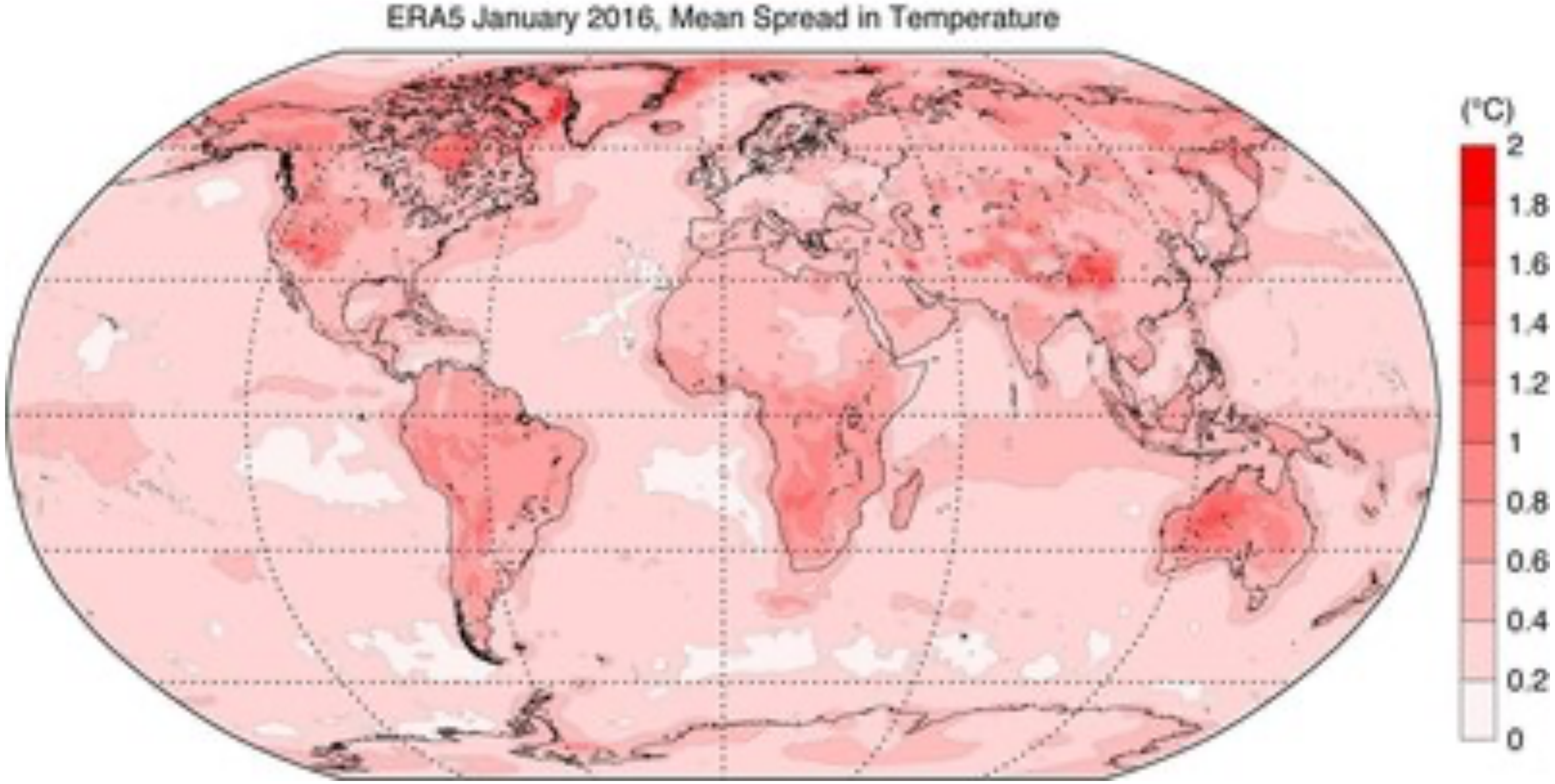
K-Means Clustering

EECE454 Intro. to Machine Learning Systems

Fall 2024

Recap: Supervised Learning

- **Given.** A labeled dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
- **Goal.** Learn $f(\cdot)$ such that $f(\mathbf{x}) \approx y$
    - Example. ERA5 dataset
        - $\mathbf{x}$: time & location
        - y: temperature
        - Goal: Predict temperature at a new time & location



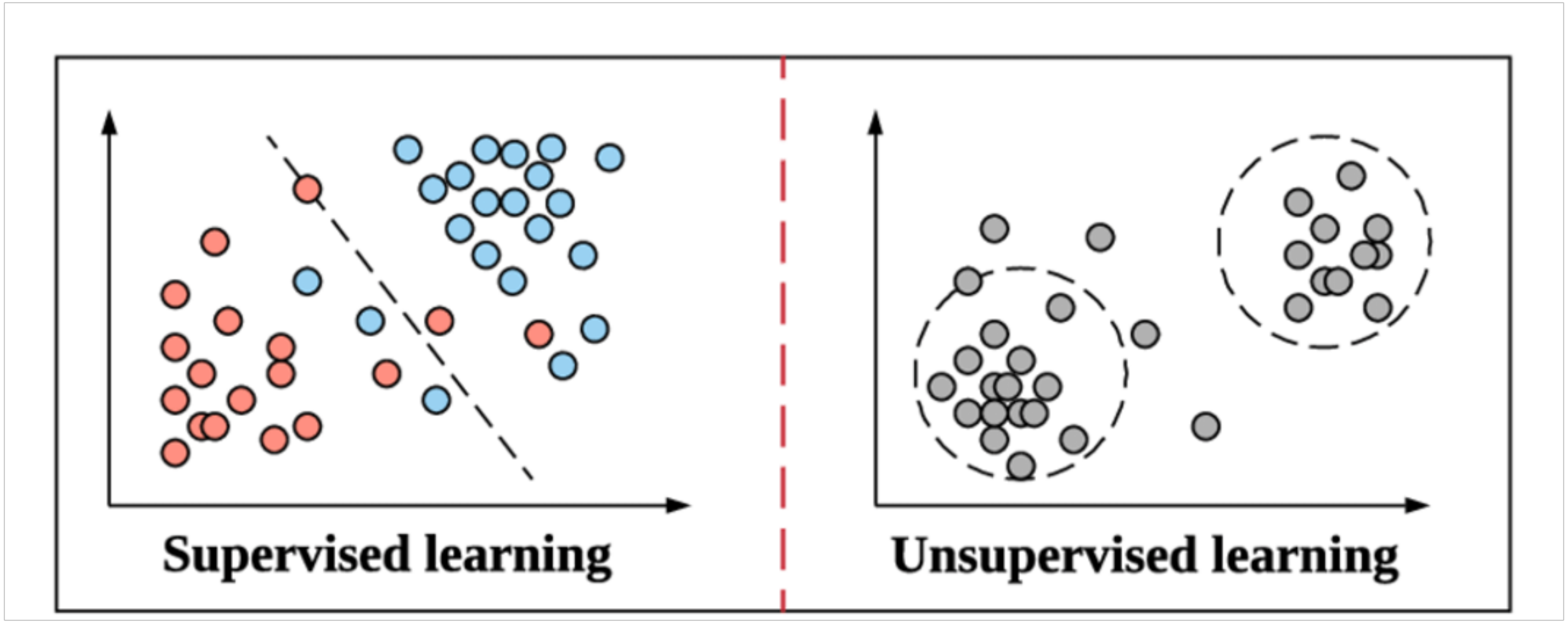
Unsupervised Learning

Unsupervised Learning

- **Given.** An unlabeled dataset $D = \{\mathbf{x}_i\}_{i=1}^{n}$
    - No labeleing cost (typically very large!
    - Example. Common Crawl — petabytes of web-crawled sentences
        - Most language models are trained on these!

