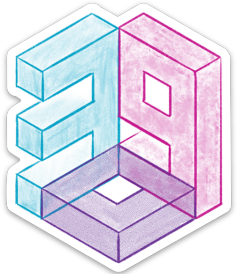


https://introml.mit.edu/

6.390 Intro to Machine Learning

Lecture 10: Clustering

Shen Shen

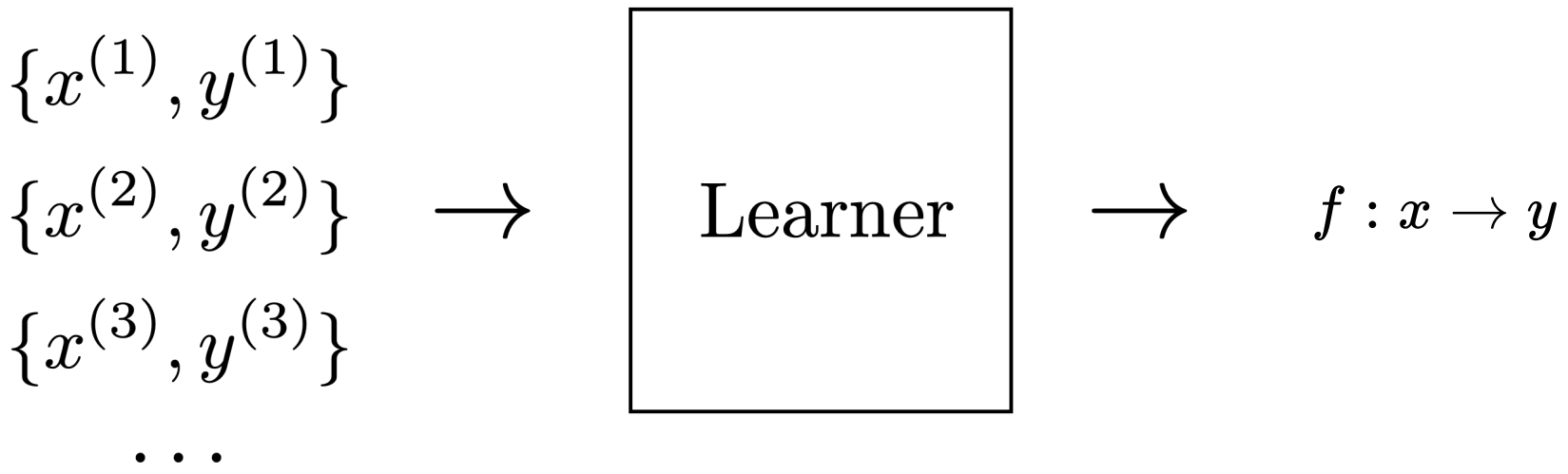
November 8, 2024

Outline

- Recap: Supervised learning and unsupervised learning
- k-means clustering:
  - k-means objective
  - k-means algorithm
    - Initialization matters
    - k matters
  - Clustering vs. classification
- Clustering and related

Recap: Supervised learning

Training data

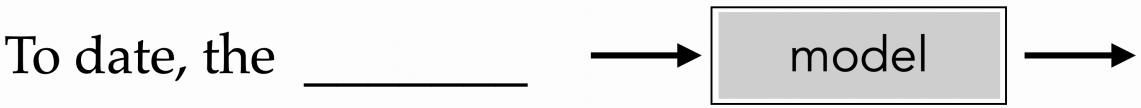


- **explicit** supervision via labels y.
- labels can be quite expensive to create.

Recap: Unsupervised/Self-supervised learning

"To date, the cleverest thinker of all time was Issac. "

feature	label
⋮	⋮
To date, the	cleverest
To date, the cleverest	thinker
To date, the cleverest thinker	was
⋮	⋮
To date, the cleverest thinker of all time was	Issac



Recap: Unsupervised/Self-supervised learning

Training Data

$$\left\{ x^{(i)} \right\}_{i=1}^{n}$$

Auto-encoder

loss/objective

$$\mathcal{L}(F(\mathbf{x}), \mathbf{x}) = \|F(\mathbf{x}) - \mathbf{x}\|^2$$

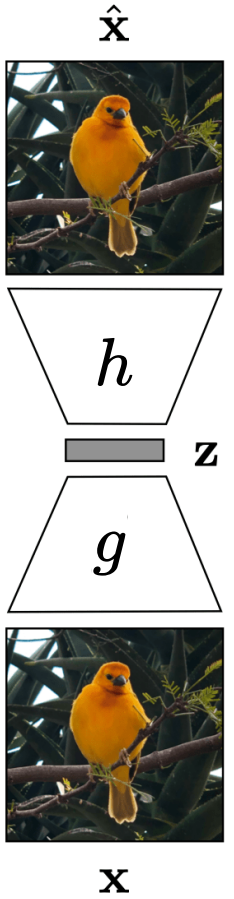
hypothesis class

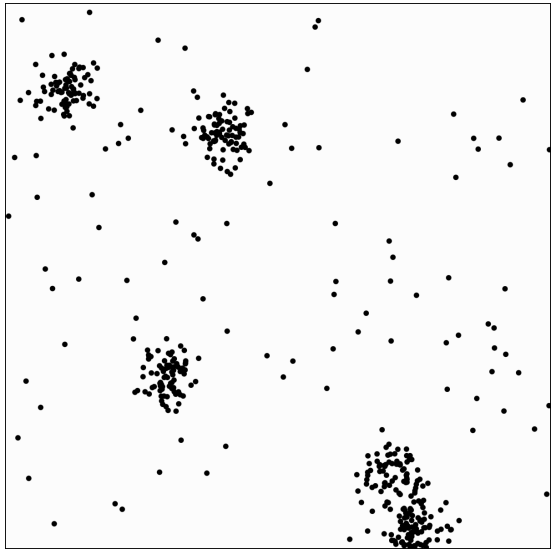
$$F = g \circ h : \mathbb{R}^d \to \mathbb{R}^m \to \mathbb{R}^d$$

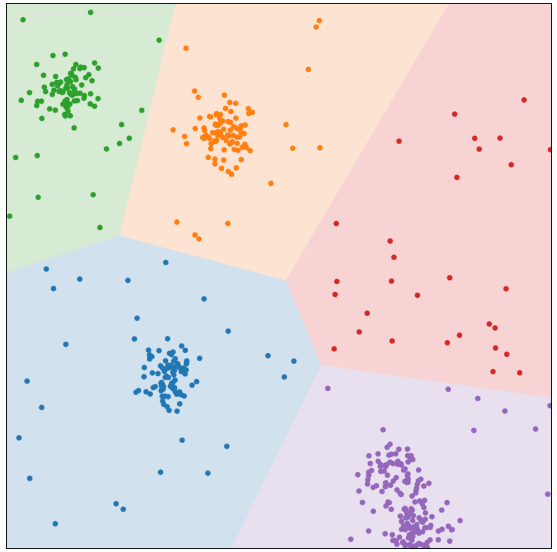
$$m < d$$

A model

f







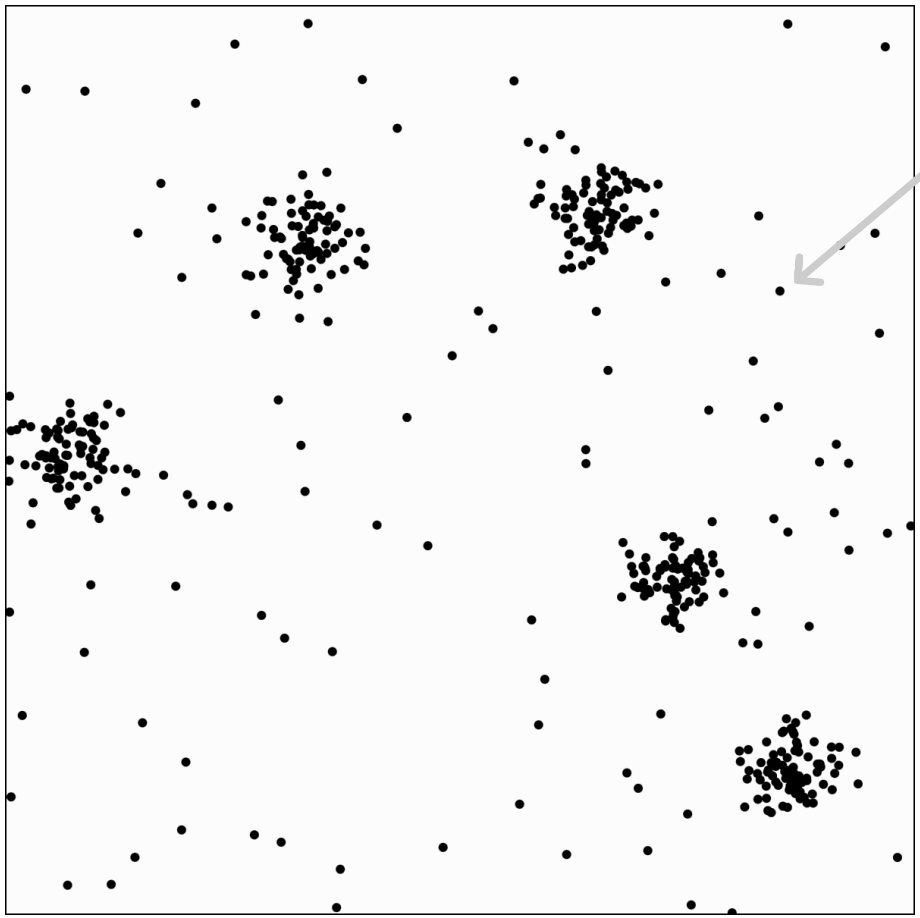
$$\{x^{(1)}\}$$

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

$$\{x^{(3)}\}$$

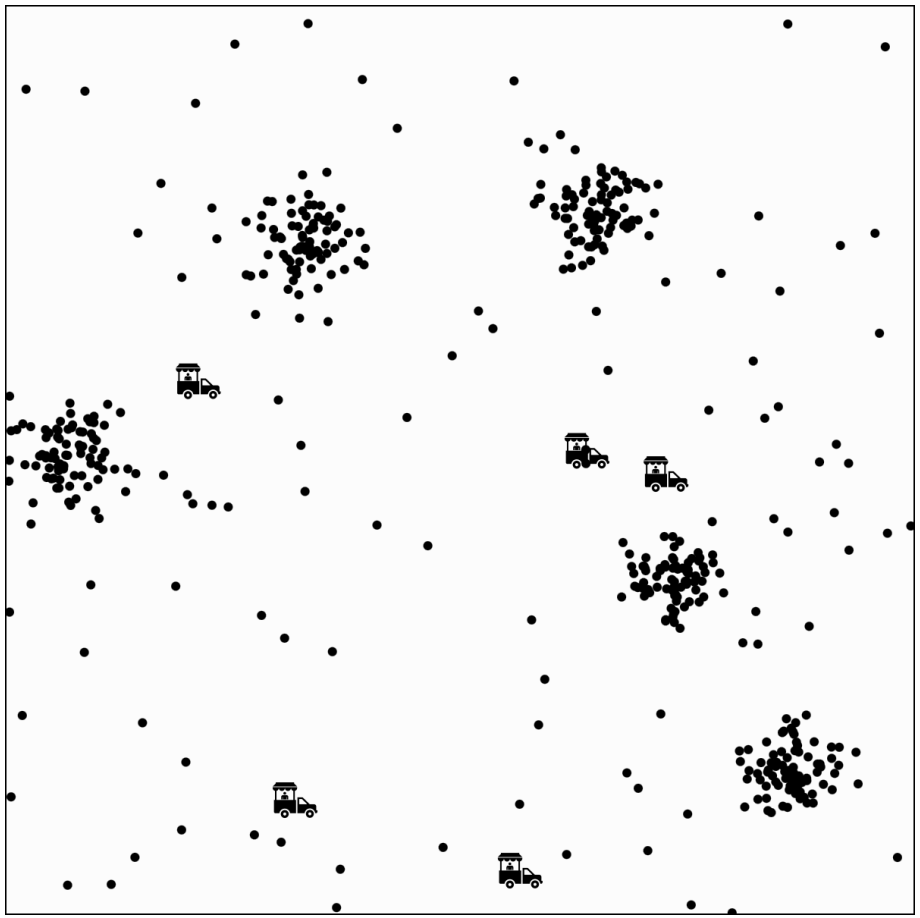
$$\ldots$$

Food-truck placement



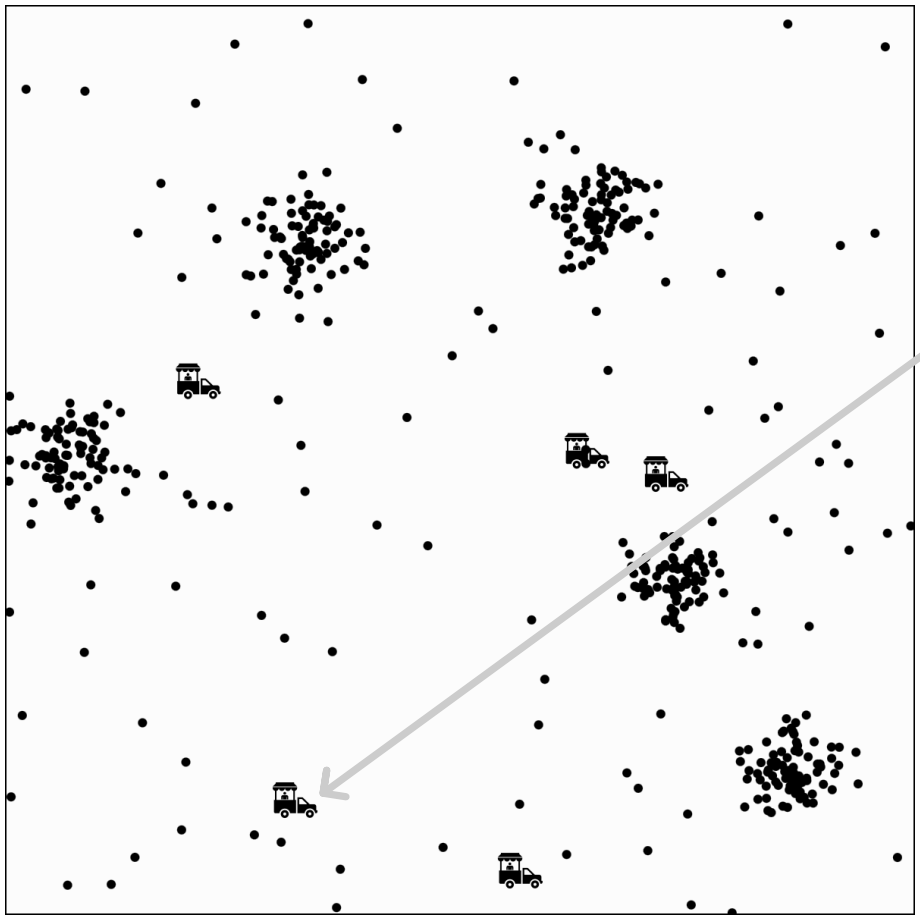
- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$

Food-truck placement



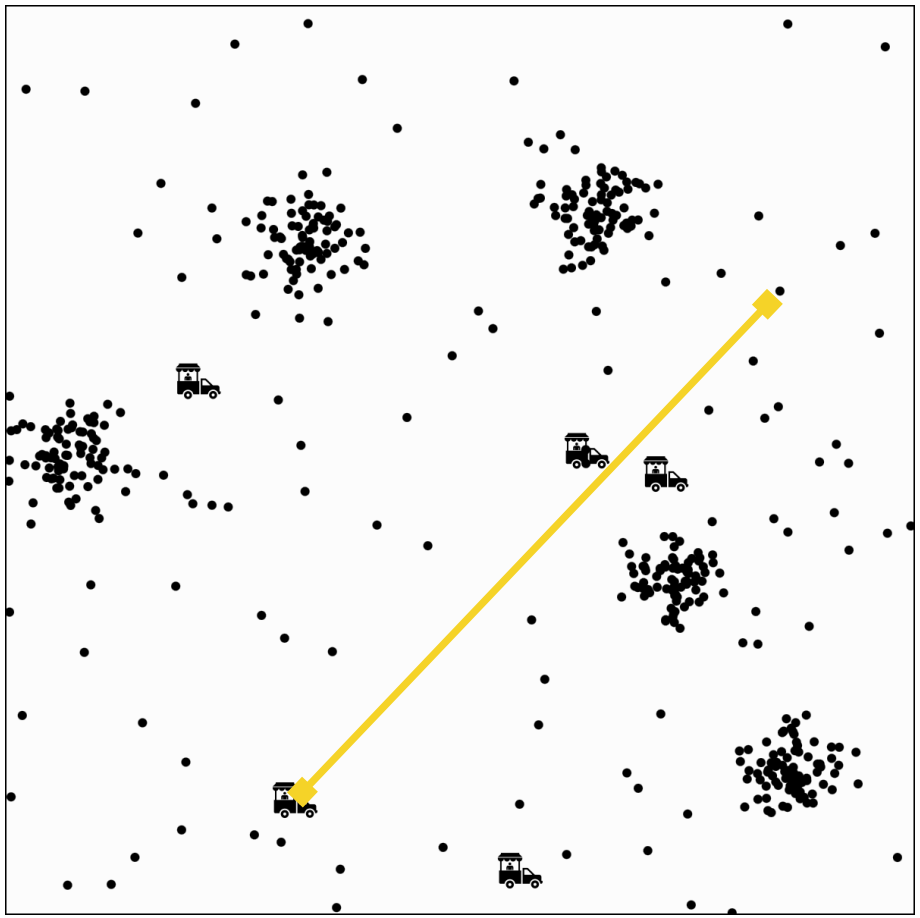
- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?

