Kmeans++

Boston University CS 506 - Lance Galletti

Sum $\sim N(n\mu, n\sigma^2)$

Average $\sim N\left(\mu, \frac{\sigma^2}{n}\right)$

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 and a finite set of points, can't have an infinite sequence of costs

1. The algorithm gets stuck in a cycle / loop : perfectly equidistant points → variance / cost remains unchanged
   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

can only reduce or keep cost equal so its impossible to have a loop

Conclusion: Lloyd's Algorithm always converges!

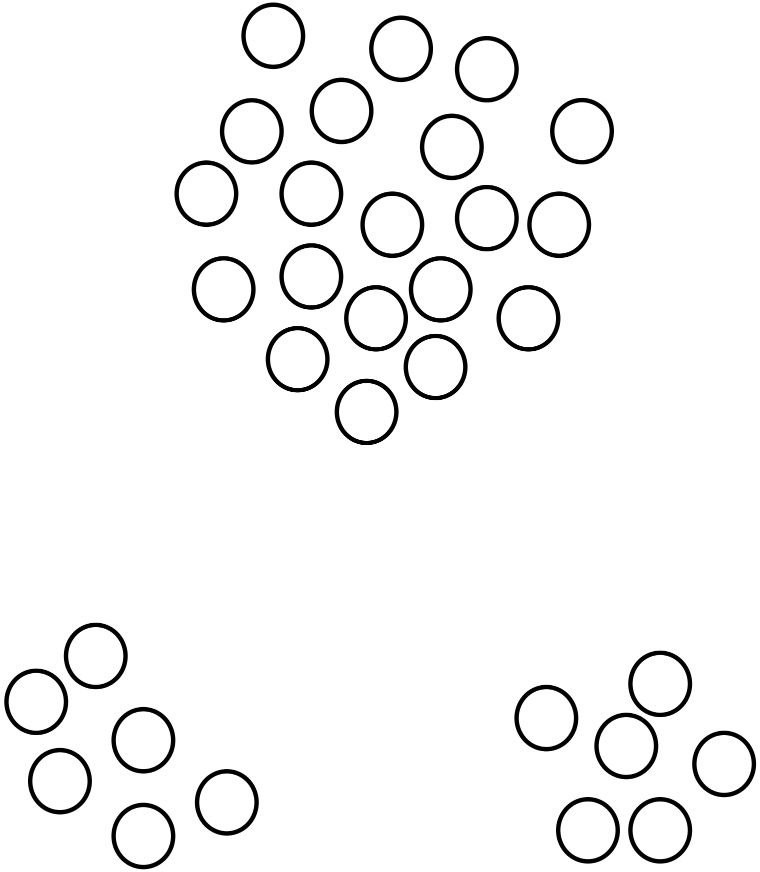
goal:

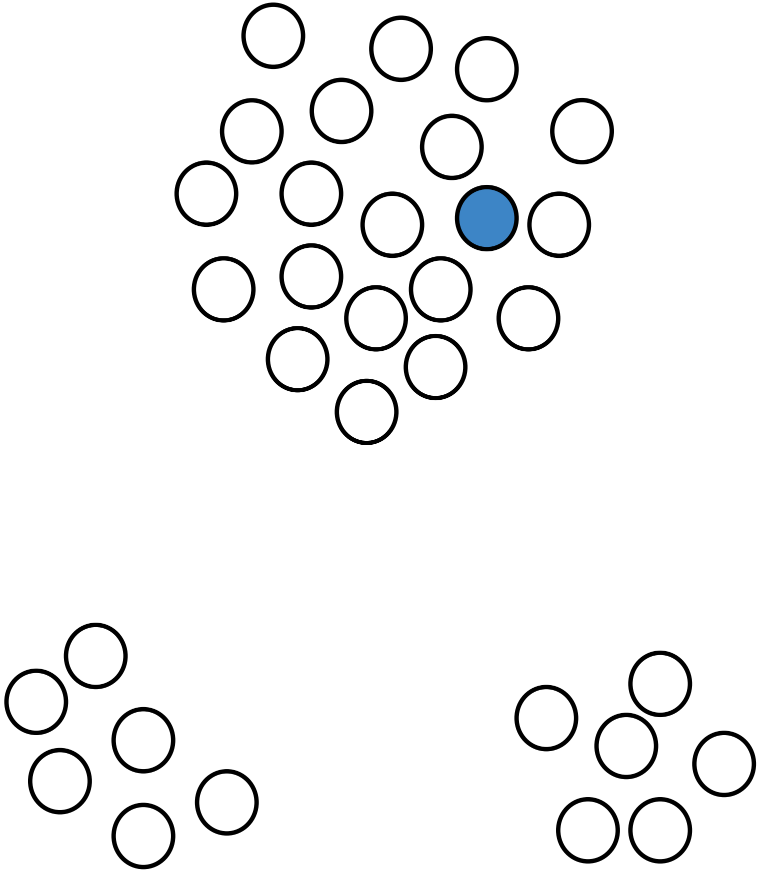
sum of the variances
created by the clustering algorithm

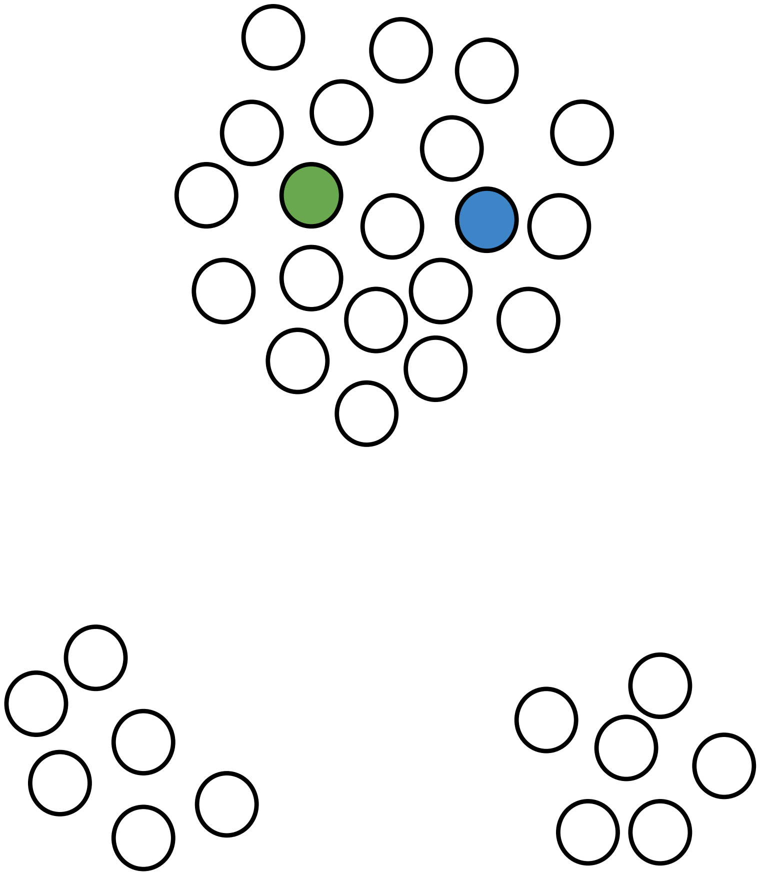
does this algorithm always converge?

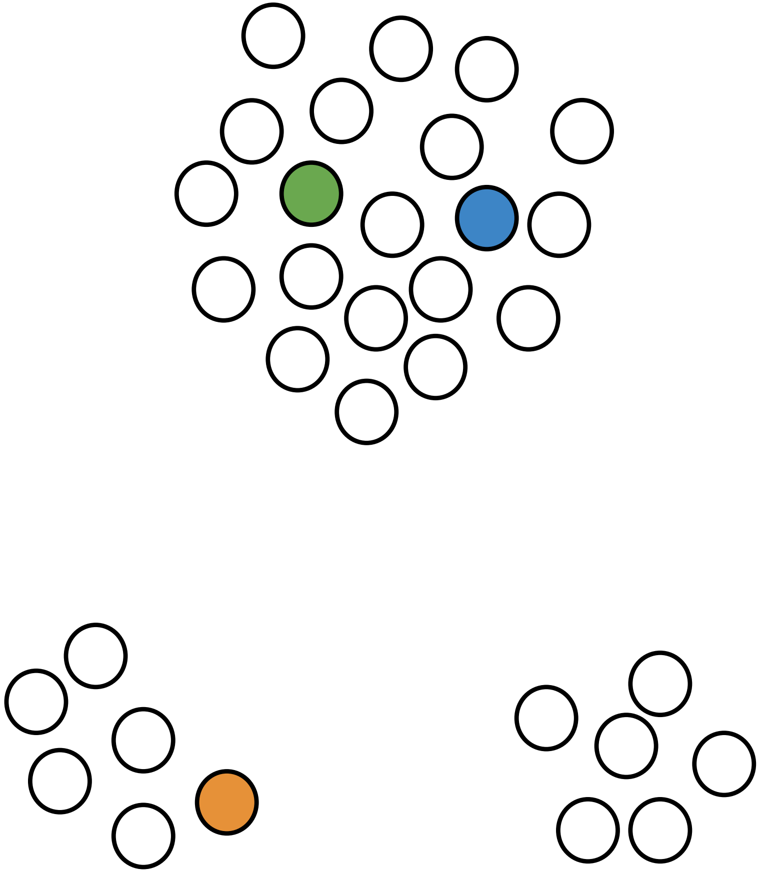
K-means - Lloyd's Algorithm

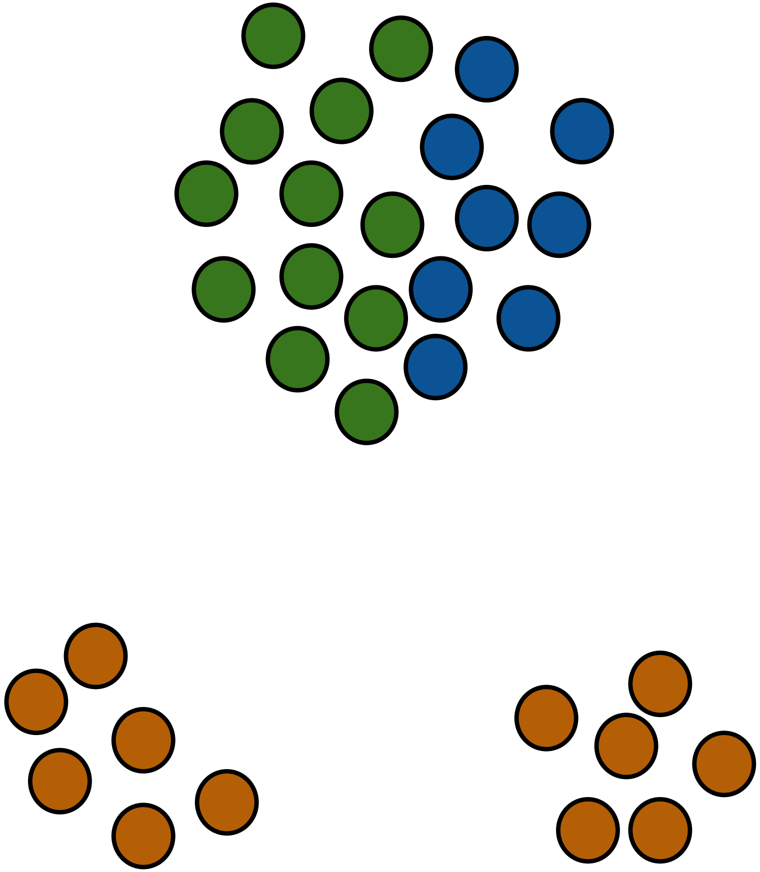
Q2: Will this always converge to the optimal solution?









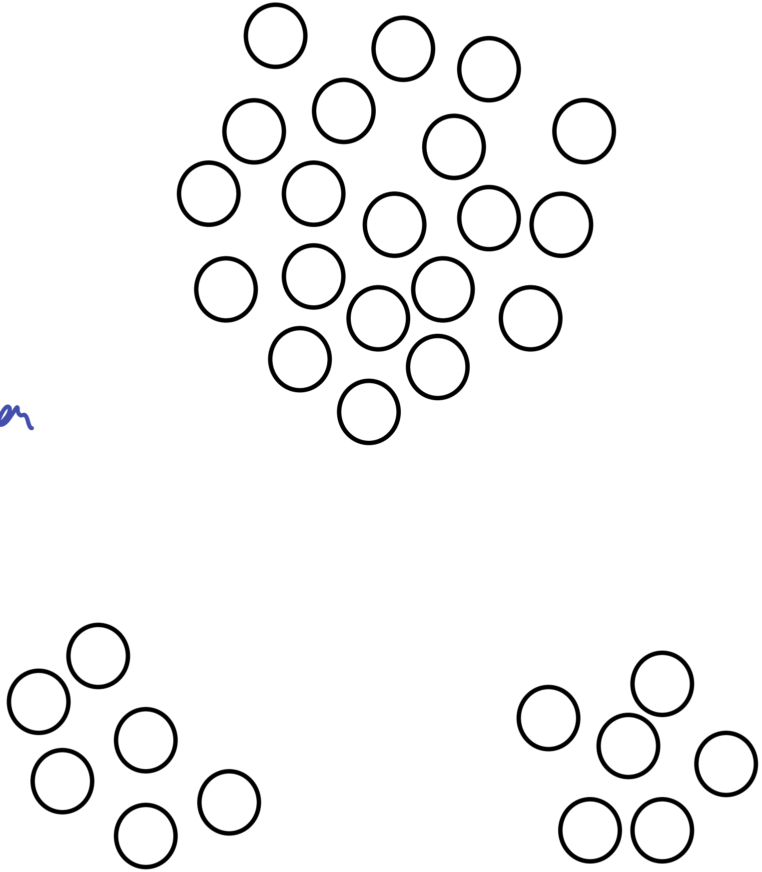


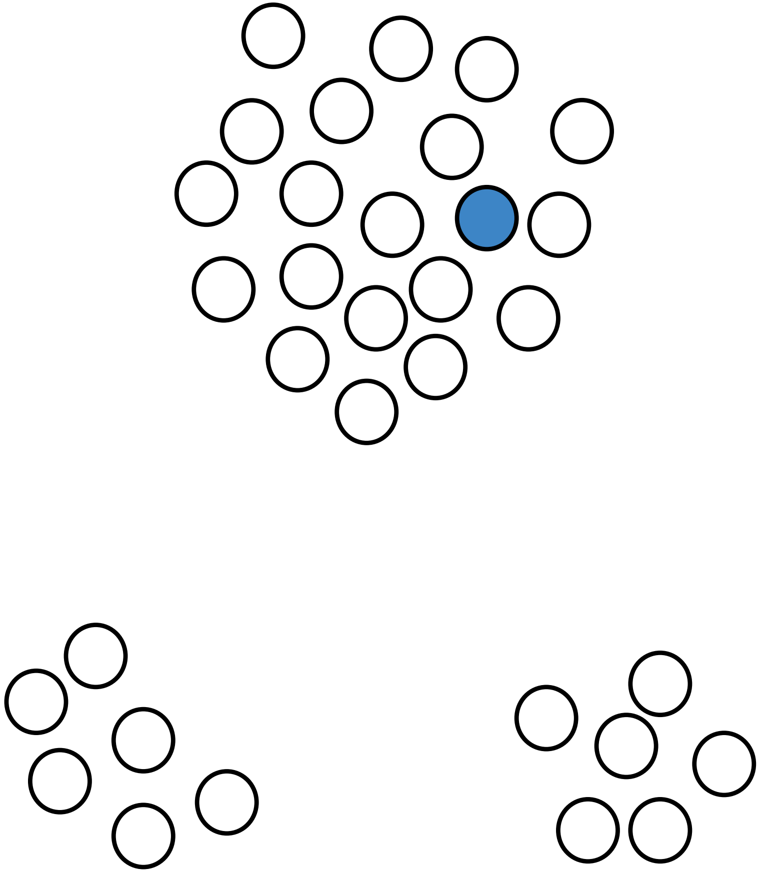
What's the problem?