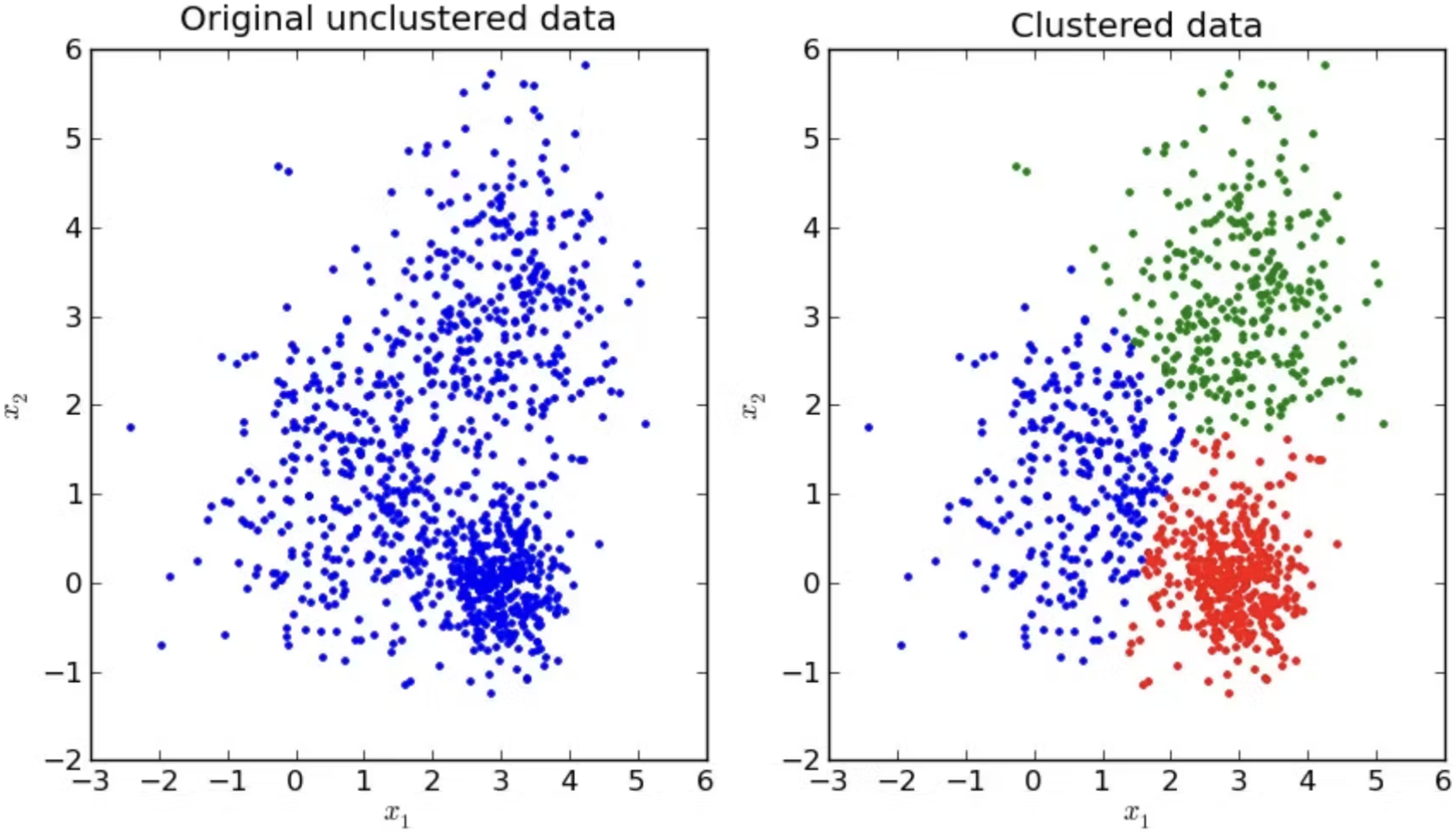
Unsupervised Learning

- **Goal.** Get insights from data, by discovering underlying structure, cause, or statistical relation
    - Learned structure can be used for supervised learning tasks
      (e.g., learning a feature map $\Phi(\cdot)$)



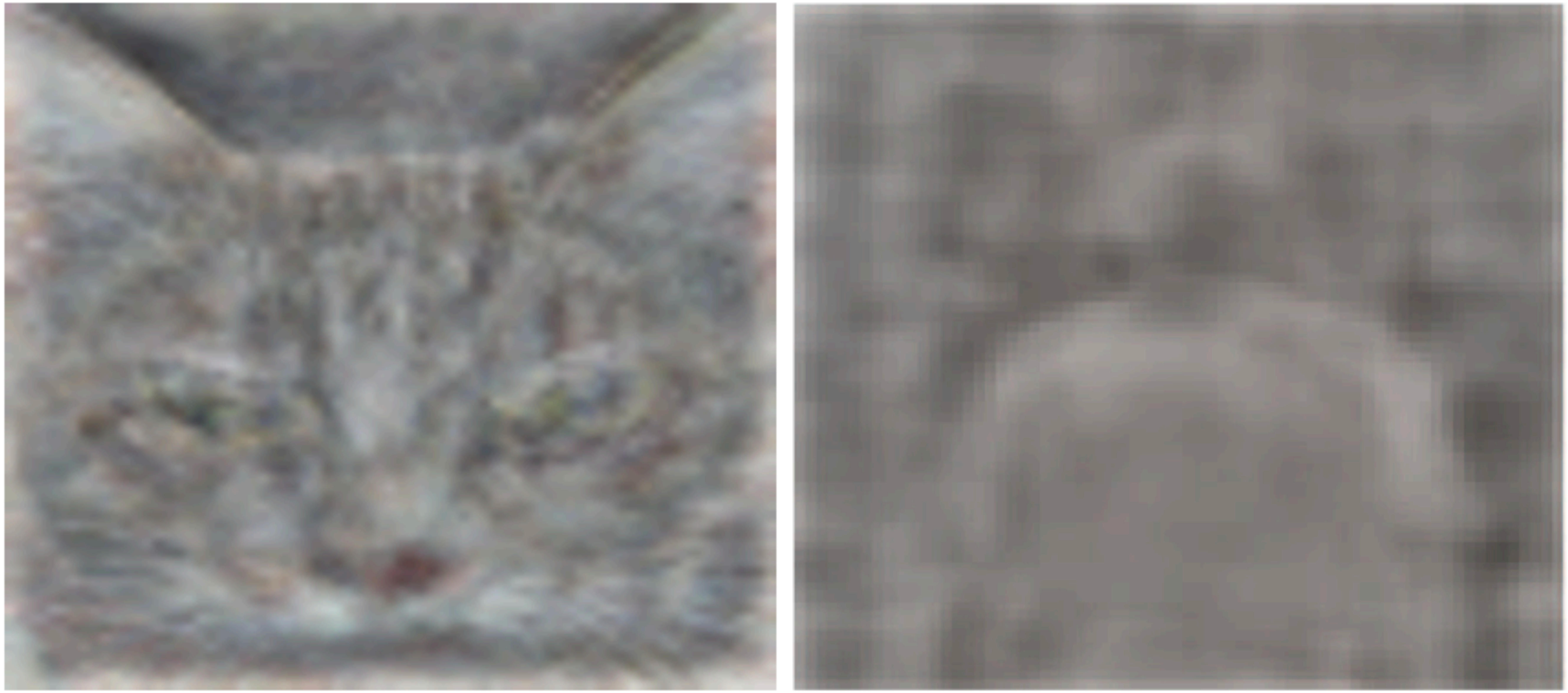
What can unsupervised learning do?

- **1957.** People were clustering many data points



What can unsupervised learning do?

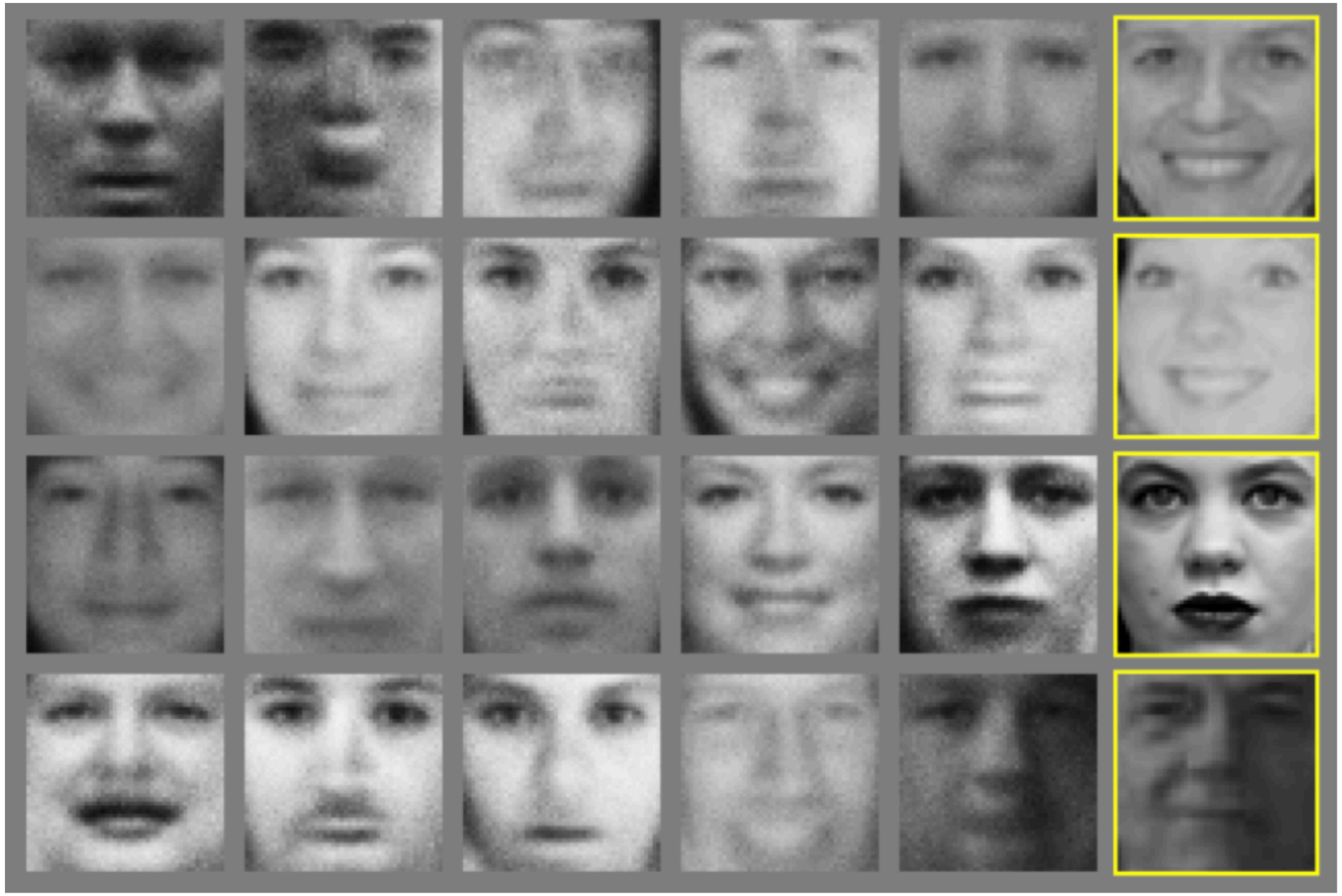
- **2012.** Discovered patterns (useful for classification) from YouTube videos without any supervision



Q. V. Le "Building High-Level Features Using Large Scale Unsupervised Learning," ICASSP 2013

What can unsupervised learning do?

- **2014.** People used face images to generate realistic(?) new faces



Goodfellow et al., "Generative Adversarial Nets" NeurIPS 2014

What can unsupervised learning do?

- **2024.** People are training awesome chatbots

OpenAI o1-preview

User

What is the pH of a 0.10 M solution of $\mathrm{NH_4F}$? The K_a **of** $\mathrm{NH_4^+}$ **is** 5.6×10^{-10} and the K_a of HF is 6.8×10^{-4}.