Food-truck placement



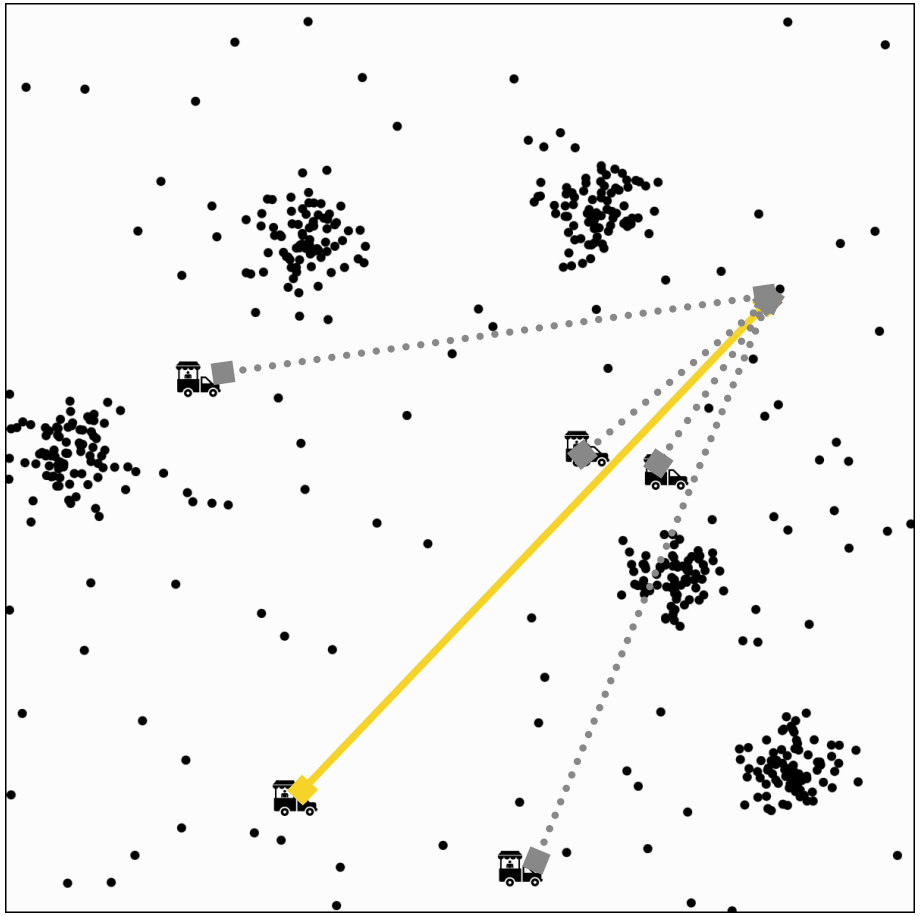
- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?
- Food truck j location $\mu^{(j)}$

Food-truck placement



- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?
- Food truck j location $\mu^{(j)}$
- Loss if i walks to truck j : $\left\|x^{(i)} - \mu^{(j)}\right\|_2^2$

Food-truck placement



- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?
- Food truck j location $\mu^{(j)}$
- Loss if i walks to truck j : $\left\|x^{(i)} - \mu^{(j)}\right\|_2^2$
- Index of the truck where person i walks: $y^{(i)}$
- Person i overall loss:

$$\sum_{j=1}^{k} \mathbf{1}\left\{y^{(i)} = j\right\} \left\|x^{(i)} - \mu^{(j)}\right\|_2^2$$

indicator function, 1 if person i is assigned to truck j, otherwise 0.

k-means objective

what we learn

clustering membership

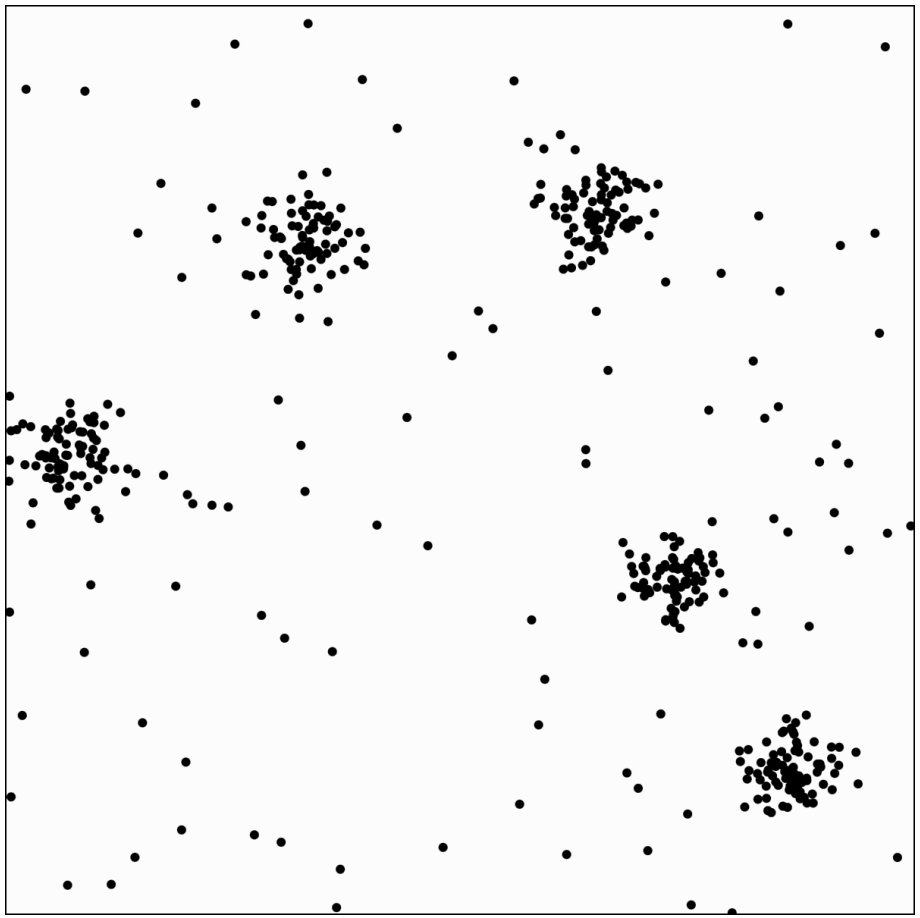
clustering centroid location

$$\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\left\{y^{(i)} = j\right\} \left\|x^{(i)} - \mu^{(j)}\right\|_2^2$$

enumerates over data

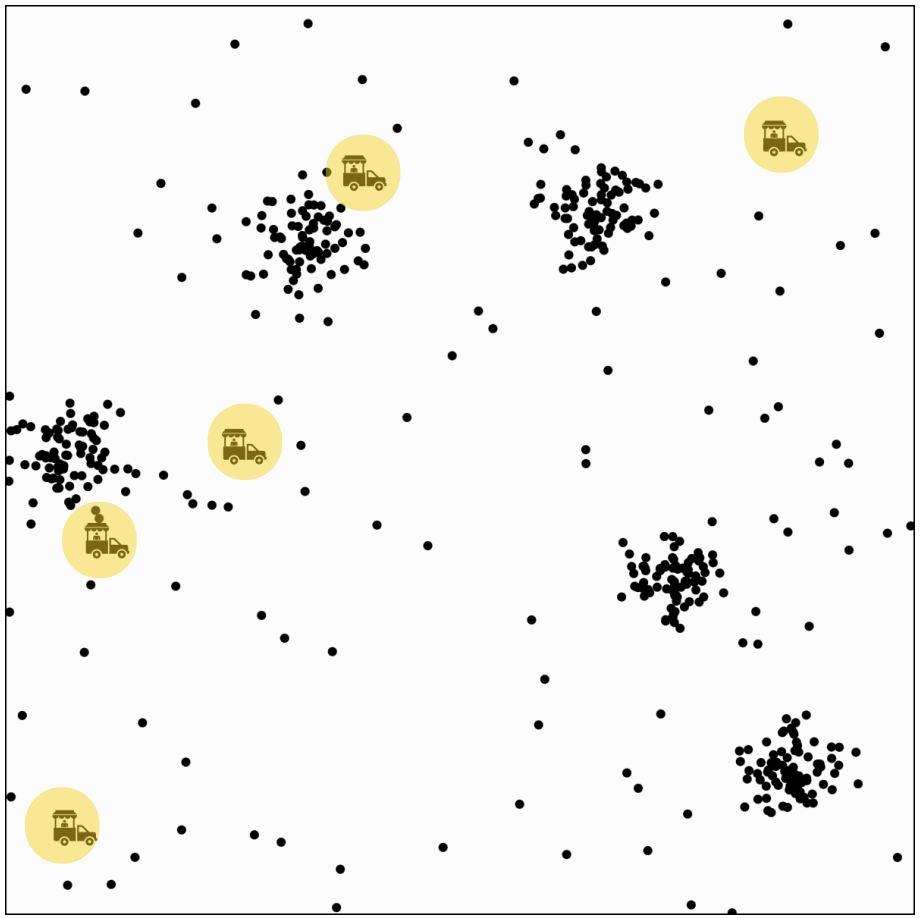
enumerates over cluster

can switch the order $= \sum_{j=1}^{k} \sum_{i=1}^{n} \mathbf{1}\left\{y^{(i)} = j\right\} \left\|x^{(i)} - \mu^{(j)}\right\|_2^2$



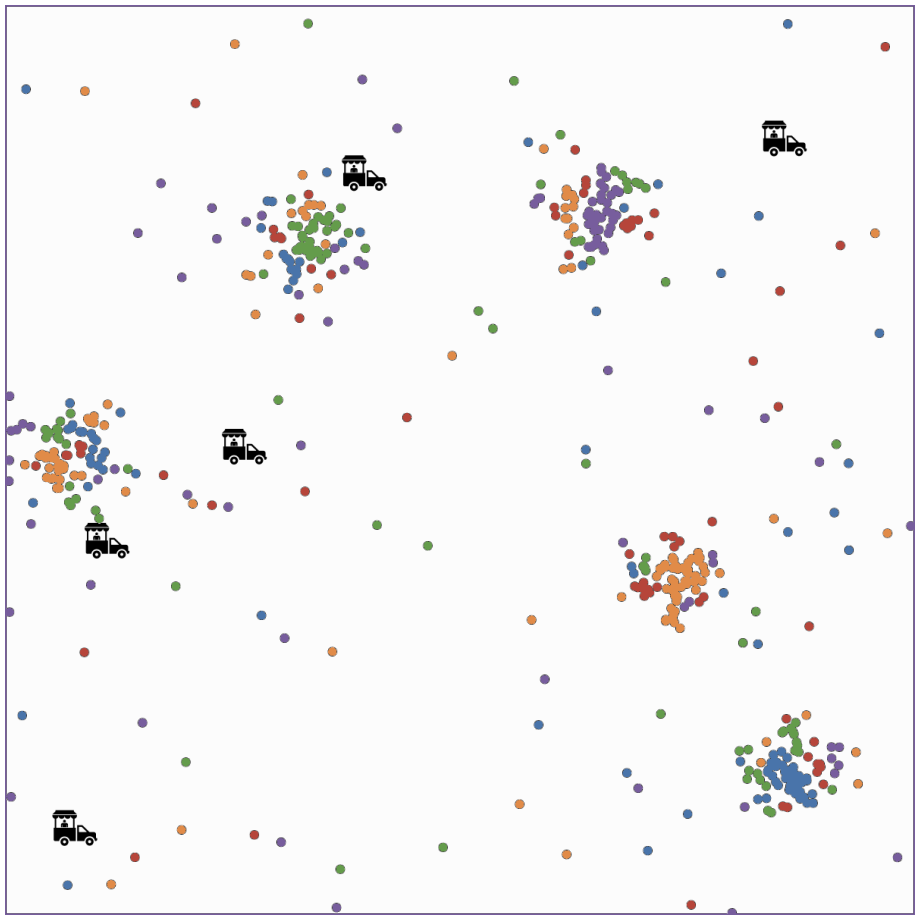
K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization
for $t = 1$ to τ
    $y_{\text{old}} = y$
    for $i = 1$ to n
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    for $j = 1$ to k
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    if $y == y_{\text{old}}$
        break
return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
3 $\quad y_{\text{old}} = y$
4 $\quad$ **for** $i = 1$ to n
5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
6 $\quad$ **for** $j = 1$ to k
7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
8 $\quad$ **if** $y == y_{\text{old}}$
9 $\quad\quad$ break
10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 μ, y = random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

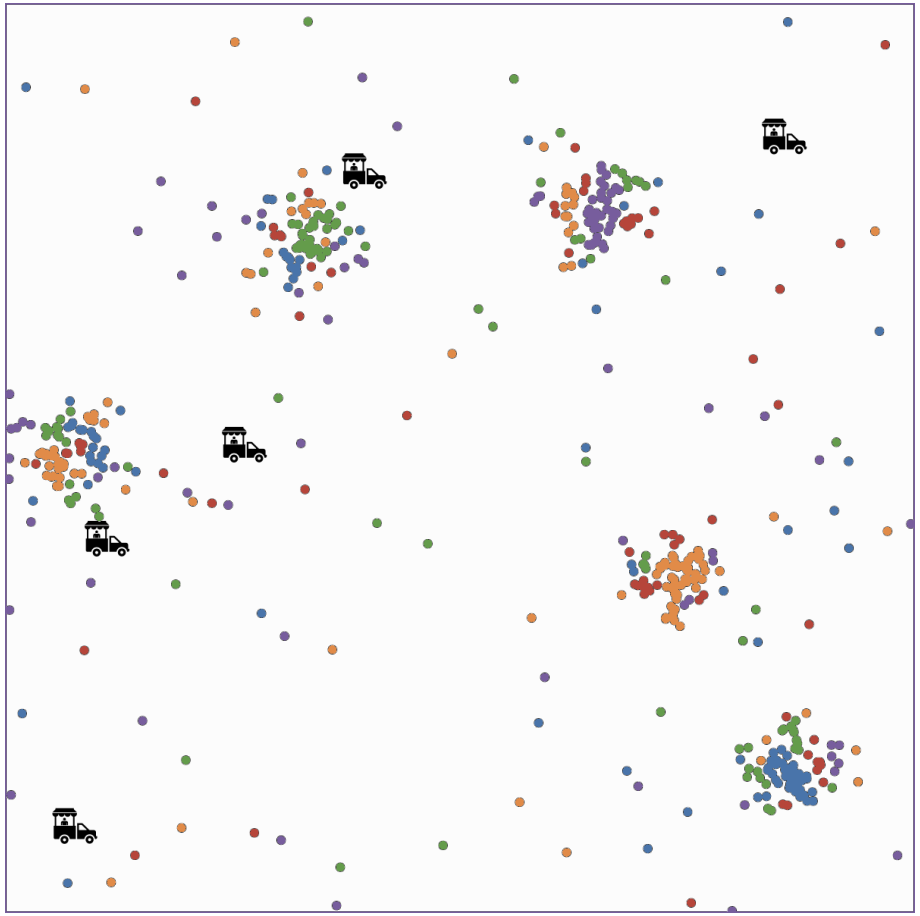
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