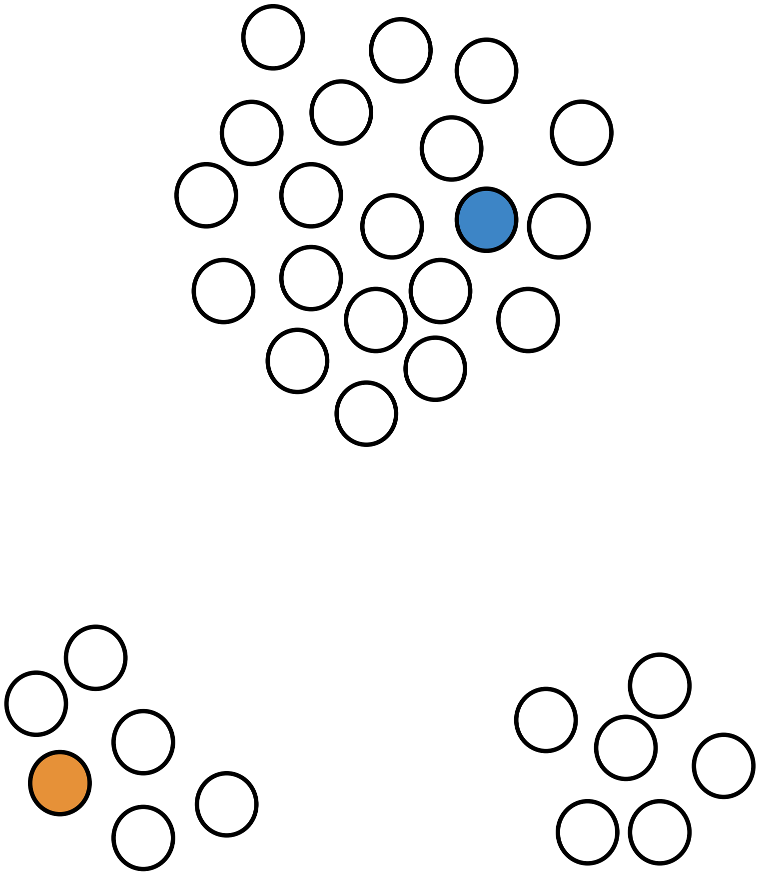
furthest first traversal

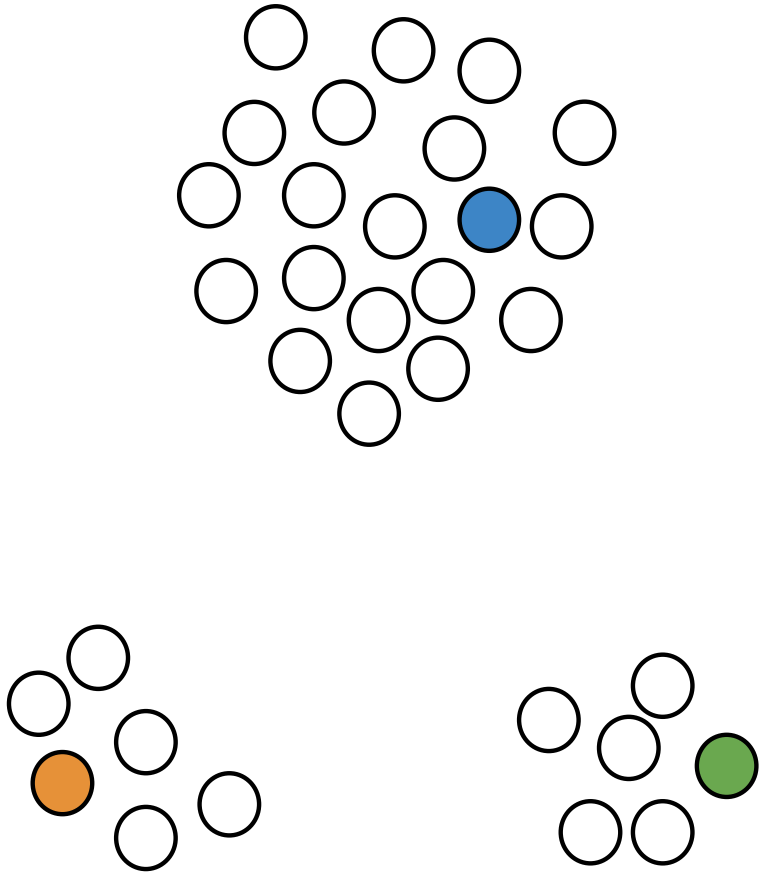
first traversal

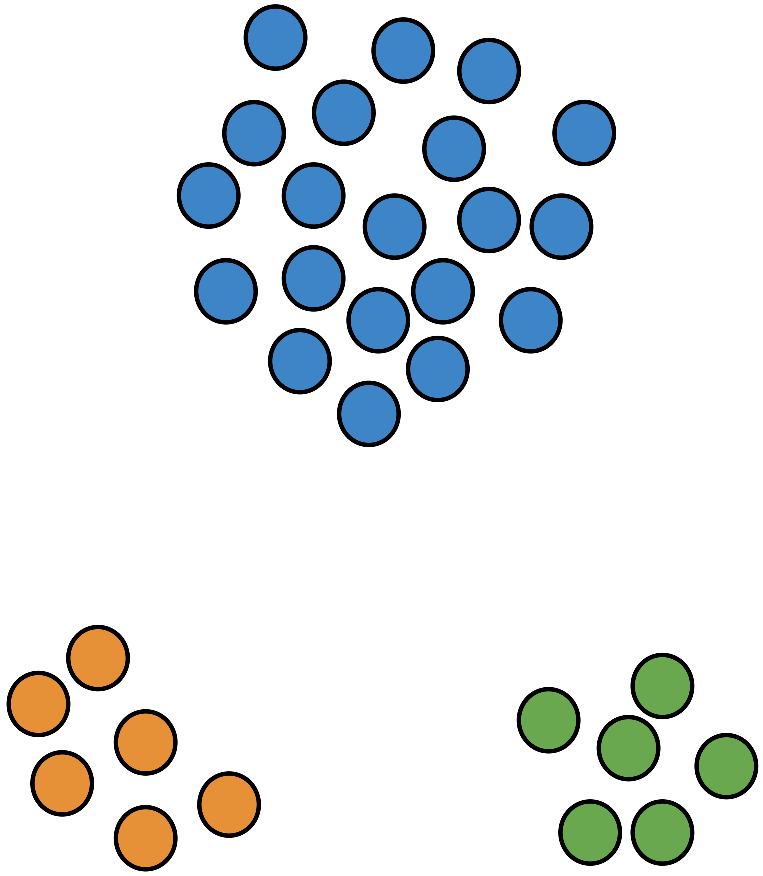
- choosing points that are far from eachother



