

机器学习

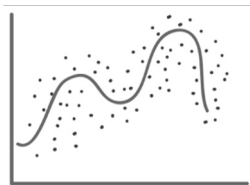
第十四讲: 聚类

顾小东

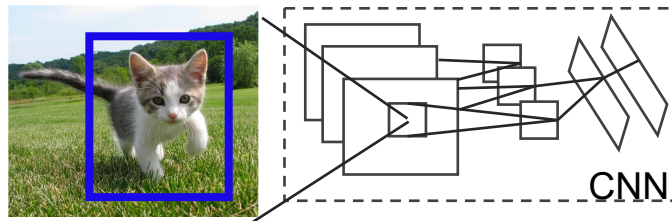
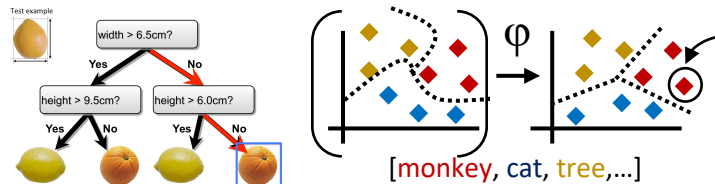
上海交通大学计算机学院



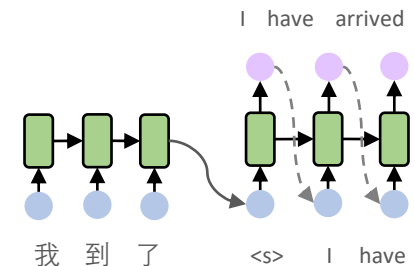
What we have learned so far?



Regression



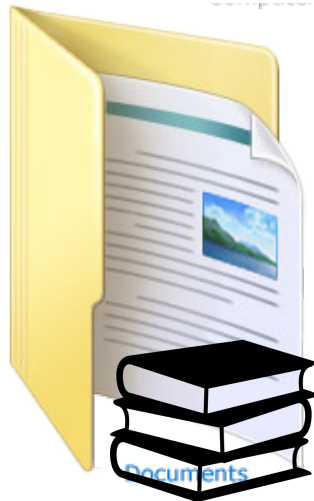
Classification



Generation

All previous lectures involve data labels (supervised learning)

What if we have no labels?



Can we still learn something from the data?

Supervised vs Unsupervised Learning

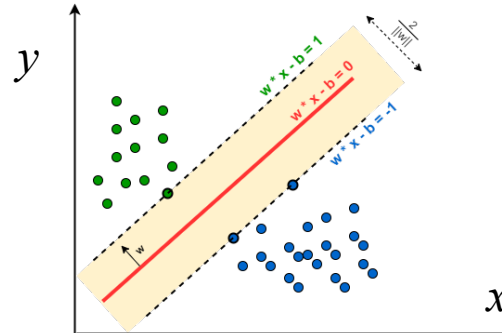
Machine Learning

Supervised Learning

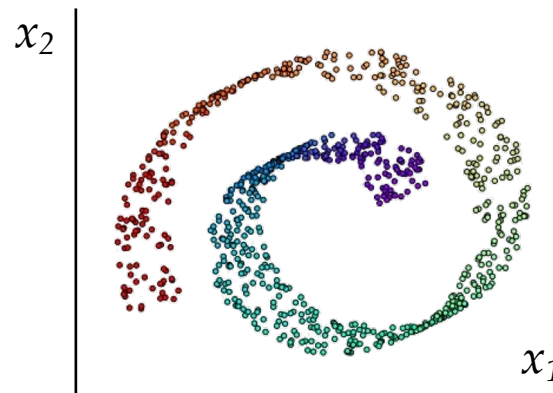
Unsupervised Learning

- Clustering
- Dimensional Reduction
- Density Estimation
- ...

Reinforcement Learning



Supervised: (have labels)



Unsupervised: (No labels)

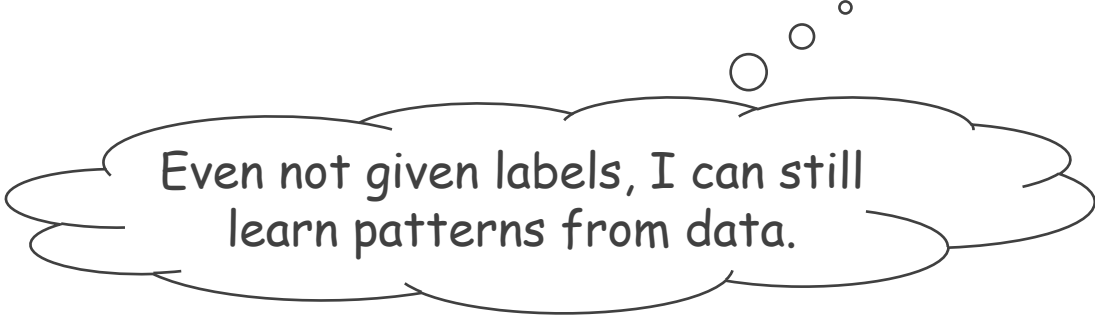
Supervised vs Unsupervised Learning

Supervised Learning

discover patterns in the data that relate data attributes **with a target (class) attribute**.

Unsupervised Learning

The data have **no target attribute**. We want to explore the data to find some **intrinsic structures**.



Even not given labels, I can still learn patterns from data.