for $t = 1$ to τ

  $y_{\text{old}} = y$

  for $i = 1$ to n

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

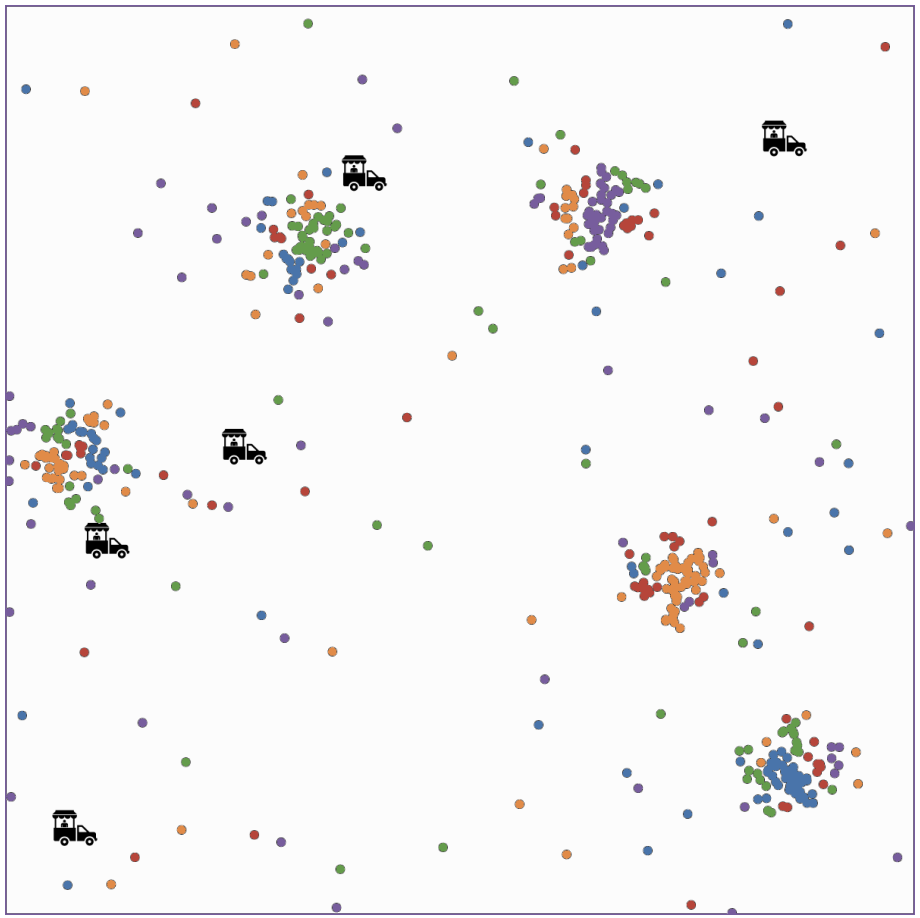
  for $j = 1$ to k

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  if $y == y_{\text{old}}$

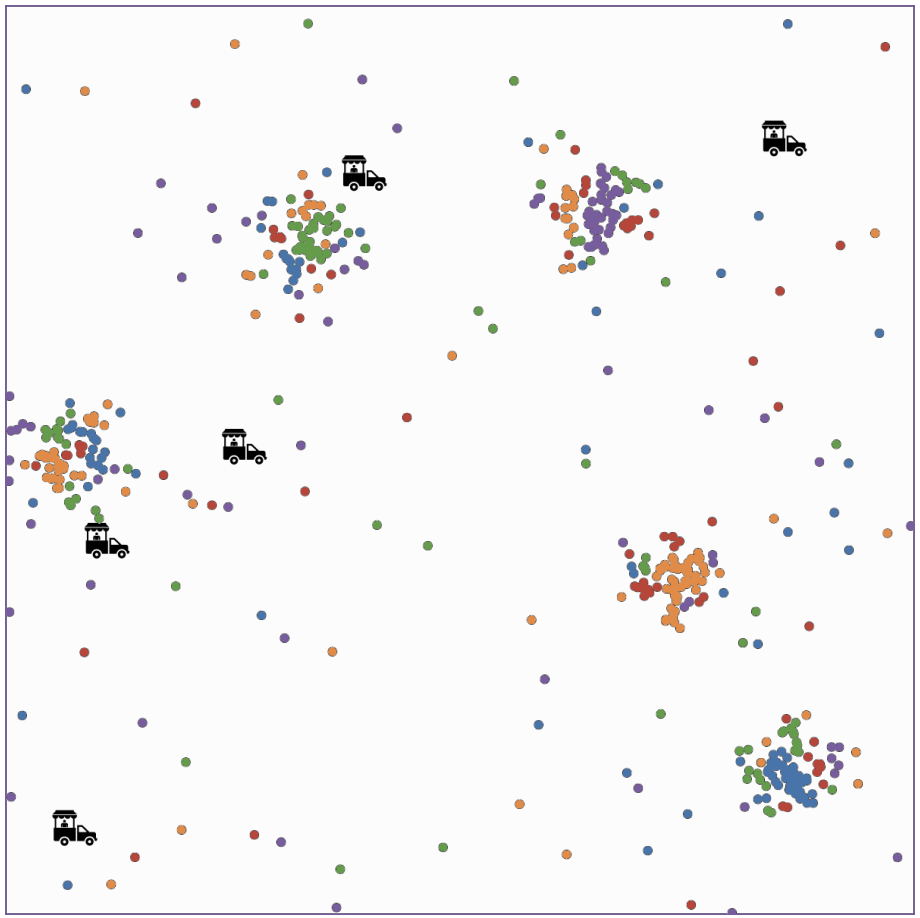
    break

return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
3 $\quad y_{\text{old}} = y$
4 $\quad$ **for** $i = 1$ to n
5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
6 $\quad$ **for** $j = 1$ to k
7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
8 $\quad$ **if** $y == y_{\text{old}}$
9 $\quad\quad$ break
10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

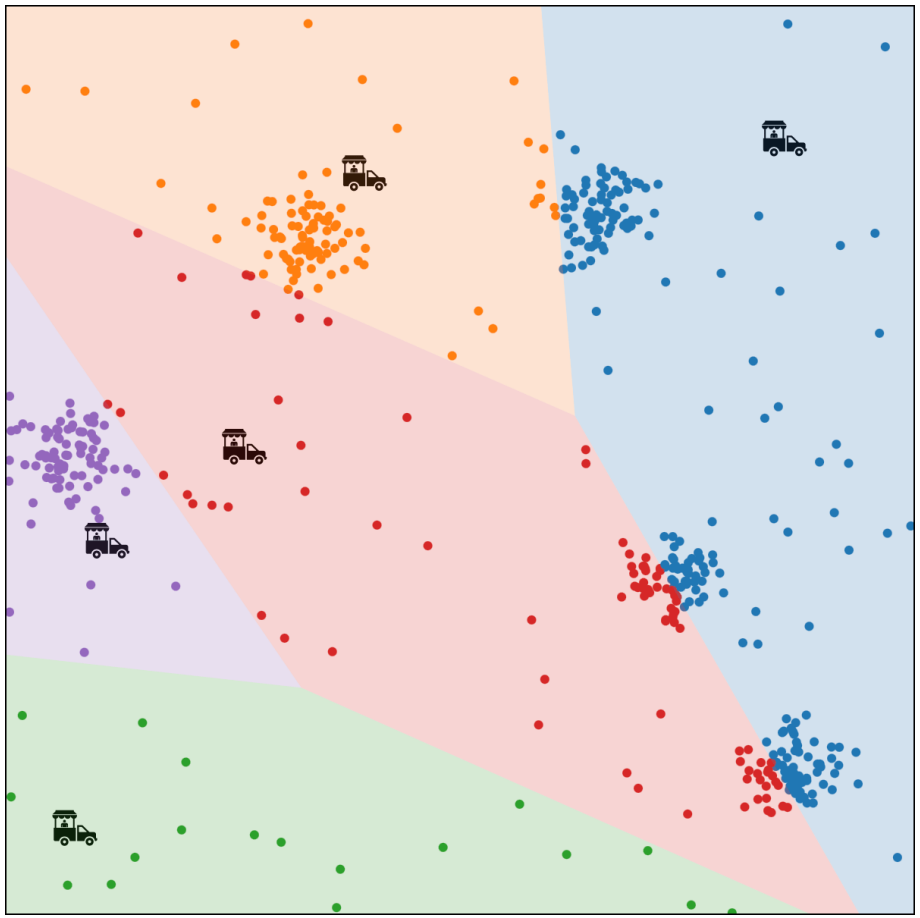
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

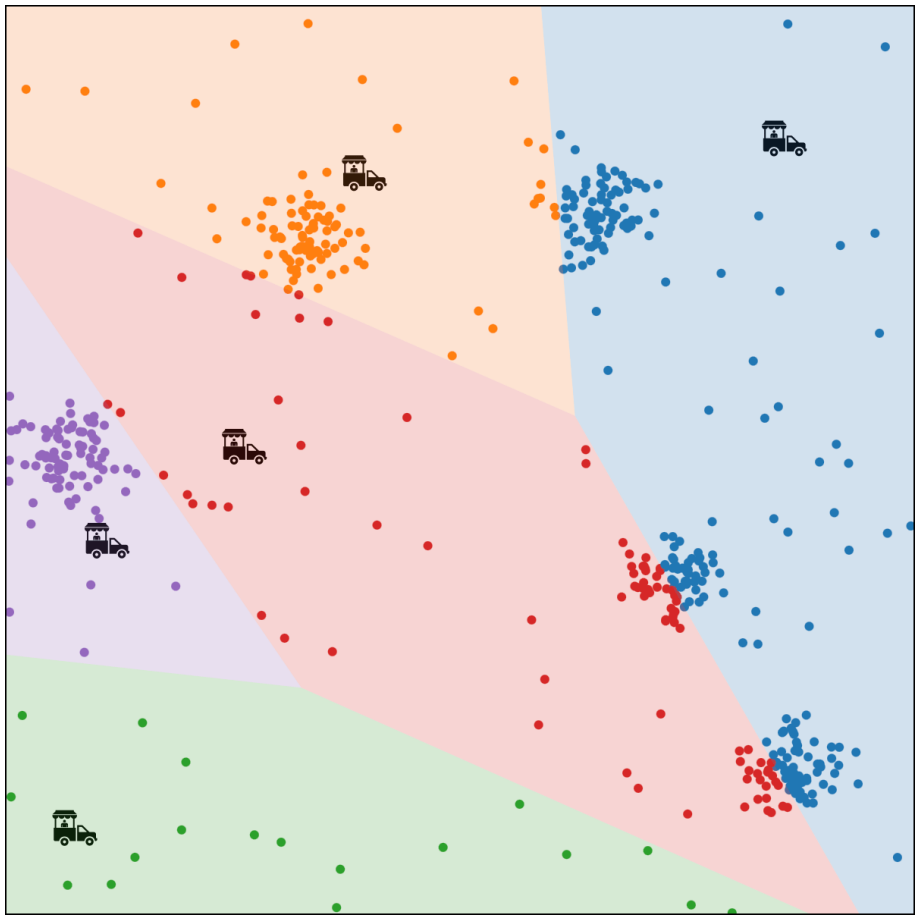
3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left|\left|x^{(i)} - \mu^{(j)}\right|\right|^2$

...

each person i gets assigned to food truck j, color-coded.



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

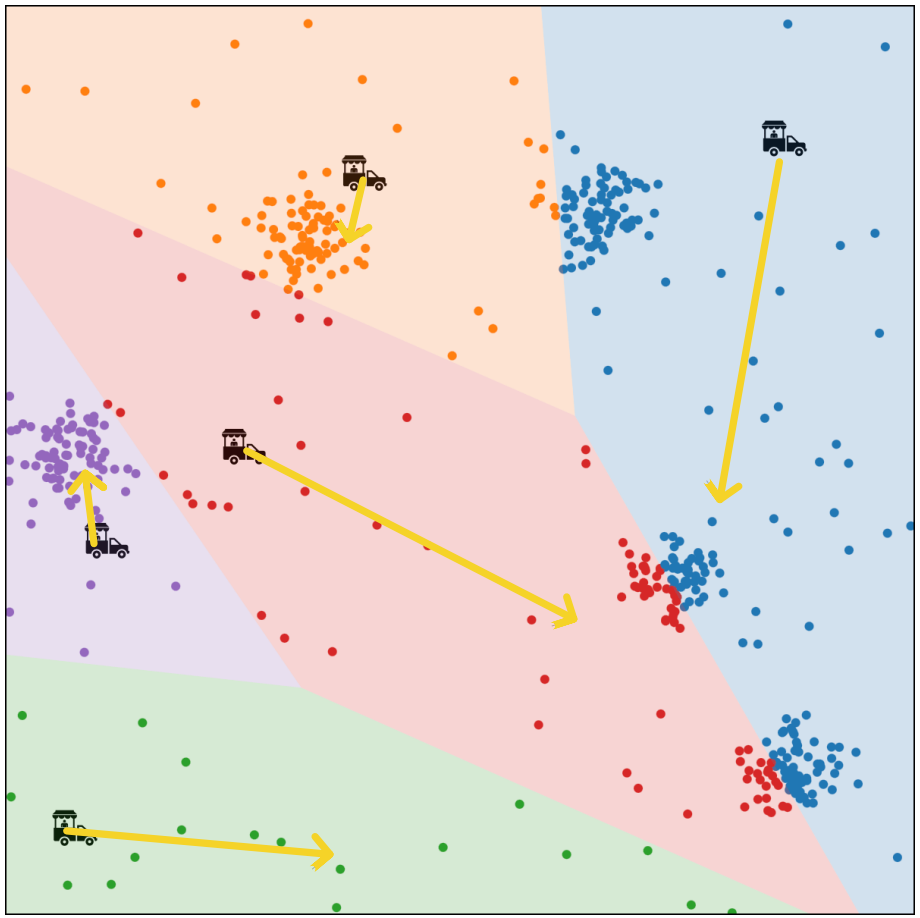
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

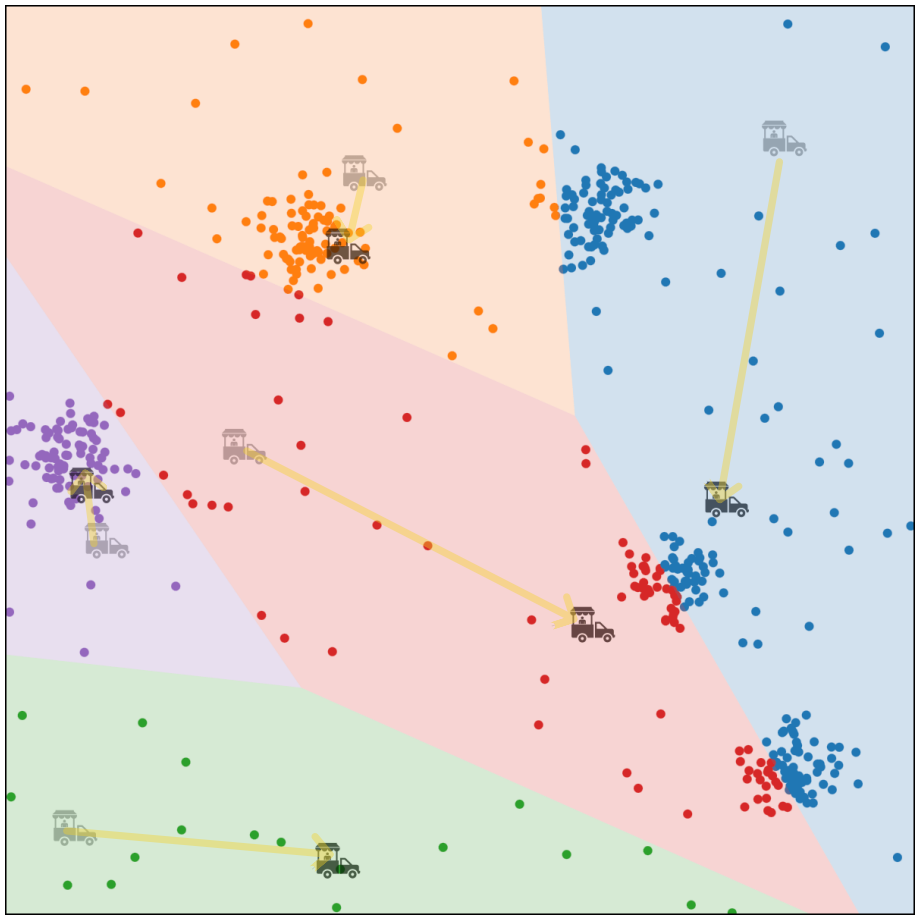
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

...

