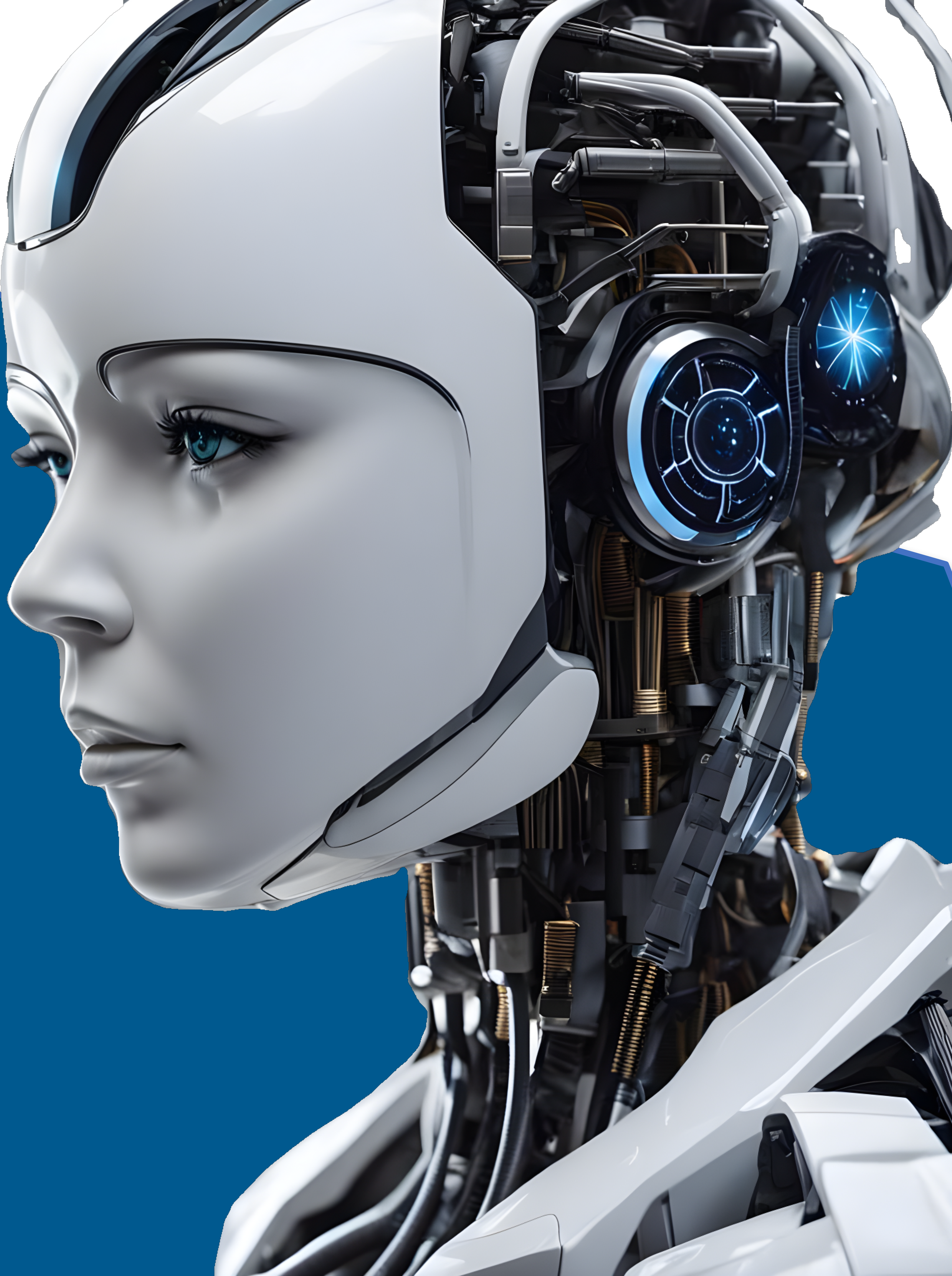
Tishk International University
IT Department
Course Code: IT-344/A

