DSC 40A

Theoretical Foundations of Data Science I

Lloyds Algorithm, or k-Means Clustering

1. Randomly initialize the k centroids.
2. Keep centroids fixed. Update groups.

   Assign each point to the nearest centroid.
3. Keep groups fixed. Update centroids.

   Move each centroid to the center of its group.
4. Repeat steps 2 and 3 until done.

In This Video

- Why does k-means clustering work?
- What are some practical considerations when using this algorithm?

Why does k-means clustering work?

$\text{Cost}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k) =$ total squared distance of each data point x_i to its nearest centroid $\boldsymbol{\mu}_j$

better choice of centroids

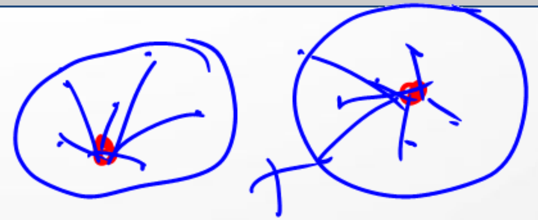
- Argue why updating the groups and centroids according to the algorithm reduces the cost with each iteration.
- With enough iterations, cost will be sufficiently small.

Why does k-means clustering work?

$\text{Cost}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k) =$ total squared distance of each data point x_i to its nearest centroid $\boldsymbol{\mu}_j$

$\text{Cost}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k) = \text{Cost}(\boldsymbol{\mu}_1) + \text{Cost}(\boldsymbol{\mu}_2) + \ldots + \text{Cost}(\boldsymbol{\mu}_k)$ where

$\text{Cost}(\boldsymbol{\mu}_j) =$ total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$



Why does k-means clustering work?

$\text{Cost}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k) = \text{Cost}(\boldsymbol{\mu}_1) + \text{Cost}(\boldsymbol{\mu}_2) + \ldots + \text{Cost}(\boldsymbol{\mu}_k)$ where

$\text{Cost}(\boldsymbol{\mu}_j) =$ total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

1. Randomly initialize the k centroids.

sets initial cost (before the process begins)

Why does k-means clustering work?

Cost($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, ... , $\boldsymbol{\mu}_k$) = Cost($\boldsymbol{\mu}_1$) + Cost($\boldsymbol{\mu}_2$) + … + Cost($\boldsymbol{\mu}_k$) where

Cost($\boldsymbol{\mu}_j$) = total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

2. Fix the centroids. Update the groups. ← consider an arbitrary iteration

Certainly Cost($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, ... , $\boldsymbol{\mu}_k$) decreases in this step because assigning each point to the **closest** centroid is best.

Why does k-means clustering work?

$\text{Cost}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k) = \text{Cost}(\boldsymbol{\mu}_1) + \text{Cost}(\boldsymbol{\mu}_2) + \ldots + \text{Cost}(\boldsymbol{\mu}_k)$ where

$\text{Cost}(\boldsymbol{\mu}_j) =$ total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

3. Fix the groups. Update the centroids.

consider an arbitrary iteration

Argue that $\text{Cost}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k)$ decreases in this step because for each group j, $\text{Cost}(\boldsymbol{\mu}_j)$ is minimized when we update the centroid.

