CS 453X: Class 17

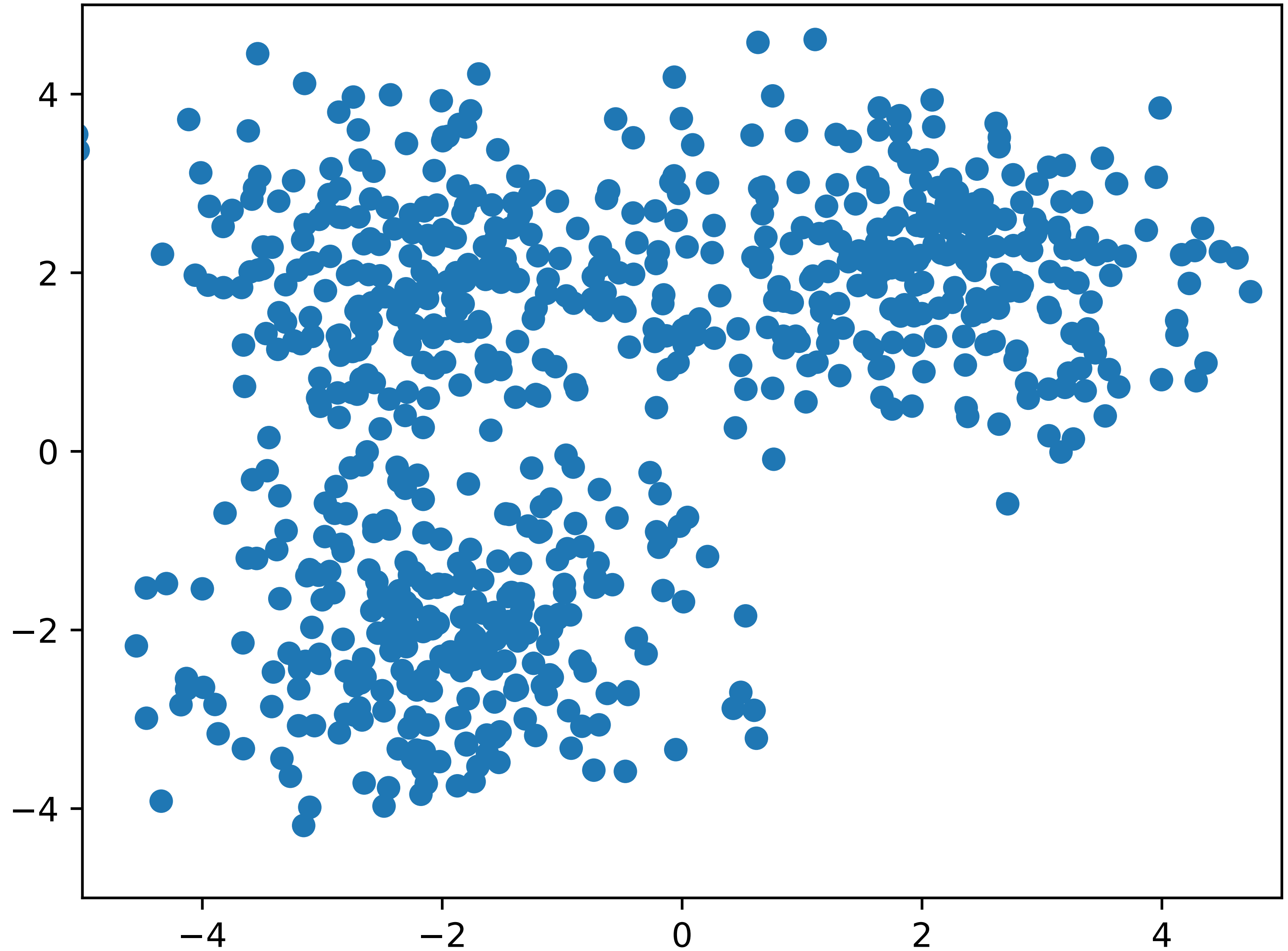
Jacob Whitehill

Finding structure in scatter

Finding structure in scatter

- What do you see in this figure?

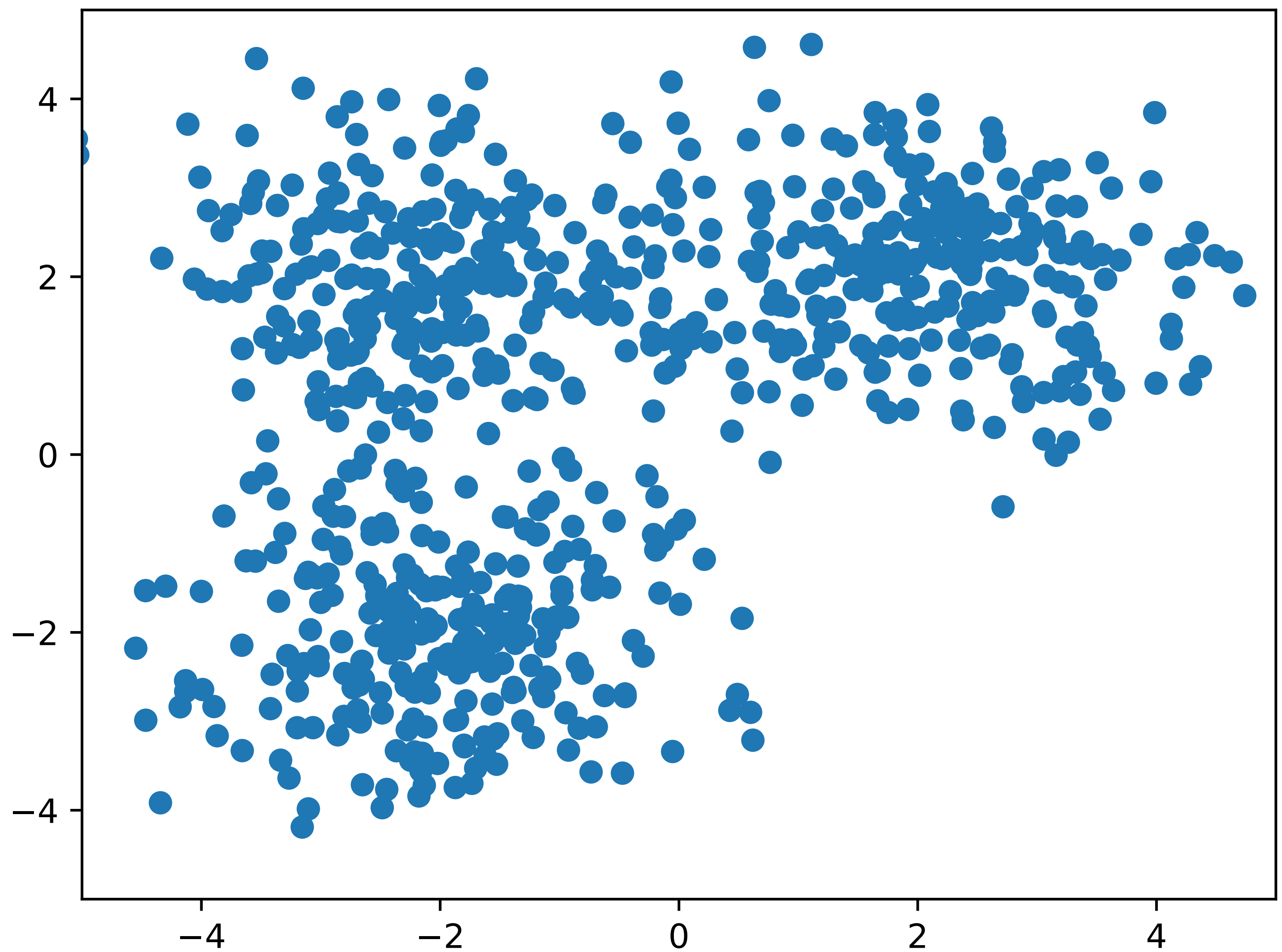
  1. A candelabra.



Finding structure in scatter

- What do you see in this figure?

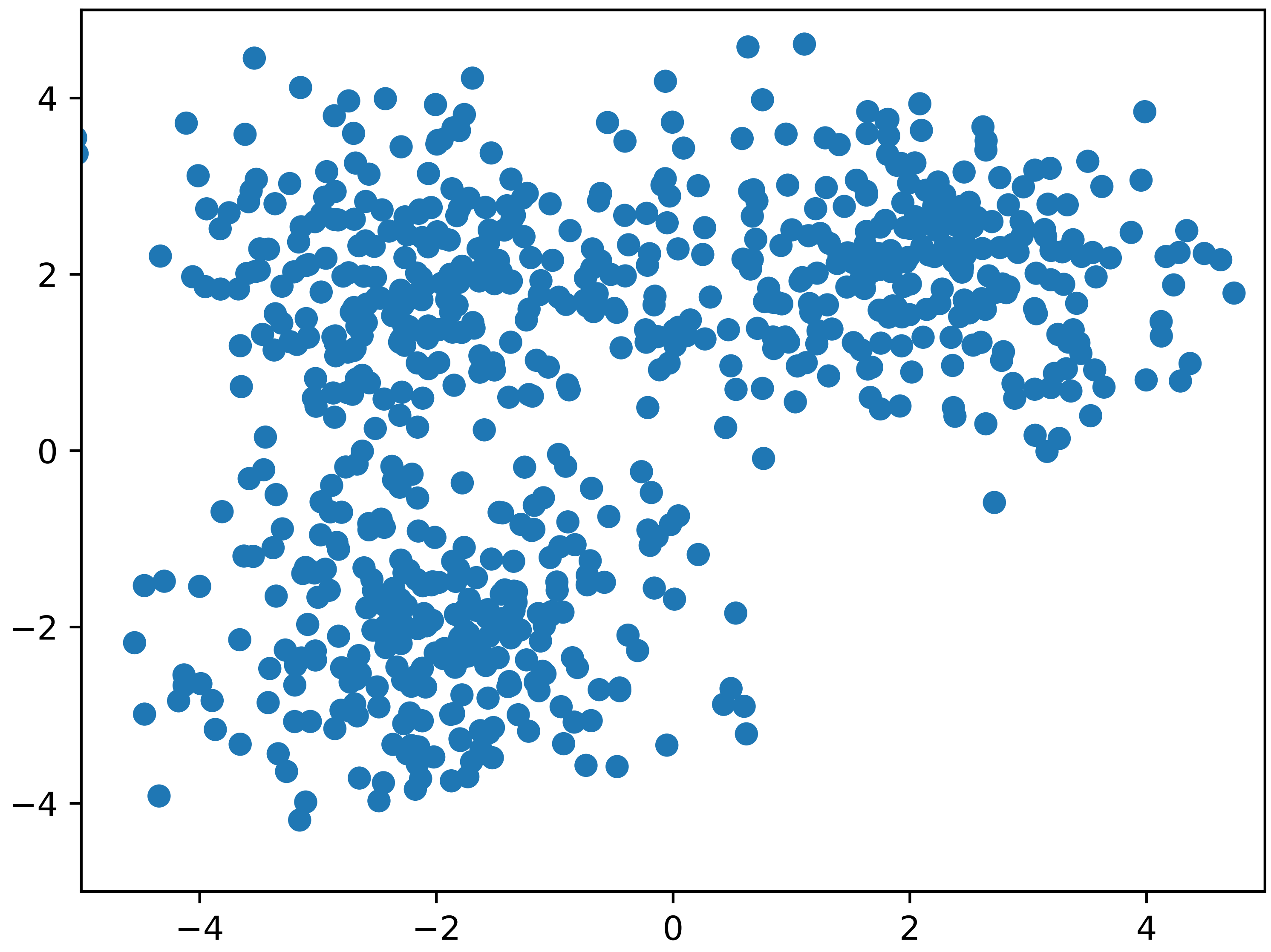
  2.A young woman frowning.



