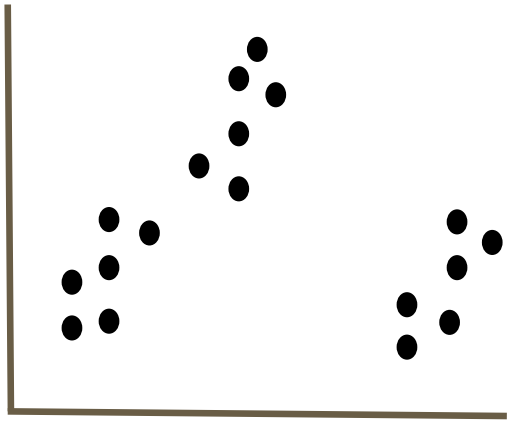
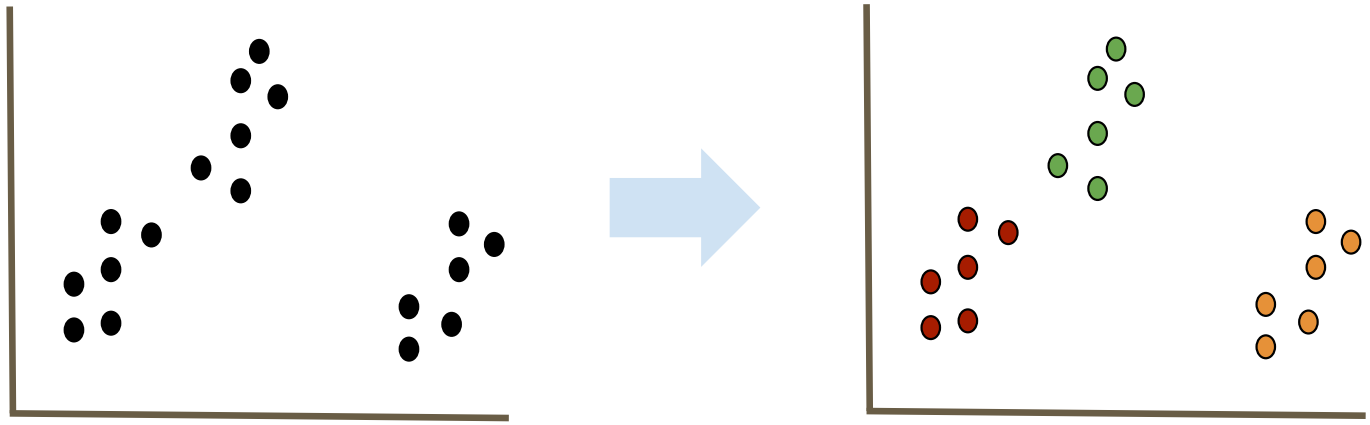
Clustering - Kmeans

Boston University CS 506 - Lance Galletti

What is a Clustering



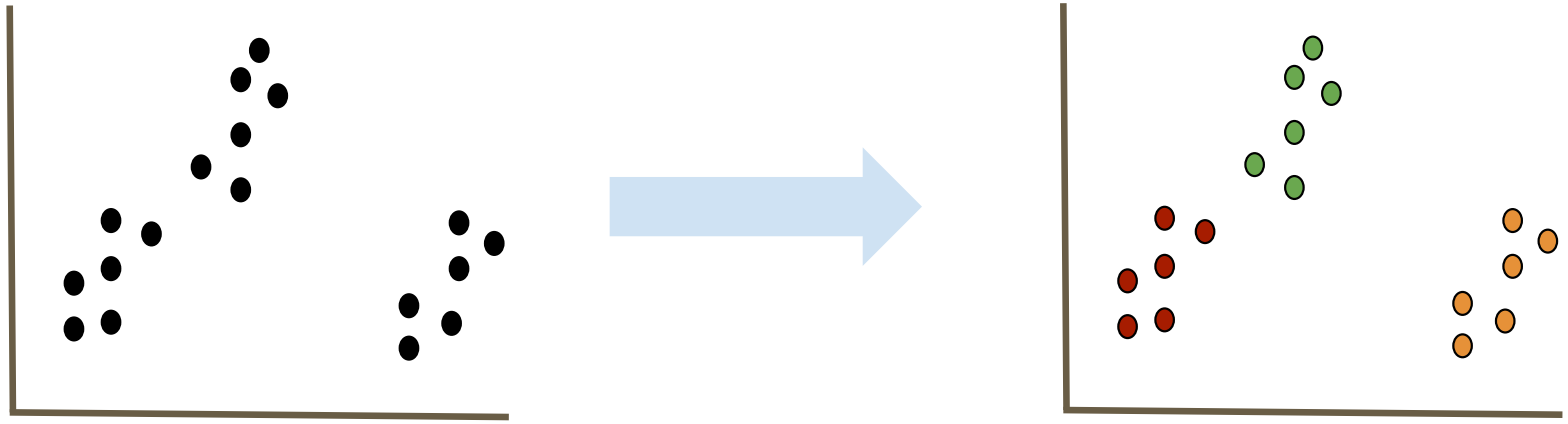
What is a Clustering



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



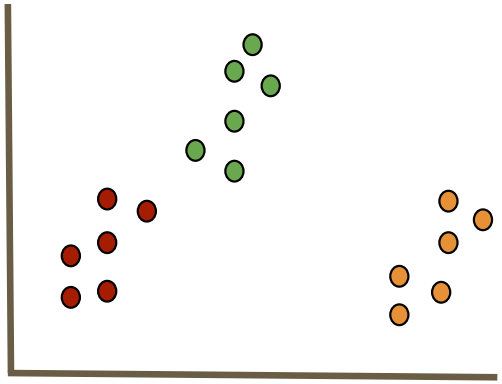
Applications

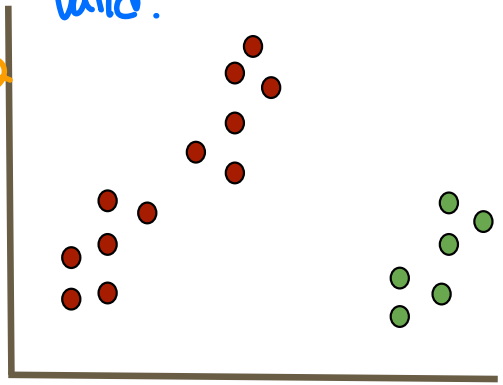
- Outlier detection / anomaly detection
  - Data Cleaning / Processing
  - Credit card fraud, spam filter etc.
- Feature Extraction
- Filling Gaps in your data
  - Using the same marketing strategy for similar people
  - Infer probable values for gaps in the data (similar users could have similar hobbies,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

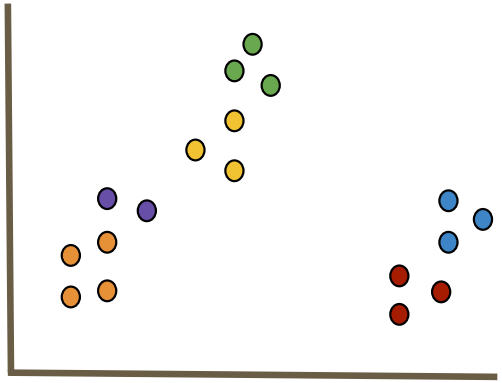
Clusters can be Ambiguous

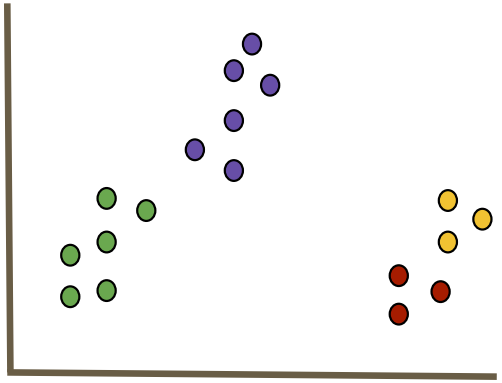
we could have multiple valid clustering

Any of these clusters could be valid.









Types of Clusterings

Partitional
Each object belongs to exactly one cluster

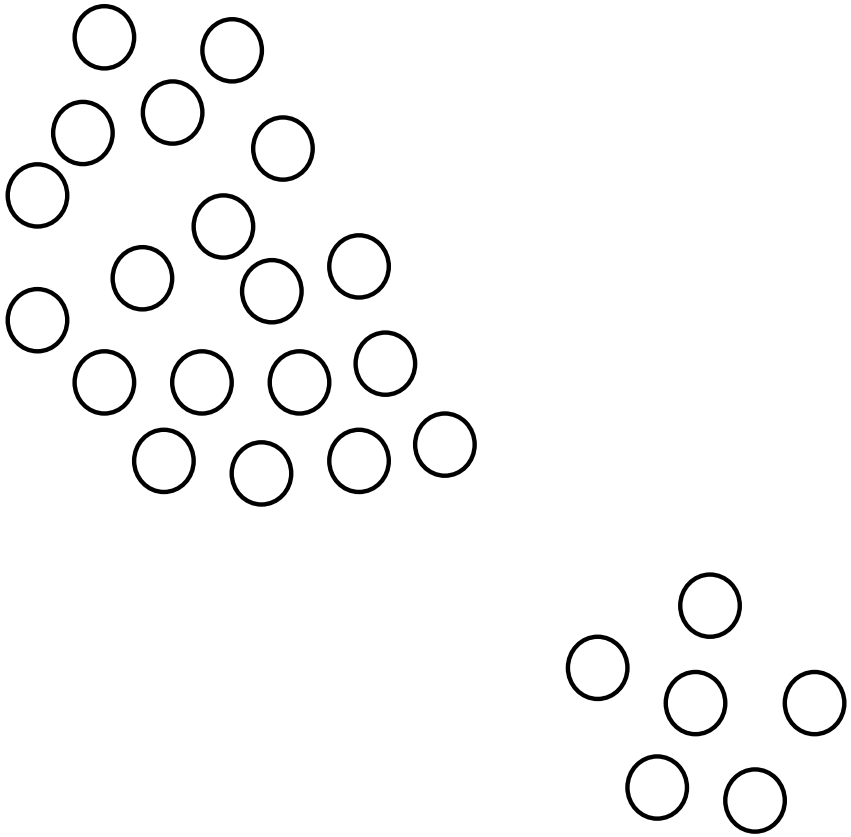
Hierarchical
A set of nested clusters organized in a tree

Density-Based
Defined based on the local density of points

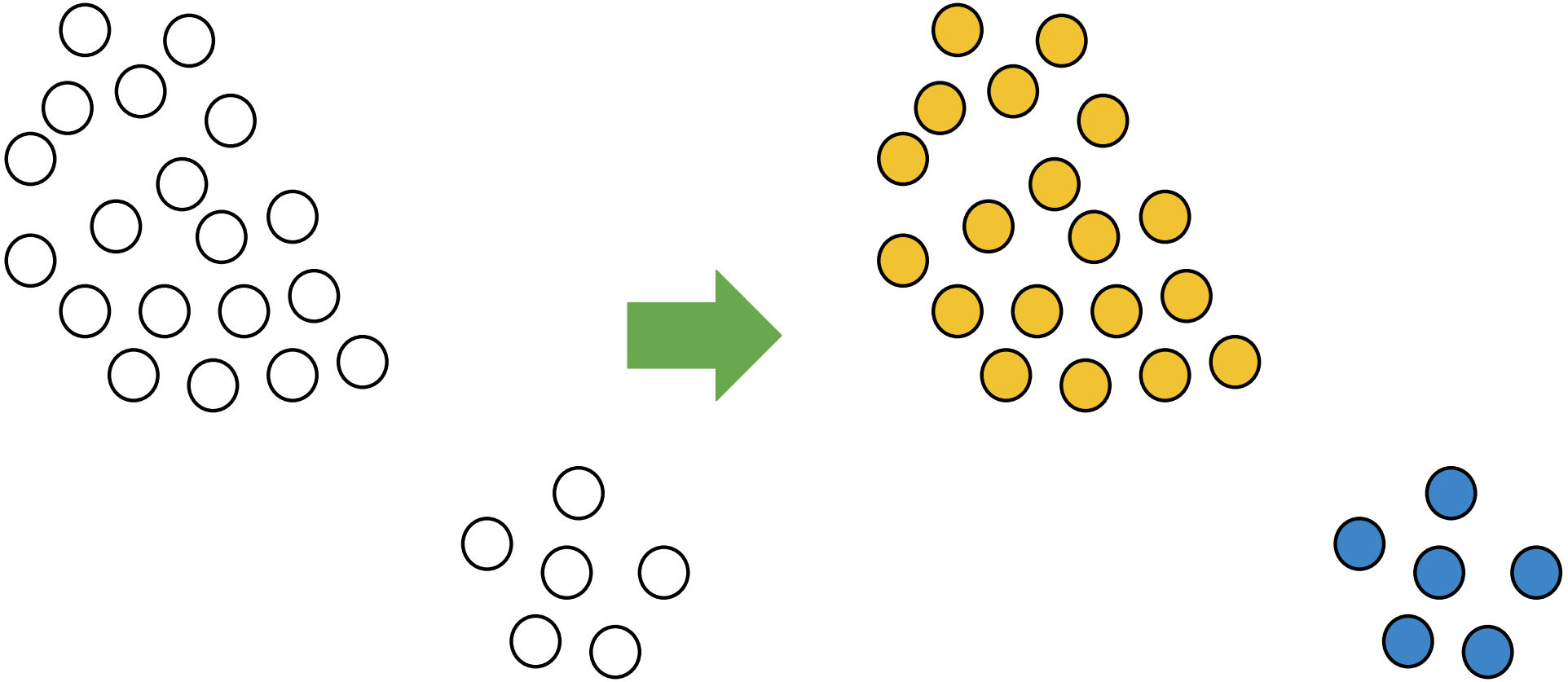
Soft Clustering
Each point is assigned to every cluster with a certain probability