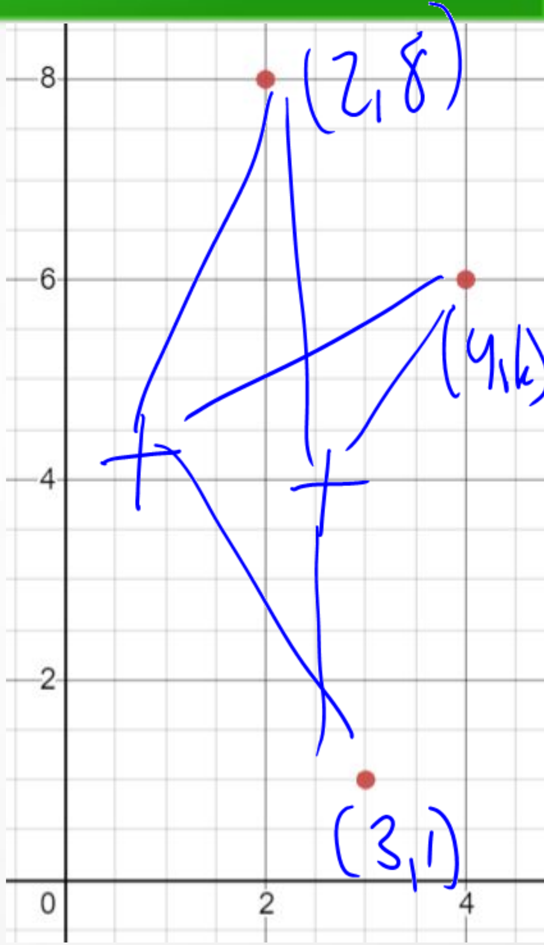
Why does k-means clustering work?

$\text{Cost}(\boldsymbol{\mu}_j) =$ total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

Example: group j contains (4, 6), (2, 8), (3,1)

How to place centroid $\boldsymbol{\mu}_j = (c_1, c_2)$ to minimize cost?

$$\text{Cost}(\mu_j) = \left(\sqrt{(4-c_1)^2+(6-c_2)^2}\right)^2 + \left(\sqrt{(2-c_1)^2+(8-c_2)^2}\right)^2 + \left(\sqrt{(3-c_1)^2+(1-c_2)^2}\right)^2$$

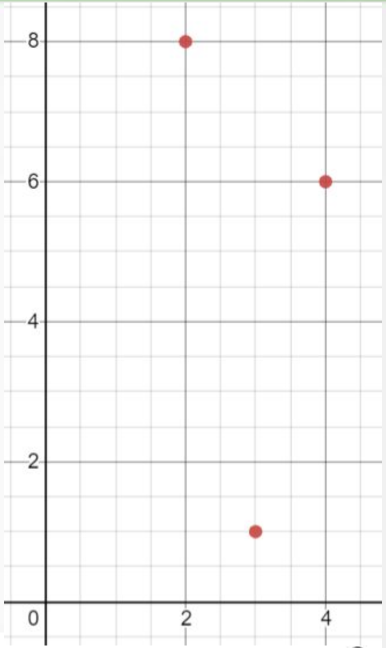


Why does k-means clustering work?

Cost($\boldsymbol{\mu}_j$) = total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

Example: group j contains (4, 6), (2, 8), (3,1)

How to place centroid $\boldsymbol{\mu}_j$ = (c_1, c_2) to minimize cost?



$$\text{Cost}\,(\mu_j) = \left(\sqrt{(4-c_1)^2+(6-c_2)^2}\right)^2 + \left(\sqrt{(2-c_1)^2+(8-c_2)^2}\right)^2 + \left(\sqrt{(3-c_1)^2+(1-c_2)^2}\right)^2$$

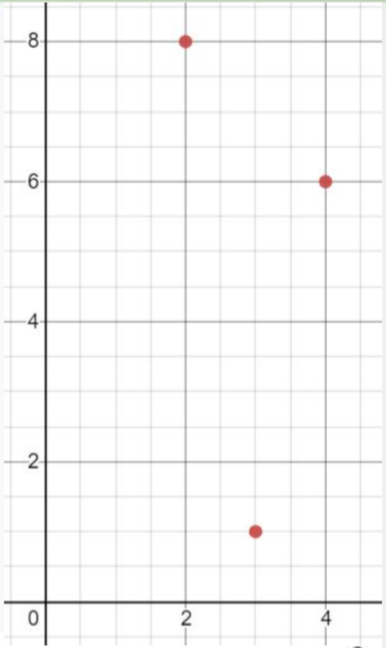
$$= (4-c_1)^2 + \cancel{(6-c_2)^2} + (2-c_1)^2 + \cancel{(8-c_2)^2} + (3-c_1)^2 + \cancel{(1-c_2)^2}$$

fcn of c_1, c_2

$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_1} = 2(4-c_1)^1\cdot(-1) + 2(2-c_1)\cdot(-1) + 2(3-c_1)\cdot(-1) = 2(c_1-4) + 2(c_1-2) + 2(c_1-3)$$

Why does k-means clustering work?