$N_j = \sum_{i=1}^{n} \mathbf{1}\left\{y^{(i)} = j\right\}$

food truck j gets moved to the "central" location of all ppl assigned to it



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

for $t = 1$ to τ

  $y_{\text{old}} = y$

  for $i = 1$ to n

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

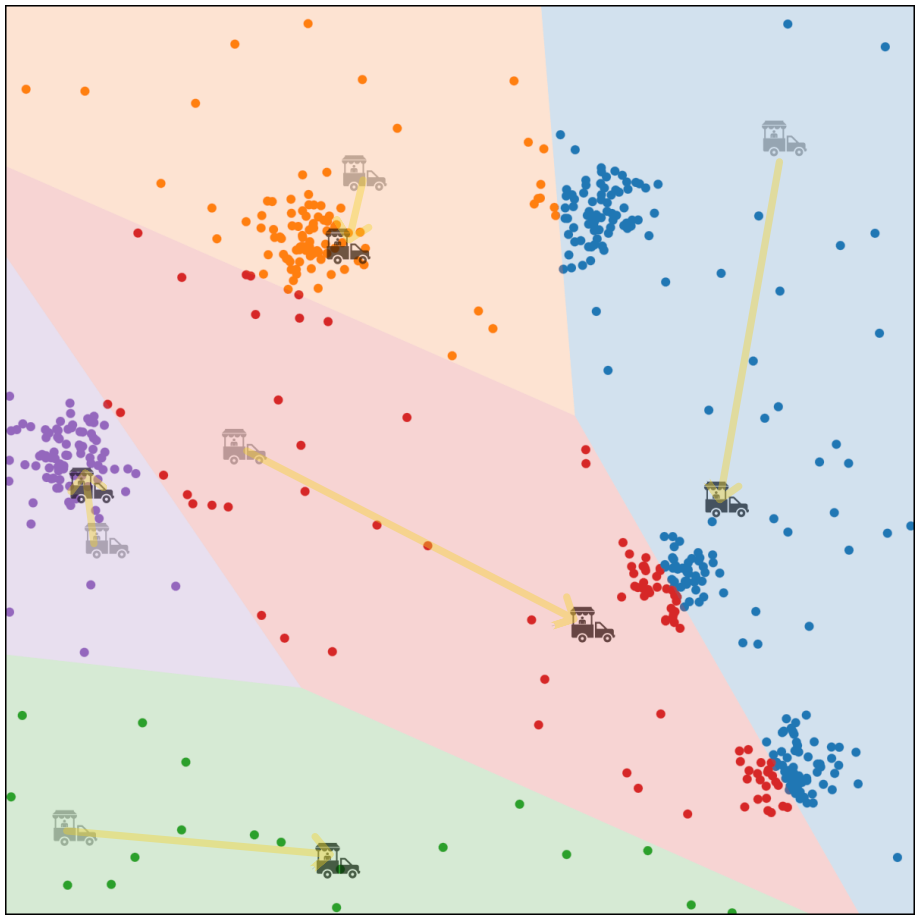
  for $j = 1$ to k

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  if $y == y_{\text{old}}$

    break

return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

for $t = 1$ to τ

  $y_{\text{old}} = y$

  for $i = 1$ to n

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

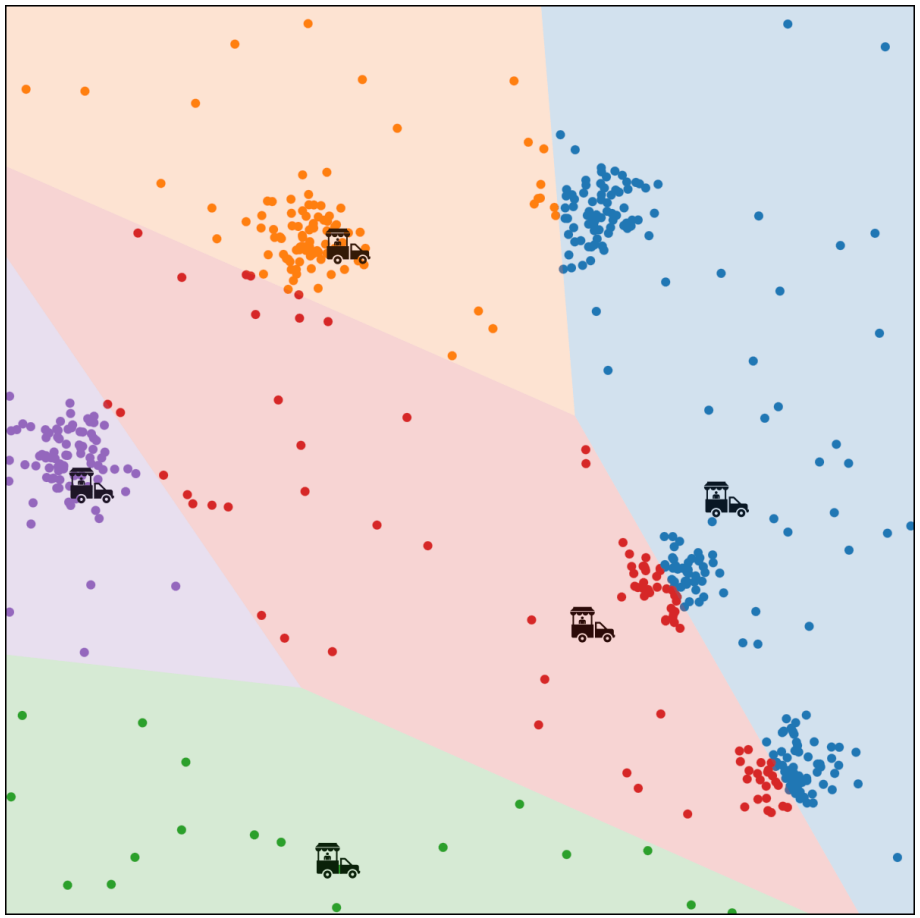
  for $j = 1$ to k

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  if $y == y_{\text{old}}$

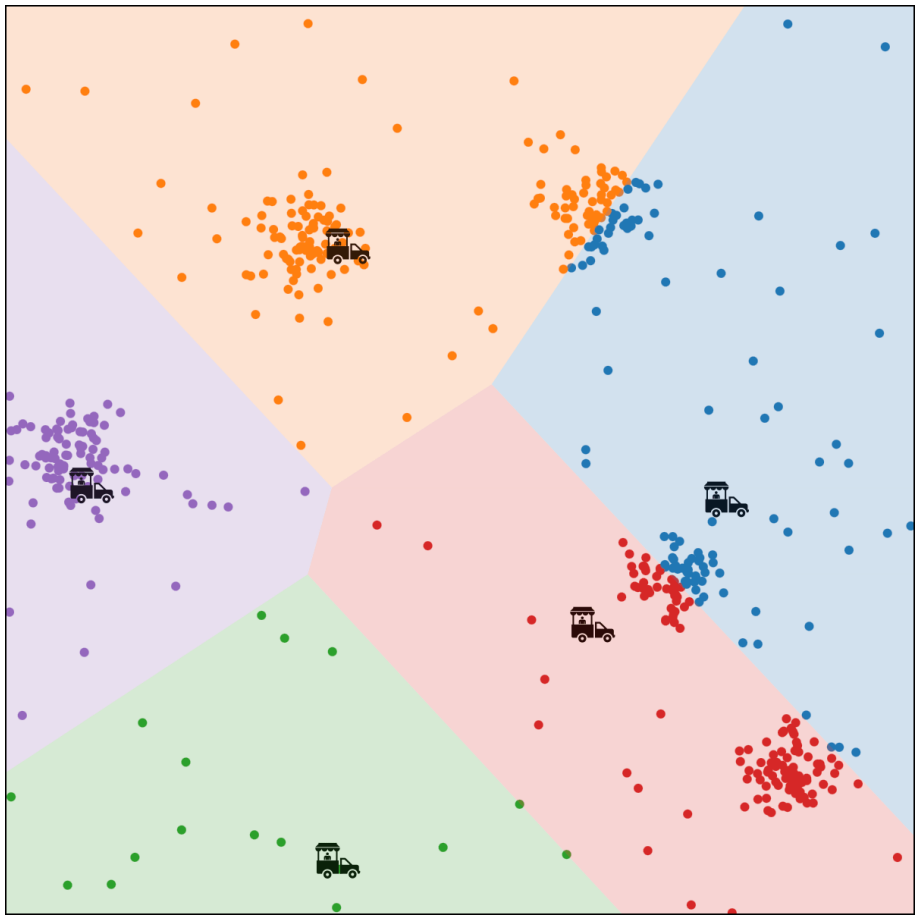
    break

return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization
for $t = 1$ to τ
    $y_{\text{old}} = y$
    for $i = 1$ to n
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    for $j = 1$ to k
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    if $y == y_{\text{old}}$
        break
return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

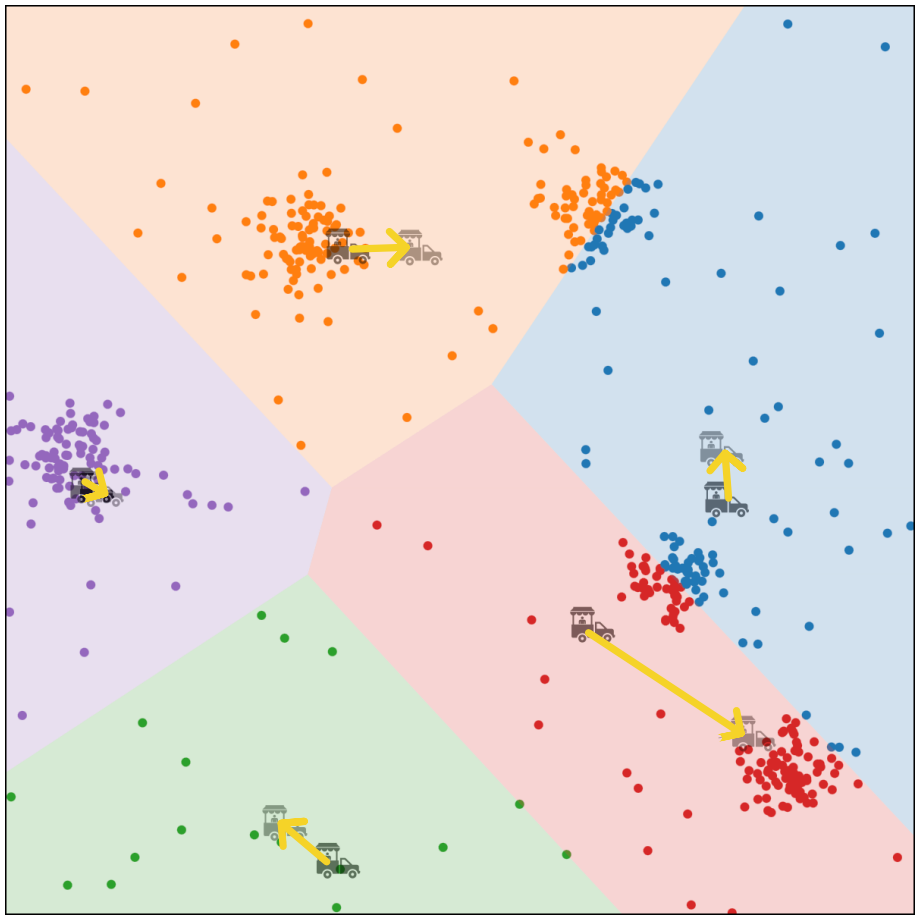
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

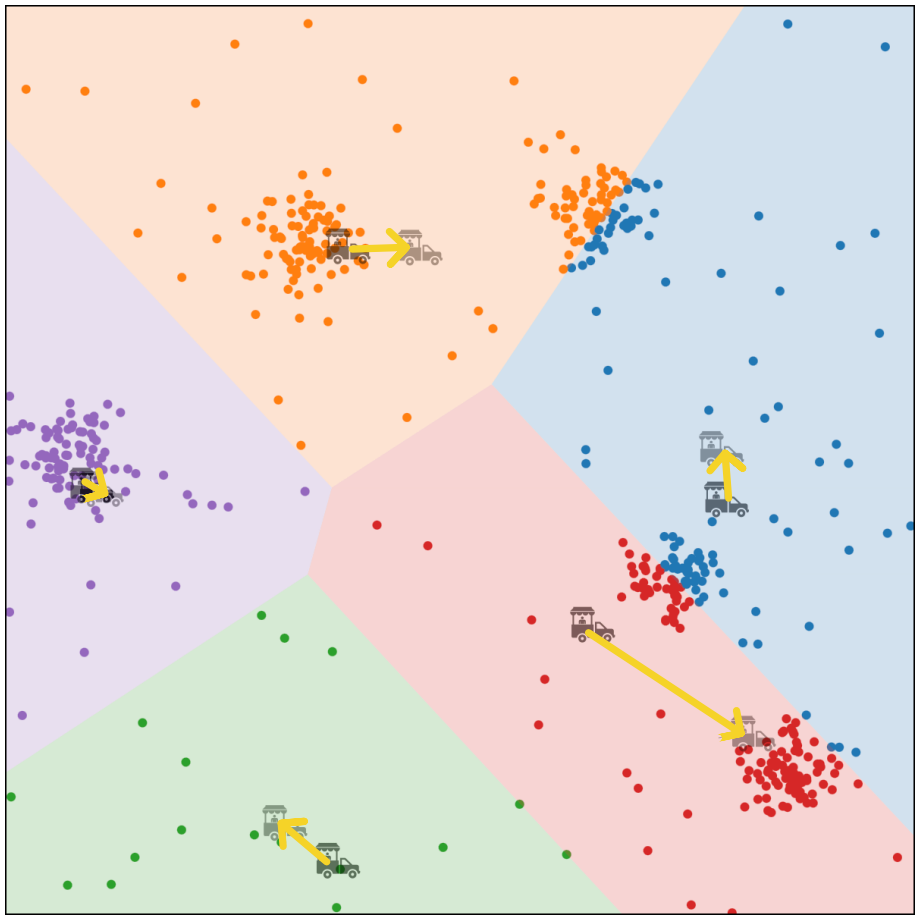
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