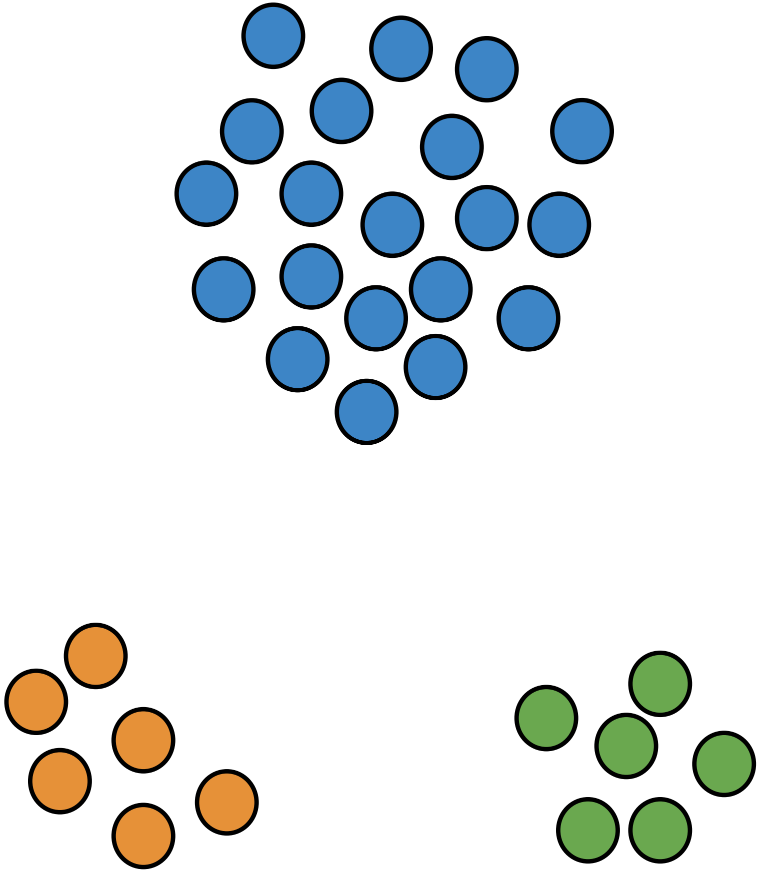




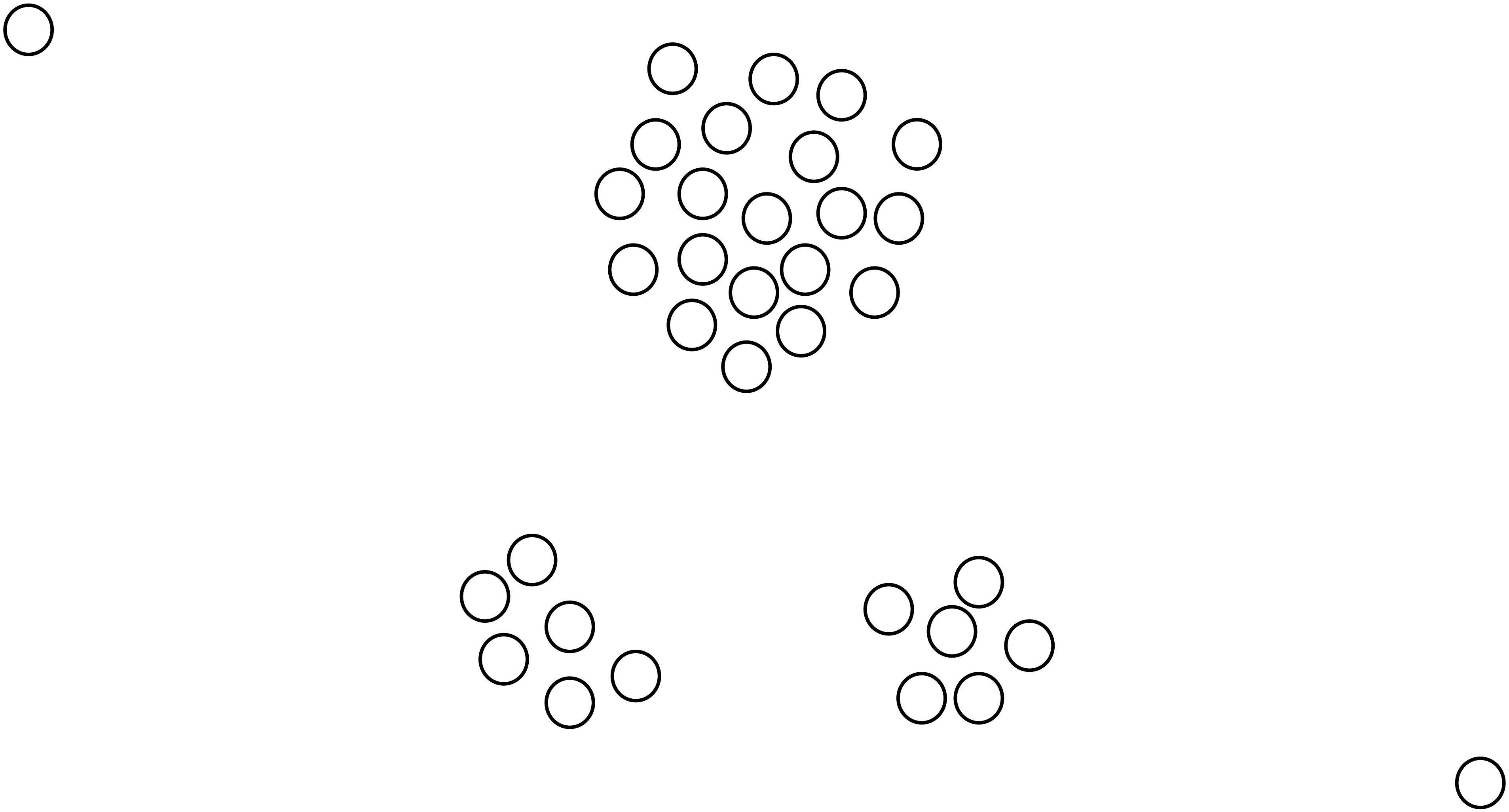


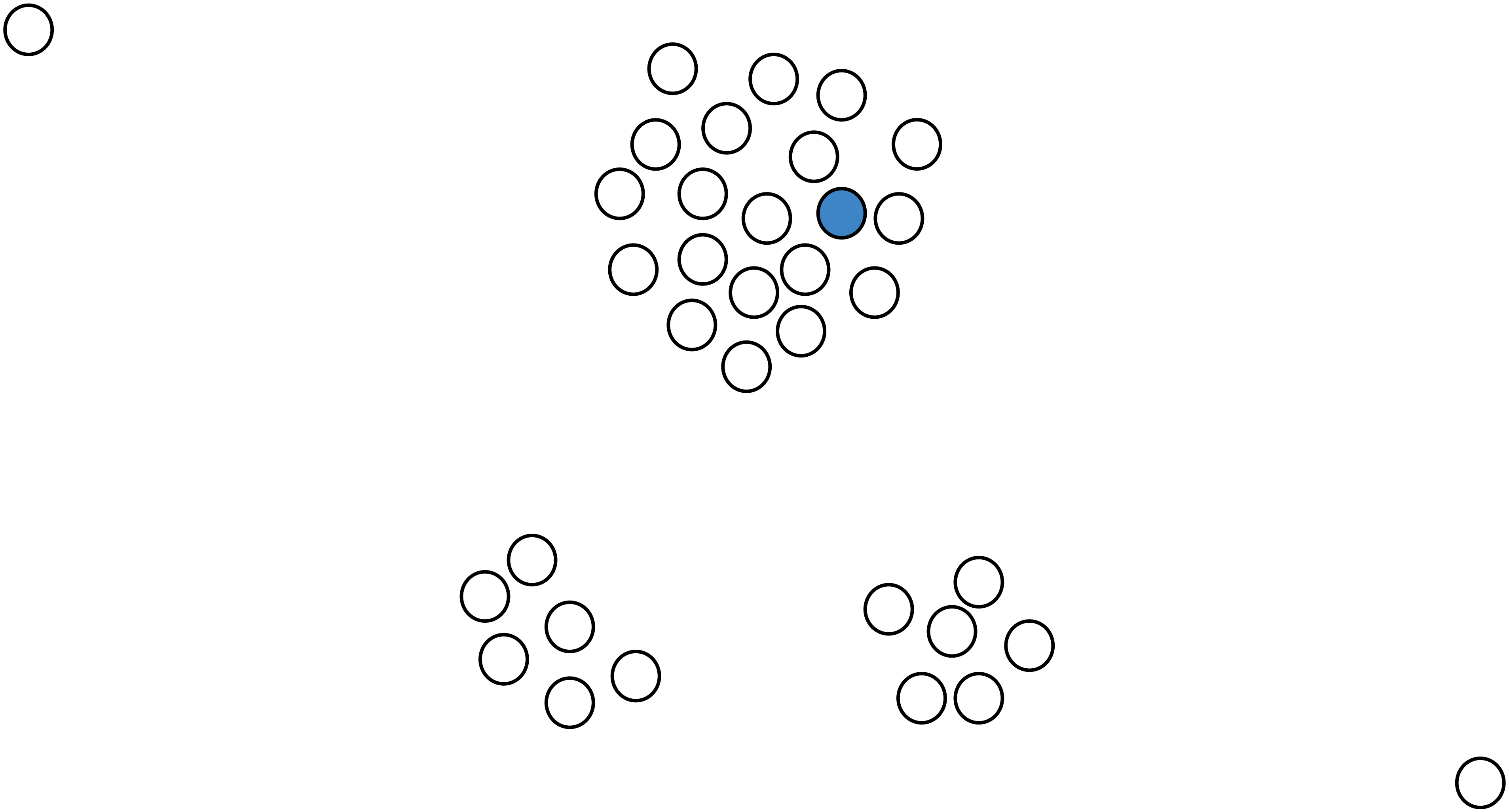


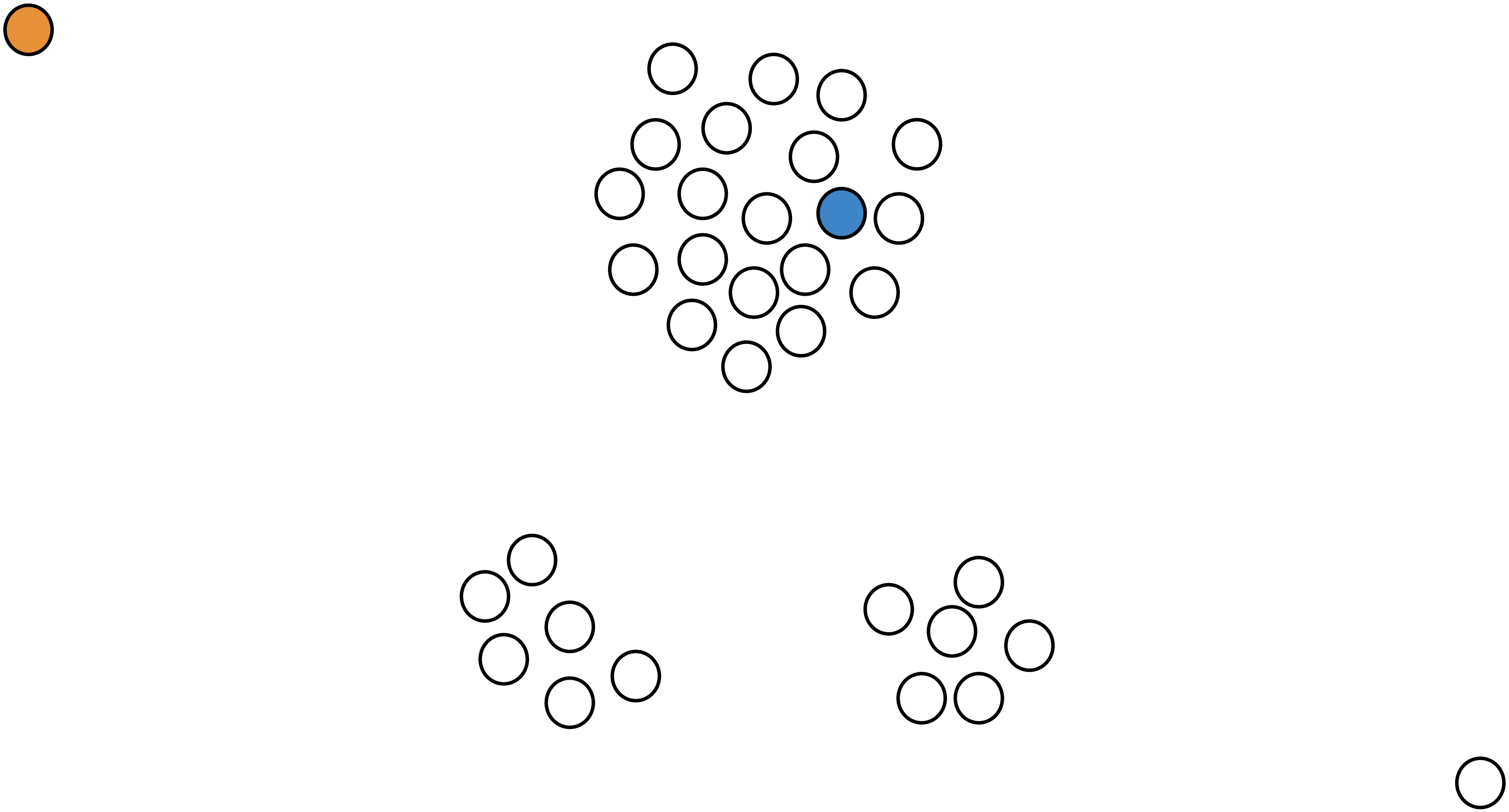
Farthest First Traversal

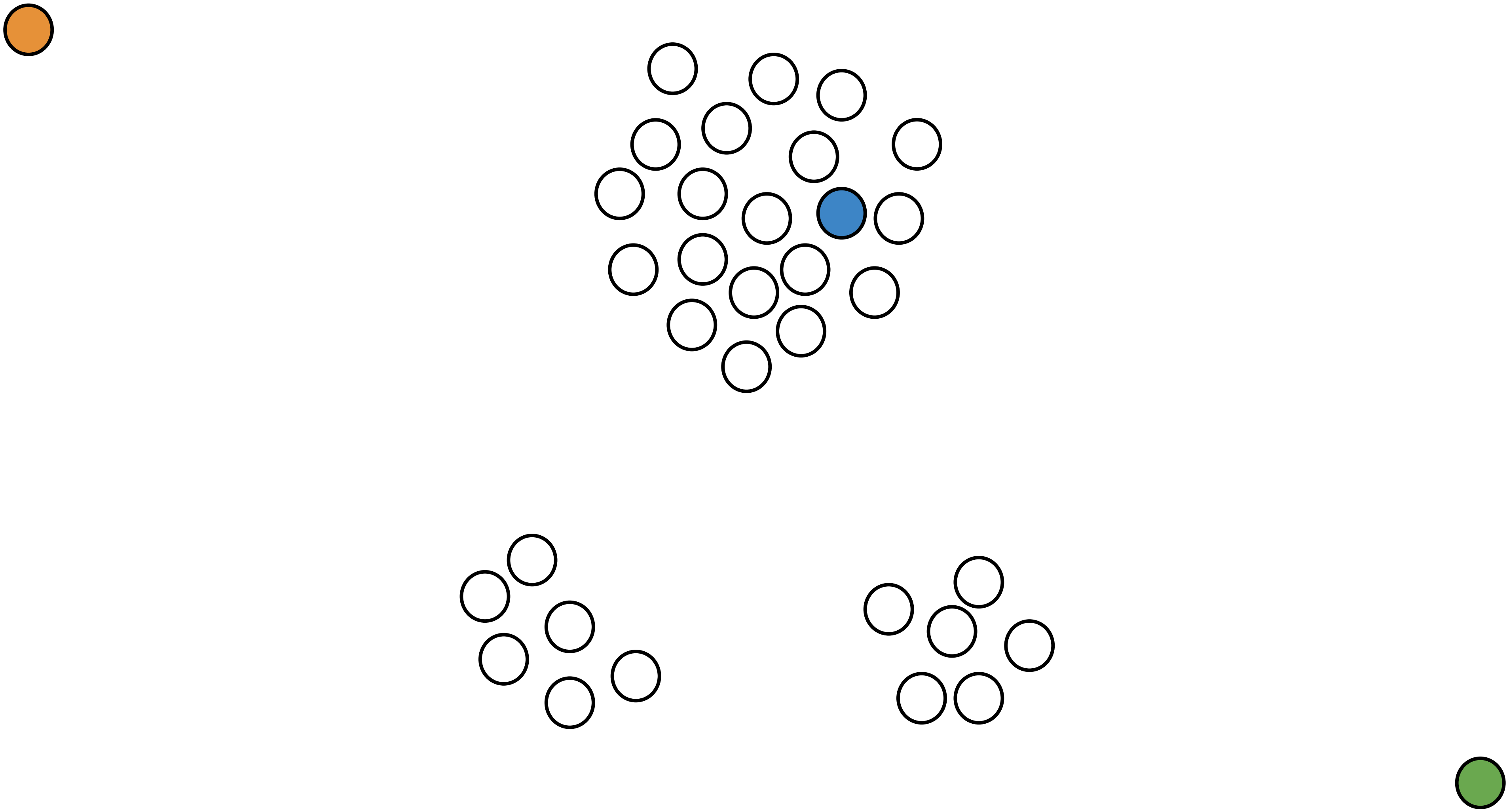


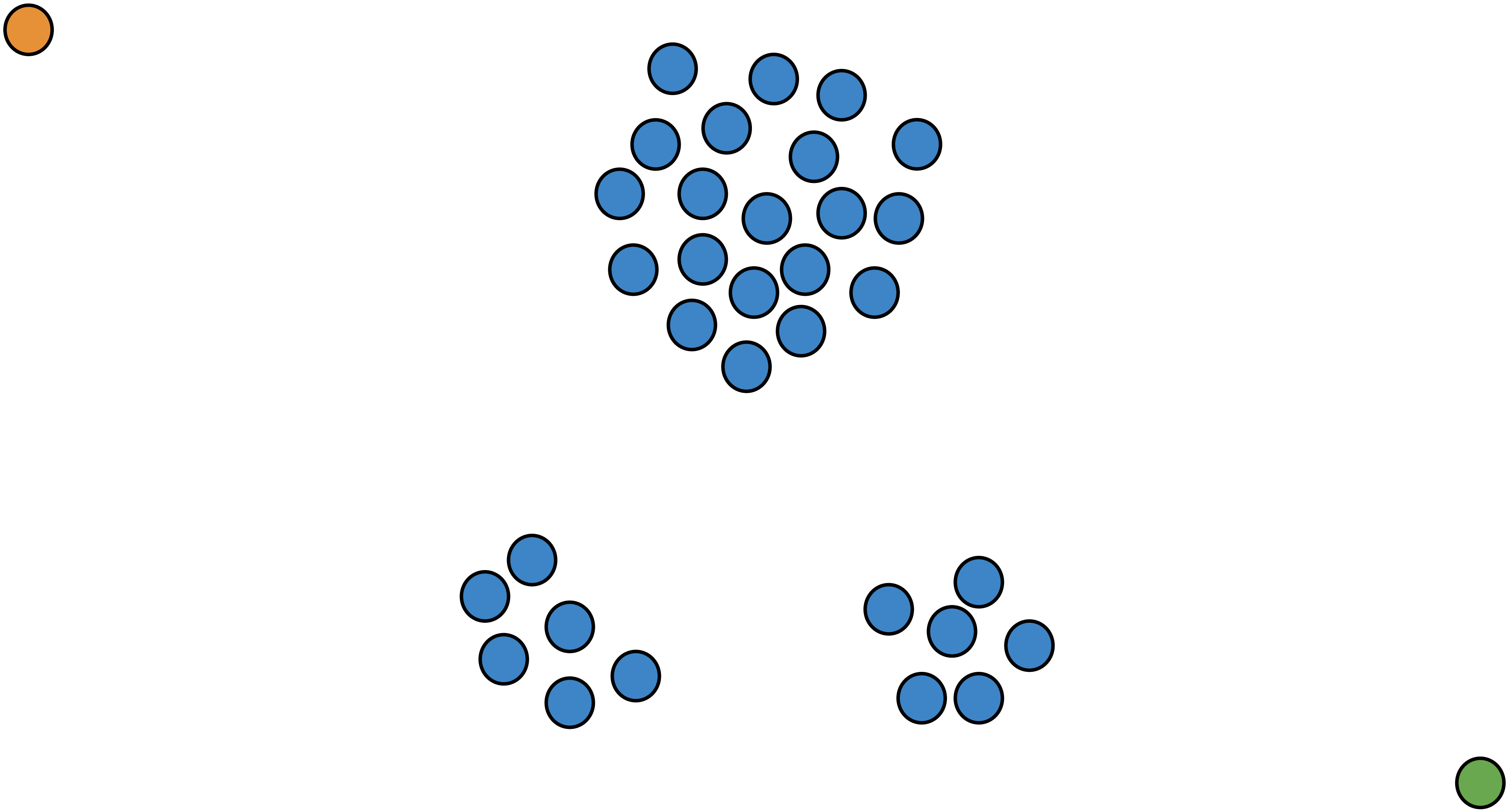
But…



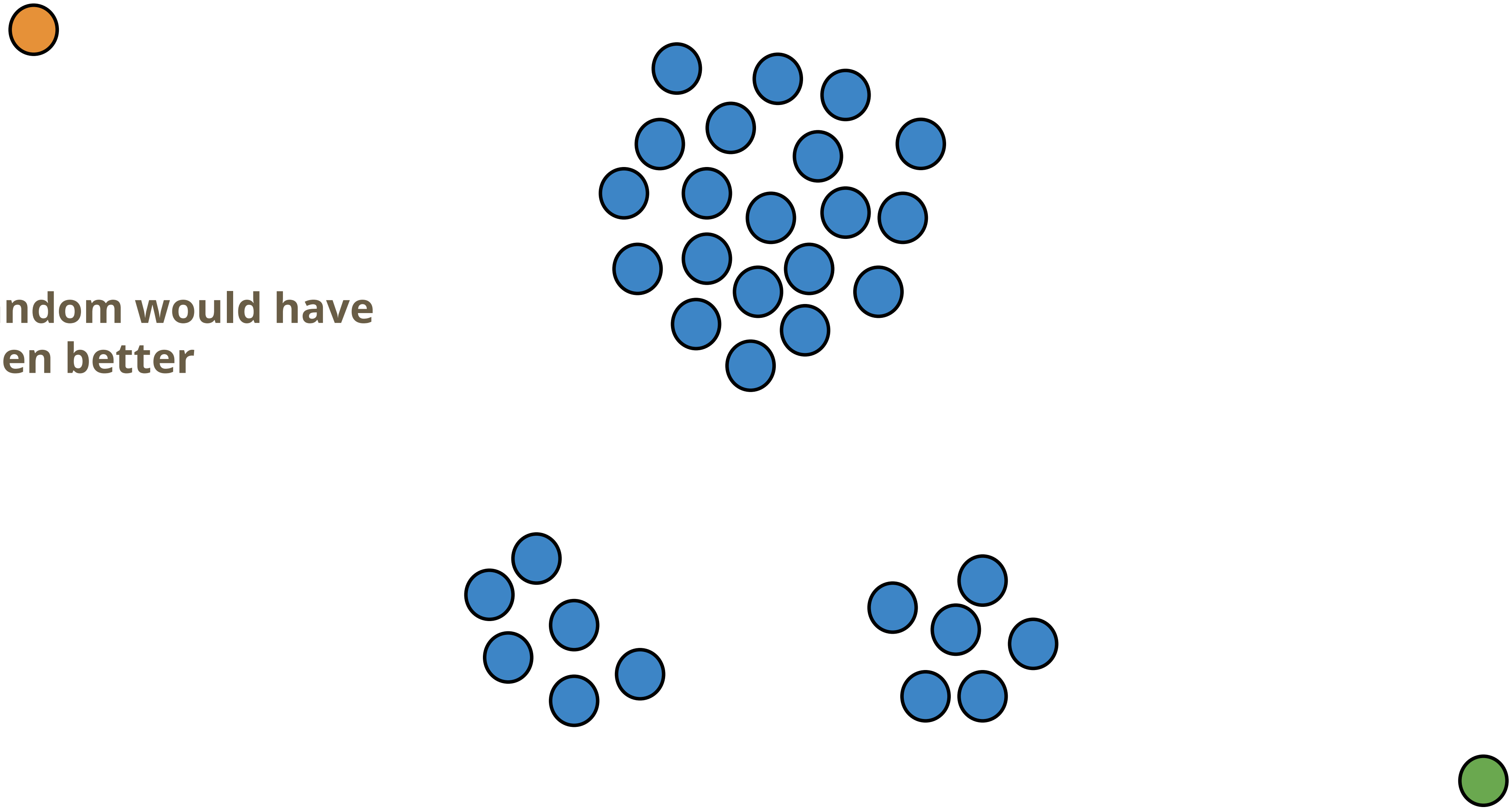








Random would have been better

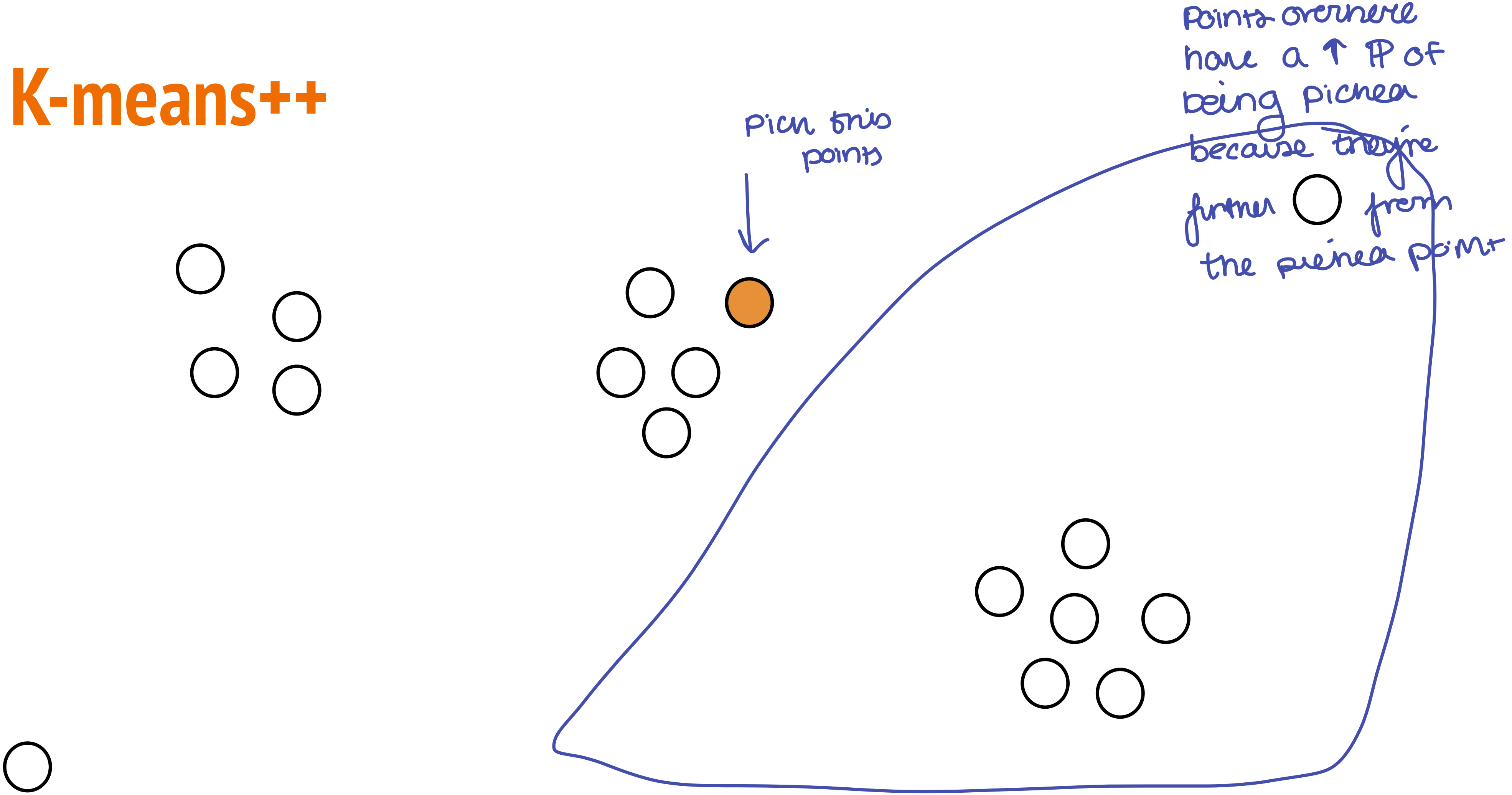


K-means++

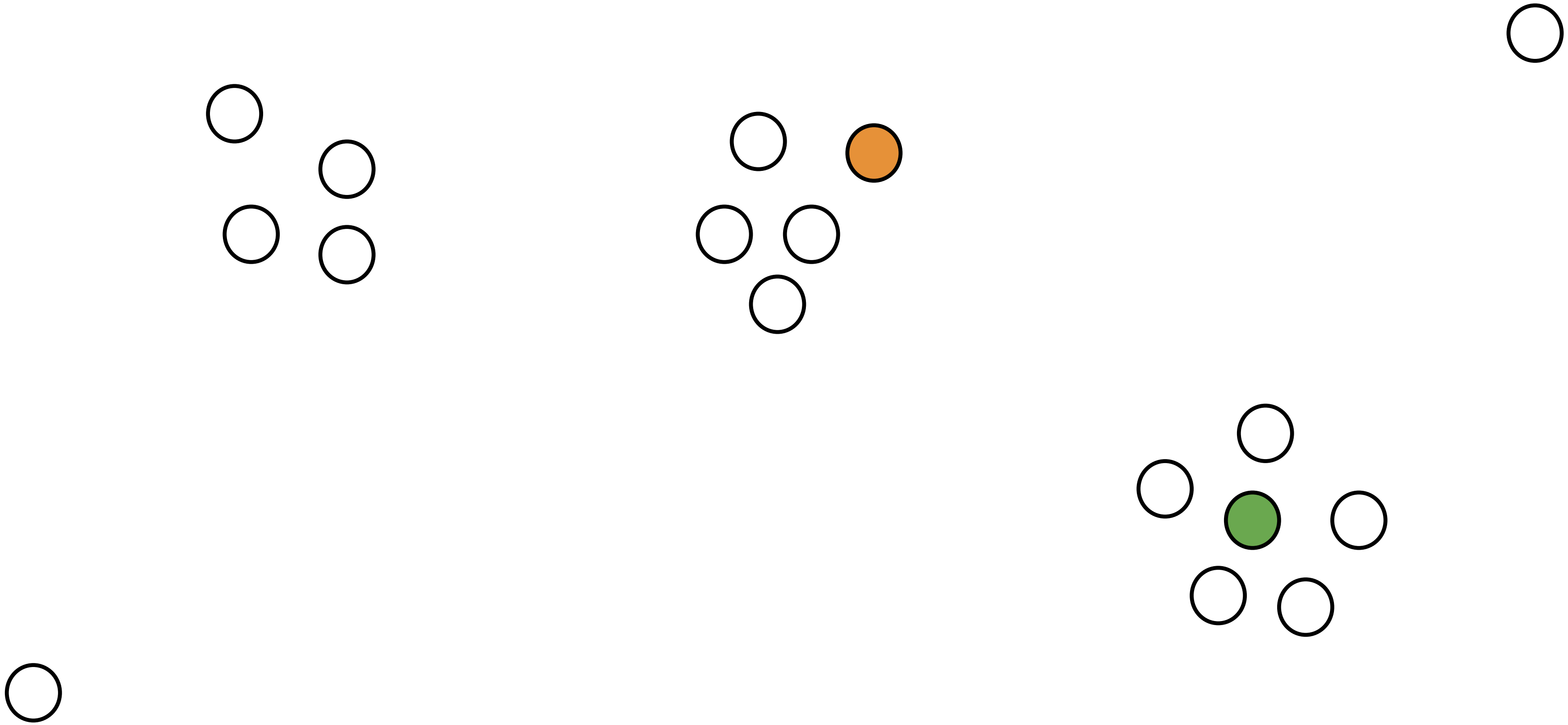
Initialize with a combination of the two methods:

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 $\mathbf{D(x)^2}$