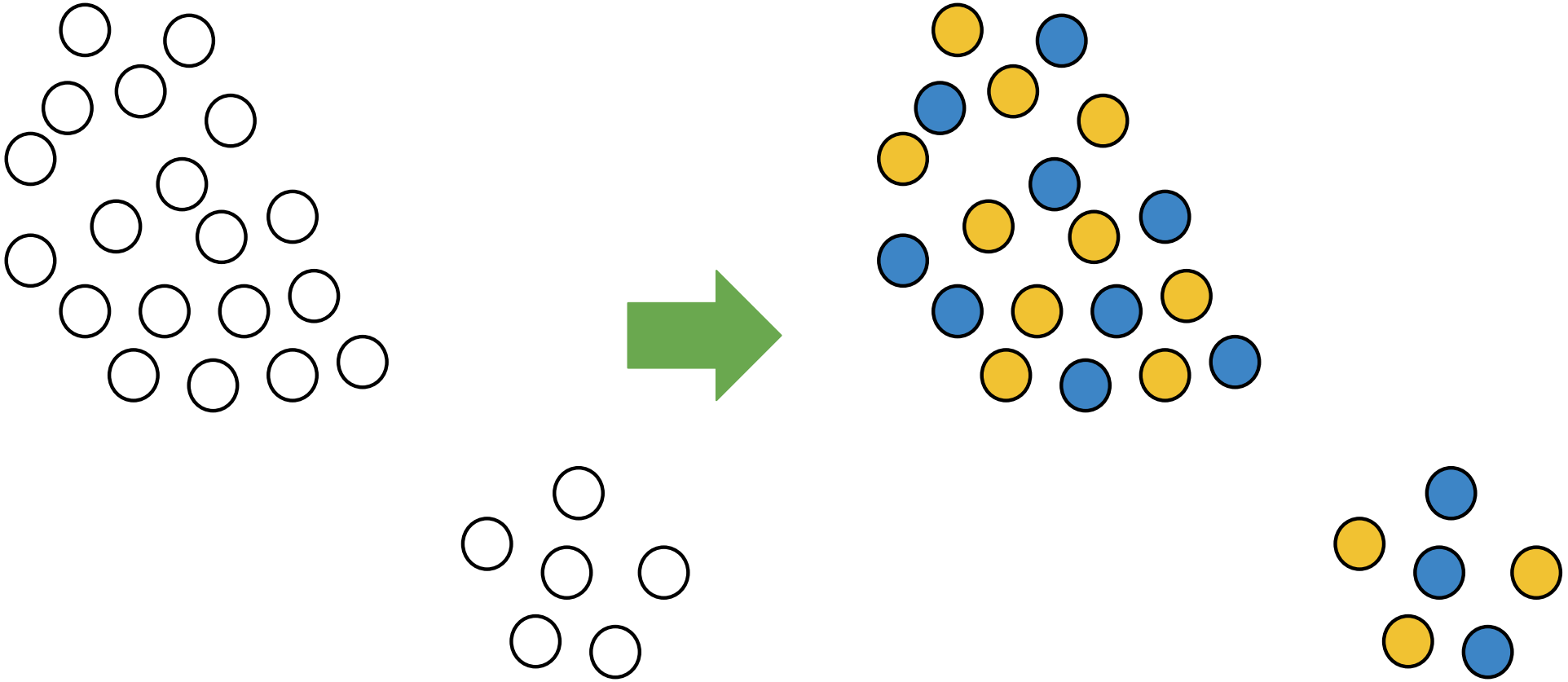
Partitional Clustering

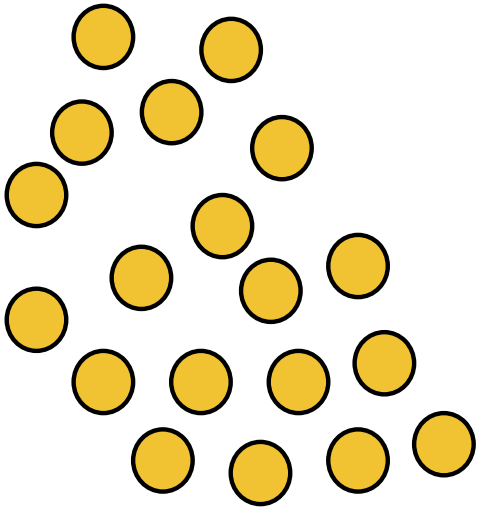
Partitional Clustering

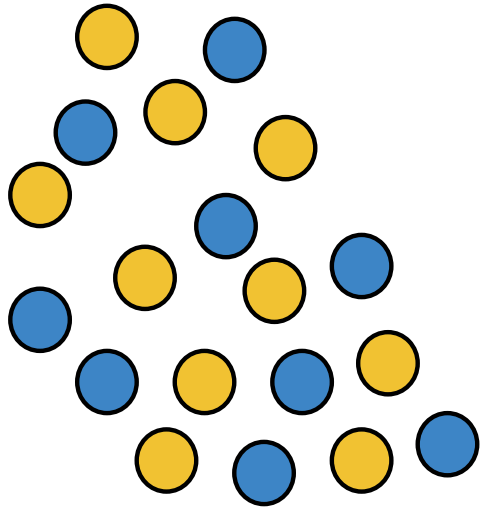


Goal: partition dataset into k partitions

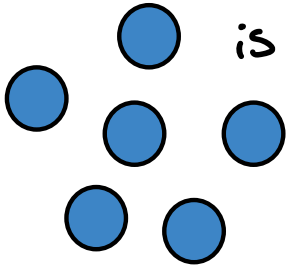






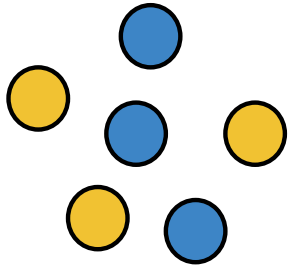


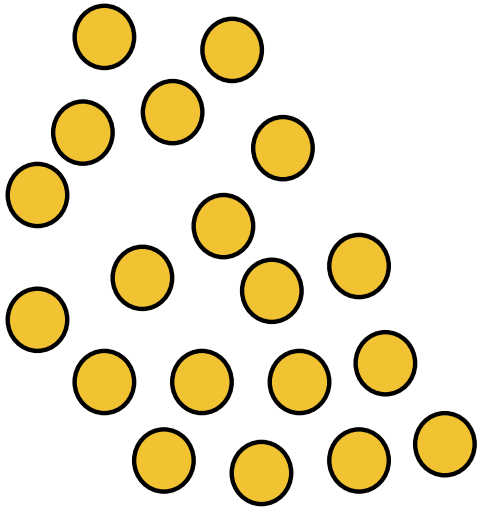
Variance of blue cluster is much smaller than



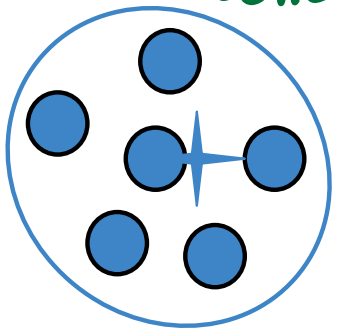
Here

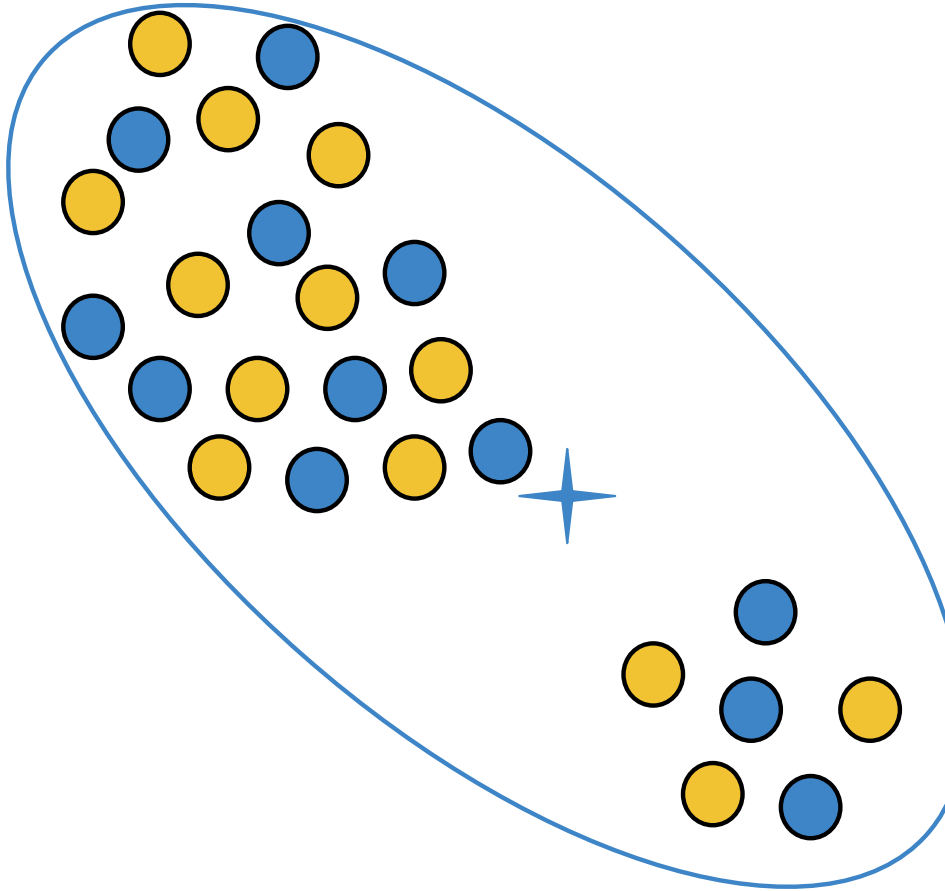
Variance of blue circles difference are huge.

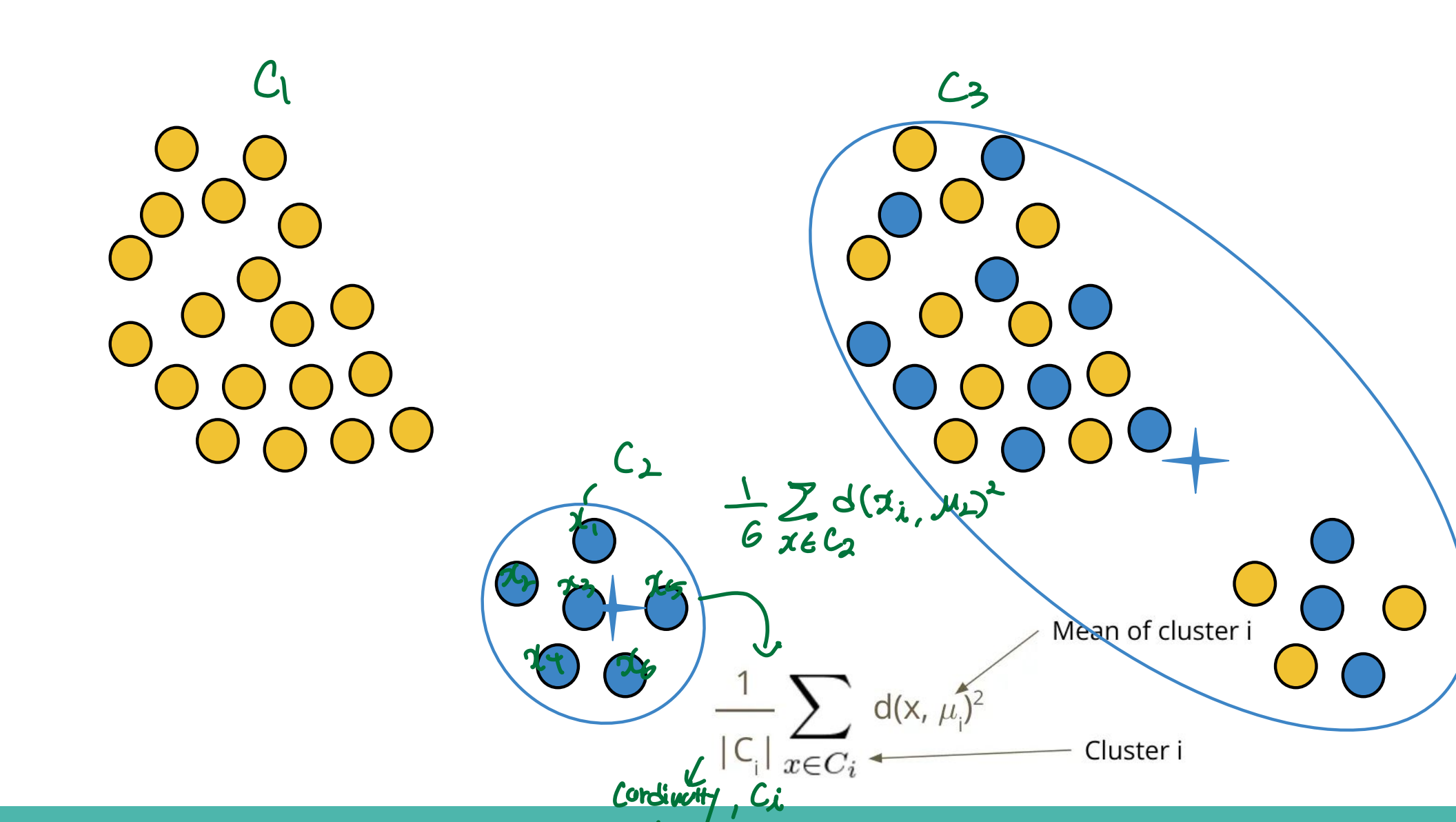




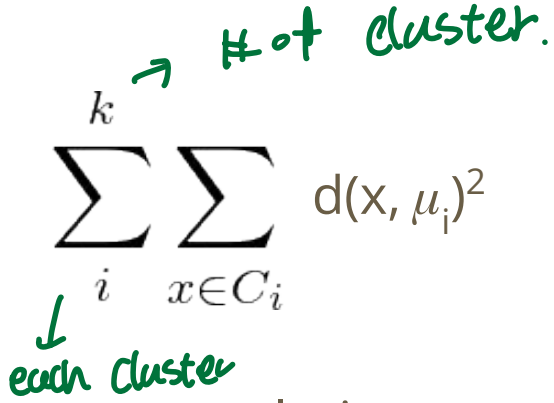
Sum of the variances are smaller, then clustering is better.






C1
C3
C2
$\frac{1}{6}\sum_{x \in C_2} d(x_i, \mu_2)^2$
Mean of cluster i
$\frac{1}{|C_i|}\sum_{x \in C_i} d(x, \mu_i)^2$
Cluster i
Cardinality, C_i

Cost Function



- 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**:

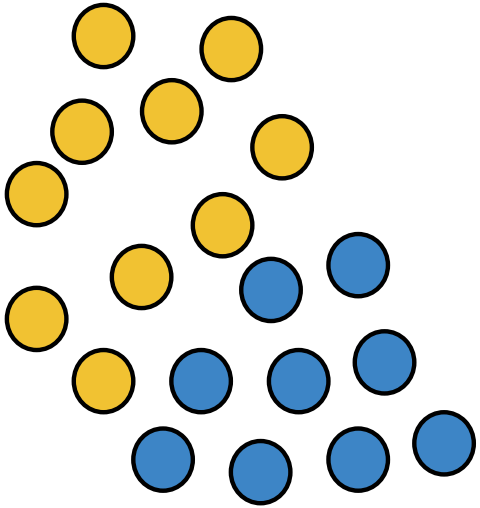
$$\sum_{i}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

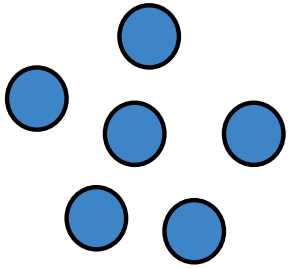
everything is their own cluster

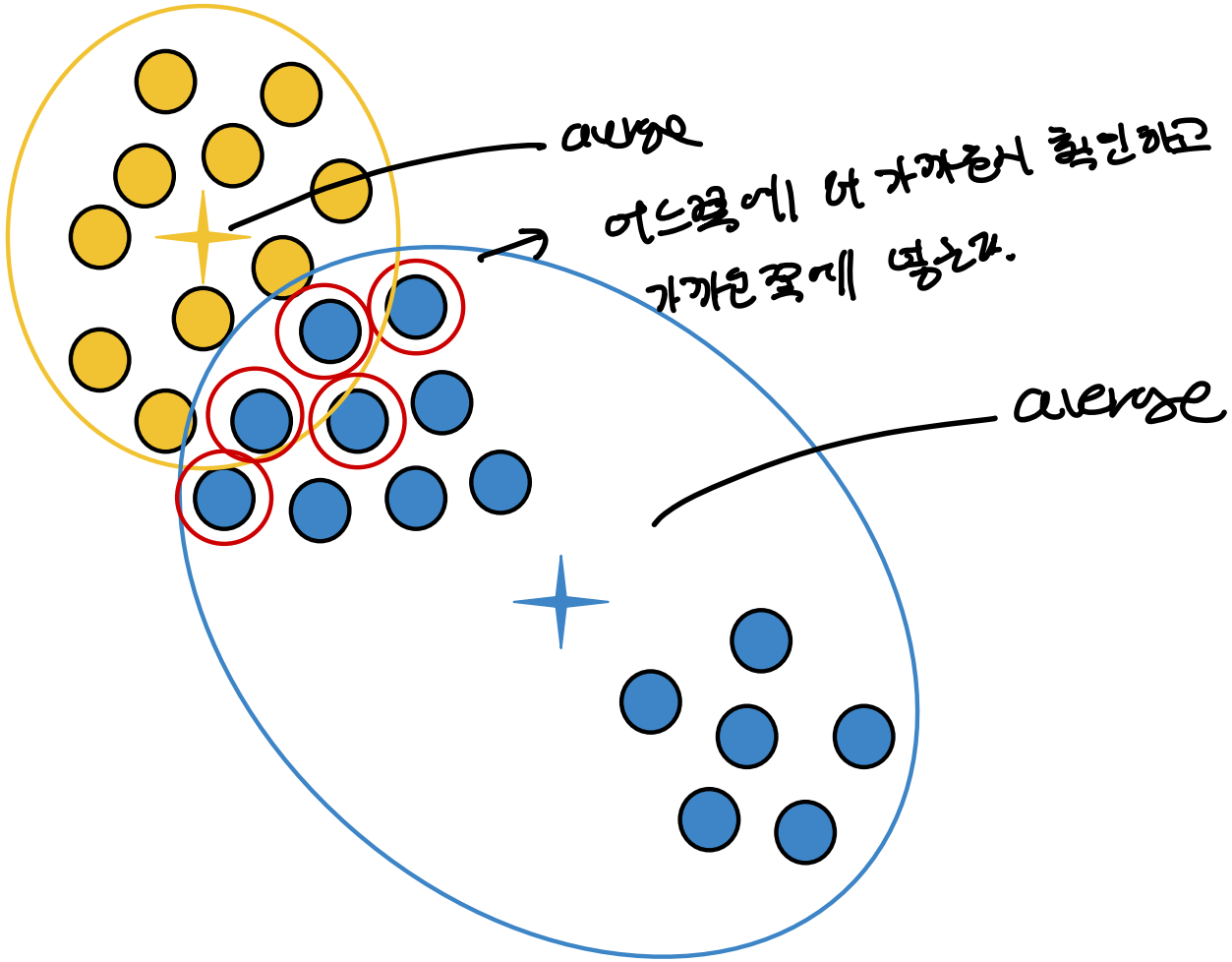
When **k=1** and **k=n** this is easy. Why?

