Kmeans++

Boston University CS 506 - Lance Galletti

K-means - Lloyd's Algorithm

Q1: Will this algorithm always converge?

Proof (by contradiction): Suppose it does not converge. Then, either:

1. The minimum of the cost function is only reached in the limit (i.e. after an infinite number of iterations).
   Impossible because we are iterating over a finite set of partitions

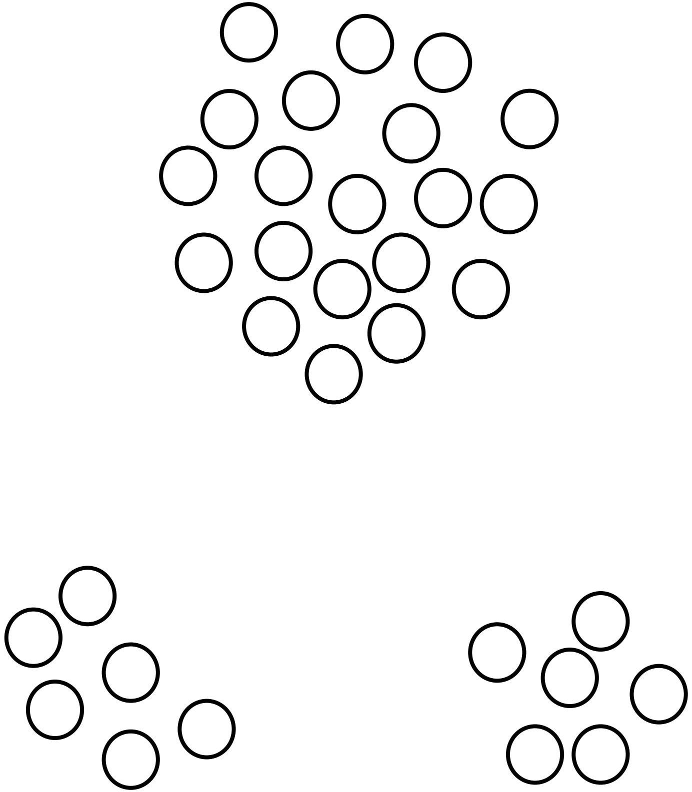
1. The algorithm gets stuck in a cycle / loop
   Impossible since this would require having a clustering that has a lower cost than itself and we know:
   - If old ≠ new clustering then the cost has improved
   - If old = new clustering then the cost is unchanged

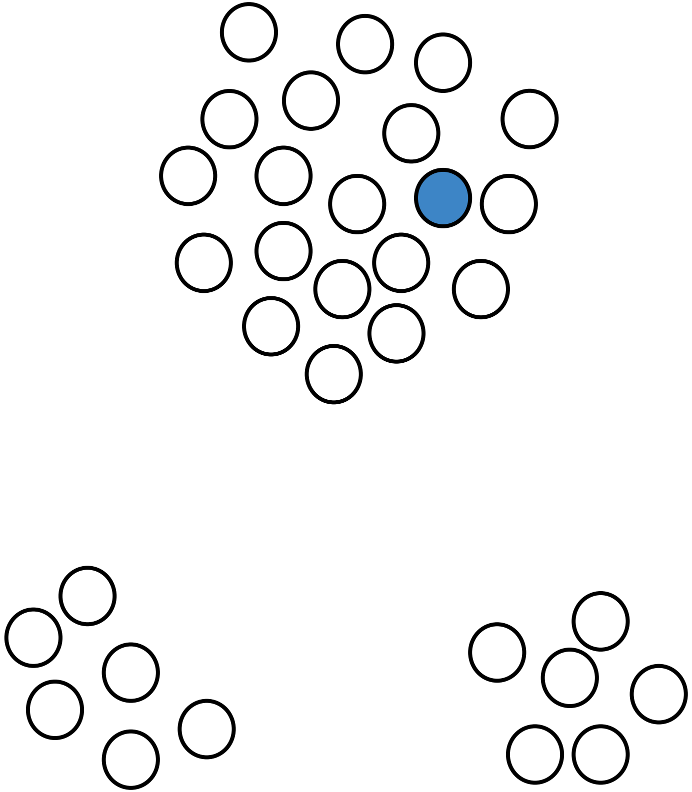
Conclusion: Lloyd's Algorithm always converges!

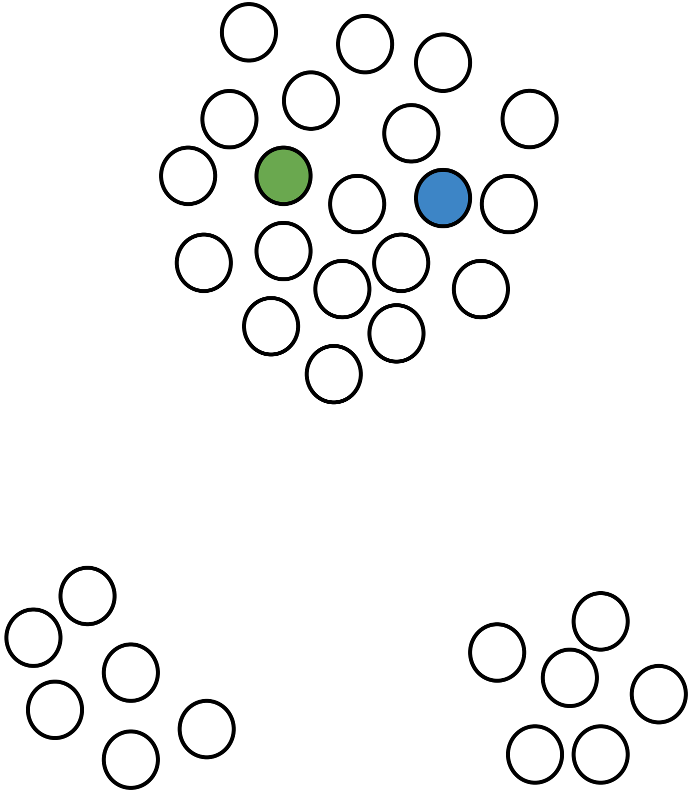
K-means - Lloyd's Algorithm

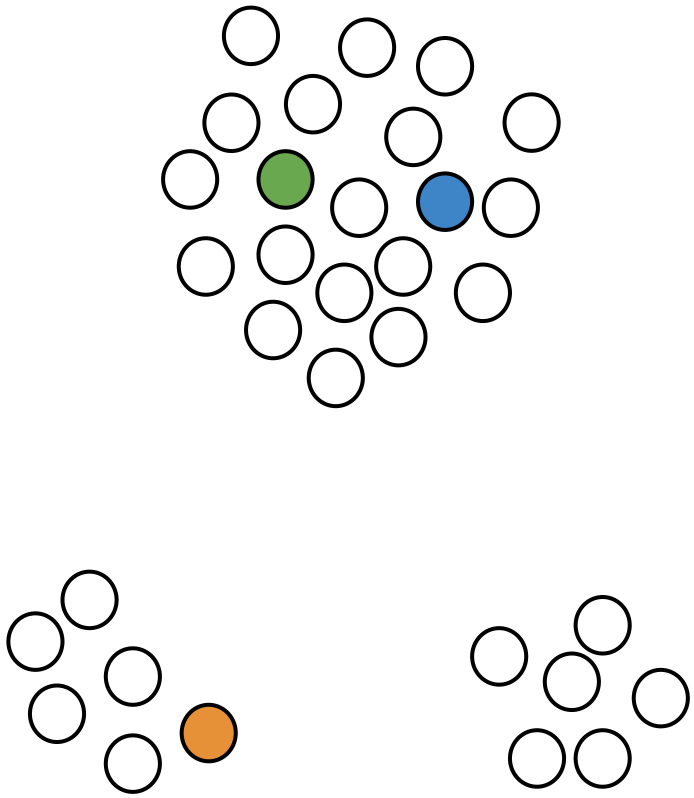
Q2: Will this always converge to the optimal solution?

No. If the centers are too closed together, they would split up a naturally occurring cluster into 2 unnatural clusters

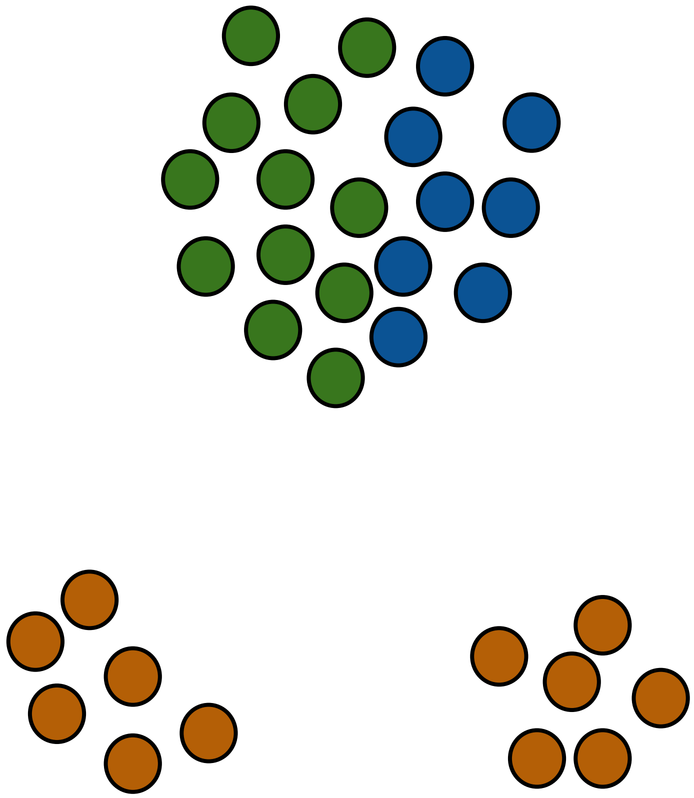




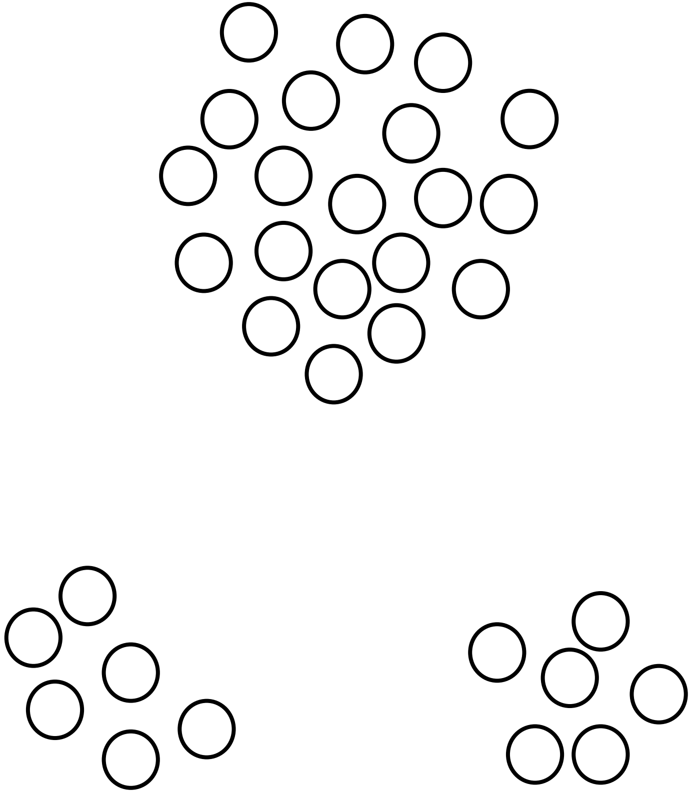


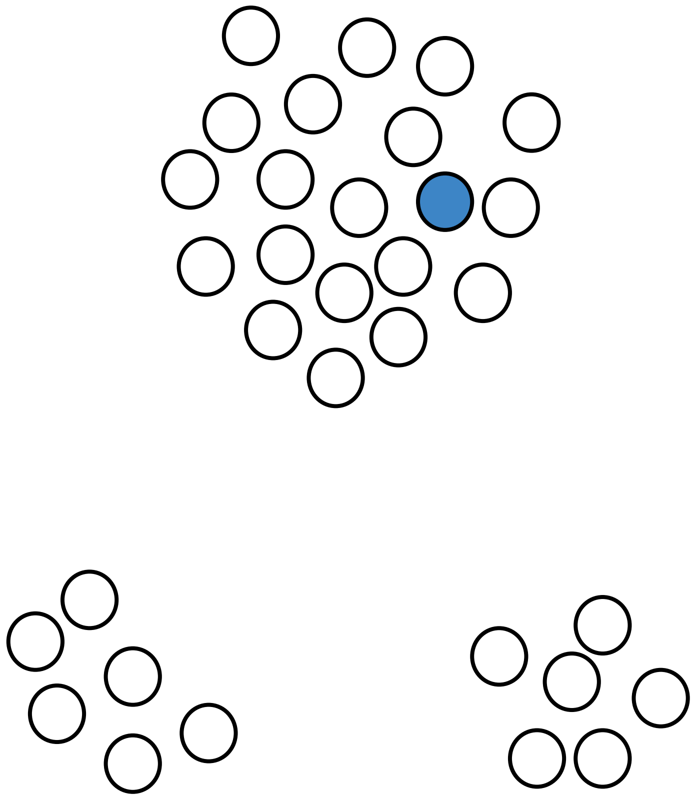


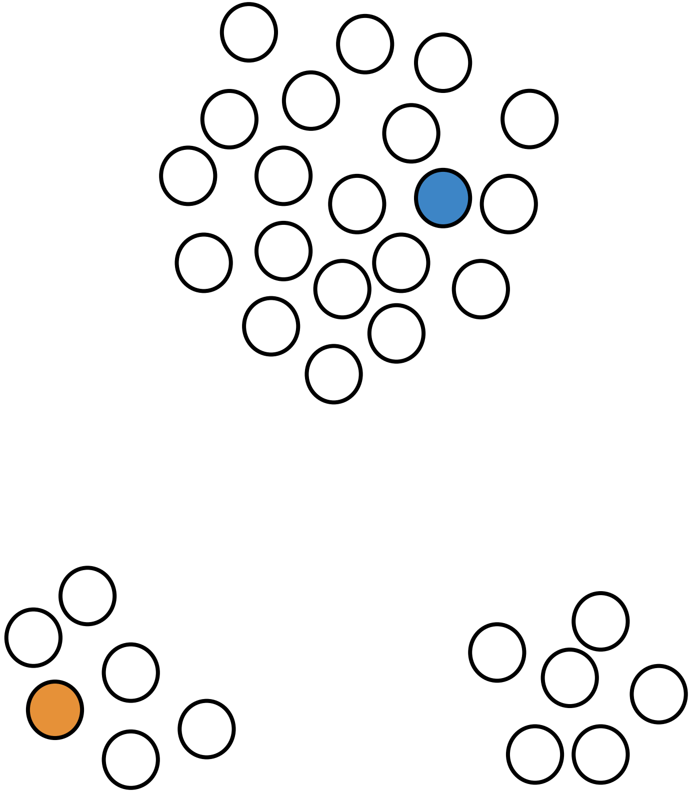
Local minima:

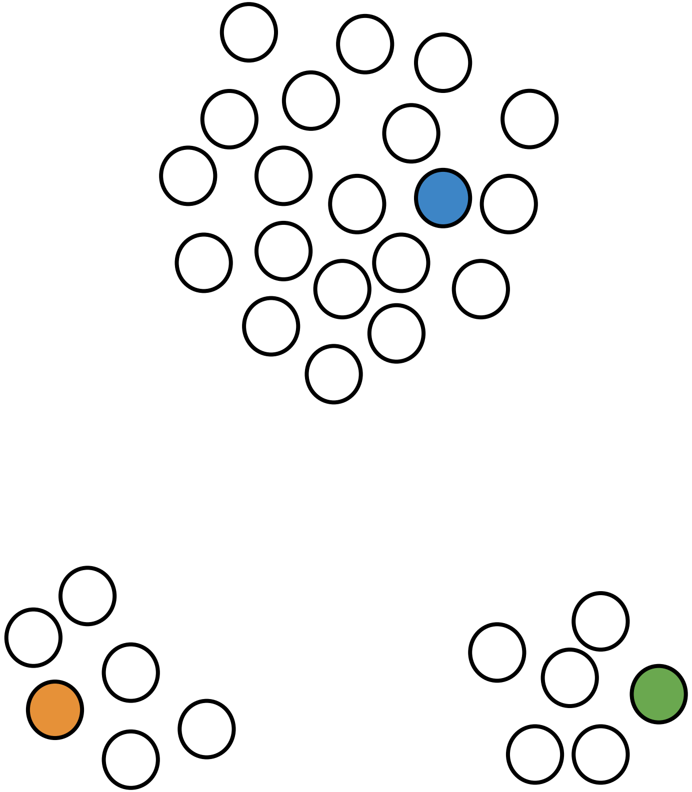


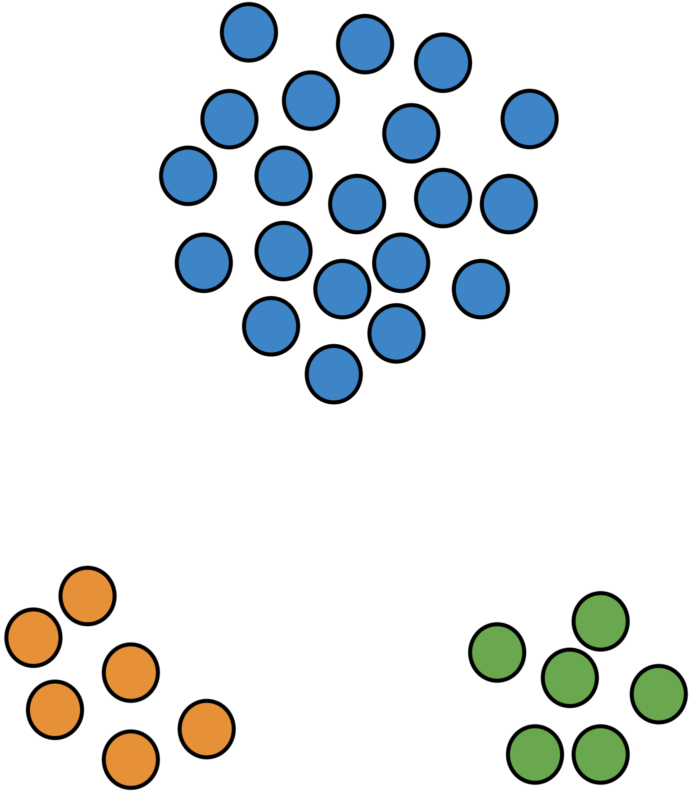
What’s the problem?





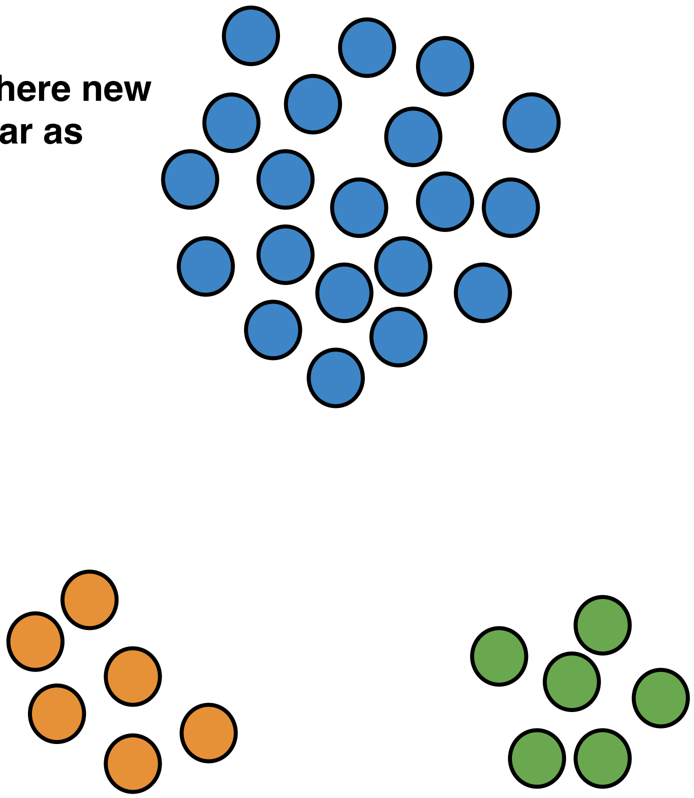






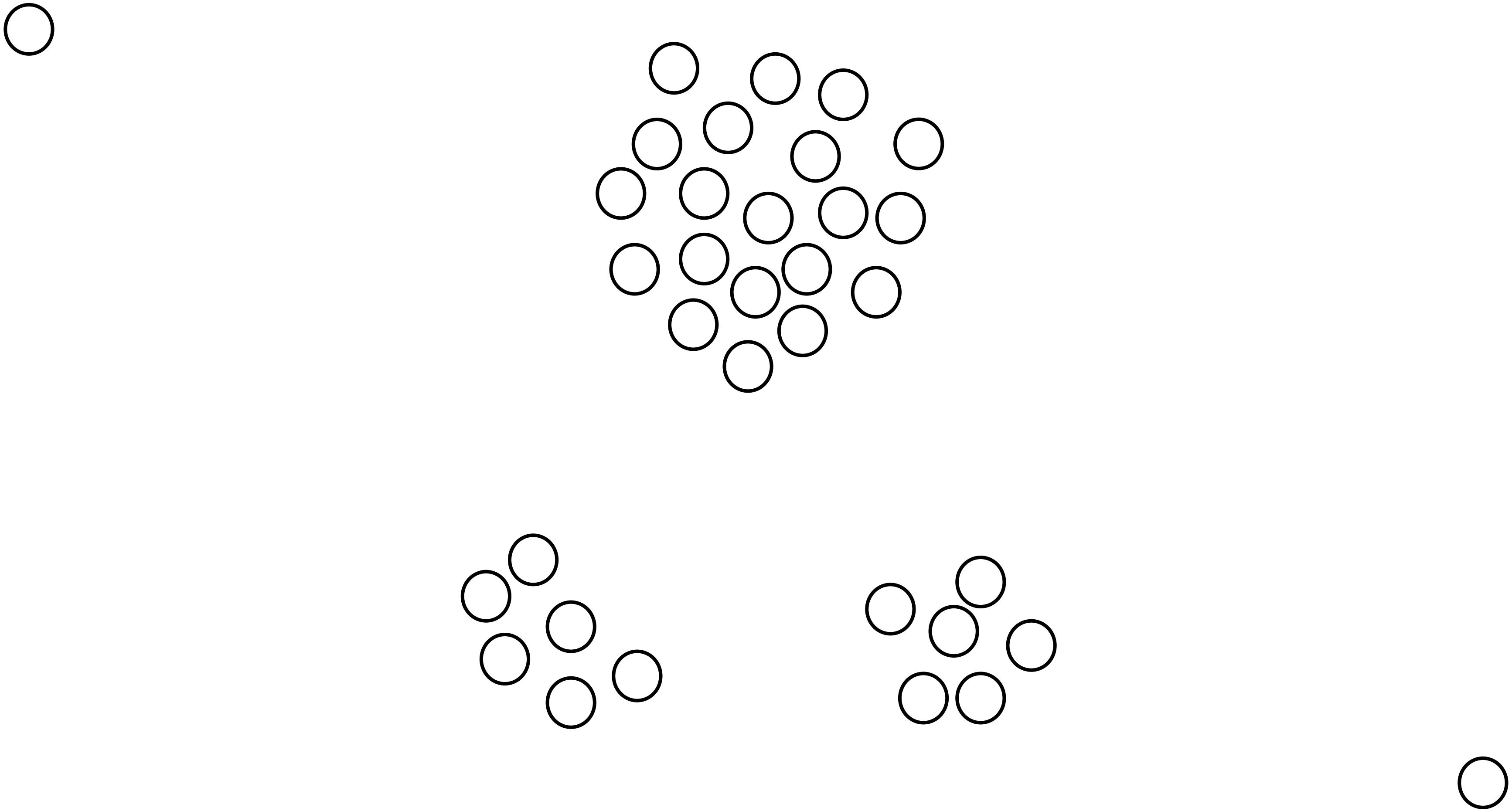
Farthest First Traversal

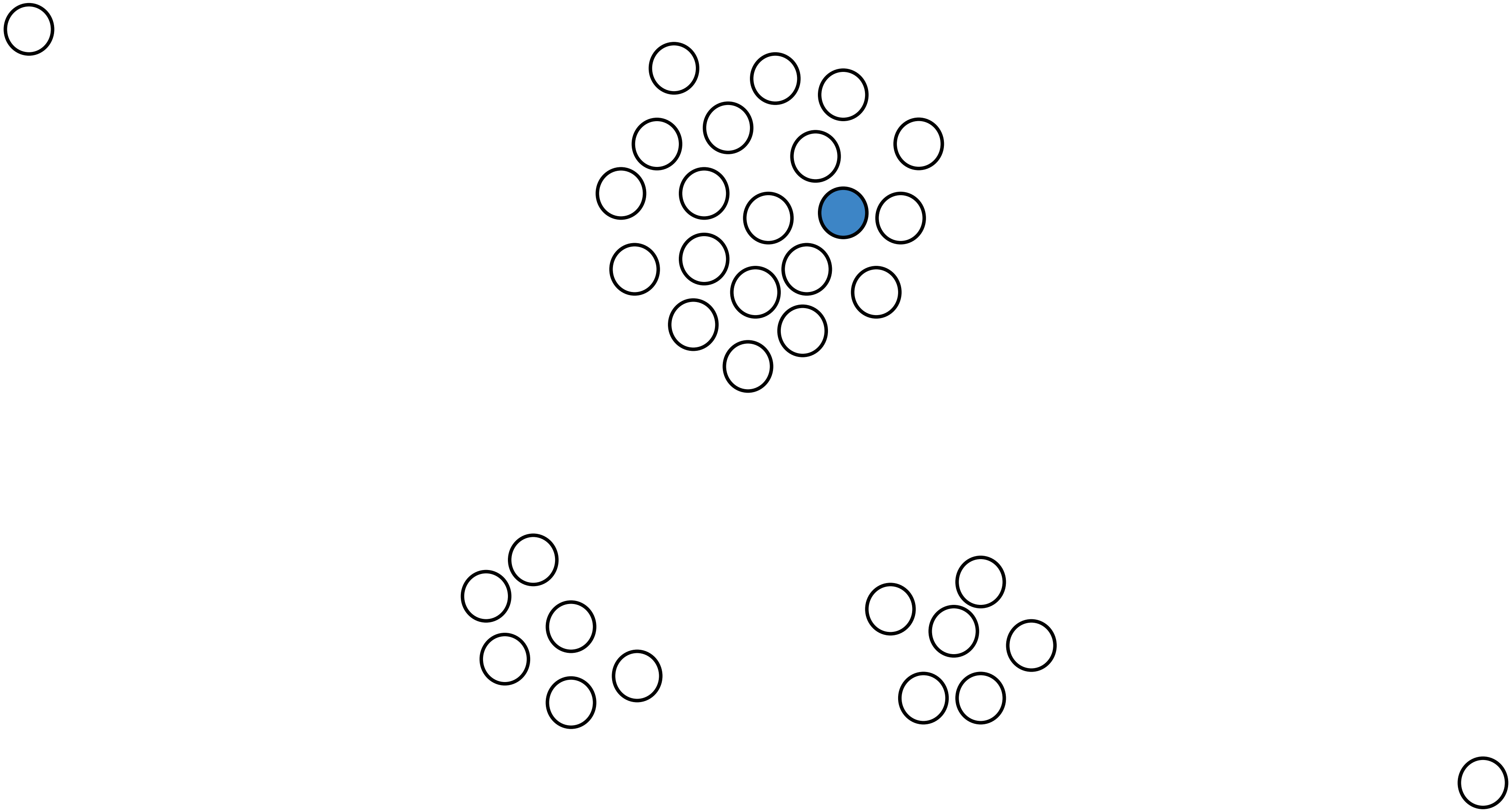
: a clustering initialization method where new cluster centers are chosen to be as far as possible from existing centers

