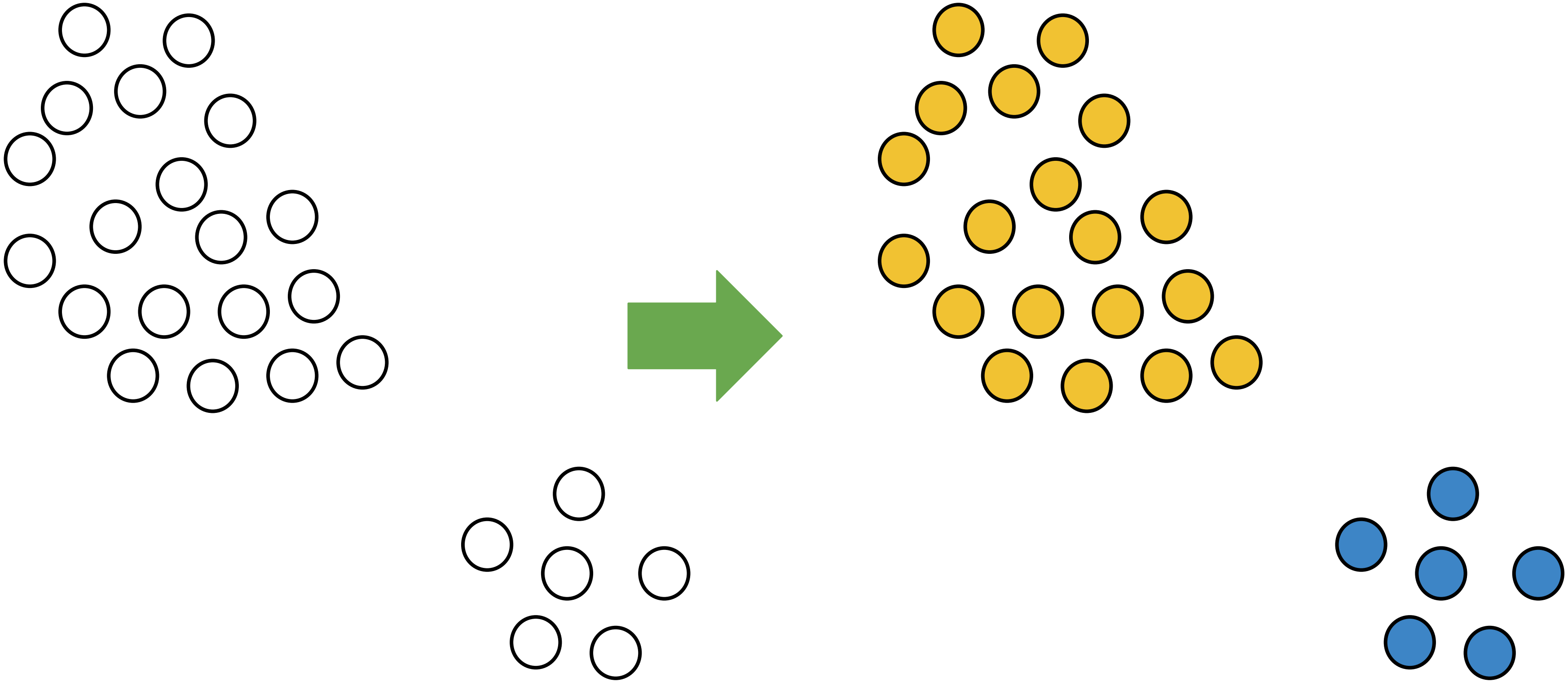
Each point is assigned to every cluster with a certain probability

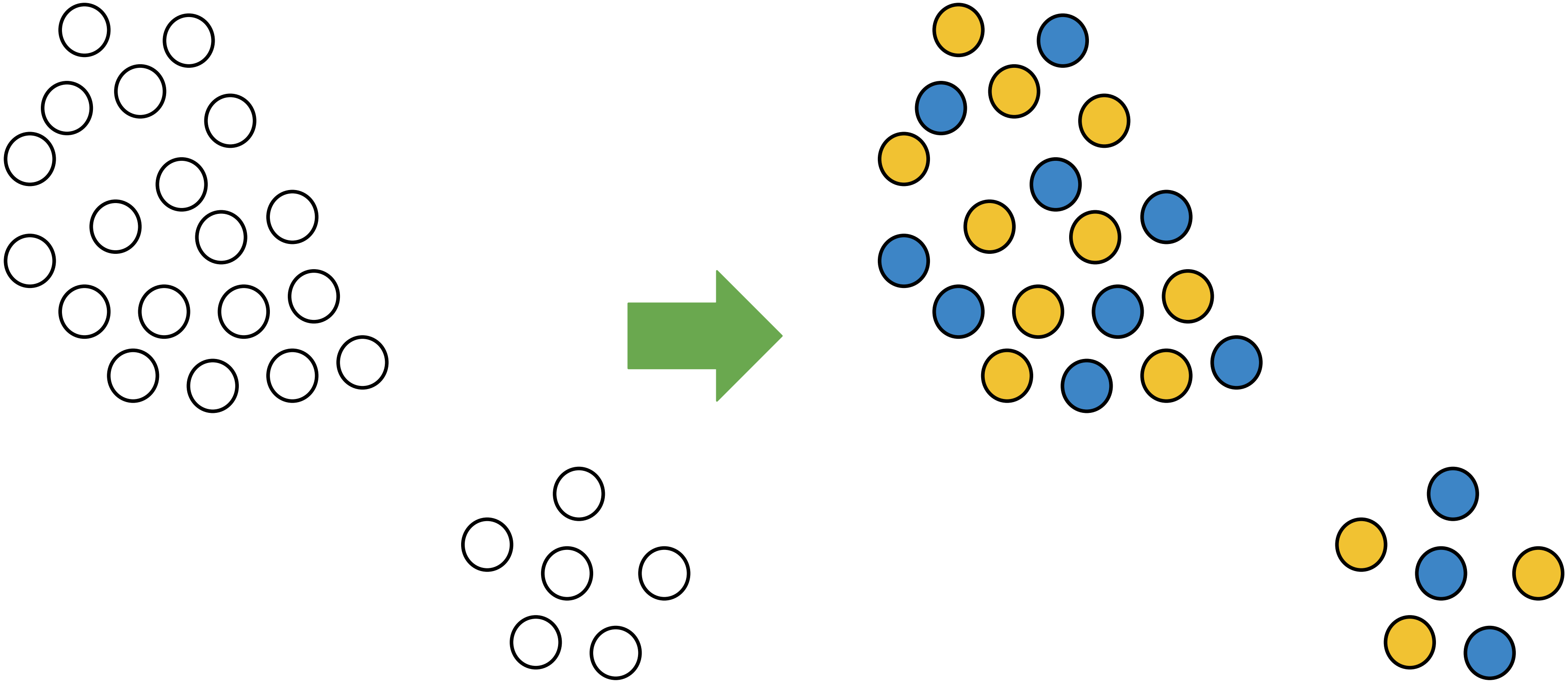
Partitional Clustering

Partitional Clustering

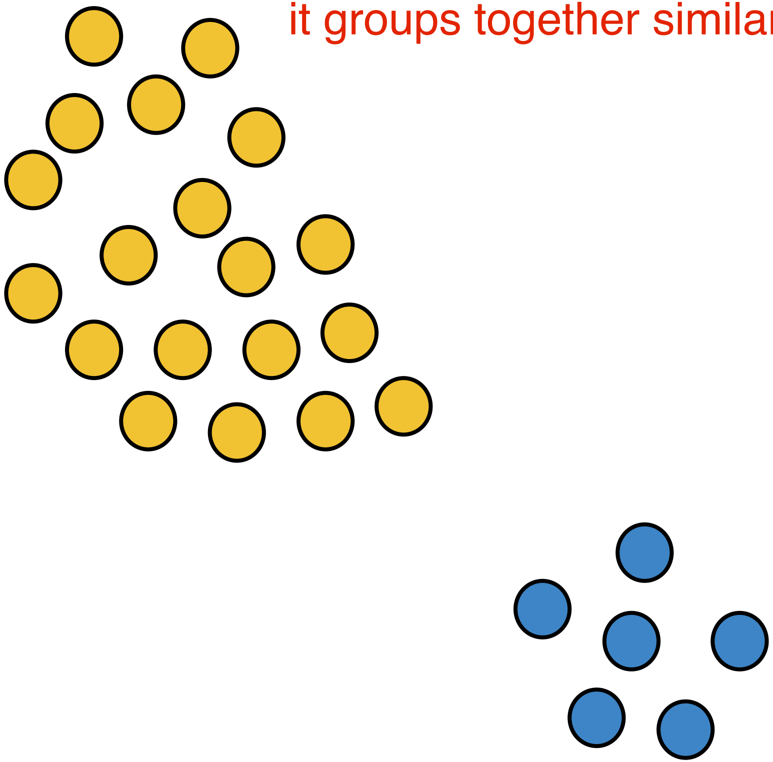


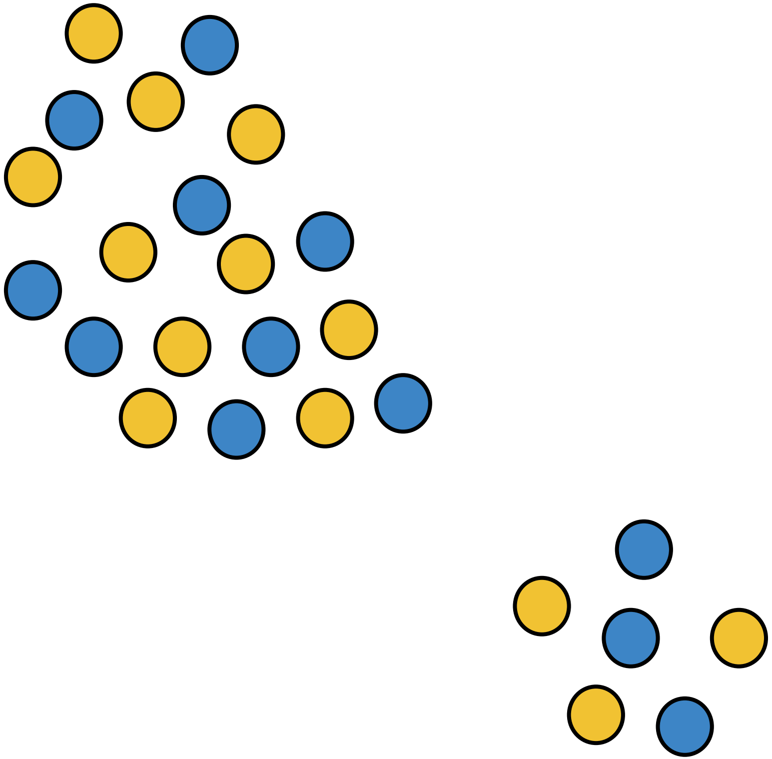
Goal: partition dataset into k partitions

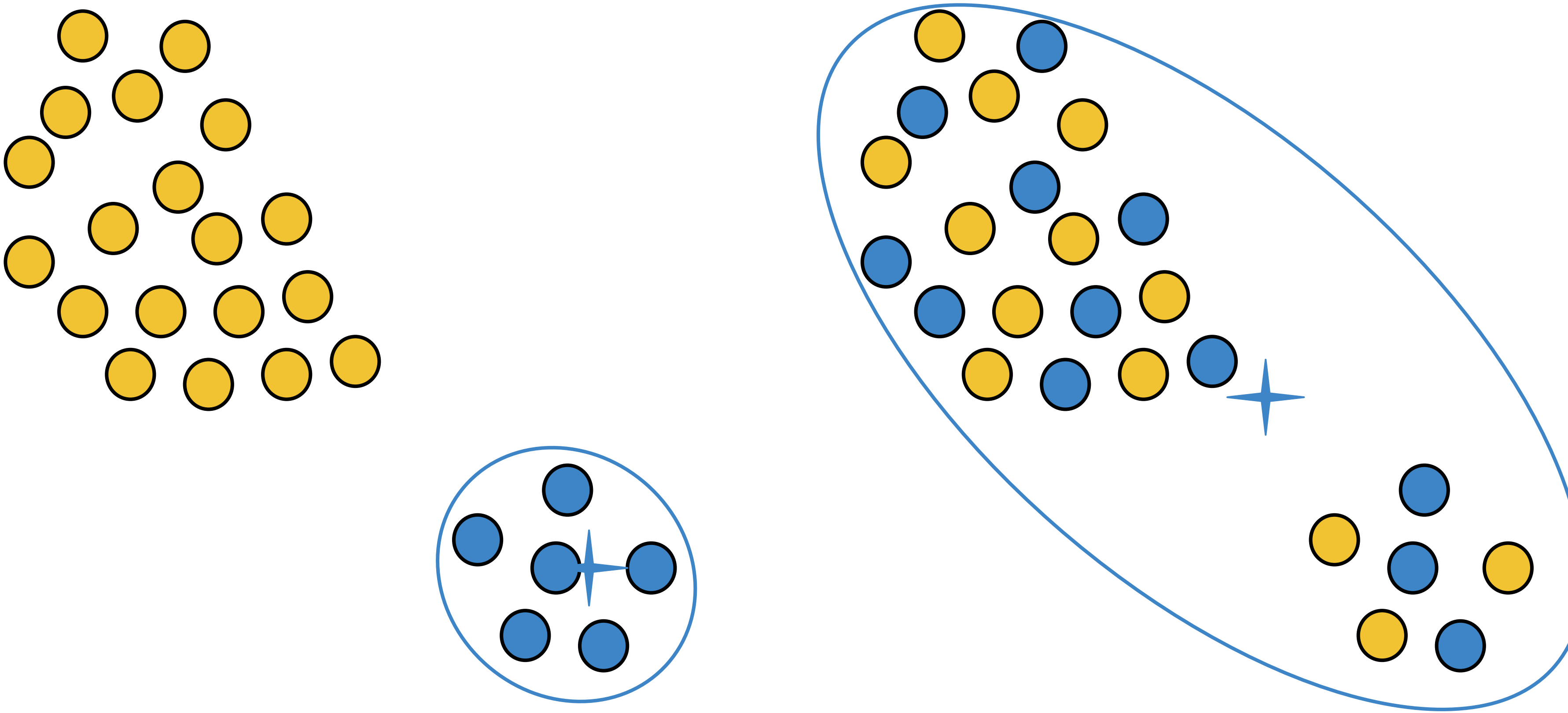




both of these are partitions,
however the first one is better because
it groups together similar data