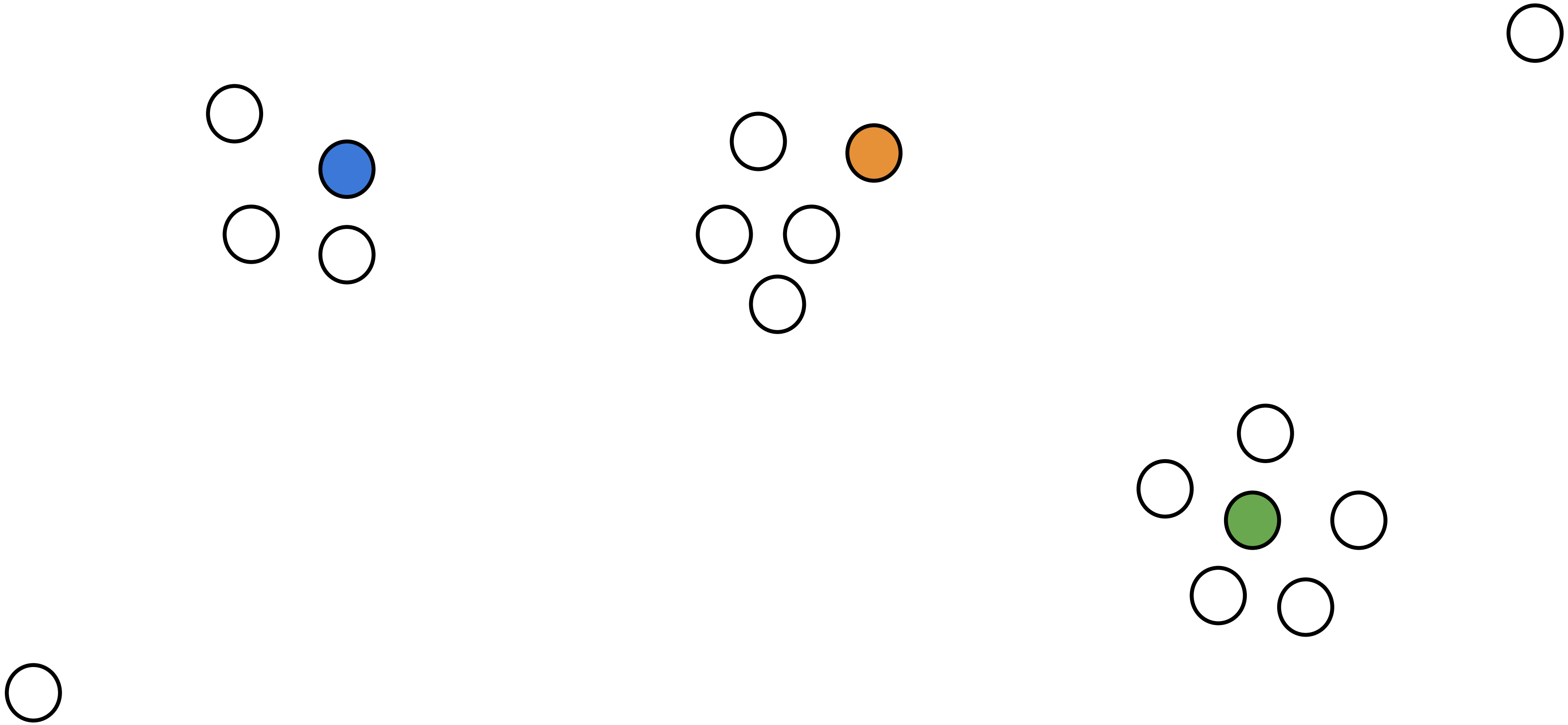
K-means++


Pick this point
Points over here have a ↑ P of being picked because they're further from the picked point

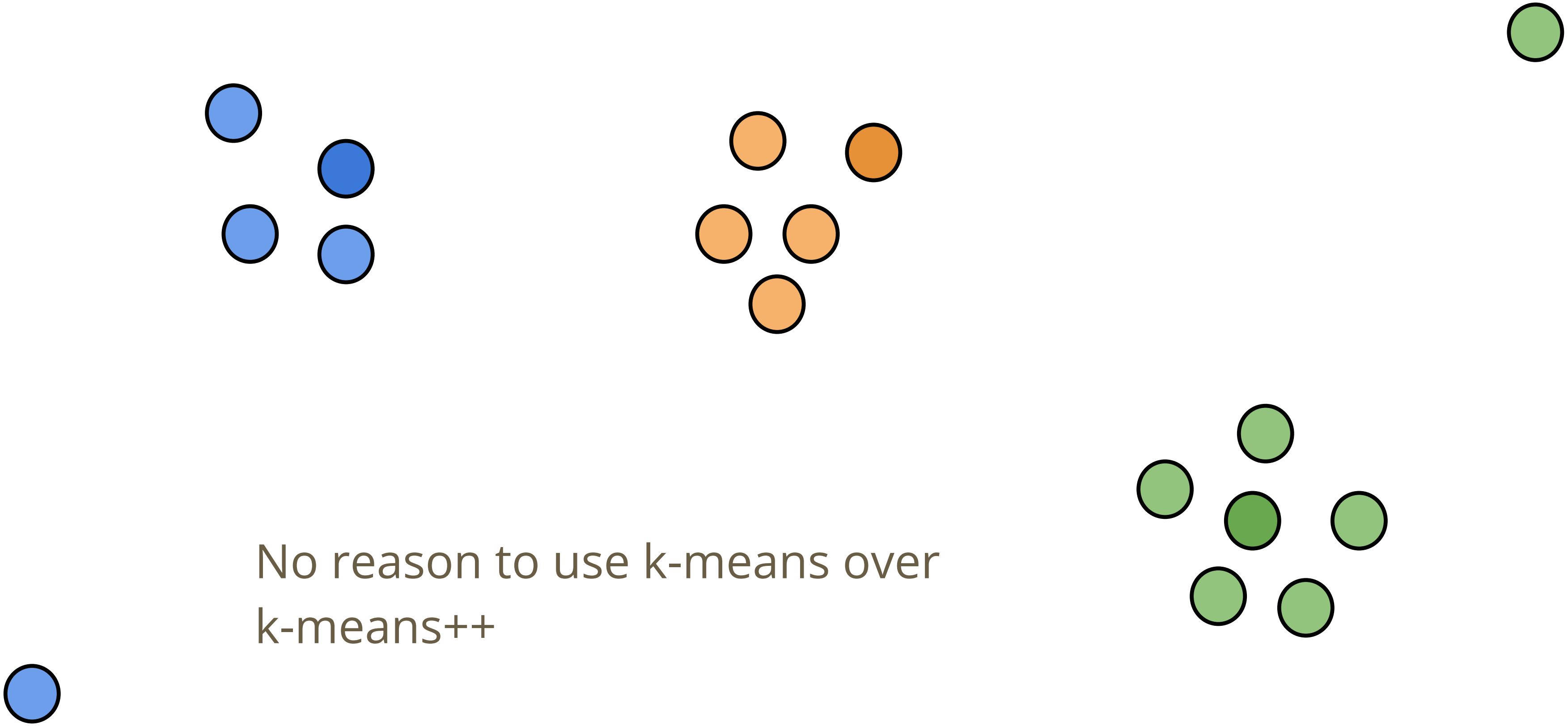
K-means++



K-means++



K-means++



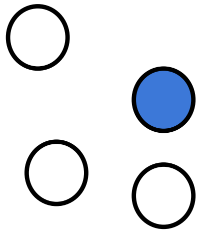
No reason to use k-means over k-means++

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 $\mathbf{D(x)^2}$?

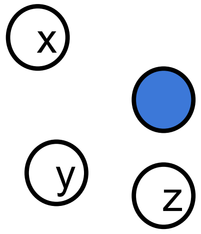
K-means++

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 $\mathbf{D(x)^2}$?