Hide chain of thought

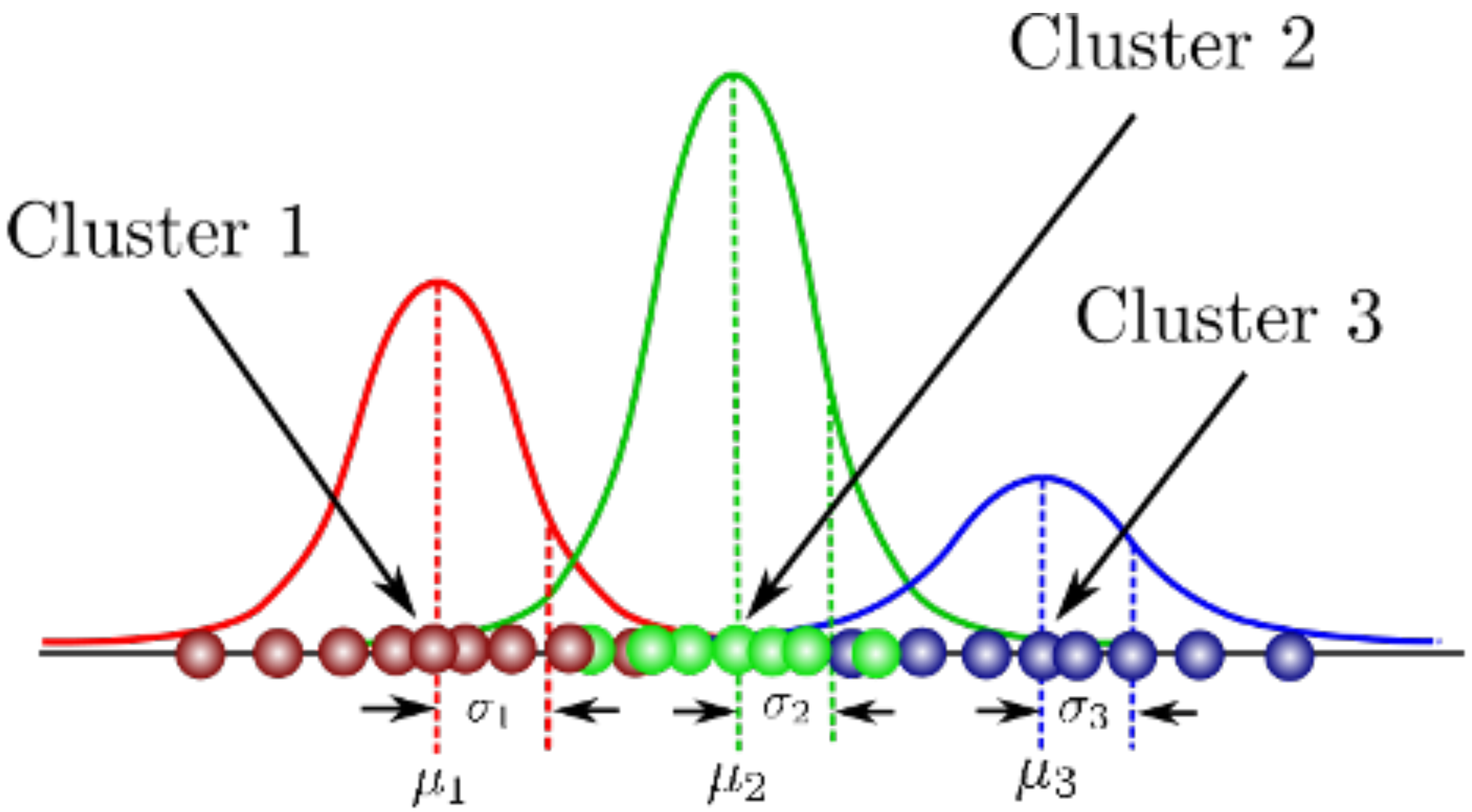
First, let's understand what is being asked.

We need to find the pH of a 0.10 M solution of ammonium fluoride, NH4F.

K-Means Clustering

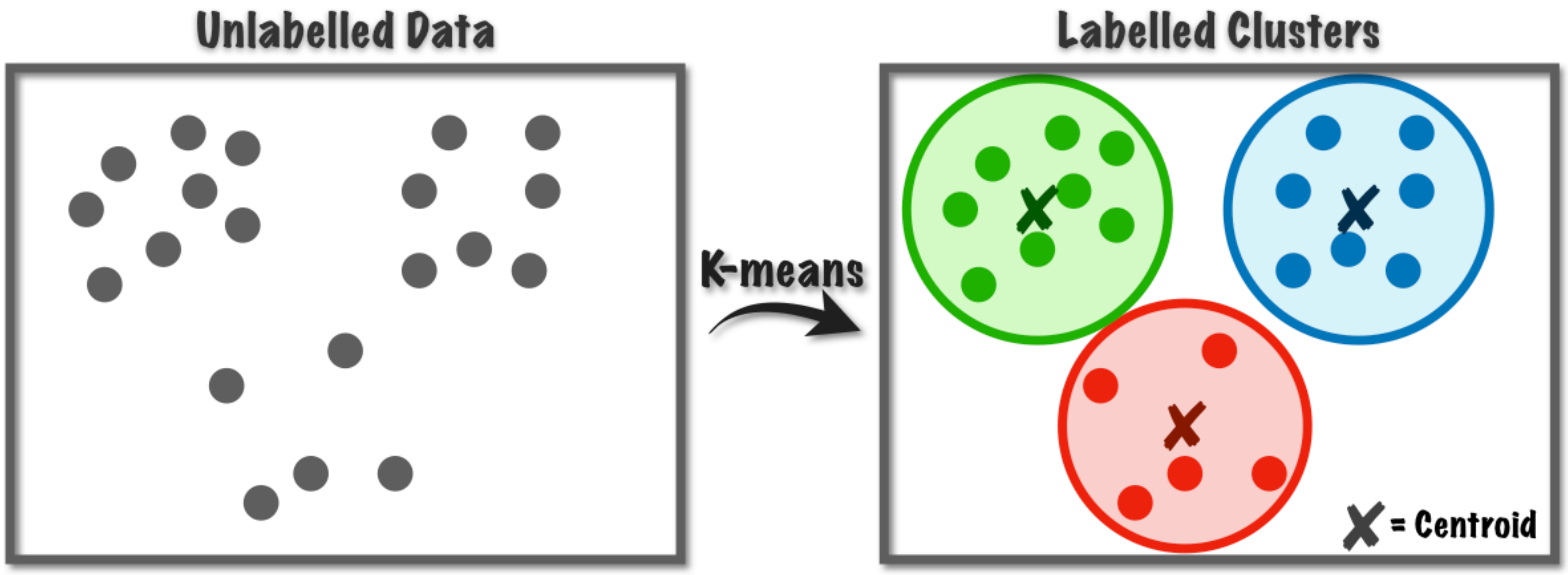
Clustering

- **Assigning** a set of unlabeled data points into pre-specified # of groups
  - K-Means, Gaussian Mixture Models, Hierarchical Clustering, Spectral Clustering, …
  - Implicitly assumes some notion of **similarity**
    - Typically maximizes the similarity of each datum to their assigned clusters



K-Means

- Each cluster is represented by **a single point in space**, called centroid
- The loss is measured by the $\mathrm{dist}(\mathrm{data}, \mathrm{centroid})$
  - i.e., maximize the centroid-data similarity



K-Means

- Suppose that we have a dataset $D = \{\mathbf{x}_i\}_{i=1}^{n}, \quad \mathbf{x}_i \in \mathbb{R}^d$

- We make two decisions:

    - We make K clusters — decide corresponding centroids $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$

K-Means

- Suppose that we have a dataset $D = \{\mathbf{x}_i\}_{i=1}^n, \quad \mathbf{x}_i \in \mathbb{R}^d$
- We make two decisions:
  - We make K clusters — decide corresponding centroids $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$
  - We assign data — decide the assignment $r_{ik} \in \{0,1\}$, $\quad \sum_{k=1}^{K} r_{ik} = 1$
    - $r_{ik} = 1$ means $\mathbf{x}_i$ belongs to k-th cluster (0 otherwise)

K-Means

- Suppose that we have a dataset $D = \{\mathbf{x}_i\}_{i=1}^n, \quad \mathbf{x}_i \in \mathbb{R}^d$
- We make two decisions:
  - We make K clusters — decide corresponding centroids $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$
  - We assign data — decide the assignment $r_{ik} \in \{0,1\}, \quad \sum_{k=1}^{K} r_{ik} = 1$
    - $r_{ik} = 1$ means $\mathbf{x}_i$ belongs to k-th cluster (0 otherwise)
- **Goal.** Choose nice $\{\mu_k\}, \{r_{ik}\}$ which minimize the mean-squared distance (or any distance), i.e.,

$$\min_{\{\mu_k\}} \min_{\{r_{ik}\}} \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$$

Algorithm

$$\min_{\{\mu_k\}} \min_{\{r_{ik}\}} \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$$

- This is a mixed optimization of **discrete & continuous** variables
    - Tricky to solve in general.