Cost($\boldsymbol{\mu}_j$) = total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$



Example: group j contains (4, 6), (2, 8), (3,1)

How to place centroid $\boldsymbol{\mu}_j$ = (c_1, c_2) to minimize cost?

$$\begin{aligned}\text{Cost}\ (\mu_j) &= \left(\sqrt{(4-c_1)^2+(6-c_2)^2}\right)^2+\left(\sqrt{(2-c_1)^2+(8-c_2)^2}\right)^2+\left(\sqrt{(3-c_1)^2+(1-c_2)^2}\right)^2\\ &= (4-c_1)^2+(6-c_2)^2+(2-c_1)^2+(8-c_2)^2+(3-c_1)^2+(1-c_2)^2\end{aligned}$$

$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_1}=2(c_1-4)+2(c_1-2)+2(c_1-3)$$

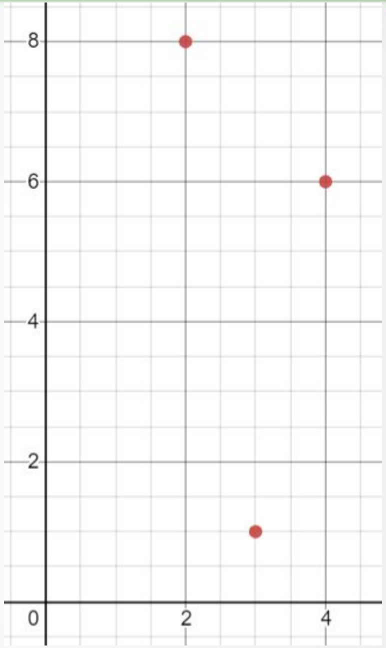
$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_2}=2(c_2-6)+2(c_2-8)+2(c_2-1)$$

Why does k-means clustering work?

Cost($\boldsymbol{\mu}_j$) = total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

Example: group j contains (4, 6), (2, 8), (3,1)

How to place centroid $\boldsymbol{\mu}_j = (c_1, c_2)$ to minimize cost?



$$
\begin{aligned}
\frac{\partial \mathrm{Cost}(\mu_j)}{\partial c_1} &= 2(c_1-4)+2(c_1-2)+2(c_1-3) \\
0 &= 2(c_1-4)+2(c_1-2)+2(c_1-3) \\
0 &= c_1-4+c_1-2+c_1-3 \\
3c_1 &= 4+2+3 \\
c_1 &= \frac{4+2+3}{3}
\end{aligned}
$$

Why does k-means clustering work?

Cost($\boldsymbol{\mu}_j$) = total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

Example: group j contains (4, 6), (2, 8), (3,1)

How to place centroid $\boldsymbol{\mu}_j$ = (c_1, c_2) to minimize cost?

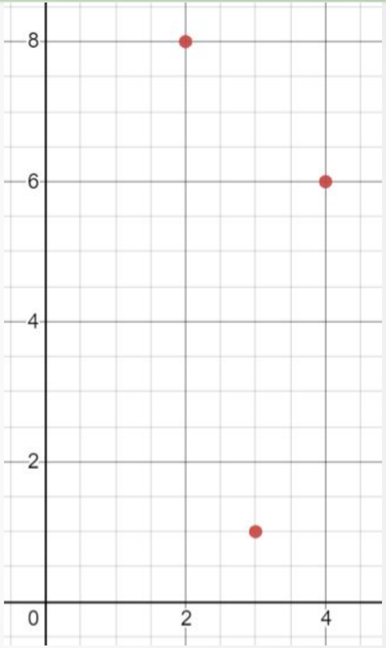
$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_2} = 2(c_2 - 6) + 2(c_2 - 8) + 2(c_2 - 1)$$