Unsupervised Learning

物以類聚

Clustering
Learning similarity

- K-means (✓)
- Gaussian Mixture

無中生有

Generative
Density estimation

- GAN
- VAE
- Diffusion

化繁為簡

Dimensional Reduction
Data compression

- Principal component analysis
- Auto-encoder

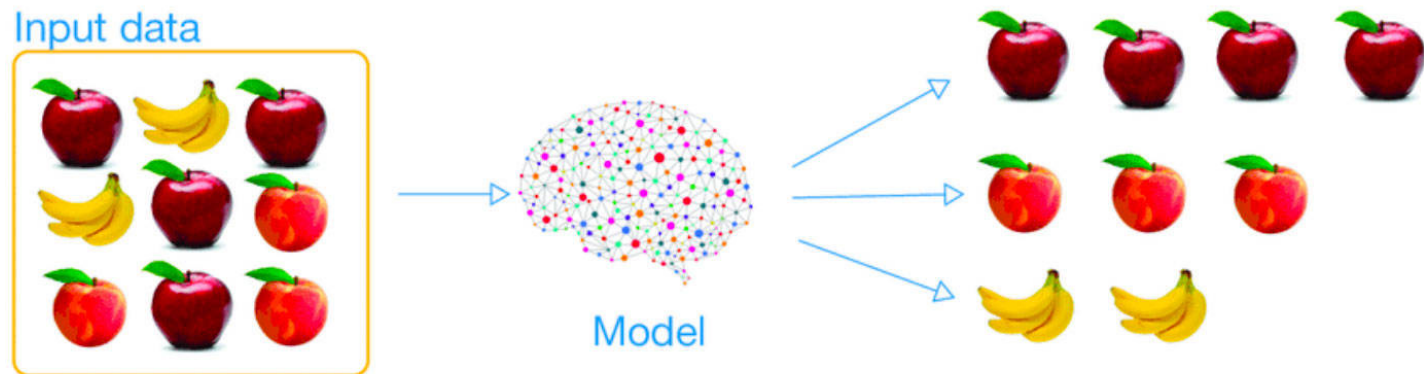
Today

- Clustering
- K-means



What is Clustering?

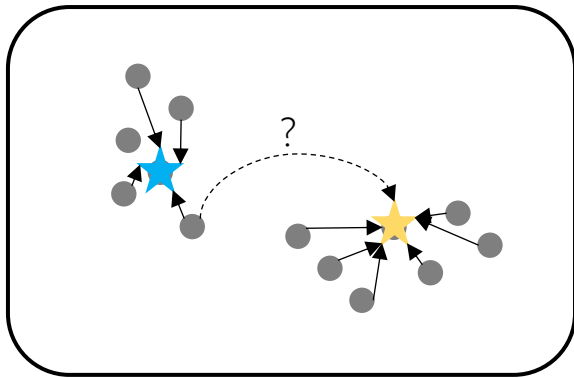
- **Clustering** = the organization of unlabeled data into similarity groups (called clusters).
- A cluster (簇) is a collection of data items which are “**similar**” between them, and “**dissimilar**” to data items in other clusters.



Q: How can we characterize “Similar” and “Dissimilar”?

How to Represent Clusters ?

- Suppose we have figured out the clusters for a dataset (e.g., by eyes), how can we represent them?



- For each cluster, elect a “**center**” point which best represents the whole cluster.
- Each data instance is assigned with a “**label**” indicating its **membership** to each cluster.



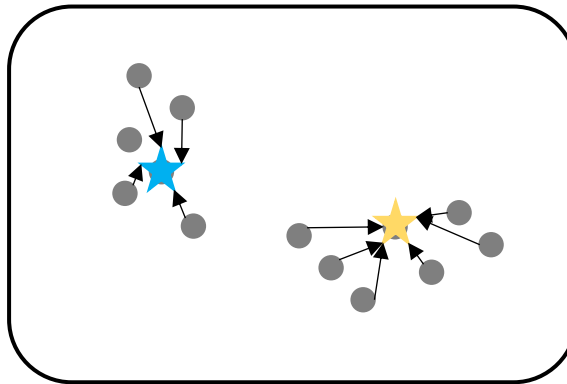
“Similar” or “Dissimilar”?

- The centroid must be more **similar** to instances in the represented cluster than other instances.
- Instances within a cluster are more **similar** to the centroid of the same cluster than centroids of other clusters.

A Machine Learning Perspective

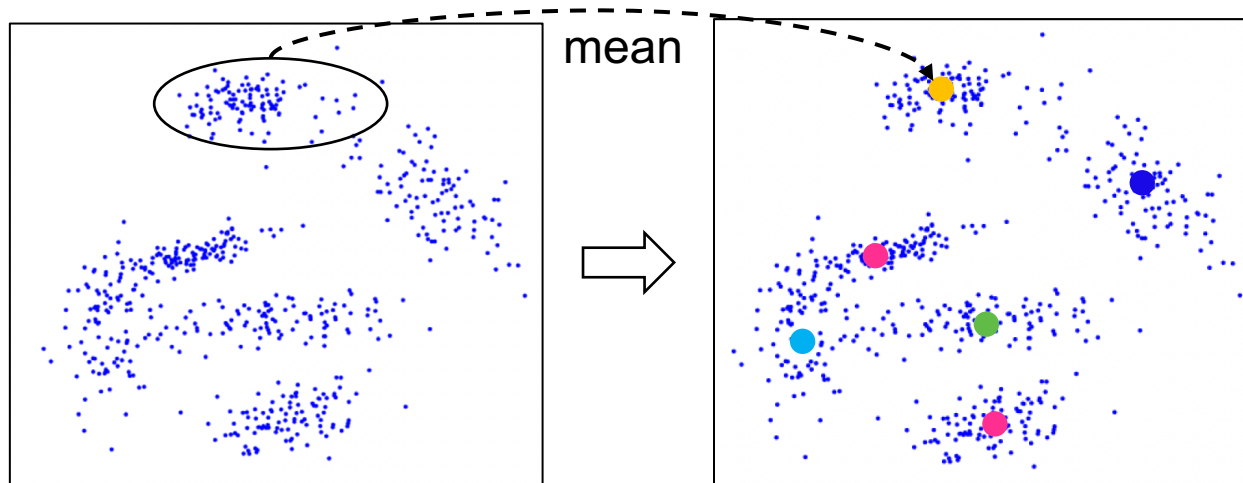
- What can be **optimized** in this “center-membership” framework?

- The “**centers**” should be **voted** according to the pairwise distances of the instances in the same cluster. [**center update**]
- Having obtained the centroids, we need to **optimize** the “**membership**” (label) of each data point. [**label update**]
- The above two steps can be repeated alternatively.