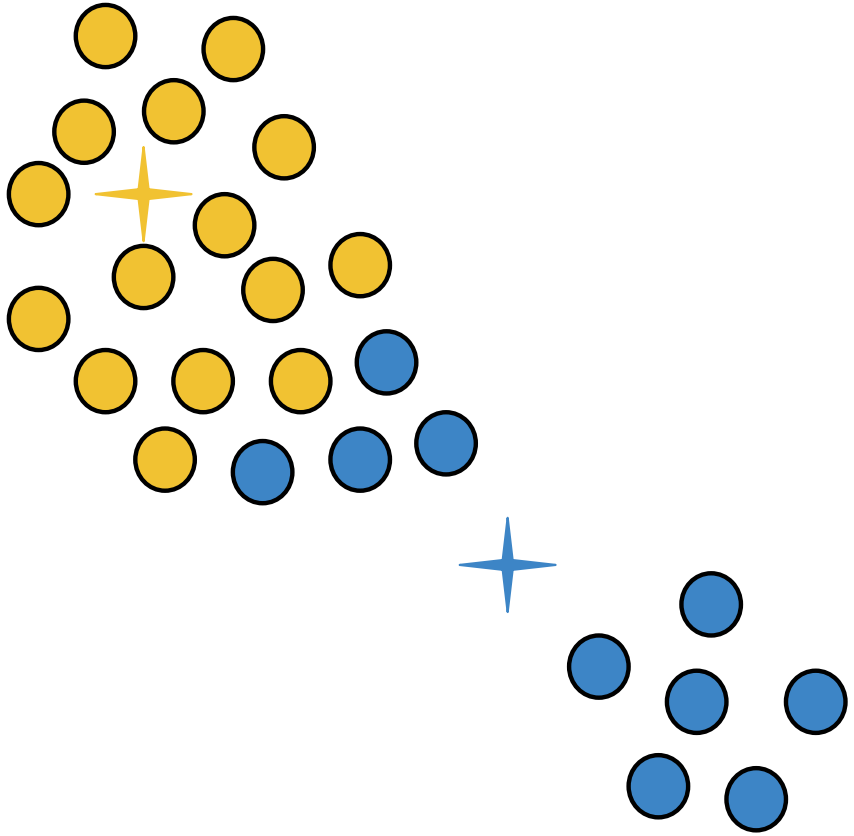
↳ Every thing is belong to 1 cluster

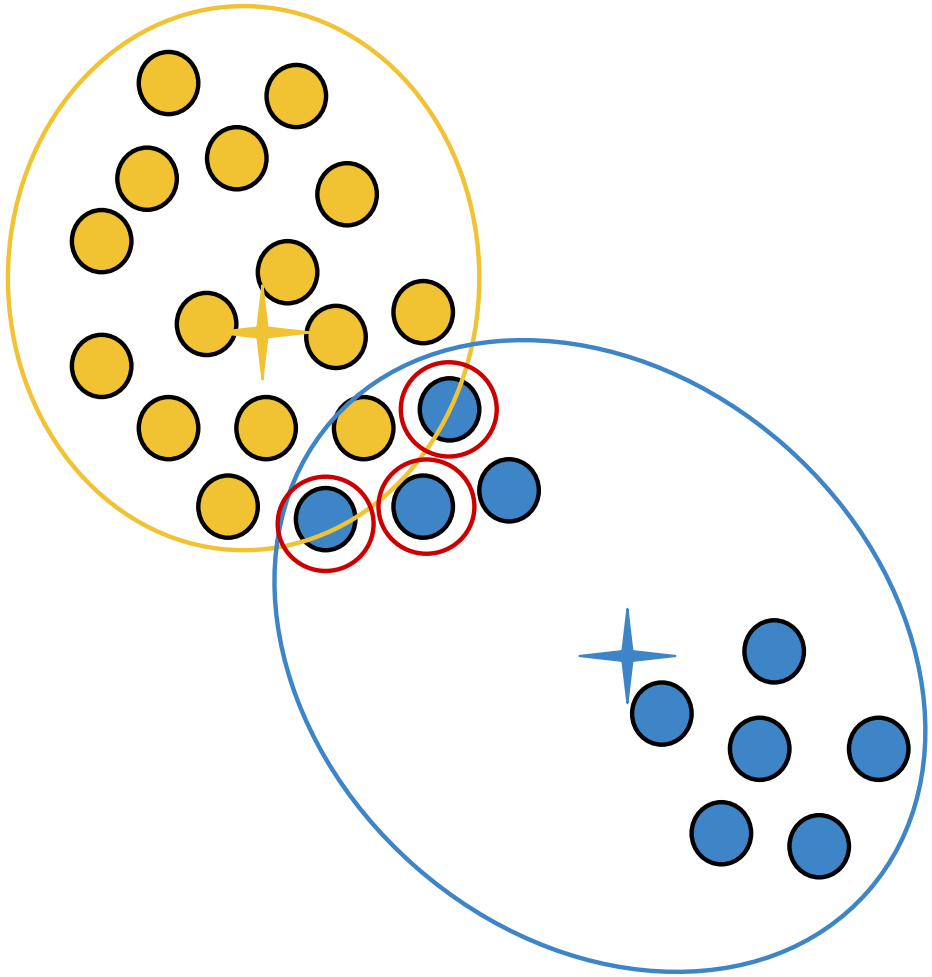
When $\mathbf{x_i}$ lives in more than 2 dimensions, this is a very difficult (**NP-hard**) problem

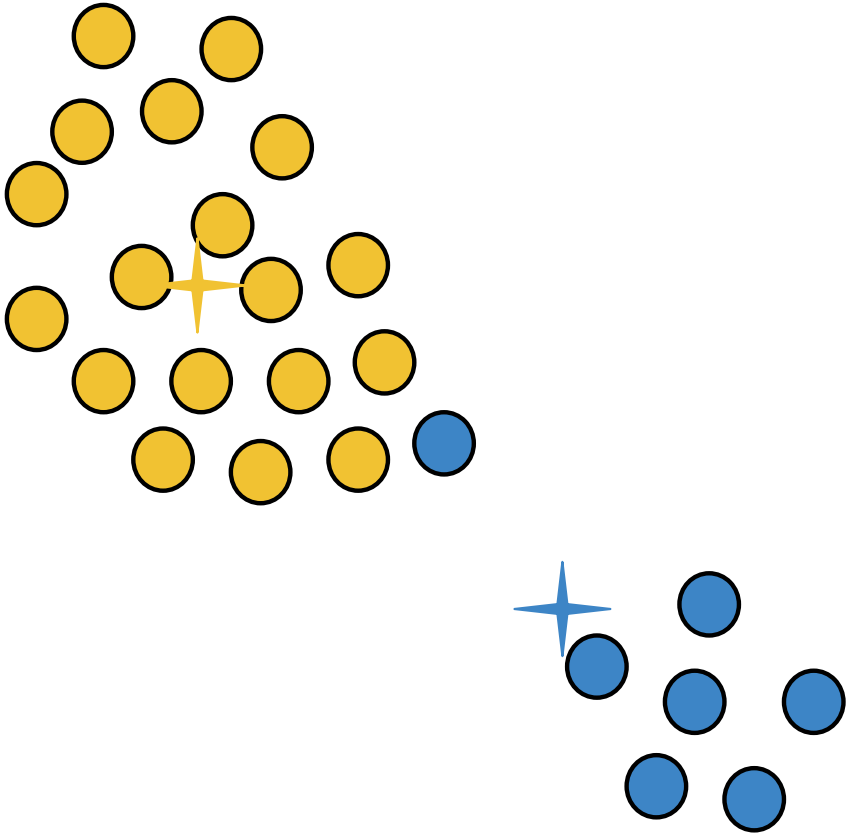


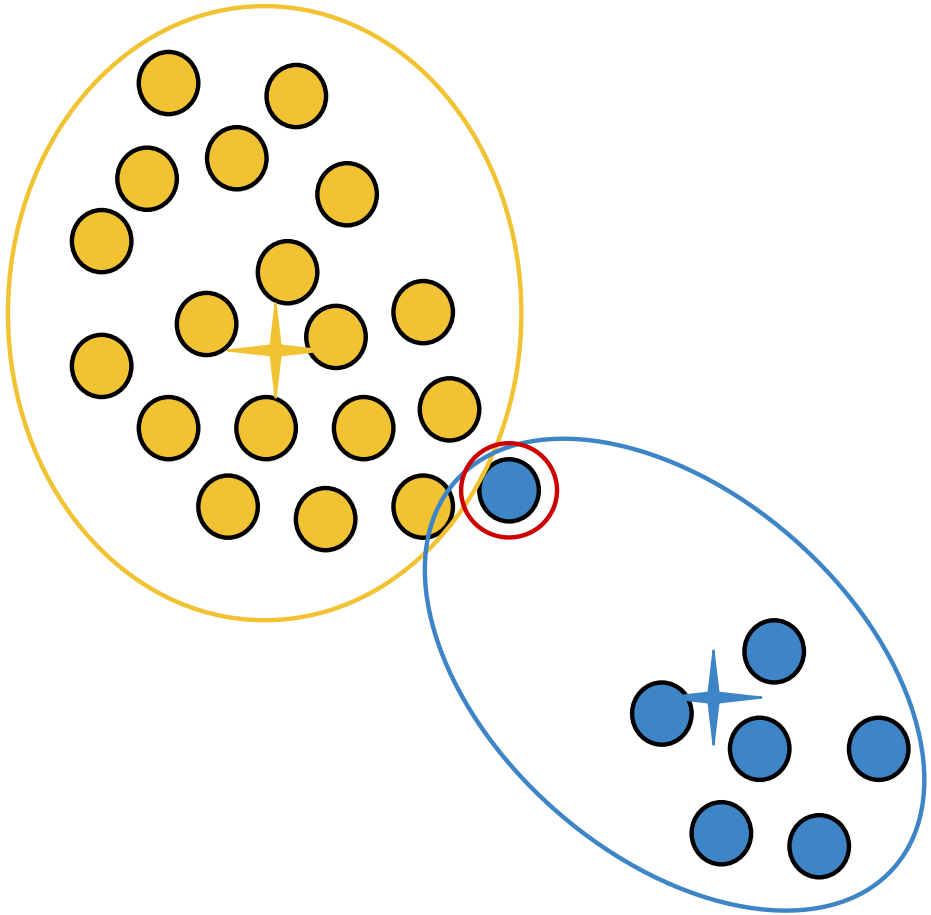


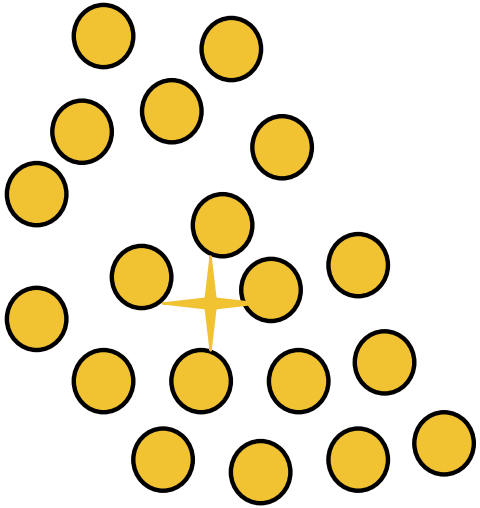

average
어느쪽에 더 가까운지 확인하고
가까운쪽에 넣는다.
average



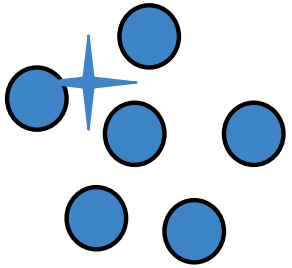






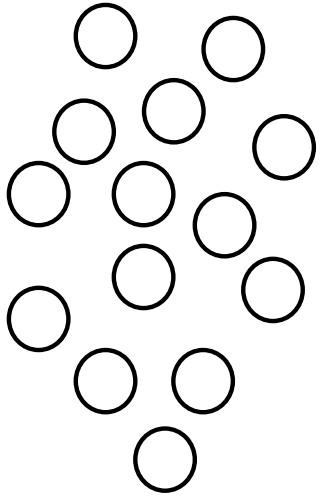


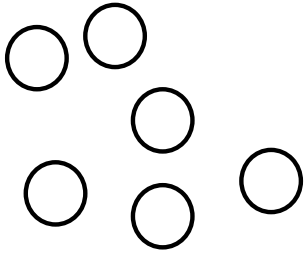
어떻게 kick start 할까?