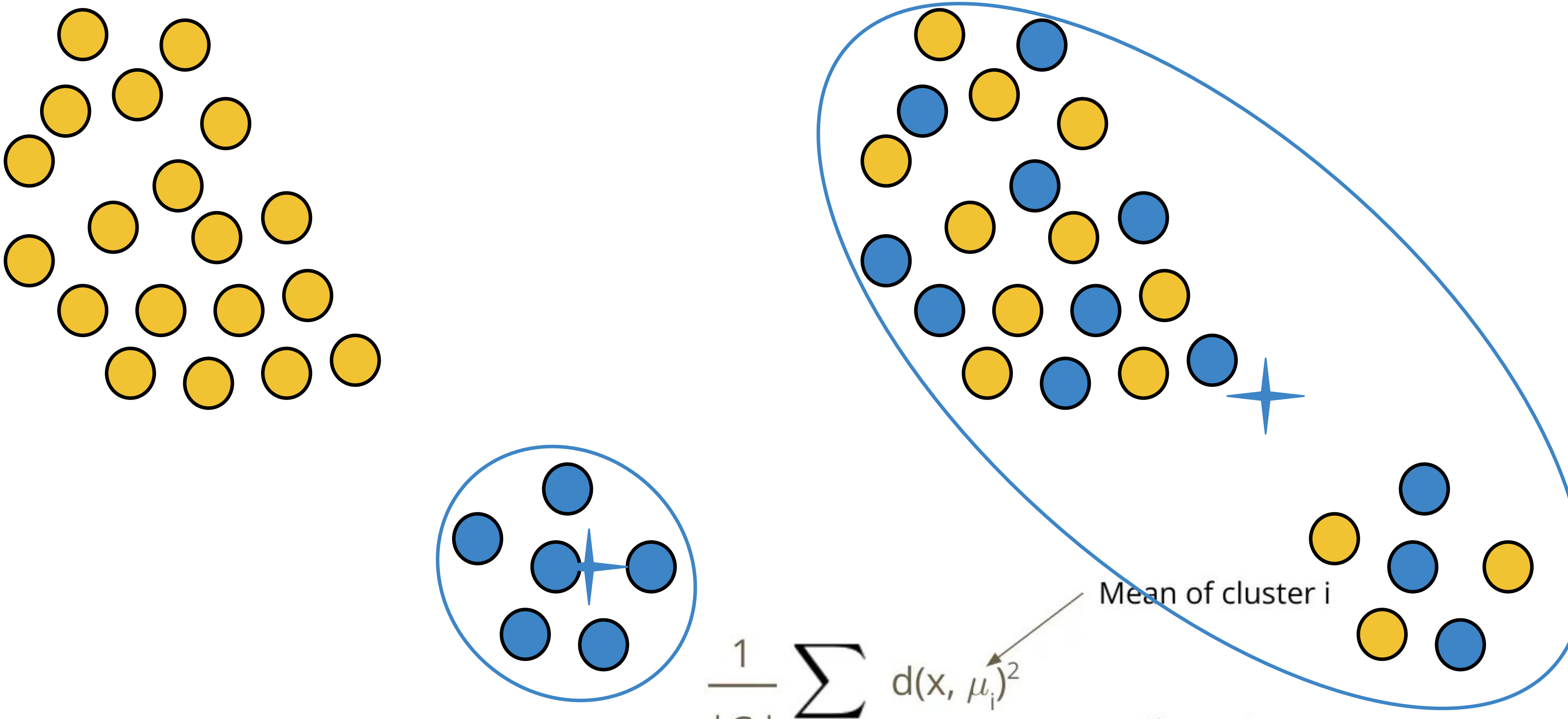






smaller value from cost function

larger value from cost function



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

Cost Function

for every cluster, take the total sum of the squared distance of each point and mean of cluster i

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

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

K-means

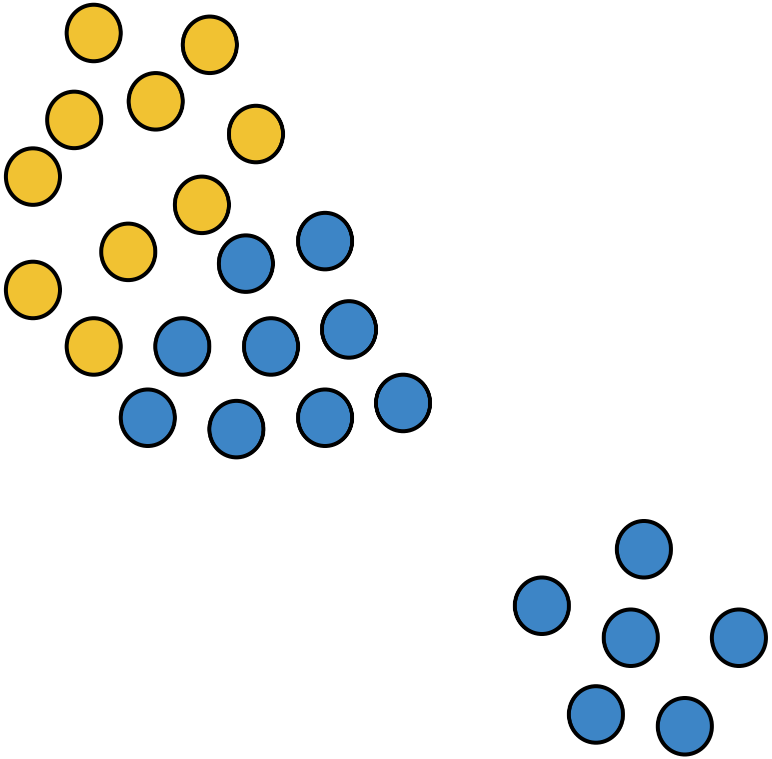
Given $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ our dataset, **d** the euclidean distance, and **k**

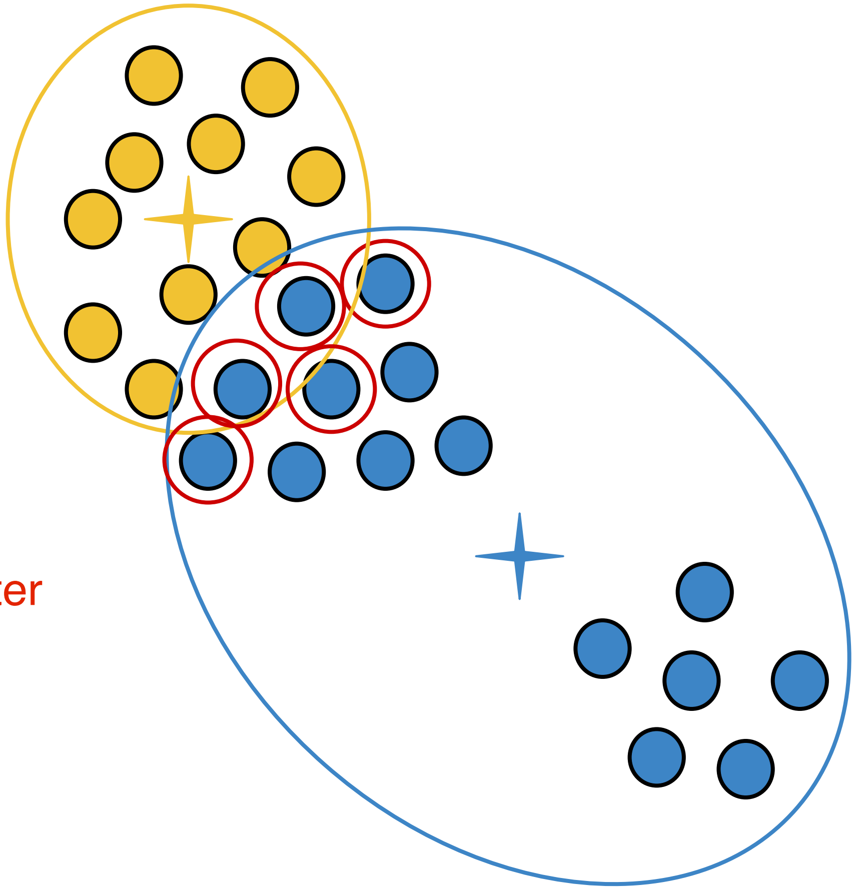
Find **k** centers $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k\}$ that minimize the **cost function**:

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

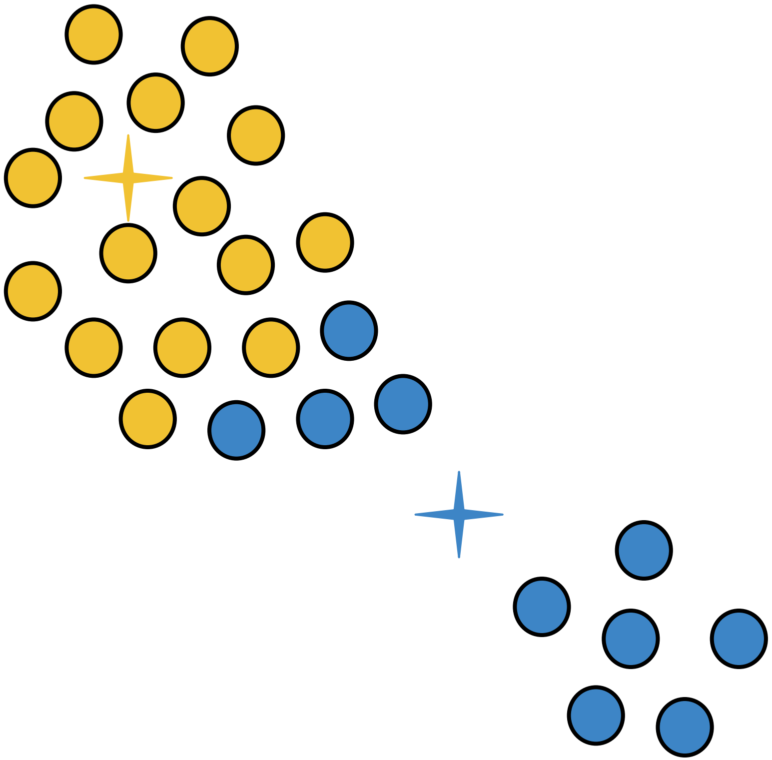
When **k=1** and **k=n** this is easy. Why? The optimal cost func. is 0. Occurs when k=n

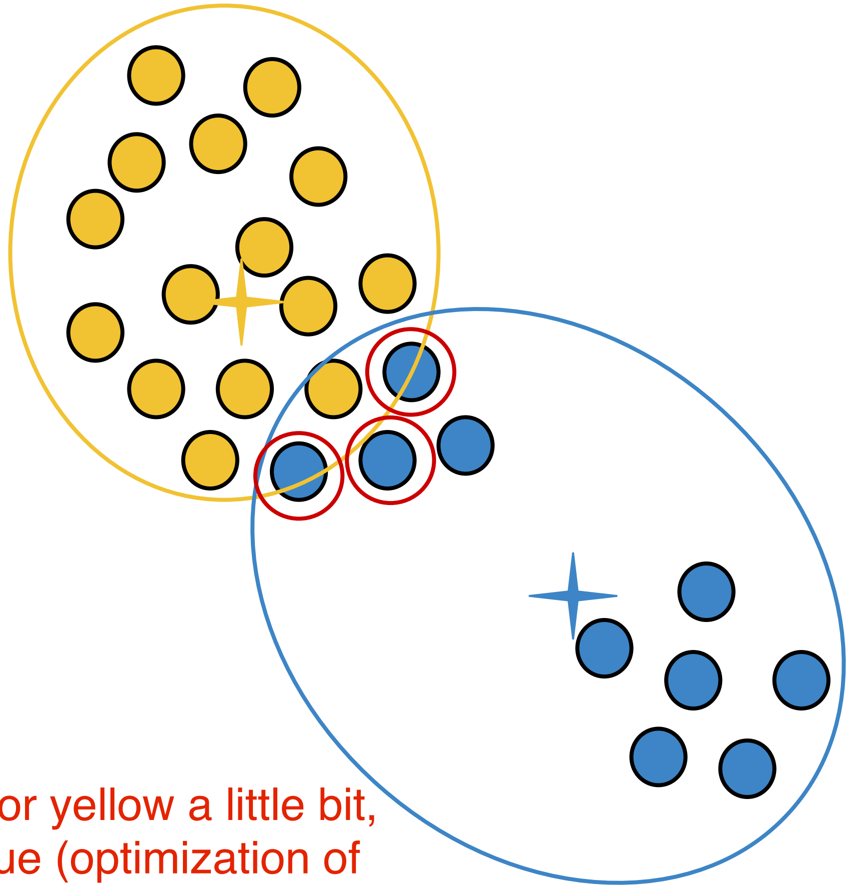
When $\mathbf{x}_i$ lives in more than 2 dimensions, this is a very difficult (**NP-hard**) problem



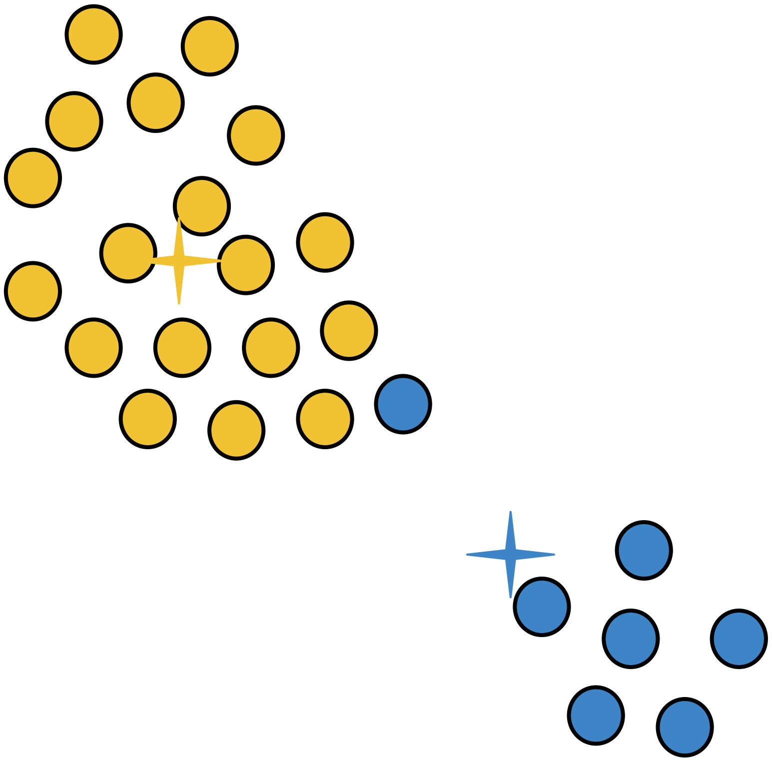


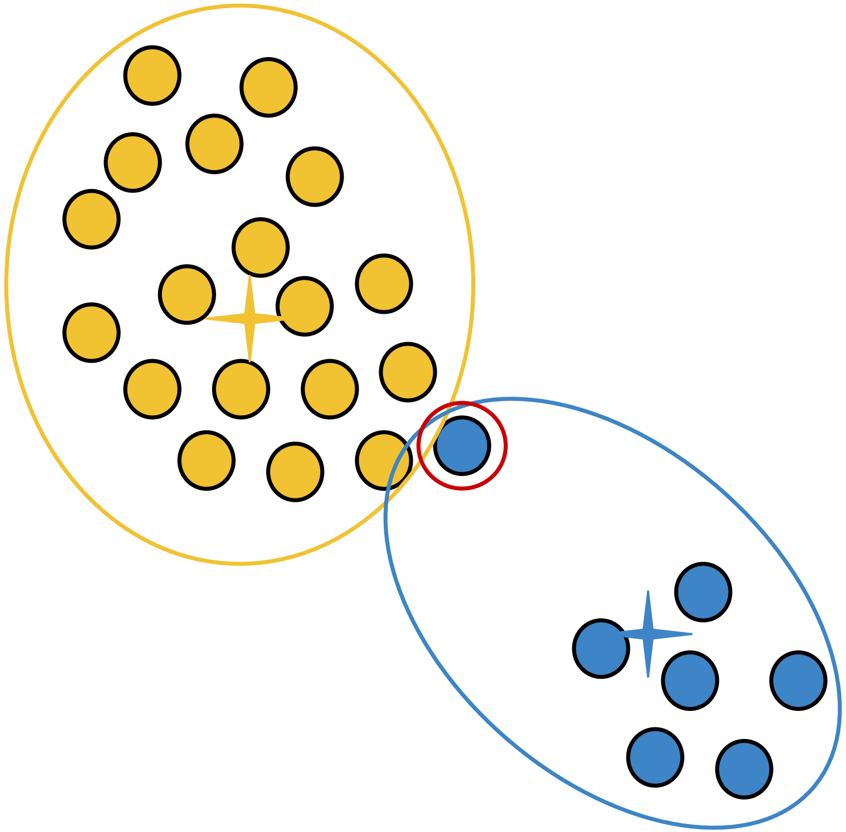
points in red are closer to
yellow mean (the center)
compared to the blue center