K-means++

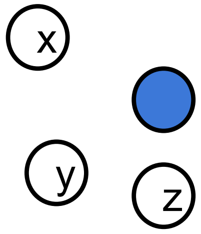
Suppose we are given a black box that will generate a uniform random number between 0 and any **N**. How can we use this black box to select points with probability proportional to $\mathbf{D(x)^2}$?



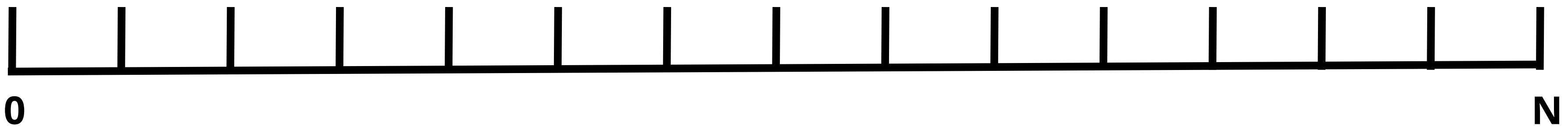
$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

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 **$D(x)^2$**?

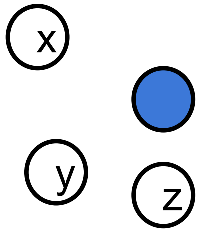


$D(x)^2 = 3^2 = 9$
$D(y)^2 = 2^2 = 4$
$D(z)^2 = 1^2 = 1$

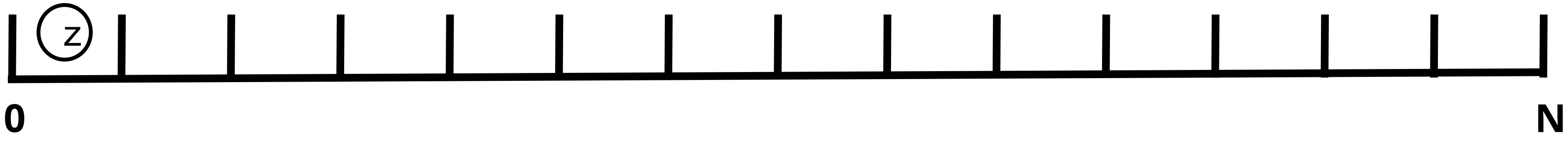


K-means++

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 **$D(x)^2$**?

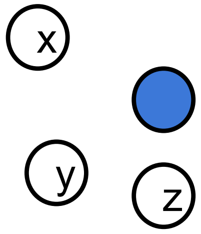


$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

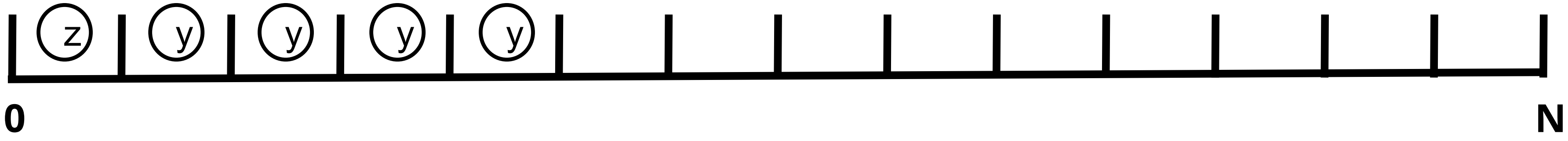


K-means++

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 $\mathbf{D(x)^2}$?

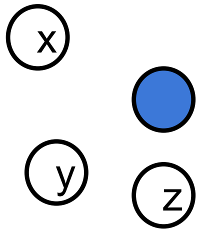


$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$

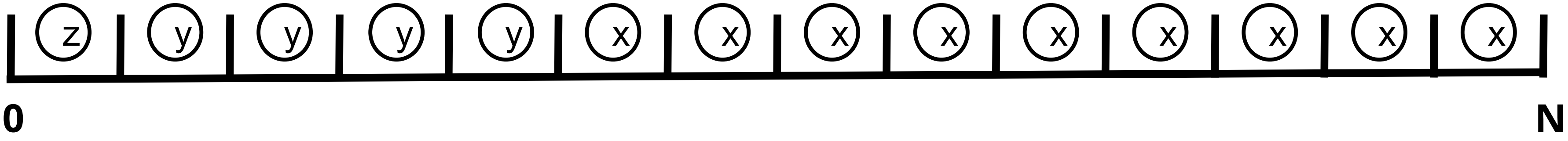


K-means++

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 $\mathbf{D(x)^2}$?



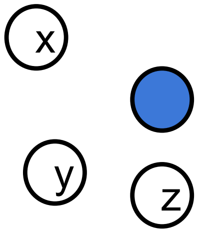
$$D(x)^2 = 3^2 = 9$$
$$D(y)^2 = 2^2 = 4$$
$$D(z)^2 = 1^2 = 1$$



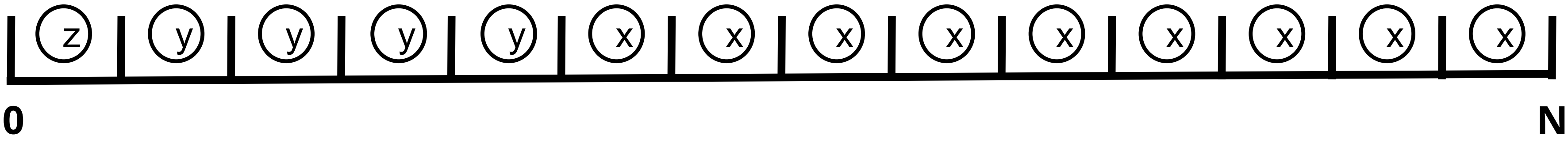
K-means++

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 **$D(x)^2$**?

lay them out in random



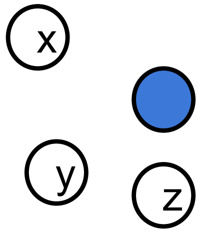
$D(x)^2 = 3^2 = 9$
$D(y)^2 = 2^2 = 4$
$D(z)^2 = 1^2 = 1$



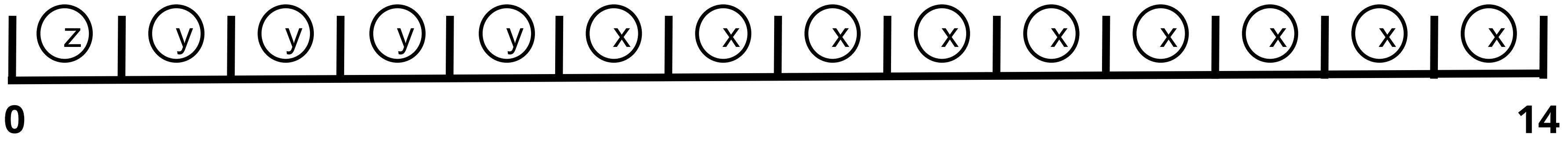
$= D(x)^2 + D(y)^2 + D(z)^2 = 14$

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 $\mathbf{D(x)^2}$?

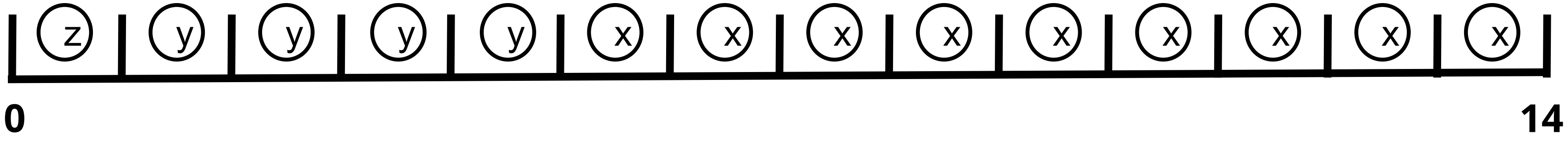


$D(x)^2 = 3^2 = 9$
$D(y)^2 = 2^2 = 4$
$D(z)^2 = 1^2 = 1$



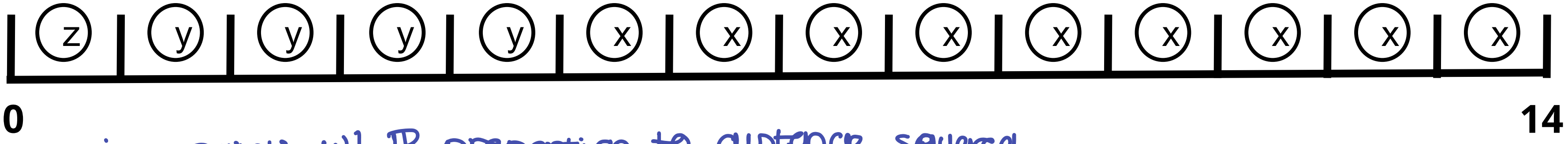
K-means++

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



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) ?



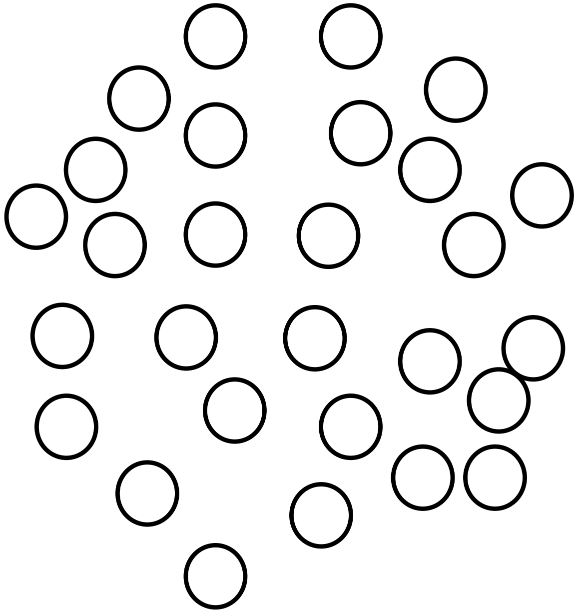
picking points w/ P proportion to distance squared

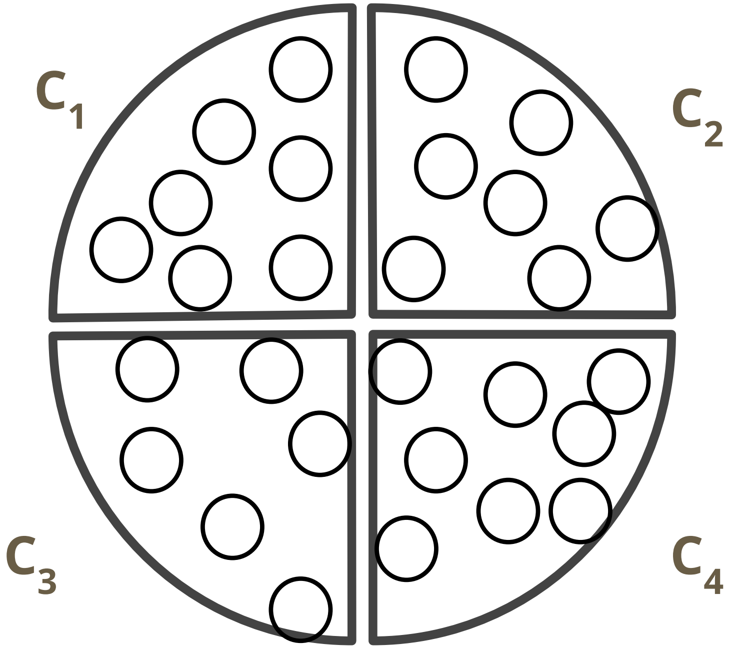
K-means++

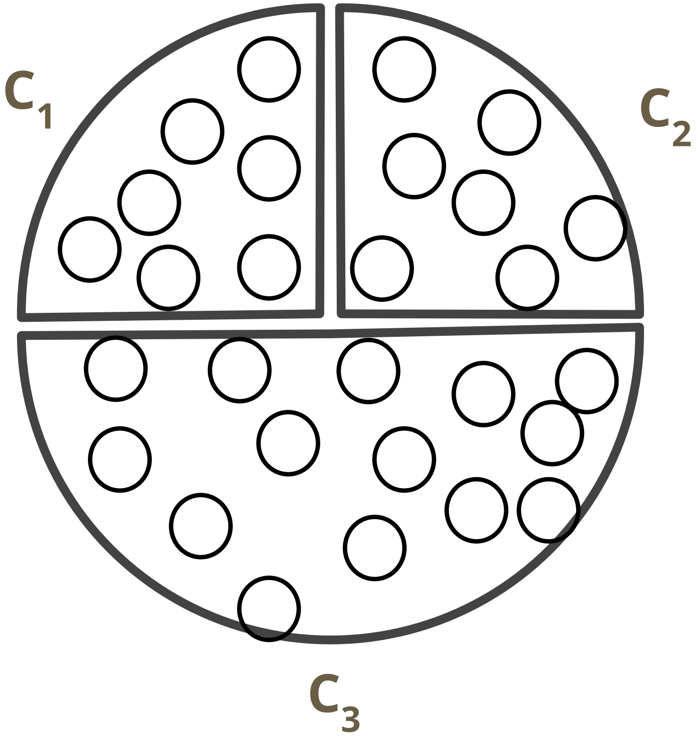
What happens if the black box can only generate numbers between 0 and 1?

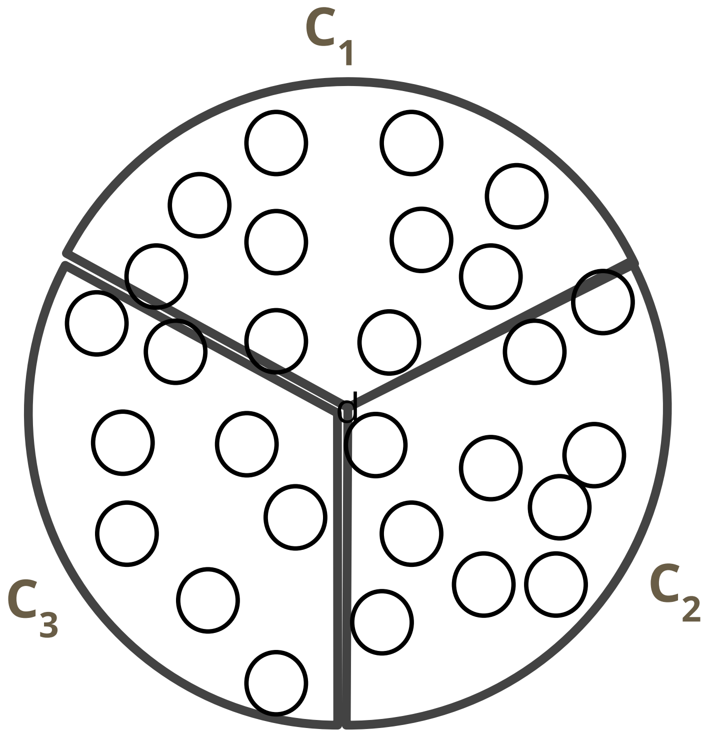
Kmeans Quizz (take 2)