Finding structure in scatter

- What do you see in this figure?

  3.Three somewhat distinct clusters of data points.

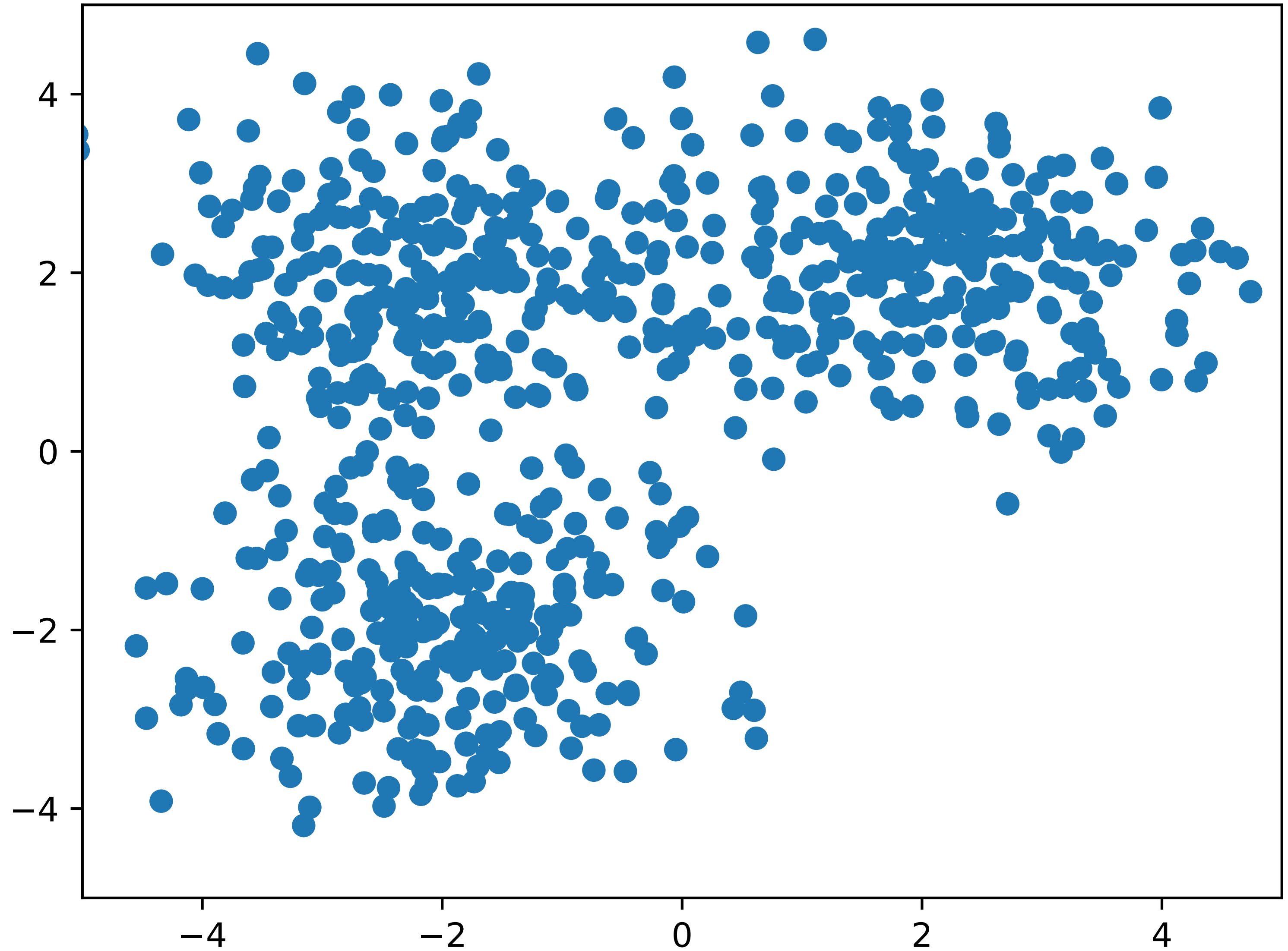


Finding structure in scatter

- What do you see in this figure?

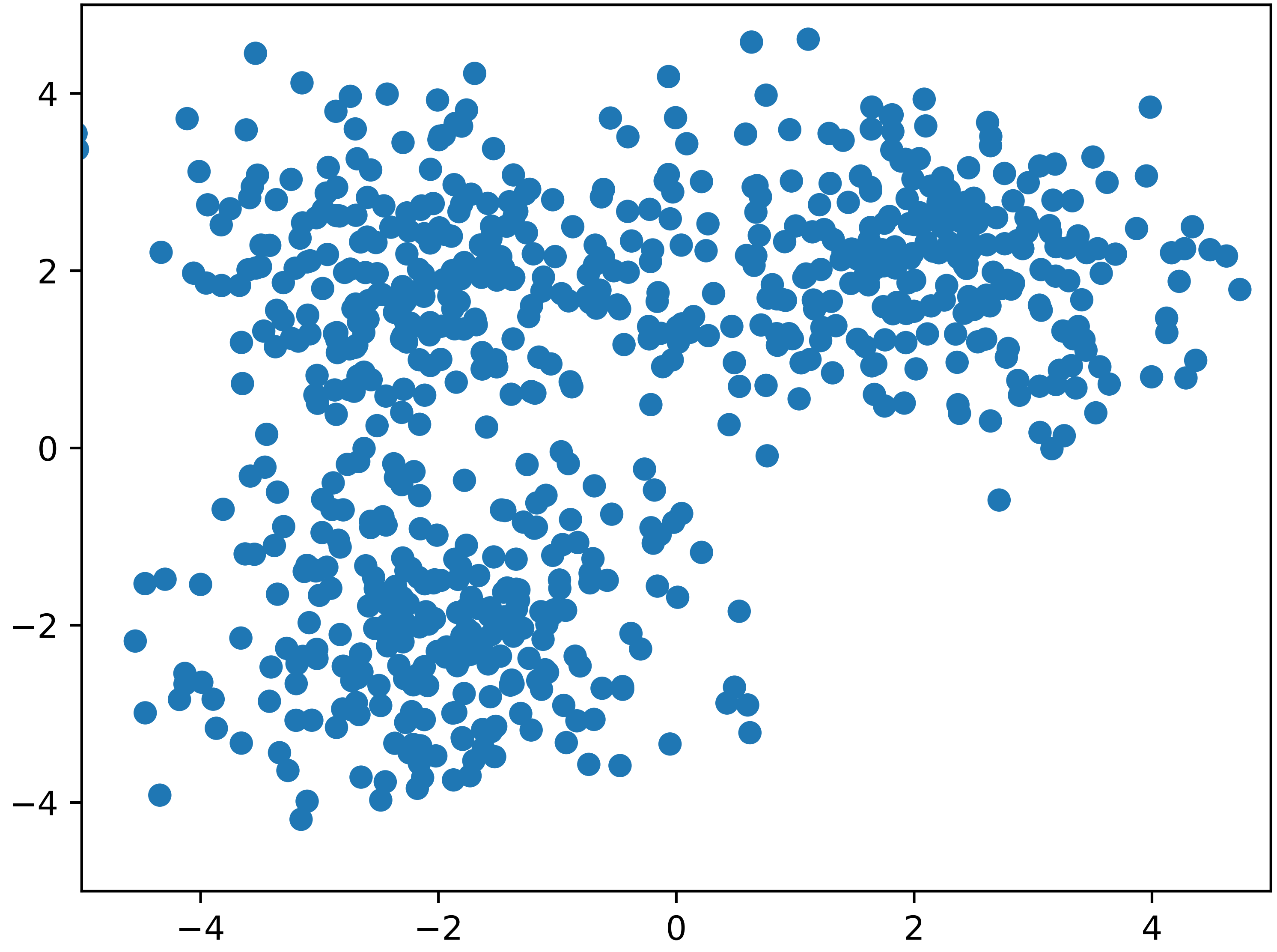
3. Three somewhat distinct clusters of data points.

Bingo.