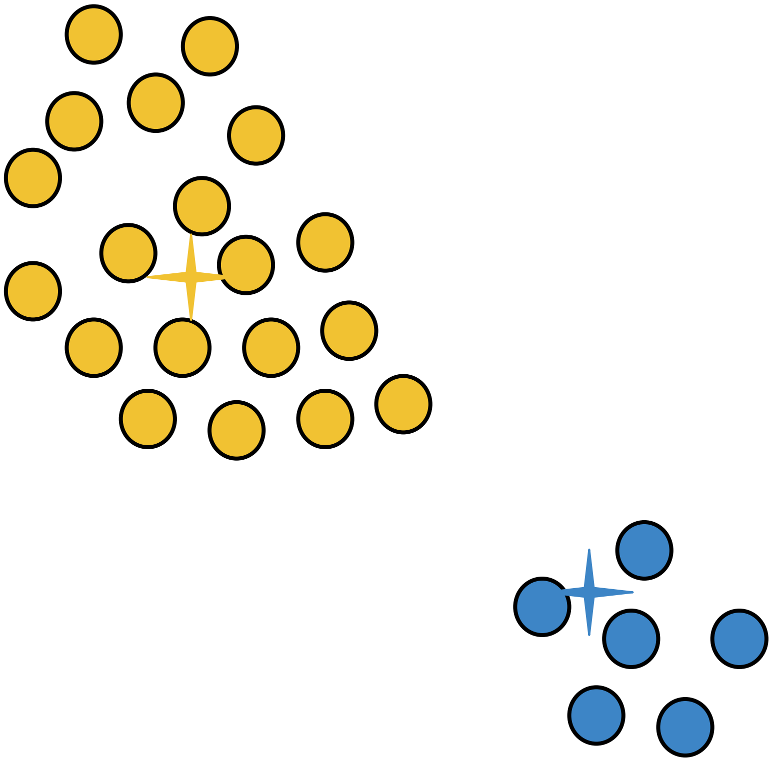


increasing the sum of squares for yellow a little bit, but drastically decreasing for blue (optimization of cost function)





optimized clusters
for cost function

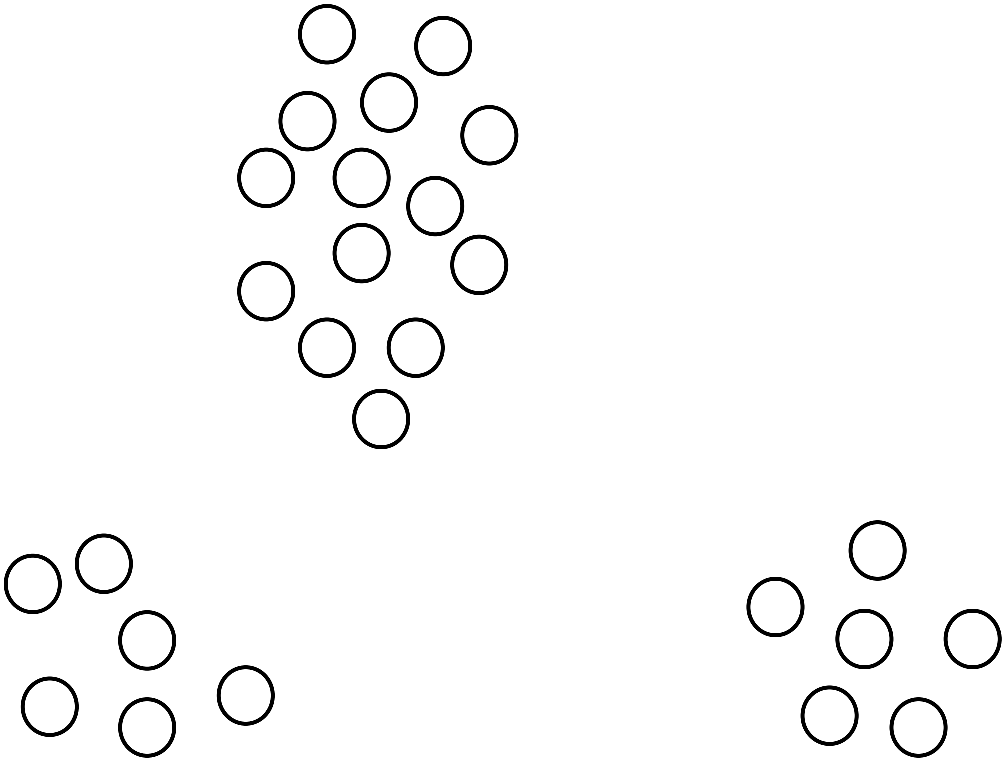


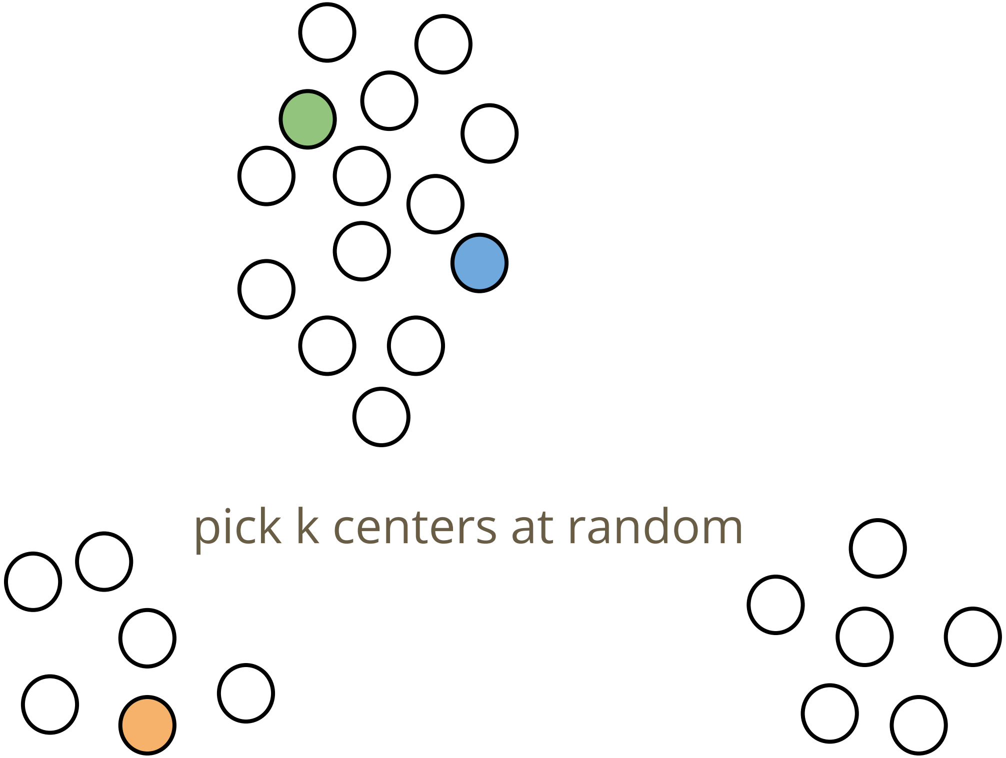
K-means - Lloyd's Algorithm

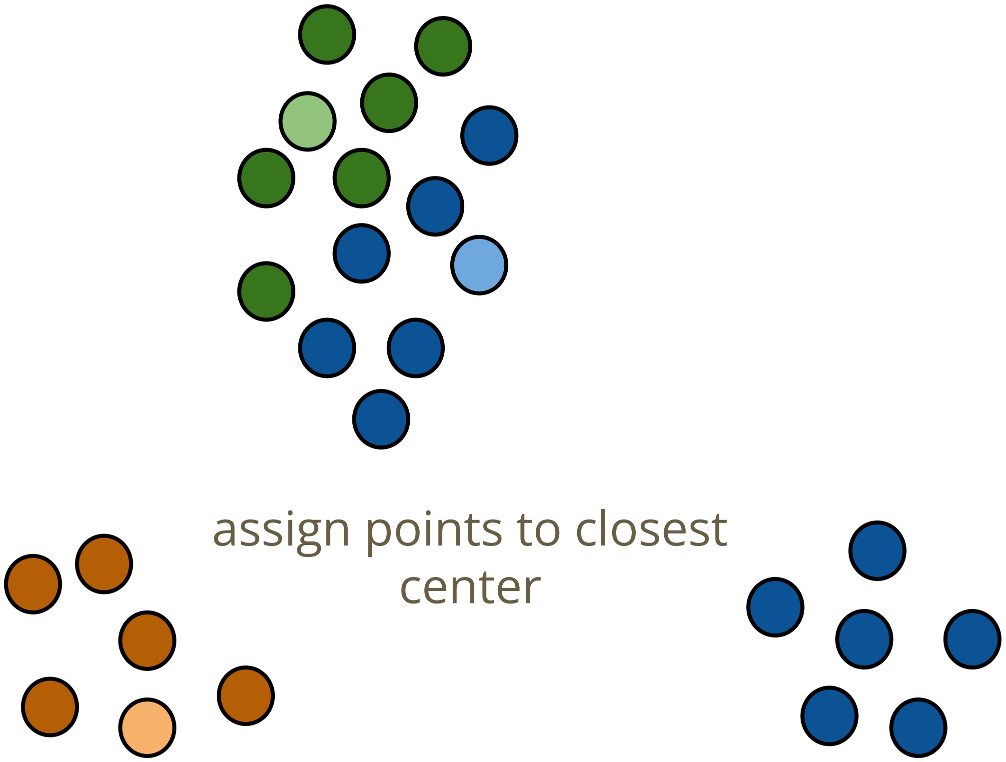
1. Randomly pick **k** centers **{$\mu_1, \ldots, \mu_k$}**
2. Assign each point in the dataset to its closest center
3. Compute the new centers as the means of each cluster
4. Repeat 2 & 3 until convergence

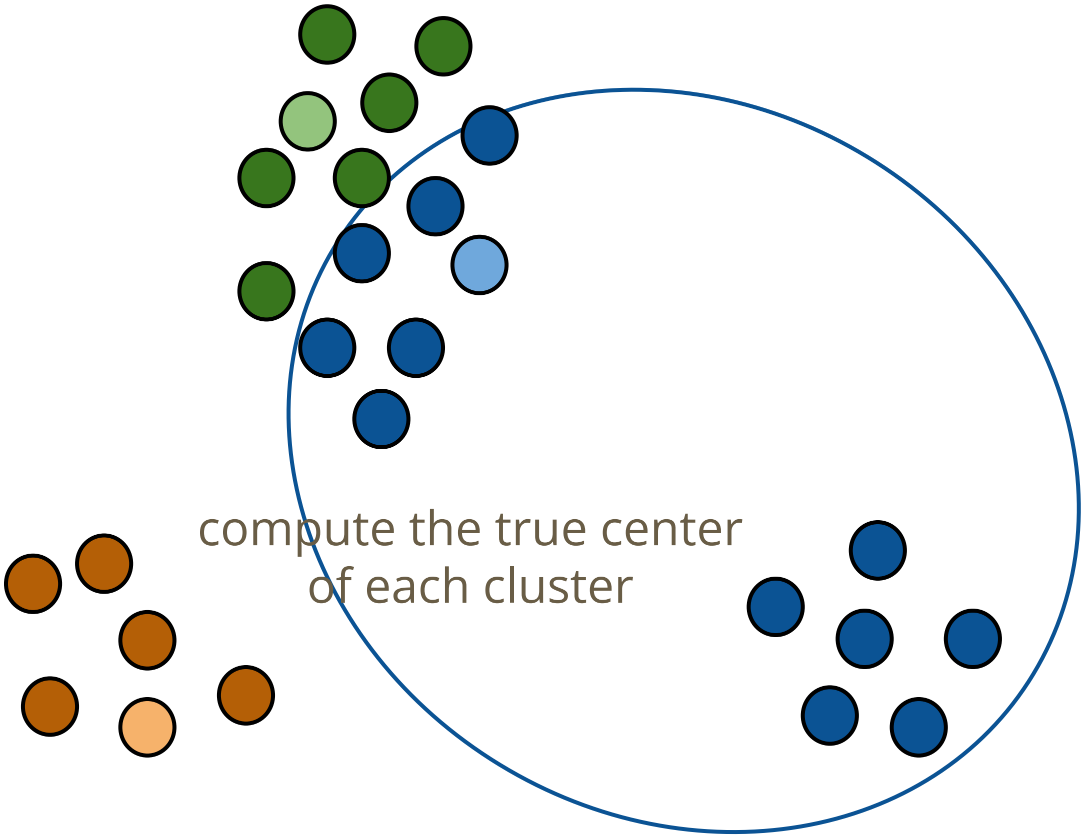
pick points closest to
center to create cluster