$$0 = 2(c_2 - 6) + 2(c_2 - 8) + 2(c_2 - 1)$$

$$0 = c_2 - 6 + c_2 - 8 + c_2 - 1$$

$$3c_2 = 6 + 8 + 1$$

$$c_2 = \frac{6 + 8 + 1}{3}$$



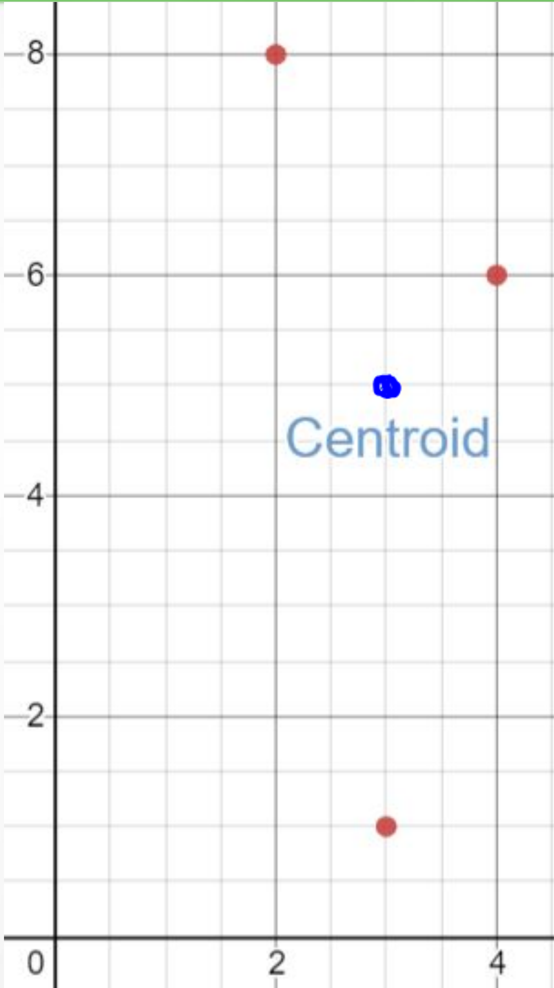
Why does k-means clustering work?

Cost($\boldsymbol{\mu}_j$) = total squared distance of each data point x_i in group j to centroid $\boldsymbol{\mu}_j$

Example: group j contains (4, 6), (2, 8), (3,1)

How to place centroid $\boldsymbol{\mu}_j = (c_1, c_2)$ to minimize cost?

$$(c_1, c_2) = \left(\frac{4+2+3}{3} , \frac{6+8+1}{3} \right) = (3, 5)$$



Minimize cost by averaging in each coordinate.

Cost, Loss, and Risk

The cost of placing the centroid at (c_1, c_2) is

$$\begin{aligned}\text{Cost}\,(\mu_j) &= \left(\sqrt{(4-c_1)^2+(6-c_2)^2}\right)^2+\left(\sqrt{(2-c_1)^2+(8-c_2)^2}\right)^2+\left(\sqrt{(3-c_1)^2+(1-c_2)^2}\right)^2\\ &= (4-c_1)^2+(6-c_2)^2+(2-c_1)^2+(8-c_2)^2+(3-c_1)^2+(1-c_2)^2\end{aligned}$$

minimize this fcn

$$\text{Cost}(\mu_j) = \underbrace{(4-c_1)^2+(2-c_1)^2+(3-c_1)^2}_{f(c_1)} + \underbrace{(6-c_2)^2+(8-c_2)^2+(1-c_2)^2}_{g(c_2)}$$