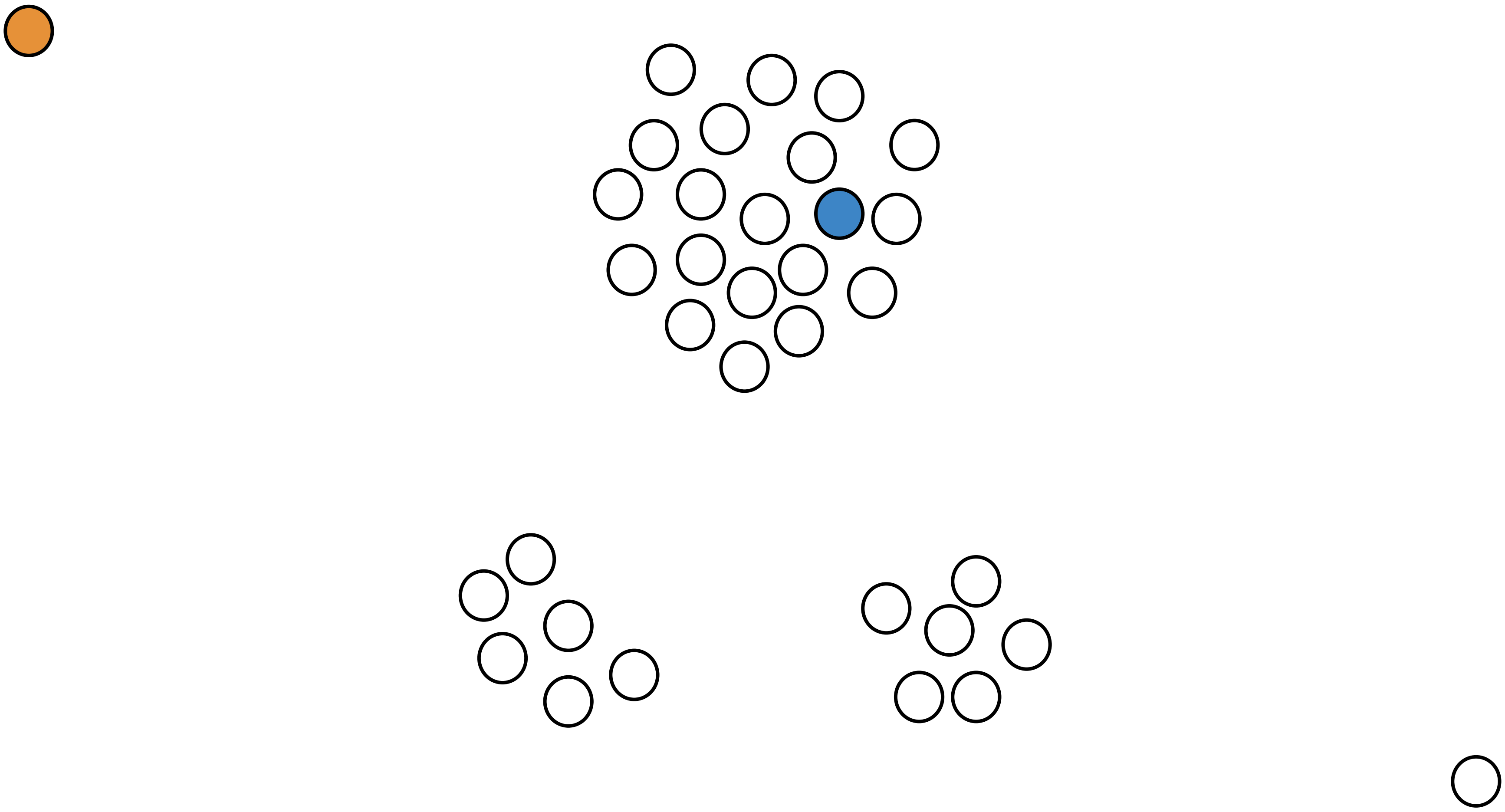


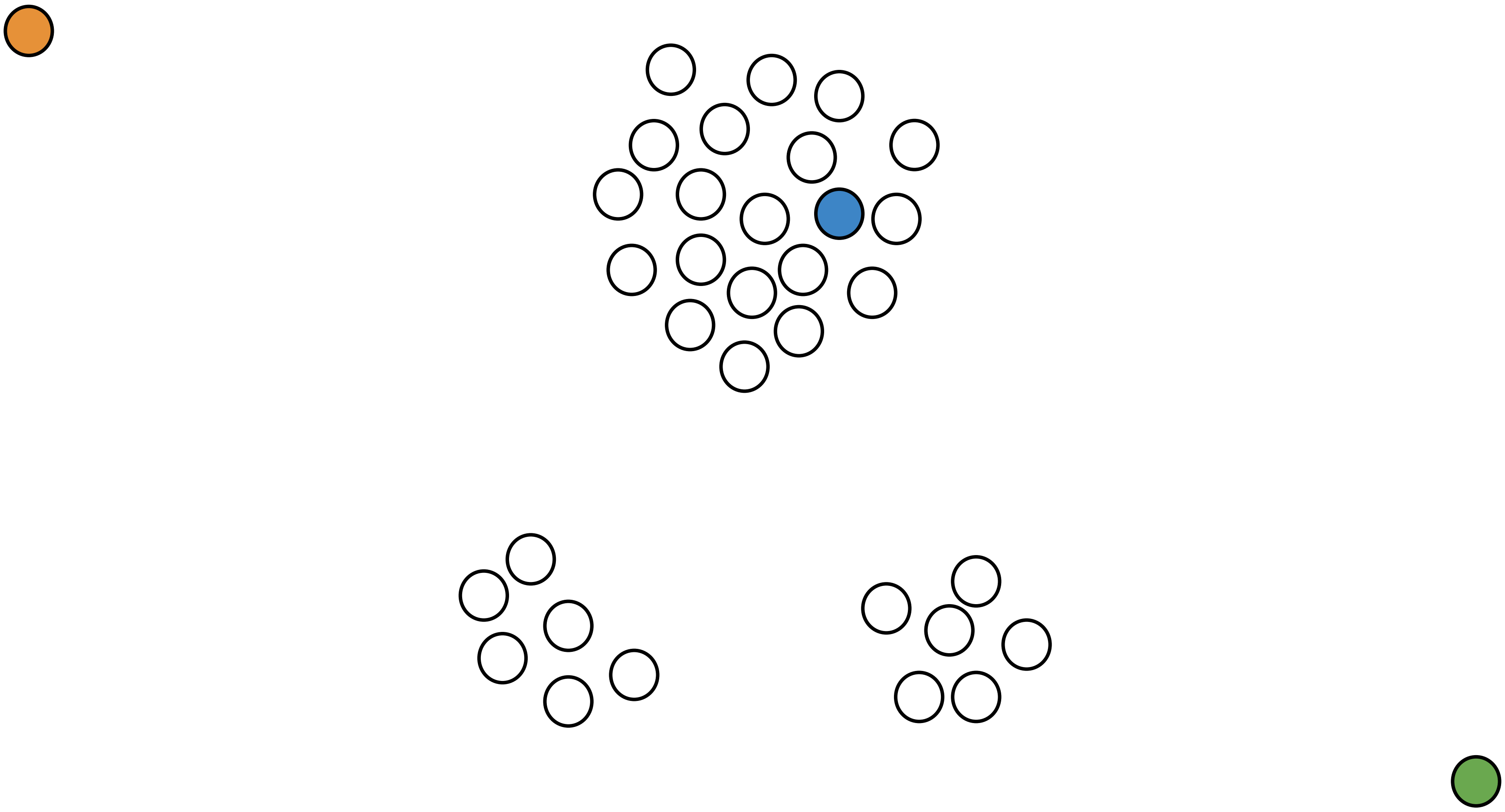
But...

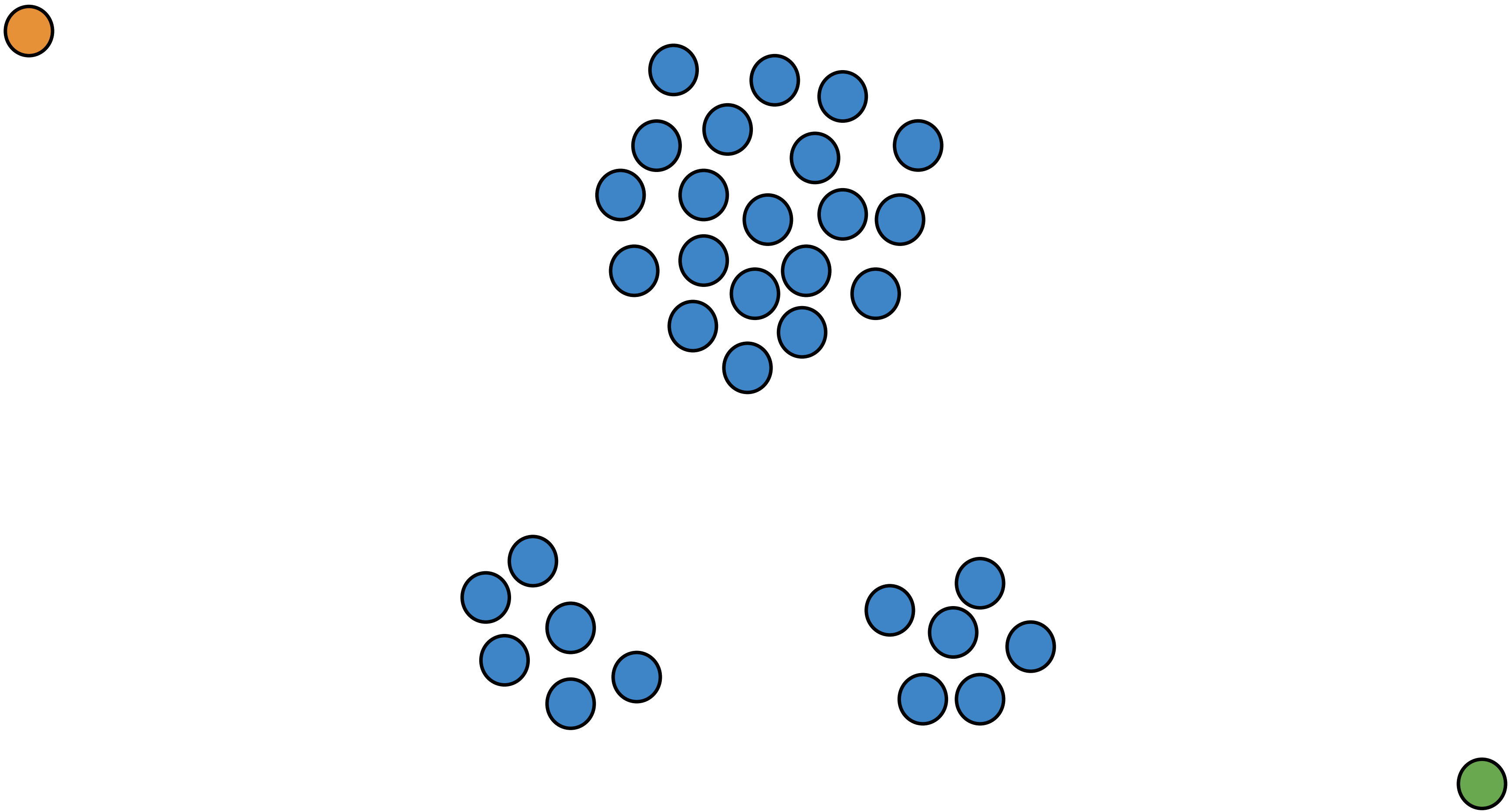
The problem is we could pick outliers as our centers