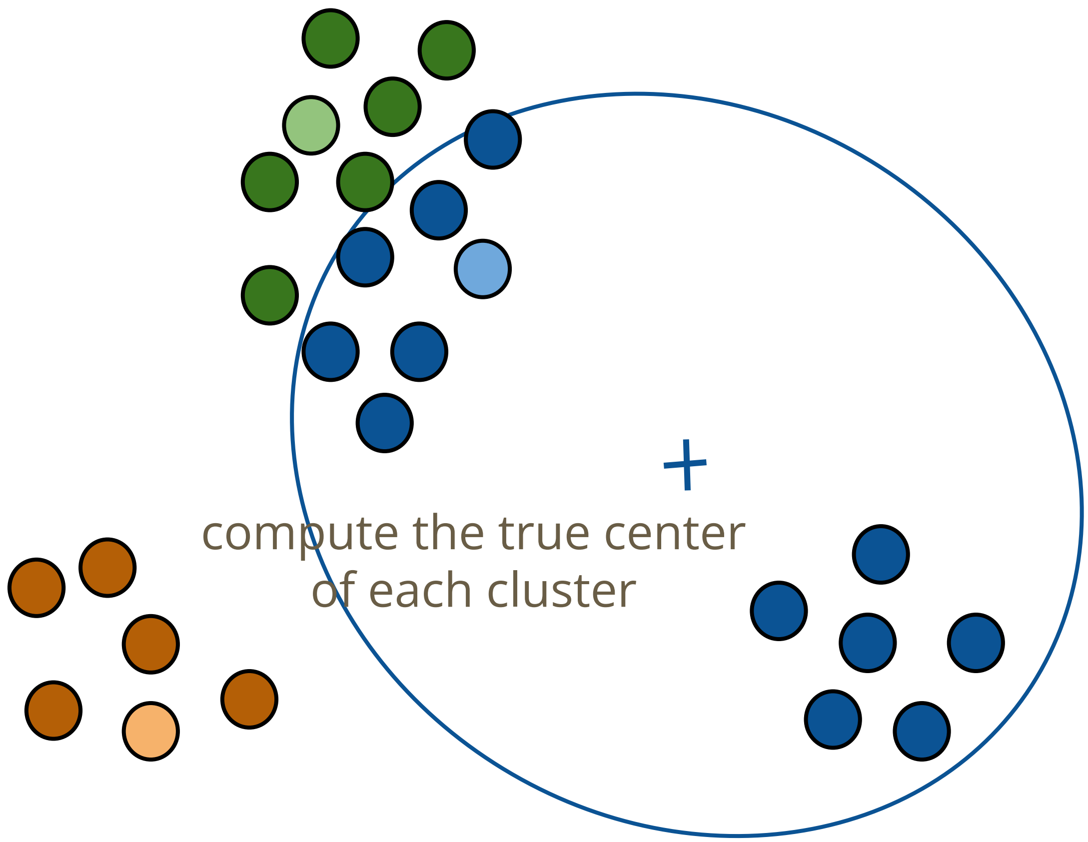
adjust centers by calculating mean
and recalculating for new centers