connection to mean squared error, R_{sq}

$$f(c_1) = (4-c_1)^2+(2-c_1)^2+(3-c_1)^2$$

$$R_{sq}(h) = \frac{1}{3}\left((4-h)^2+(2-h)^2+(3-h)^2\right)$$

← minimized at mean of 4, 2, 3

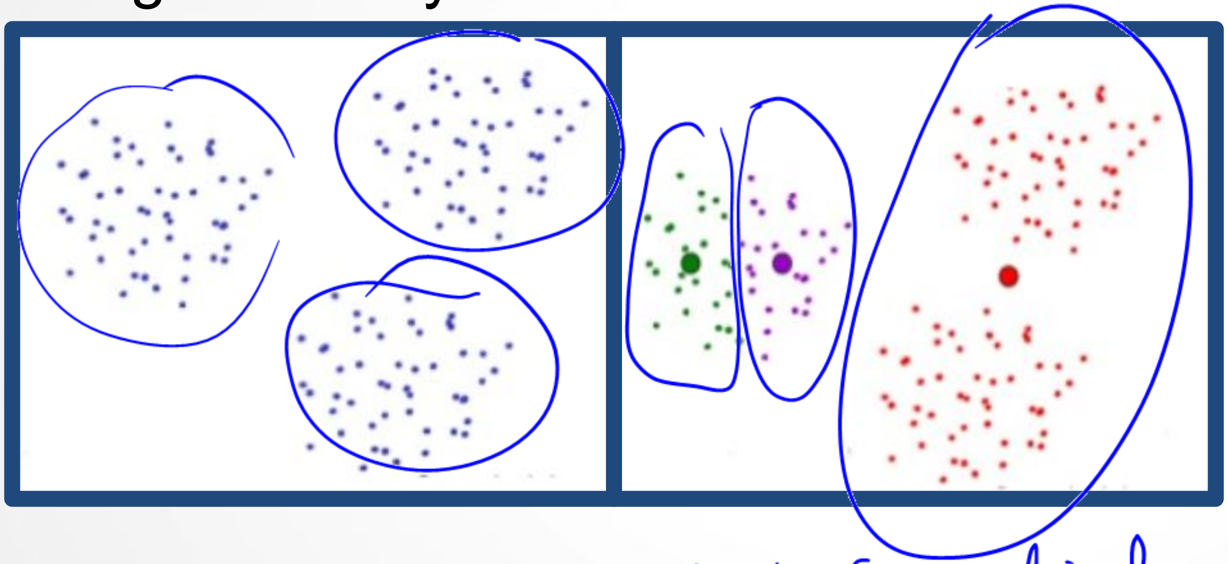
Why does k-means clustering work?

Cost($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, ... , $\boldsymbol{\mu}_k$) = total squared distance of each data point x_i to its nearest centroid $\boldsymbol{\mu}_j$

- Argue why updating the groups and centroids according to the algorithm reduces the cost with each iteration.
- With enough iterations, cost will be sufficiently small.

k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.