K-Means [MacQueen, 1967]

- Partition the data into k clusters, where the **center** for each cluster is defined as the **mean** of all instances within the cluster, called **centroid**.



Problem Formulation

- Given a dataset $D = \{x_1, \dots, x_N\}$, where $x_i \in \mathbb{R}^d$ is a d dimensional vector.
- Find K reference vectors which best represent the clusters
- Assign each $x^{(i)}$ with a variable $m_i \in \{1, \dots, K\}$ to indicate its cluster membership.

Steps

1. Choose k (random) data points (**seeds**) to be the initial **centroids**, cluster centers.
2. Assign each data point to the **closest centroid**.

$$L_2(x, \mu^k) = \|x - \mu^k\| = \sqrt{\sum_{m=1}^d (x_i - \mu_m^k)^2}$$

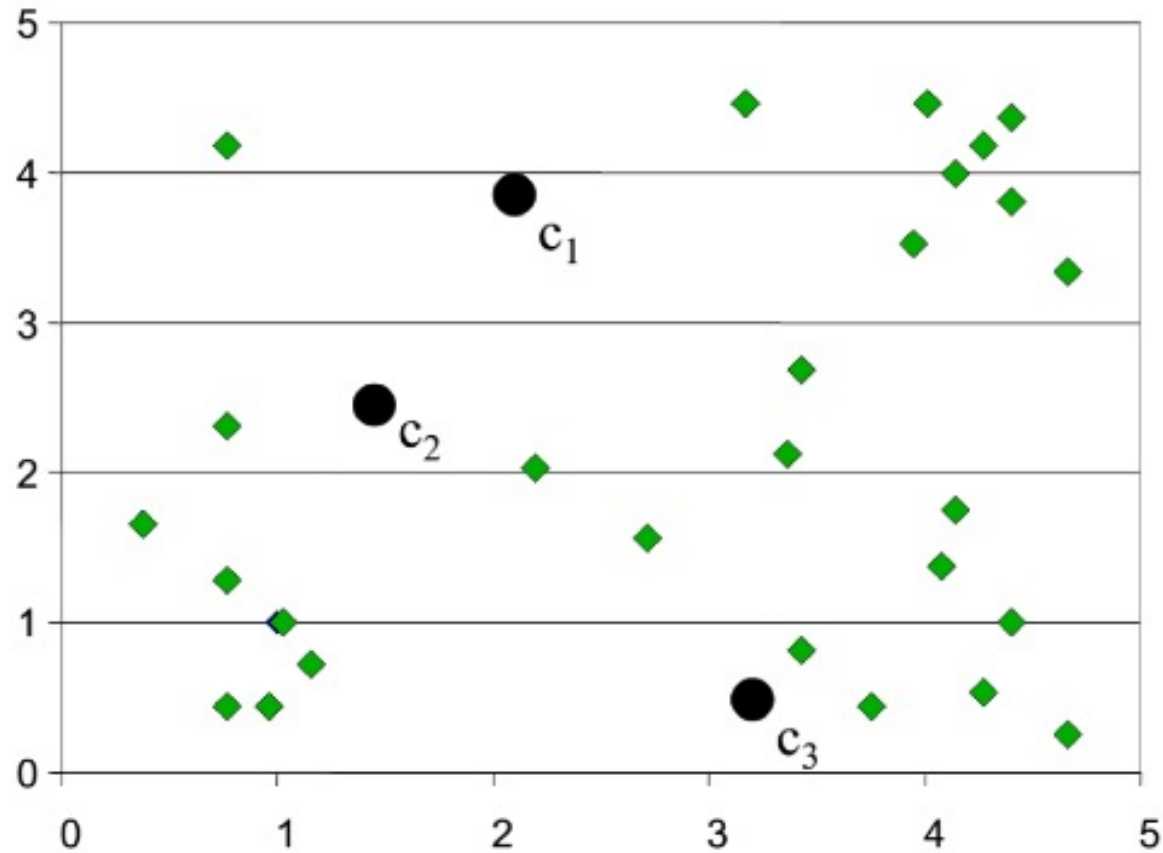
3. Re-compute the **centroids** using the current cluster memberships.