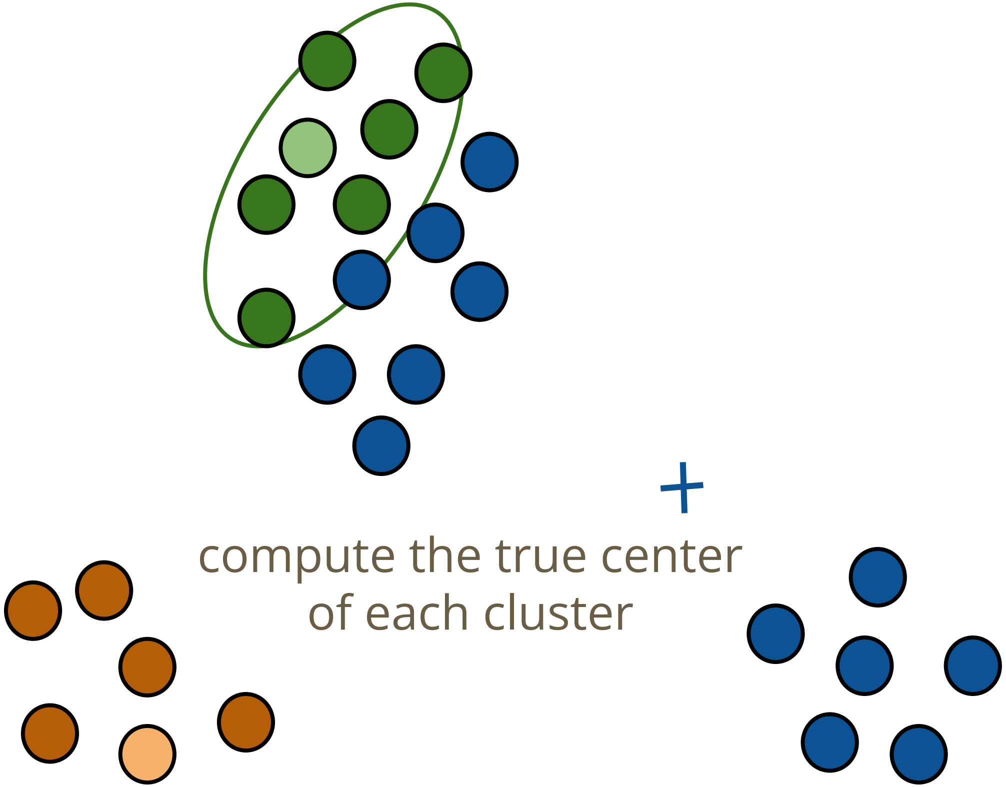


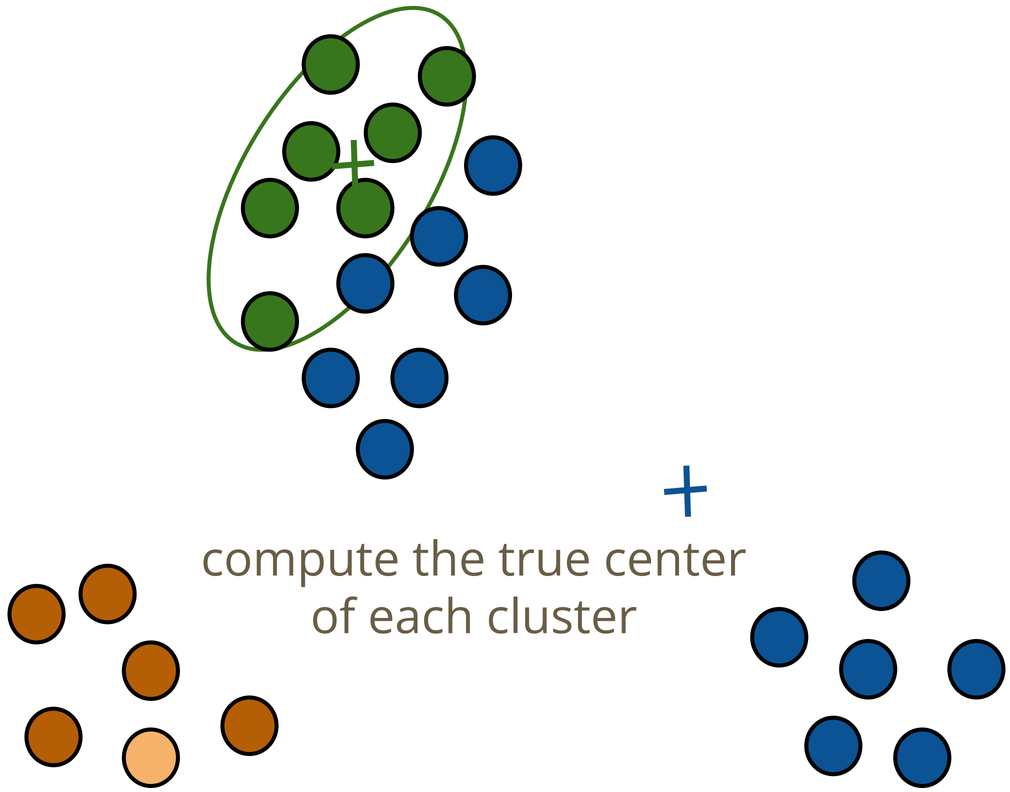

pick k centers at random

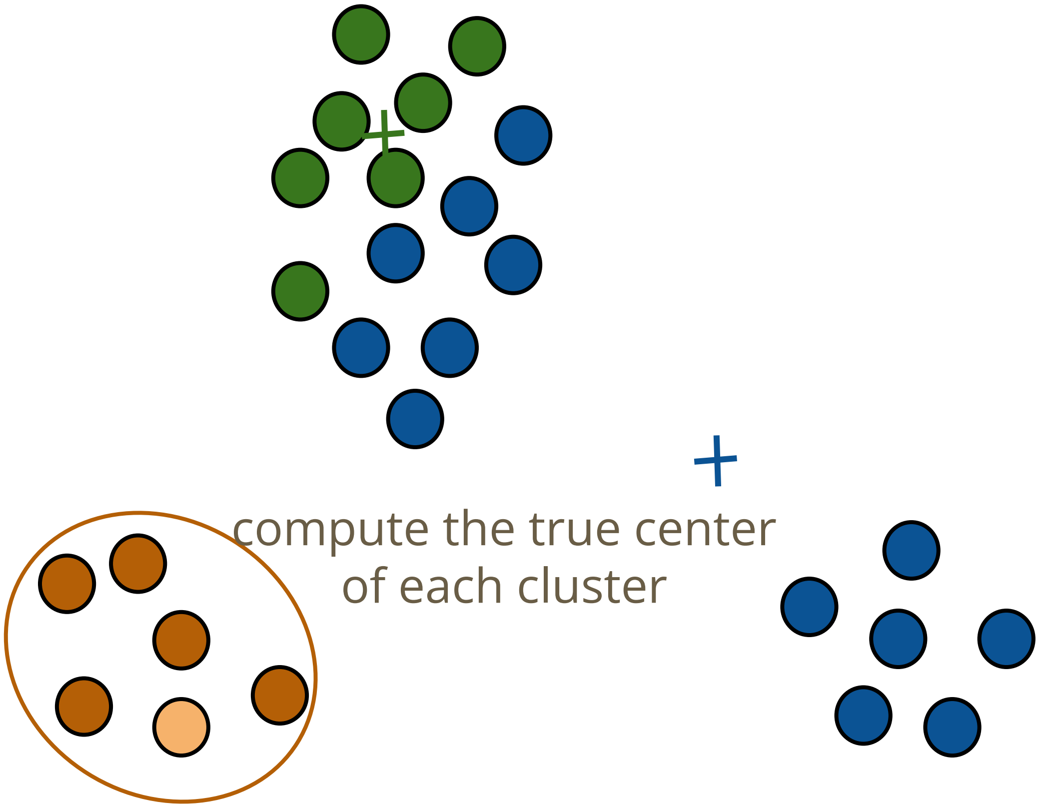

assign points to closest center

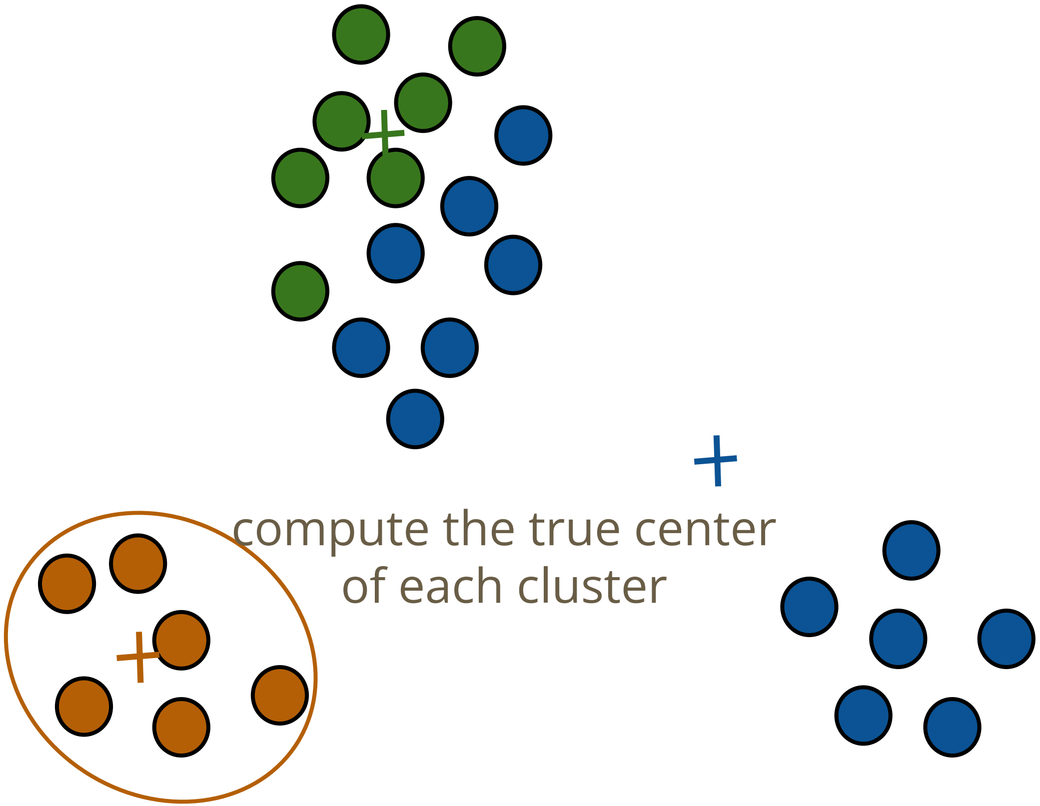

compute the true center
of each cluster

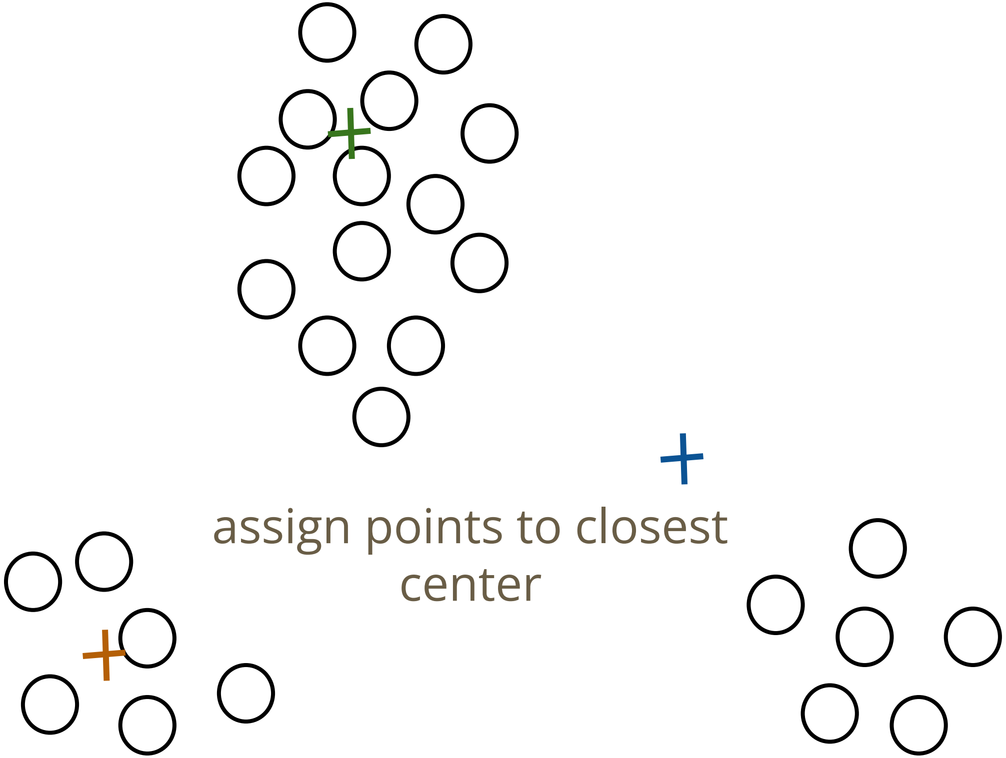

compute the true center
of each cluster

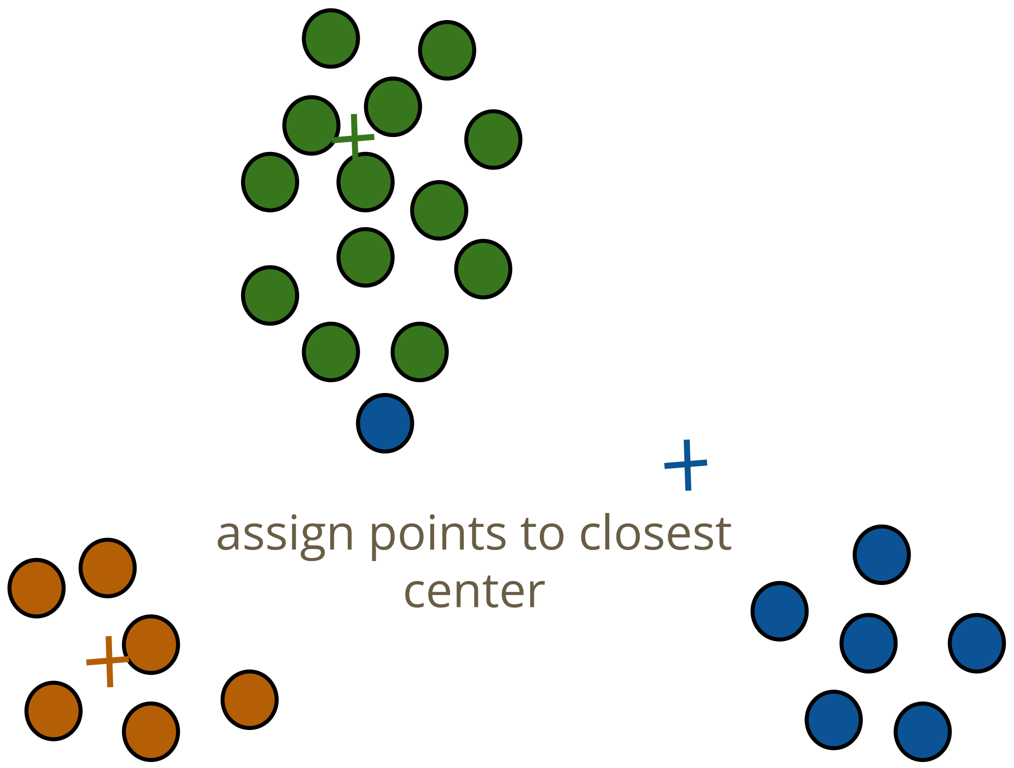

compute the true center
of each cluster

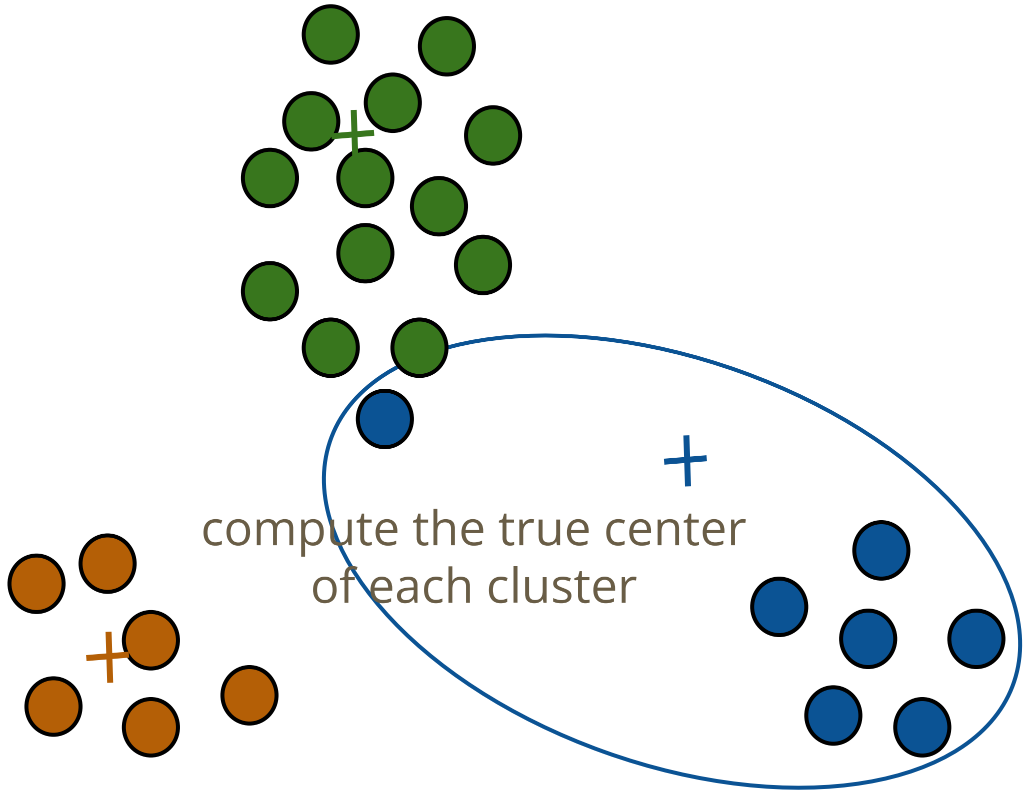