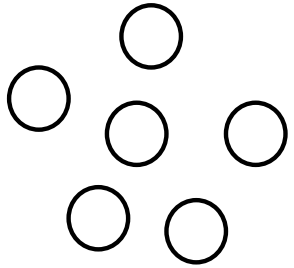


K-means - Lloyd's Algorithm → Randomly pick k centers

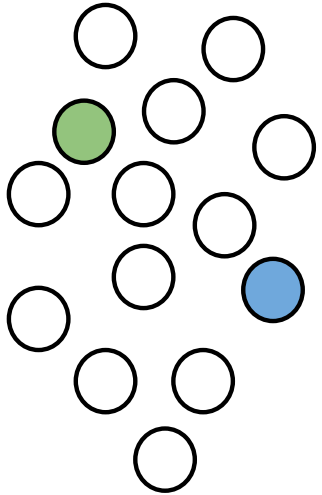
1. Randomly pick **k** centers $\{\mu_1, \ldots, \mu_k\}$
2. Assign each point in the dataset to its closest center
3. Compute the new centers as the means of each cluster
4. Repeat 2 & 3 until convergence



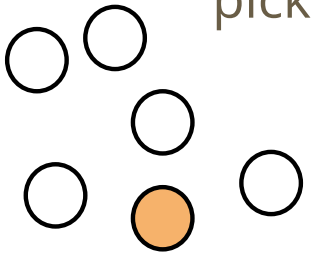


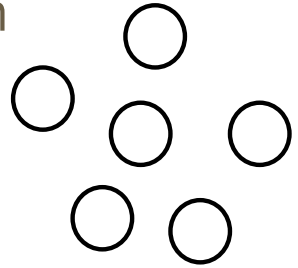


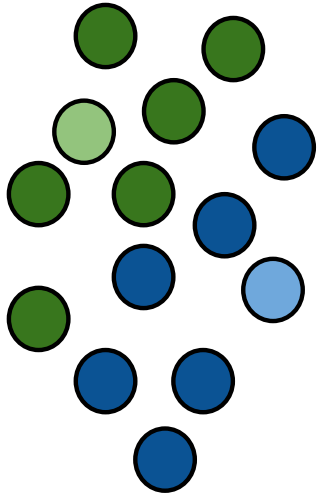
First kick start



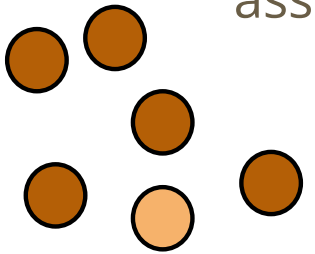
pick k centers at random

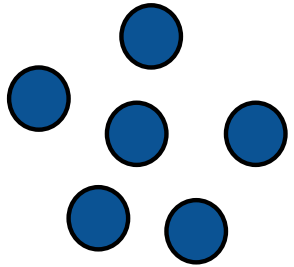