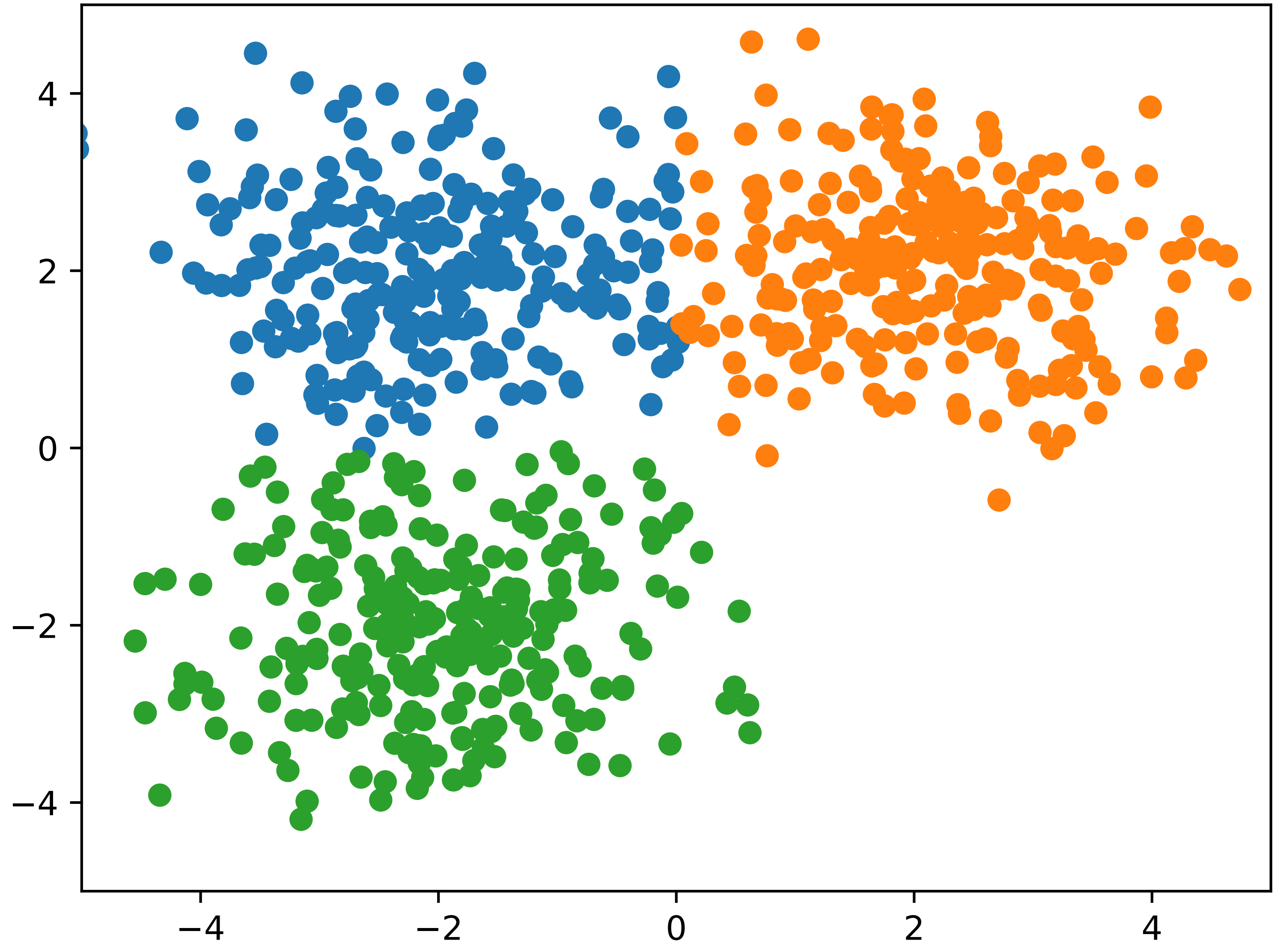
Finding structure in scatter

- Intuitively, we can define **clustering** as putting data into groups such that: data *within* each group are more similar than data *between* groups.



Finding structure in scatter

- Wouldn't it be nice to be able to cluster data automatically?
  - Maybe the clusters align with certain natural structure in the data (e.g., digit classes in MNIST dataset)?

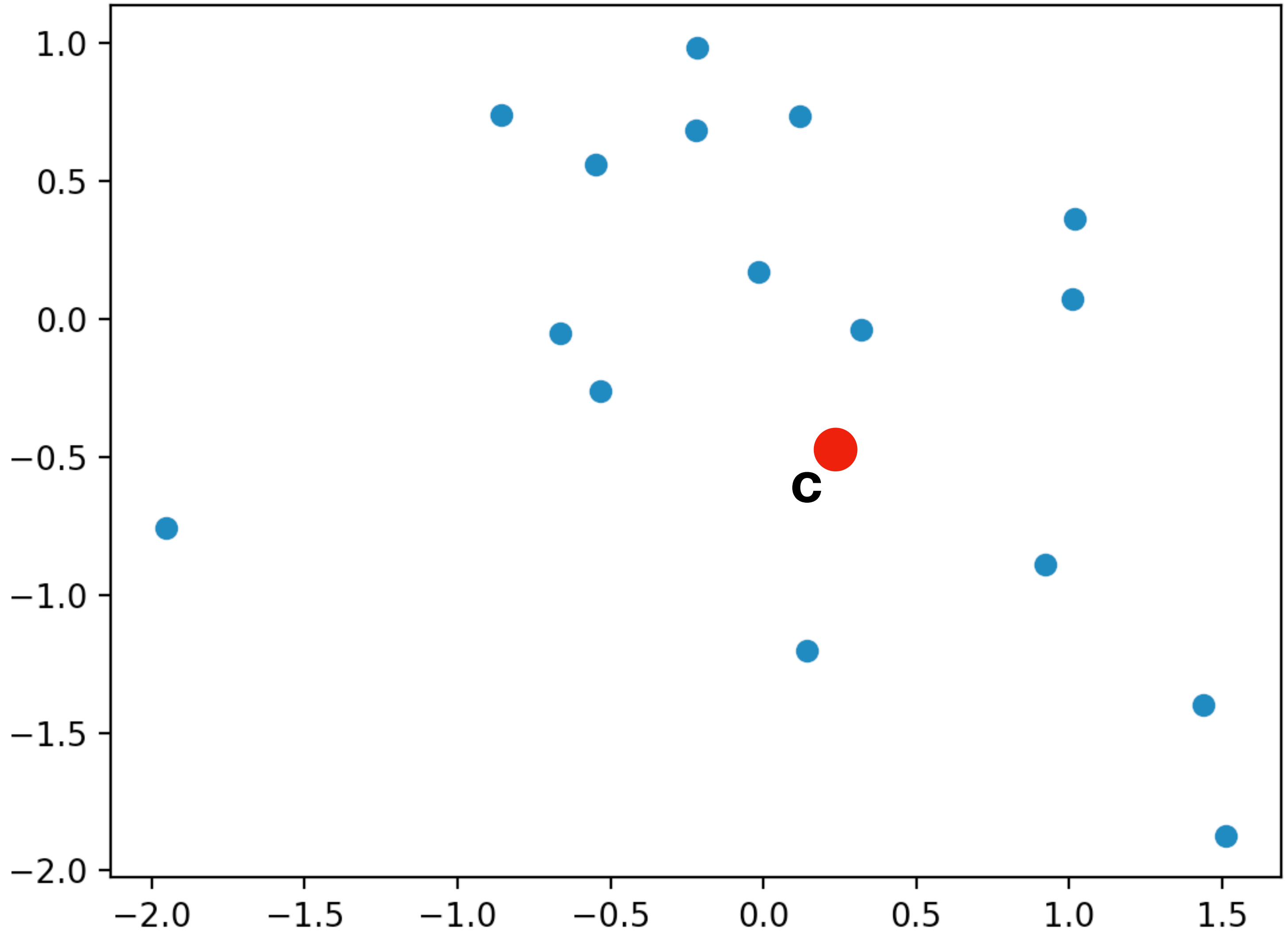


Show demo.