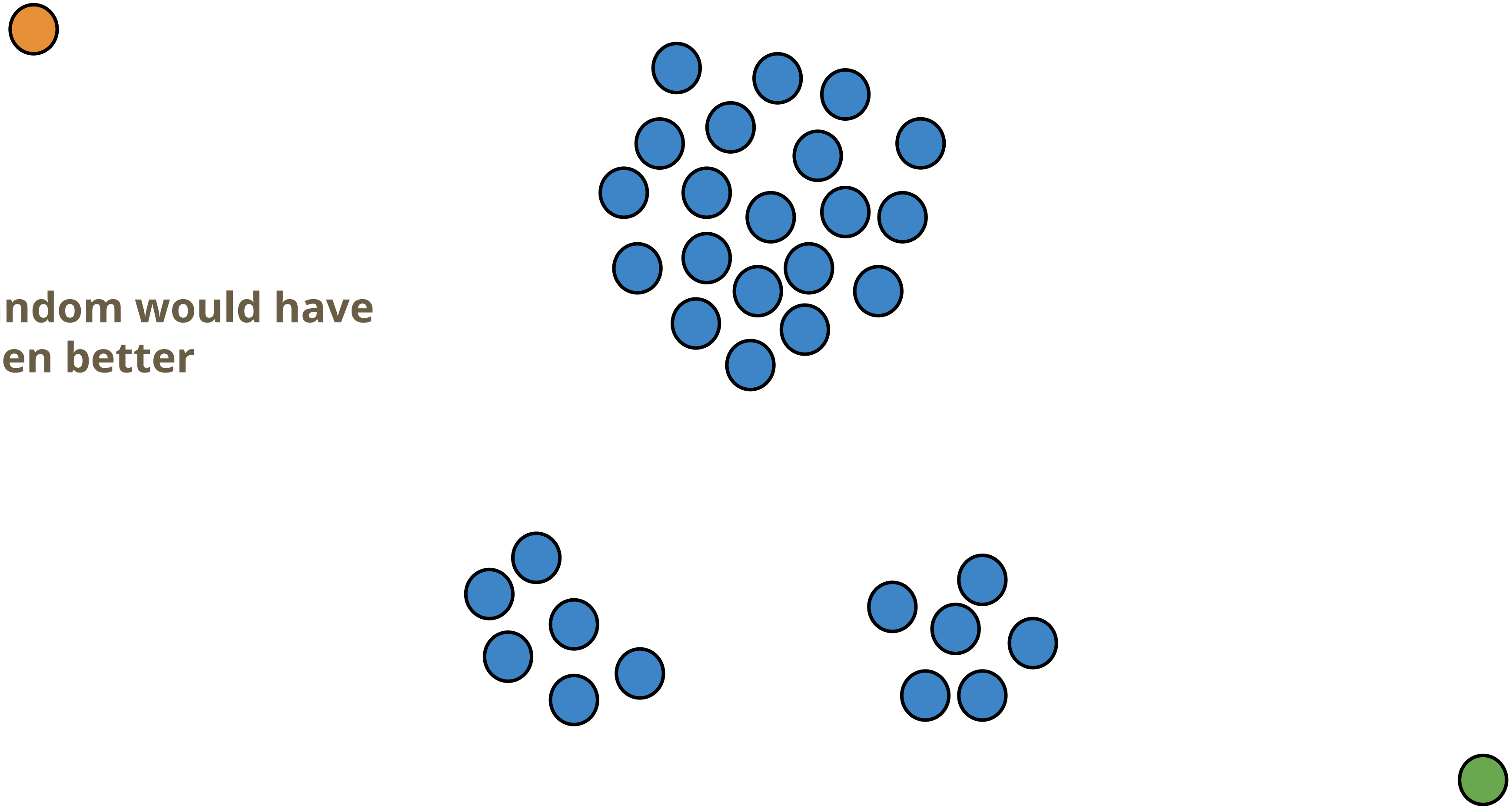








Random would have been better

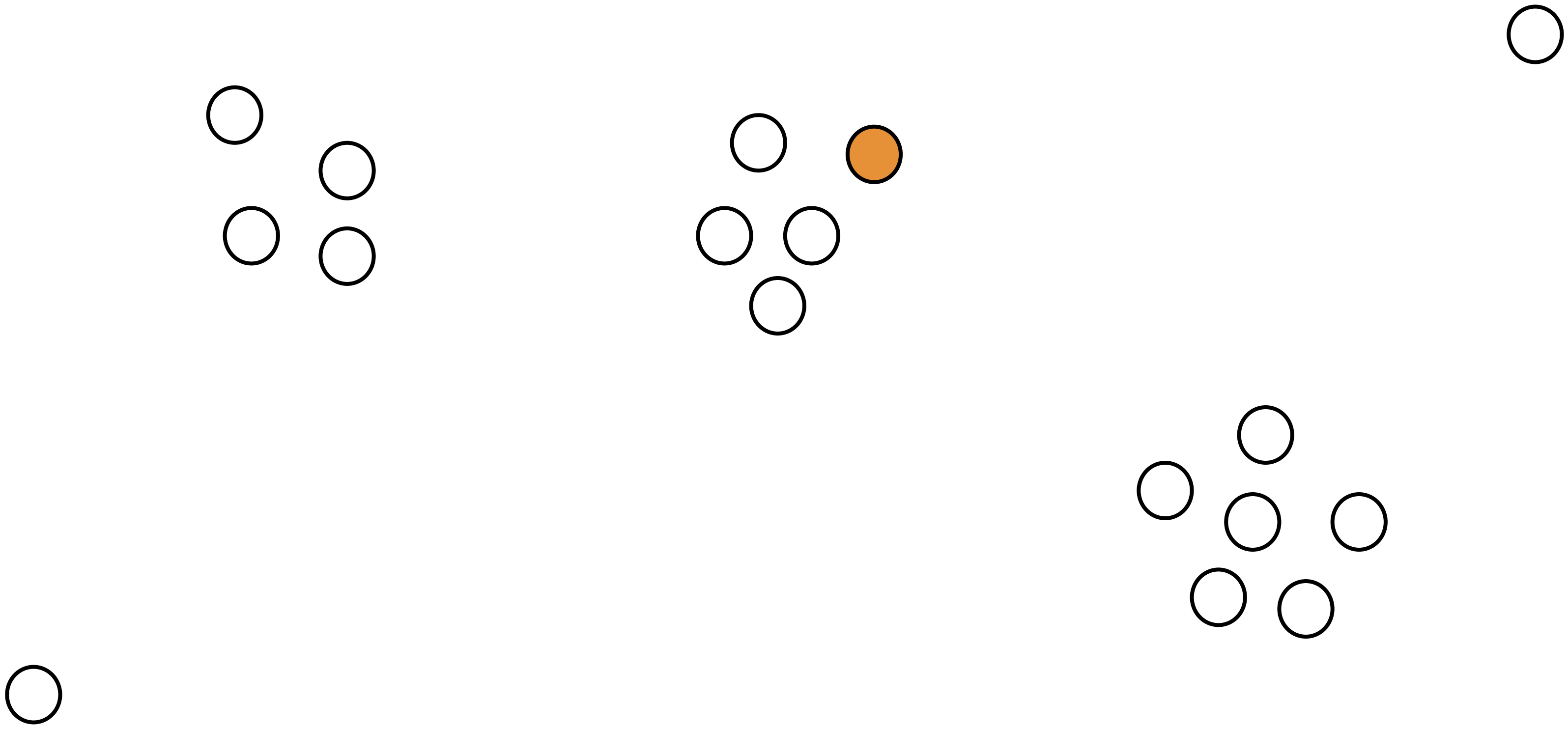


K-means++

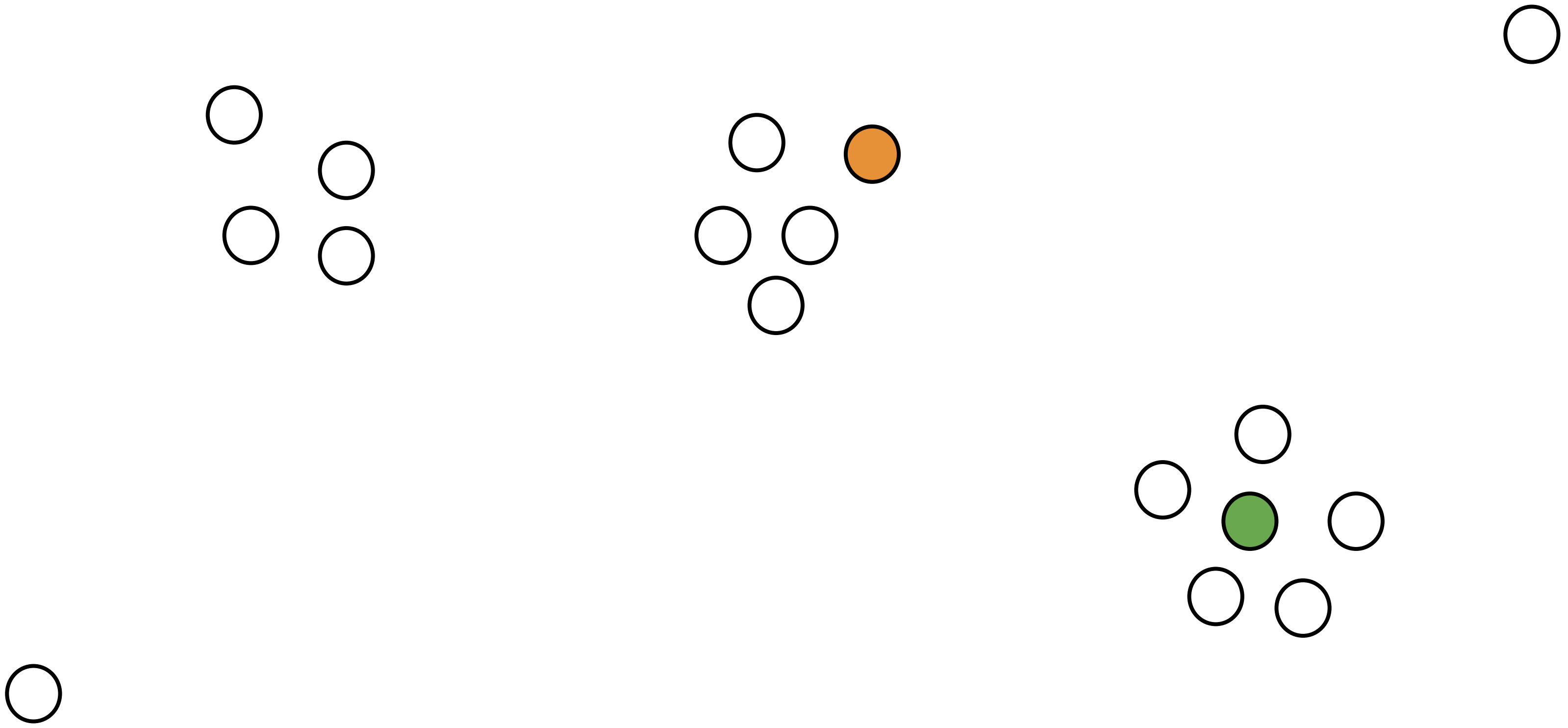
Initialize with a combination of the two methods:

1. Start with a random center
2. Let **D(x)** be the distance between **x** and the closest of the centers picked so far. Choose the next center with probability proportional to **$D(x)^2$**

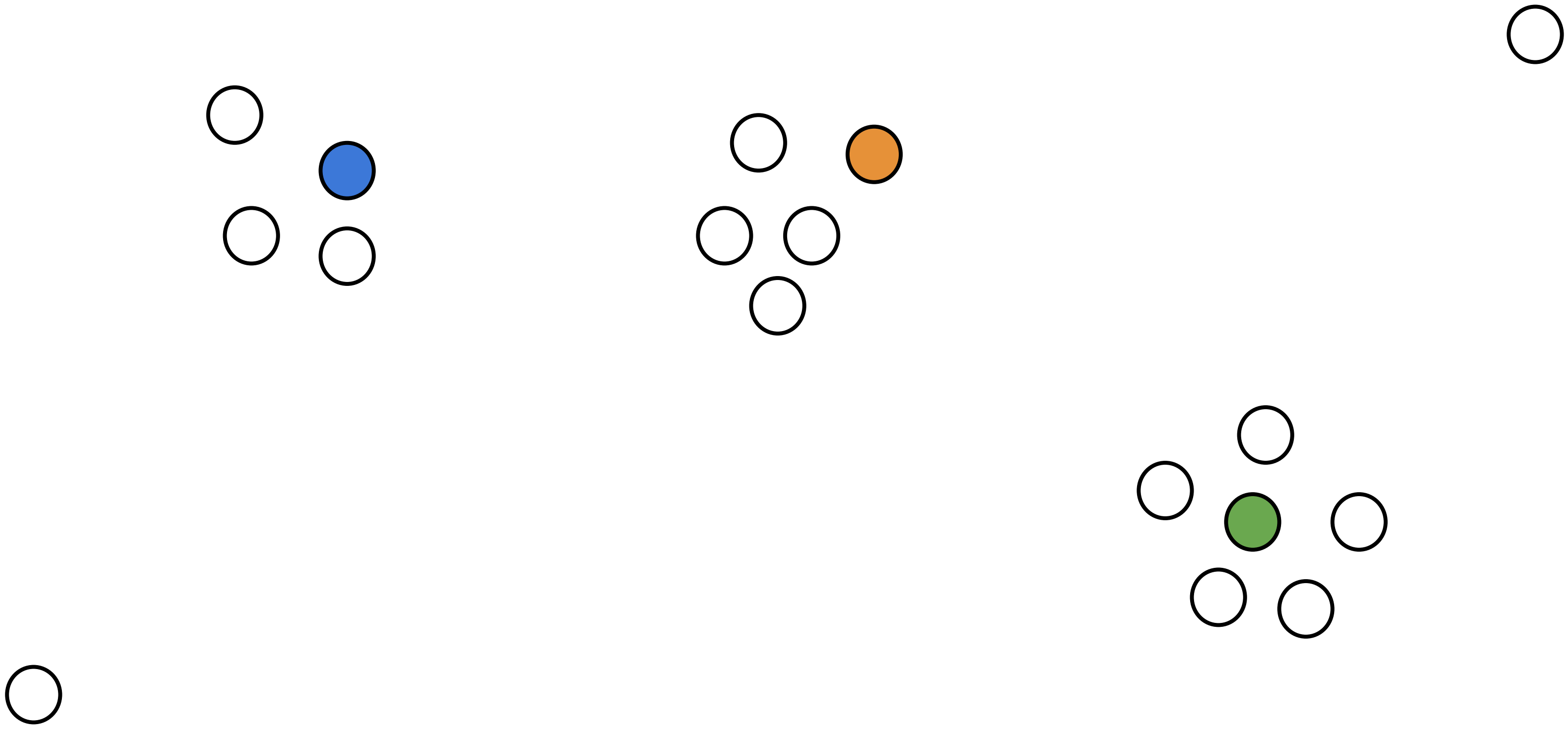
K-means++



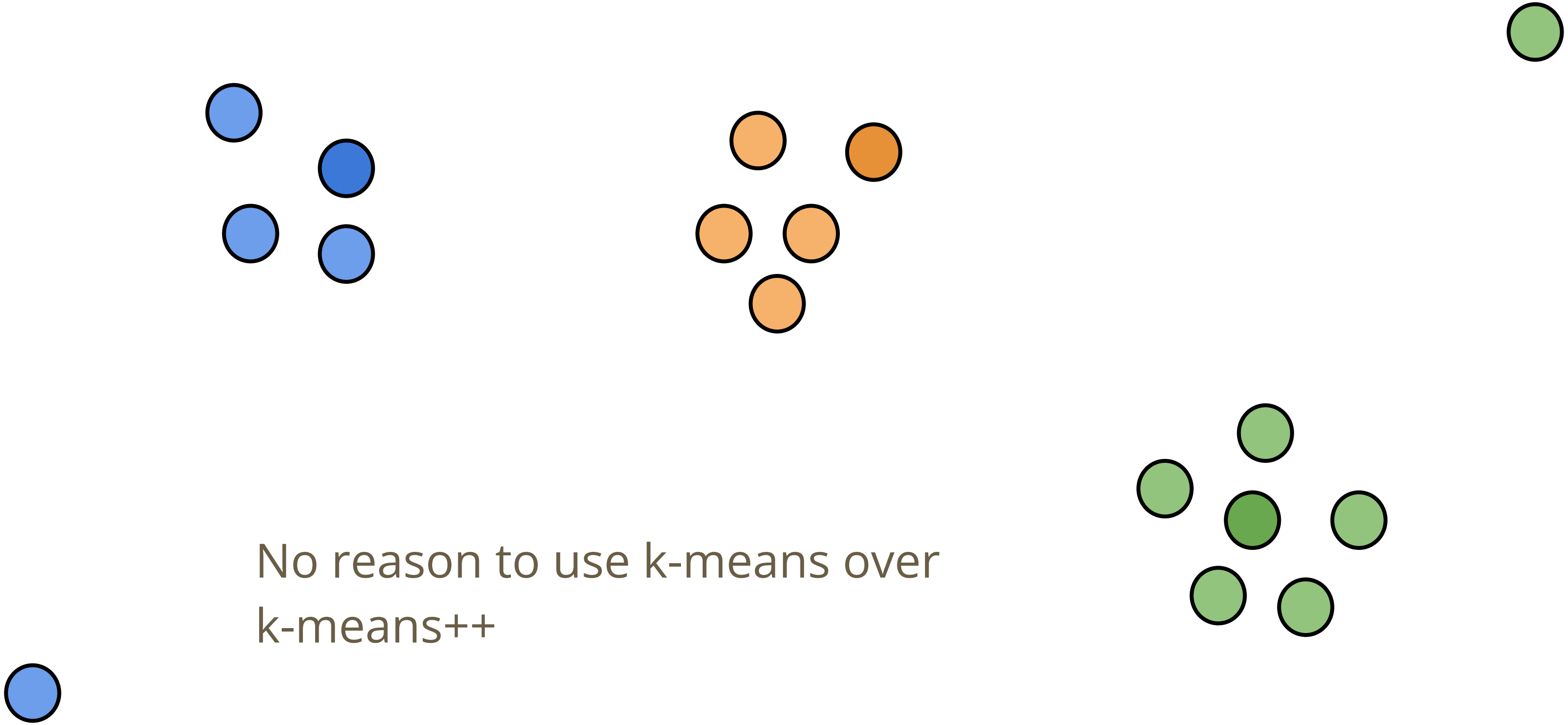
K-means++



K-means++



K-means++



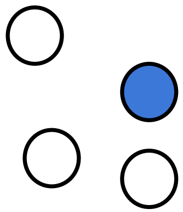
No reason to use k-means over k-means++

K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to **$D(x)^2$**?

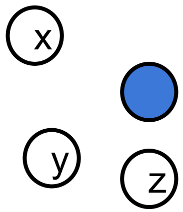
K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to $\mathbf{D(x)^2}$?



K-means++

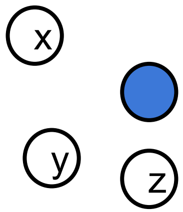
Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to $\mathbf{D(x)^2}$?



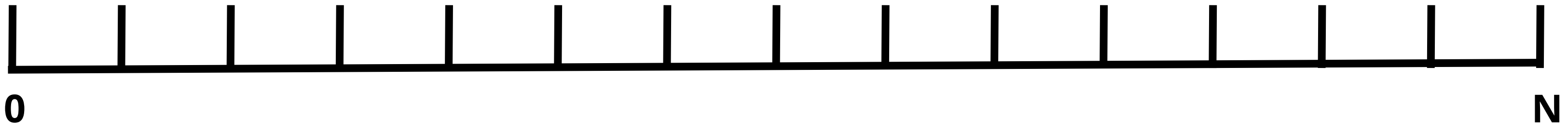
$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to $\mathbf{D(x)^2}$?