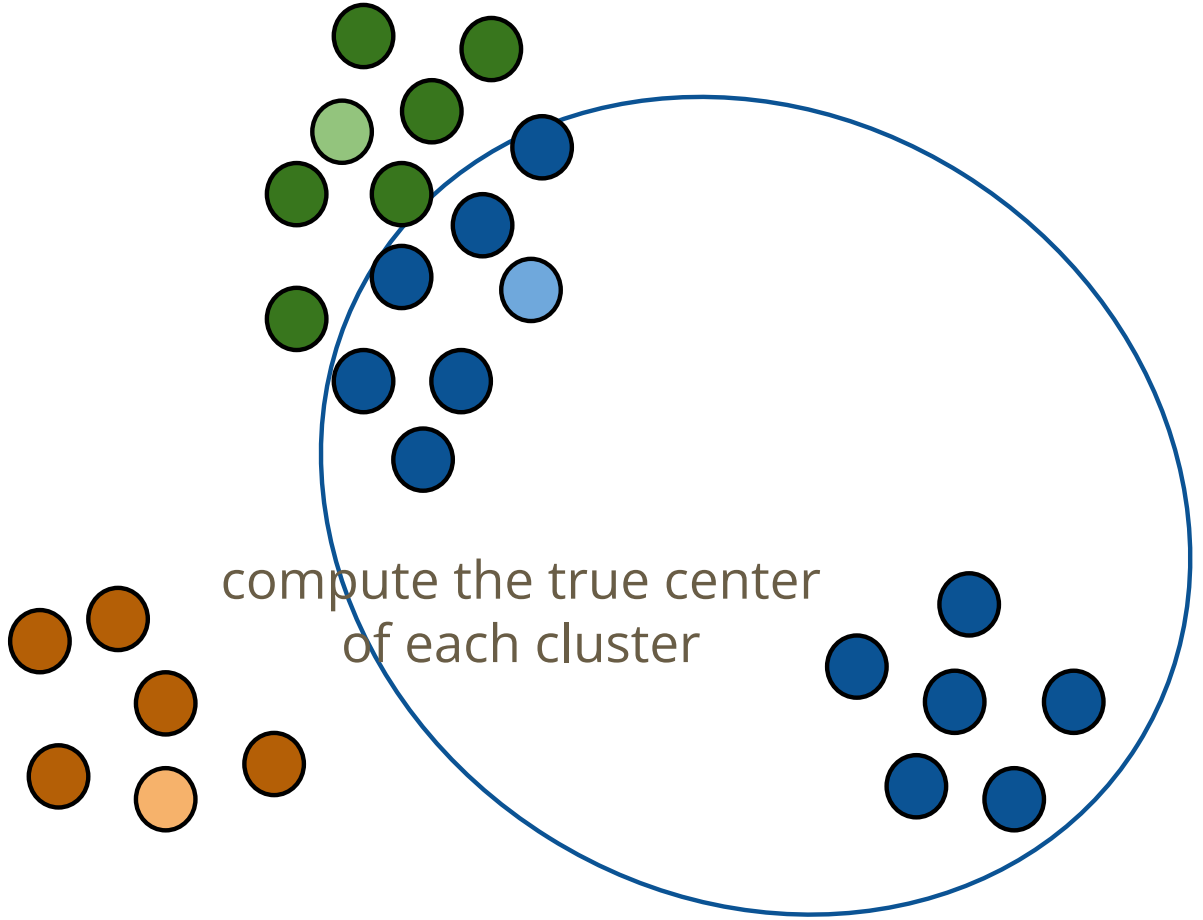


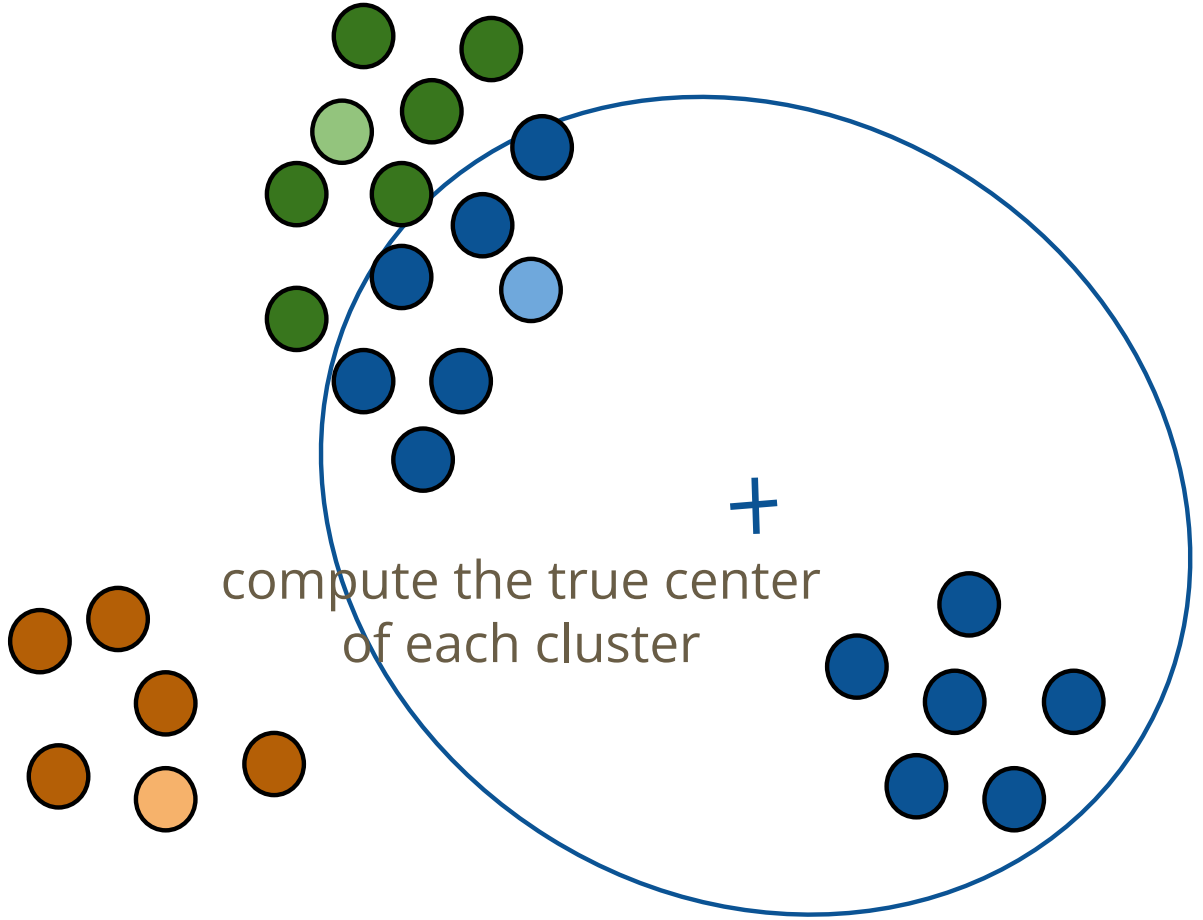


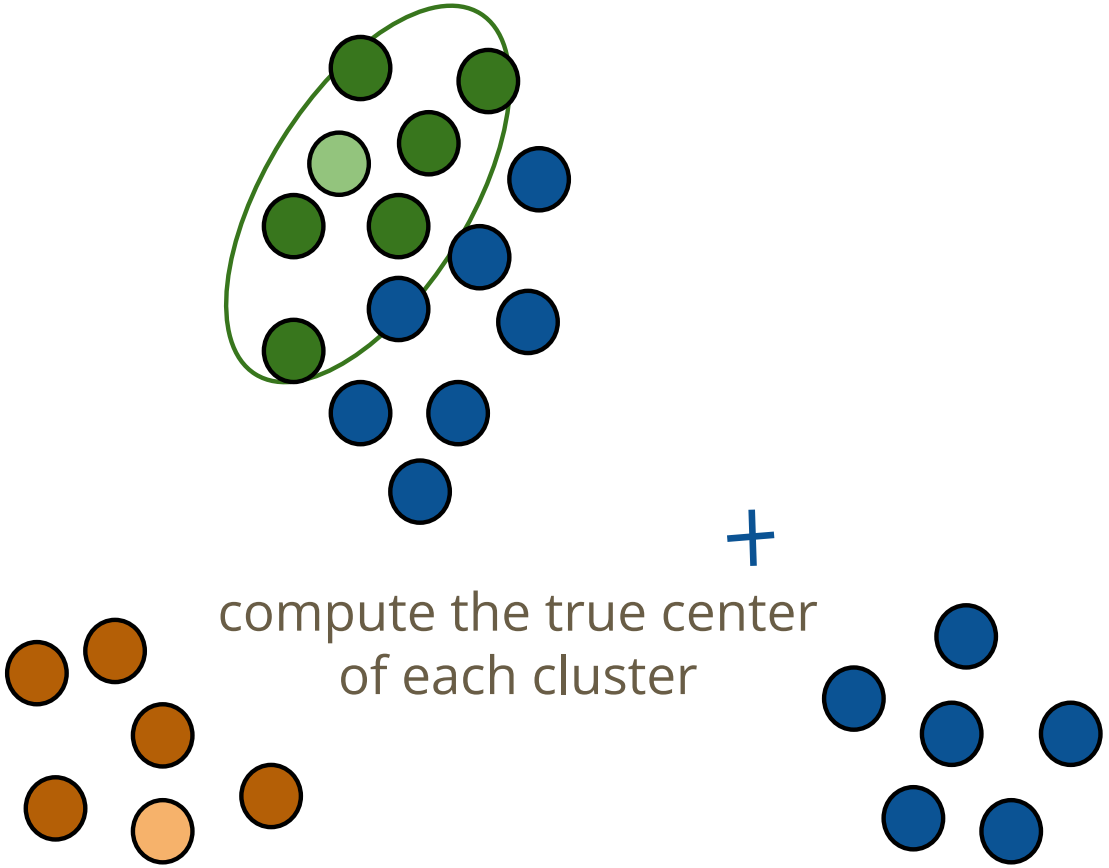
assign points to closest center

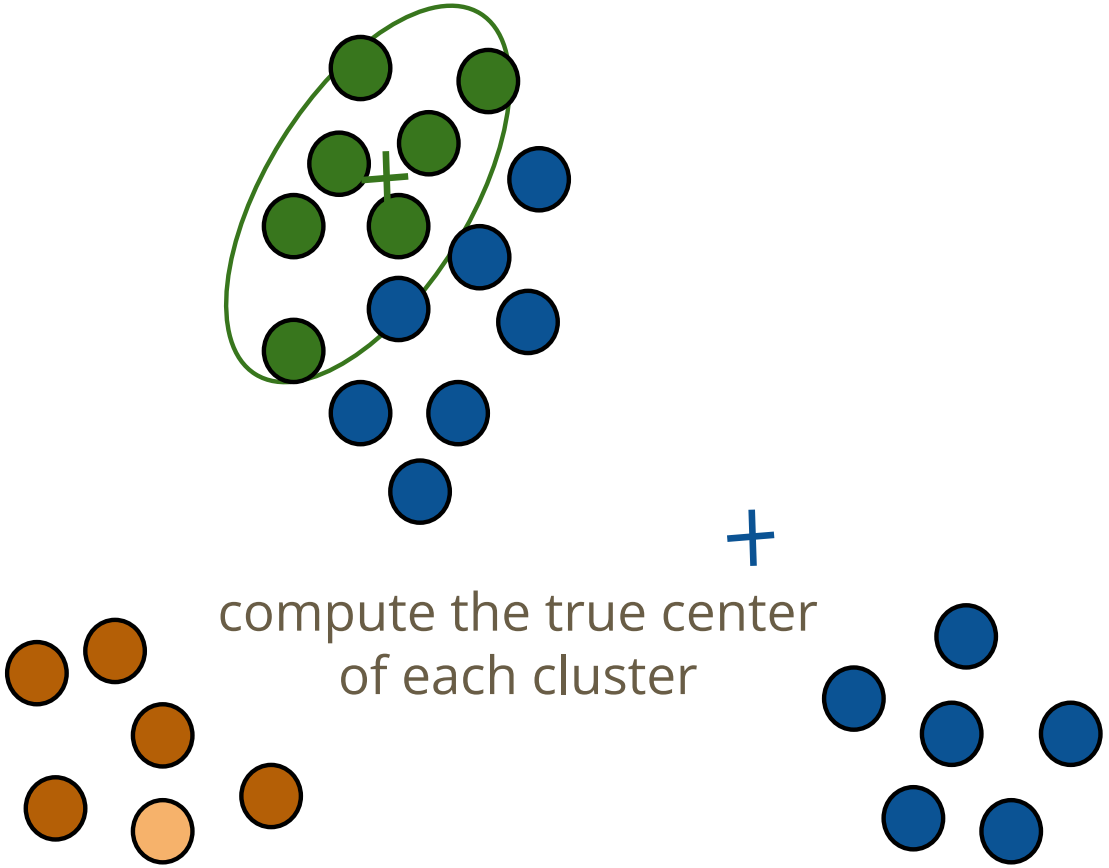


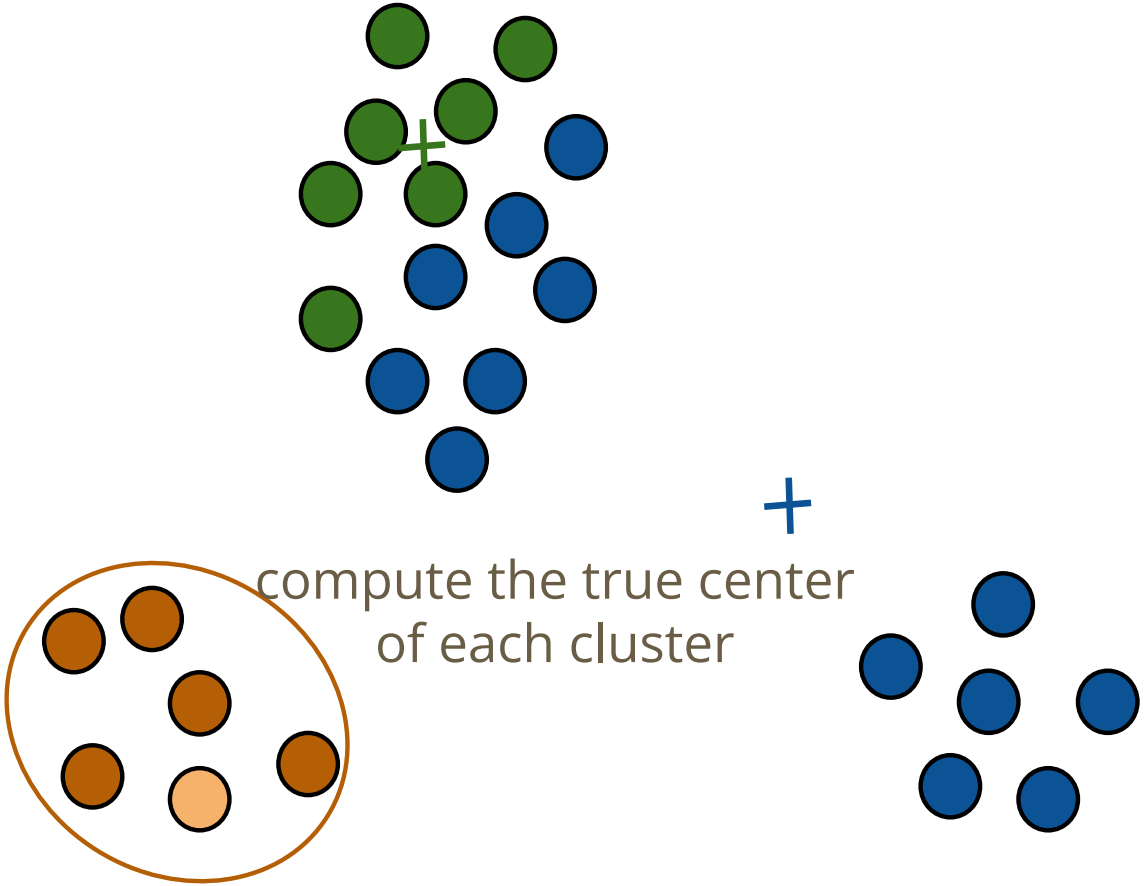


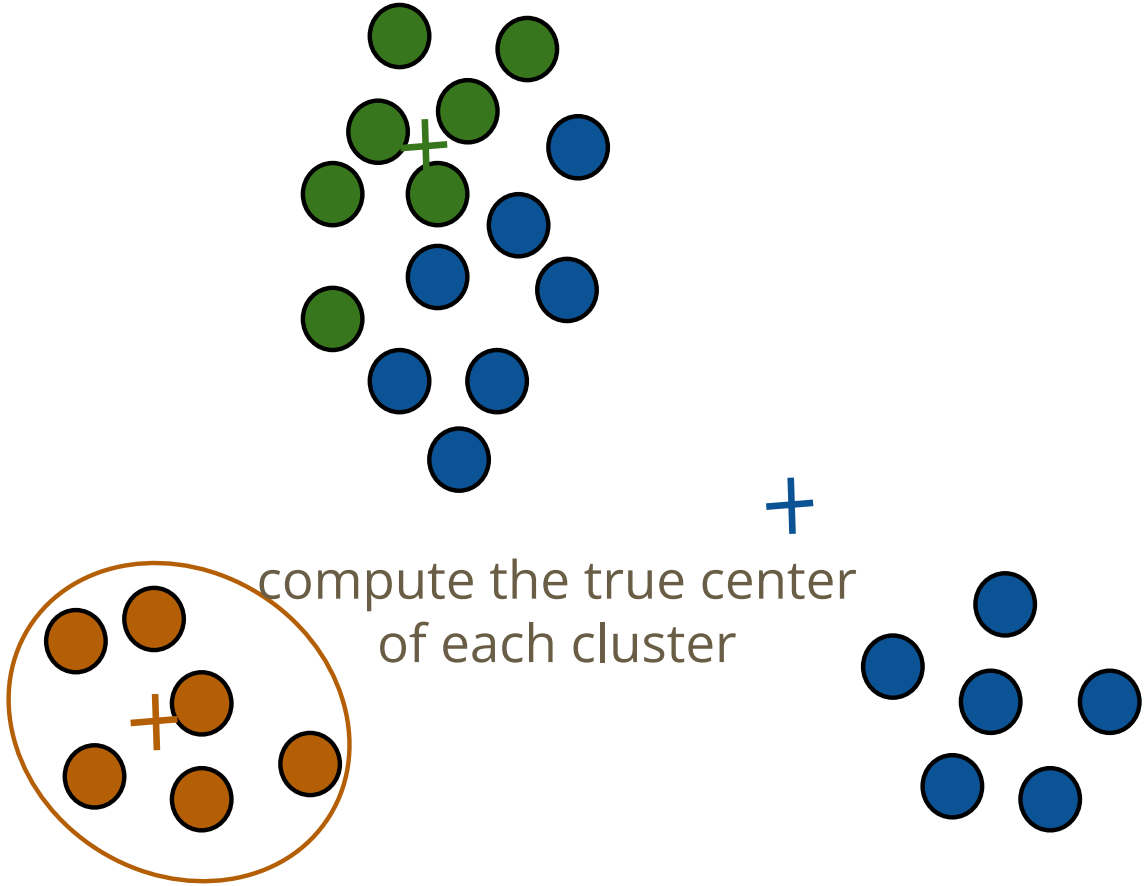

compute the true center
of each cluster


compute the true center
of each cluster

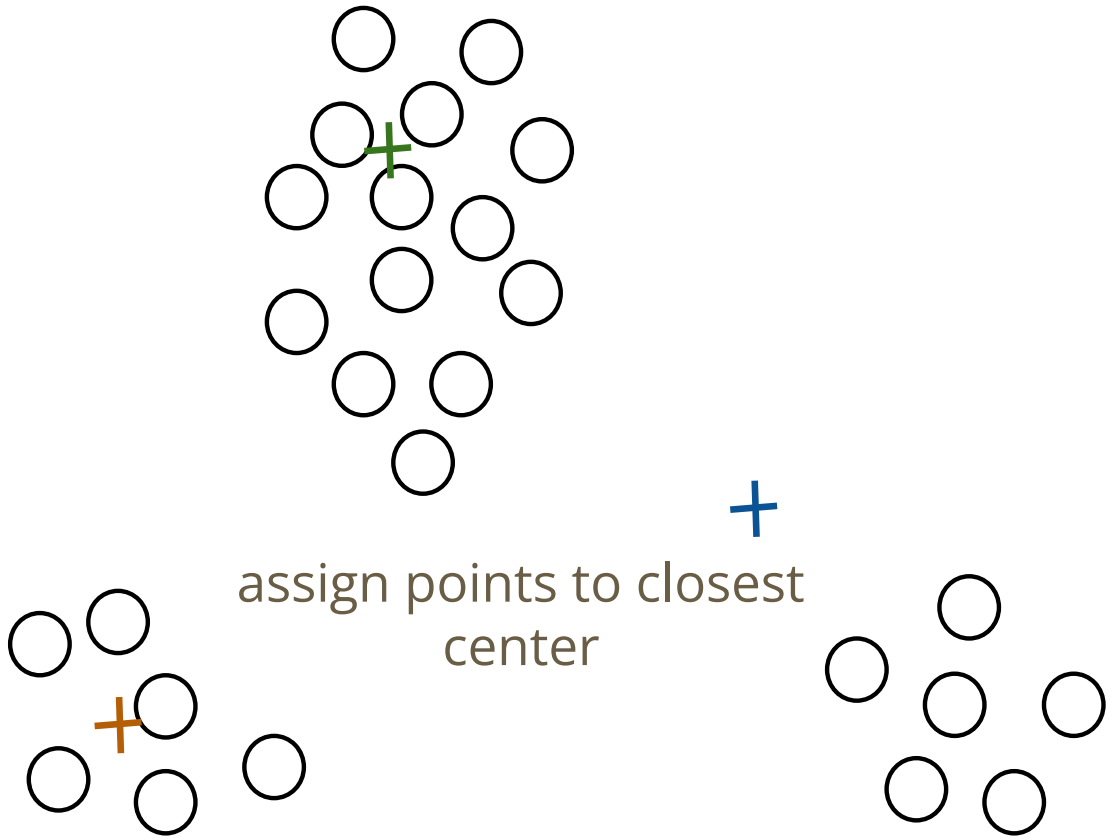

+
compute the true center
of each cluster

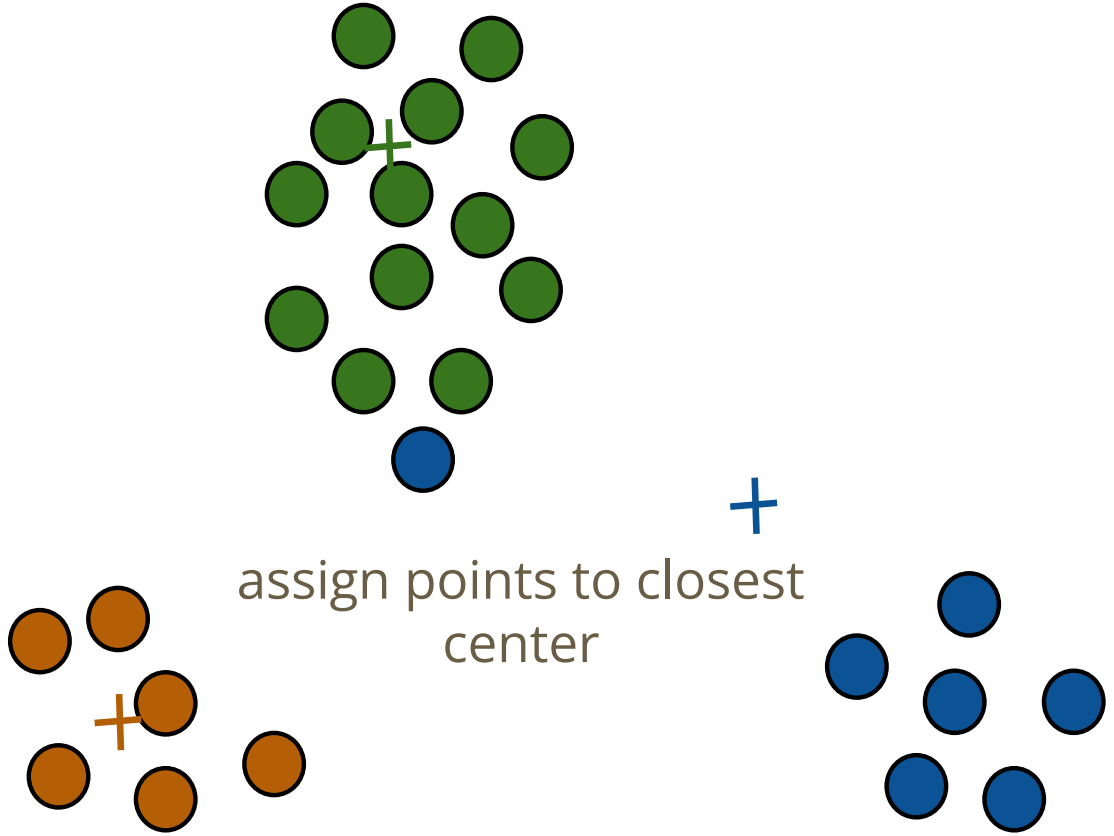

compute the true center
of each cluster

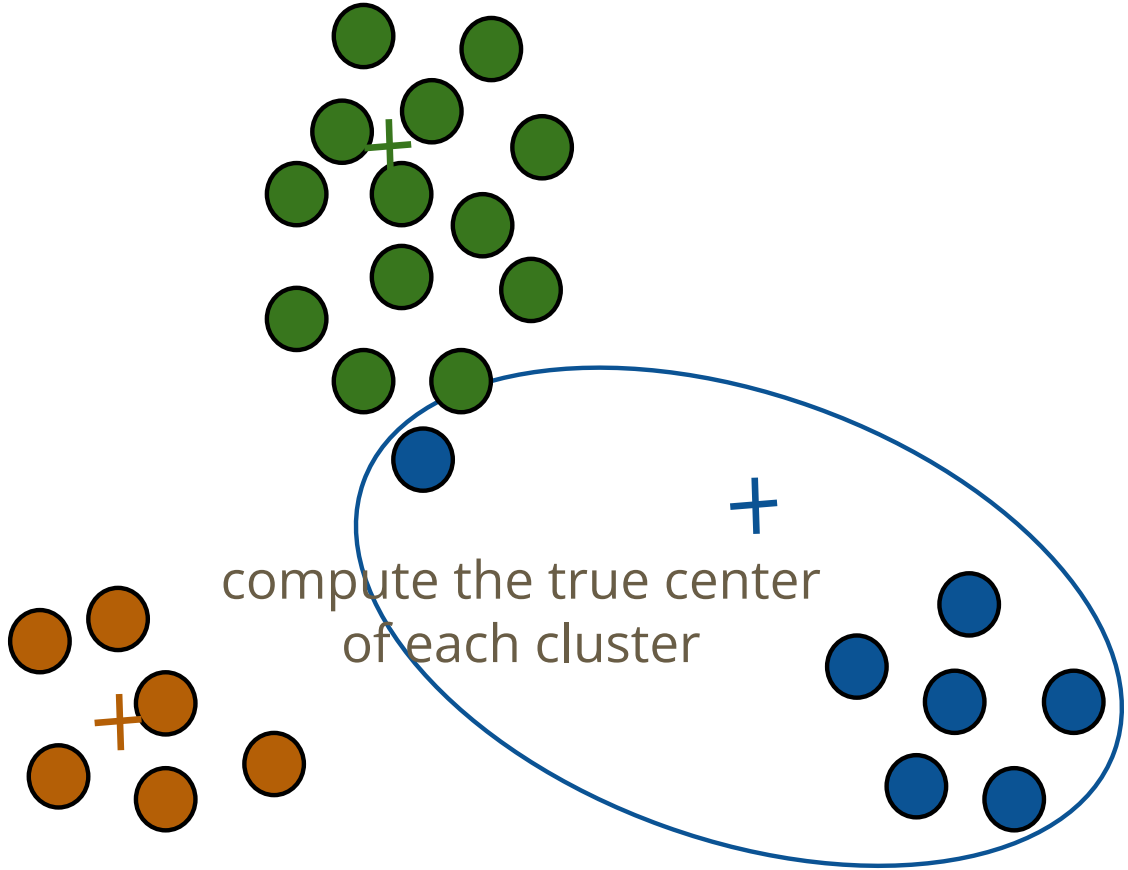

compute the true center
of each cluster

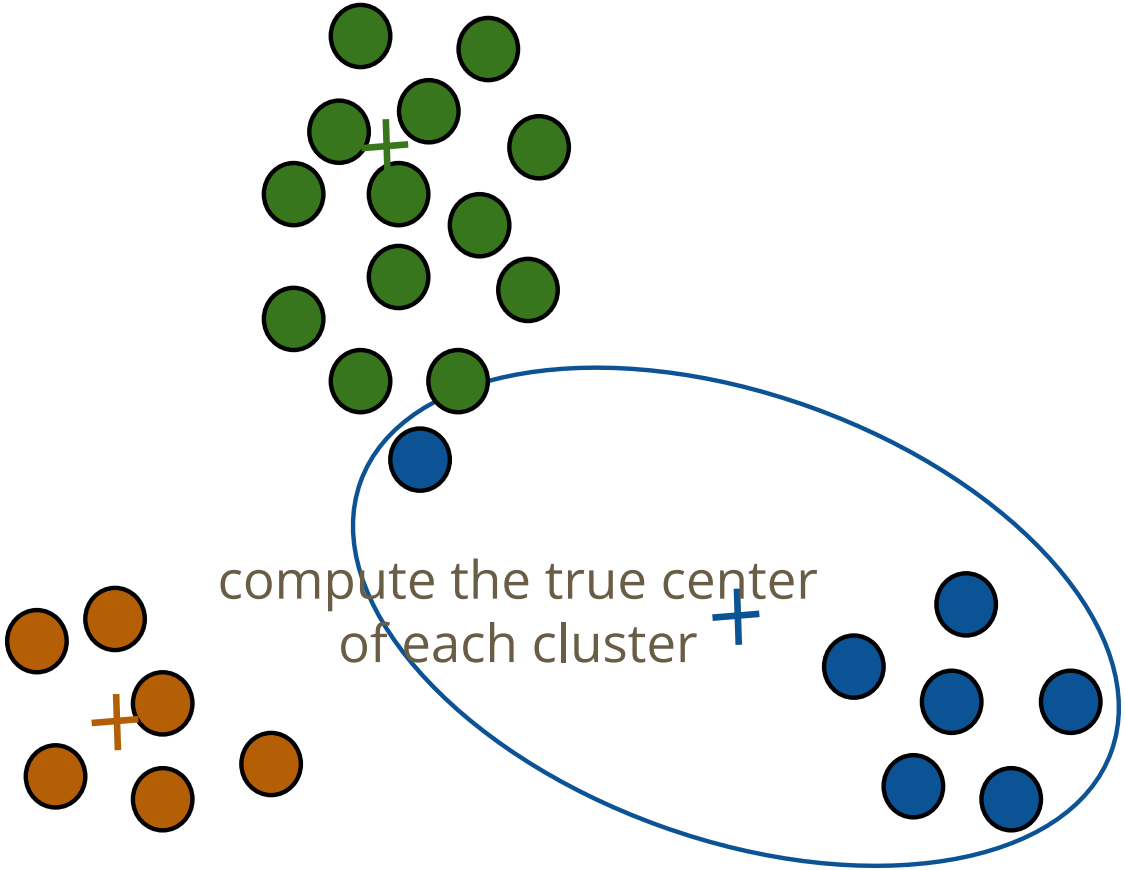