for $t = 1$ to τ

  $y_{\text{old}} = y$

  for $i = 1$ to n

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

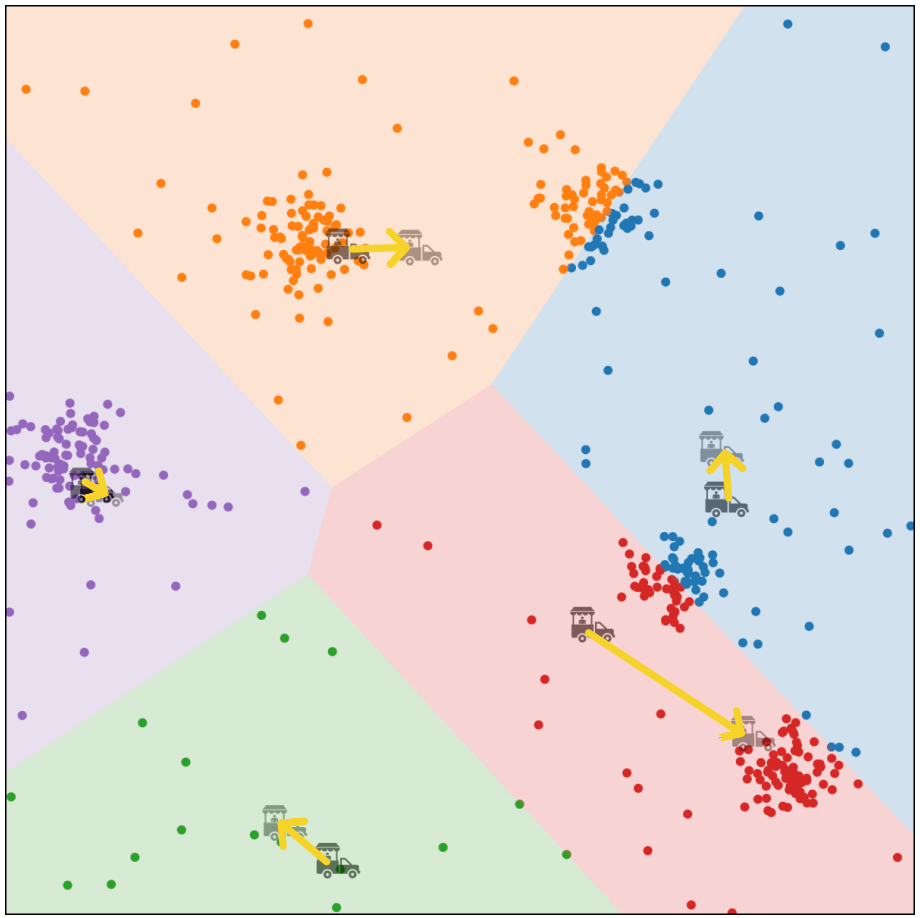
  for $j = 1$ to k

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  if $y == y_{\text{old}}$

    break

return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

for $t = 1$ to τ

    $y_{\text{old}} = y$

    for $i = 1$ to n

        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

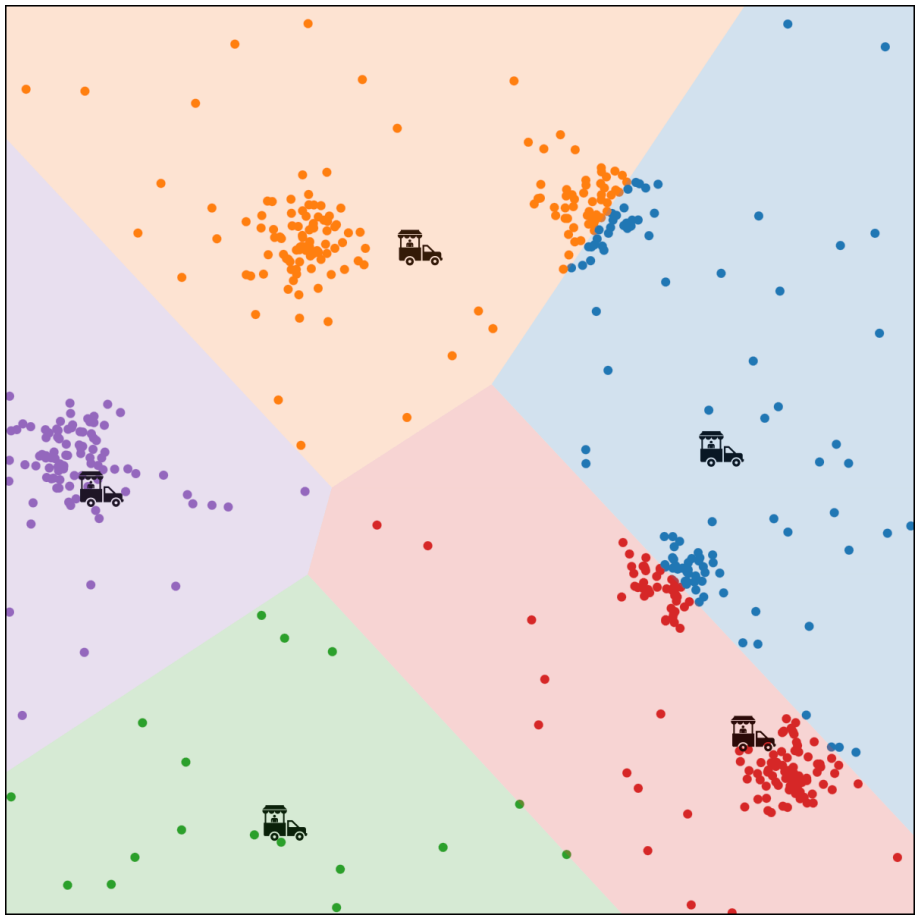
    for $j = 1$ to k

        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

    if $y == y_{\text{old}}$

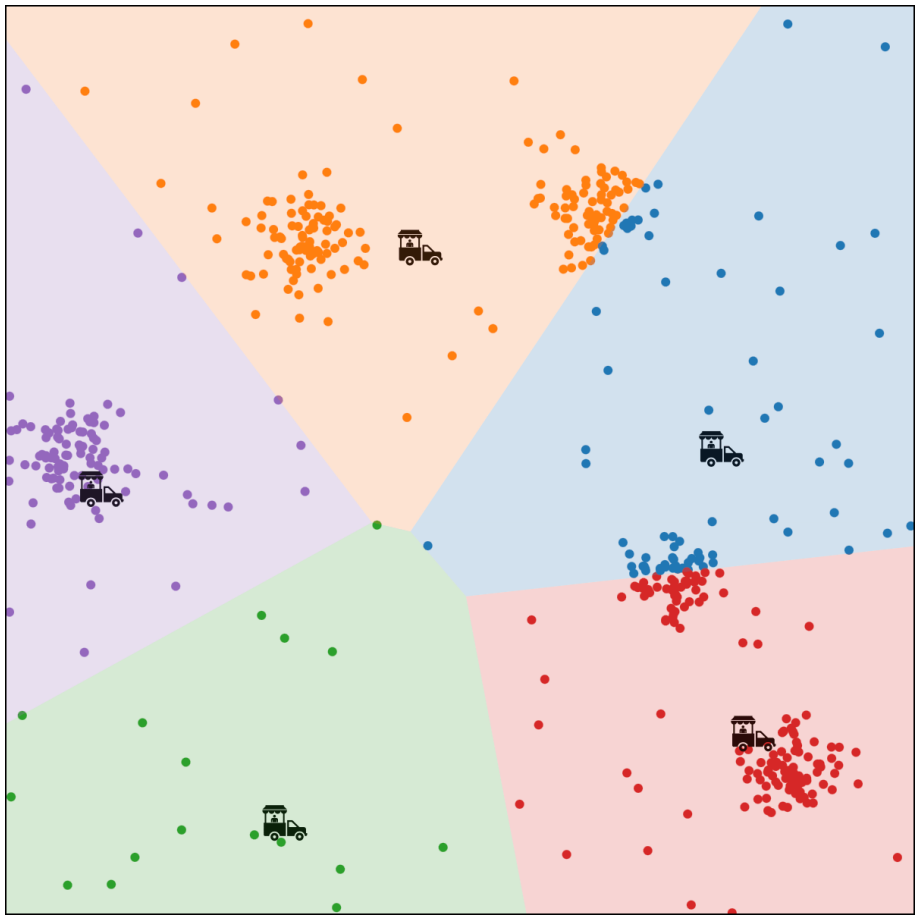
        break

return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization
for $t = 1$ to τ
    $y_{\text{old}} = y$
    for $i = 1$ to n
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    for $j = 1$ to k
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    if $y == y_{\text{old}}$
        break
return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

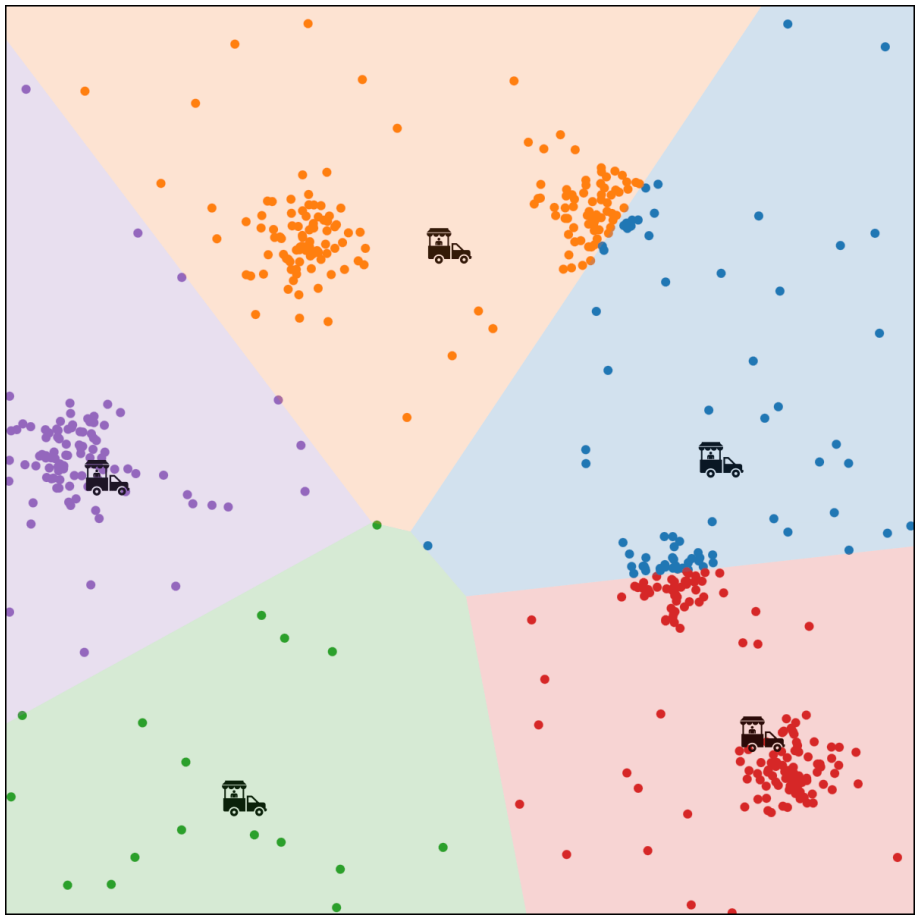
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

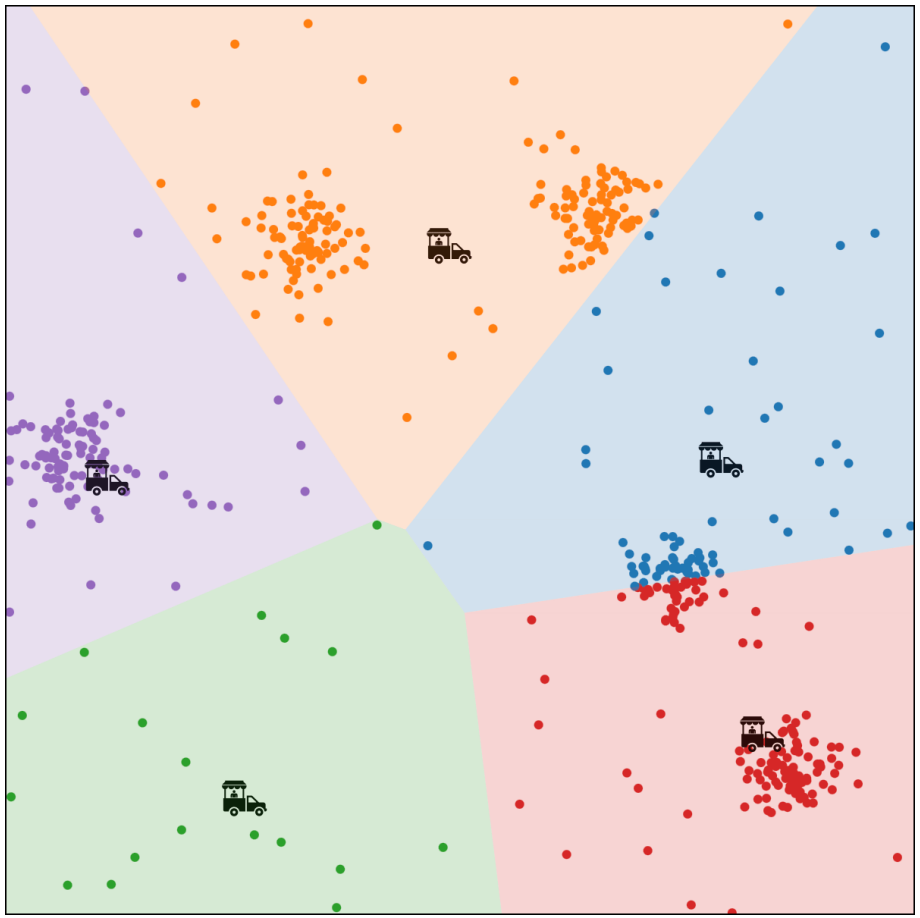
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

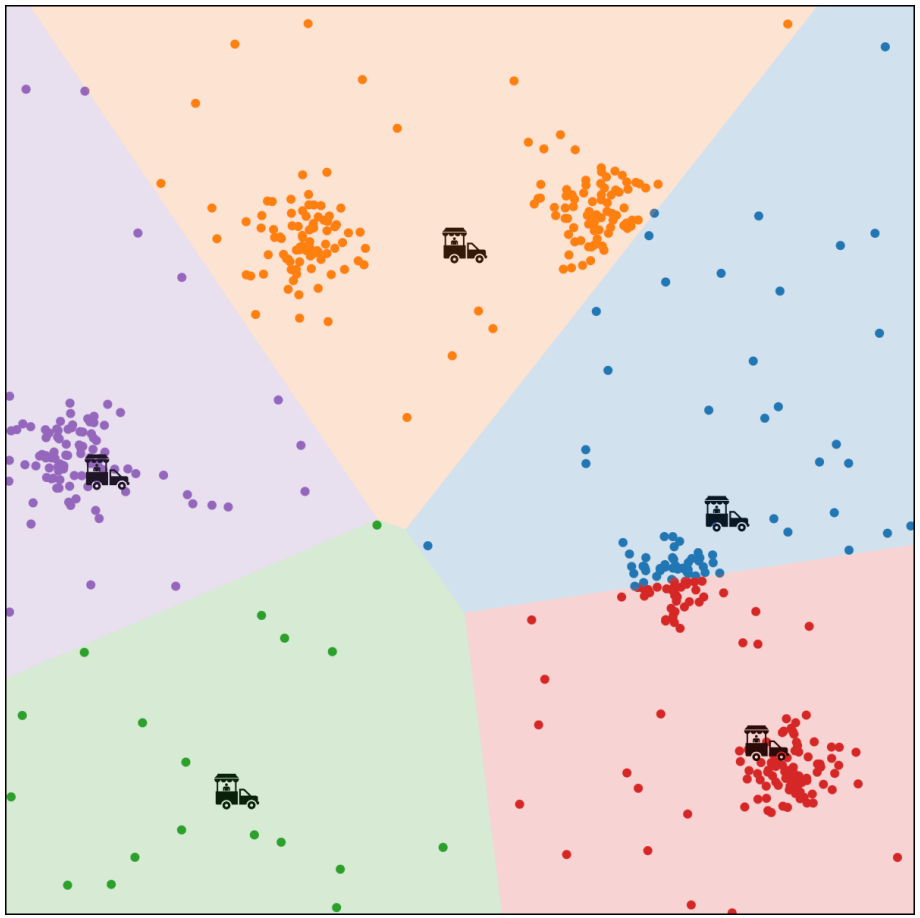
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

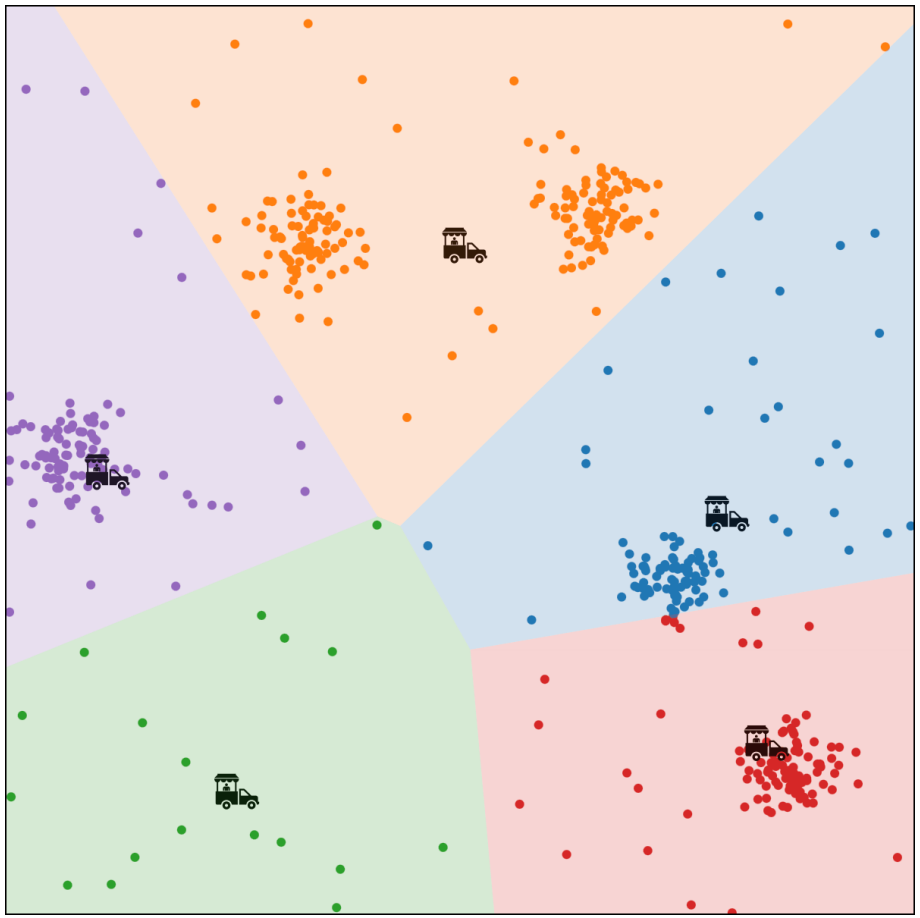
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