Clustering: background

Clustering: background

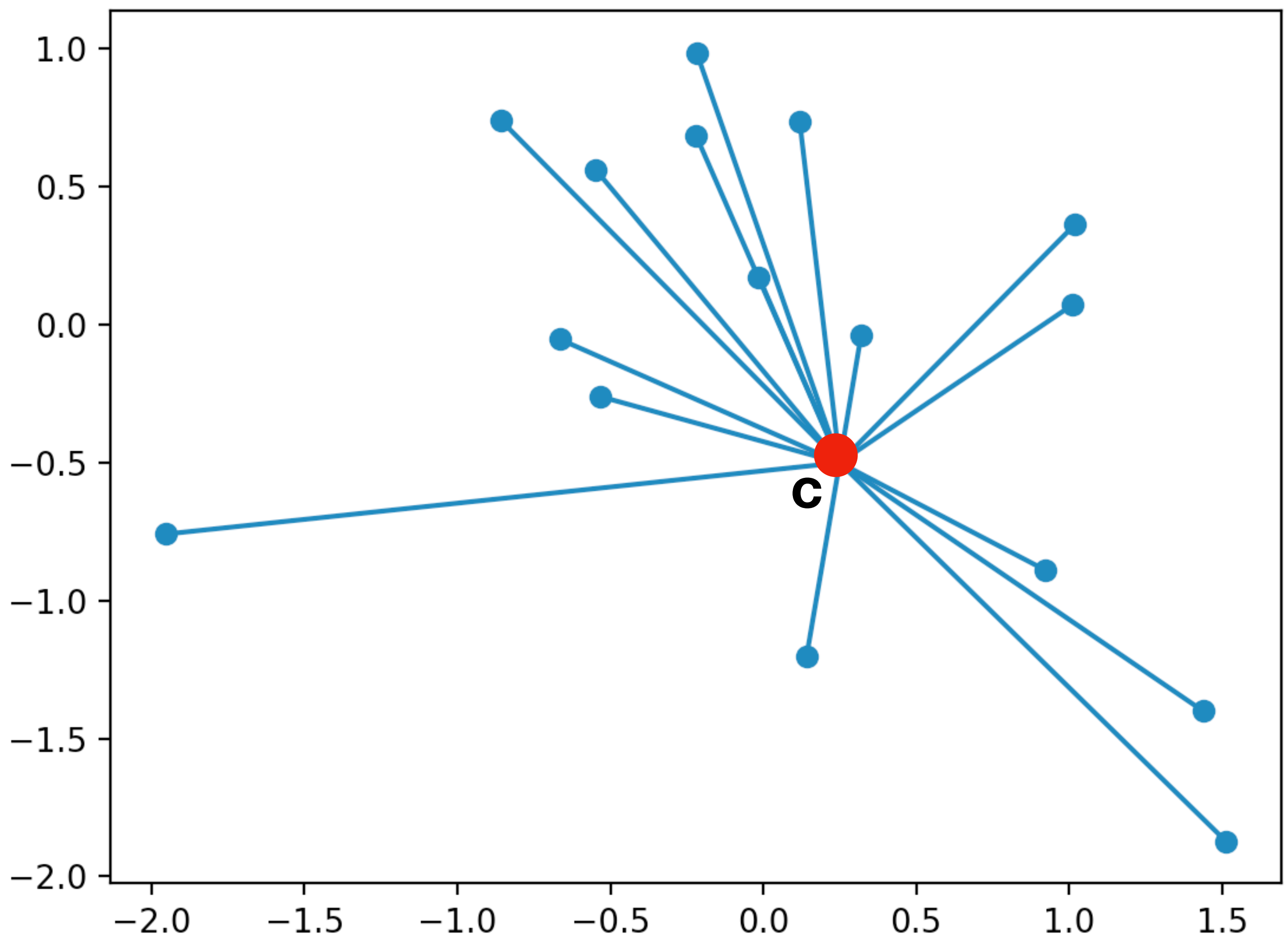
- Consider a set of n data points $\{ \mathbf{x}^{(i)} \}$, and another point $\mathbf{c}$:



Clustering: background

- Let's define a cost function f_{SSD} as the sum of squared distances between each data point $\mathbf{x}^{(i)}$ and **c**.

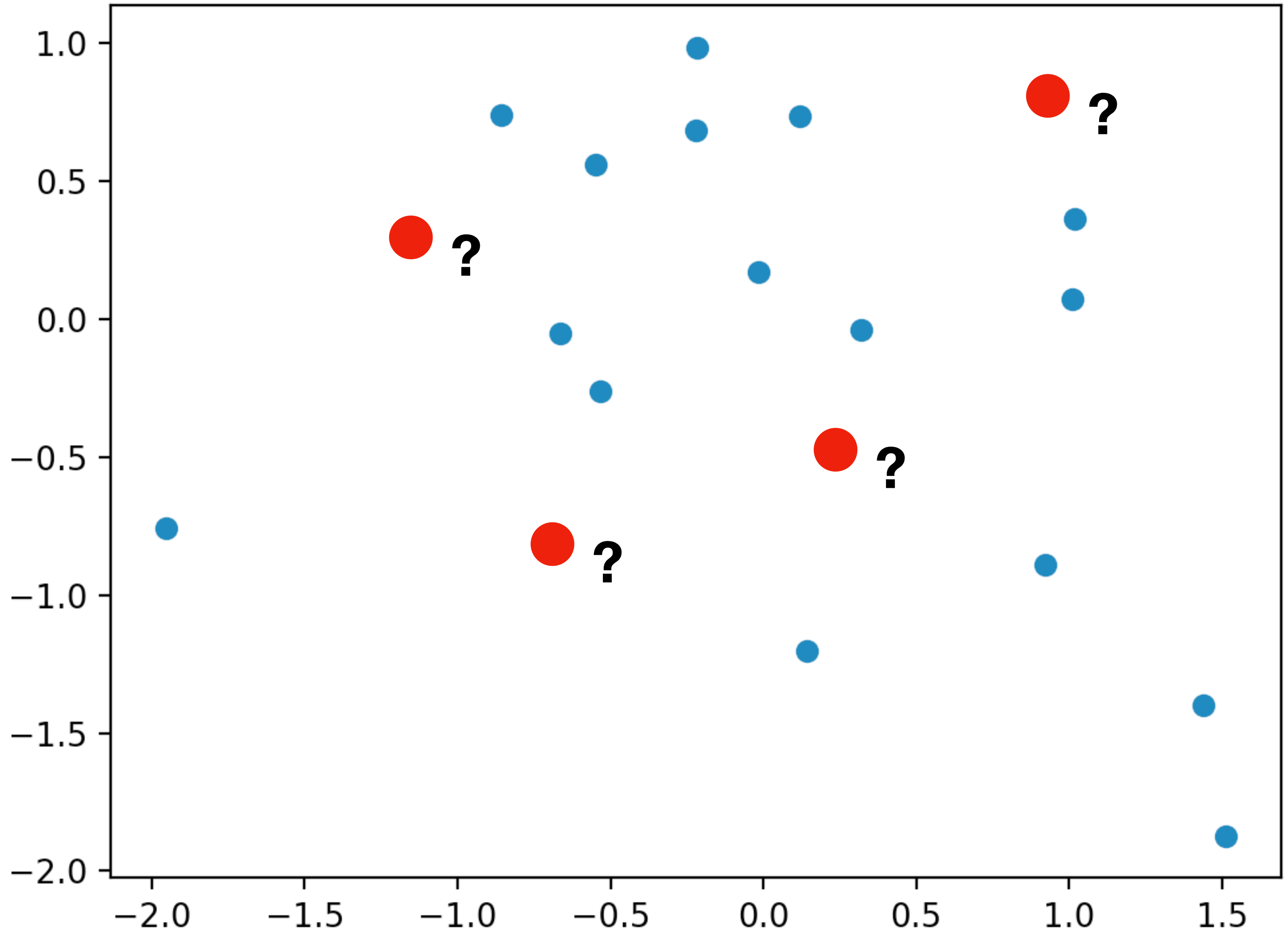
$$f_{\text{SSD}}(\mathbf{c}) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c} - \mathbf{x}^{(i)})^2$$



Clustering: background

- Which point **c** minimizes f_{SSD} for { $\mathbf{x}^{(i)}$ }?

$$f_{\mathrm{SSD}}(\mathbf{c}) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c} - \mathbf{x}^{(i)})^2$$



Center of *n* points

$$f_{\mathrm{SSD}}(\mathbf{c}) \quad = \quad \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})^2$$

Center of *n* points

$$
\begin{aligned}
f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
\nabla_{\mathbf{c}} f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}\nabla_{\mathbf{c}}(\mathbf{c}-\mathbf{x}^{(i)})^2
\end{aligned}
$$

Center of *n* points

$$
\begin{aligned}
f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
\nabla_{\mathbf{c}} f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}\nabla_{\mathbf{c}}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
&= \sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})
\end{aligned}
$$

Center of *n* points

$$
\begin{aligned}
f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
\nabla_{\mathbf{c}} f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}\nabla_{\mathbf{c}}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
&= \sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)}) \\
&= \sum_{i=1}^{n}\mathbf{c}-\sum_{i=1}^{n}\mathbf{x}^{(i)}
\end{aligned}
$$

Center of *n* points

$$
\begin{aligned}
f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
\nabla_{\mathbf{c}} f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}\nabla_{\mathbf{c}}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
&= \sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)}) \\
&= \sum_{i=1}^{n}\mathbf{c}-\sum_{i=1}^{n}\mathbf{x}^{(i)} \\
&= n\mathbf{c}-\sum_{i=1}^{n}\mathbf{x}^{(i)} = 0
\end{aligned}
$$

Center of *n* points

$$
\begin{aligned}
f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
\nabla_{\mathbf{c}} f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}\nabla_{\mathbf{c}}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
&= \sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)}) \\
&= \sum_{i=1}^{n}\mathbf{c}-\sum_{i=1}^{n}\mathbf{x}^{(i)} \\
&= n\mathbf{c}-\sum_{i=1}^{n}\mathbf{x}^{(i)} = 0 \\
n\mathbf{c} &= \sum_{i=1}^{n}\mathbf{x}^{(i)}
\end{aligned}
$$

Center of *n* points

$$
\begin{aligned}
f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
\nabla_{\mathbf{c}} f_{\mathrm{SSD}}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{n}\nabla_{\mathbf{c}}(\mathbf{c}-\mathbf{x}^{(i)})^2 \\
&= \sum_{i=1}^{n}(\mathbf{c}-\mathbf{x}^{(i)}) \\
&= \sum_{i=1}^{n}\mathbf{c}-\sum_{i=1}^{n}\mathbf{x}^{(i)} \\
&= n\mathbf{c}-\sum_{i=1}^{n}\mathbf{x}^{(i)} = 0 \\
n\mathbf{c} &= \sum_{i=1}^{n}\mathbf{x}^{(i)} \\
\mathbf{c} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}
\end{aligned}
$$

Center of n points

- In other words, the point $\mathbf{c}$ that minimizes f_{SSD} is the mean of the n points $\{ \mathbf{x}^{(i)} \}$.
  - Every other point $\mathbf{c}' \neq \mathbf{c}$ must have a *higher* f_{SSD}.

k-means clustering algorithm

k-means clustering algorithm

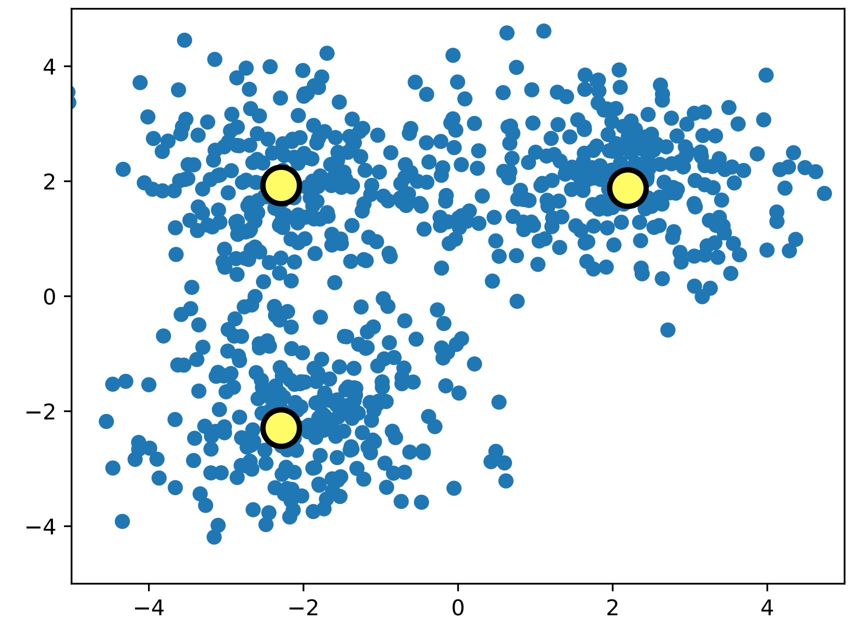
- Probably the simplest and most commonly used clustering algorithm is called ***k*-means**.