compute the true center
of each cluster

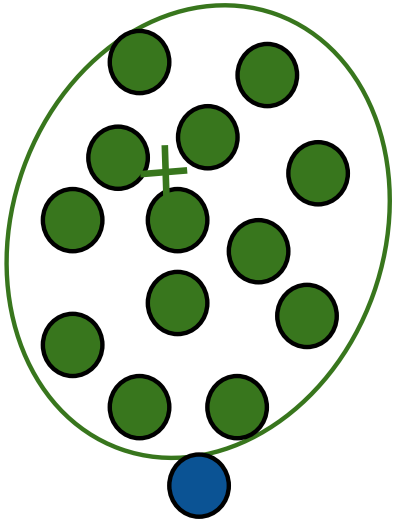
* unassign Everything



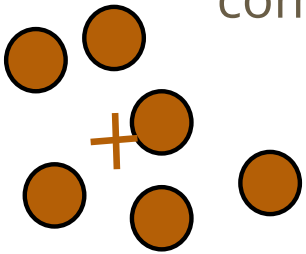

assign points to closest
center

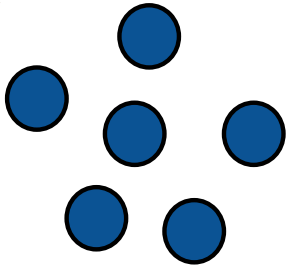

compute the true center
of each cluster

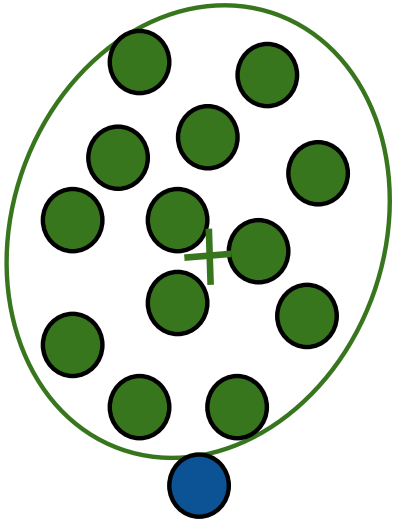

compute the true center
of each cluster



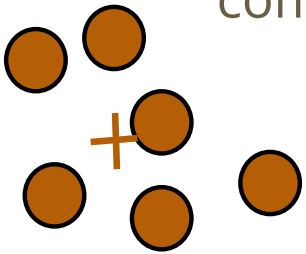
compute the true center
of each cluster

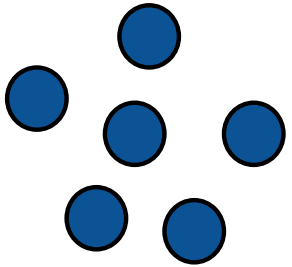


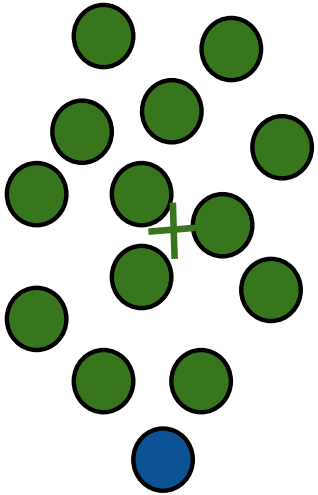




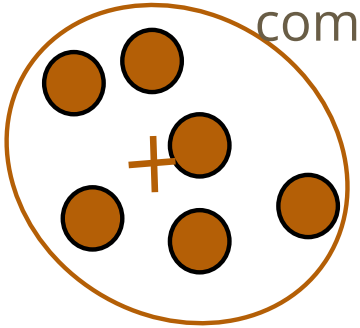
compute the true center
of each cluster