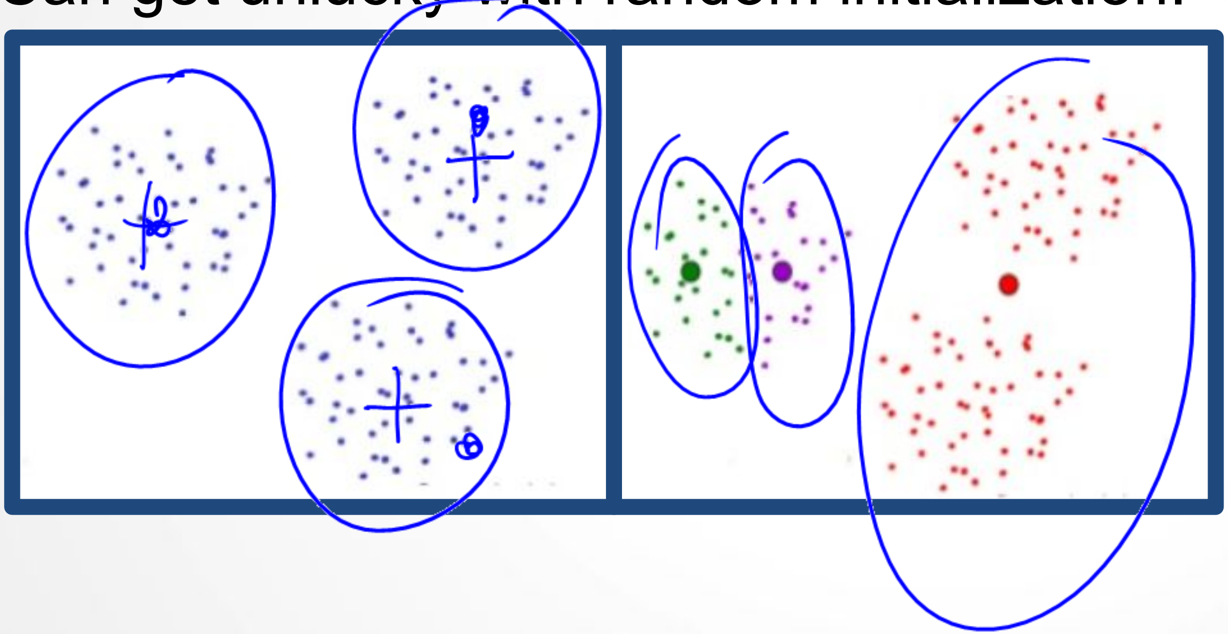
Cost fcn high

In general, how do we assess which result is the best?

A. Clusters appear how we expect them to
B. Clusters are evenly sized
C. Cost function is lowest

k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.



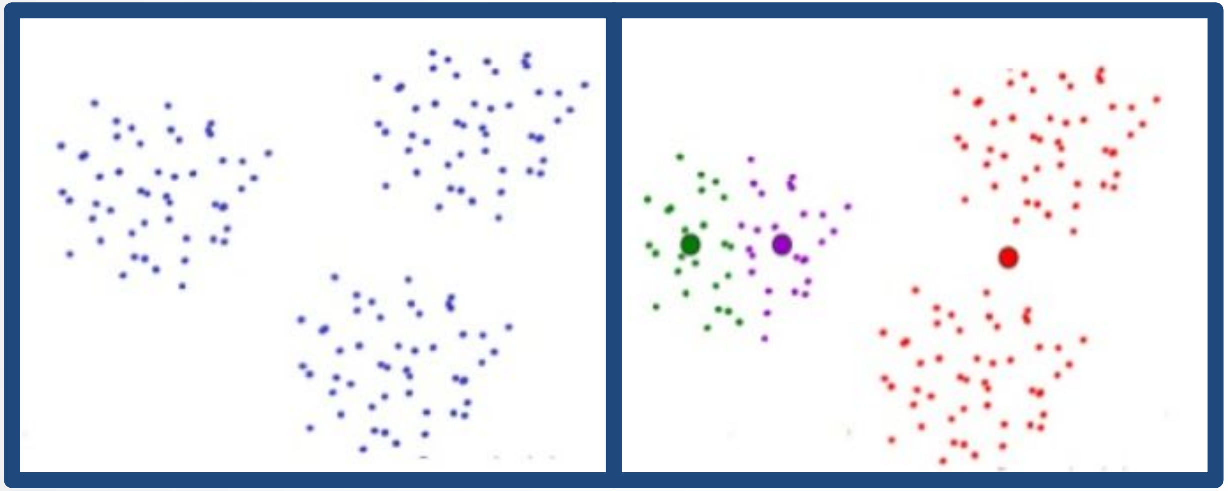
In general, how do we assess which result is the best?

A. Clusters appear how we expect them to
B. Clusters are evenly sized
C. Cost function is lowest

Solution?
- Try algorithm several times, pick the best result.
- Similar approach used in gradient descent.