compute the true center
of each cluster

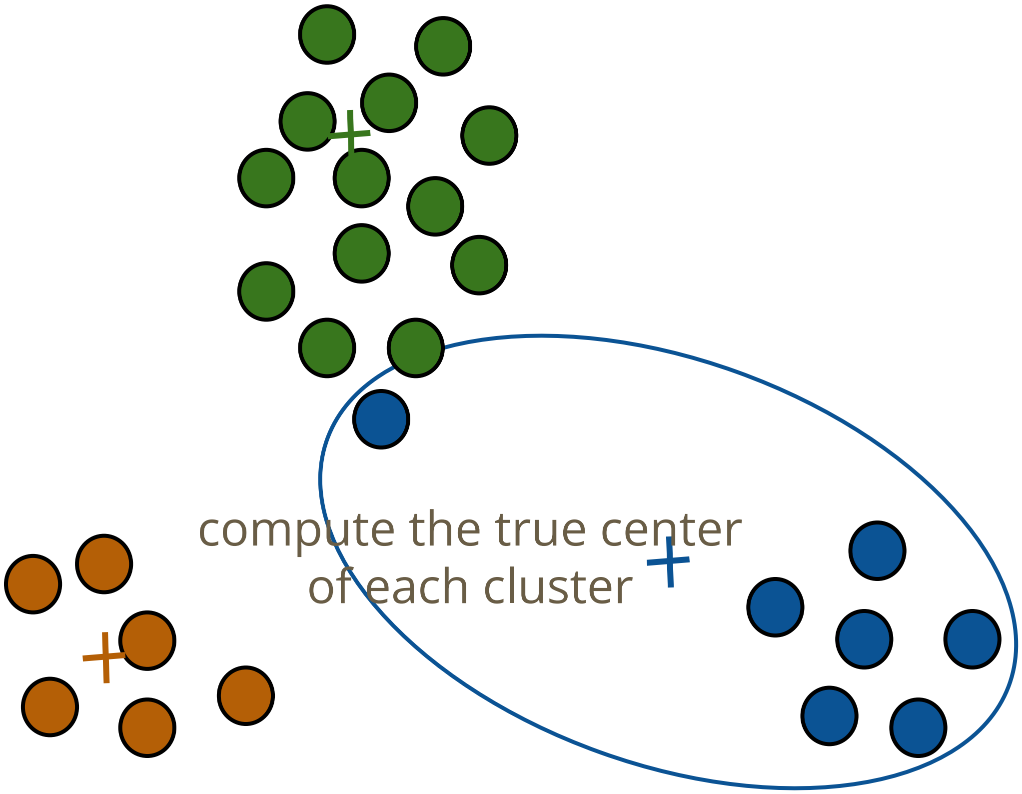

compute the true center
of each cluster

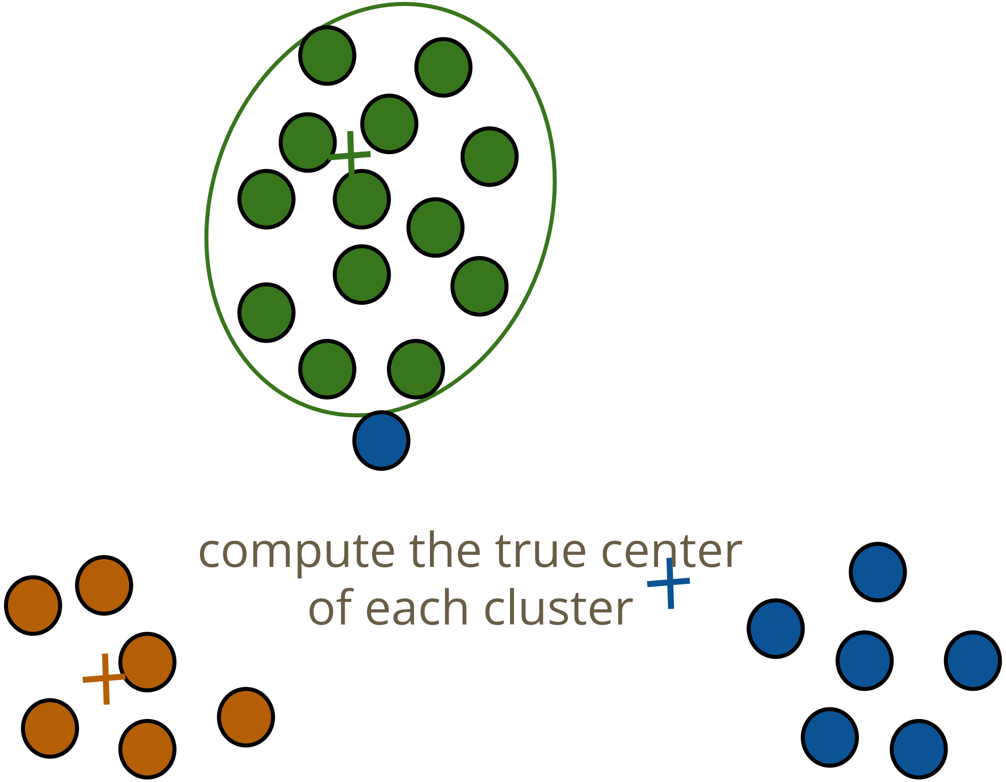

compute the true center
of each cluster

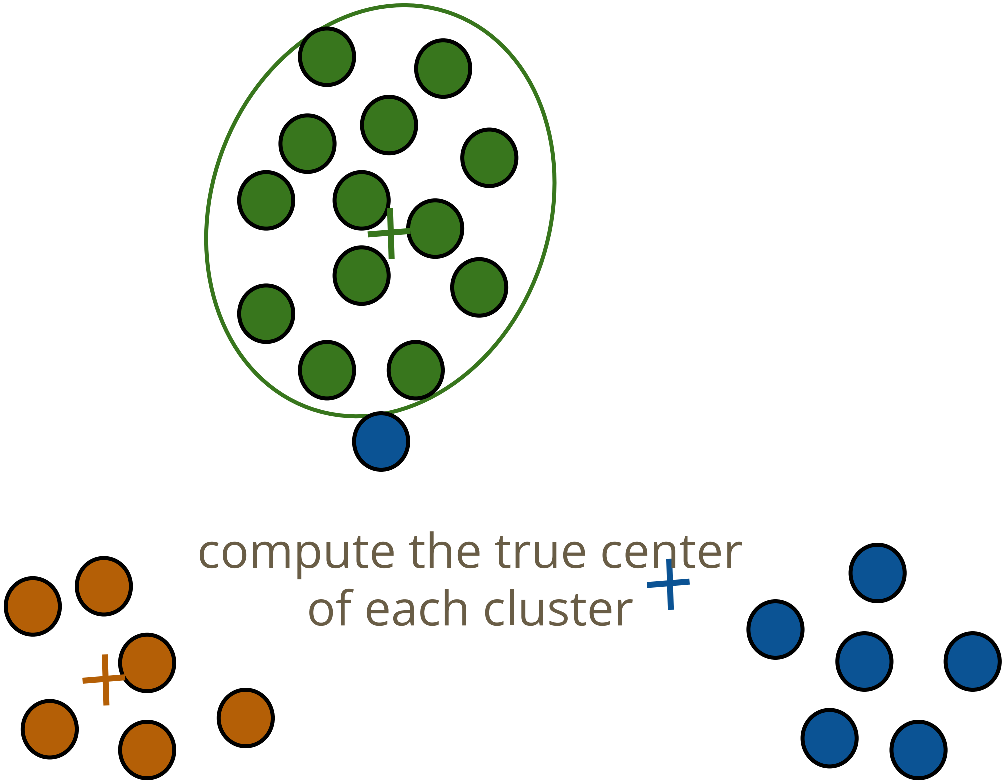

assign points to closest center

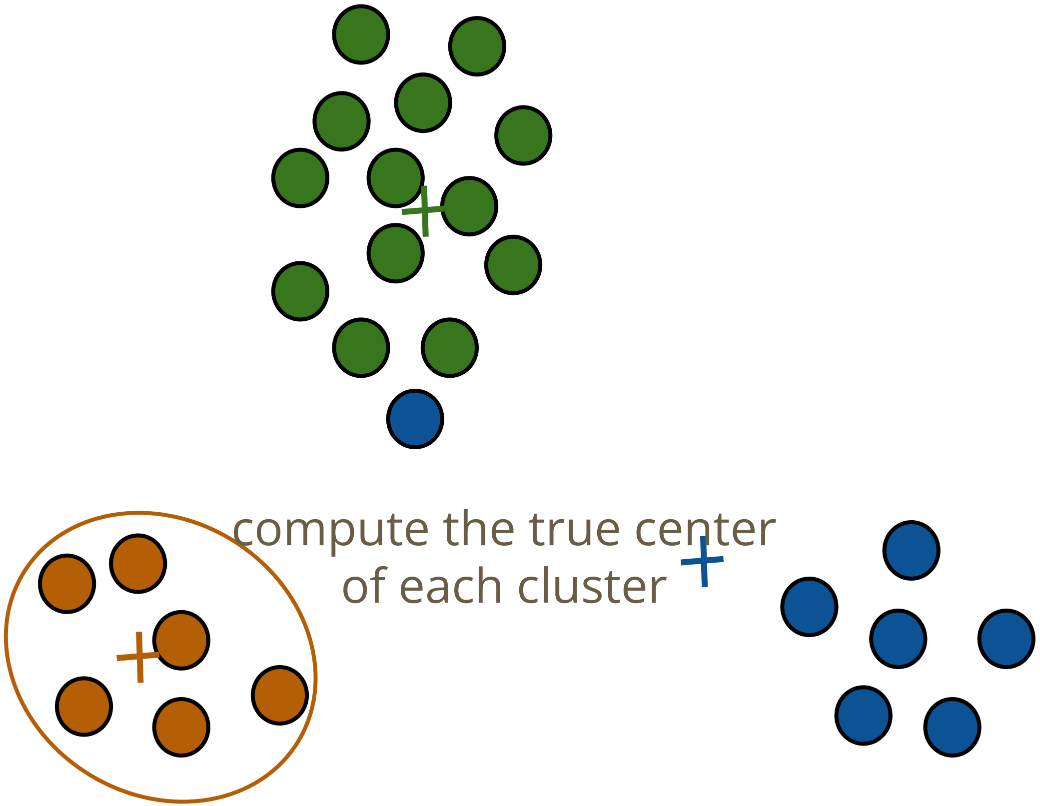

assign points to closest
center

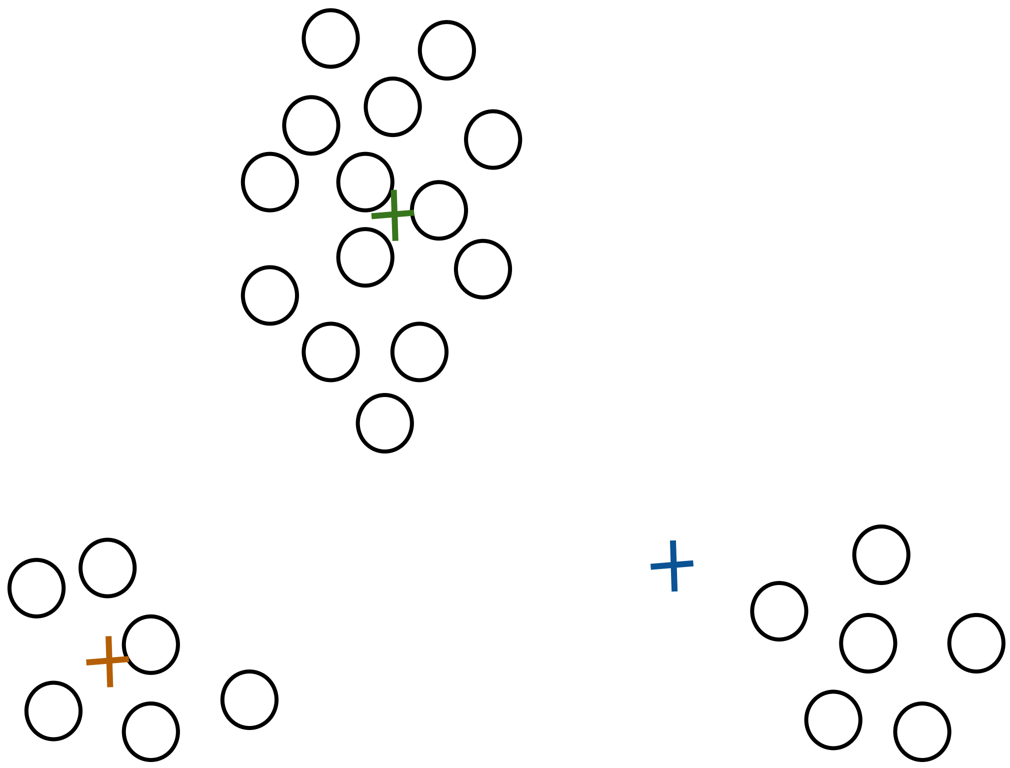

compute the true center
of each cluster

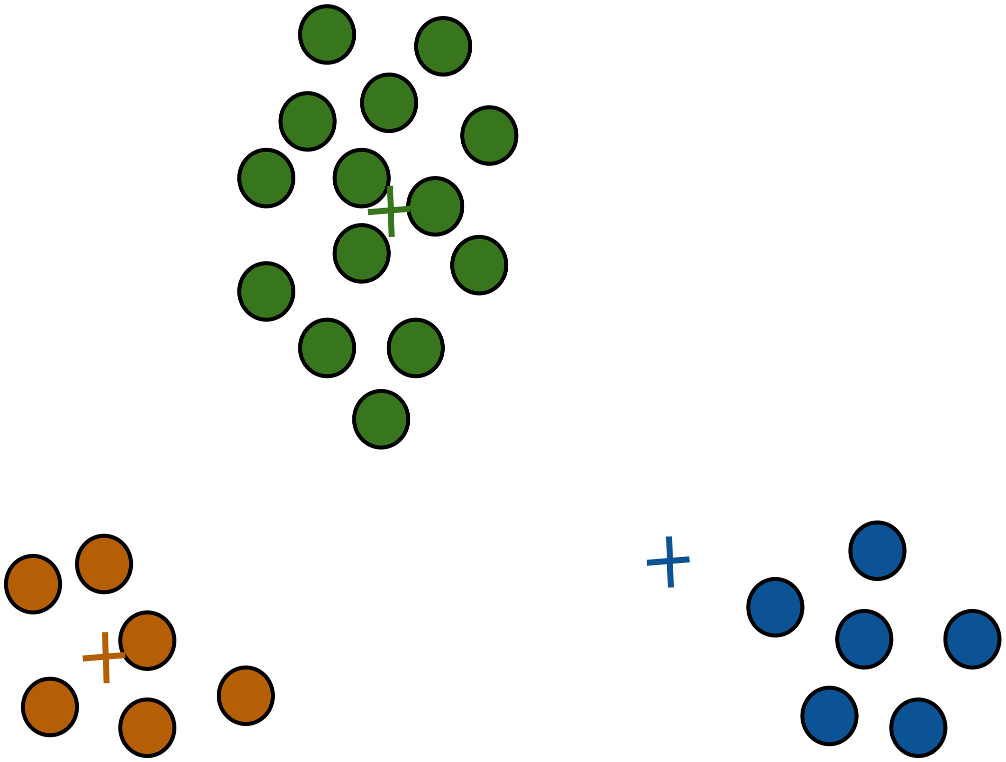