- pick points
- pick centers
-

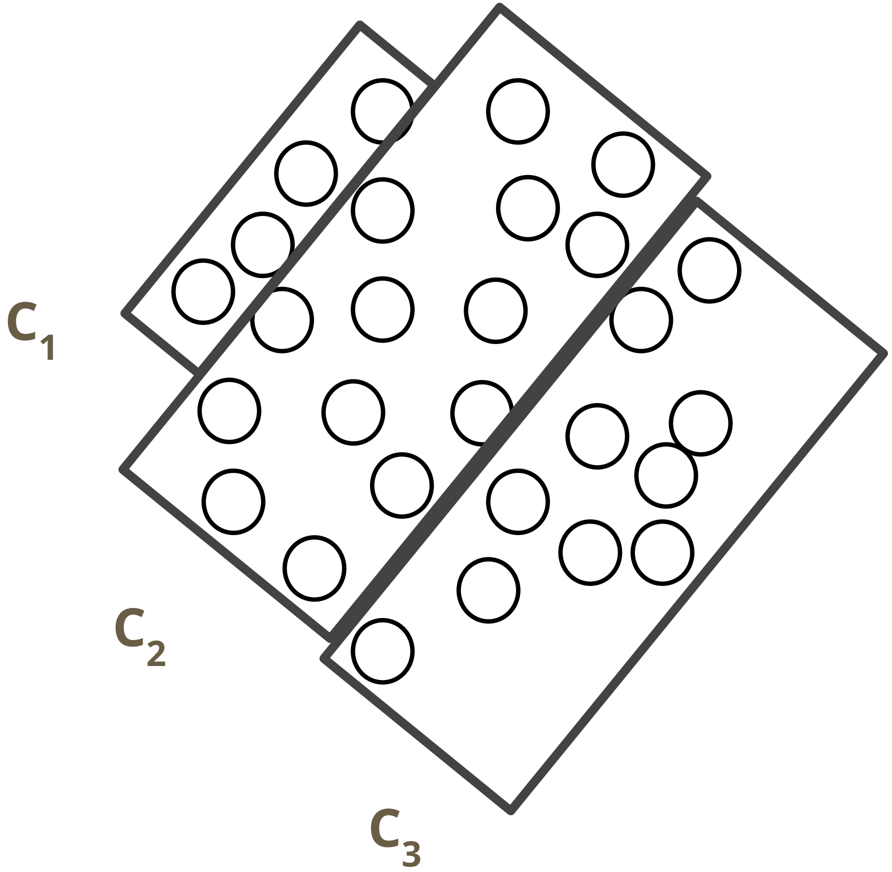


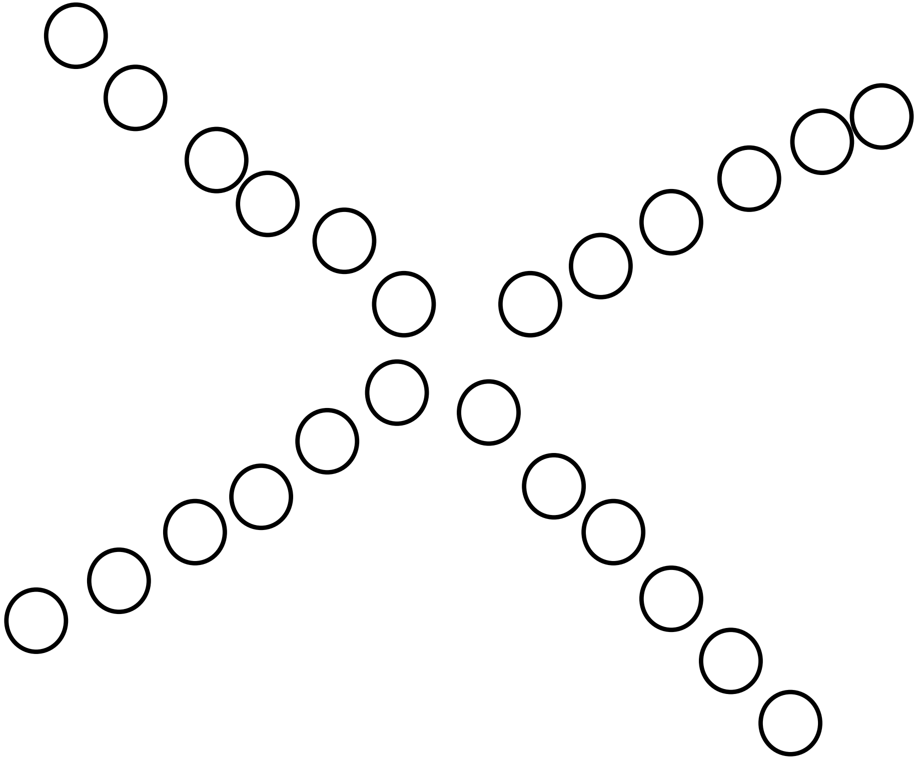

C_1
C_2
C_3
C_4

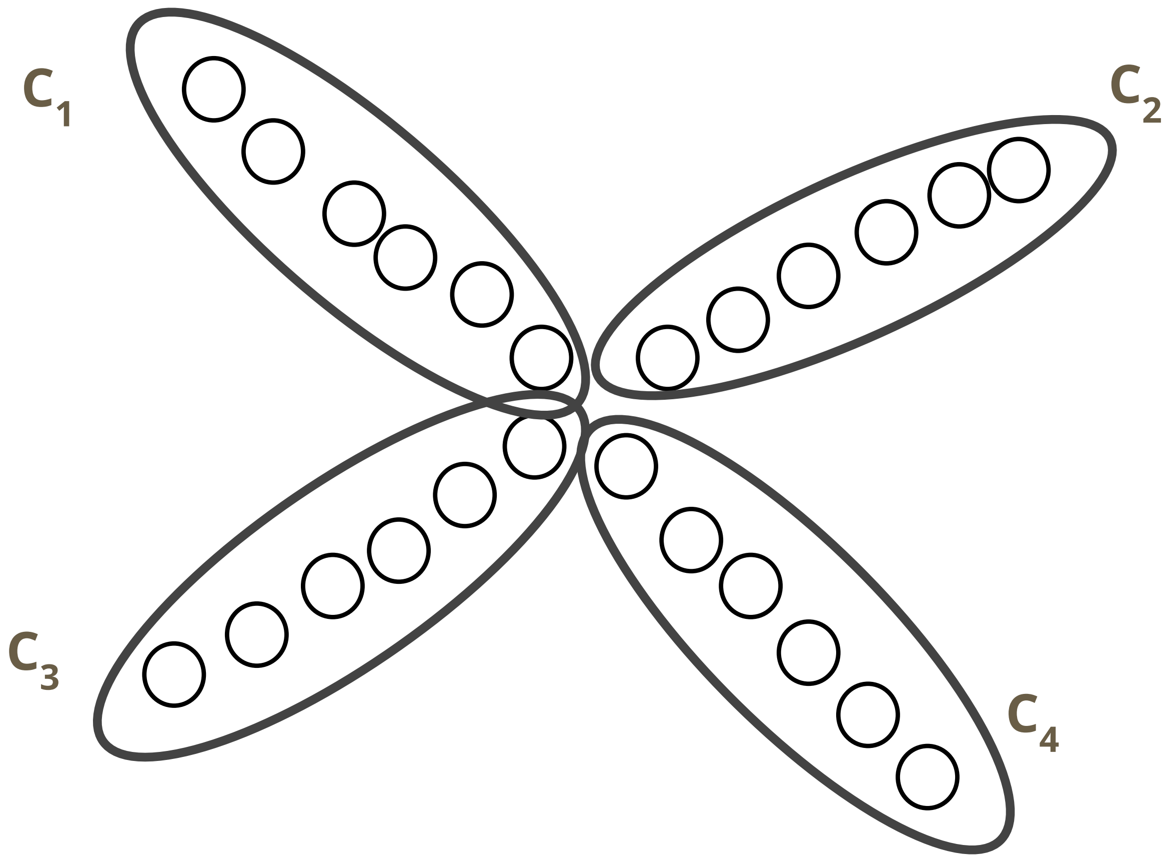

C_1
C_2
C_3



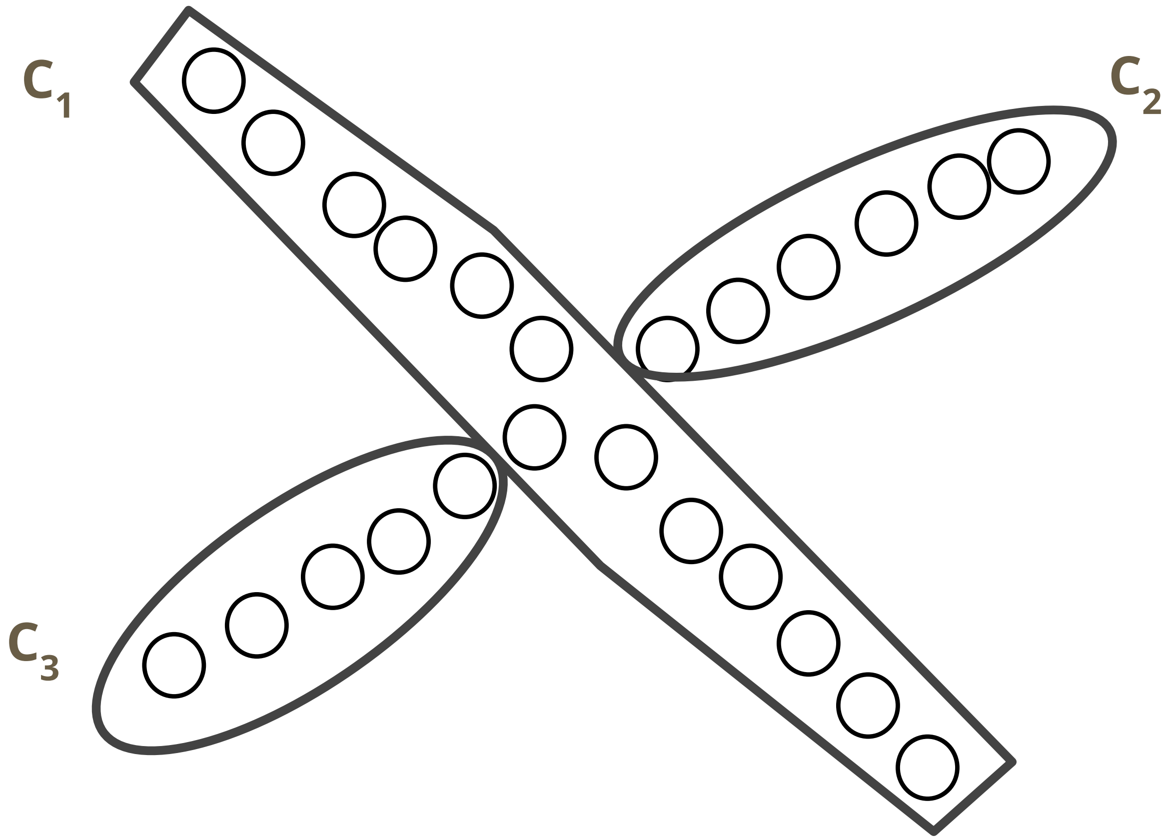
C_1
C_2
C_3

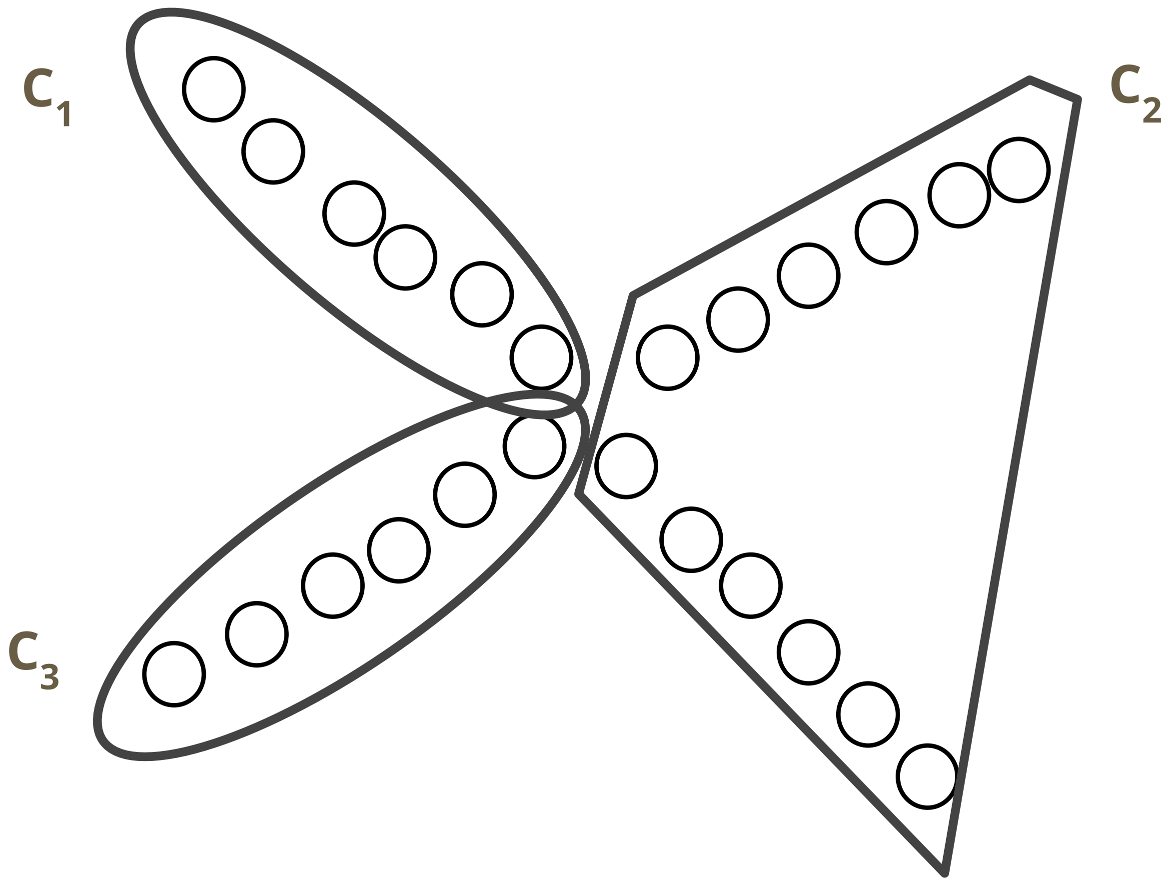

C_1
C_2
C_3



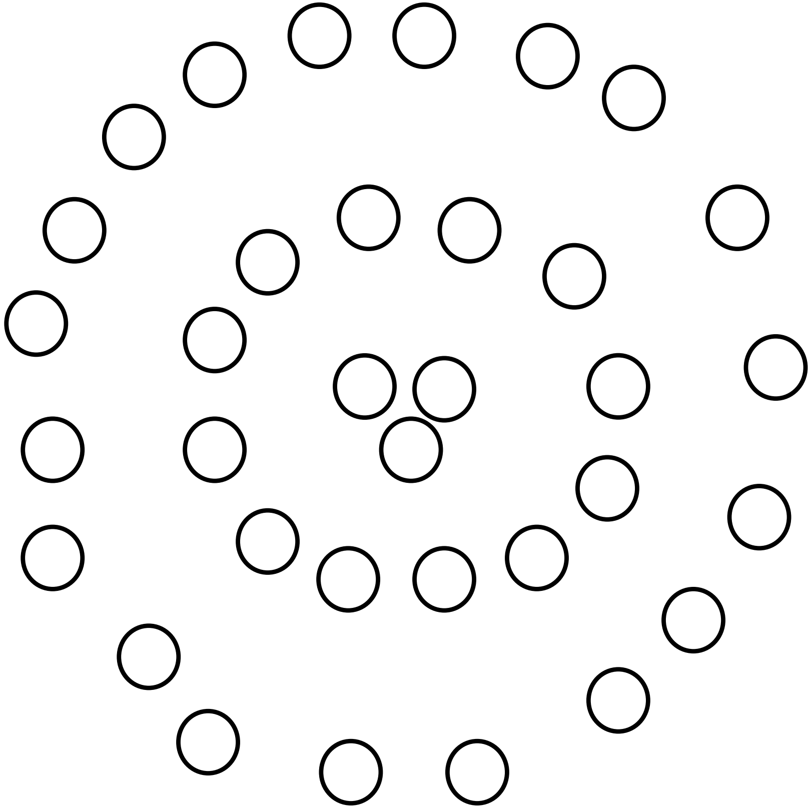


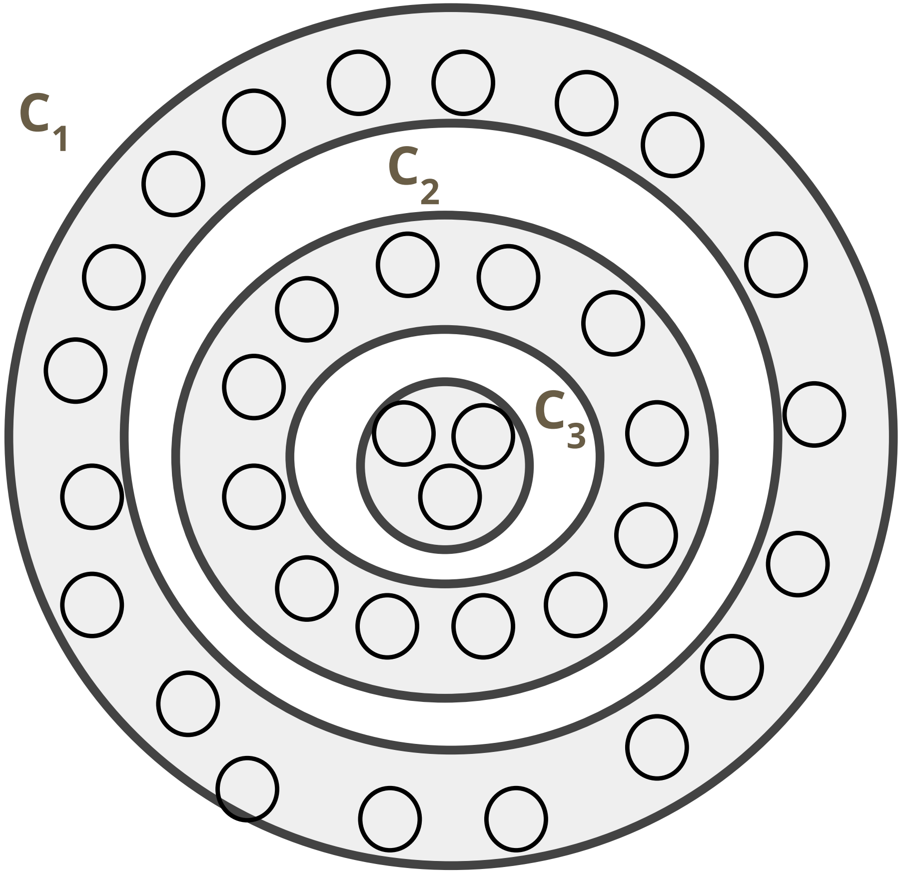
C_1
C_2
C_3
C_4


C_1
C_2
C_3

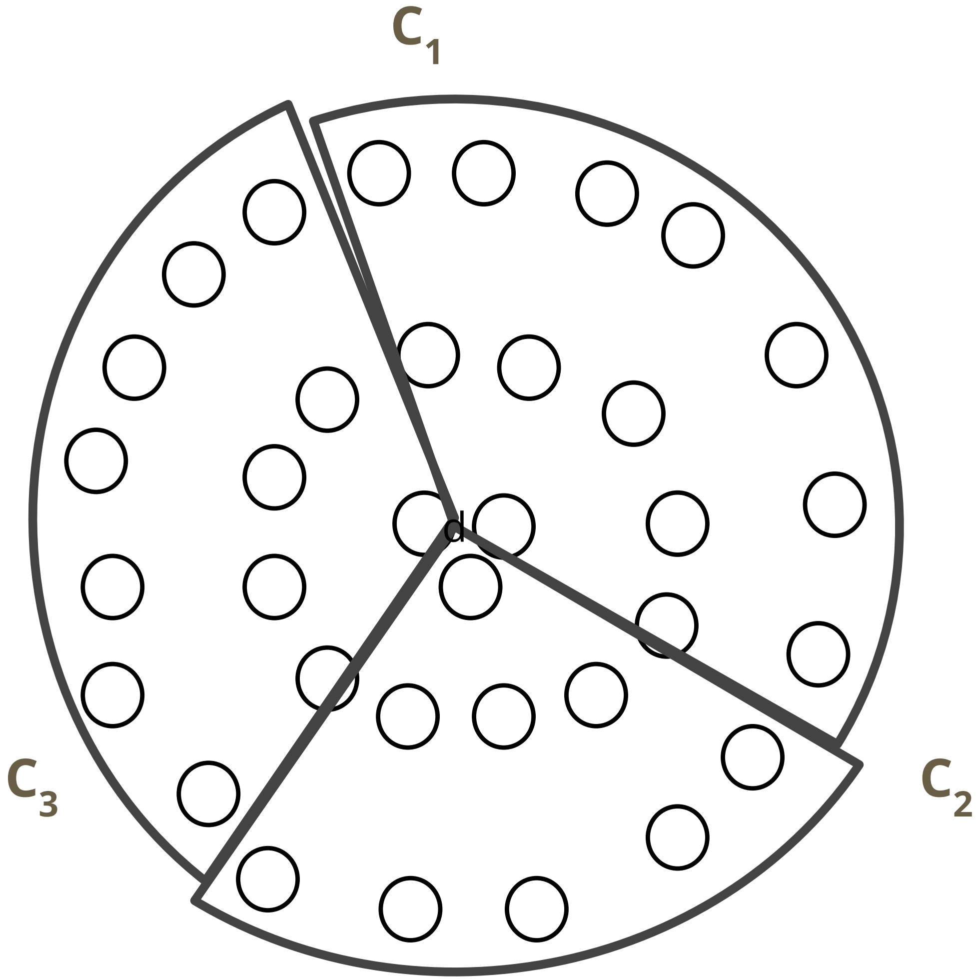