k-Means Clustering in Practice: Initialization

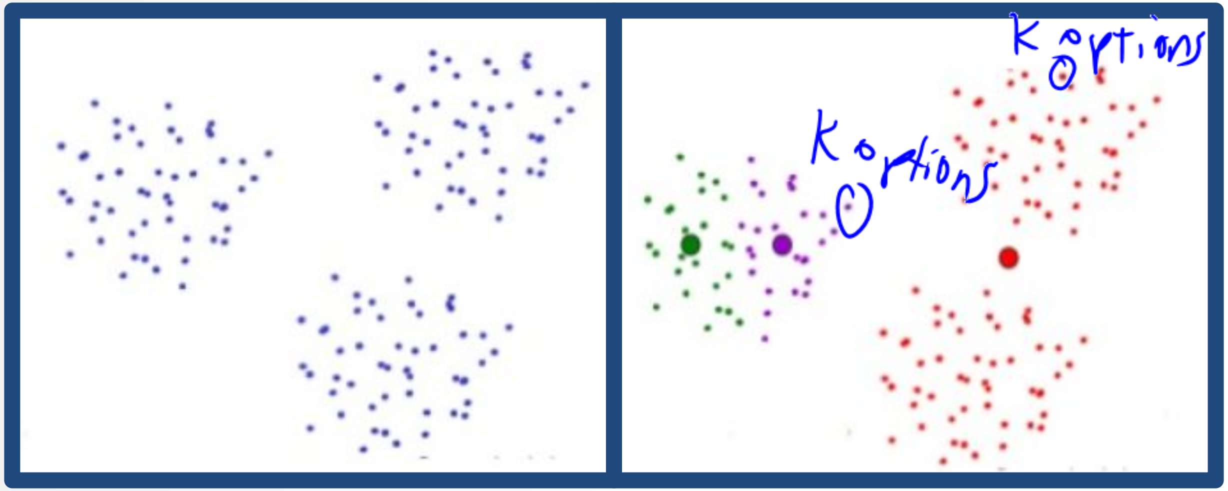
Can get unlucky with random initialization.



- No guarantees of a satisfactory solution with this algorithm.
- Brute force algorithm would try all assignments of points to clusters and choose the one with the lowest cost.

k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.



How many ways to assign n points to k clusters?

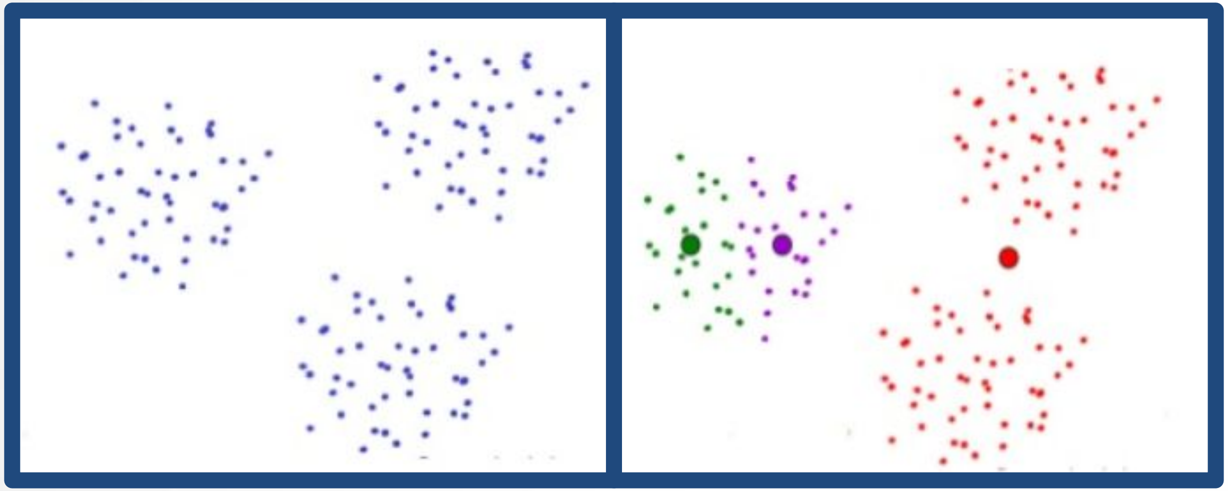
k = # colors

$$\underbrace{k * k * k * \dots * k}_{n \text{ points}} = \boxed{k^n}$$

- No guarantees of a satisfactory solution with this algorithm.
- Brute force algorithm would try all assignments of points to clusters and choose the one with the lowest cost.