compute the true center
of each cluster

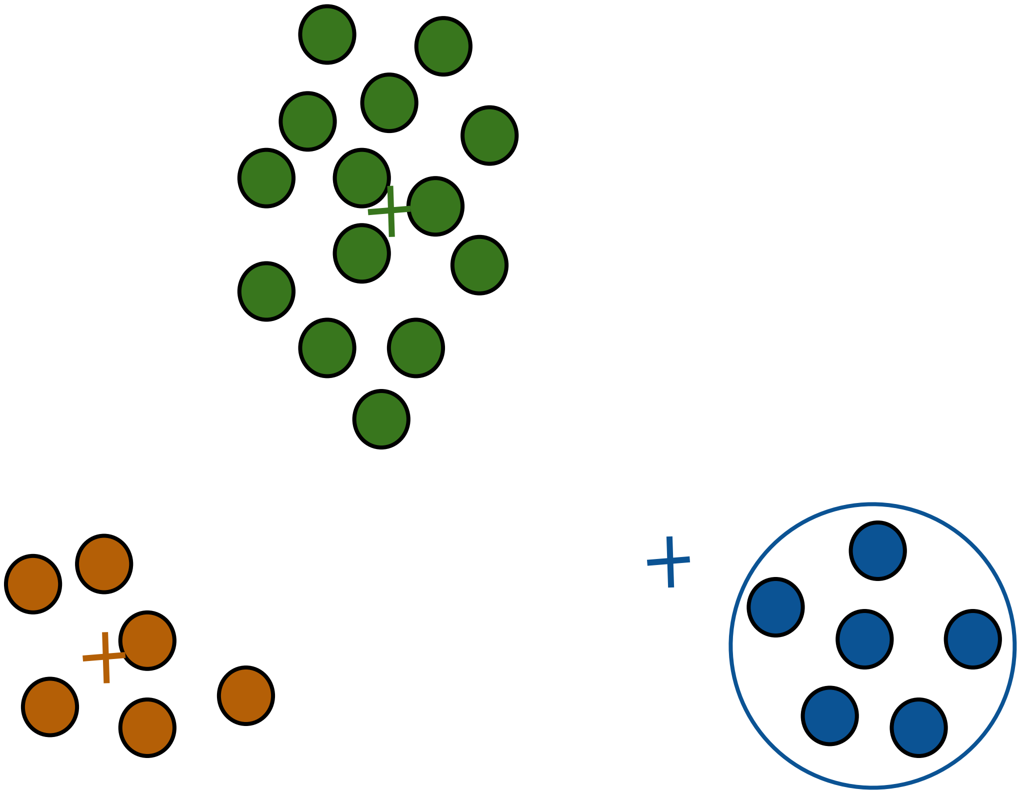

compute the true center
of each cluster

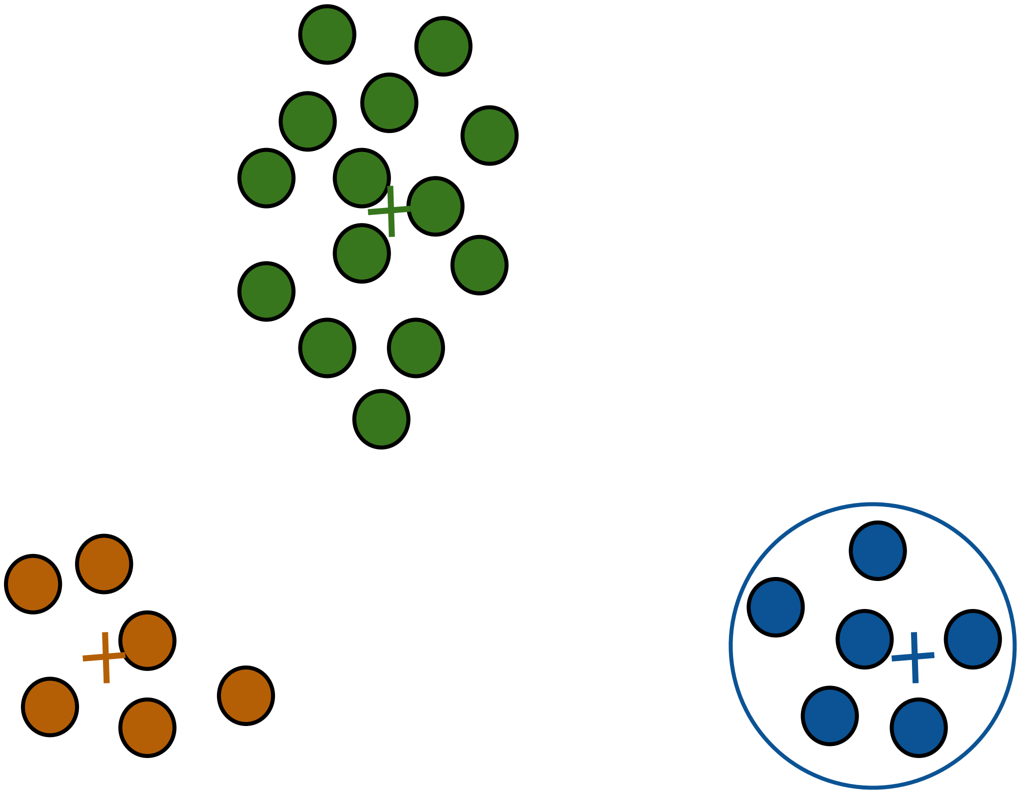

compute the true center
of each cluster

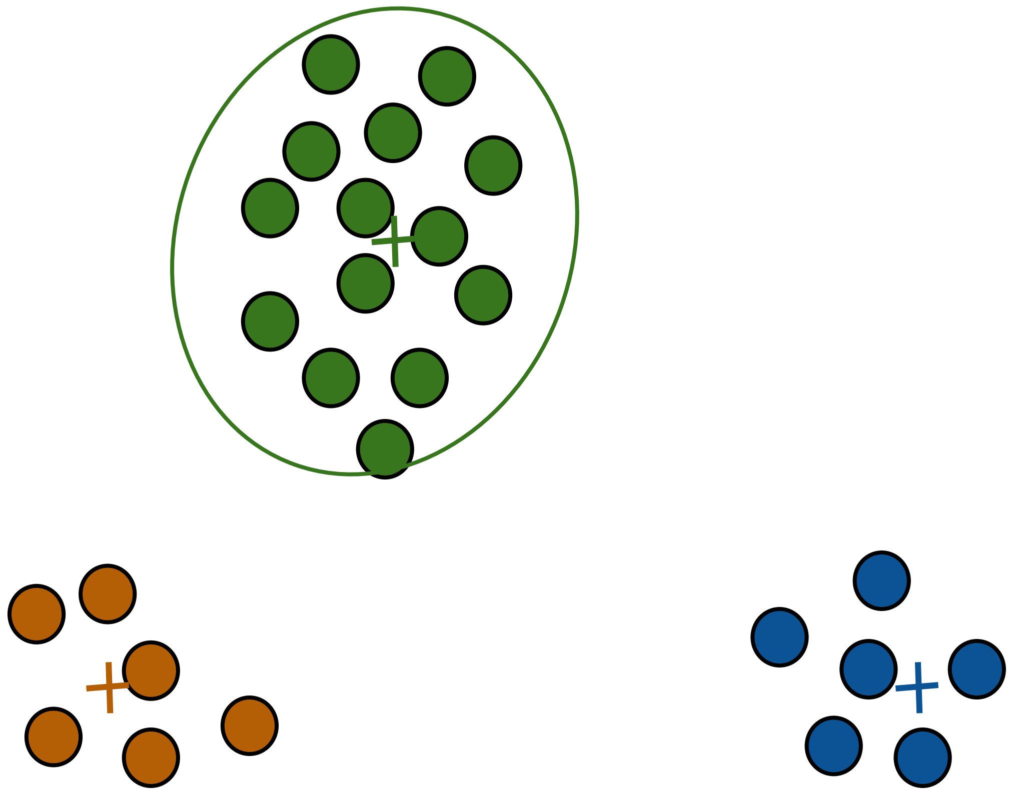

compute the true center
of each cluster

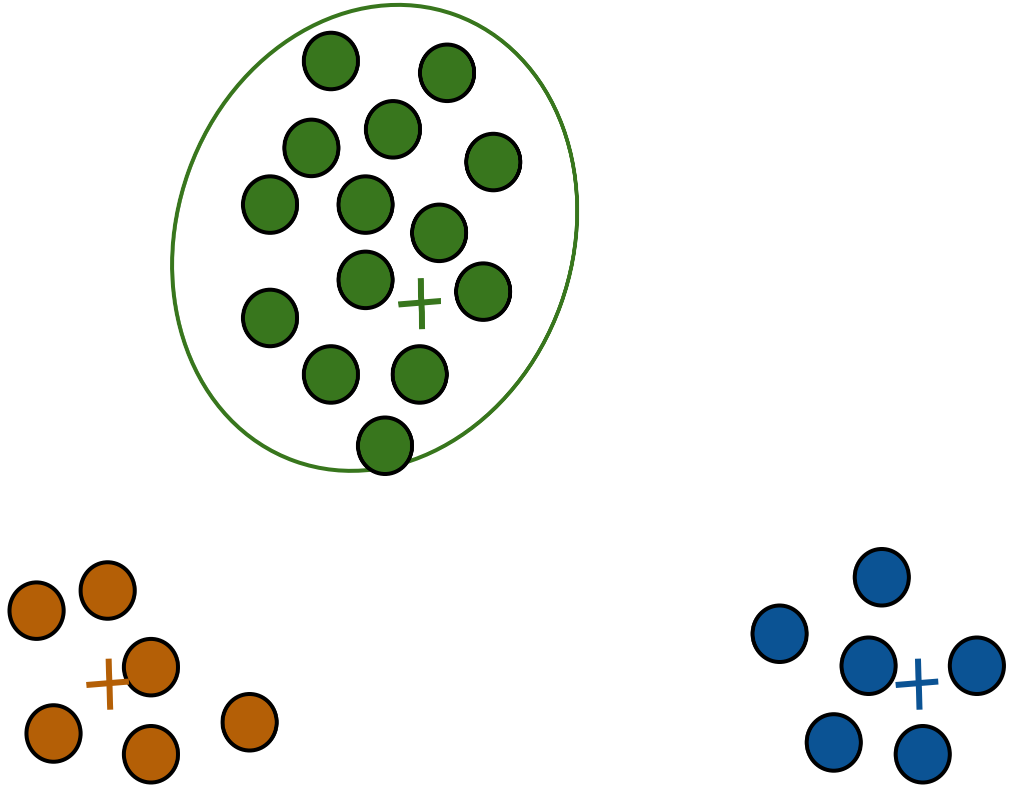


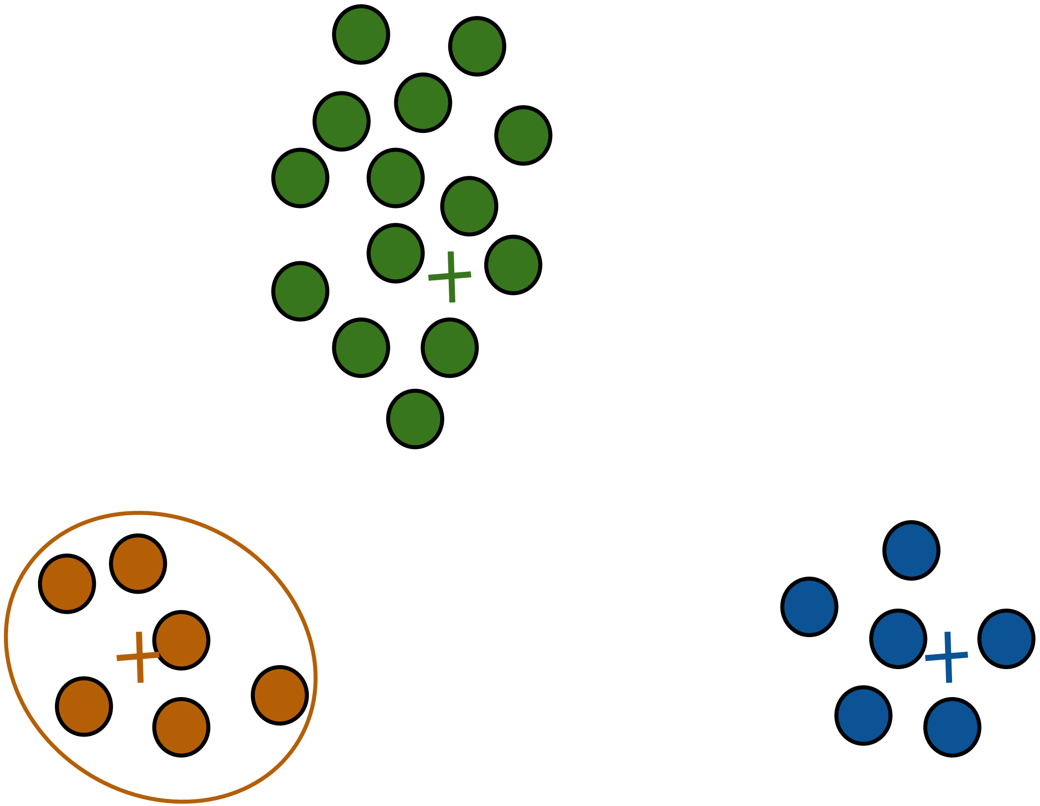


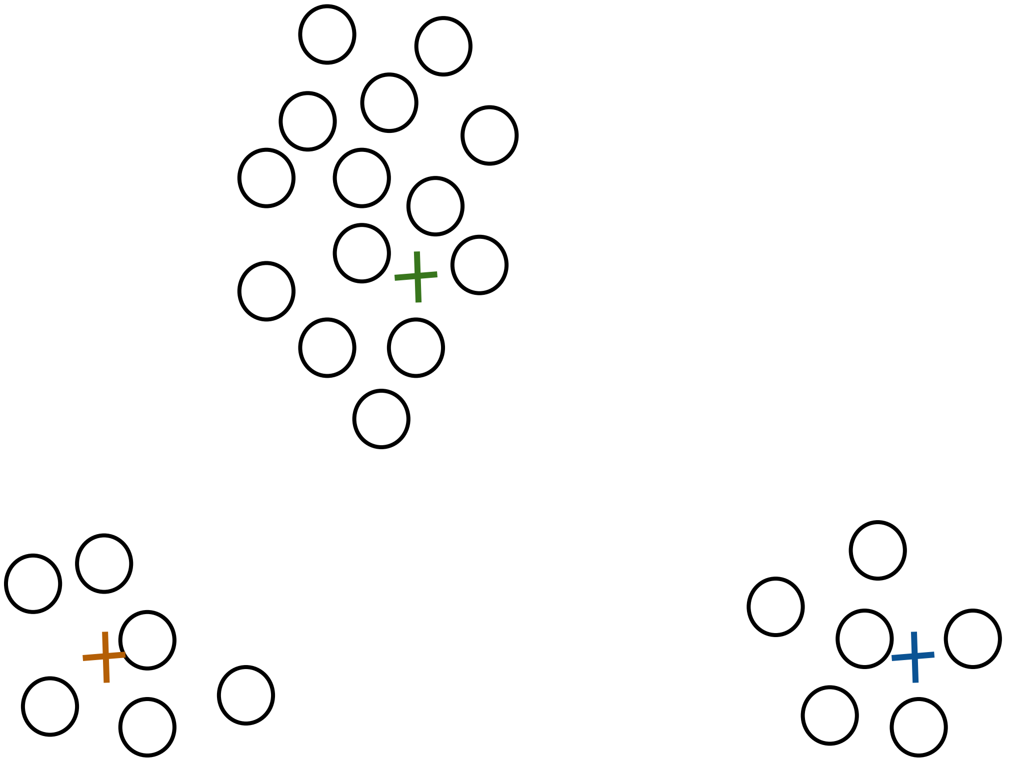


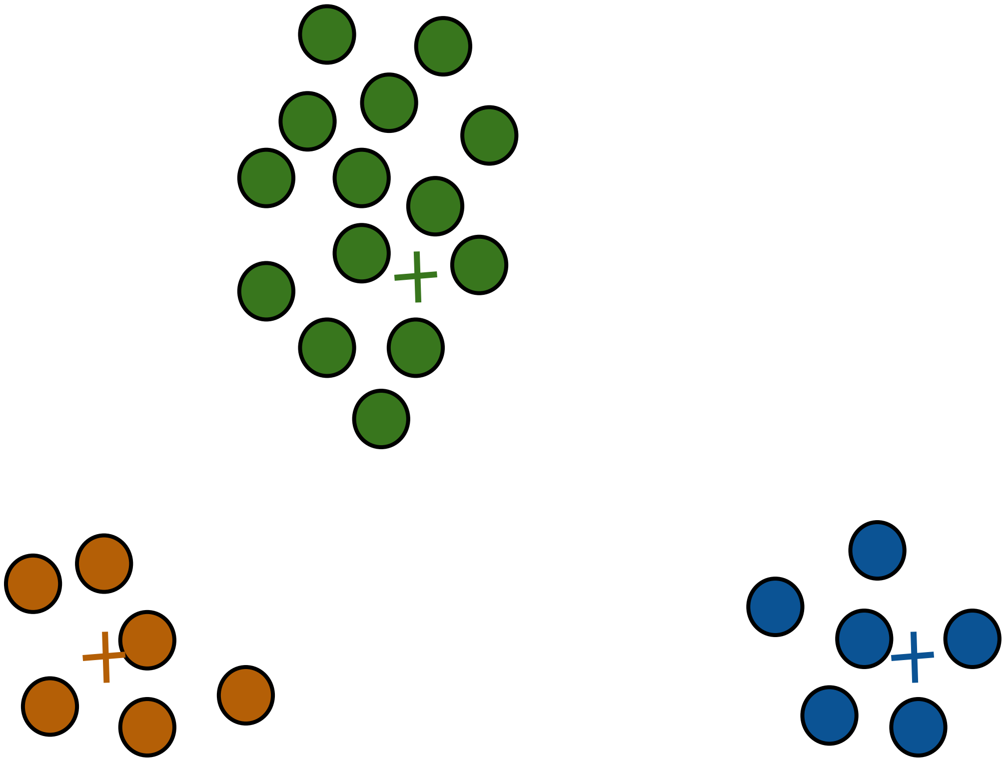