Algorithm

$$\min_{\{\mu_k\}} \min_{\{r_{ik}\}} \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$$

- This is a mixed optimization of **discrete & continuous** variables
  - Tricky to solve in general.
- **Strategy.** Look at the optimality conditions of each subproblem
  - Principle 1. Centroid —> assignment: Assign to the closest centroid
    - Given the centroid, optimal assignment is obvious:

$$r_{ik} = \begin{cases} 1 & \cdots \quad k = \mathrm{argmin}_k \|\mathbf{x}_i - \mu_k\|_2^2 \\ 0 & \cdots \quad \text{otherwise} \end{cases}$$

Algorithm

$$\min_{\{\mu_k\}} \min_{\{r_{ik}\}} \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$$

- This is a mixed optimization of **discrete & continuous** variables
  - Tricky to solve in general.
- **Strategy.** Look at the optimality conditions of each subproblem
  - Principle 1. Centroid —> assignment: Assign to the closest centroid
  - Principle 2. Assignment —> centroid: Take an average
    - Given the assignments, optimal centroid is obvious:
      If $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(n_k)}$ are assigned to the kth cluster, let

$$\mu_k = \mathrm{argmin}_{\mu \in \mathbb{R}^d} \sum_{i=1}^{n_k} \|\mu - \mathbf{x}_{(i)}\|_2^2 = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{x}_{(i)}$$

Algorithm

$$\min_{\{\mu_k\}} \min_{\{r_{ik}\}} \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$$

- This is a mixed optimization of **discrete & continuous** variables
  - Tricky to solve in general.
- **Strategy.** Look at the optimality conditions of each subproblem
  - Principle 1. Centroid —> assignment: Assign to the closest centroid
  - Principle 2. Assignment —> centroid: Take an average
- In other words, the optimal solution should satisfy both:
  - Data are assigned to the nearest centroid
  - Centroids are average of assigned data
- **Question.** How do we find a solution that satisfies these?

Lloyd's algorithm

- **Algorithm.** Apply P1 —> Apply P2 —> Apply P1 —> ... —> Until convergence
  - Assignment step. Given $\{\mu_k\}$, find $\{r_{ik}\}$
  - Update step. Given $\{r_{ik}\}$, find $\{\mu_k\}$

Algorithm 1 k-means algorithm

```
Specify the number k of clusters to assign.
Randomly initialize k centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.
```

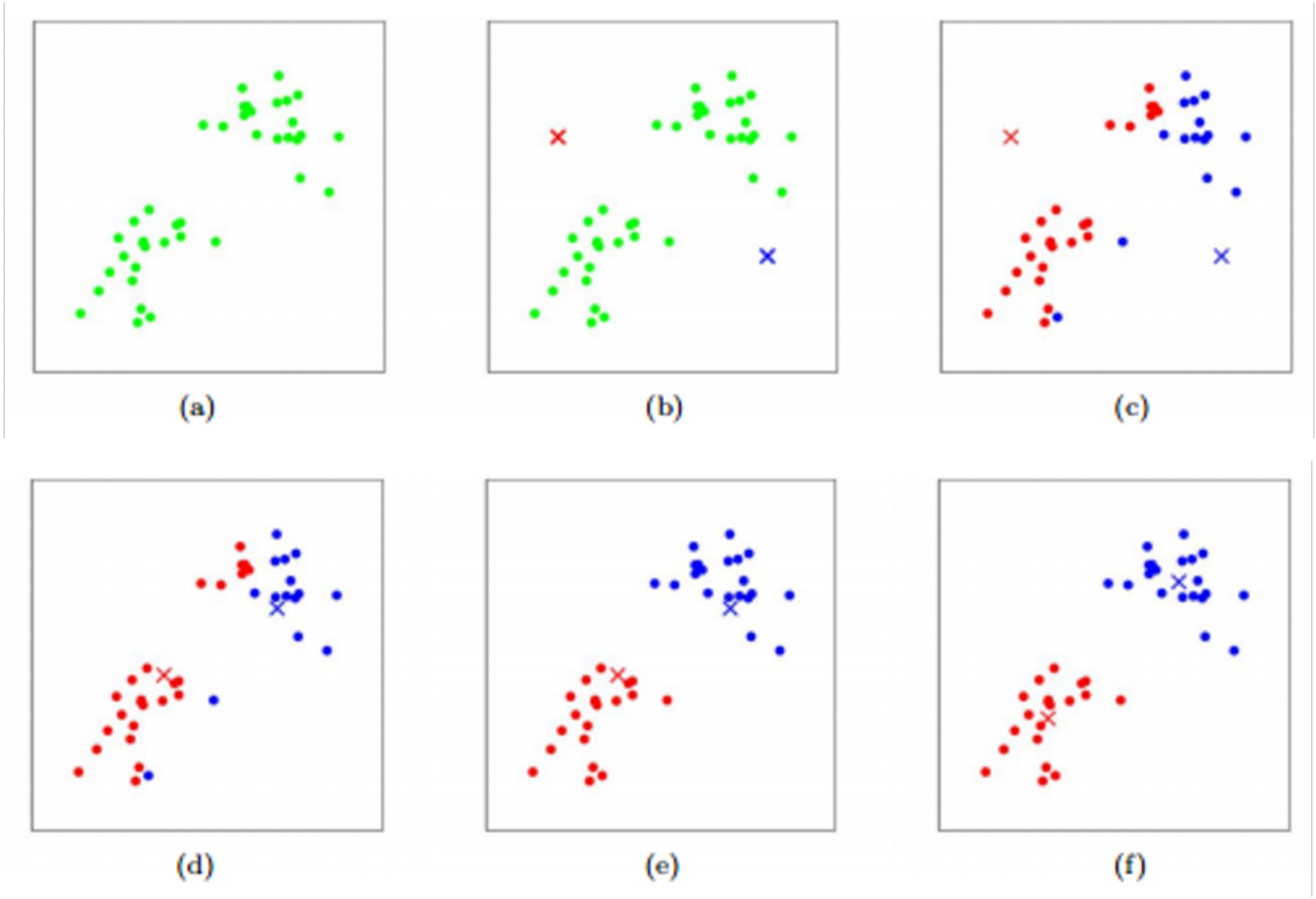
Lloyd's algorithm

- **Algorithm.** Apply P1 —> Apply P2 —> Apply P1 —> … —> Until convergence
  - Assignment step. Given $\{\mu_k\}$, find $\{r_{ik}\}$
  - Update step. Given $\{r_{ik}\}$, find $\{\mu_k\}$
- This is called the Lloyd's algorithm (originally proposed for pulse-code modulation)
  - which is a special case of the **expectation-maximization** (EM) algorithm

Algorithm 1 k-means algorithm

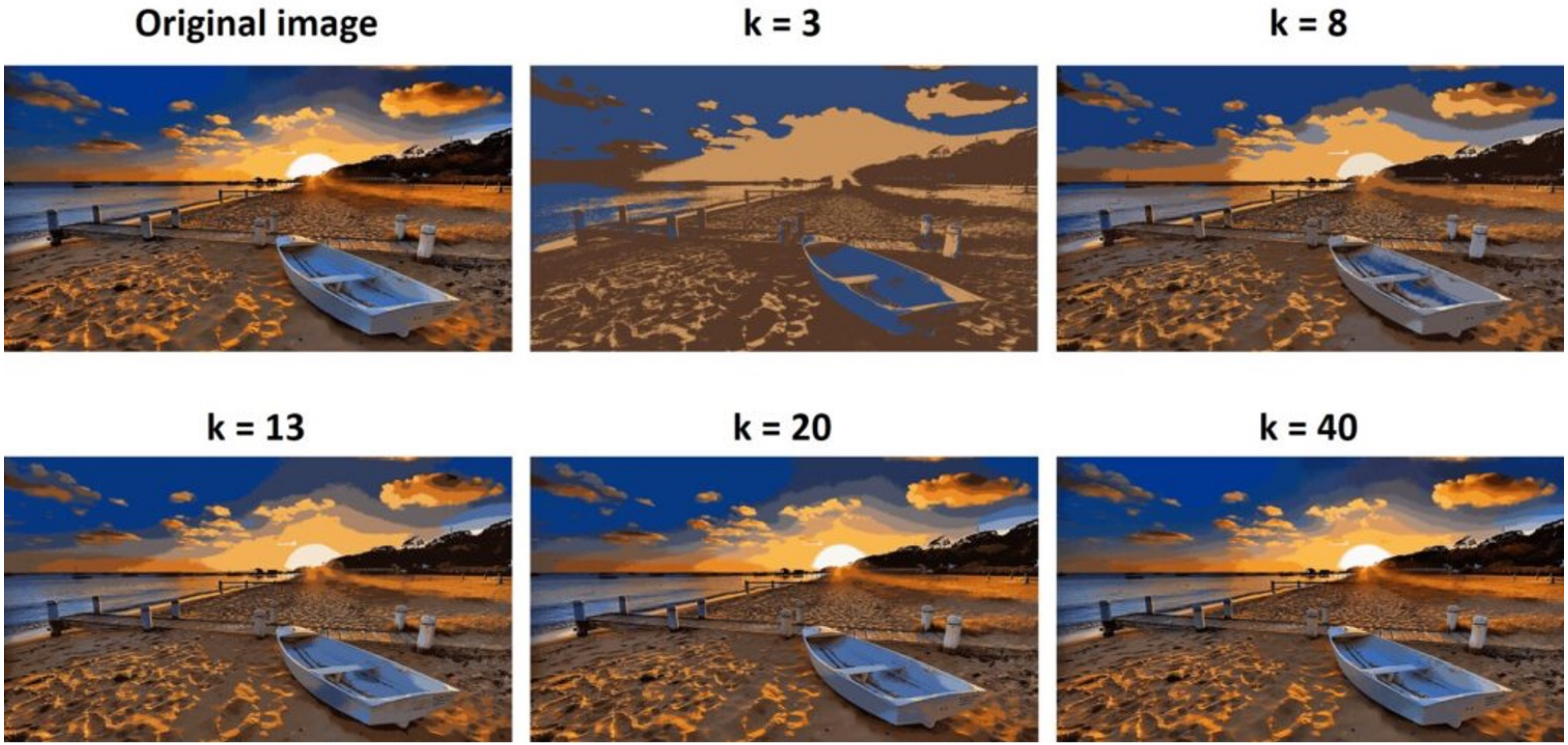
```
Specify the number k of clusters to assign.
Randomly initialize k centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.
```

Lloyd's algorithm



A simple application

- An easy application is to compress an image.
  - Reduce the number of colors —> representable with low bit