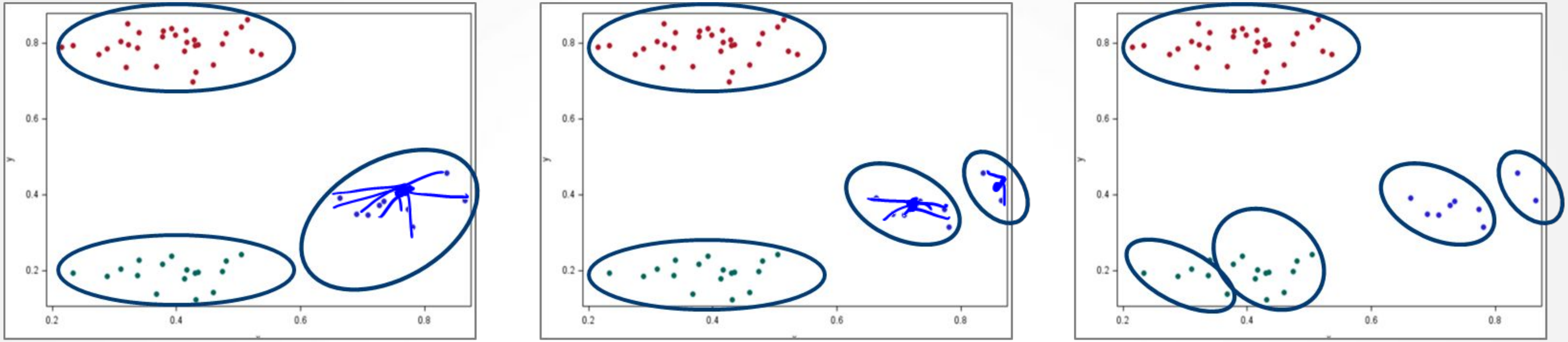
k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.



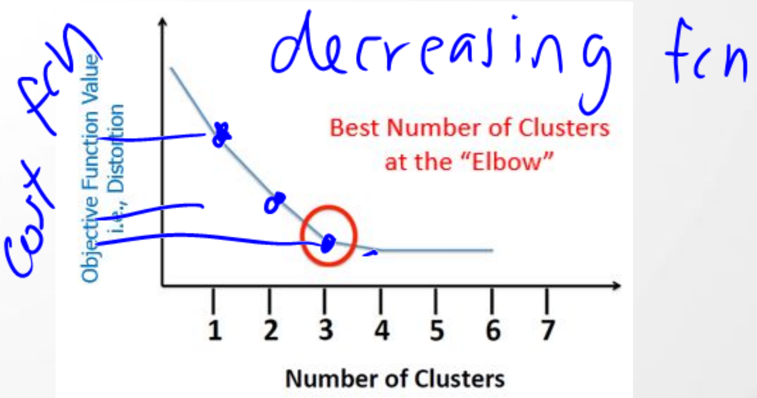
- No guarantees of a satisfactory solution with this algorithm.
- Any algorithm that is guaranteed to find the best coloring of data points takes exponential time (computationally infeasible).

k-Means Clustering in Practice: Choosing k



k=3 k=4 k=5

- Most commonly done by hand (visualizations, trial and error)
- Elbow method
- Context or domain knowledge
- Use a different clustering algorithm



What if a centroid has no points in its group?

What should we do if a centroid has no points in its group?

A. Terminate the algorithm.
B. Wait for points get added to the group in a subsequent iteration.
C. Set the centroid to be a data point, chosen at random.
D. Set the centroid to be one of the other centroids, chosen at random.

What if a centroid has no points in its group?

What should we do if a centroid has no points in its group?

A. Terminate the algorithm.
B. Wait for points get added to the group in a subsequent iteration.
C. Set the centroid to be a data point, chosen at random.
D. Set the centroid to be one of the other centroids, chosen at random.

Two options:

- Eliminate that centroid and find k-1 clusters instead
- Randomly re-initialize that centroid

Summary

- We saw that k-means clustering works because each step of the algorithm reduces the cost function, which measures the quality of a set of centroids.
- We discussed some practical considerations, including random initialization and choice of k.