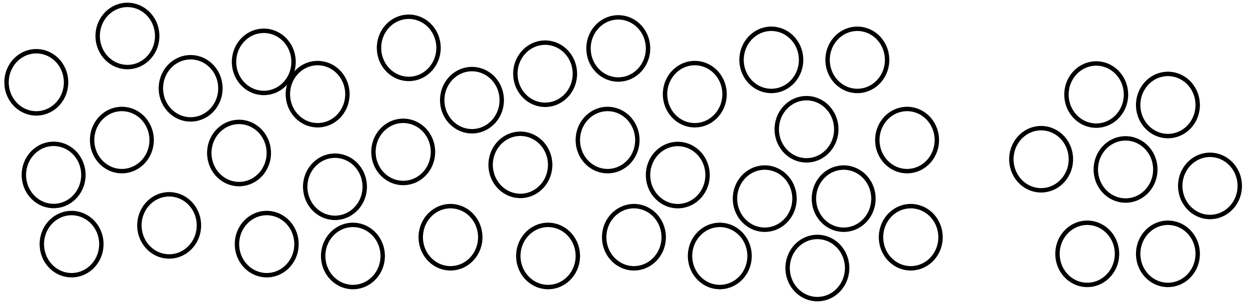
C_1
C_2
C_3

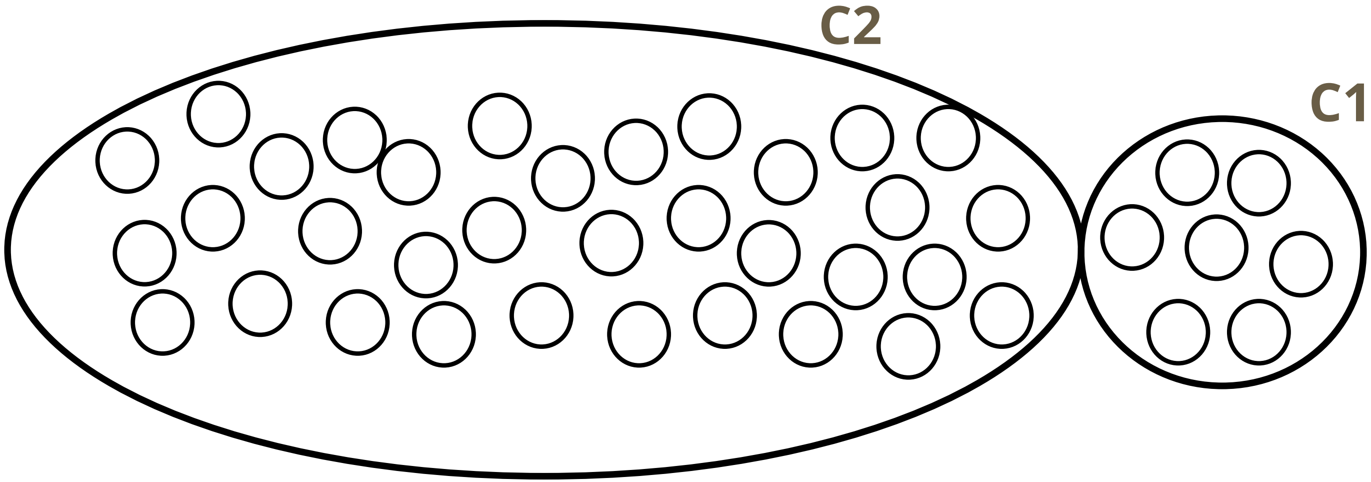




C_1
C_2
C_3

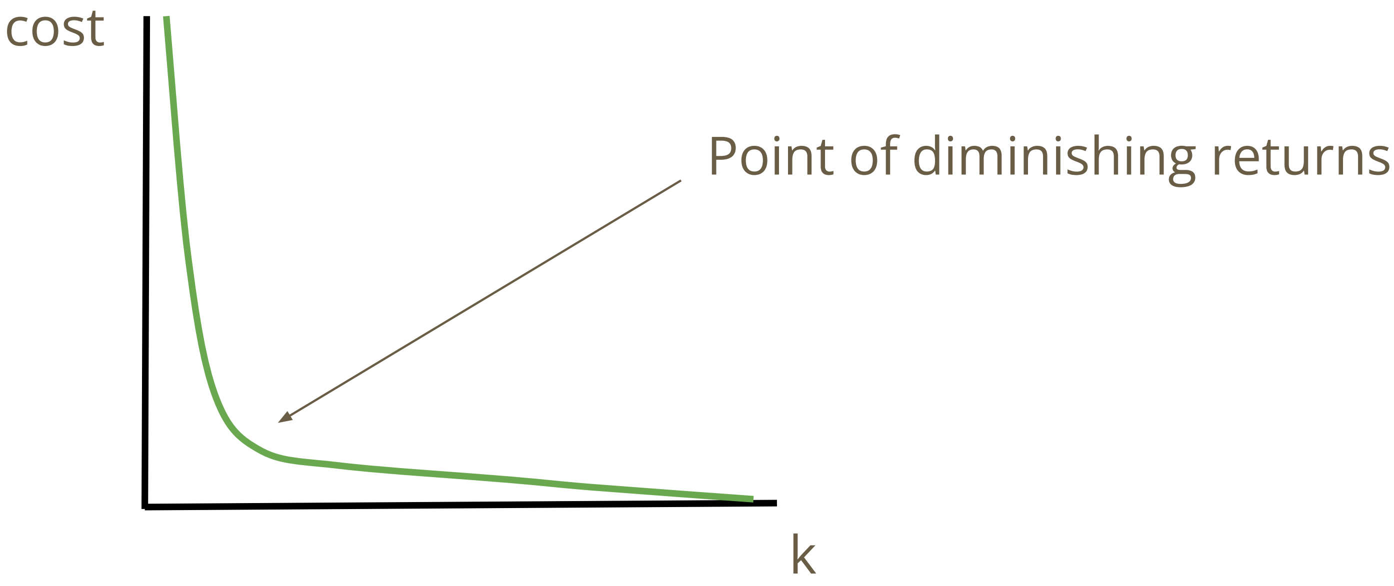

C_1
d
C_2
C_3




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**

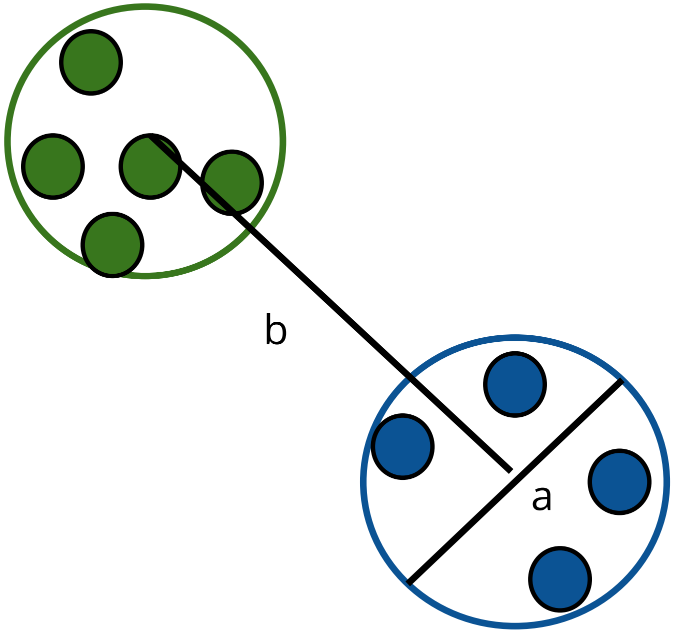
Evaluation

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

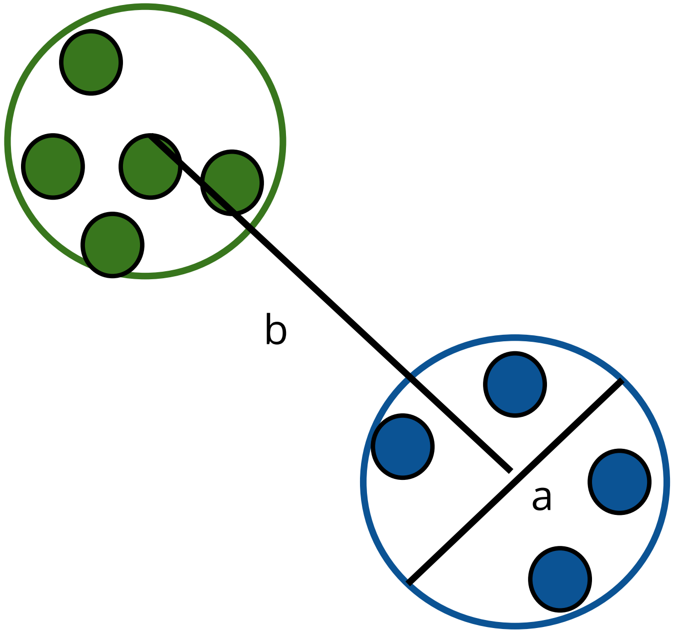
But what about the intra-cluster distance? Are the clusters we created far? How far? Relative to what?