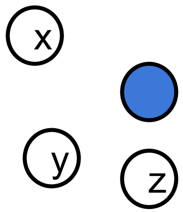


$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

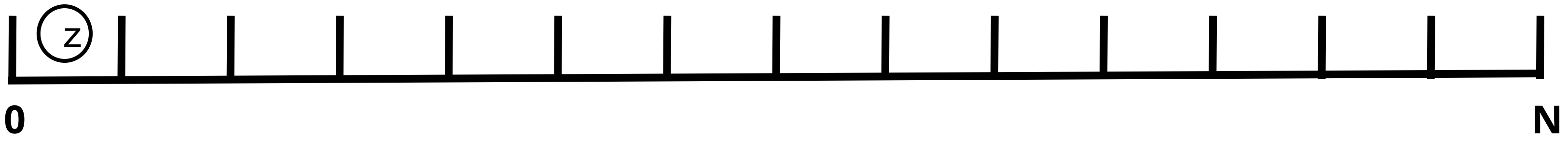


K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to $\mathbf{D(x)^2}$?

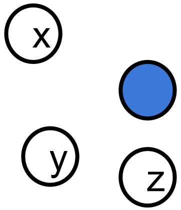


$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

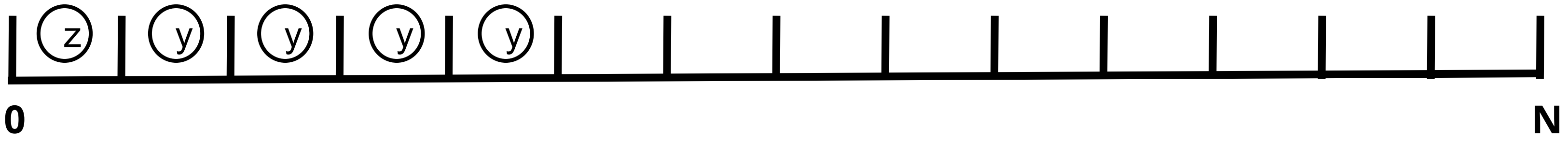


K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to $\mathbf{D(x)^2}$?

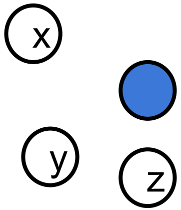


$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

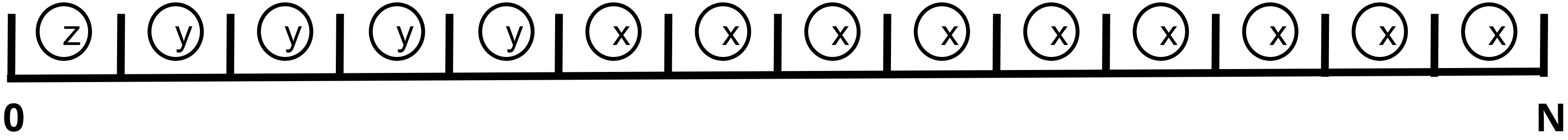


K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to **$D(x)^2$**?

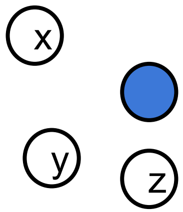


$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

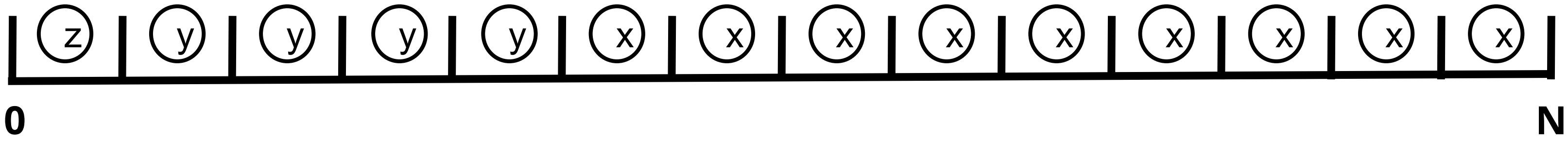


K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to **$D(x)^2$**?



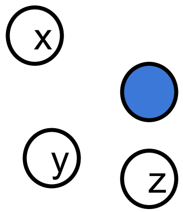
$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$



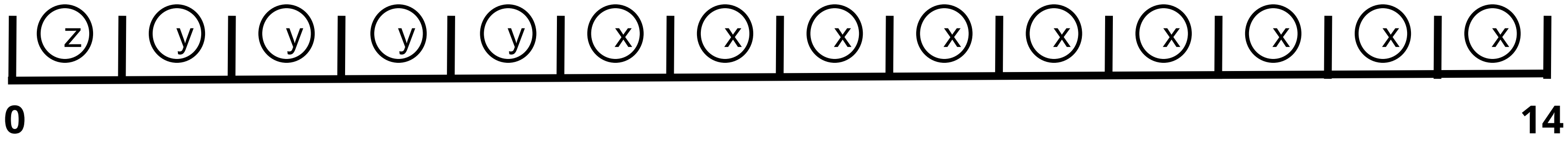
$= D(x)^2 + D(y)^2$
$+ D(z)^2 = 14$

K-means++

Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to $\mathbf{D(x)^2}$?



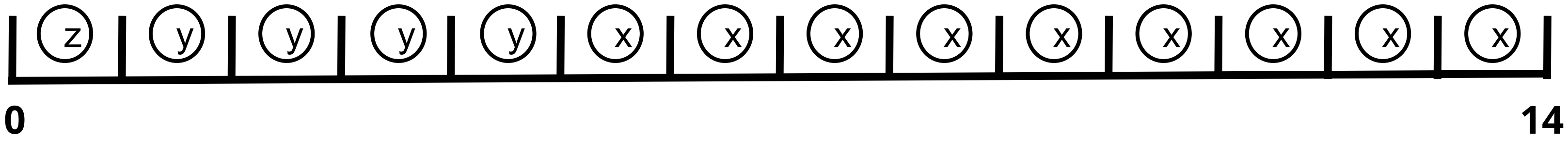
$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$



K-means++

Q3: the black box returns “12” as the random number generated. Which point do we choose for the next center (x, y, or z) ?

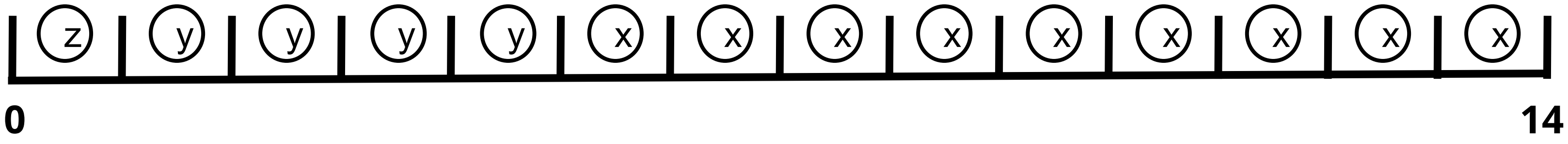
x



K-means++

Q4: the black box returns “4” as the random number generated. Which point do we choose for the next center (x, y, or z) ?

y



K-means++

What happens if the black box can only generate numbers between 0 and 1?

Let's say we have 6 data points:

$$(1,1),(2,2),(10,10),(11,11),(30,30),(50,50)$$

Step 1: Pick First Centroid Randomly

- Suppose we randomly pick **(1,1)** as the first centroid c_1.

Step 2: Compute Distances from c_1

Point	Distance to $c_1 = (1,1)$	Squared Distance $D(x)^2$
(2,2)	$\sqrt{(2-1)^2+(2-1)^2} = \sqrt{2}$	**2**
(10,10)	$\sqrt{(10-1)^2+(10-1)^2} = \sqrt{162}$	**162**
(11,11)	$\sqrt{(11-1)^2+(11-1)^2} = \sqrt{200}$	**200**
(30,30)	$\sqrt{(30-1)^2+(30-1)^2} = \sqrt{1682}$	**1682**
(50,50)	$\sqrt{(50-1)^2+(50-1)^2} = \sqrt{4802}$	**4802**

- The probability of choosing a new centroid is proportional to the squared distances.
- Since **(50,50) has the largest squared distance**, it has the highest probability of being the next centroid.

with like random number generator
-> see the uniform random number part

Step 3: Pick the Next Centroid

- Let's say **(50,50)** is chosen as c_2.

Step 4: Compute New Distances (Now Using 2 Centroids)

- Now, for each remaining point, compute its squared distance **to the nearest centroid** (either $c_1 = (1,1)$ or $c_2 = (50,50)$).

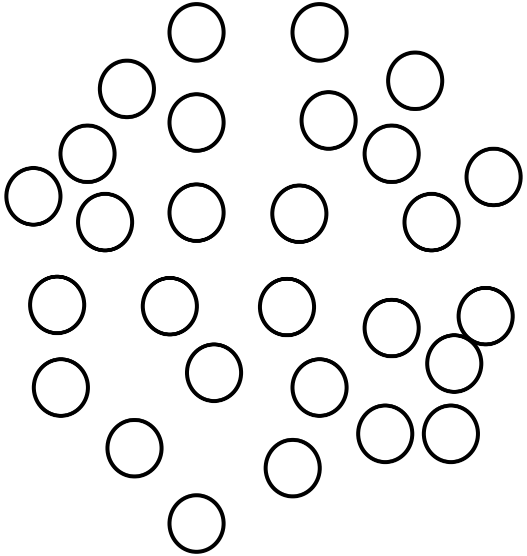
Point	Distance to c_1	Distance to c_2	Nearest Centroid	$D(x)^2$
(2,2)	$\sqrt{2}$	$\sqrt{4608}$	c_1 (1,1)	**2**
(10,10)	$\sqrt{162}$	$\sqrt{3200}$	c_1 (1,1)	**162**
(11,11)	$\sqrt{200}$	$\sqrt{3024}$	c_1 (1,1)	**200**
(30,30)	$\sqrt{1682}$	$\sqrt{800}$	c_2 (50,50)	**800**

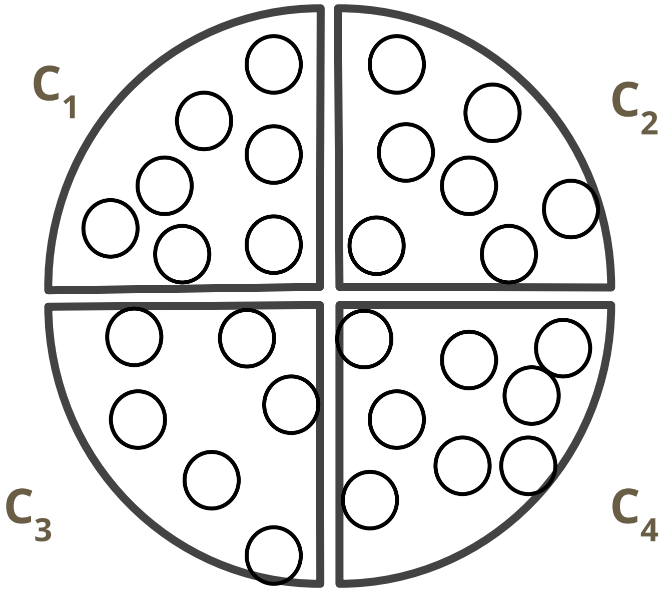
- The point **(30,30)** has the largest squared distance from its nearest centroid, so it is **most likely to be picked next.**

Repeat Until k Centroids are Selected

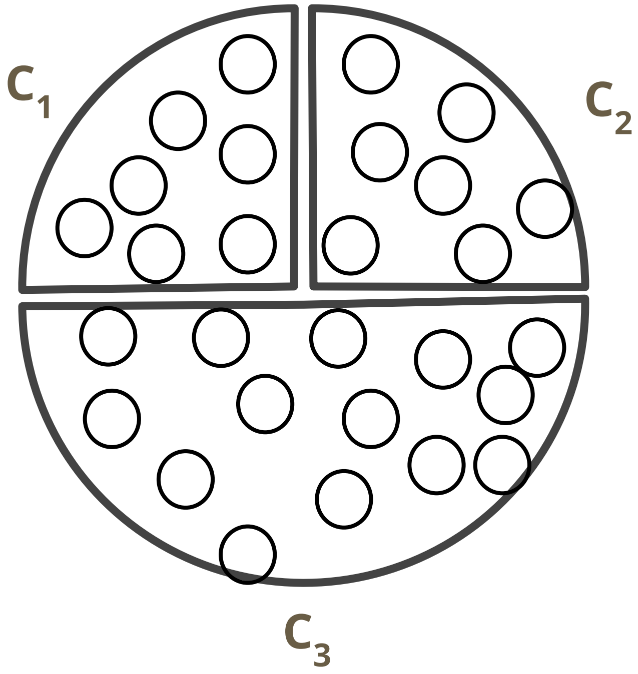
- Keep iterating until we get k centroids.

Kmeans Quizz (take 2)