Properties

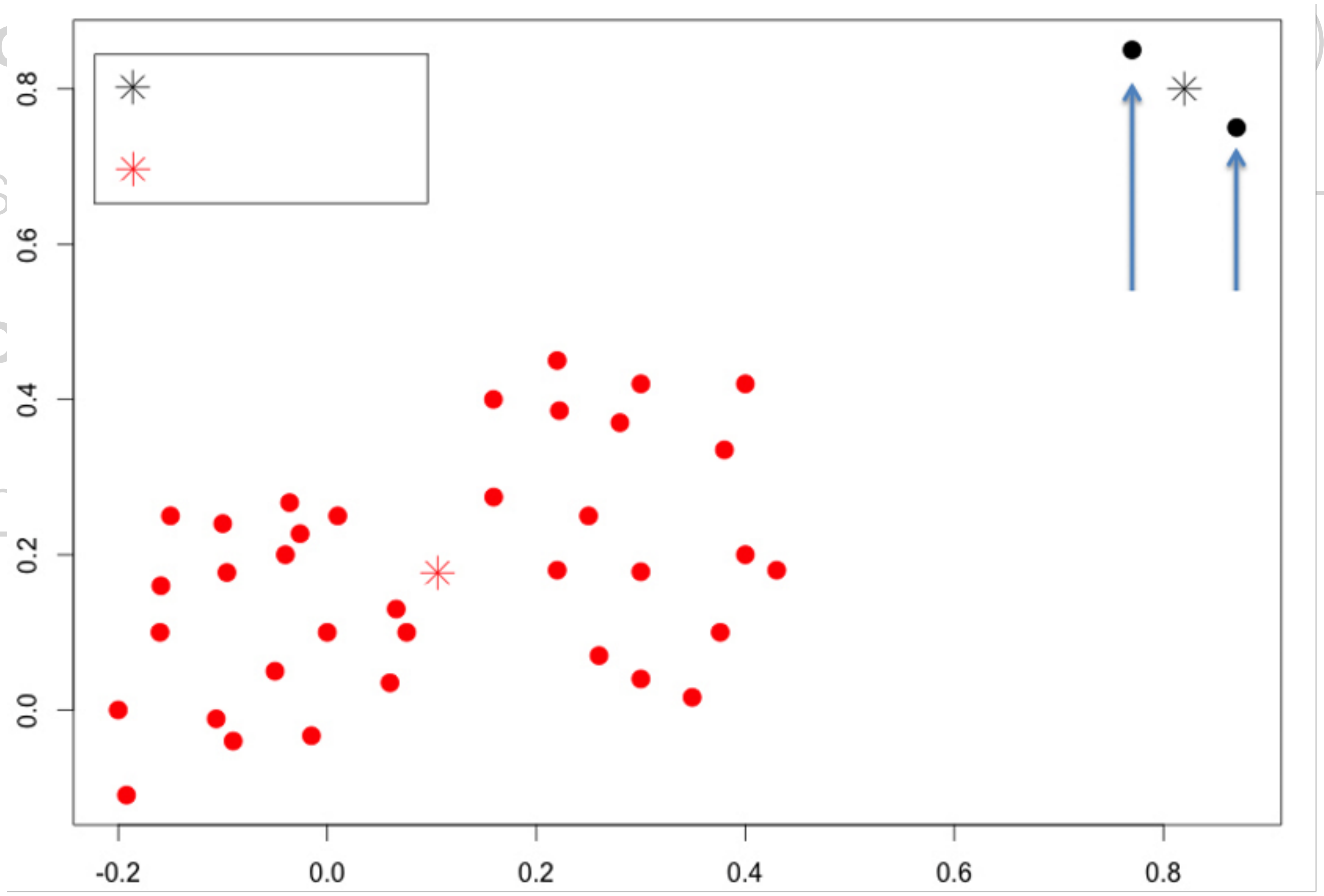
- **Convergence.** Provably converges to some (local) optimum.
  - Success largely depends on the initialization — use K-means++ for better results

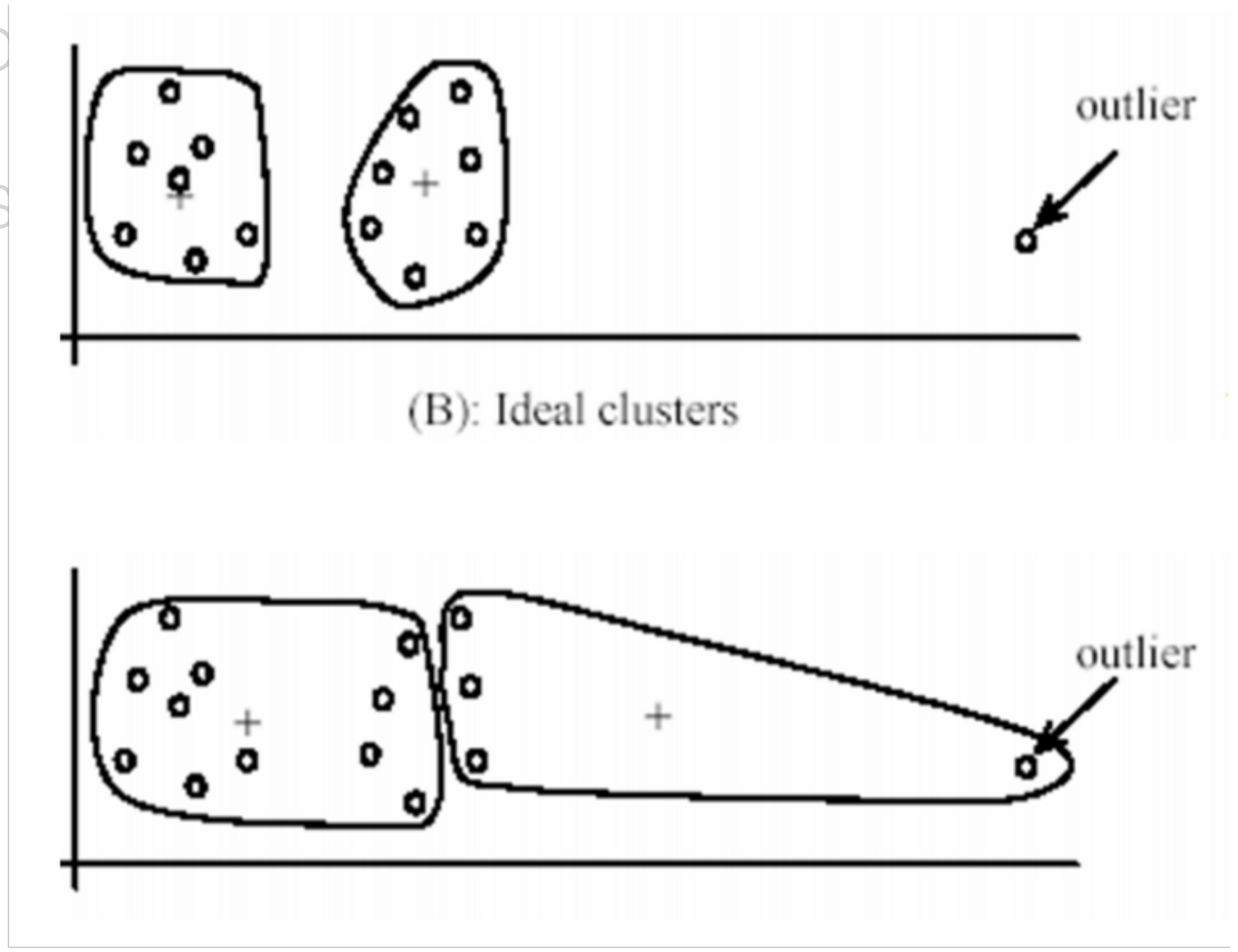
Properties

- **Convergence.** Provably converges to some (local) optimum.
  - Success largely depends on the initialization — use K-means++ for better results
- **Computation.** The training requires...
  - <u>Assignment</u>. $\mathcal{O}(d \cdot k \cdot n)$
  - <u>Update</u>. $\mathcal{O}(n)$
  - The testing requires $\mathcal{O}(d \cdot k)$ per sample

Properties

- Convergenc ... op
  - Success ... us
- Computatic
  - Assignm
  - Update.
  - The test