- It partitions a set of n data $\{ \mathbf{x}^{(i)} \}$ into k clusters.

k-means clustering algorithm

- Chicken-and-the-egg problem:
  - If we knew the *mean* $\mathbf{c}^{(j)}$ of each cluster j, we could assign each data point $\mathbf{x}$ to the closest cluster center.

$$a(\mathbf{x}) = \arg\min_j \left(\mathbf{x} - \mathbf{c}^{(j)}\right)^2$$



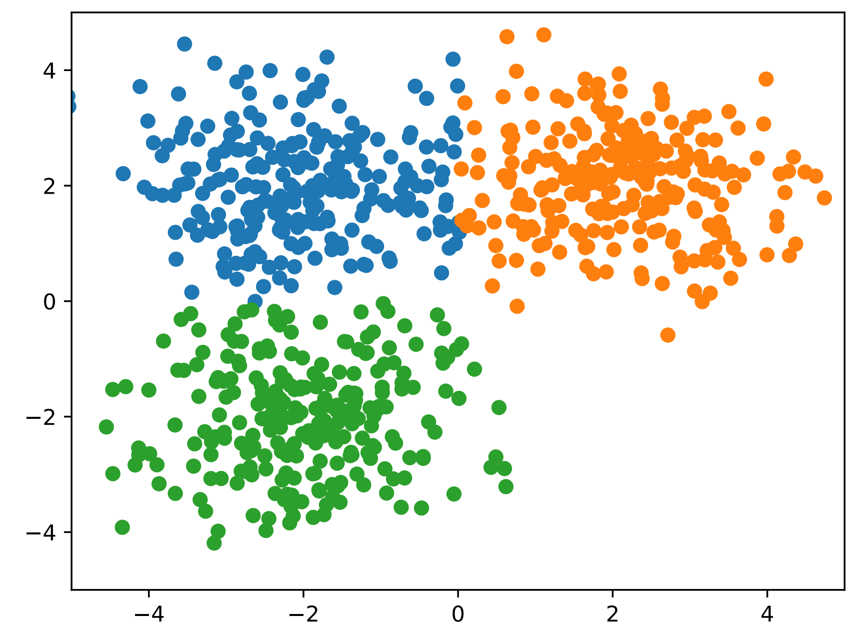
k-means clustering algorithm

- Chicken-and-the-egg problem:
  - If we knew the *mean* $\mathbf{c}^{(j)}$ of each cluster j, we could assign each data point **x** to the closest cluster center.

$$a(\mathbf{x}) = \arg\min_j \left(\mathbf{x} - \mathbf{c}^{(j)}\right)^2$$

  - If we knew which data were in which cluster, we could compute the mean of each cluster.

$$\mathbf{c}_j = \frac{1}{n_j} \sum_{i:a(\mathbf{x}^{(i)})=j} \mathbf{x}^{(i)}$$



k-means clustering algorithm

- Chicken-and-the-egg problem:
  - If we knew the *mean* $\mathbf{c}^{(j)}$ of each cluster *j*, we could assign each data point **x** to the closest cluster center.

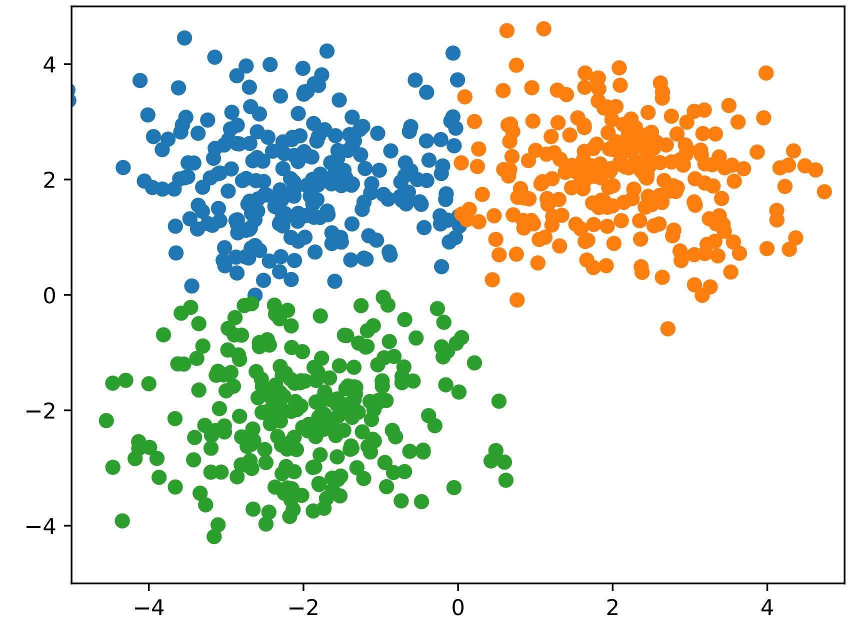
$$a(\mathbf{x}) = \arg\min_j \left(\mathbf{x} - \mathbf{c}^{(j)}\right)^2$$

  - If we knew which data were in which cluster, we could compute the mean of each cluster.

$$\mathbf{c}_j = \frac{1}{n_j} \sum_{i:a(\mathbf{x}^{(i)})=j} \mathbf{x}^{(i)}$$

data in cluster *j*

The data in cluster *j*.



k-means clustering algorithm

- The *k*-means algorithm seeks to optimize the assignment of data points to clusters so as to minimize:

$$\sum_{j=1}^{k} \sum_{i:a(\mathbf{x}^{(i)})=j} \left(\mathbf{x}^{(i)} - \mathbf{c}^{(j)}\right)^2$$

- Algorithm:

1. Randomly assign the data points to clusters.
2. Repeat until the cost does not change:
   A. Compute $\mathbf{c}^{(j)}$ of each cluster j as the mean of the points assigned to it.
   B. Assign each point $\mathbf{x}^{(i)}$ to the nearest cluster mean $\mathbf{c}^{(j)}$.

k-means clustering: convergence

- Why is the “Repeat until the cost does not change” loop guaranteed to converge?

k-means clustering: convergence

- Why is the "Repeat until the cost does not change" loop guaranteed to converge? $$\sum_{j=1}^{k} \sum_{i:a(\mathbf{x}^{(i)})=j} \left(\mathbf{x}^{(i)} - \mathbf{c}^{(j)}\right)^2$$
  - The *k*-means cost function has a *lower bound* (0) since the sum can never be negative.
  - Each step within the loop can only *lower* the cost:

k-means clustering: convergence

- Why is the "Repeat until the cost does not change" loop guaranteed to converge?

$$\sum_{j=1}^{k} \sum_{i:a(\mathbf{x}^{(i)})=j} \left(\mathbf{x}^{(i)} - \mathbf{c}^{(j)}\right)^2$$

  - The *k*-means cost function has a *lower bound* (0) since the sum can never be negative.
  - Each step within the loop can only *lower* the cost:

    A. Compute $\mathbf{c}^{(j)}$ of each cluster j as the mean of the points assigned to it.

As shown earlier, the mean of the data { $\mathbf{x}^{(i)}$ } minimizes f_{SSD}.

k-means clustering: convergence

- Why is the "Repeat until the cost does not change" loop guaranteed to converge?

$$\sum_{j=1}^{k} \sum_{i:a(\mathbf{x}^{(i)})=j} \left(\mathbf{x}^{(i)} - \mathbf{c}^{(j)}\right)^2$$

  - The *k*-means cost function has a *lower bound* (0) since the sum can never be negative.
  - Each step within the loop can only *lower* the cost:

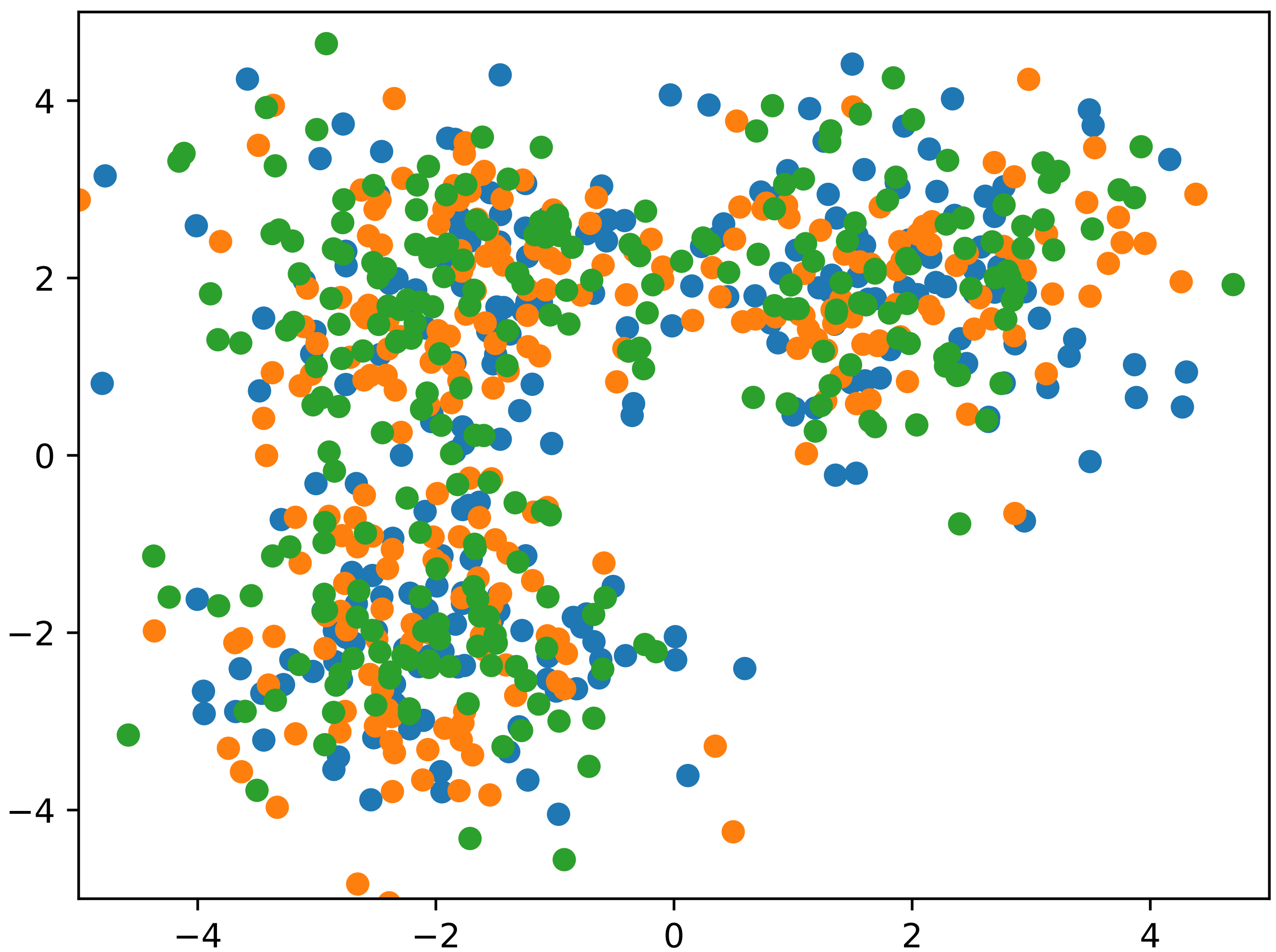
    A. Compute $\mathbf{c}^{(j)}$ of each cluster j as the mean of the points assigned to it.

    As shown earlier, the mean of the data $\{ \mathbf{x}^{(i)} \}$ minimizes f_{SSD}.

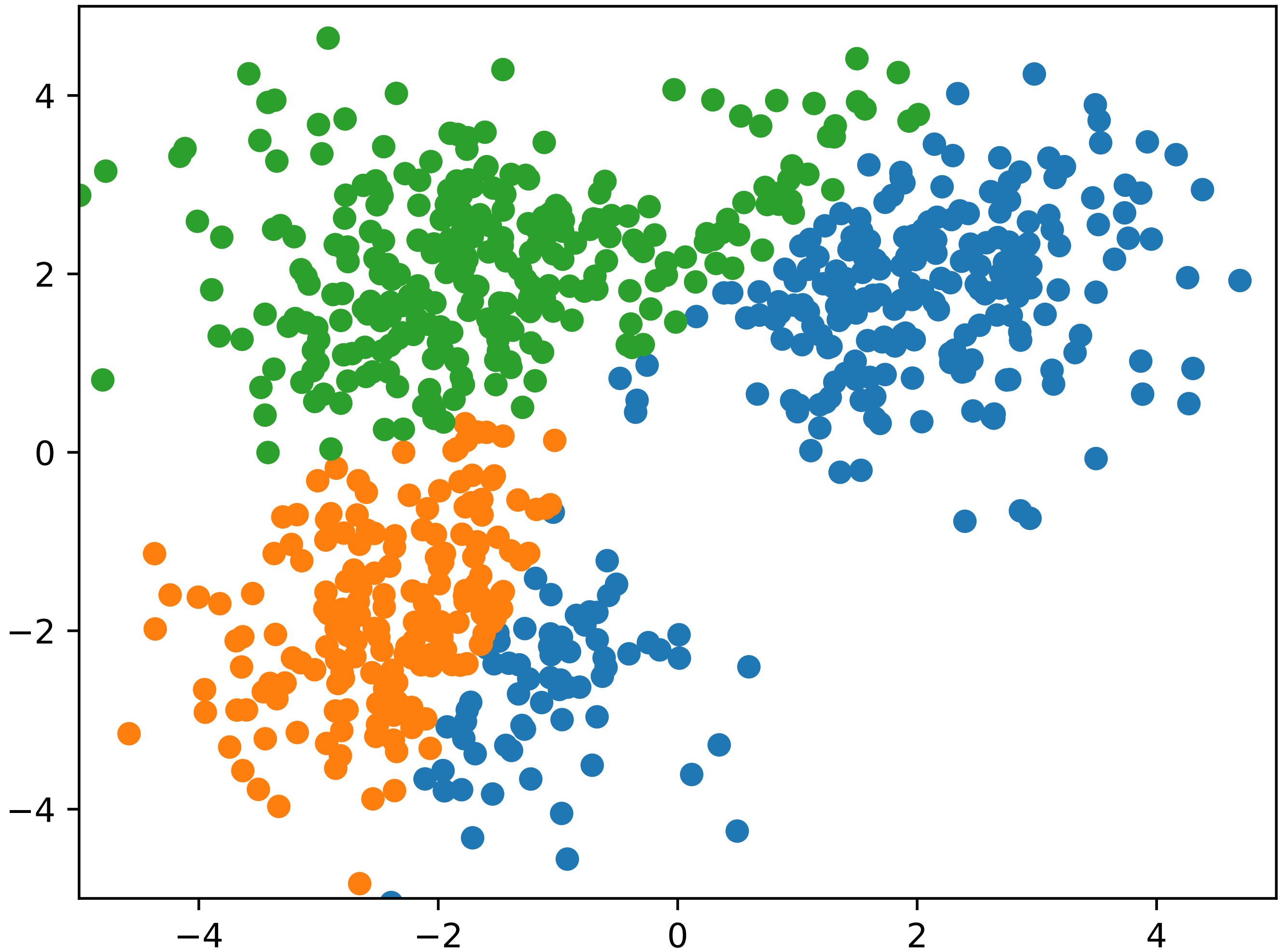
    B. Assign each point $\mathbf{x}^{(i)}$ to the nearest cluster mean $\mathbf{c}^{(j)}$.

    Assigning $\mathbf{x}^{(i)}$ to the closest cluster mean $\mathbf{c}^{(j)}$ can only decrease the cost due to $\mathbf{x}^{(i)}$.