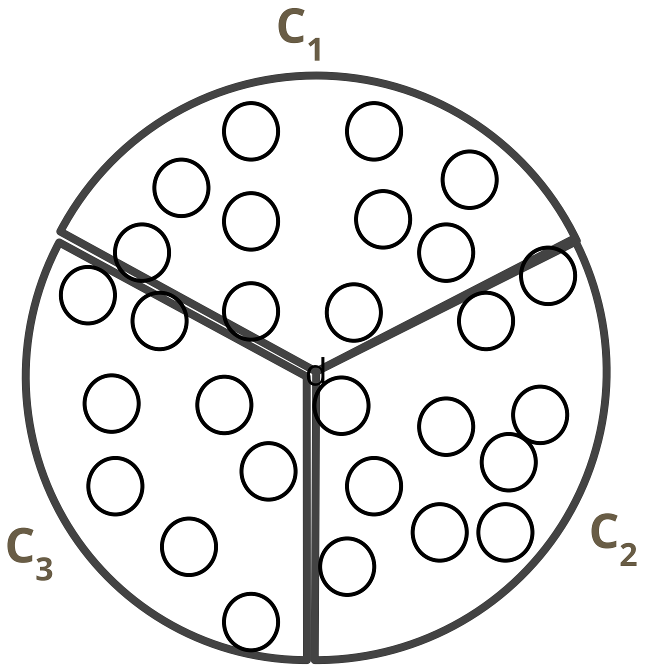




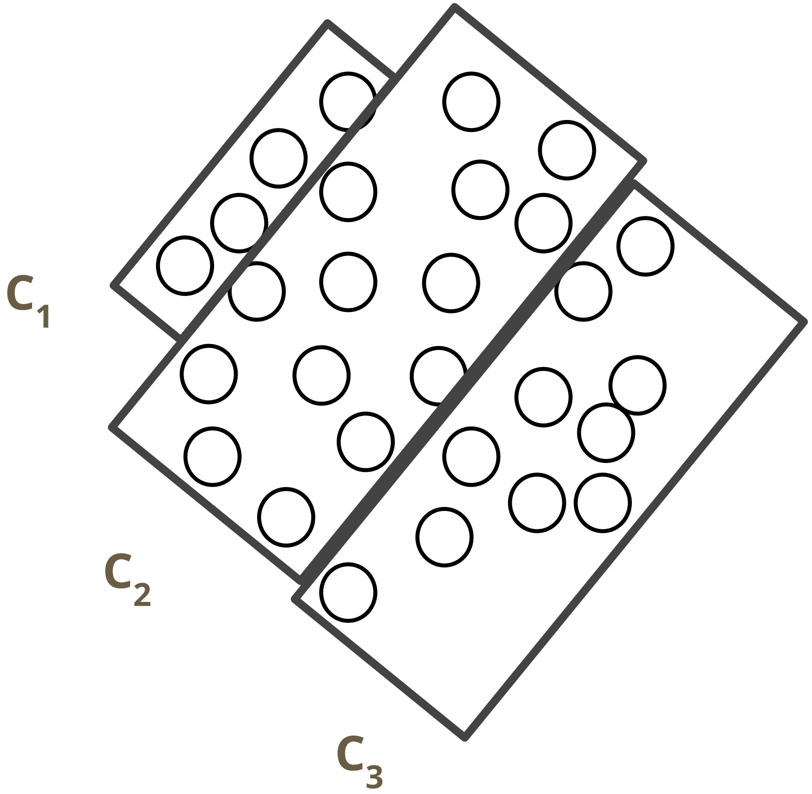
C_1
C_2
C_3
C_4

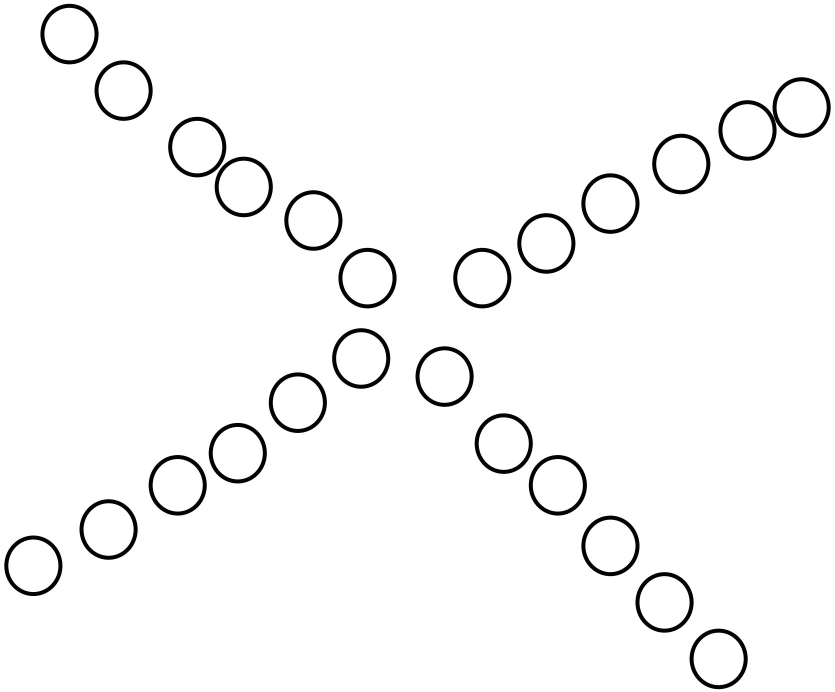


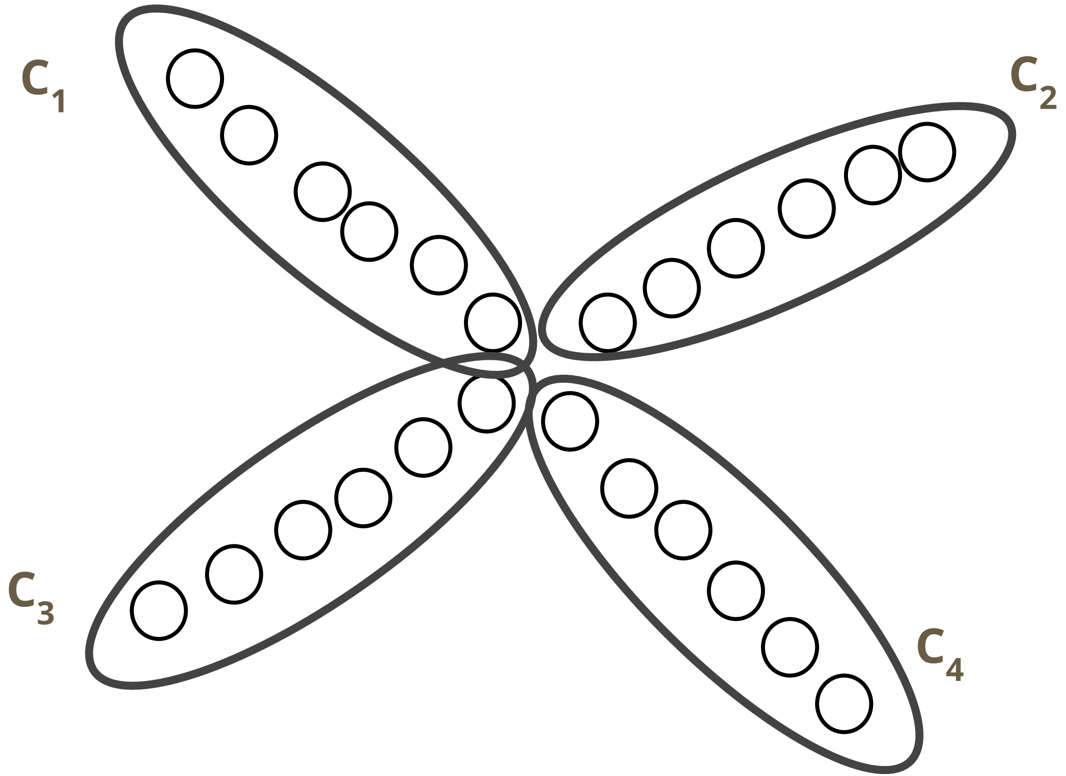
C_1
C_2
C_3

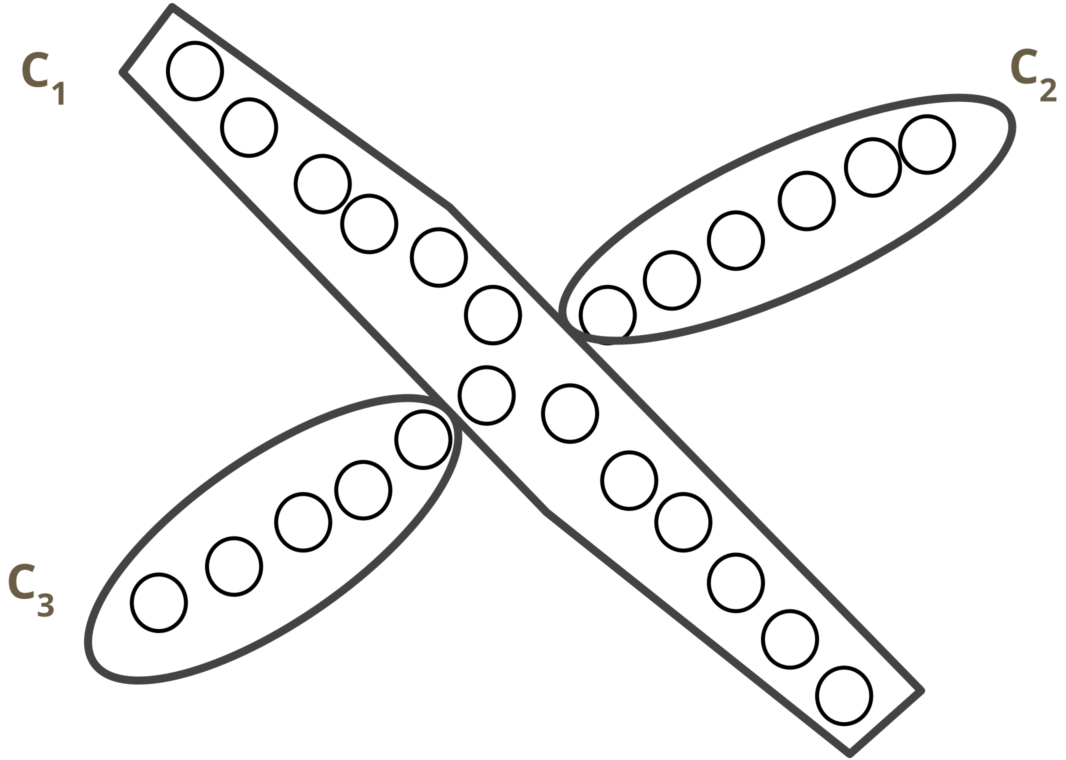


C_1
C_2
C_3

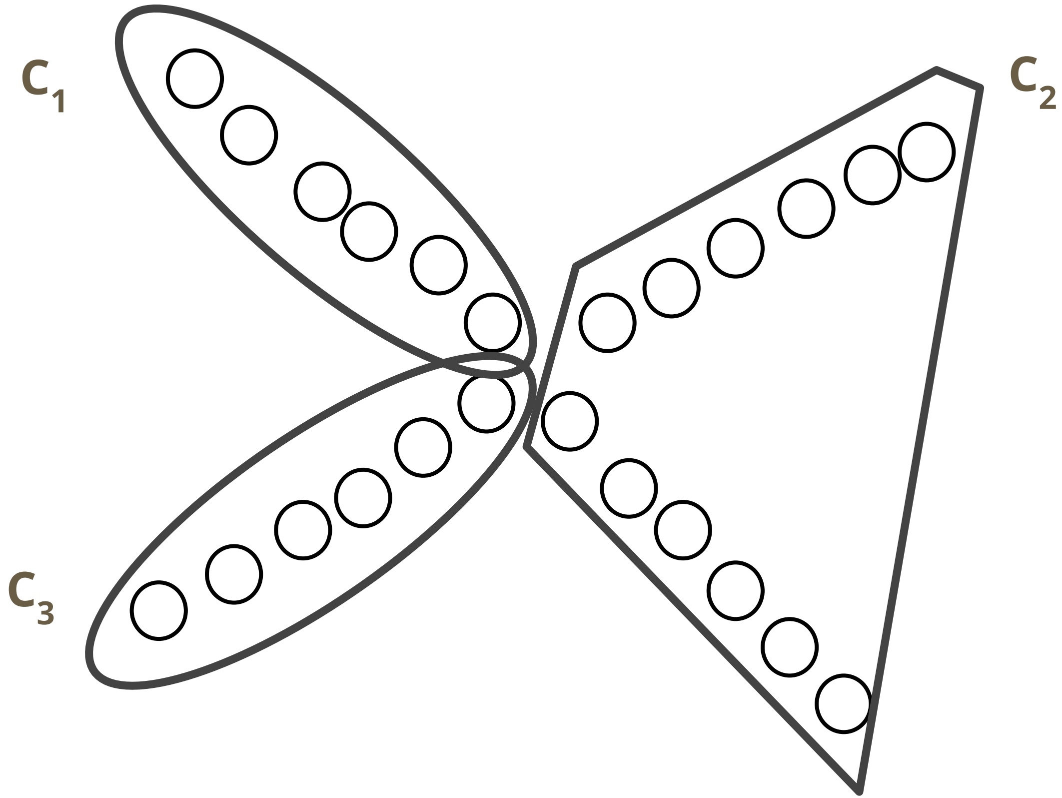

C_1
C_2
C_3

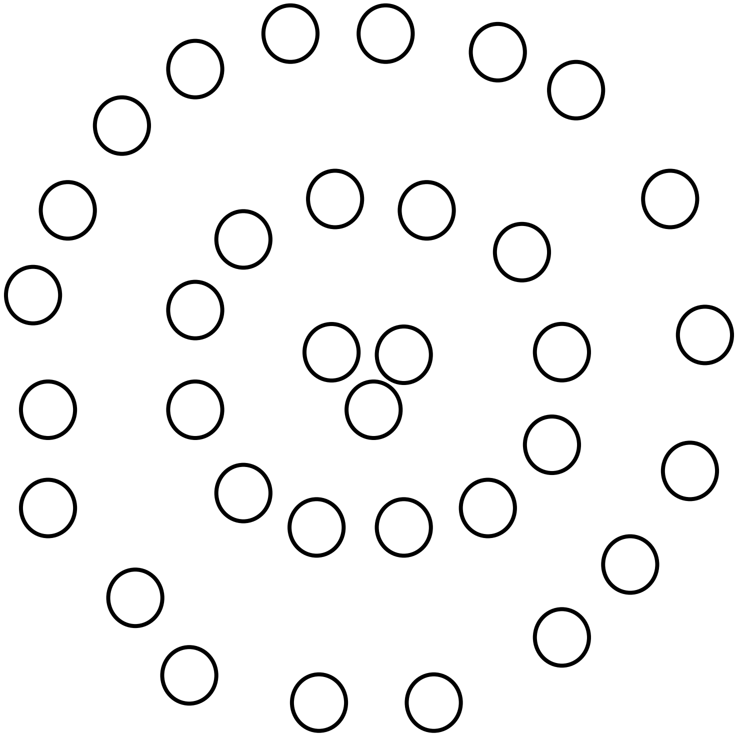


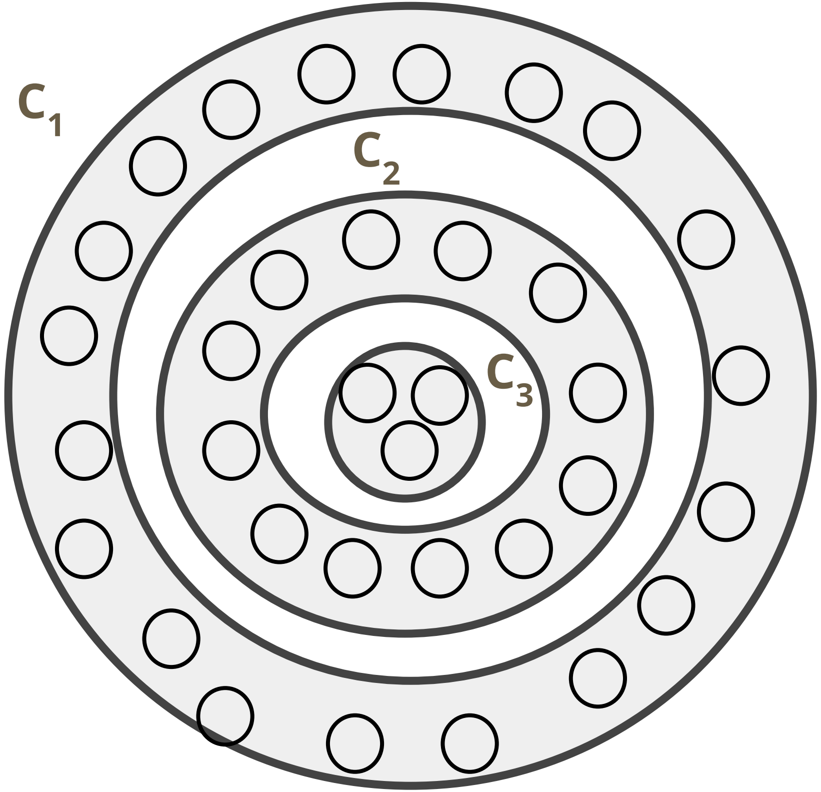

C_1
C_2
C_3
C_4



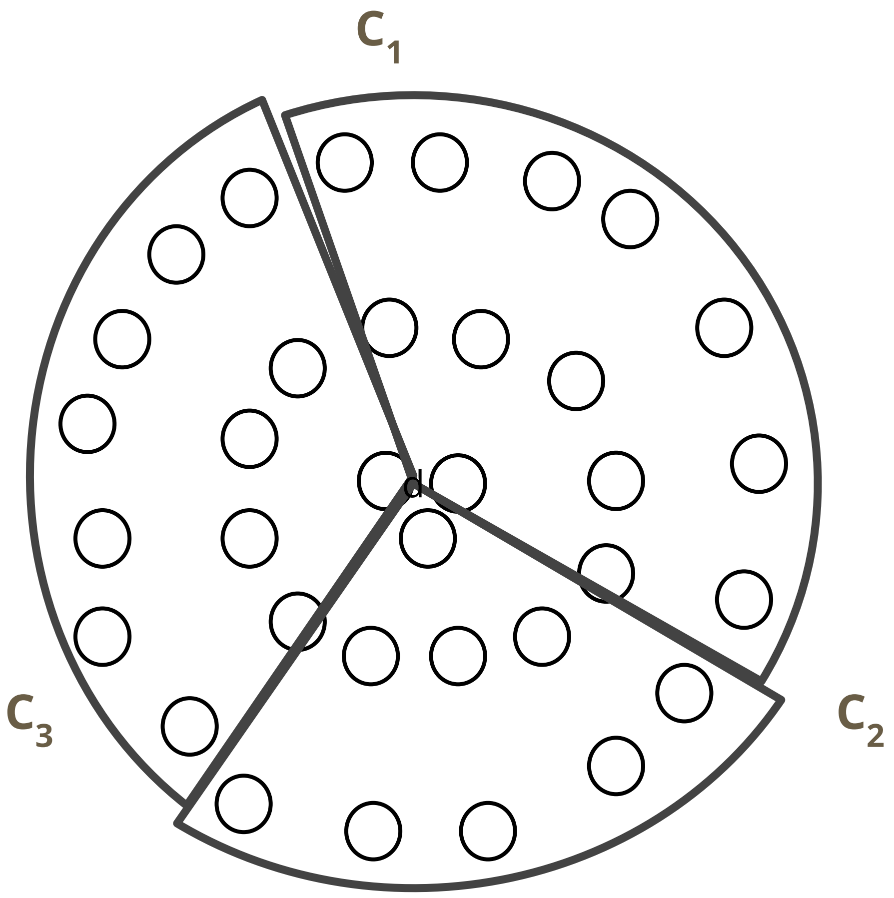
C_1
C_2
C_3


C_1
C_2
C_3

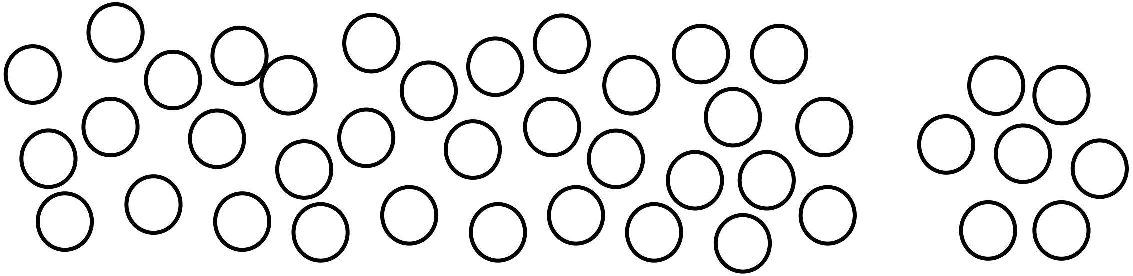


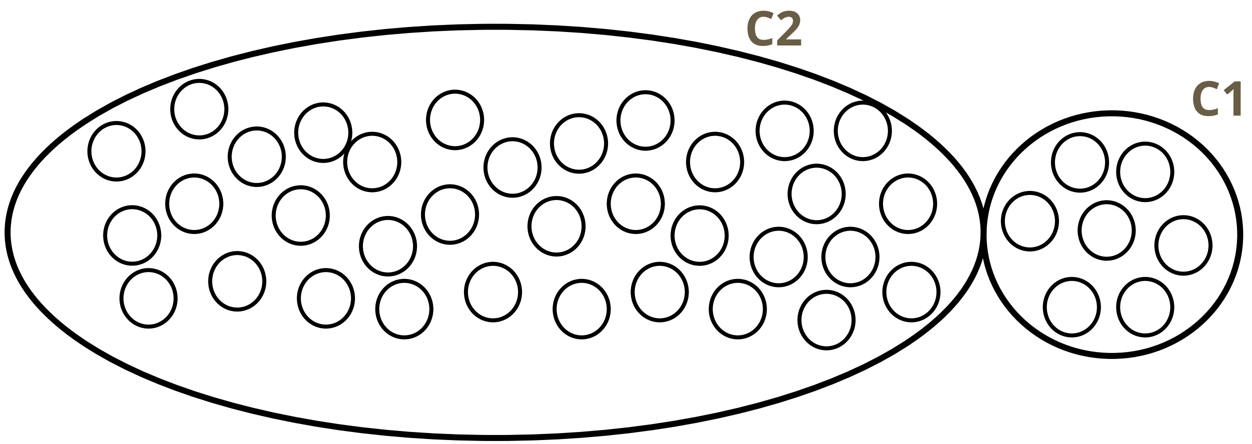


C_1
C_2
C_3



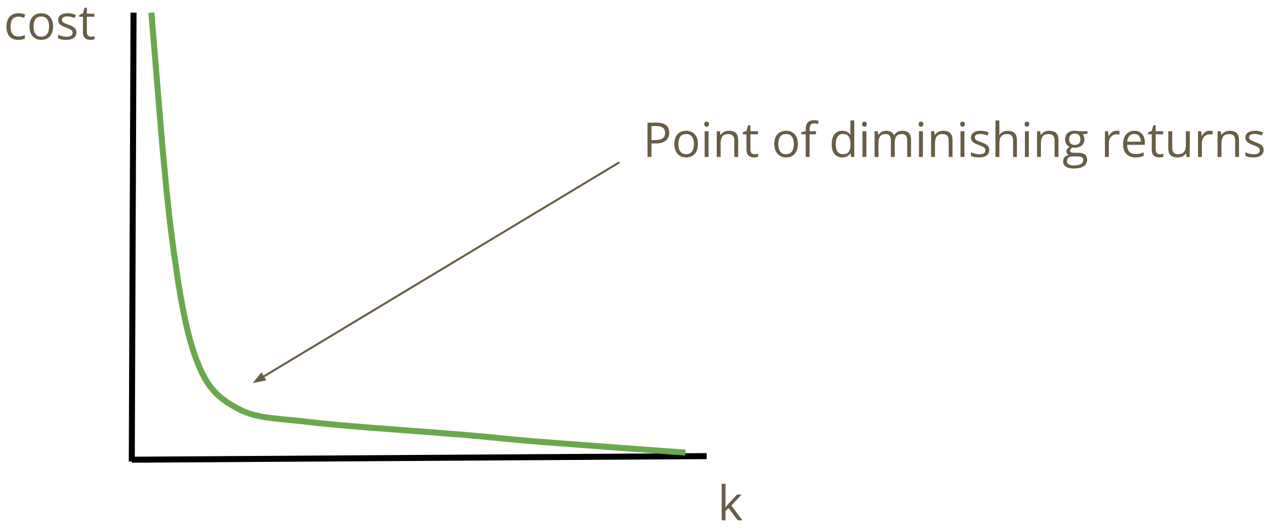
C_1
d
C_3
C_2




C2
C1

How to choose the right k?

1. Iterate through different values of k (elbow method)



How to choose the right k?

1. Iterate through different values of k (elbow method)
2. Use empirical / domain-specific knowledge

   Example: Is there a known approximate distribution of the data? (K-means is good for spherical gaussians)
3. Metric for evaluating a clustering output

Evaluation

Recall our goal: Find a clustering such that

- **Similar** data points are in the **same cluster**
- **Dissimilar** data points are in **different clusters**

Evaluation

Recall our goal: Find a clustering such that

- **Similar** data points are in the **same cluster** ✅