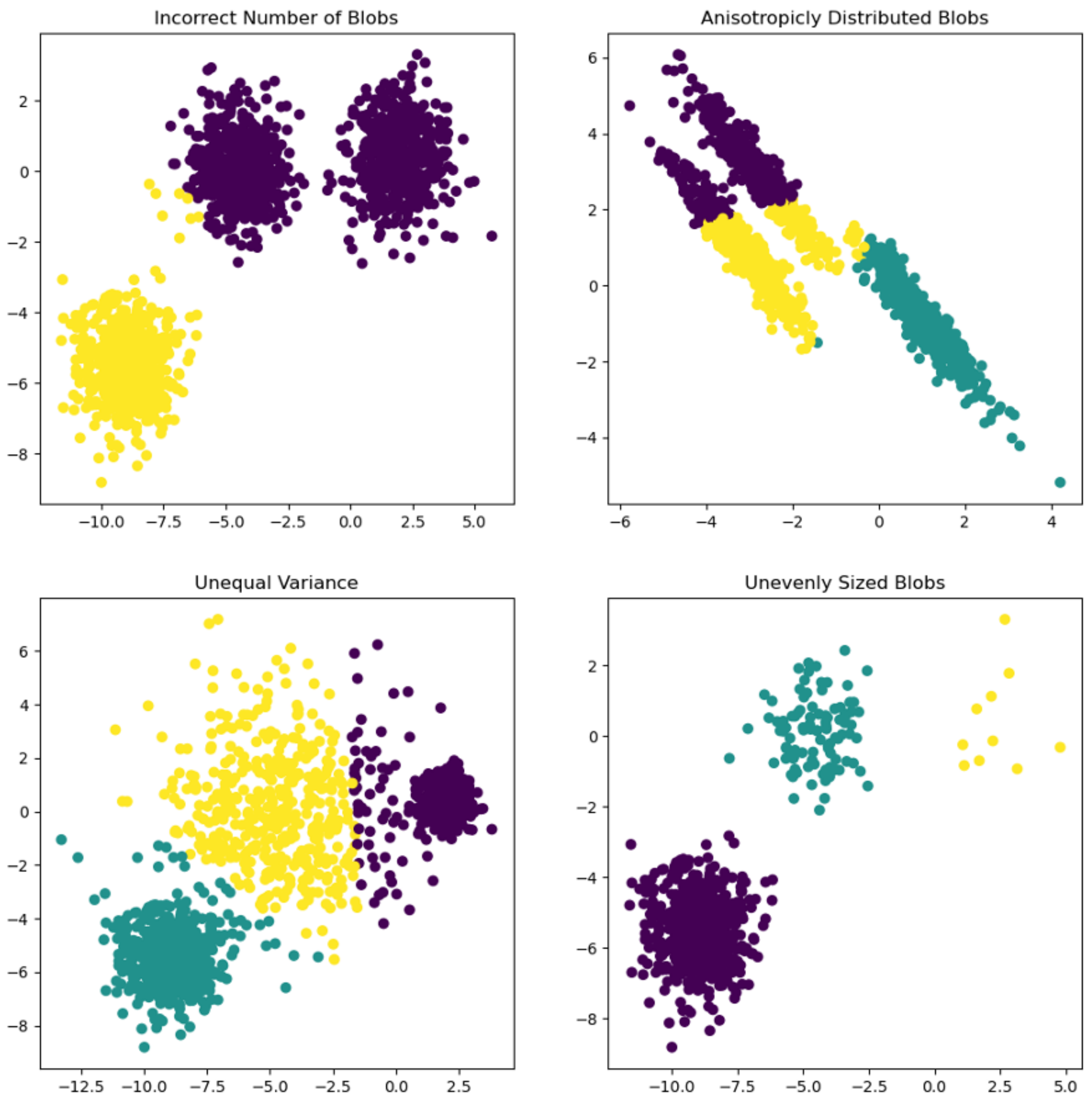
- **Limitation#1.** Quite sensitive to outliers
  - Leads to suboptimal cluster assignments

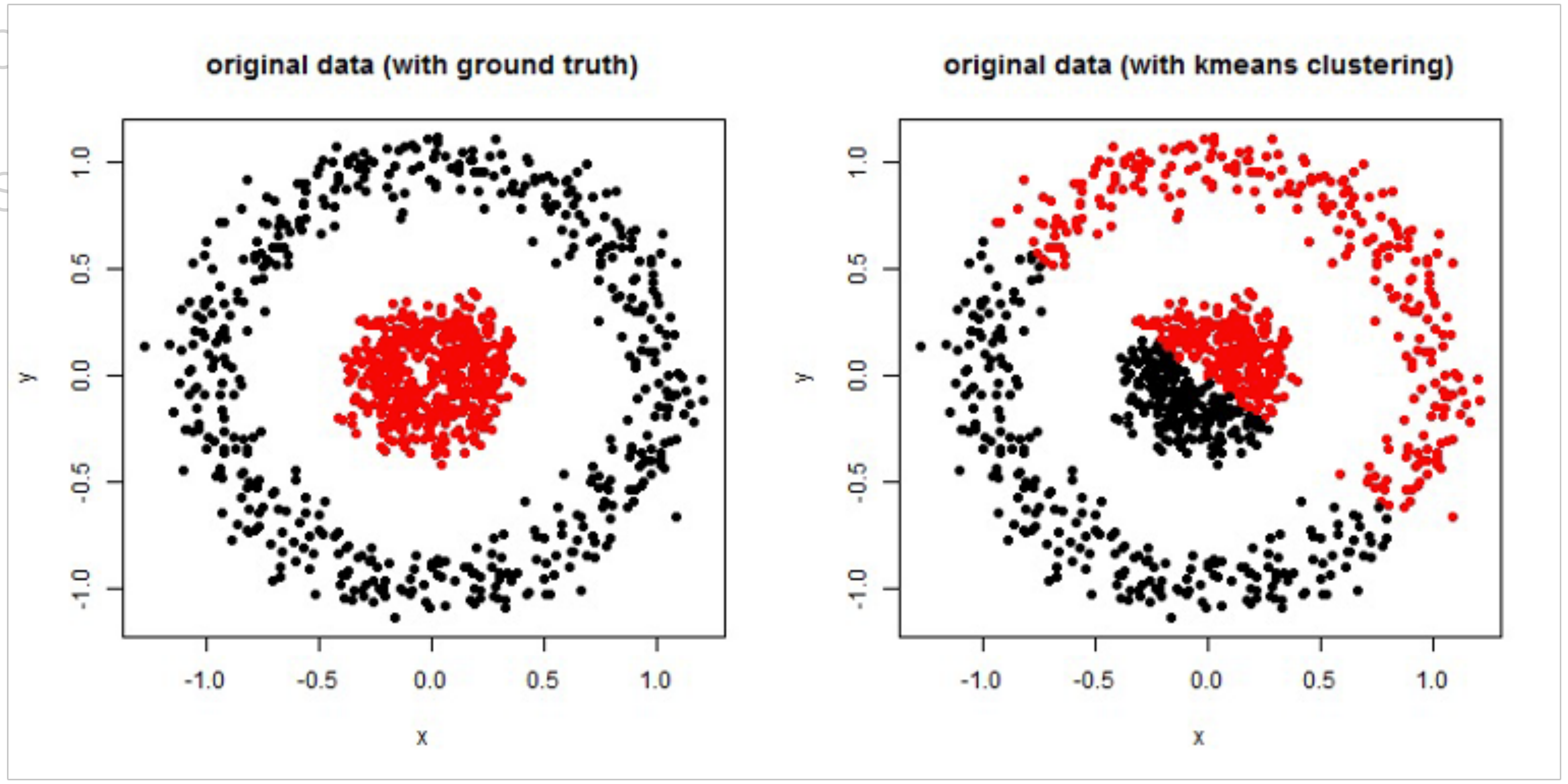
Properties

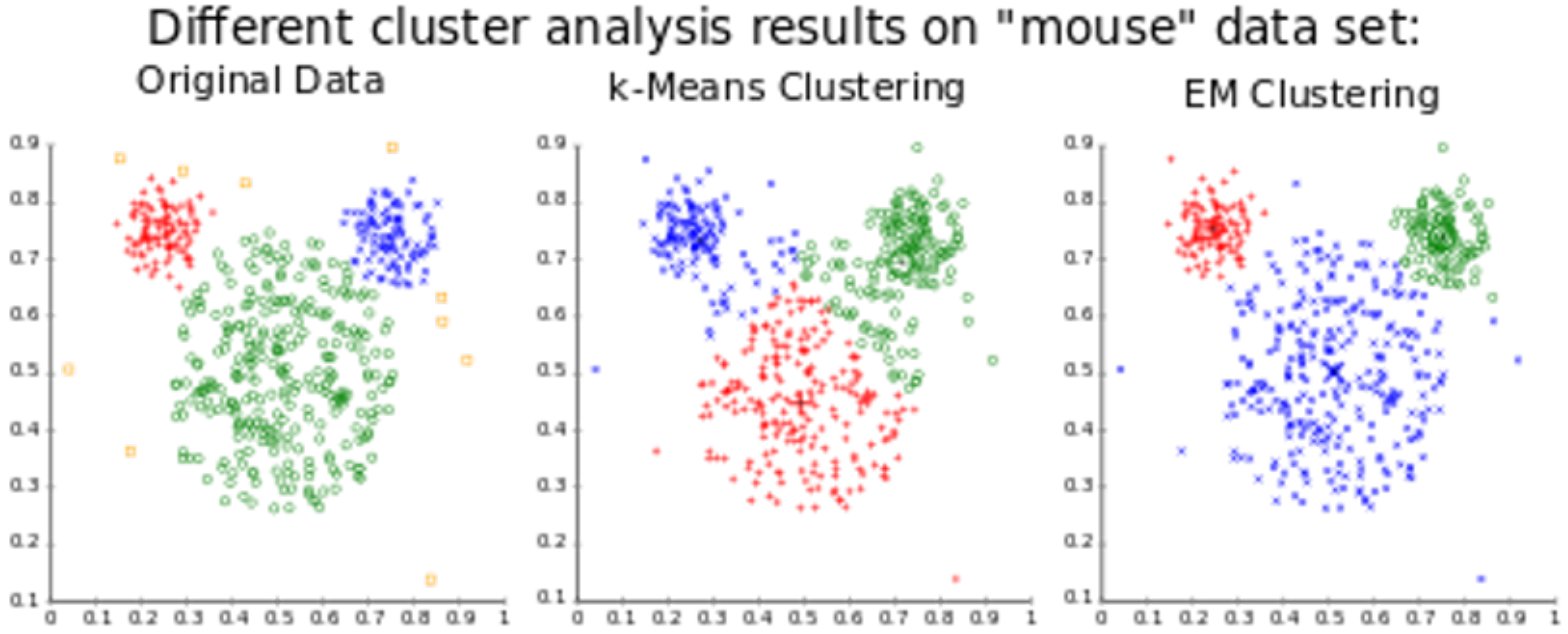
- Converger
  - Succes
- Computati
  - Assigni
  - Update
  - The tes
- Limitation
  - Leads t

l) op

— us







- **Limitation#2.** May not work for certain datasets
  - e.g., overlapping clusters