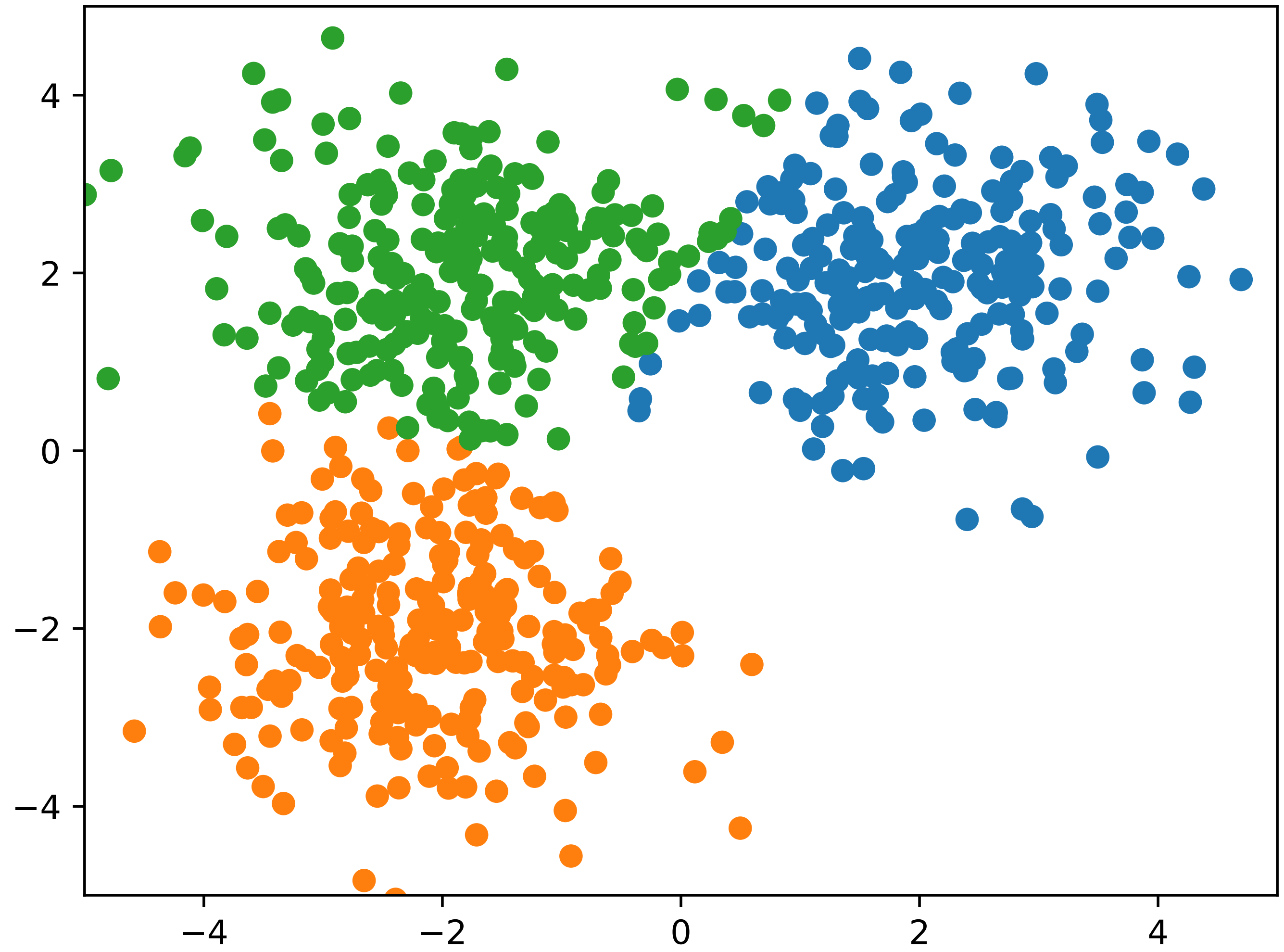
Example



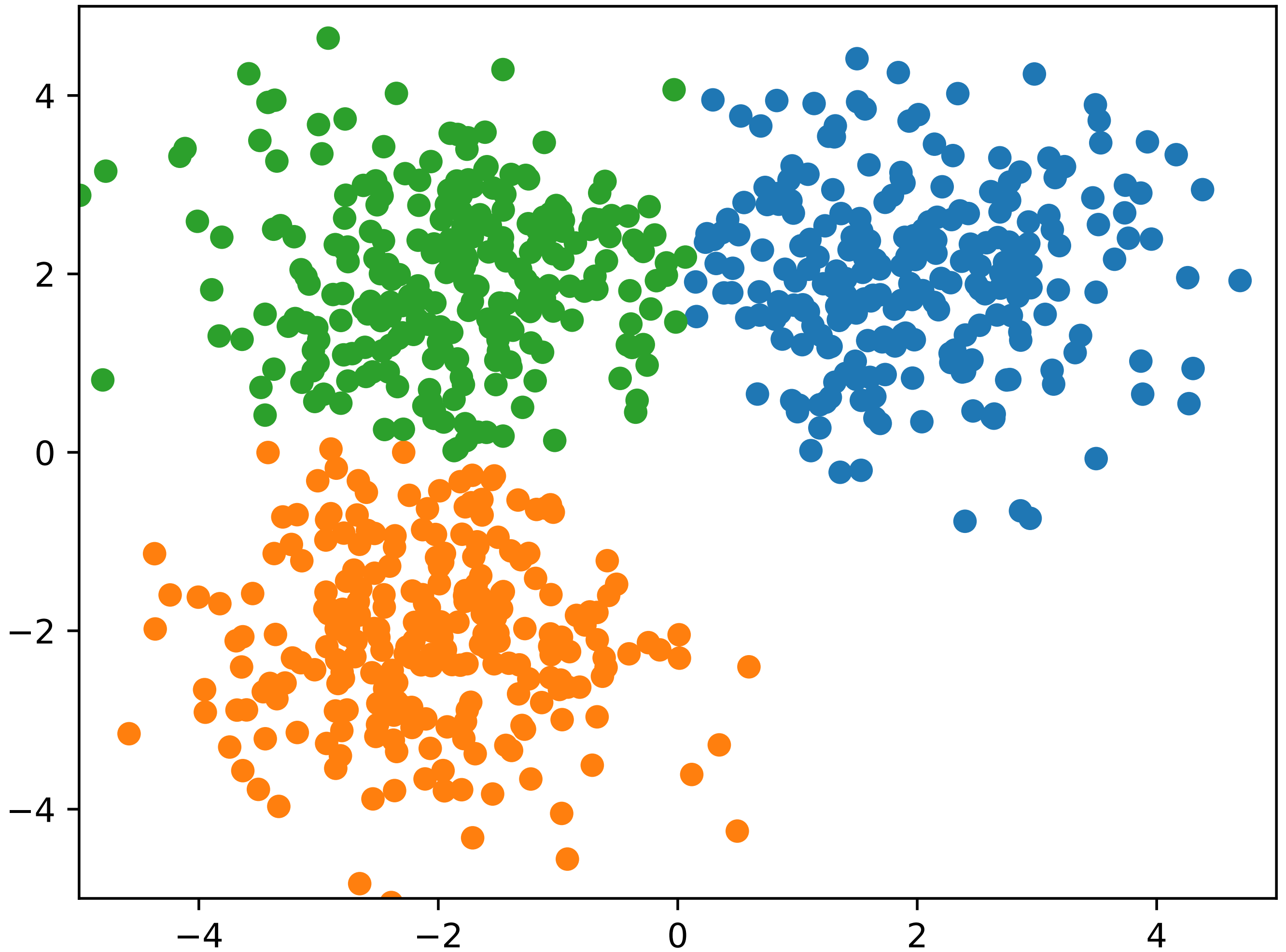
Example



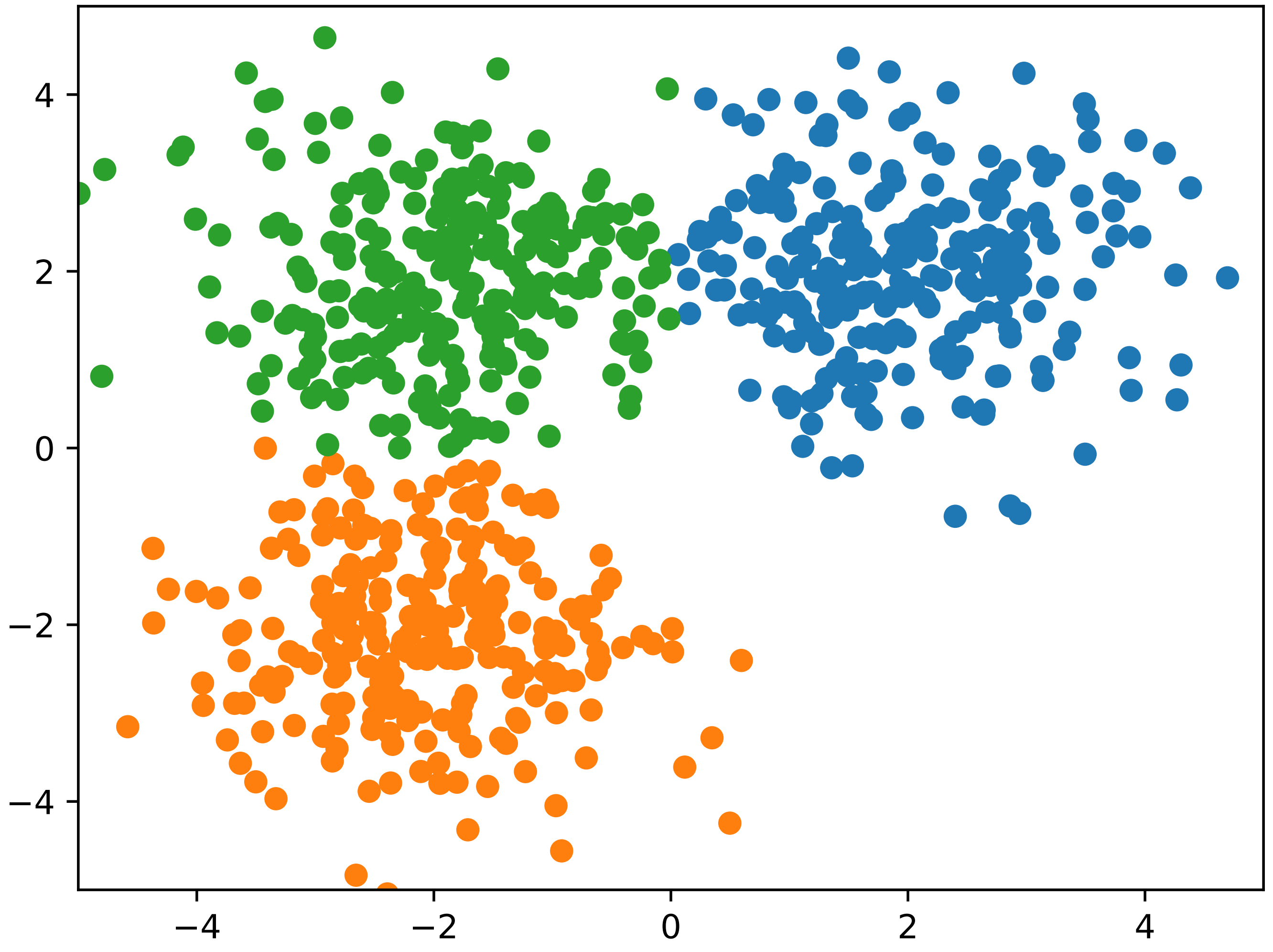
Example



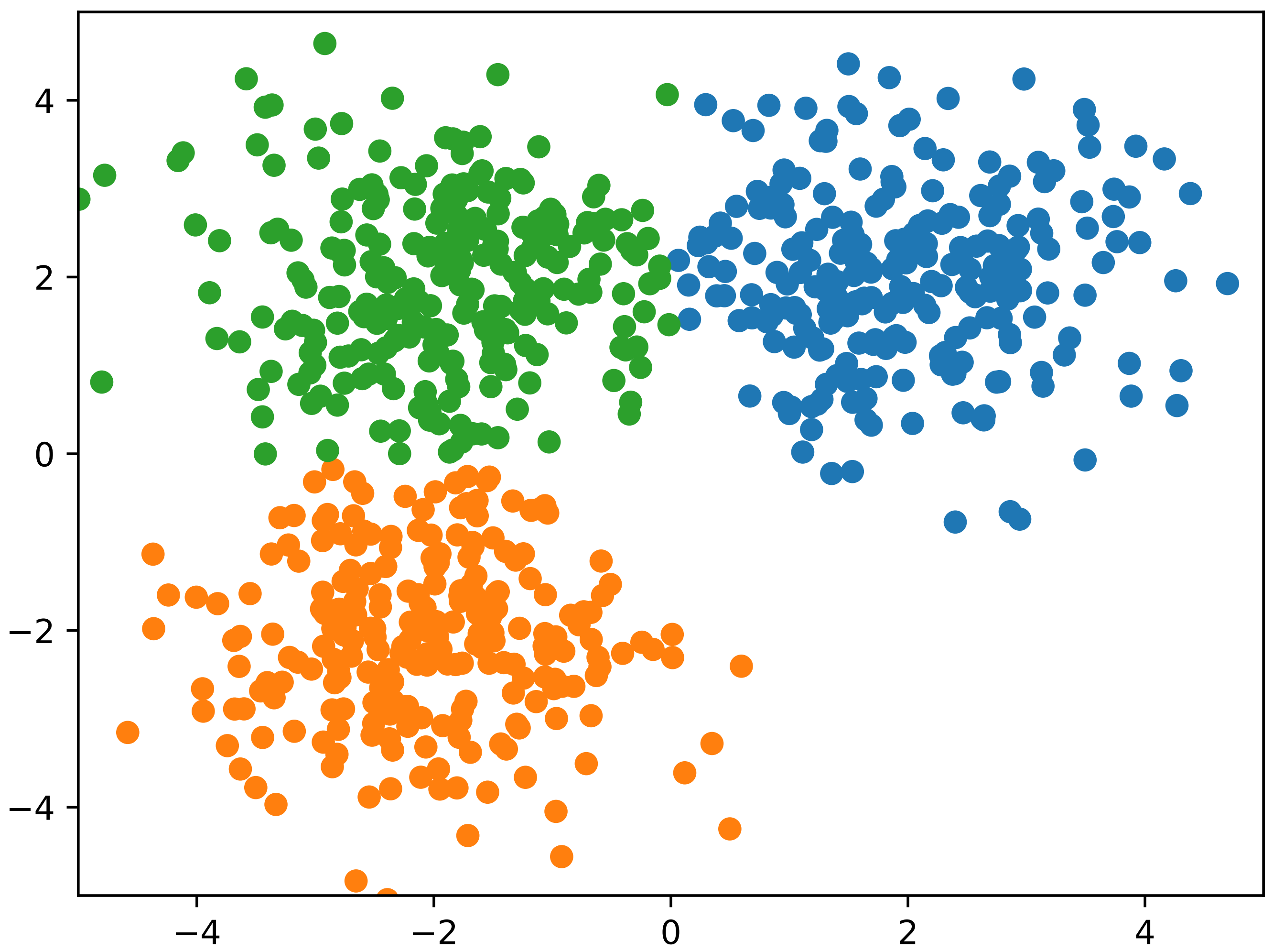
Example



Example

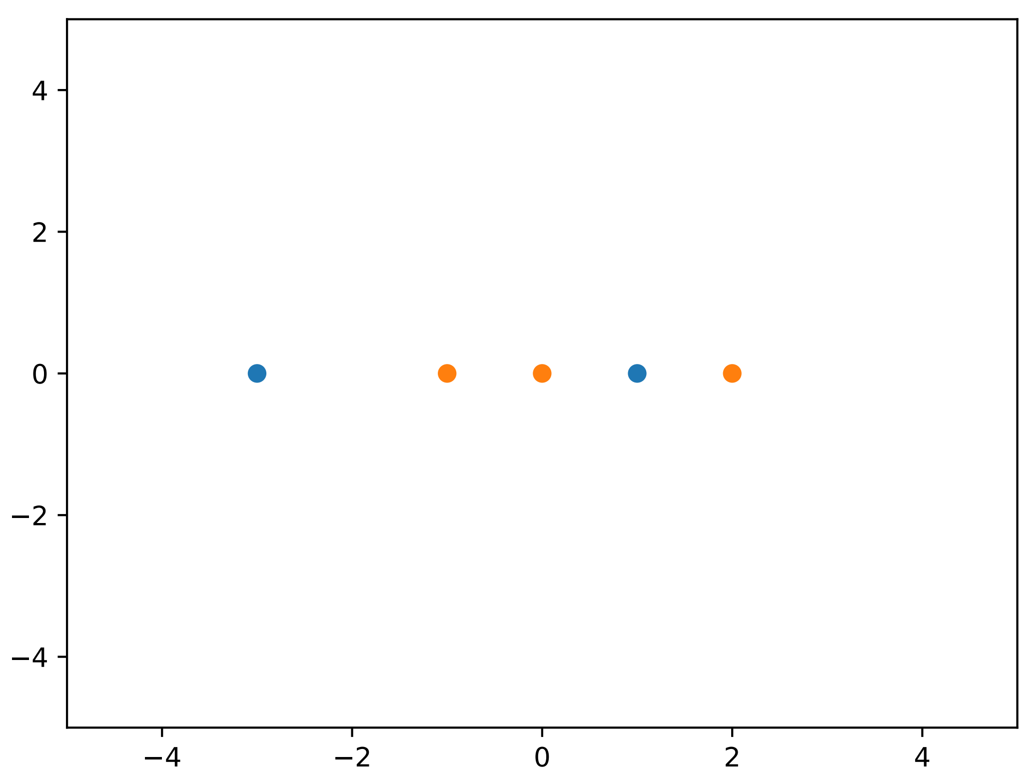


Example



Exercise: *k*-means on 1-D data

- Suppose the (1-D) data consist of { -3, -1, 0, 1, 2 }.
- Suppose we initialize the clusters as:



$$\sum_{j=1}^{k} \sum_{i:a(\mathbf{x}^{(i)})=j} \left(\mathbf{x}^{(i)} - \mathbf{c}^{(j)}\right)^2$$

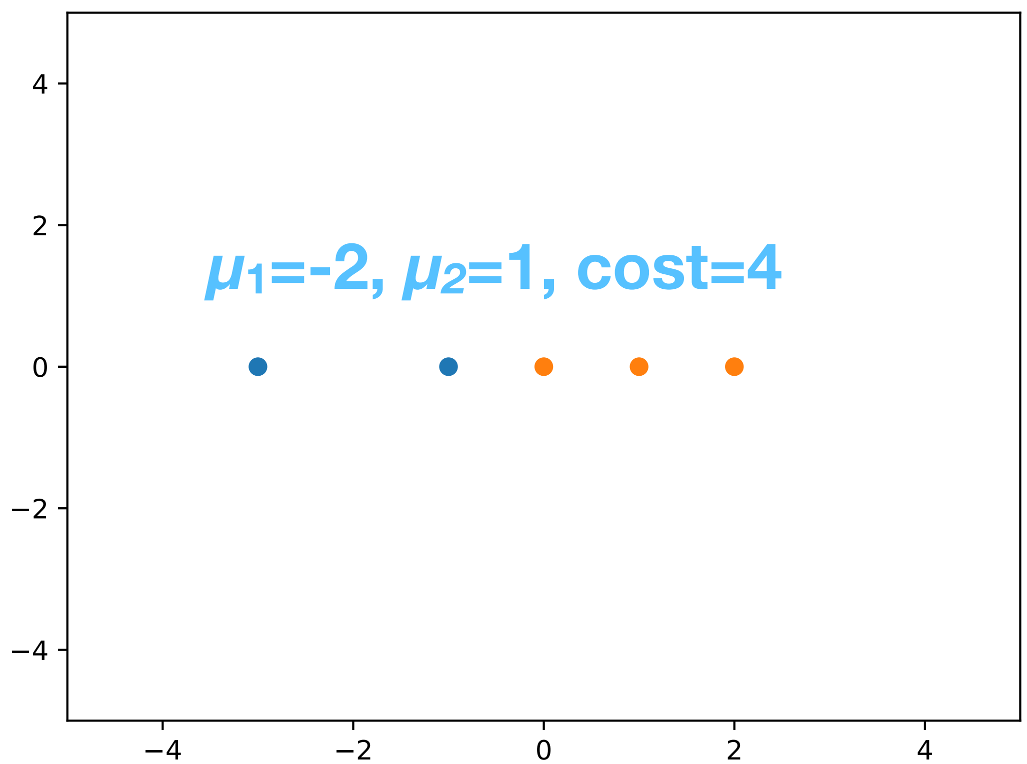
- Repeat until the cost does not change:
  A. Compute $\mathbf{c}^{(j)}$ of each cluster j as the mean of the points assigned to it.
  B. Assign each point $\mathbf{x}^{(i)}$ to the nearest cluster mean $\mathbf{c}^{(j)}$.

- What will *k*-means output (cluster assignments & means)?

Exercise: *k*-means on 1-D data

- Suppose the (1-D) data consist of { -3, -1, 0, 1, 2 }.

- Suppose we initialize the clusters as:



$$\sum_{j=1}^{k} \sum_{i:a(\mathbf{x}^{(i)})=j} \left(\mathbf{x}^{(i)} - \mathbf{c}^{(j)}\right)^2$$

- Repeat until the cost does not change:
  A. Compute $\mathbf{c}^{(j)}$ of each cluster j as the mean of the points assigned to it.
  B. Assign each point $\mathbf{x}^{(i)}$ to the nearest cluster mean $\mathbf{c}^{(j)}$.

- What will *k*-means output (cluster assignments & means)?

Caveats

- The output of *k*-means can differ depending on how the algorithm was initialized.

Caveats

- The output of *k*-means can differ depending on how the algorithm was initialized.
- There is no guarantee that the algorithm will converge to a global minimum.

Caveats

- The output of *k*-means can differ depending on how the algorithm was initialized.
- There is no guarantee that the algorithm will converge to a global minimum.
- Sometimes, clusters can become empty.
    - In this case, we have to eliminate one (or more) of the clusters to avoid dividing by 0:

$$\mathbf{c}_j = \frac{1}{n_j} \sum_{i:a(\mathbf{x}^{(i)})=j} \mathbf{x}^{(i)}$$