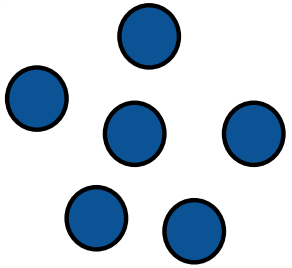


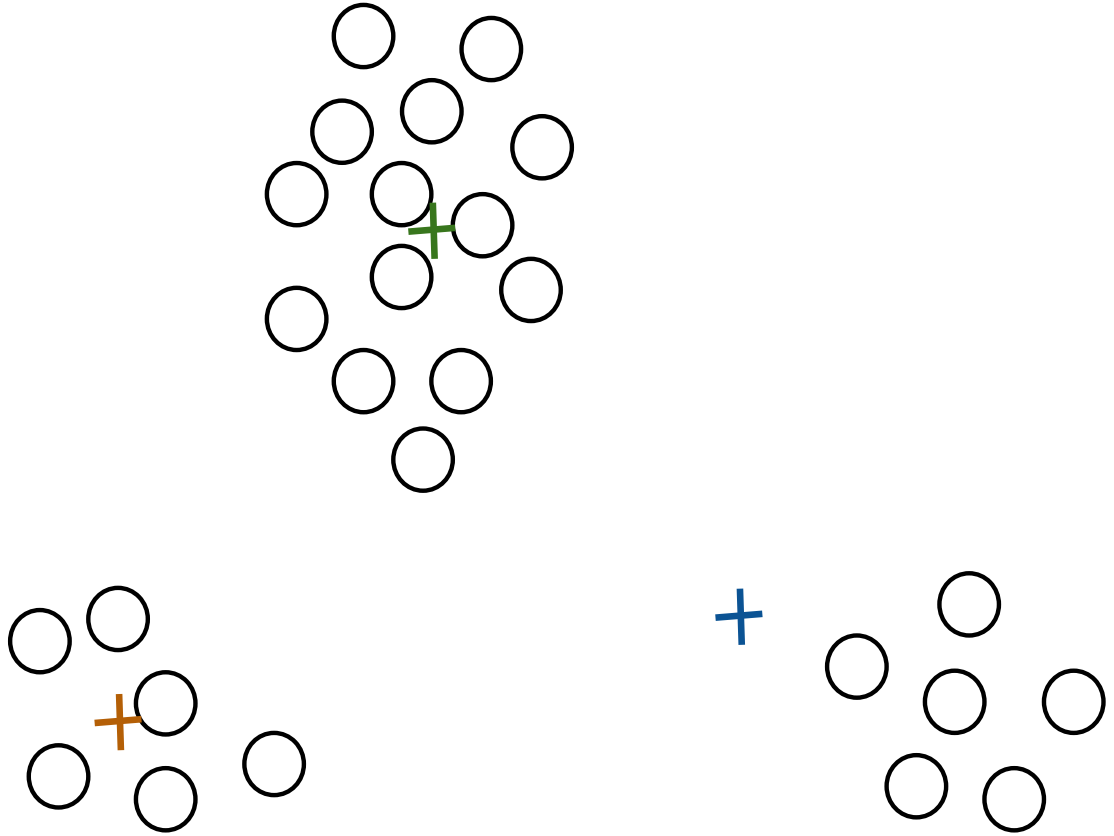


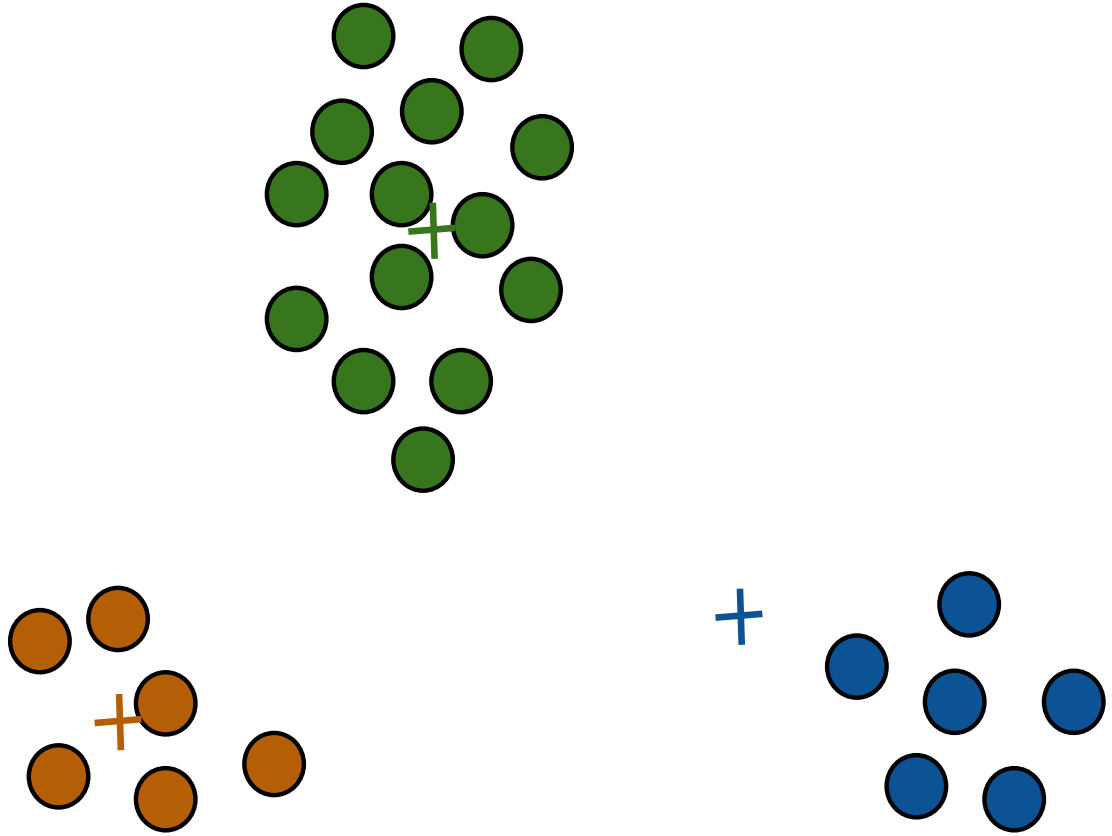


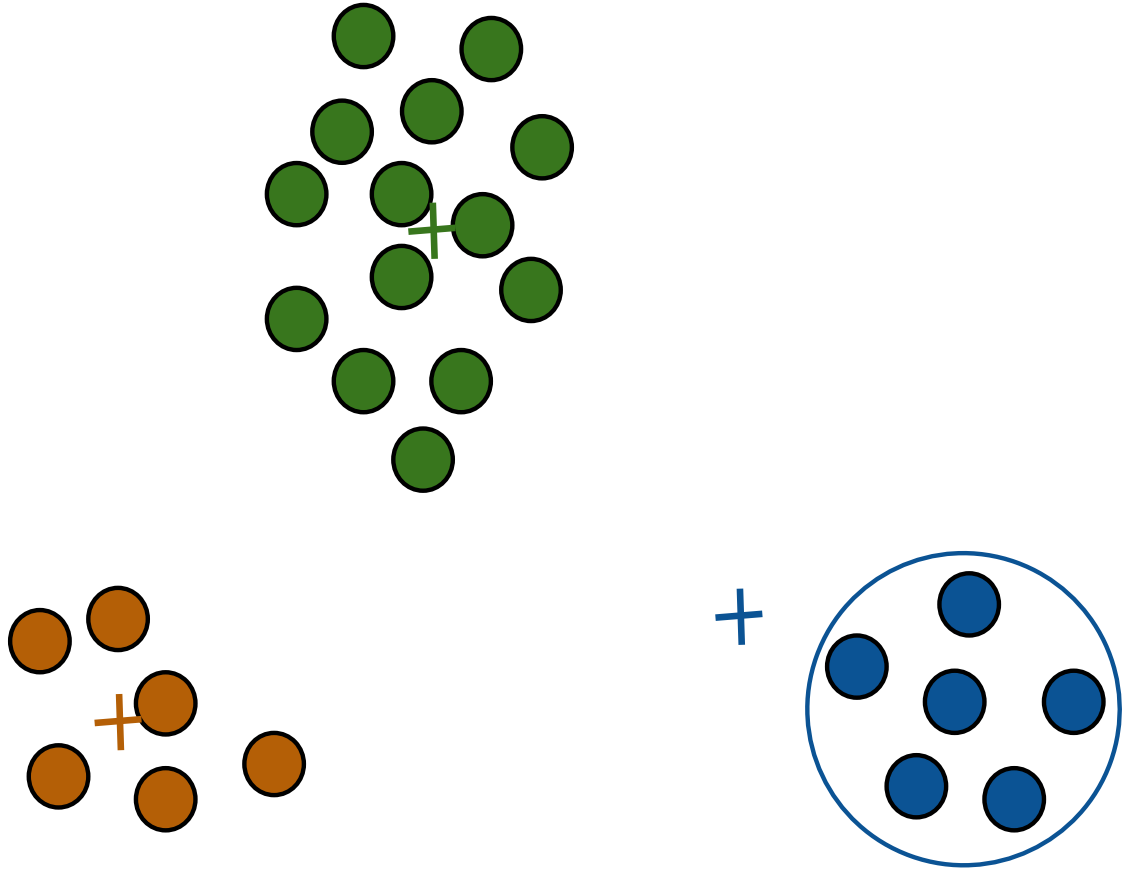
compute the true center
of each cluster

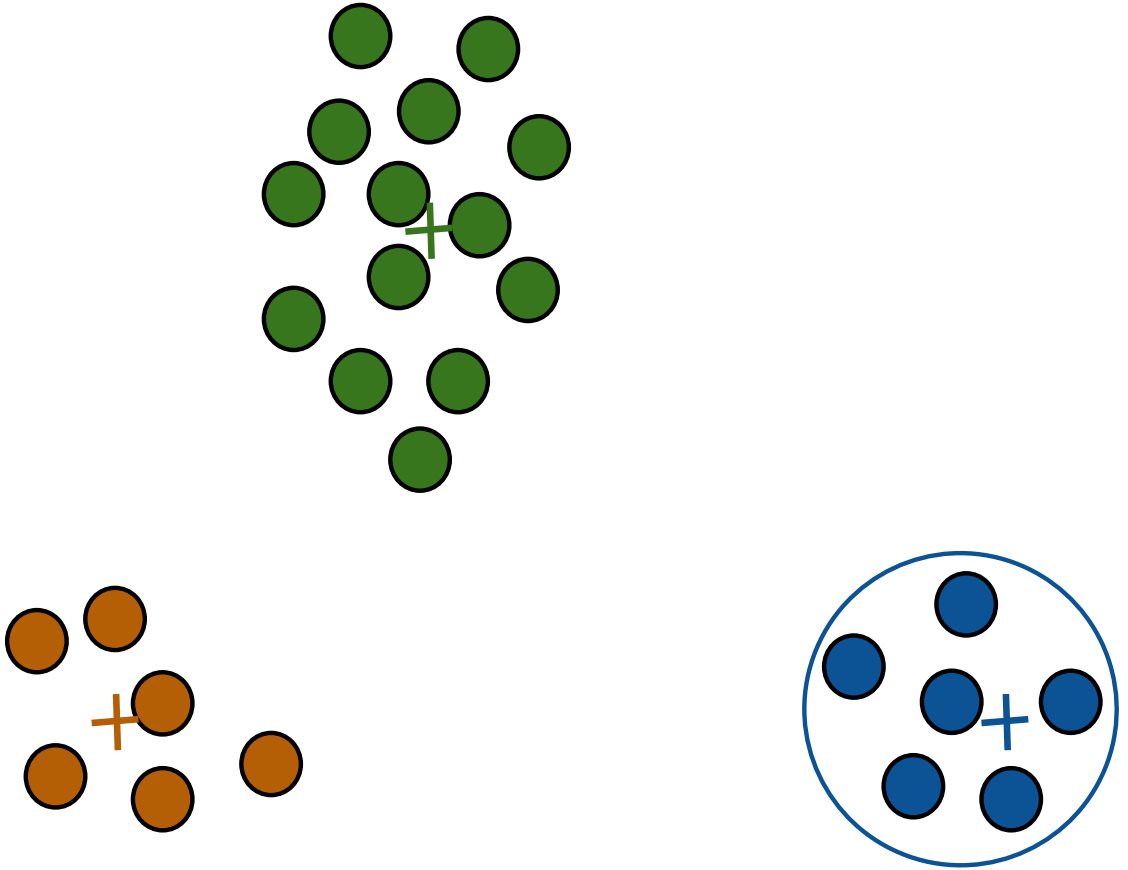


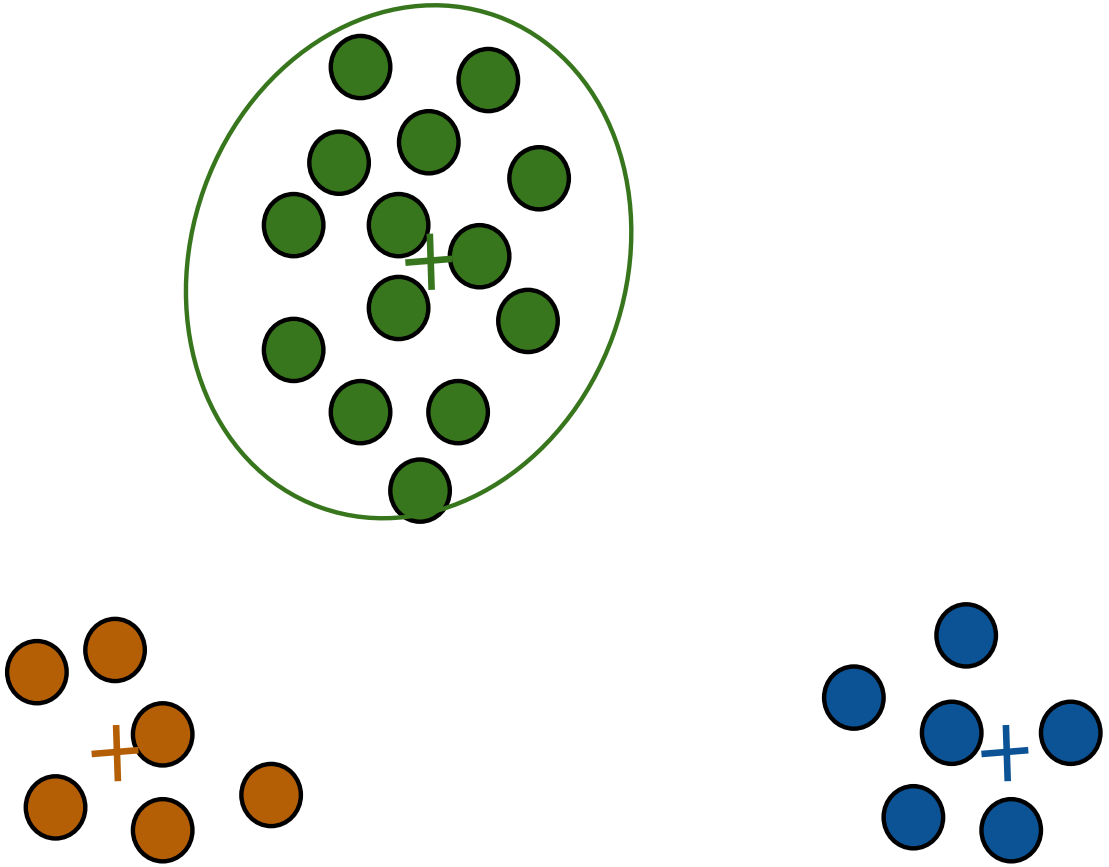


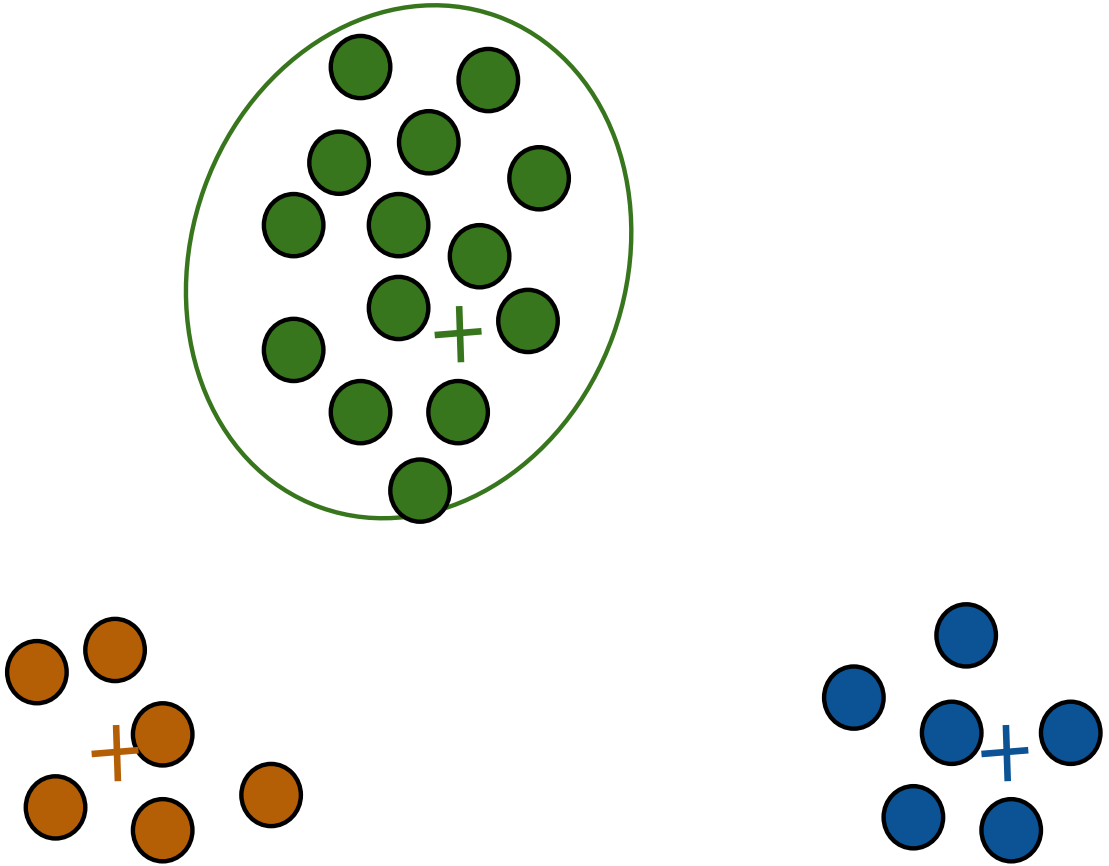


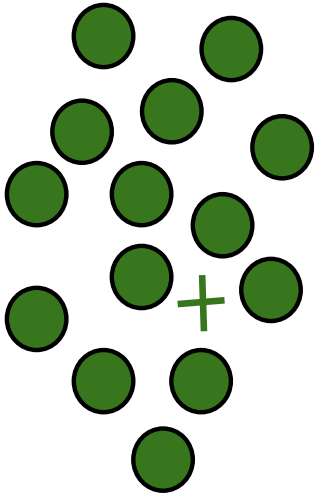


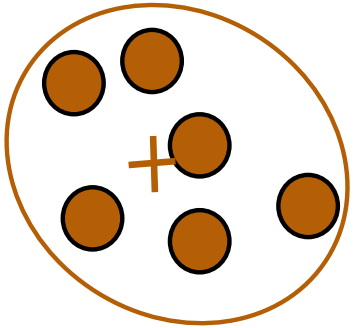


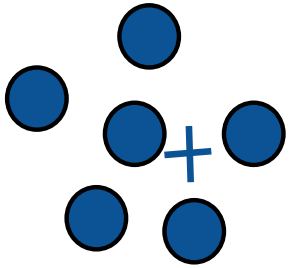


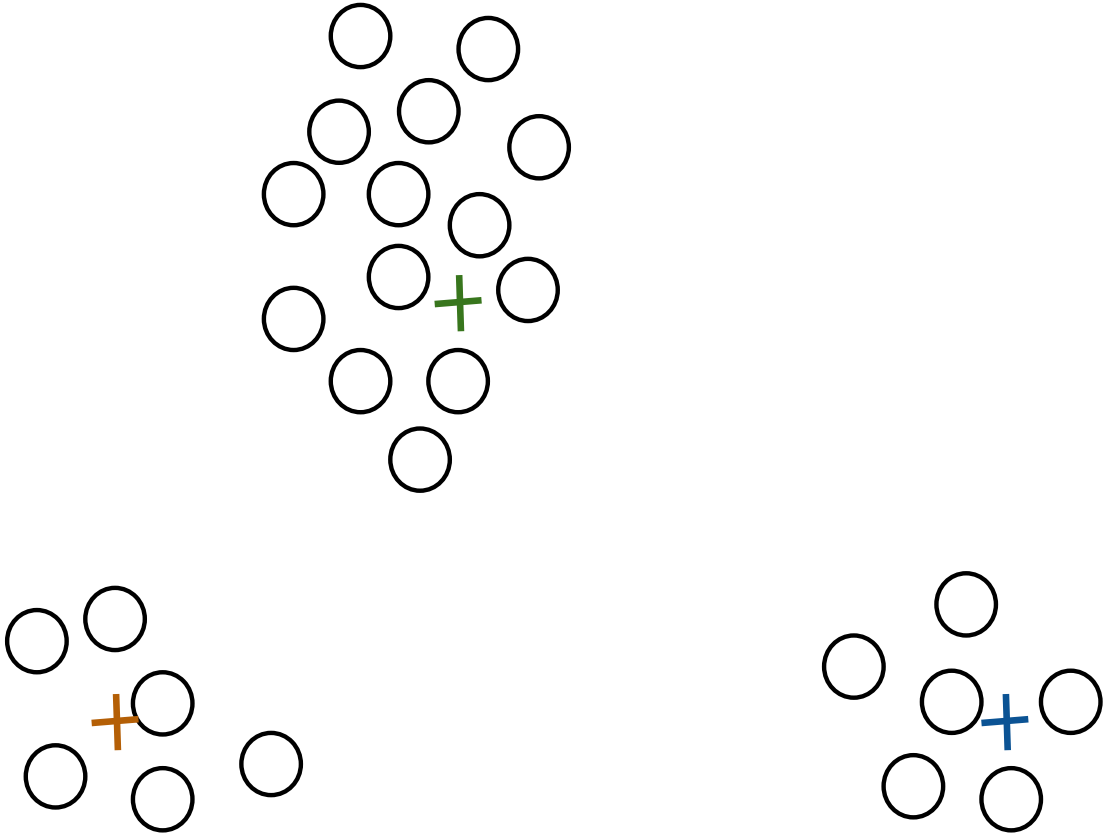


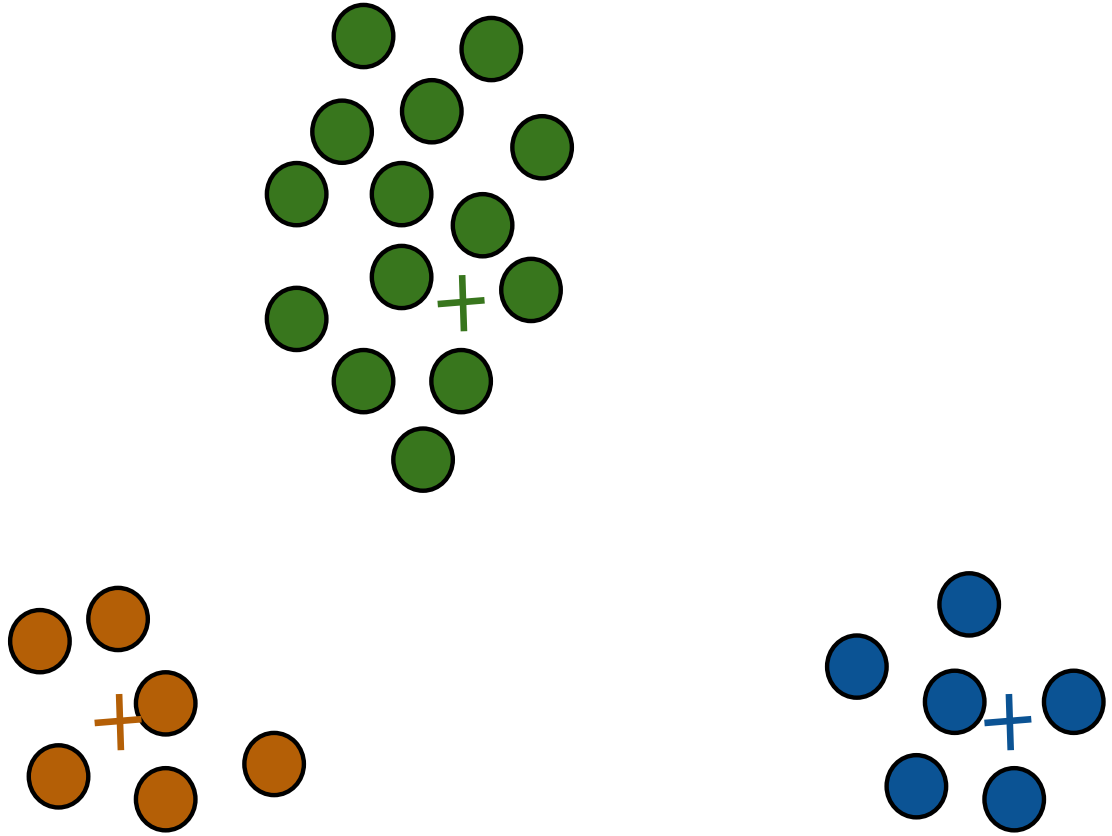


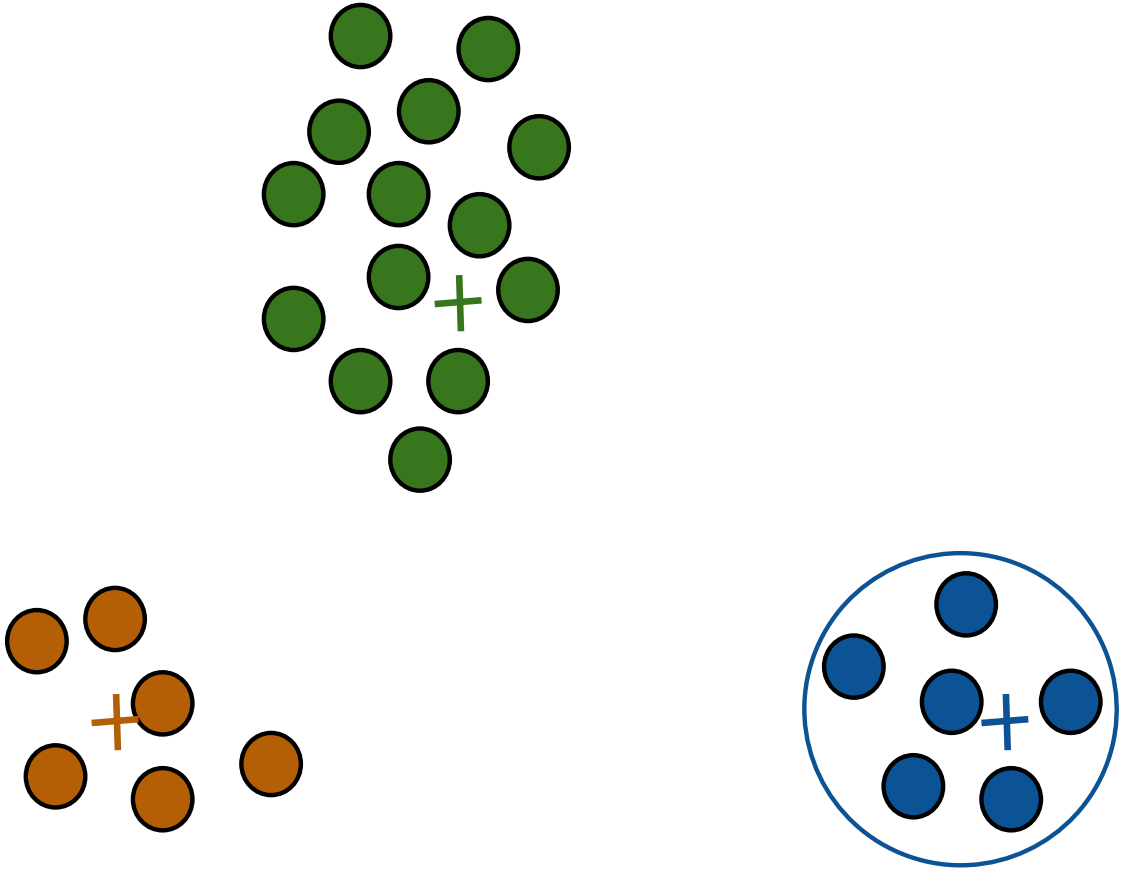


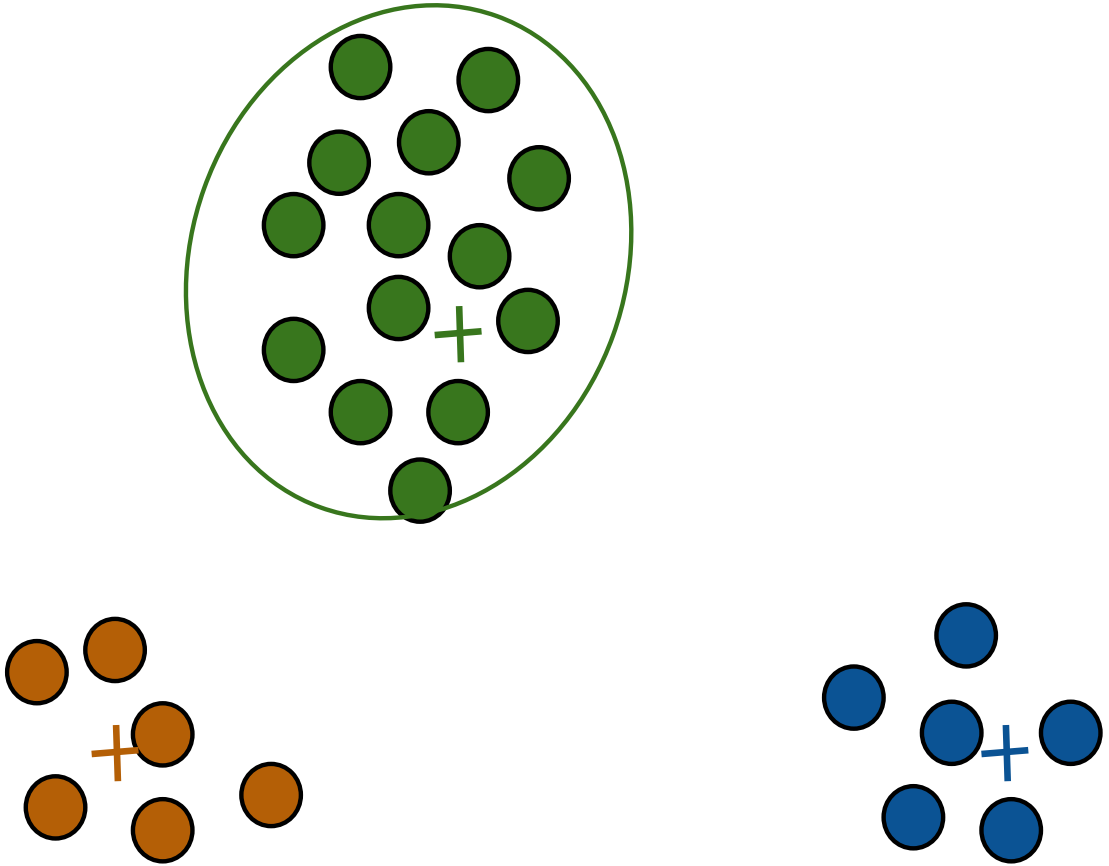