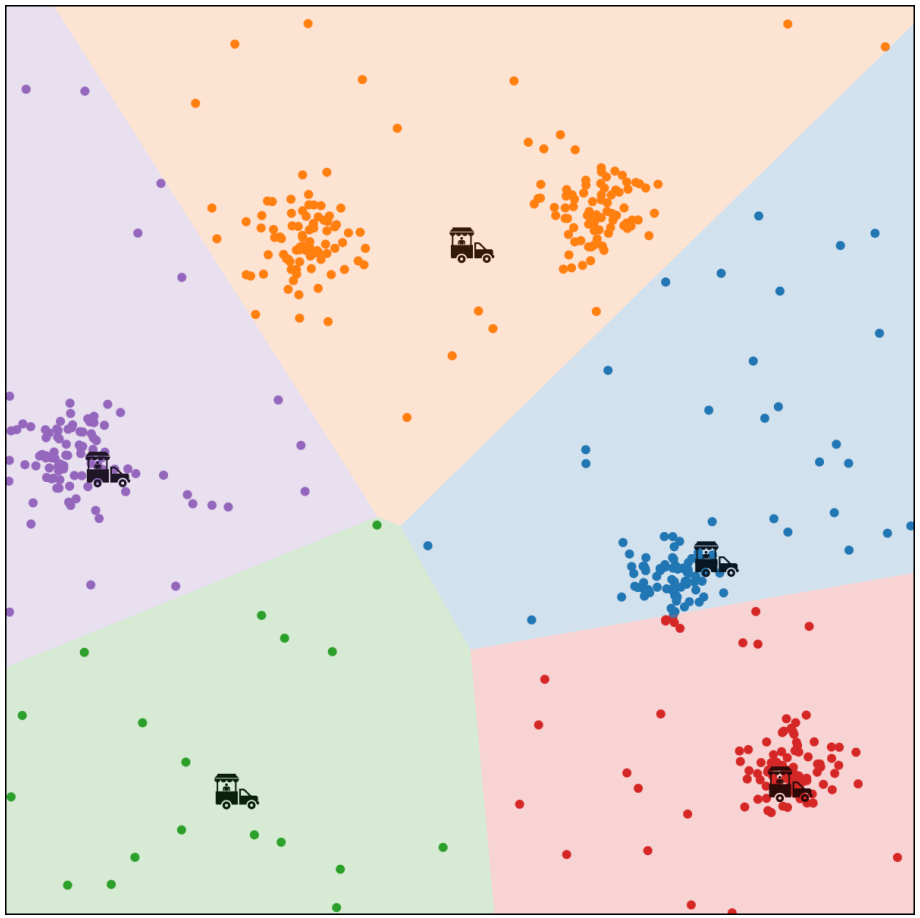
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

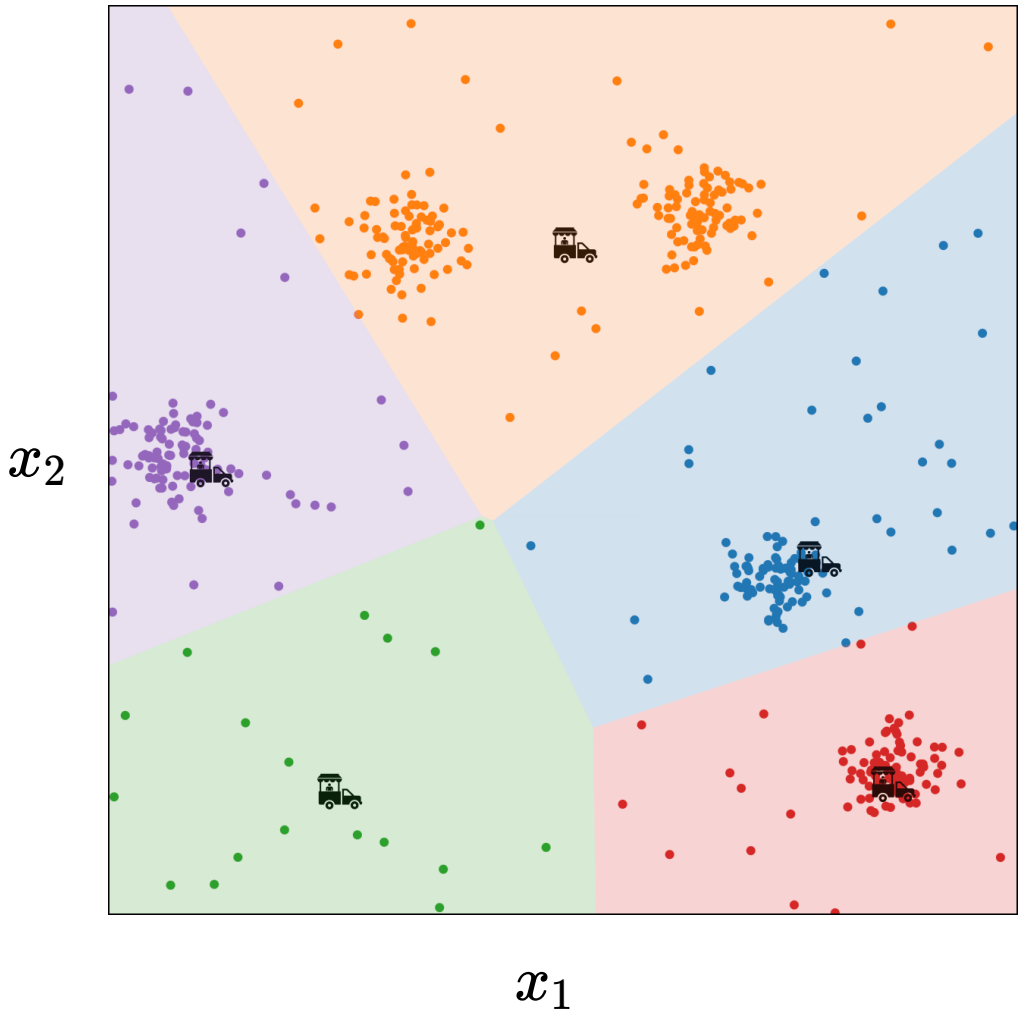
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

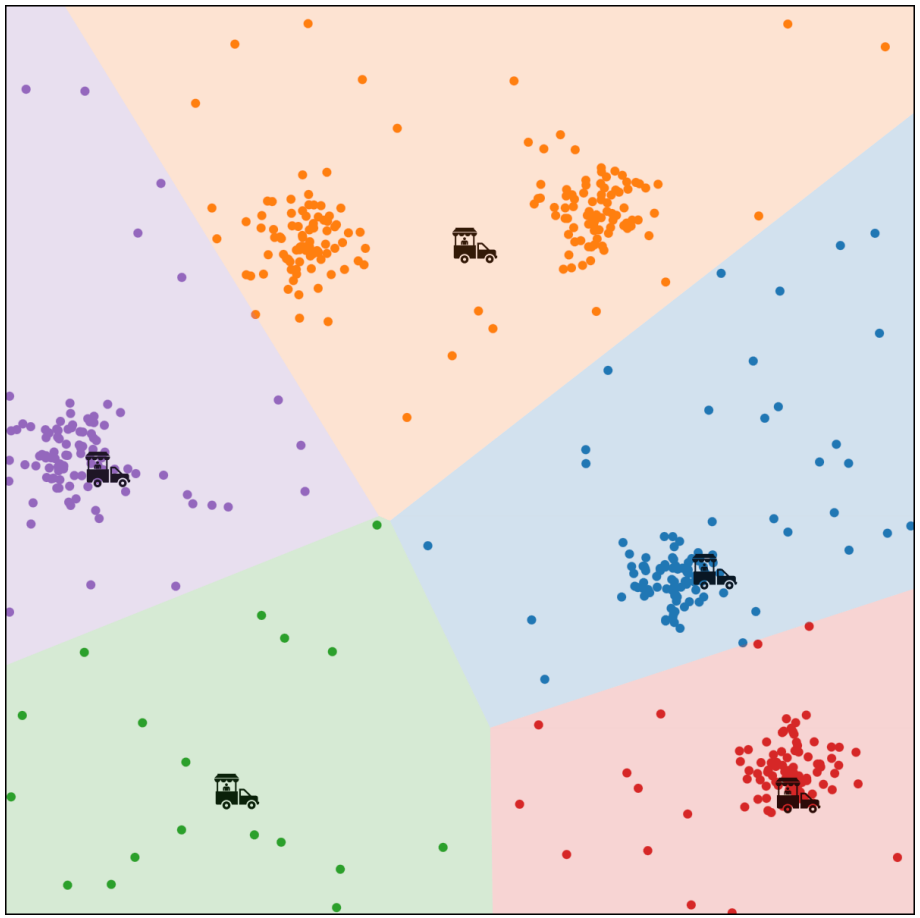
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

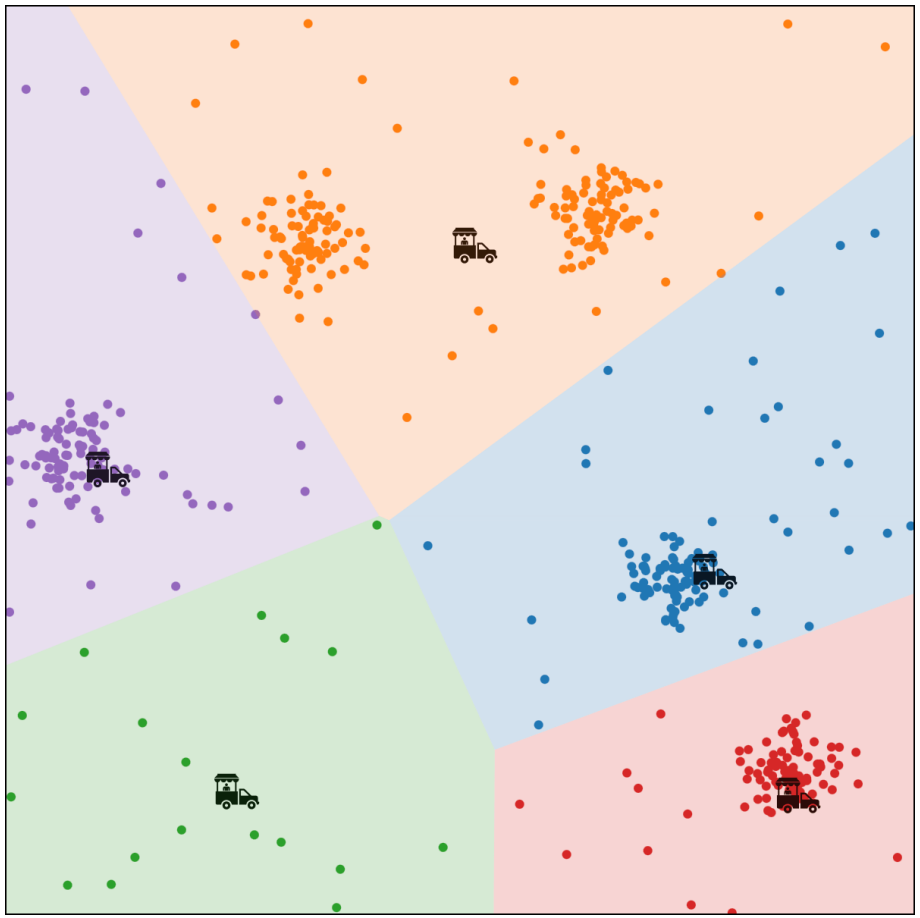
9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
3 $\quad y_{\text{old}} = y$
4 $\quad$ **for** $i = 1$ to n
5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
6 $\quad$ **for** $j = 1$ to k
7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
8 $\quad$ **if** $y == y_{\text{old}}$
9 $\quad\quad$ break
10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

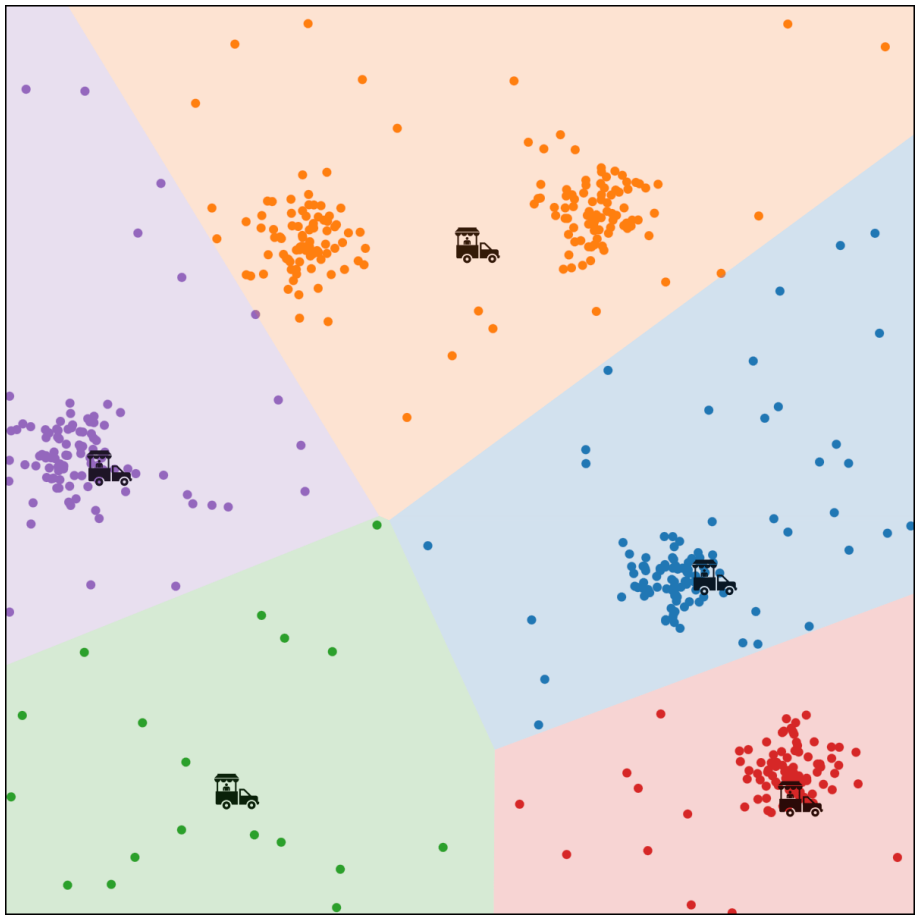
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

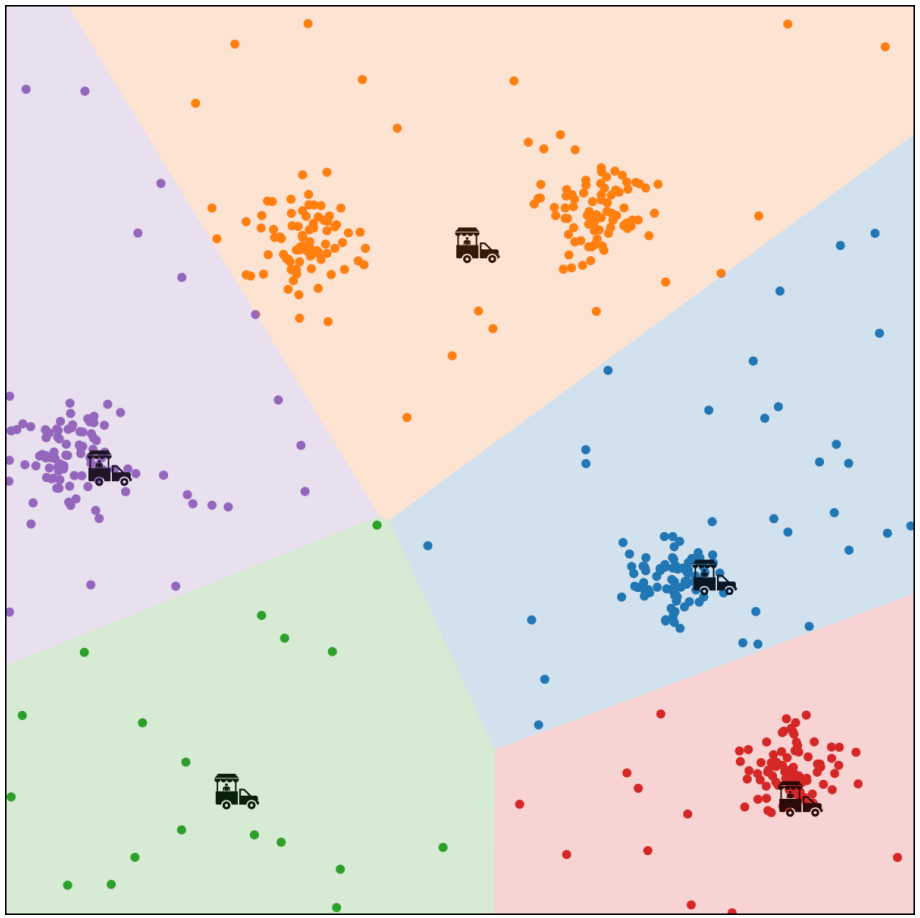
9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
3 $\quad y_{\text{old}} = y$
4 $\quad$ **for** $i = 1$ to n
5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
6 $\quad$ **for** $j = 1$ to k
7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
8 $\quad$ **if** $y == y_{\text{old}}$
9 $\quad\quad$ break
10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