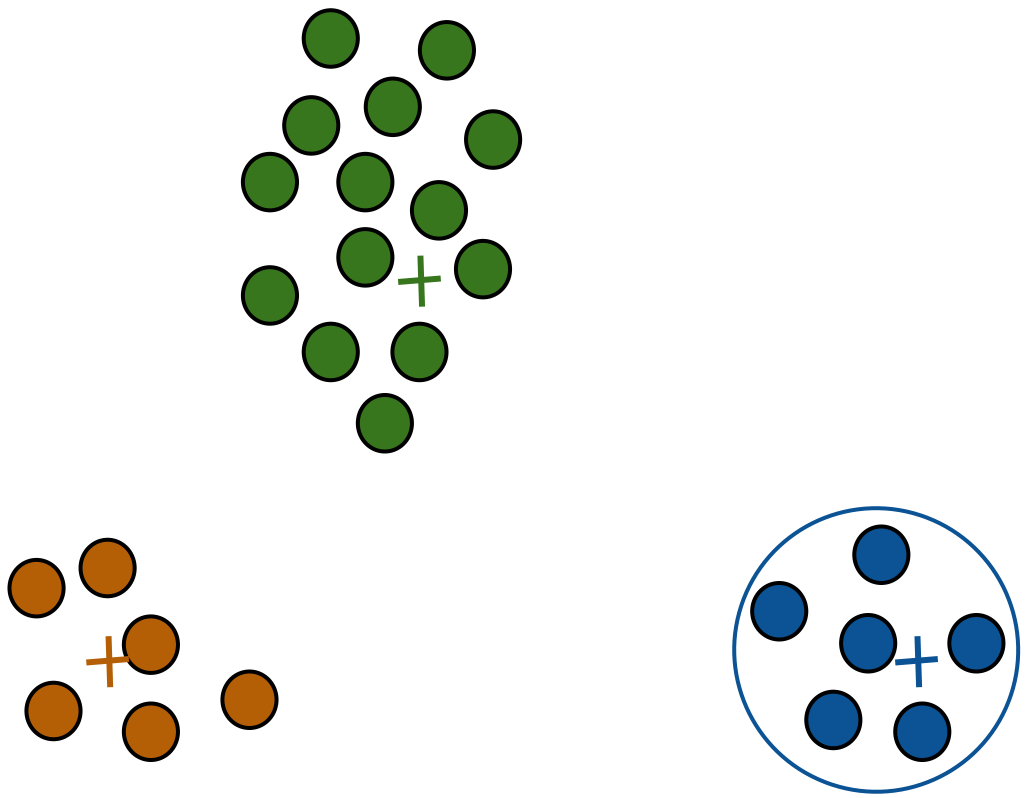


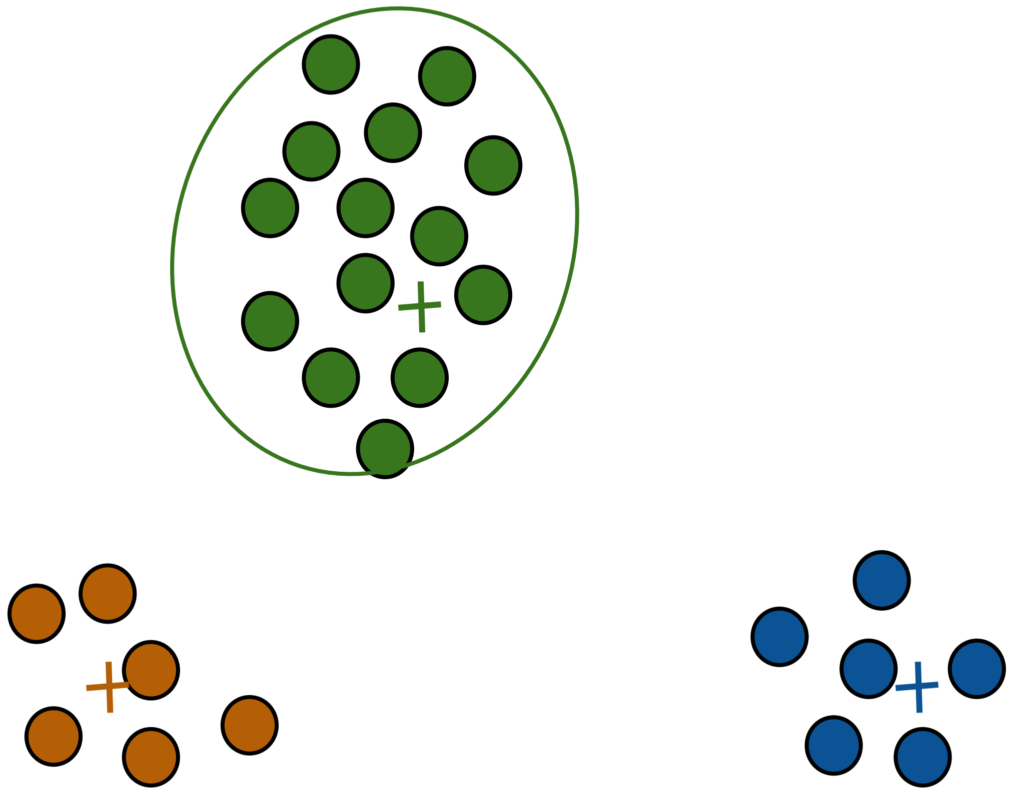


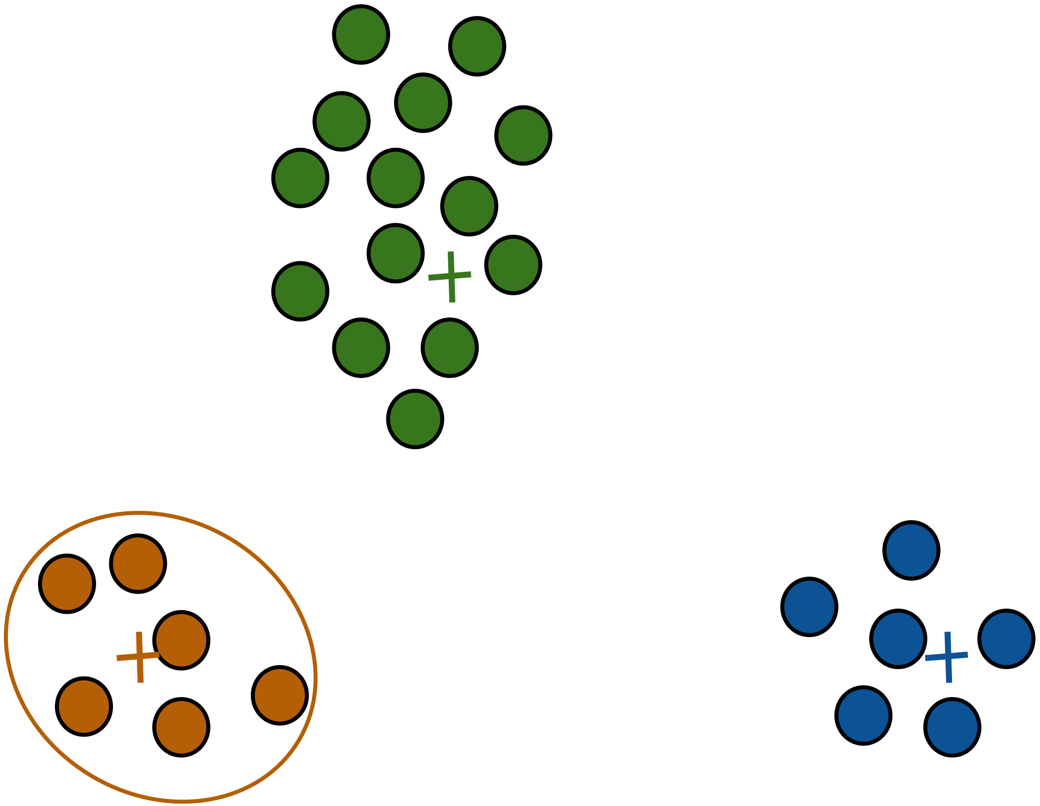


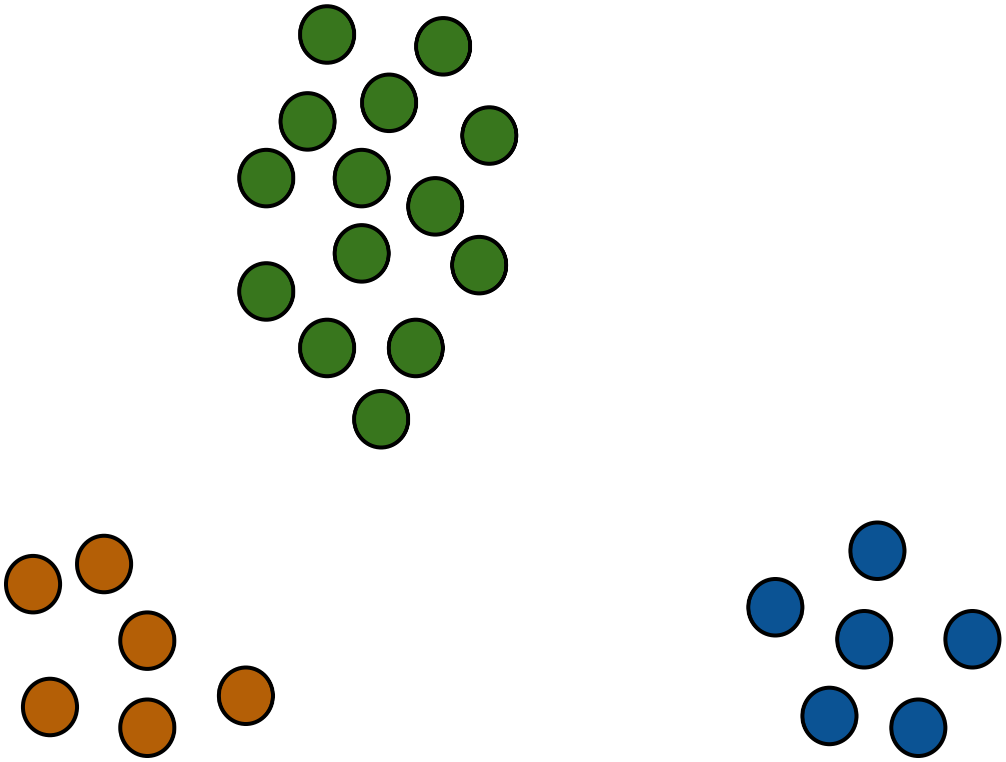




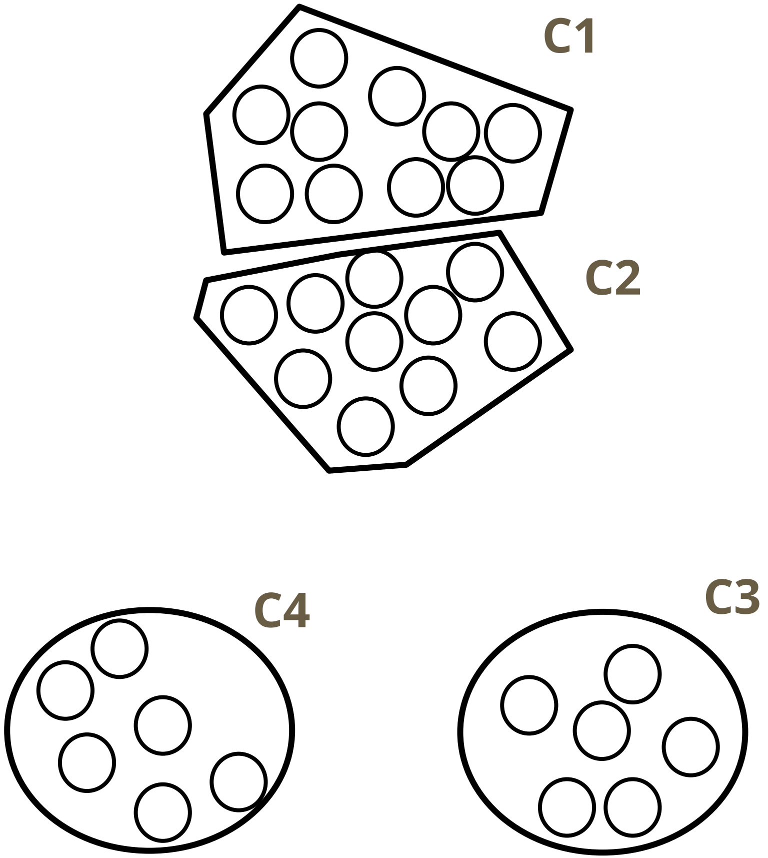


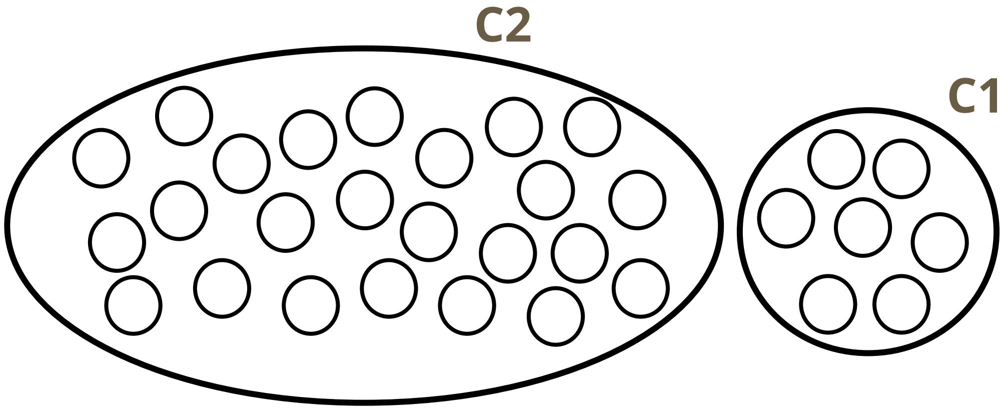


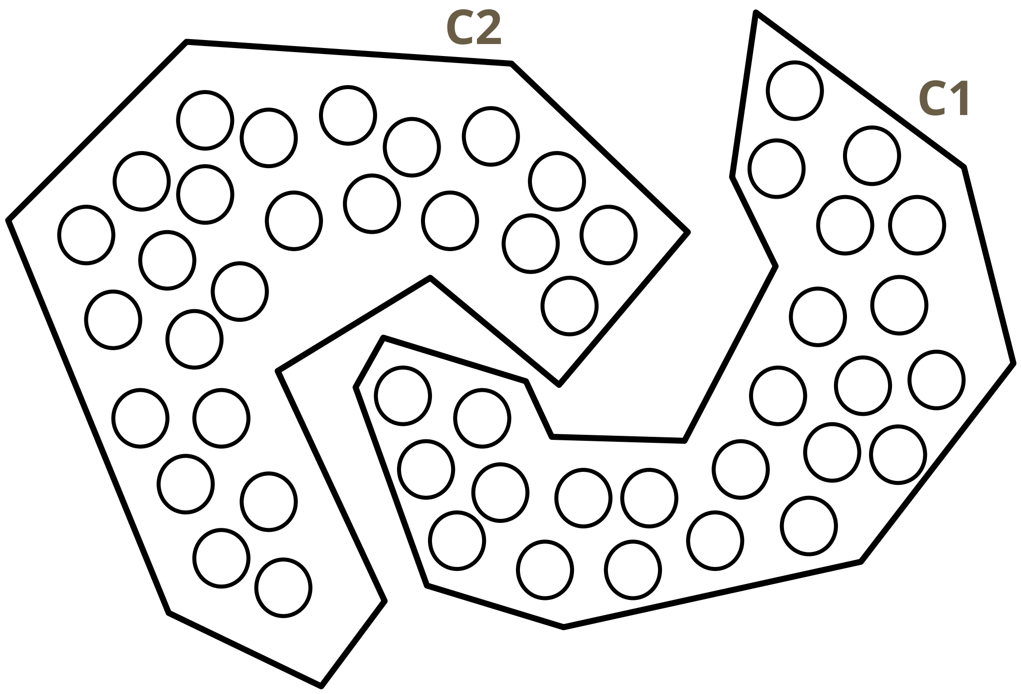


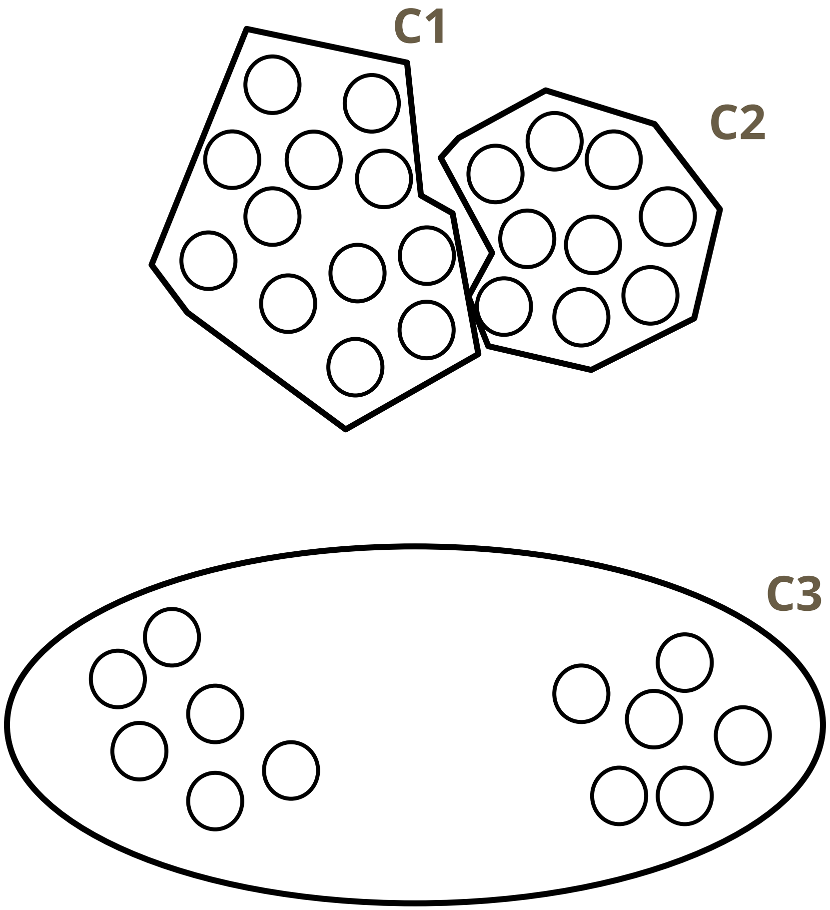


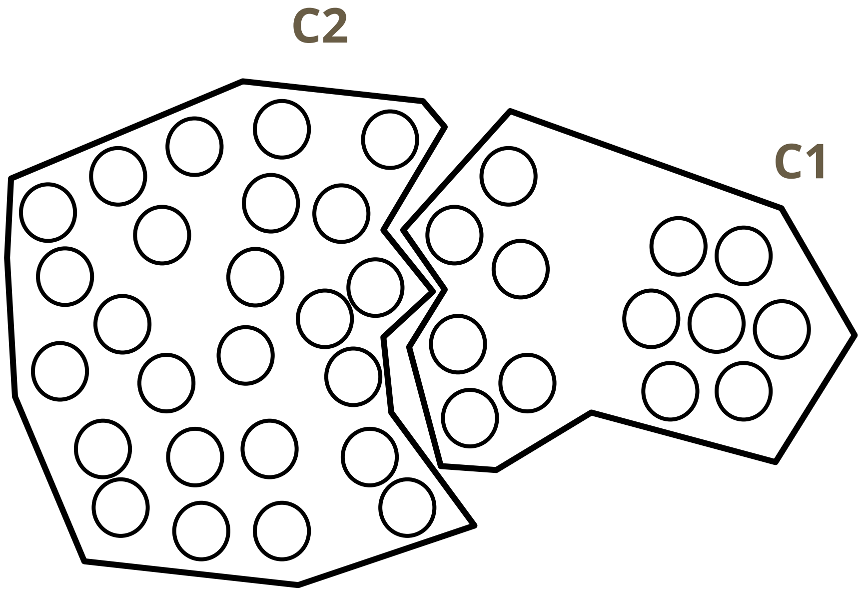
Questions

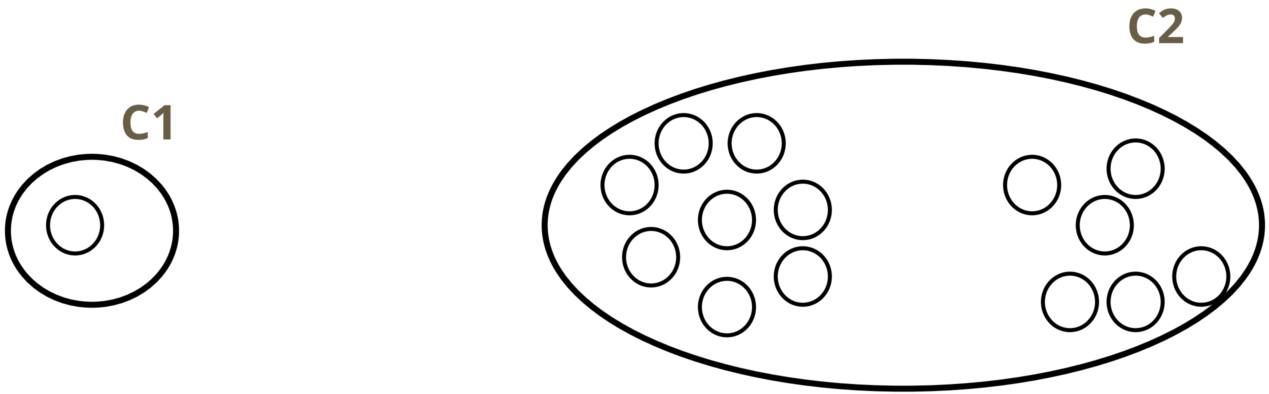

C1
C2
C4
C3


C2
C1


C2
C1


C1
C2
C3


C2
C1


C1
C2