Soft K-Means

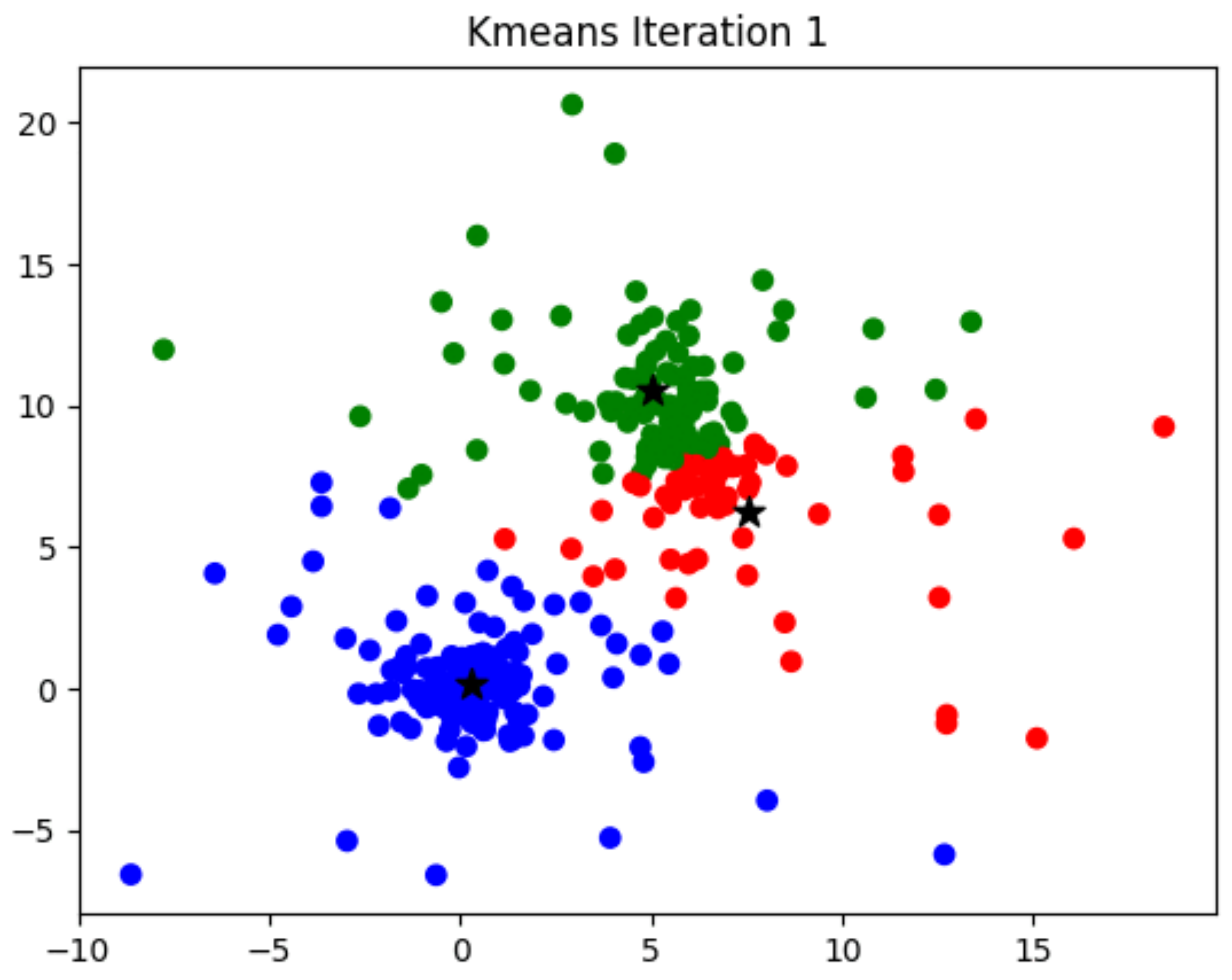
Soft K-Means

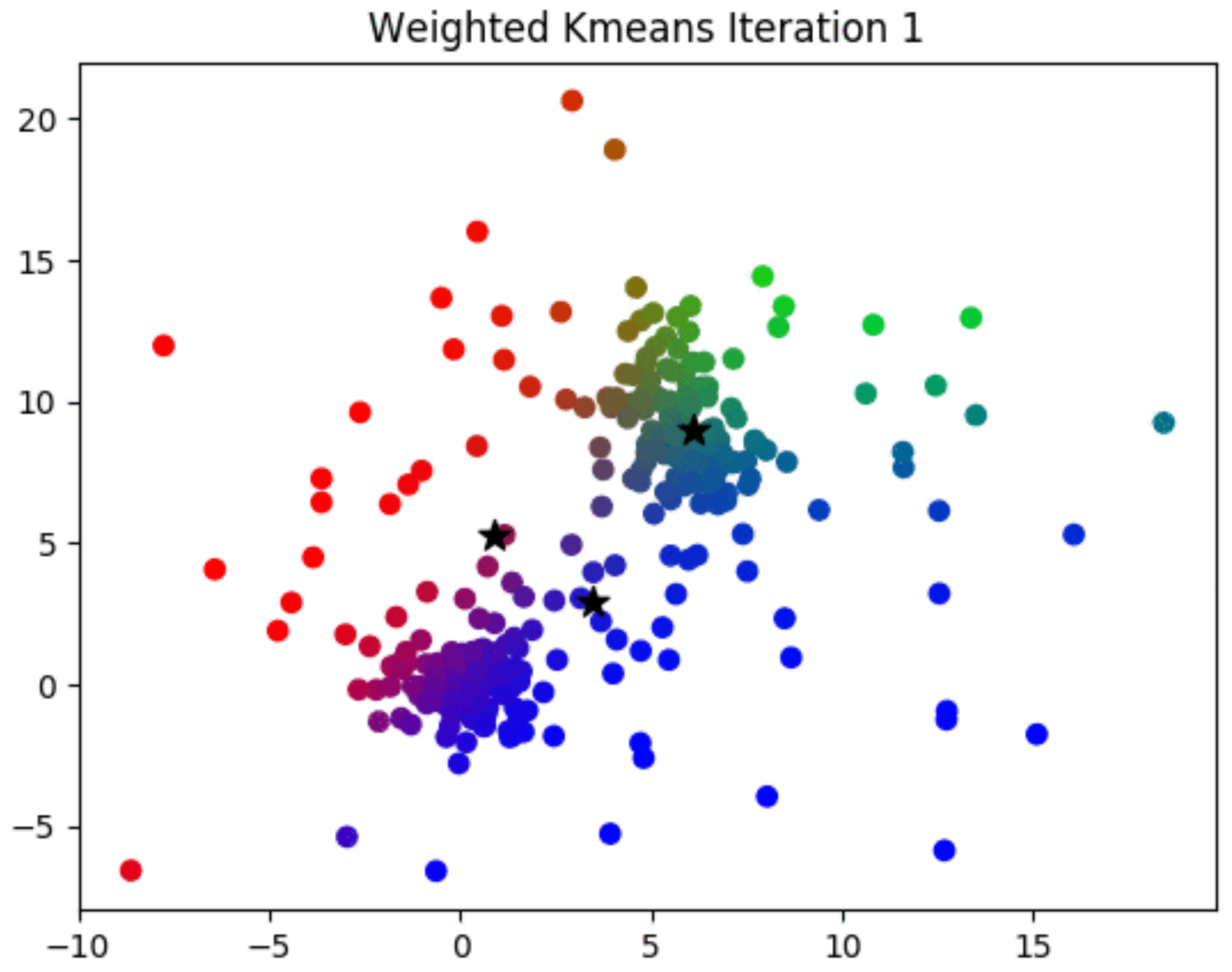
- One version of K-means that can handle overlapping clusters
- **Idea.** Make the assignment **soft**
  - Hard. A point belongs to a specific cluster

$$r_{ik} \in \{0,1\}, \quad \sum_{k=1}^{K} r_{ik} = 1$$

  - Soft. A point may belong 90% to one, and 10% to another

$$r_{ik} \in [0,1], \quad \sum_{k=1}^{K} r_{ik} = 1$$





Algorithm

- **Assignment.** The larger **responsibility** for closer centroid
  - with some hardness hyperparameter β

$$r_{ik} = \frac{\exp(-\beta \|\mathbf{x}_i - \mu_k\|_2^2)}{\sum_j \exp(-\beta \|\mathbf{x}_i - \mu_j\|_2^2)}$$

  - will discuss why such form, in GMM

Algorithm

- **Assignment.** The larger **responsibility** for closer centroid
  - with some hardness hyperparameter β

$$r_{ik} = \frac{\exp(-\beta\|\mathbf{x}_i - \mu_k\|_2^2)}{\sum_j \exp(-\beta\|\mathbf{x}_i - \mu_j\|_2^2)}$$

  - will discuss why such form, in GMM

- Note. If we let $\beta \to \infty$, this becomes the hard assignment

$$r_{ik} = \begin{cases} 1 & \cdots \quad k = \mathrm{argmin}_k \|\mathbf{x}_i - \mu_k\|_2^2 \\ 0 & \cdots \quad \text{otherwise} \end{cases}$$

  - Thus we call such r_{ik} a softmax

Algorithm

- **Update.** Take a weighted average of the data
  - the weight comes from the responsibility