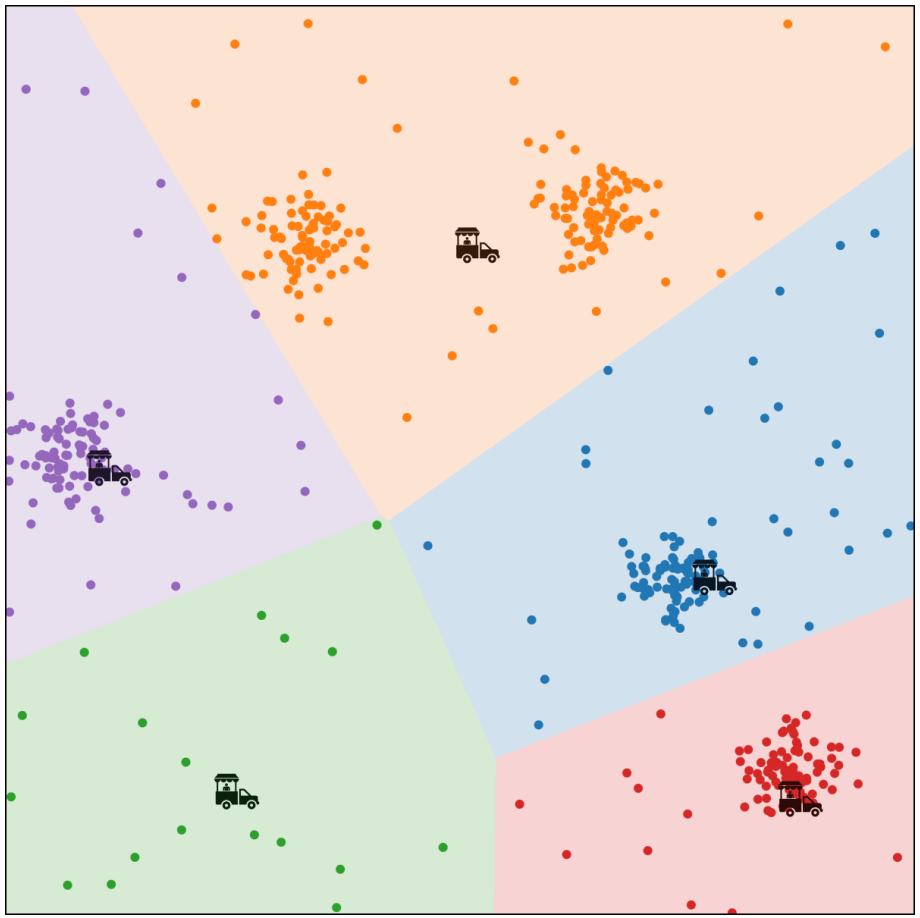
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

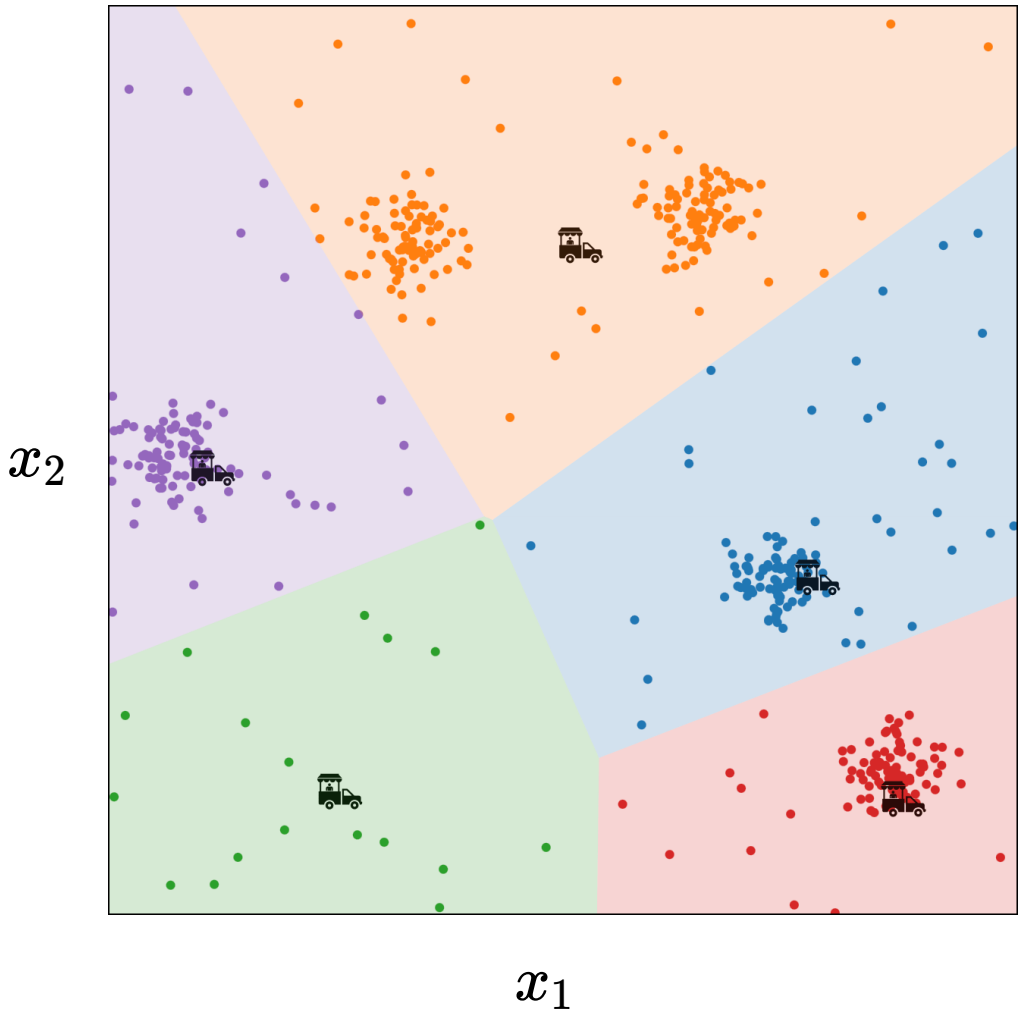
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

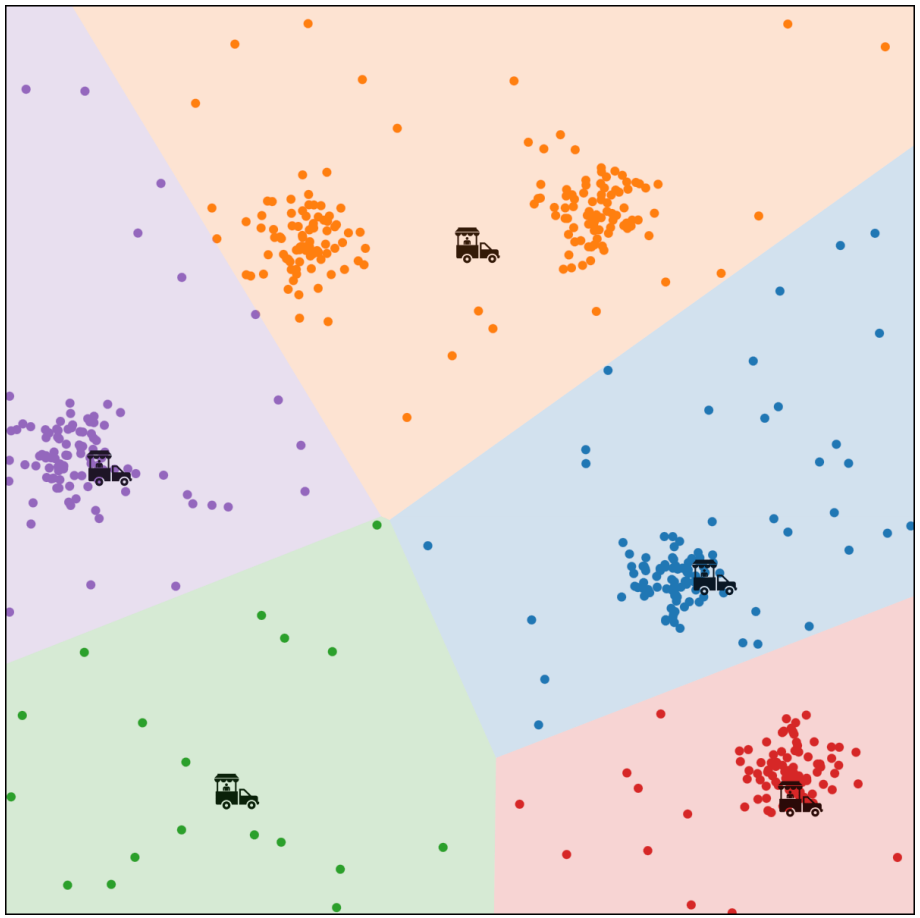
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to n

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

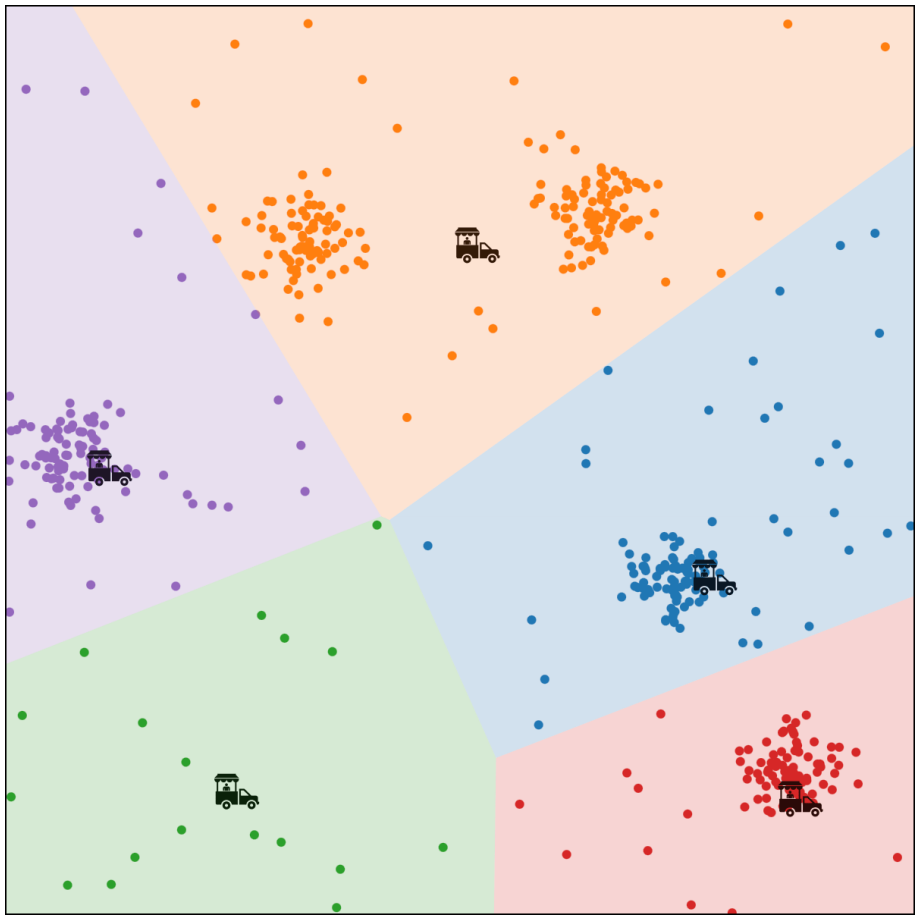
6 $\quad$ **for** $j = 1$ to k

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return μ, y



K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

for $t = 1$ to τ

  $y_{\text{old}} = y$

  for $i = 1$ to n

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

  for $j = 1$ to k

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  if $y == y_{\text{old}}$

    break

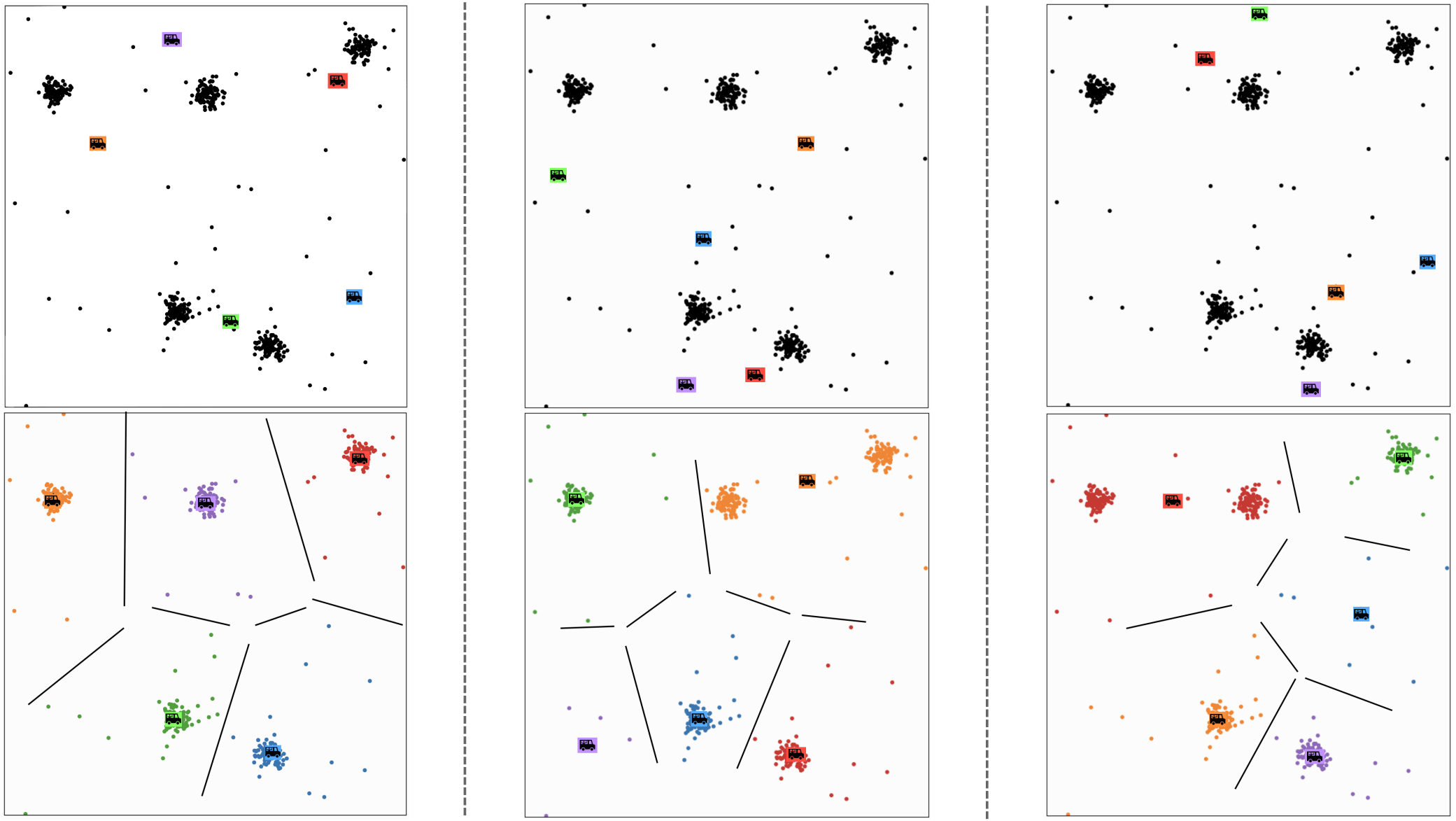
return μ, y

- if run for enough outer iterations, the algorithm will converge to a local minimum of the k-means objective.
- that local minimum could be bad!