$$\mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

4. Repeat steps 2 and 3 until a convergence criterion is met.

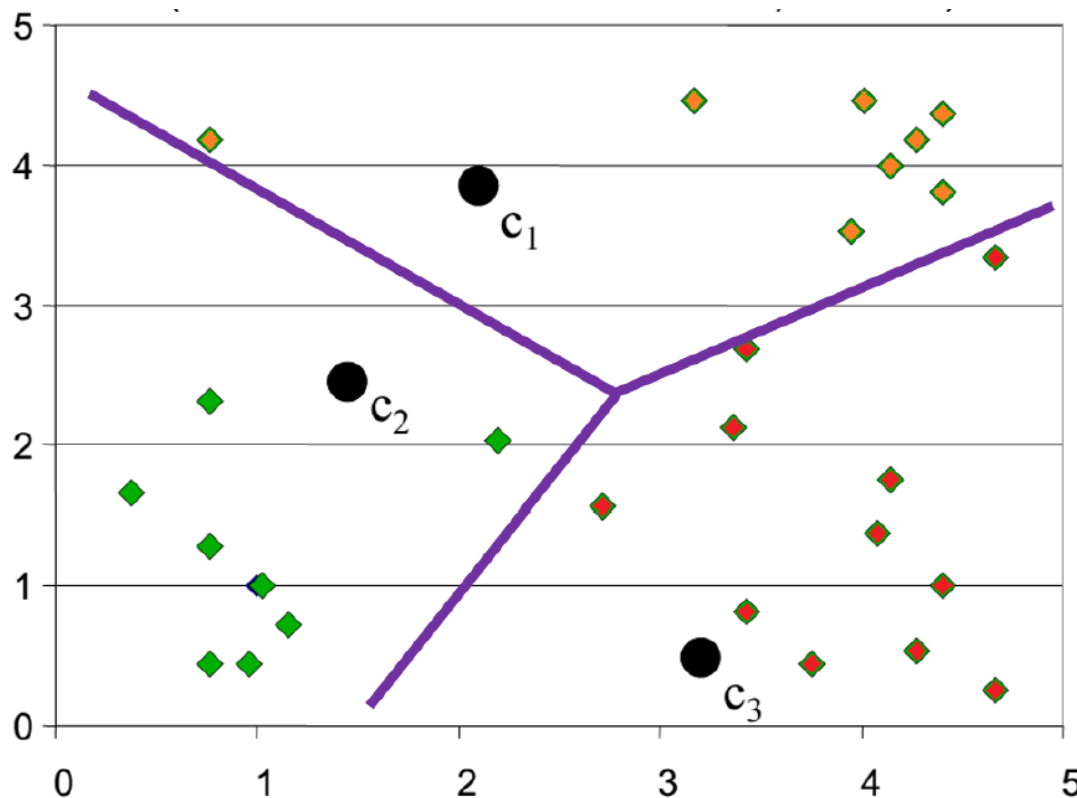
Example

Step1: randomly initialize the cluster centers.



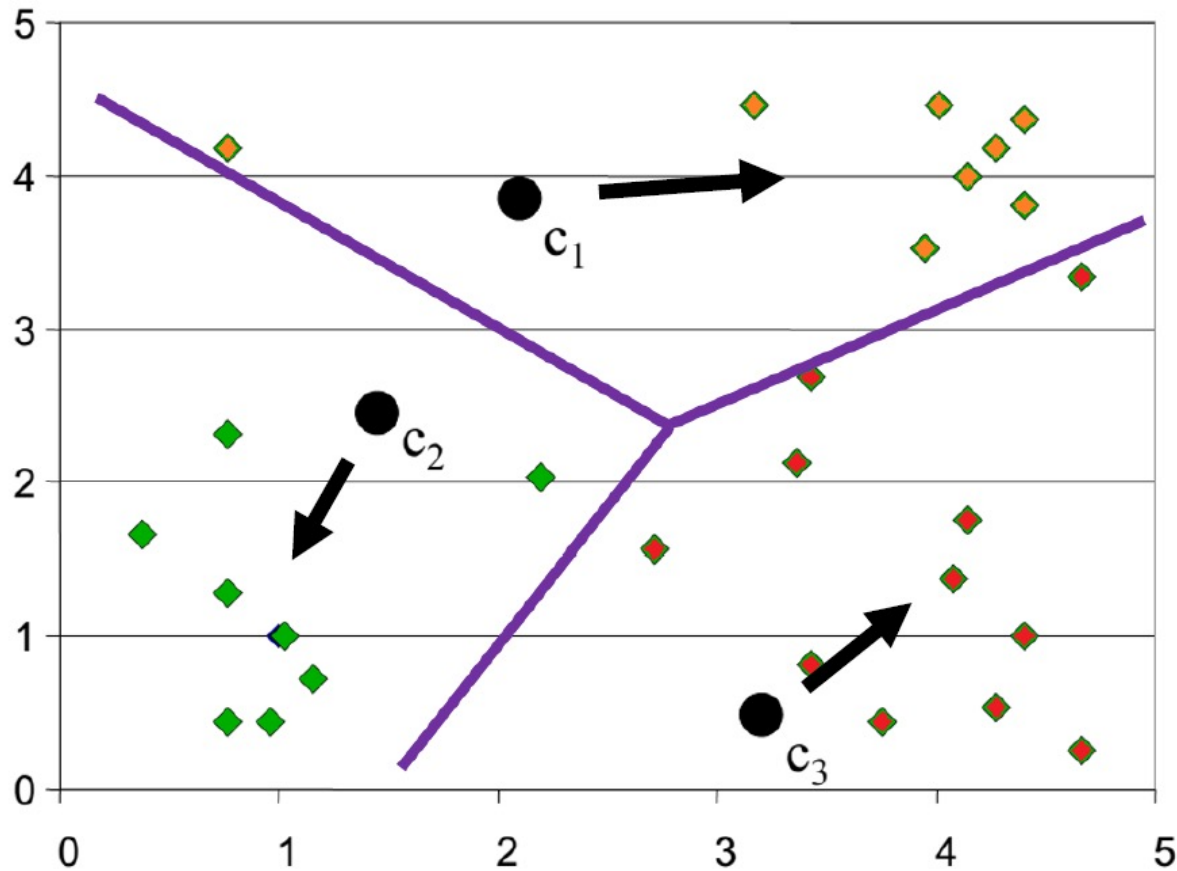
Example

Step2: determine cluster membership for each input
("winner-takes-all" inhibitory circuit)