Introduction to Machine Learning

Unsupervised Learning

Lecture 9

Spring 2024
Hemin Ibrahim, PhD
hemin.ibrahim@tiu.edu.iq



Outline

- Supervised vs Unsupervised learning
- Unsupervised learning
- Clustering
- Distance Measures
- Similarity Measures
- K-means Clustering

localhost

File Edit View Run Kernel Tabs Settings Help

my_plot_demo.ipynb

Code

[13]:
```python
from datetime import datetime
import matplotlib.pyplot as plt
from meteostat import Point, Daily


def foo(x, y):
    x['Average Temperature'] = x['Average Temperature'].rolling(window=y).mean()
    x['Minimum Temperature'] = x['Minimum Temperature'].rolling(window=y).mean()
    x['Maximum Temperature'] = x['Maximum Temperature'].rolling(window=y).mean()
    return x


def bar(ax, events):
    for date, label in events.items():
        ax.annotate(label, xy=(date, data.loc[date, 'Maximum Temperature']),
                    xytext=(date, data.loc[date, 'Maximum Temperature'] + 5),
                    arrowprops=dict(facecolor='black', arrowstyle='->'))


start = datetime(2018, 1, 1)
end = datetime(2018, 12, 31)

location = Point(49.2497, -123.1193, 70)

data = Daily(location, start, end)
data = data.fetch()

data = data.rename(columns={
    'tavg': 'Average Temperature',
```

Simple 0 2 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 my_plot_demo.ipynb 0

Supervised vs. Unsupervised Learning

- Up to now we considered supervised learning scenario, where we are given
  - samples $x_1,..., x_n$
  - class labels for all samples $x_1,..., x_n$
  - This is also called learning with **teacher**, since correct answer (**the true class**) is provided.

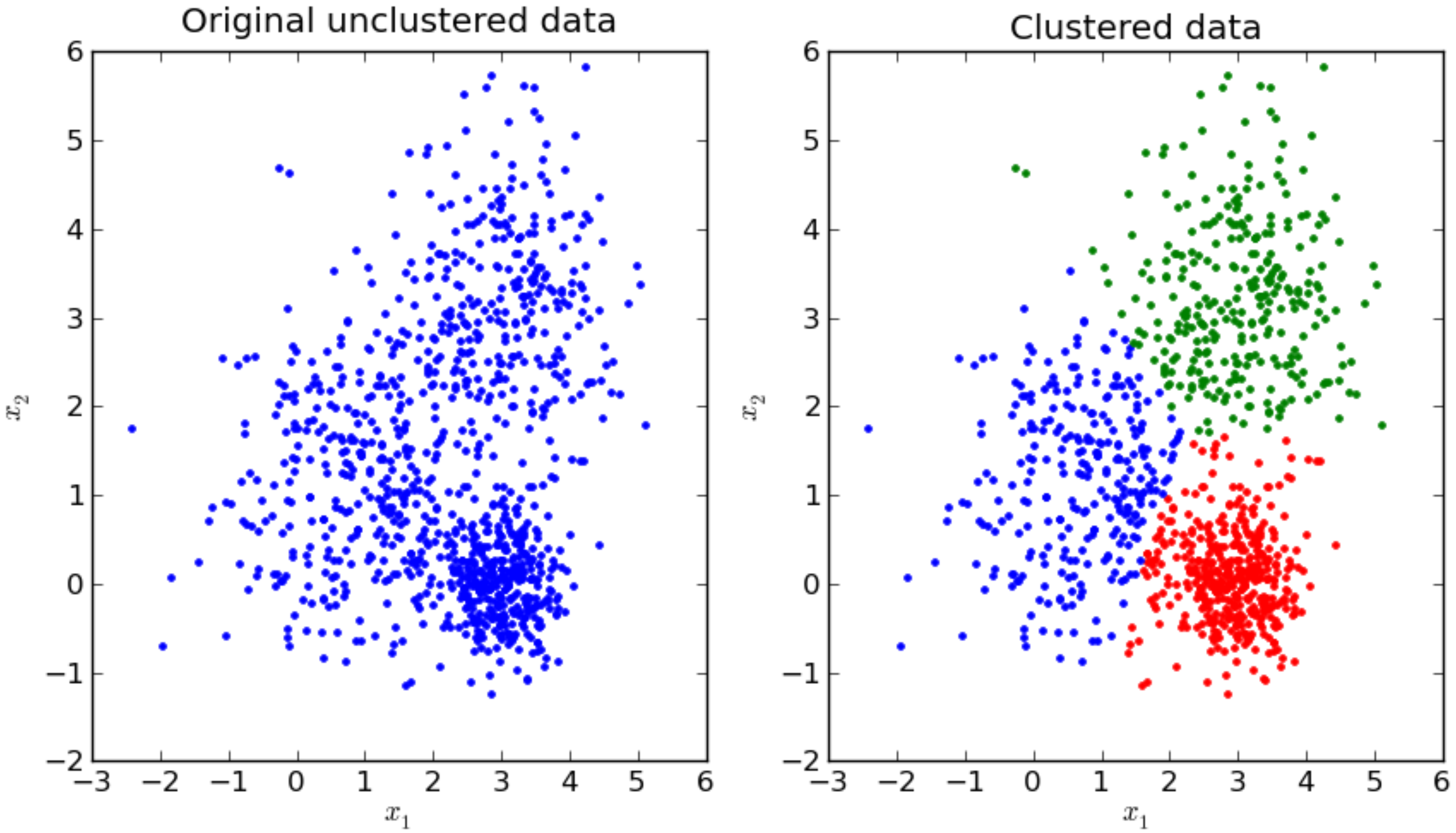
- In this lectures we consider unsupervised learning scenario, where we are only given samples $x_1,..., x_n$
  - This is also called learning **without teacher**, since correct answer is not provided
- <u>We do not split data into training and test sets</u>