K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

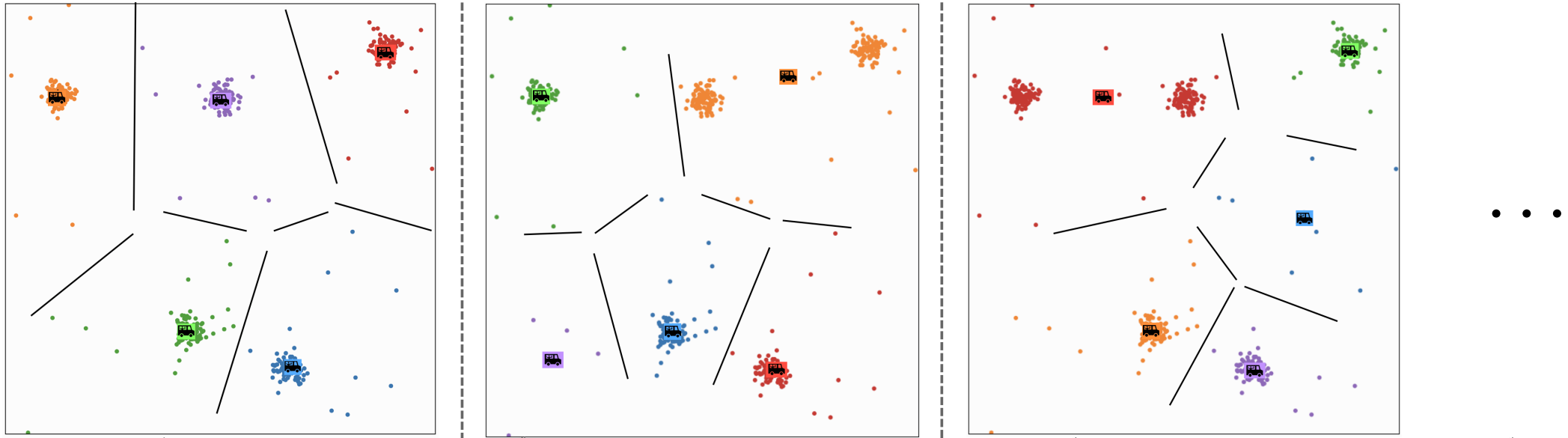
$\mu, y =$ random initialization
for $t = 1$ to τ
    $y_{\text{old}} = y$
    for $i = 1$ to n
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    for $j = 1$ to k
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    if $y == y_{\text{old}}$
        break
return μ, y

Effect of initialization



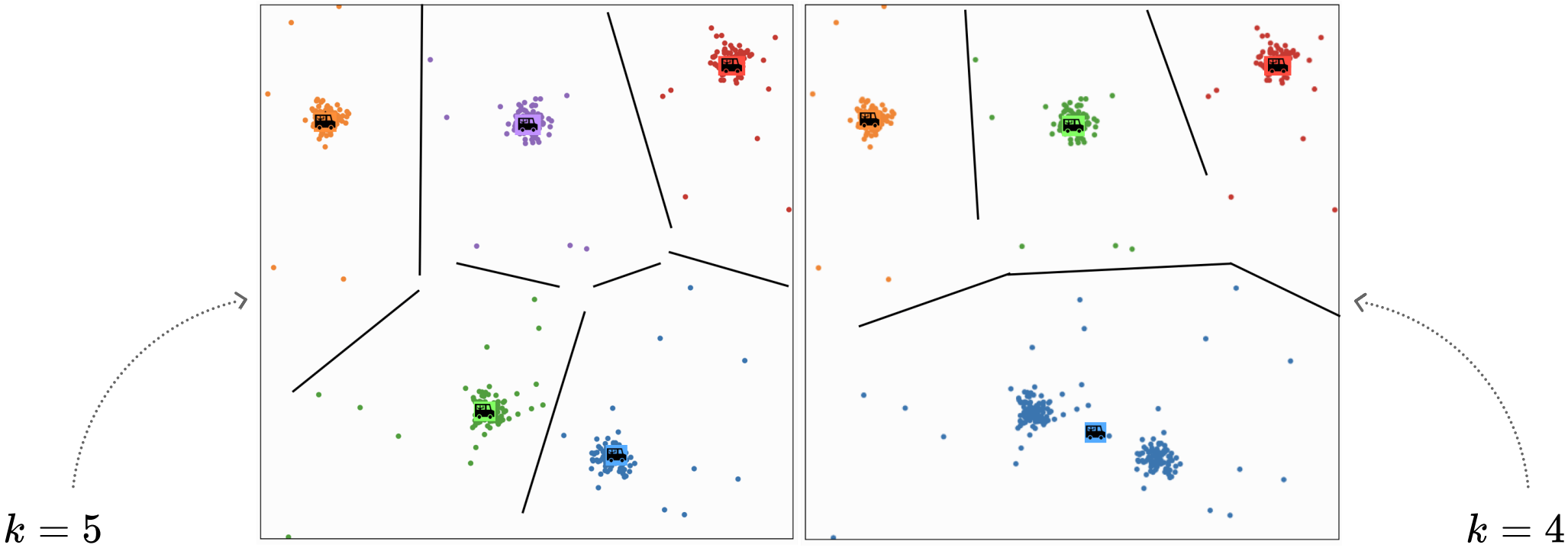
Effect of initialization - one remedy:

Run random initializations multiple times,



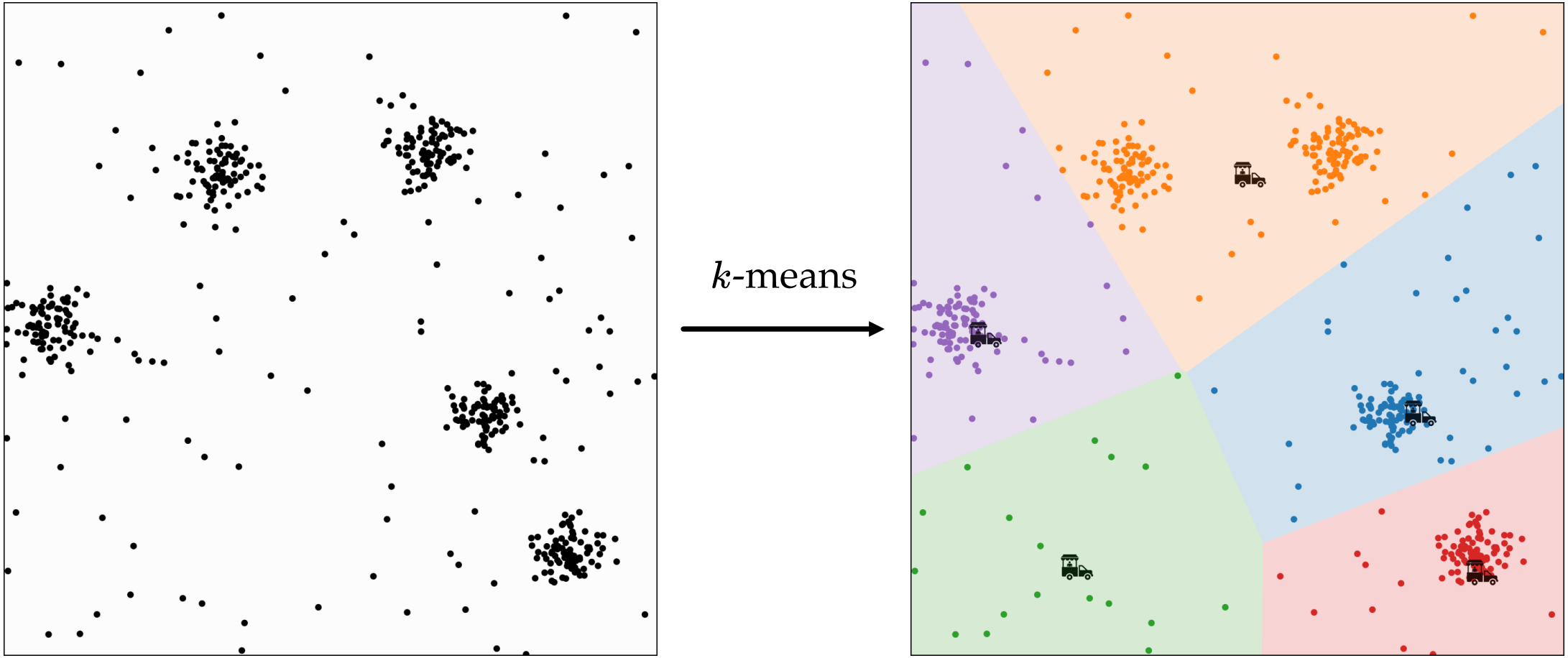
Compare their k-means objective values, choose the lowest one

Effect of k

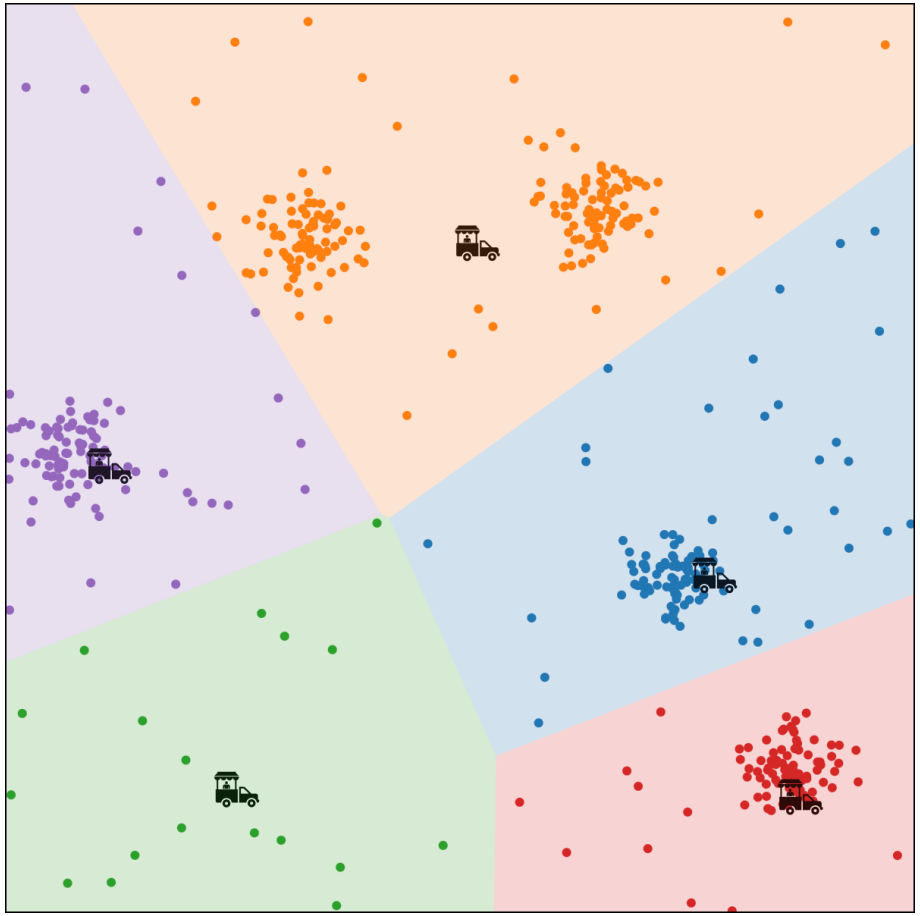


- Choosing of k is a judgment call. Cross-validation.

Compare to classification

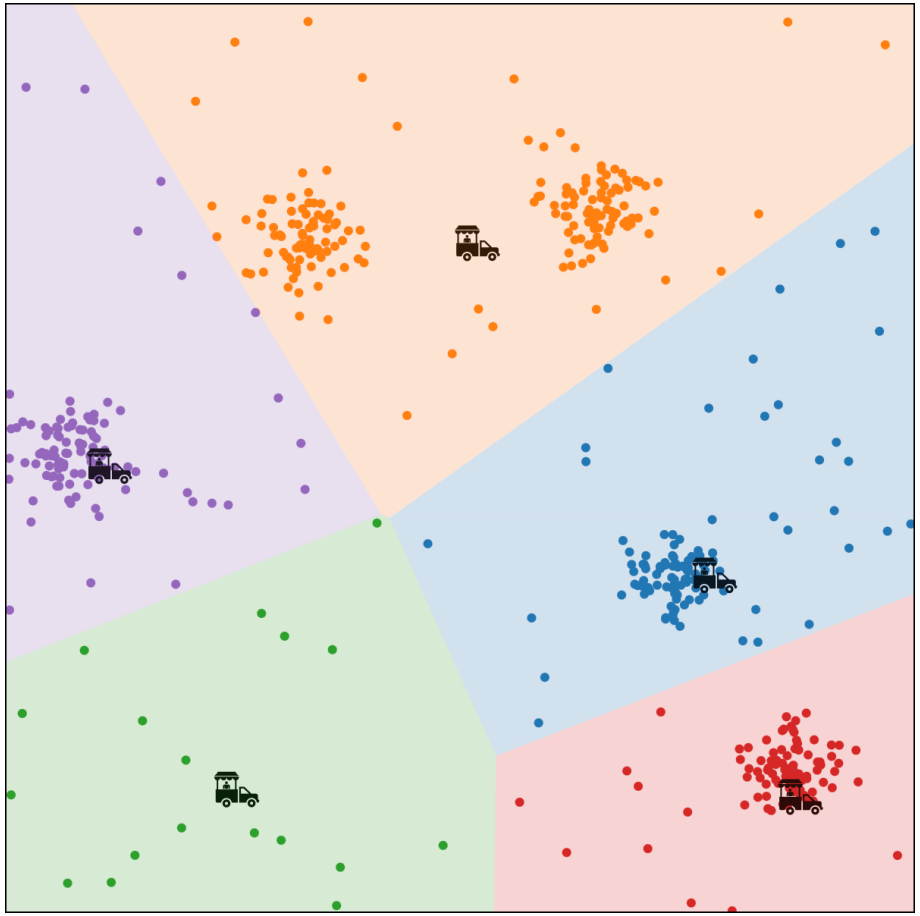


Compare to classification

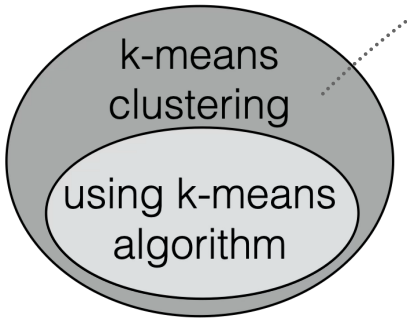


- Did we just do k-class classification?
- Looks like we assigned label $y^{(i)}$, which takes k different values, to each feature vector $x^{(i)}$
- But we didn't use any *labeled* data
- The "labels" here don't have meaning; we could permute them and have the same result.
- Output is really a *partition* of the data/features.

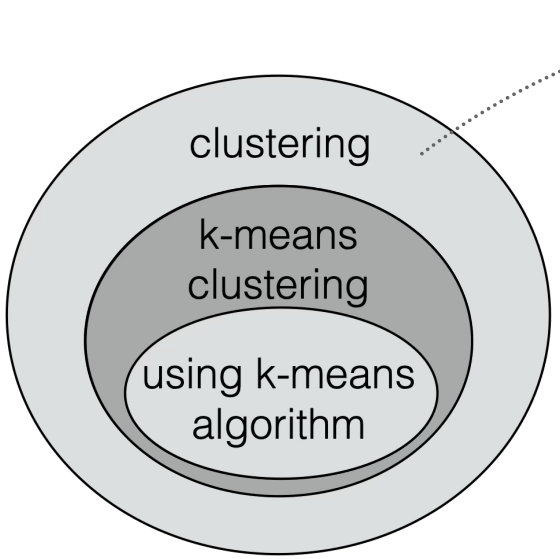
Compare to classification



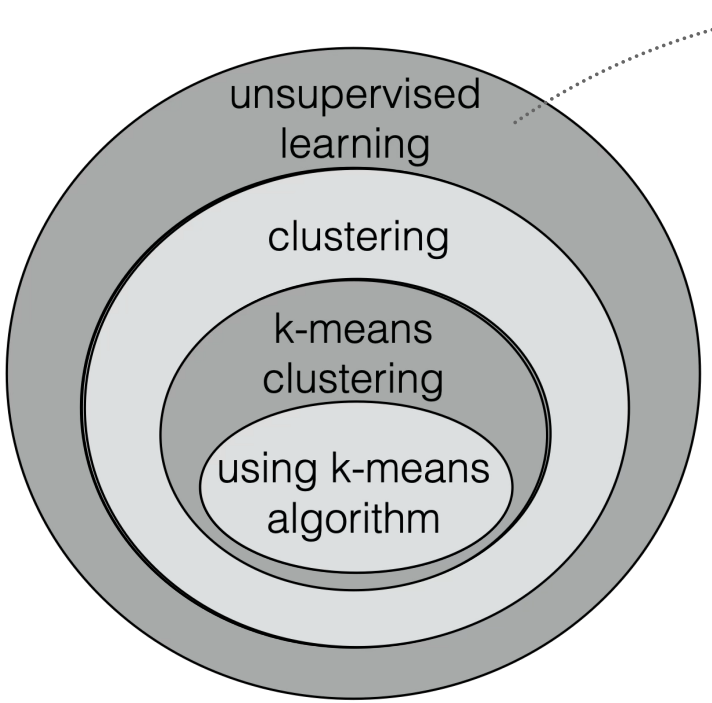
- So what did we do?
- We clustered the data: we grouped the data by similarity
- Why not just plot the data? We should -- whenever we can!
- But also: Precision, big data, high dimensions, high volume.
- An example of unsupervised learning: no labeled data, and we're finding patterns.



- k-means ++
- integre programming
- enumeration
- ...



- Hierarchical Clustering
- Gaussian mixture model (GMMs)
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)



- More broadly, self-supervised learning
- Auto-encoder
- Variational auto-encoder
- Dimensionality reduction (PCA, t-SNE
- Rich world of generative models

...

Y. LeCun

How Much Information is the Machine Given during Learning?

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**

- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**

- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

[Slide Credit: Yann LeCun]

Summary

- Clustering is an important kind of unsupervised learning in which we try to divide the x's into a finite set of groups that are in some sense similar.
- A widely used clustering objective is the k-means. It also requires a distance metric on x's.
- There's a convenient special-purpose method for finding a local optimum: the k-means algorithm.
- The solution obtained by k-means algorithm is sensitive to initialization.
- The solution obtained by k-means algorithm is sensitive to the number of clusters chosen.

https://forms.gle/yMPvCB19C4WkG1s68

We'd love to hear
your thoughts.

Thanks!