- **Dissimilar** data points are in **different clusters**

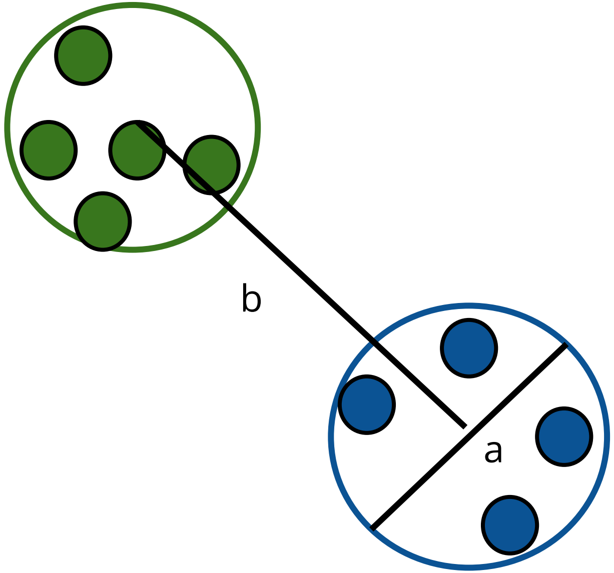
Evaluation

K-means cost function tells us the within-cluster distances between points will be small overall.

But what about the intra-cluster distance? Are the clusters we created far? How far? Relative to what?

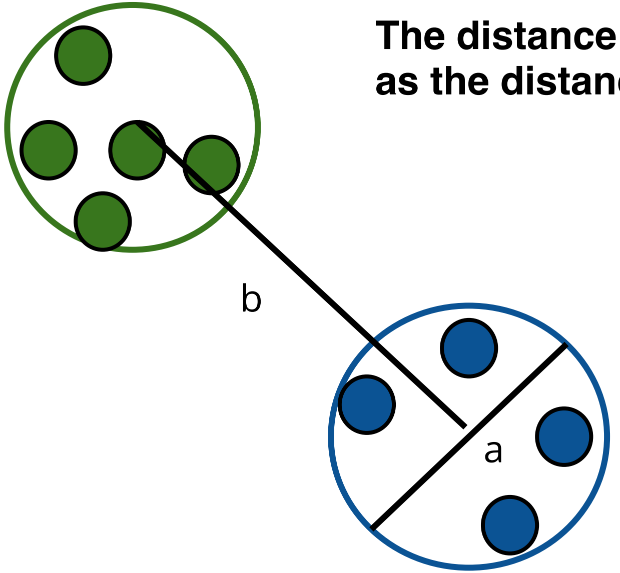
Discuss - 5min

Define a metric that evaluates how spread out the clusters are from one another.



a: average within-cluster distance

b: average intra-cluster distance



The distance within the cluster is the same as the distance between the clusters

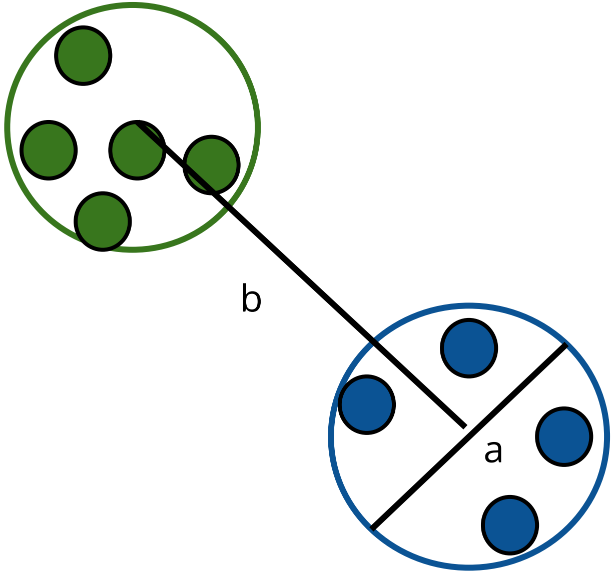
a: average within-cluster distance
b: average intra-cluster distance

What does it mean for (b - a) to be 0?

means clusters are right next to each other

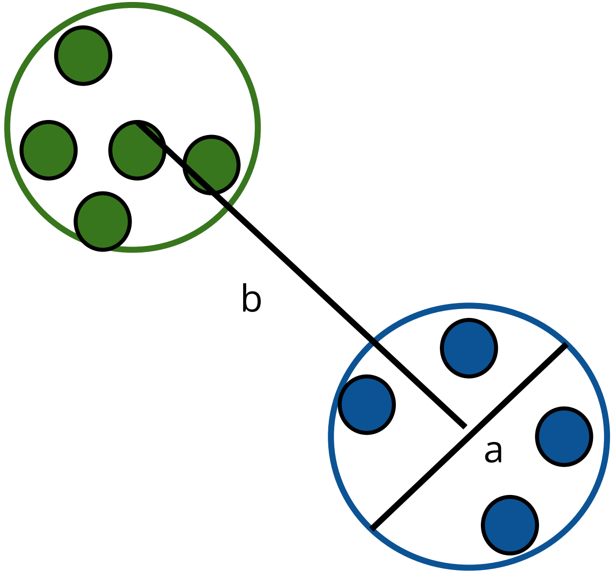
Clusters are well-separated: A large b suggests that the clusters are far apart, meaning the data points from different clusters are distinctly different.

Clusters are compact: A small a indicates that the points within each cluster are tightly packed, meaning the intra-cluster similarity is high.

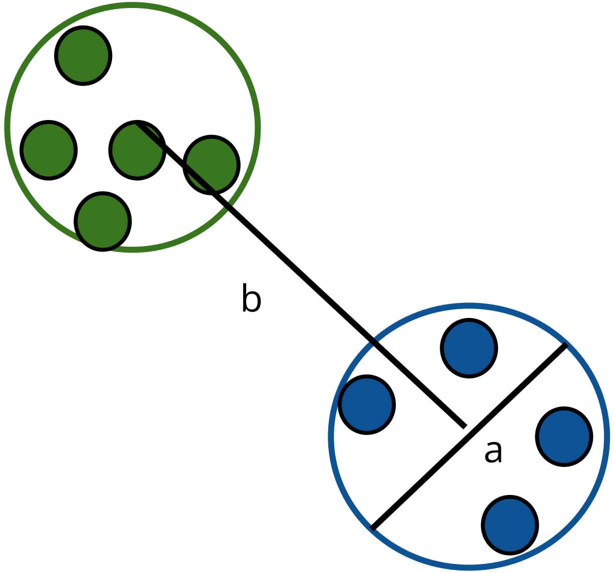


a: average within-cluster distance
b: average intra-cluster distance

What does it mean for (b - a) to be large?