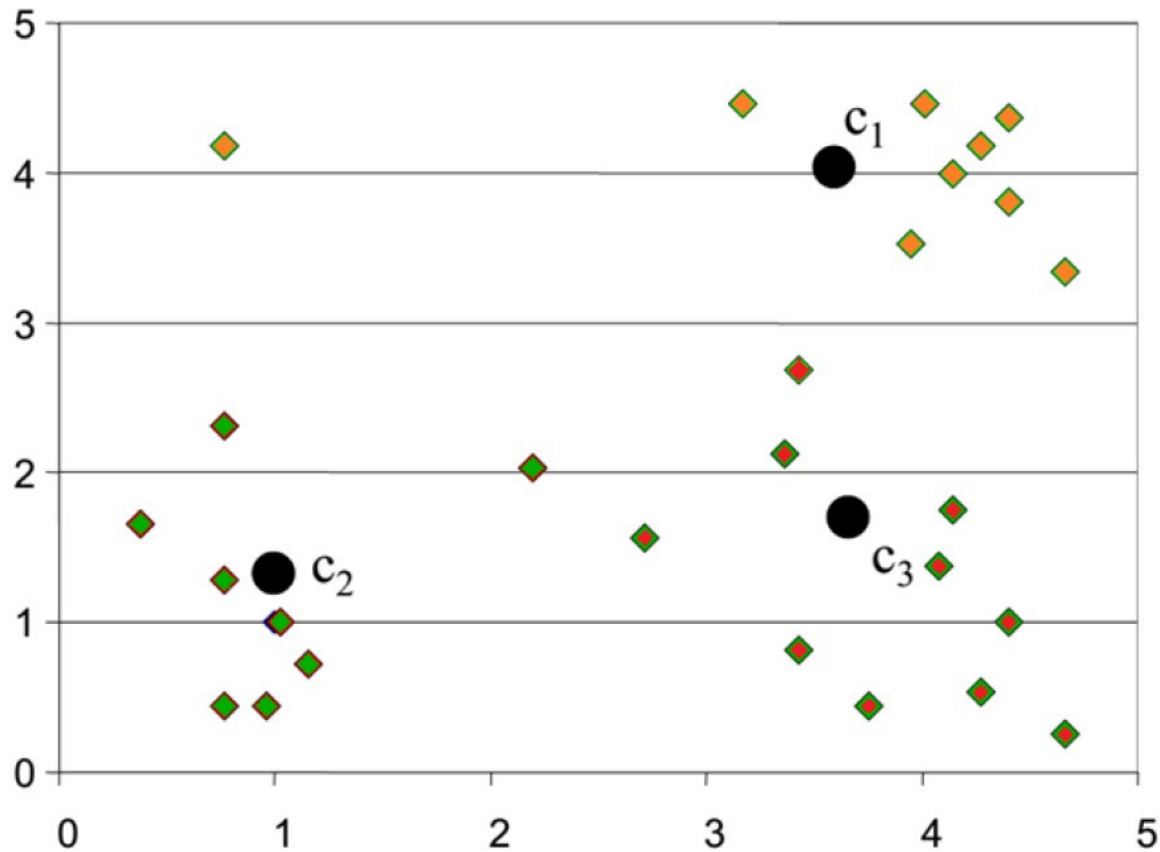
Example

Step3: re-estimate cluster centers (adapt synaptic weights)