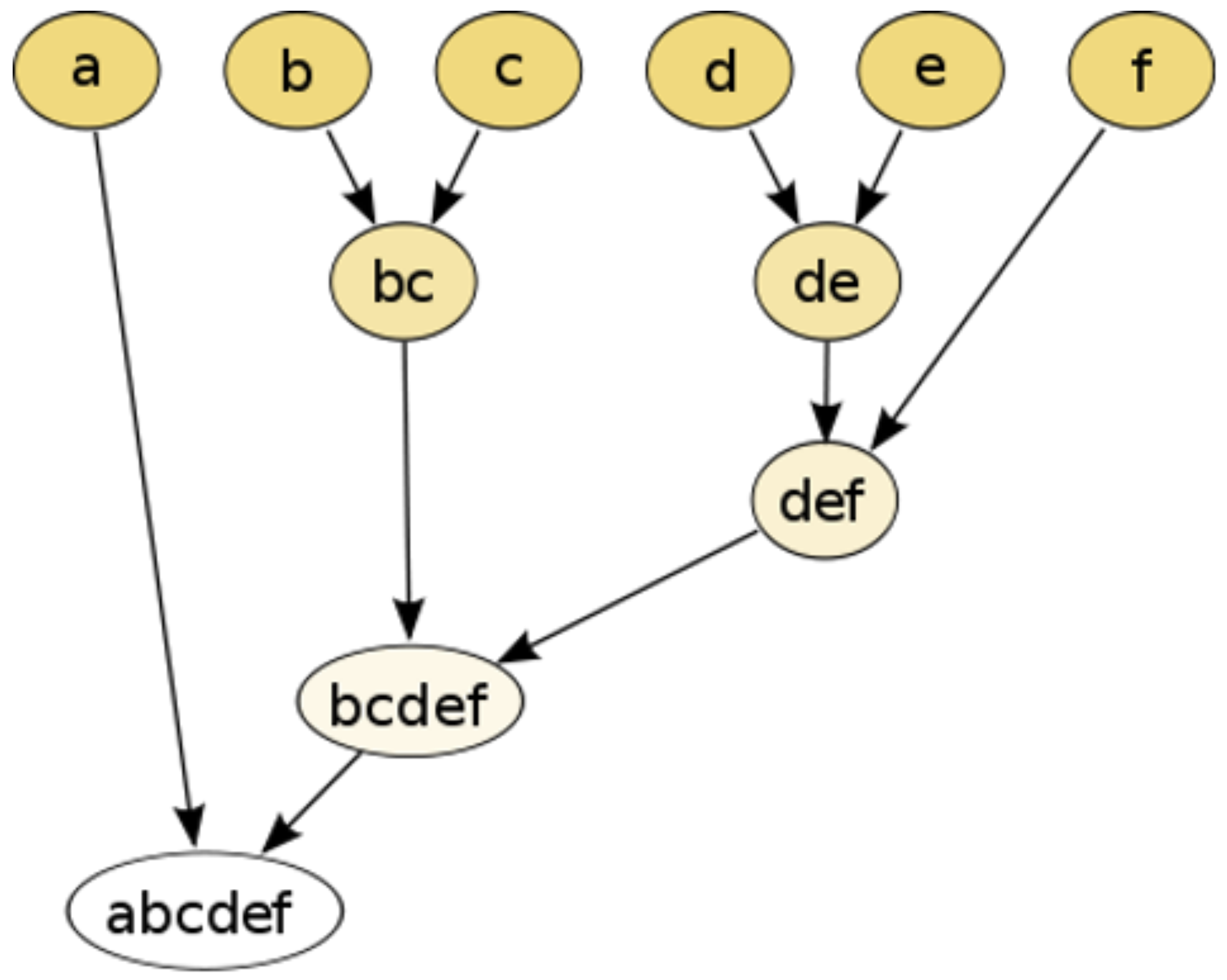
$$\mu_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_j r_{jk}}$$

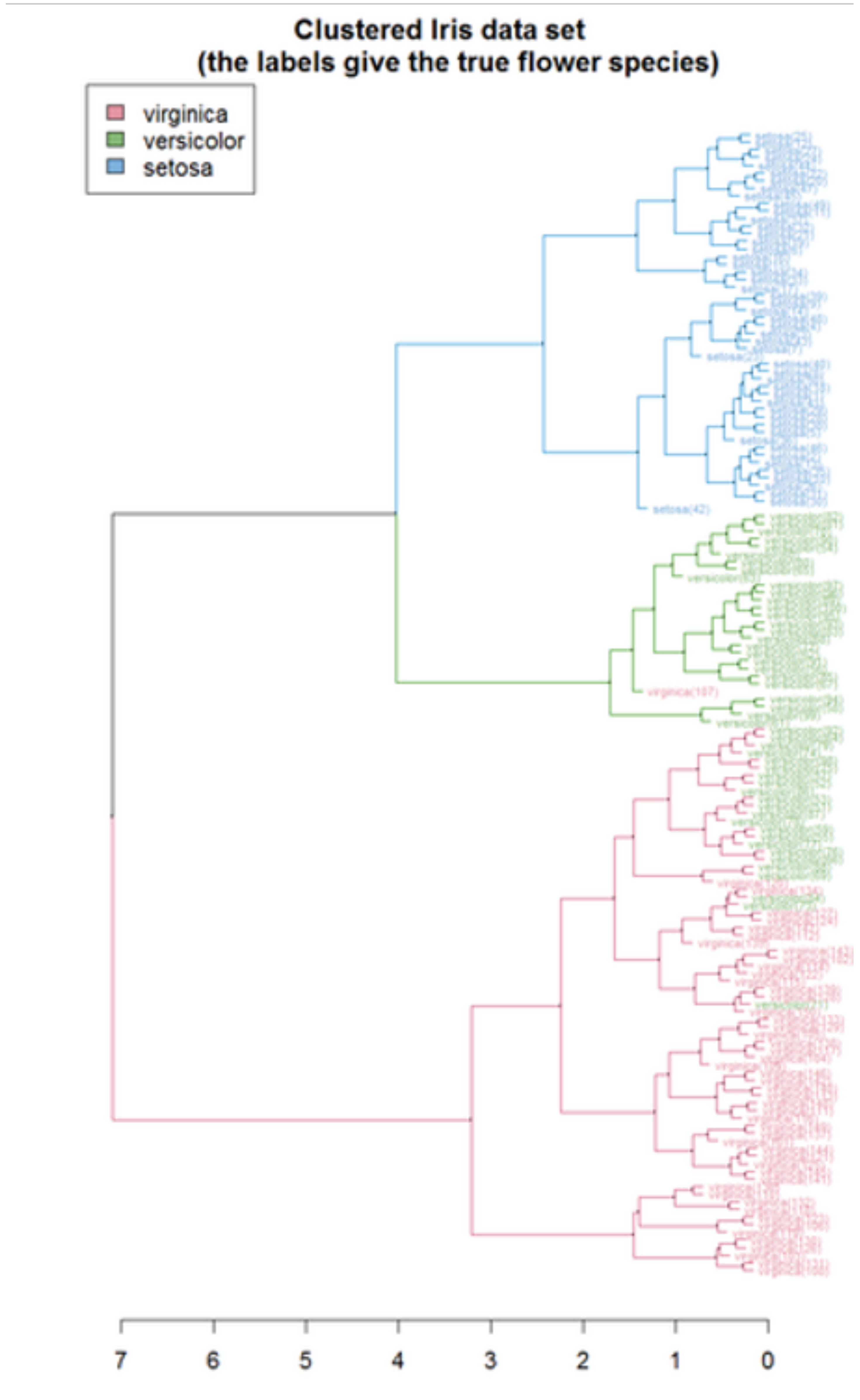
  - can be derived similarly as in hard K-means

Others (informally)

Hierarchical Clustering

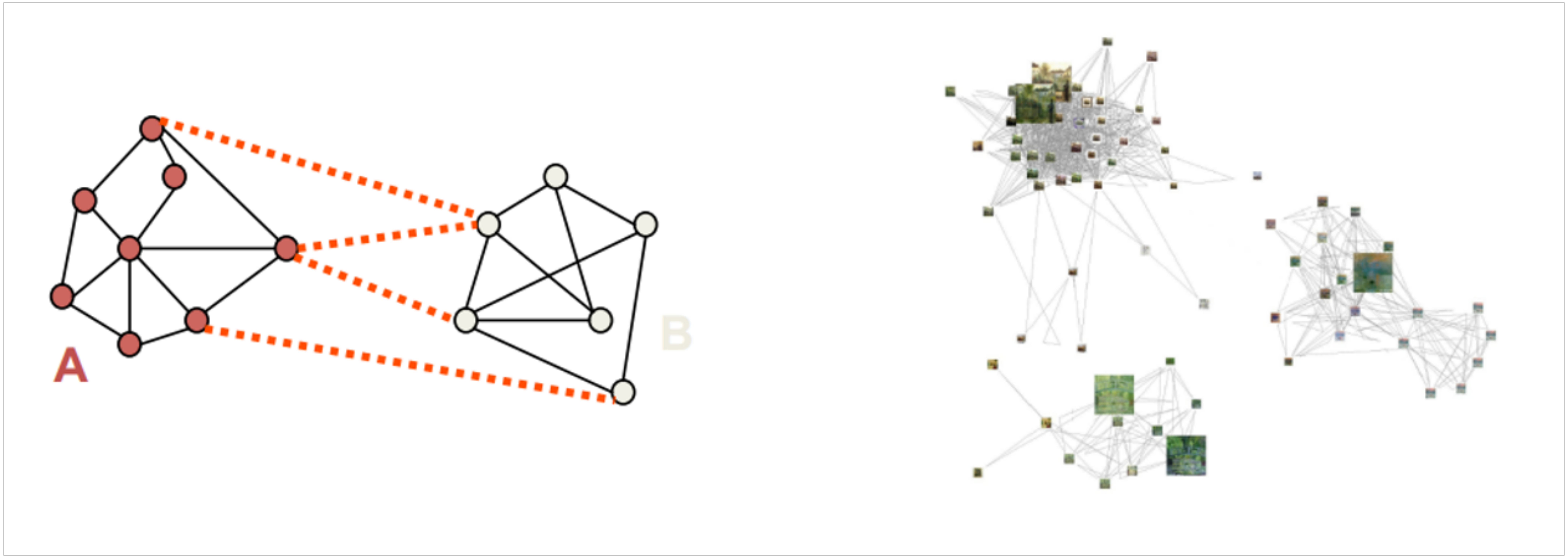
- **Idea.** Clusters inside clusters
  - Discovers hierarchical structures
  - Relax strict assumptions (e.g., distributions)
  - leverage faster heuristic algorithms
  - Waive strict decision of K





Spectral Clustering

- **Idea.** Data lies on a graph.
  - Similarity is measured by the distance on graph
  - Solve via graph algorithms, e.g., min-cut



Next up

- Mixture models

Cheers