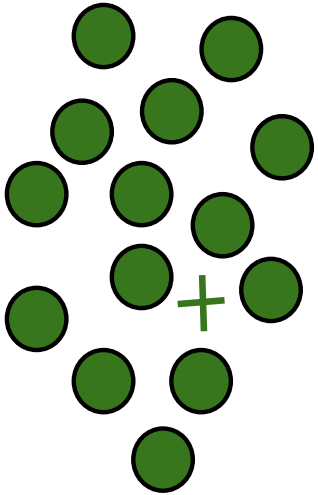


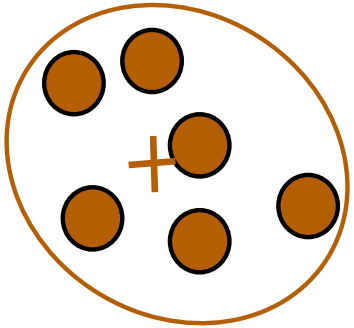


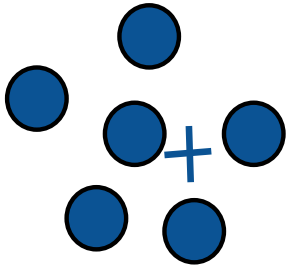


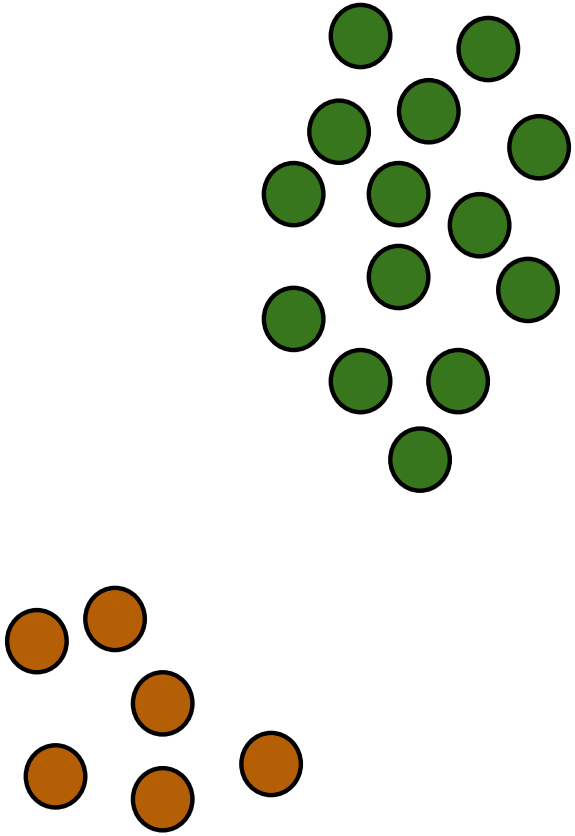








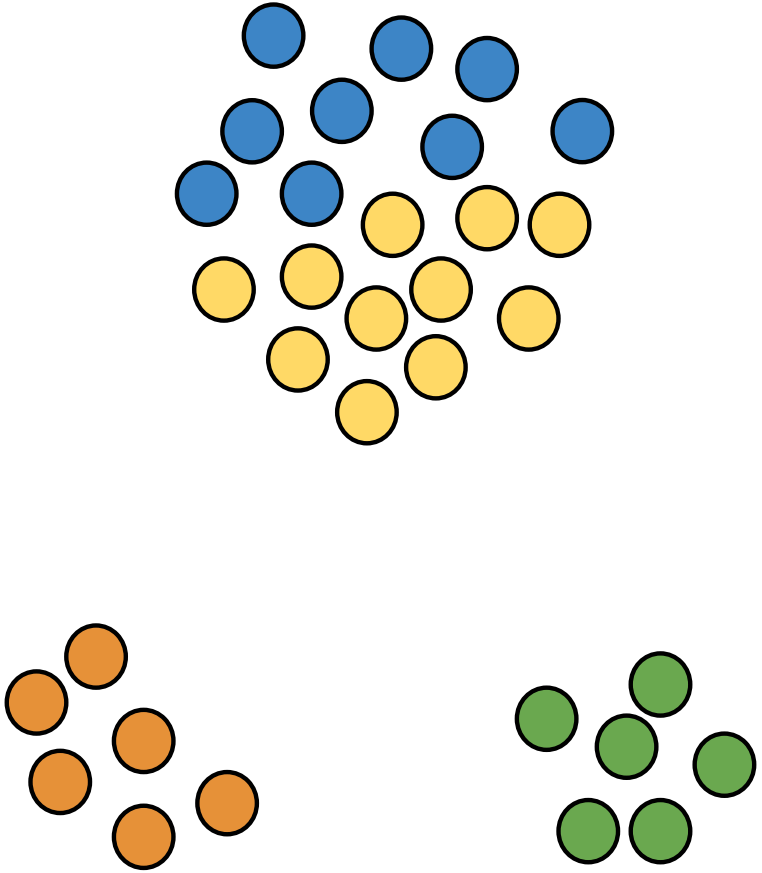




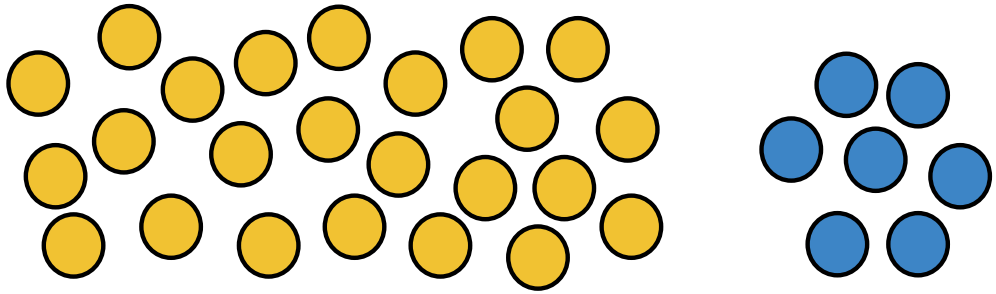
converge when the center doesn't change.

Questions

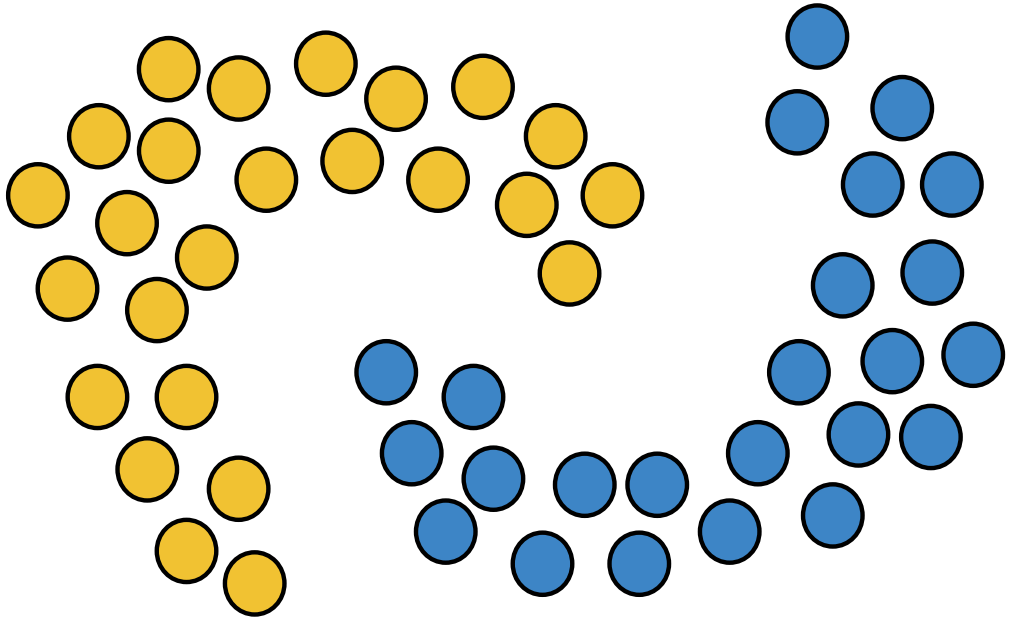
Q1: Is this a possible output of Kmeans?



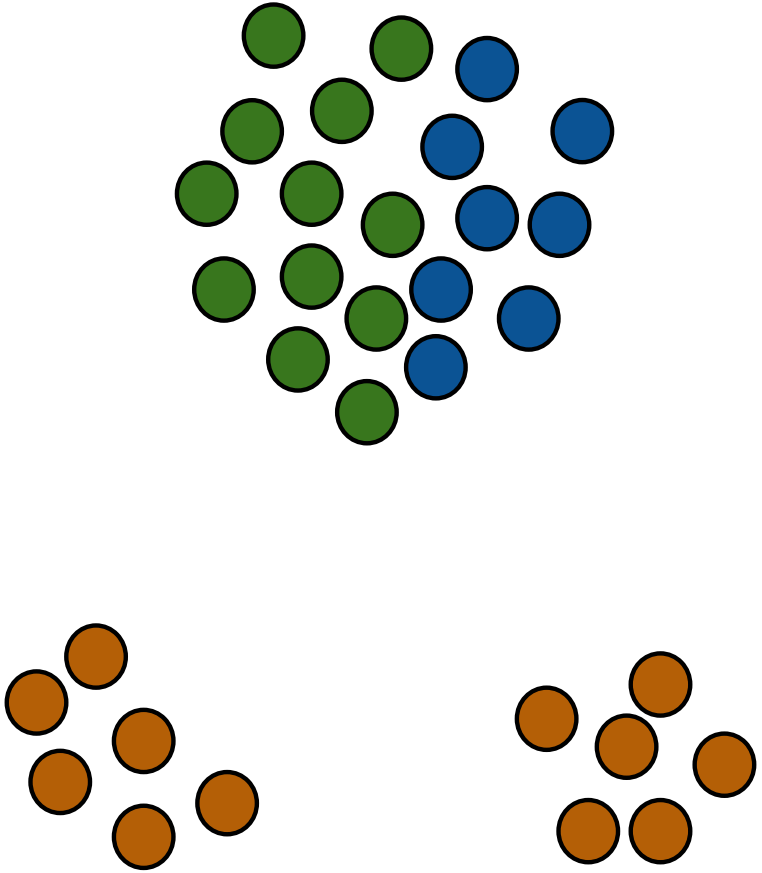
Q2: Is this a possible output of Kmeans?



Q3: Is this a possible output of Kmeans?



Q4: Is this a possible output of Kmeans?



Q5: Is this a possible output of Kmeans?