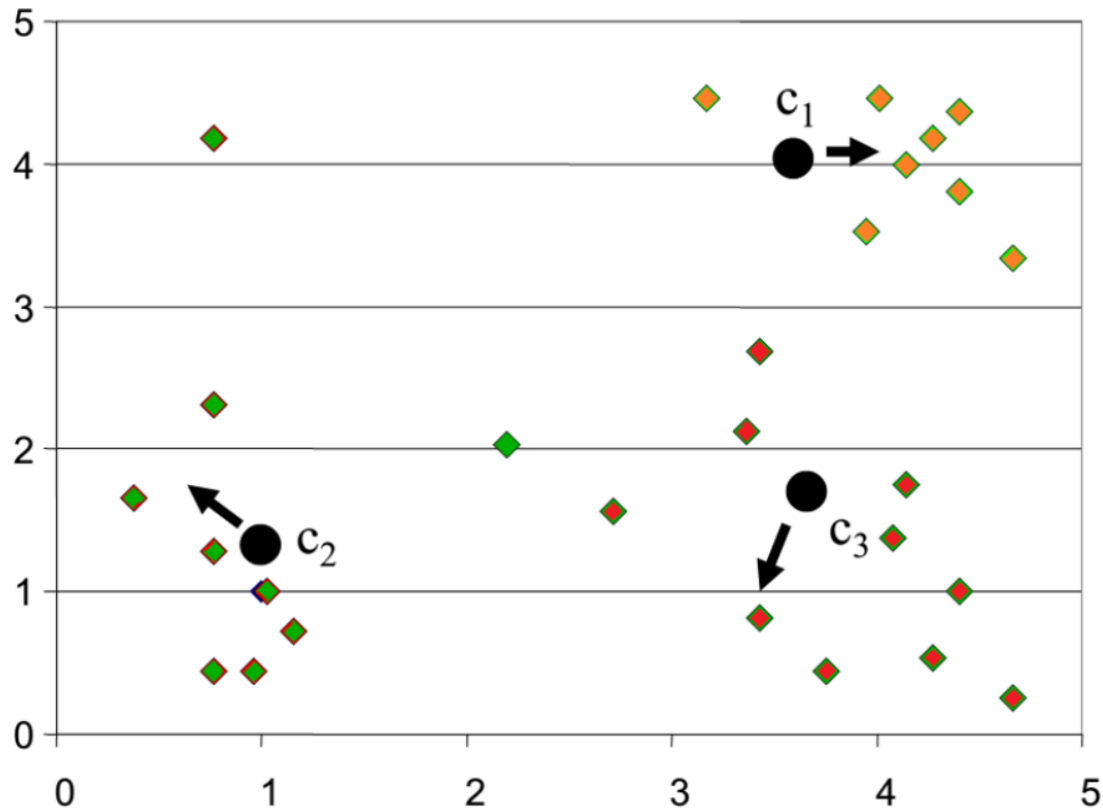
Example

Result of the first iteration



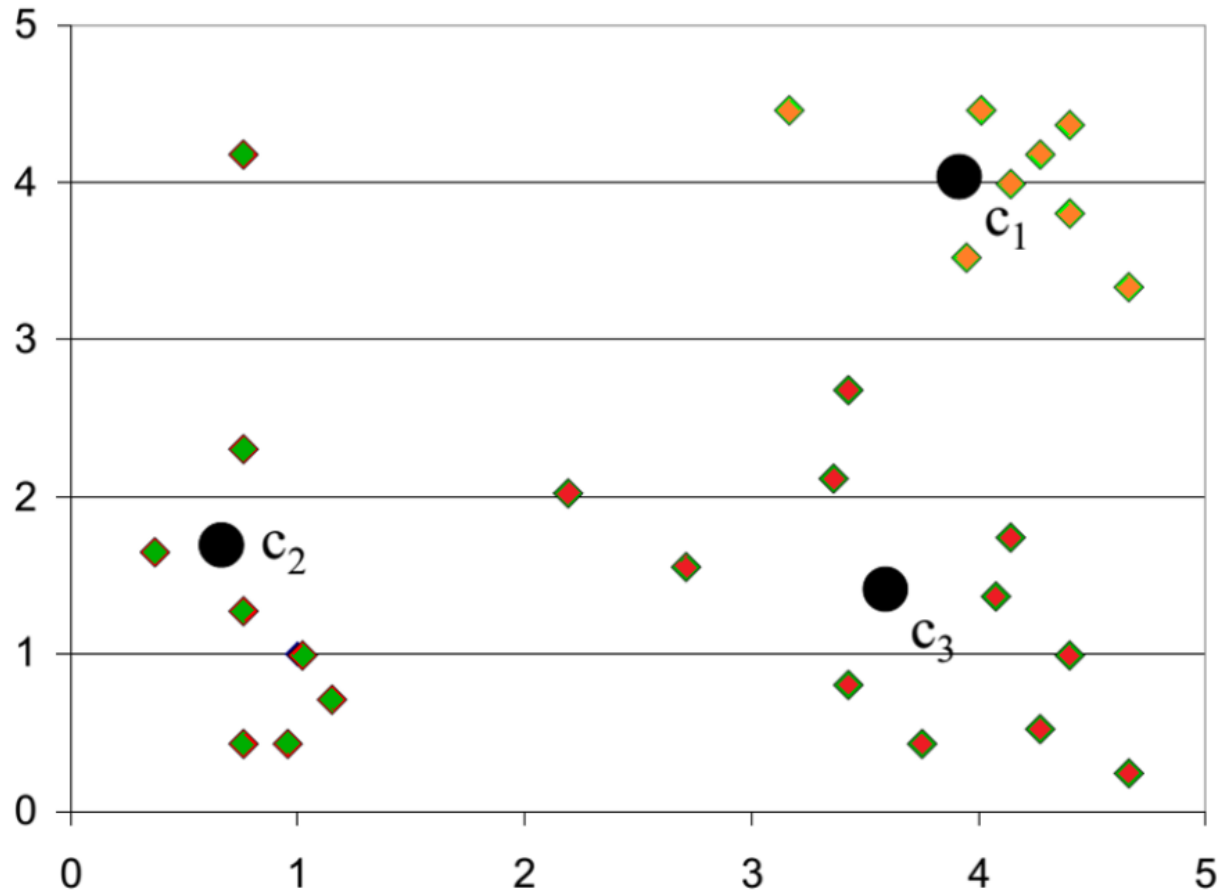
Example

The second iteration



Example

Result of the second iteration

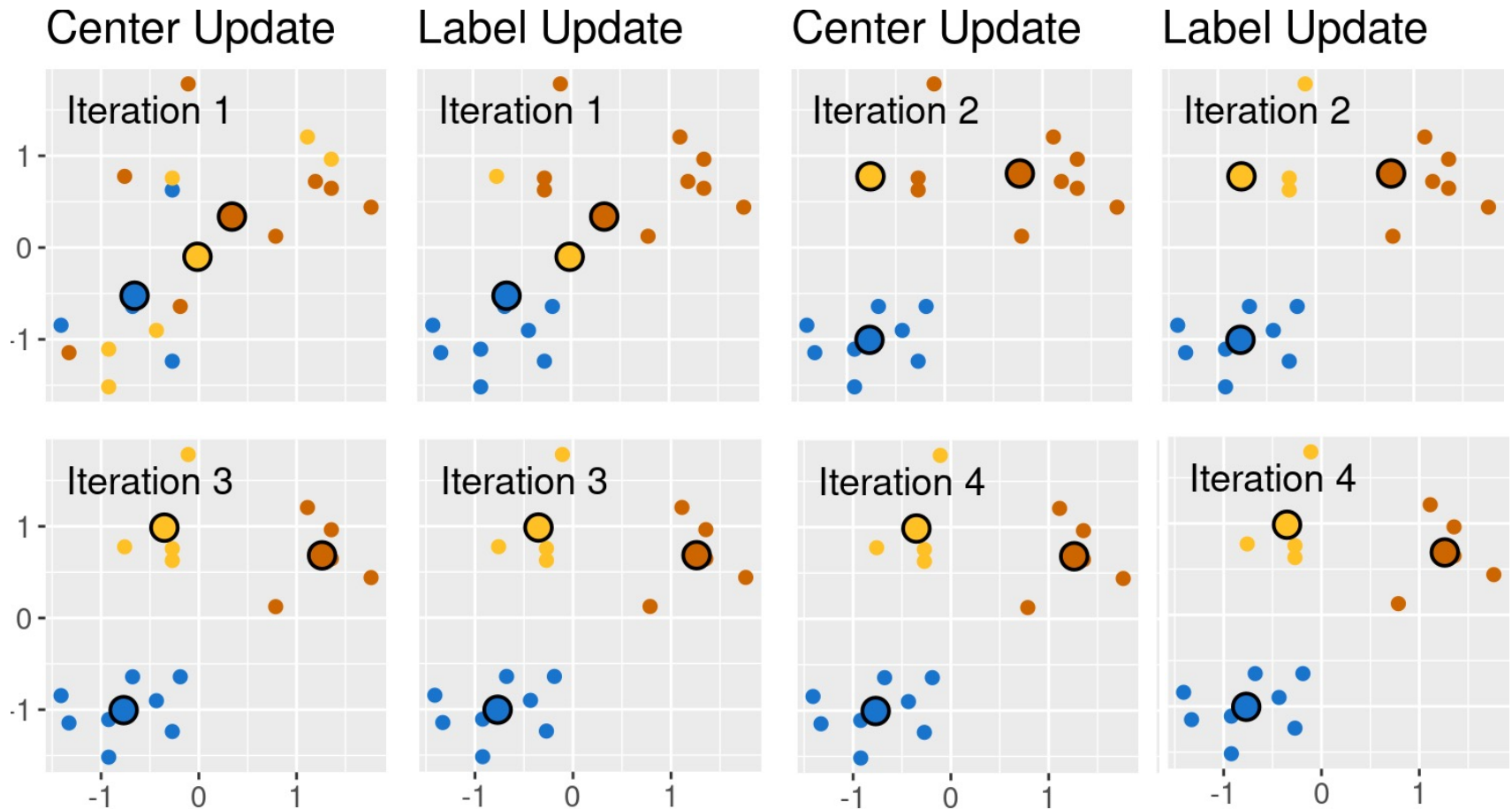


Convergence Criterion

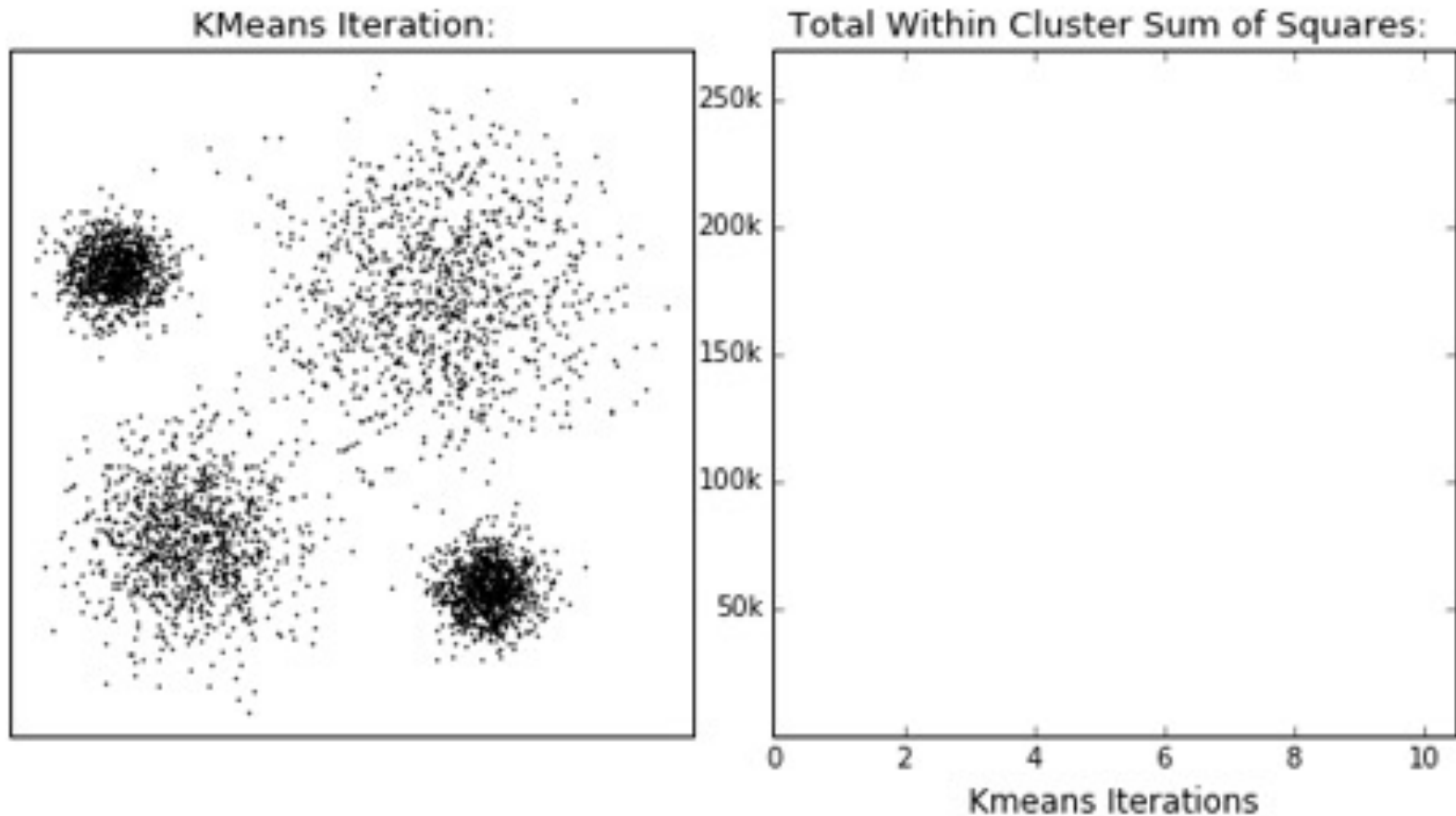
- No (or minimum) re-assignments of data points to different clusters, *or*
- No (or minimum) change of centroids, *or*
- Minimum decrease in the **sum of squared error (SSE)**:

$$\min_{\{\mu^k\}_{k=1}^K} \sum_{k=1}^K \sum_{x \in C_k} L(x - \mu^k) \qquad \mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

Example – A Big Picture



Example: Live K-Means



Time Complexity