Discuss - 5min

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



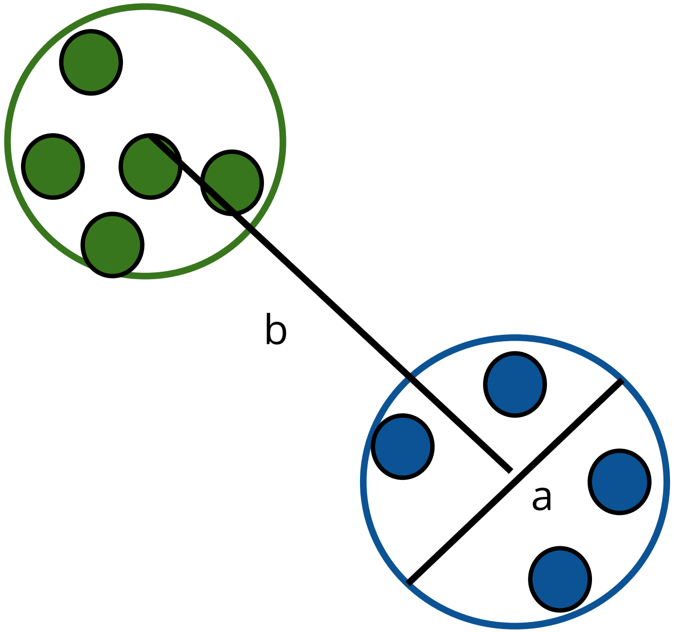
a: average within-cluster distance
b: average intra-cluster distance



a: average within-cluster distance

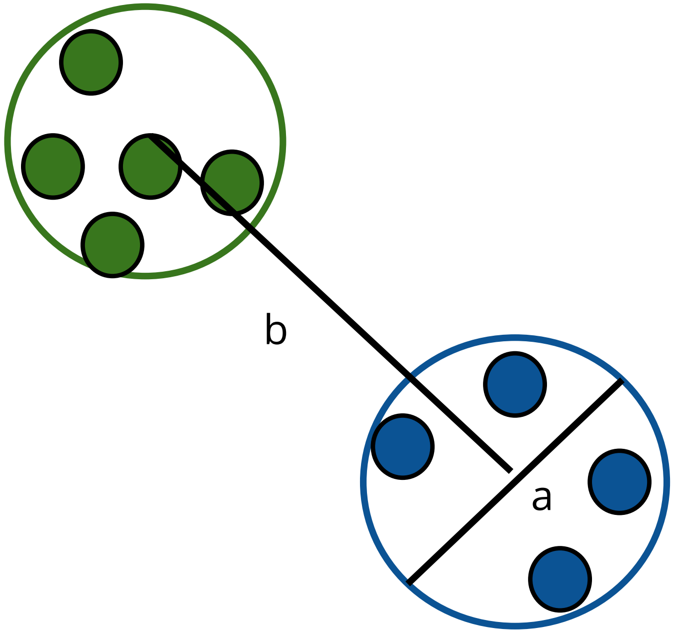
b: average intra-cluster distance

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

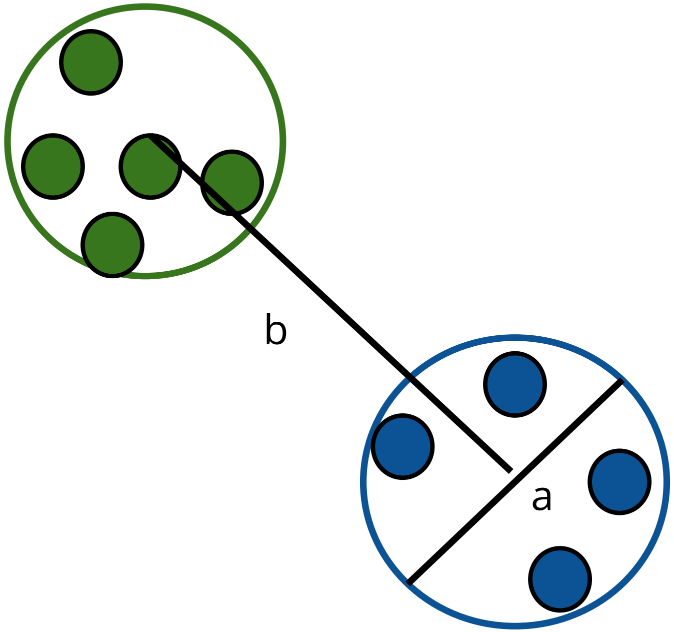


a: average within-cluster distance
b: average intra-cluster distance

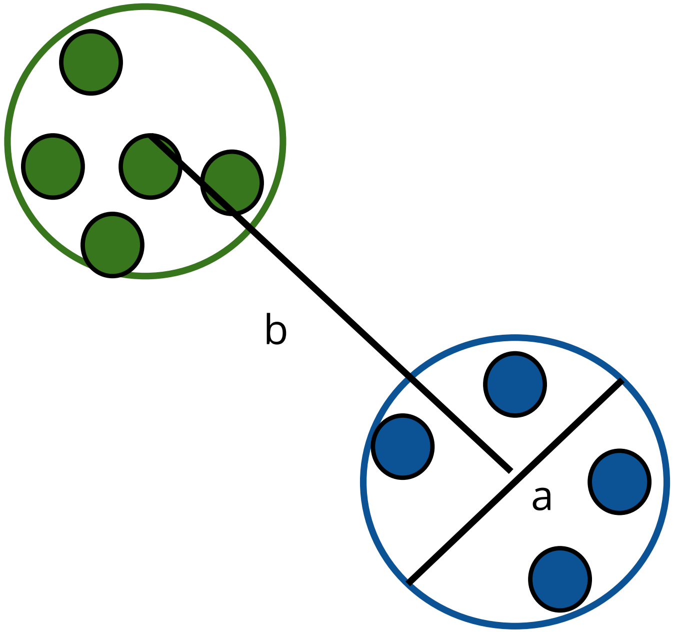
What does it mean for (b - a) to be large?



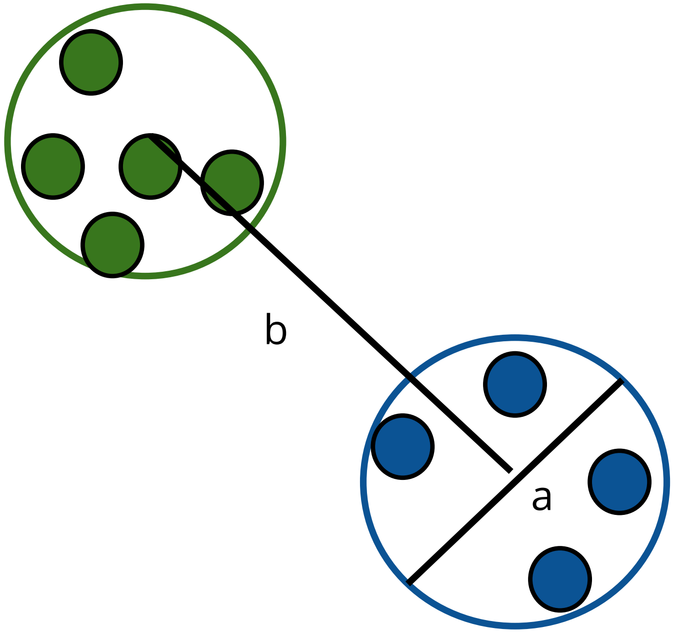
The value of (b-a) doesn't mean much by itself. Can we compare it to something so that the ratio becomes a value between 0 and 1?



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

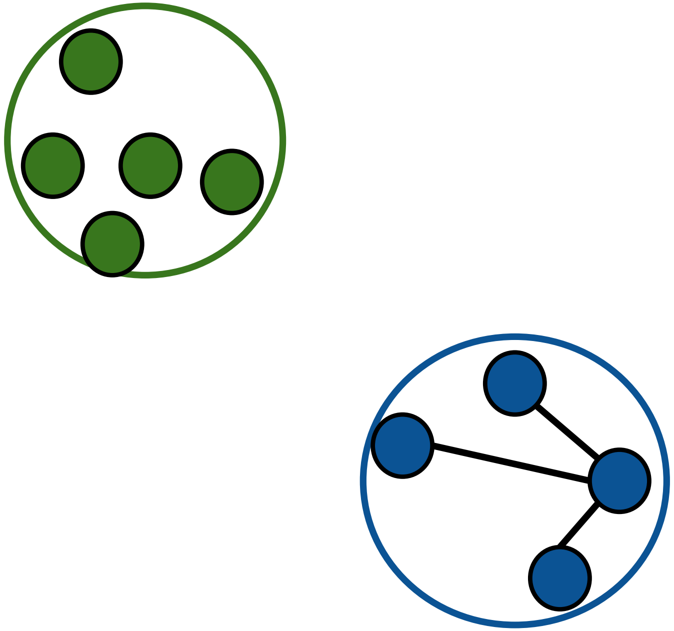


What does it mean for (b - a) / max(a, b) to be close to 1?



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

Silhouette Scores



For each data point i:

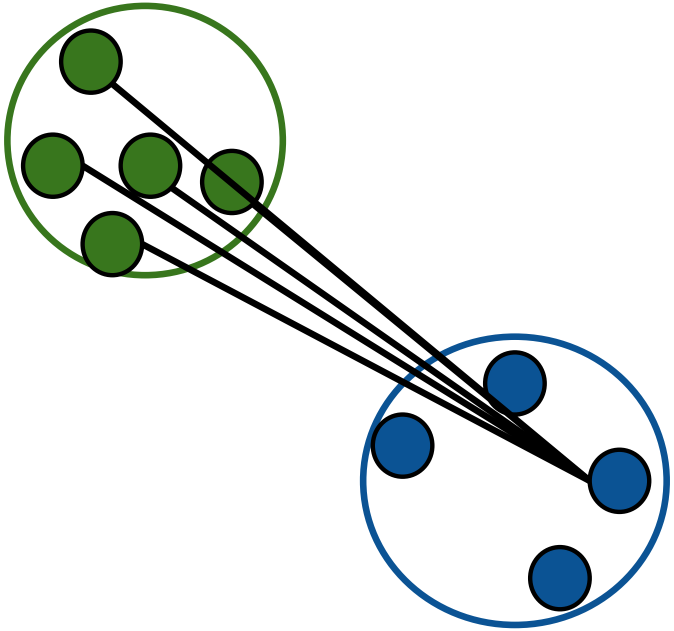
a_i : mean distance from point i to every other point in its cluster

Silhouette Scores

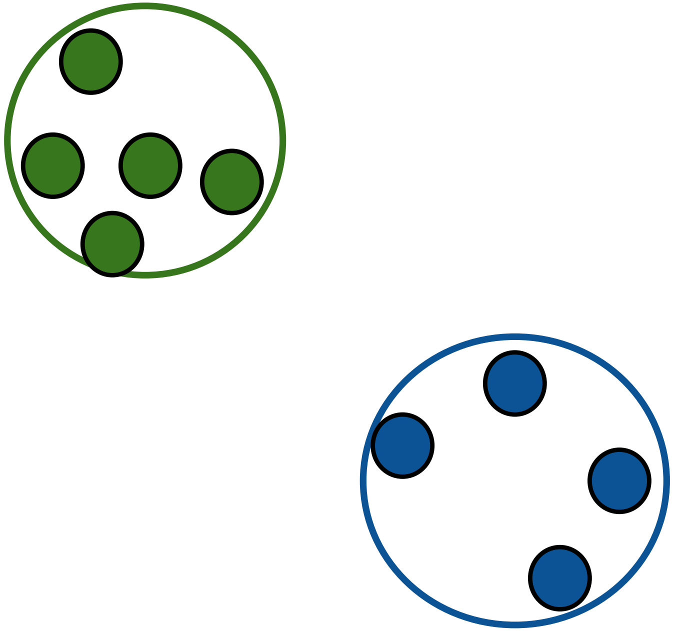
For each data point i:

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



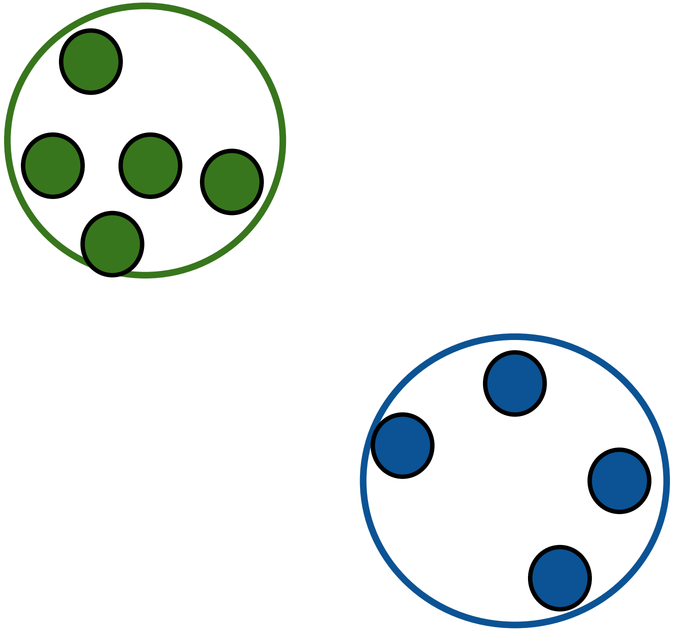
Silhouette Scores



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

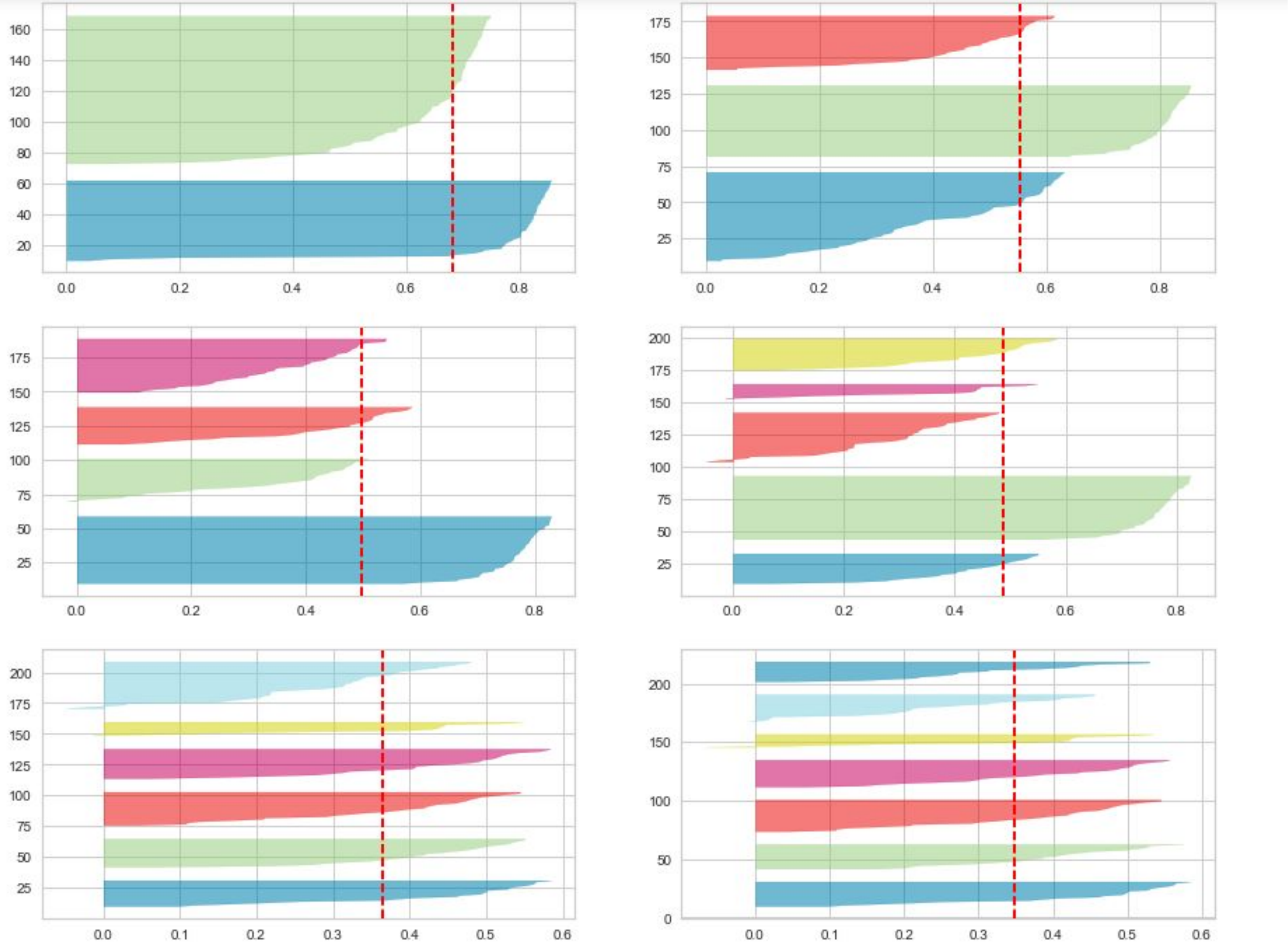


$s_i = (b_i - a_i) / \max(a_i, b_i)$

Silhouette score plot

OR

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



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)