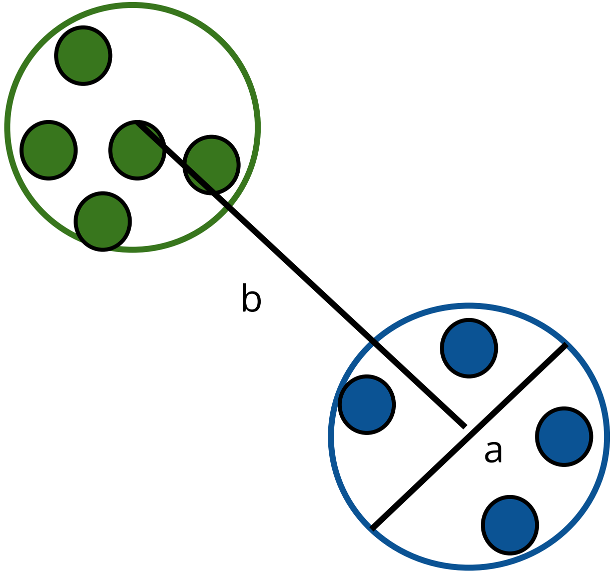


The value of (b-a) doesn't mean much by itself. Can we compare it to something so that the ratio becomes a value between 0 and 1?



Value between 0 and 1

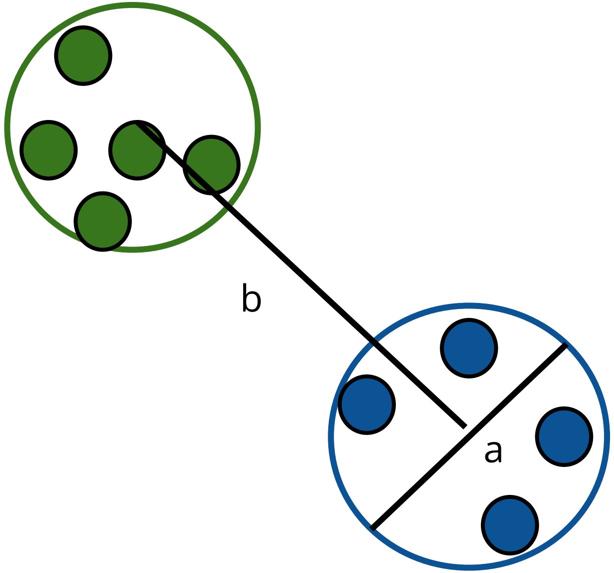
(b - a) / max(a, b)



close to 1: good
close to 0: not good

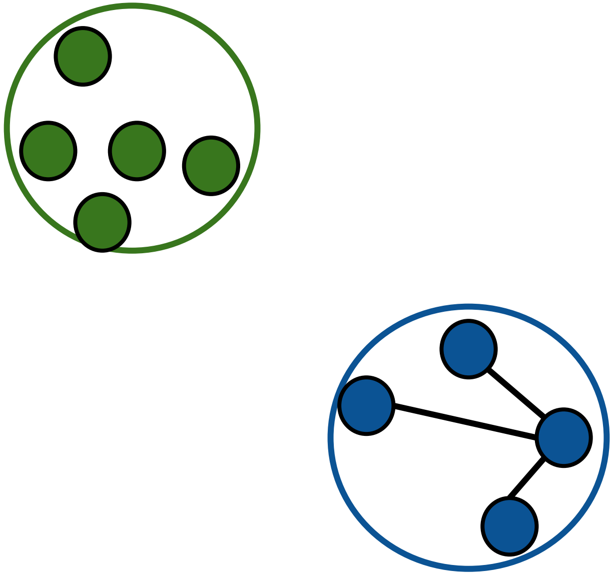
What does it mean for (b - a) / max(a, b) to be close to 1?

close to 1: b-a is large so the data points are well-separated from each other and from other clusters
close to 0: not good; opposite



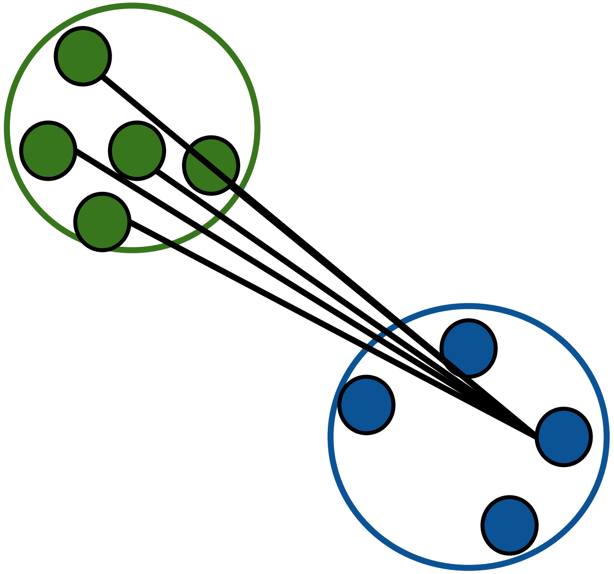
What does it mean for (b - a) / max(a, b) to be close to 0?

Silhouette Scores



For each data point i:
a_i : mean distance from point i to every other point in its cluster

Silhouette Scores

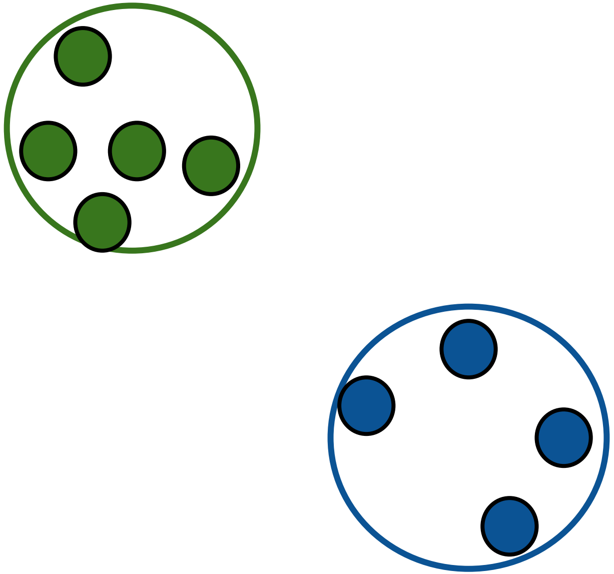


For each data point i:

a_i : mean distance from point i to every other point in its cluster

b_i : smallest mean distance from point i to every point in another cluster

Silhouette Scores



For each data point i:

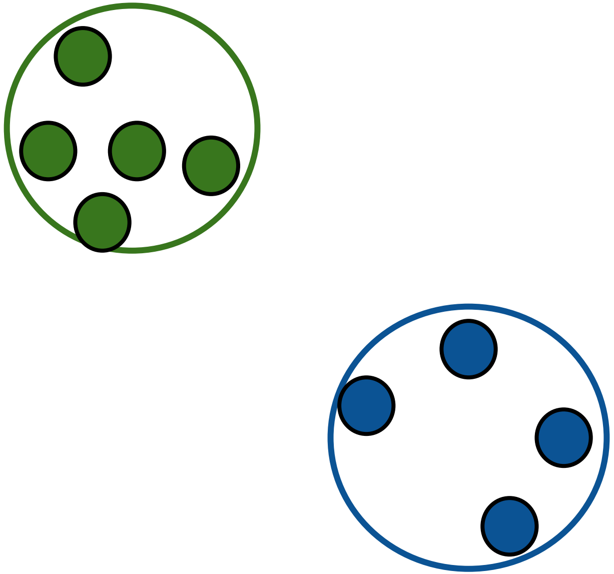
a_i : mean distance from point i to every other point in its cluster

b_i : smallest mean distance from point i to every point in another cluster

$s_i = (b_i - a_i) / \max(a_i, b_i)$

Silhouette Scores

silhouette score ranges from -1 to 1
close to 1: the samples are well-clustered, a score
close to 0: overlapping clusters
negative score: the samples might have been assigned to the wrong cluster



$s_i = (b_i - a_i) / \max(a_i, b_i)$

Silhouette score plot

OR

return the mean s_i over the entire dataset as a measure of goodness of fit

Let's say we have **two clusters**:

Cluster 1: $A(1,1), B(2,2), C(3,3)$
Cluster 2: $D(10,10), E(11,11), F(12,12)$

Now, let's calculate the silhouette score for point $A(1,1)$.

1. Compute $a(i)$: Intra-cluster distance

$a(i)$ is the average distance between $A(1,1)$ and the other points in **Cluster 1**:

$$a(A) = \frac{\text{dist}(A,B) + \text{dist}(A,C)}{2}$$

Using Euclidean distance:

$$\text{dist}(A,B) = \sqrt{(2-1)^2 + (2-1)^2} = \sqrt{2} \approx 1.41$$

$$\text{dist}(A,C) = \sqrt{(3-1)^2 + (3-1)^2} = \sqrt{8} \approx 2.83$$

$$a(A) = \frac{1.41 + 2.83}{2} = 2.12$$

cluster가 여러개라면 거기서 나오는 mean과 비교

2. Compute $b(i)$: Nearest-cluster distance

$b(A)$ is the average distance from A to **all** points in the **nearest cluster (Cluster 2)**.

$$b(A) = \frac{\text{dist}(A, D) + \text{dist}(A, E) + \text{dist}(A, F)}{3}$$

$$\text{dist}(A, D) = \sqrt{(10-1)^2 + (10-1)^2} = \sqrt{162} \approx 12.73$$

$$\text{dist}(A, E) = \sqrt{(11-1)^2 + (11-1)^2} = \sqrt{200} \approx 14.14$$

$$\text{dist}(A, F) = \sqrt{(12-1)^2 + (12-1)^2} = \sqrt{242} \approx 15.56$$

$$b(A) = \frac{12.73 + 14.14 + 15.56}{3} = 14.14$$

---

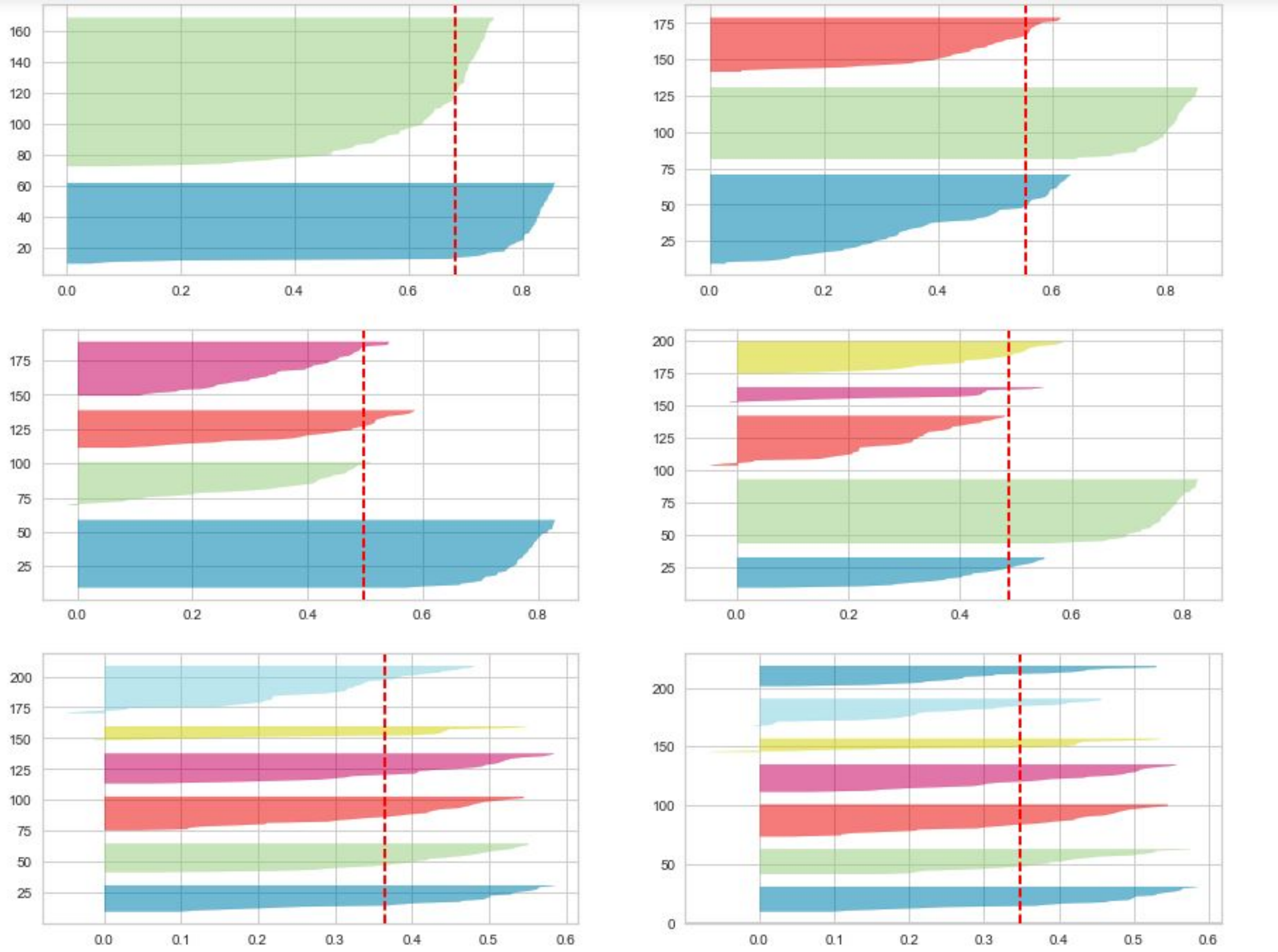
3. Compute $s(A)$: Silhouette Score

$$s(A) = \frac{b(A) - a(A)}{\max(a(A), b(A))}$$

$$s(A) = \frac{14.14 - 2.12}{\max(2.12, 14.14)} = \frac{12.02}{14.14} = 0.85$$

Step 3: Interpret the Result

- $s(A) = 0.85$ is **close to 1**, meaning A is **well-clustered** and far from the other cluster.
- If we repeat this for all points and take the **average**, we get the overall **Silhouette Score** for the clustering.



160 140 120 100 80 60 40 20
0.0 0.2 0.4 0.6 0.8
175 150 125 100 75 50 25
0.0 0.2 0.4 0.6 0.8
175 150 125 100 75 50 25
0.0 0.2 0.4 0.6 0.8
200 175 150 125 100 75 50 25
0.0 0.2 0.4 0.6 0.8
200 175 150 125 100 75 50 25
0.0 0.1 0.2 0.3 0.4 0.5 0.6
200 150 100 50 0
0.0 0.1 0.2 0.3 0.4 0.5 0.6

K-means Variations

1. K-medians (uses the L_1 norm / manhattan distance)
2. K-medoids (any distance function + the centers must be in the dataset)
3. Weighted K-means (each point has a different weight when computing the mean)