- Assume computing distance between two instances is $O(d)$ where d is the dimensionality of the vectors.
- **Reassigning clusters:** $O(knd)$ distance computations.
- **Computing centroids:** Each instance vector gets added once to some centroid: $O(nd)$.
- Assume these two steps are each done once for I iterations: $O(Iknd)$

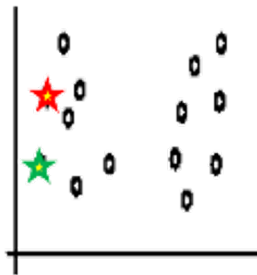
Local Optimum

- Finding the global optimum is NP-hard.
- The k -means algorithm is guaranteed to converge to a local optimum.

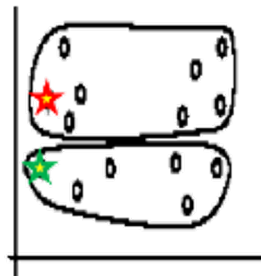
Q: What causes the uncertainty?

Sensitivity to Initial Centroids (Seeds)

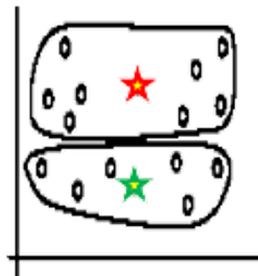
- Results can vary based on random seed selection.
- Some seeds can result in slow convergence or sub-optimal clusters.



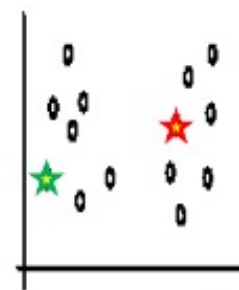
Random selection of seeds (centroids)



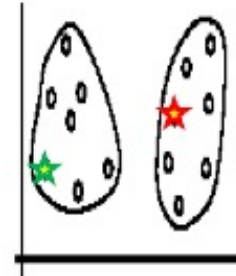
Iteration 1



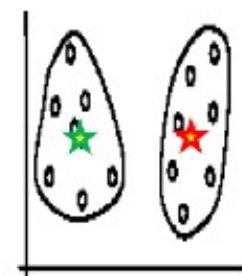
Iteration 2



Random selection of seeds (centroids)



Iteration 1

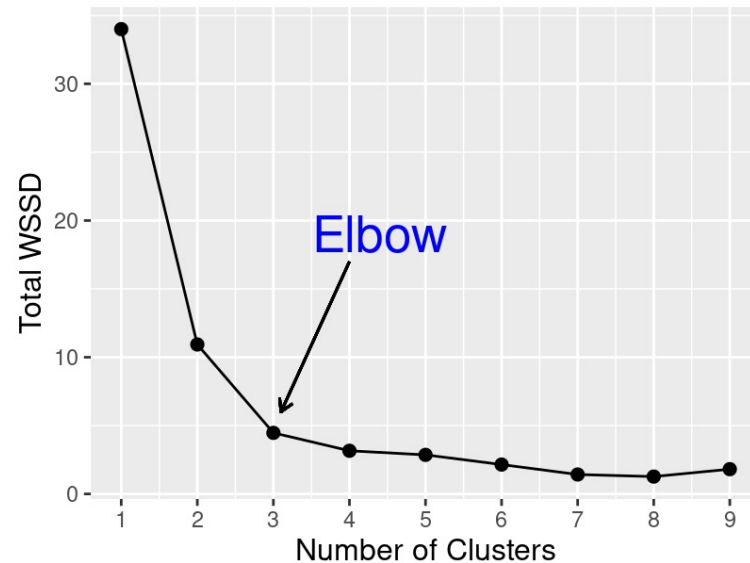


Iteration 2

How to choose the value of K?

- The Elbow method is the best way to find K.
- For each K, the *within-sum-of-squares* (WSS) is defined as the sum of the squared distance between each member of the cluster and its centroid.
- K with the least amount of WSS is taken as the optimum.

$$WSS = \sum_{i=1}^m (x_i - c_i)^2$$



How to choose the initial centroids?

- Select good seeds using a heuristic or the results of another method.

Example: [k-means++](#)

-
1. Choose one center uniformly at random from among the data points.
 2. For each data point x , compute $D(x)$, the distance between x and the nearest center that has already been chosen.
 3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to $D(x)^2$.
 4. Repeat Steps 2 and 3 until k centers have been chosen.
 5. Proceed using standard k -means clustering.
-

Pros and Cons

Advantages:

- **Simple:** easy to understand and implement,
- **Efficient:** k-means is considered a linear algorithm.

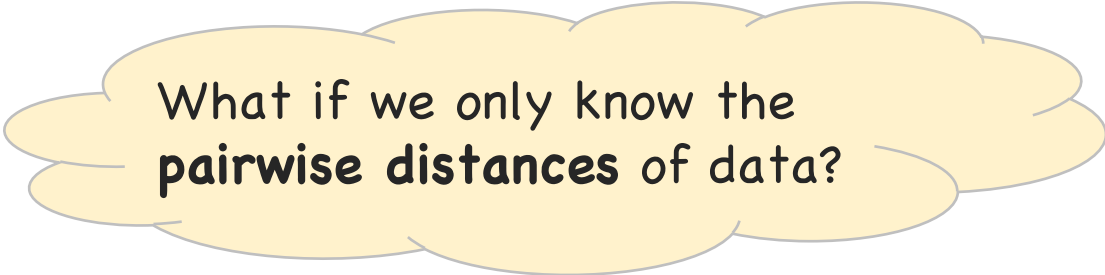


K-means is ~~one of~~ the most popular clustering algorithm.

Pros and Cons

Disadvantages:

- Assume that each cluster is a sphere (球形)
- The algorithm is only applicable if the **mean** is defined.
- The user needs to specify k .
- The algorithm is sensitive to **outliers**.



What if we only know the
pairwise distances of data?

TIME for Some Coding



<https://www.kaggle.com/code/adinishad/kmeans-clustering-from-scratch>

<https://www.kaggle.com/code/satishgunjal/tutorial-k-means-clustering>

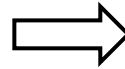
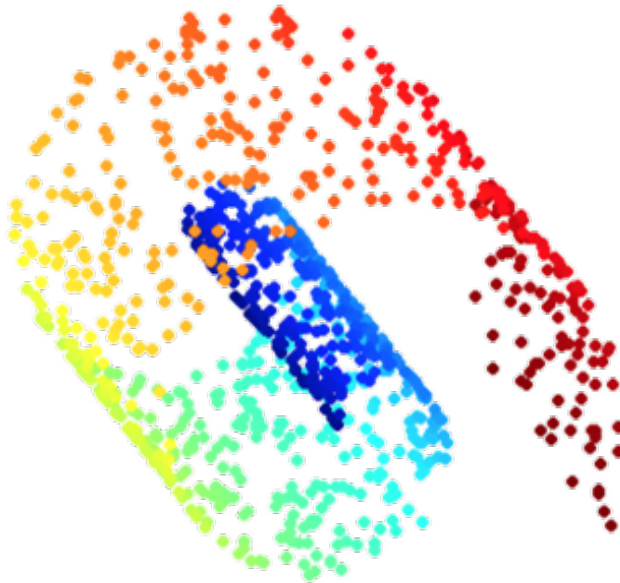


What's Next?

WHAT'S
NEXT?

Dimensionality Reduction

Looks 3D



Actually 2D