Unsupervised Learning

- **Data is not labeled**
  - Non-Parametric Approach: group the data into clusters, each cluster (hopefully) says something about categories (classes) present in the data.
- **Unsupervised learning is harder**
  - How do we know if results are meaningful? No answer labels are available.
  - Let the expert look at the results (external evaluation)
  - Define an objective function on clustering (internal evaluation)
- **We nevertheless need it because**
  - Labeling large datasets is very costly (speech recognition), sometimes can label only a few examples by hand
  - May have no idea what/how many classes there are (data mining)
  - May want to use clustering to gain some insight into the structure of the data before designing a classifier, **Clustering** as data description

Clustering

- **Seek "natural" clusters in the data**
- What is a **good clustering?**
  - internal (within the cluster) distances <u>should be small</u>
  - external (intra-cluster) <u>should be large</u>
  - Clustering is a way to discover new categories (classes)

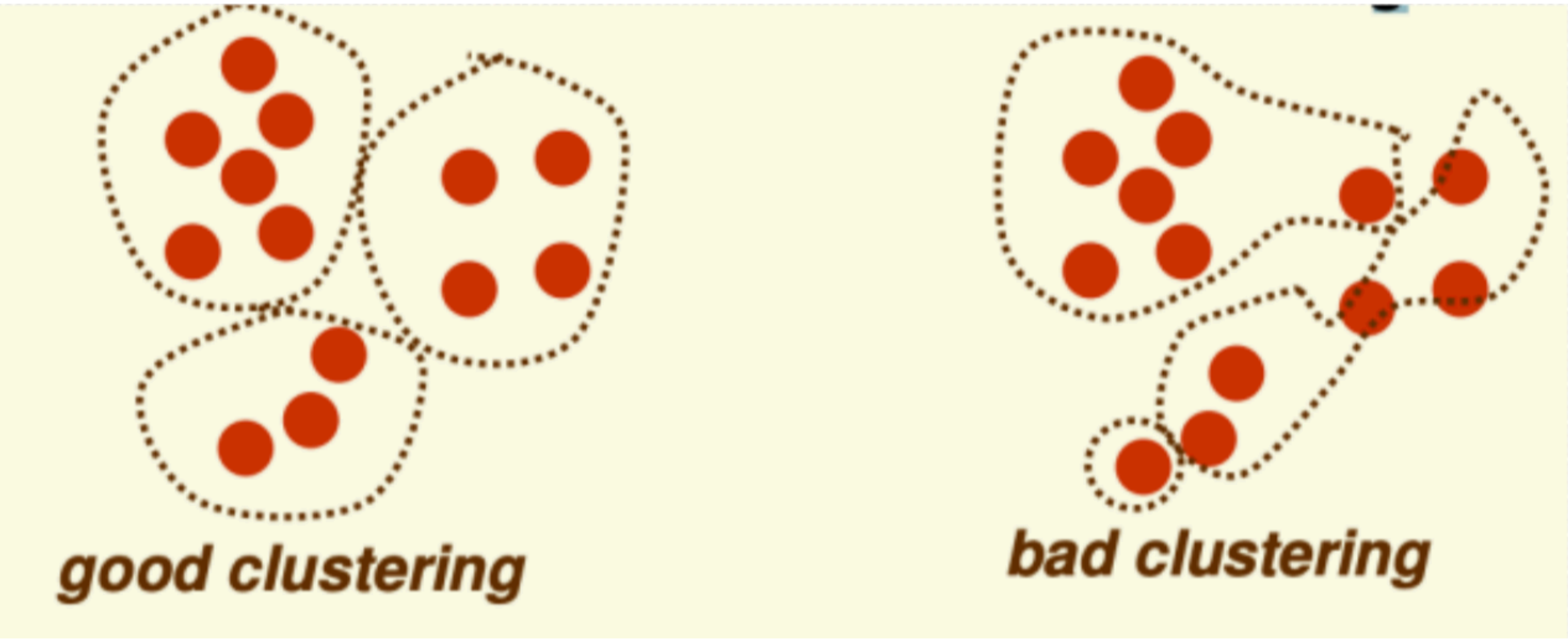


What we Need for Clustering

- **Proximity measure, either**
  - *similarity measure* $\mathbf{s}(x_i,x_k)$*: large if* x_i,x_k *are similar*
  - *dissimilarity(or distance) measure* $\mathbf{d}(x_i,x_k)$*: small if* x_i,x_k *are similar*

- **Criterion function to evaluate a clustering (clustering evaluation measure)**
  - How well a clustering algorithm has grouped the data points.

- **Algorithm to compute clustering**
  - For example, by optimizing the criterion function such as K-means, Hierarchical clustering..

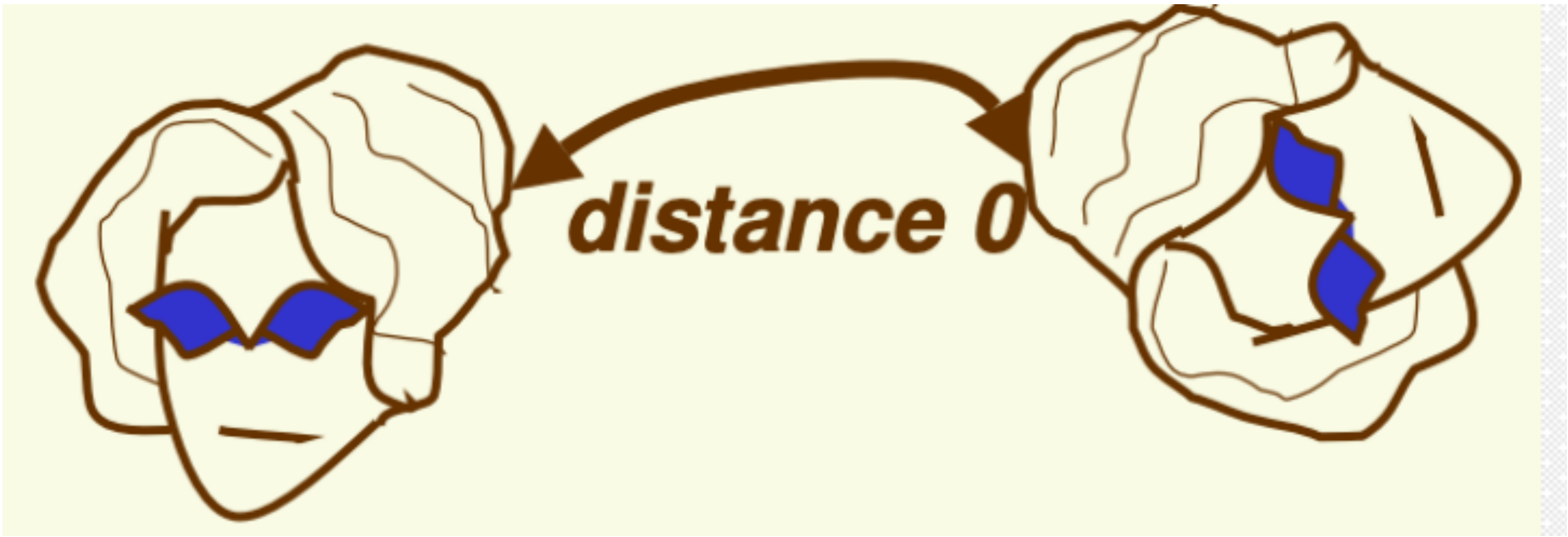


How Many Clusters?

- **Possible approaches**
  - fix the number of clusters to **k**
  - find the best clustering according to the criterion function (number of clusters may vary)

Proximity Measures

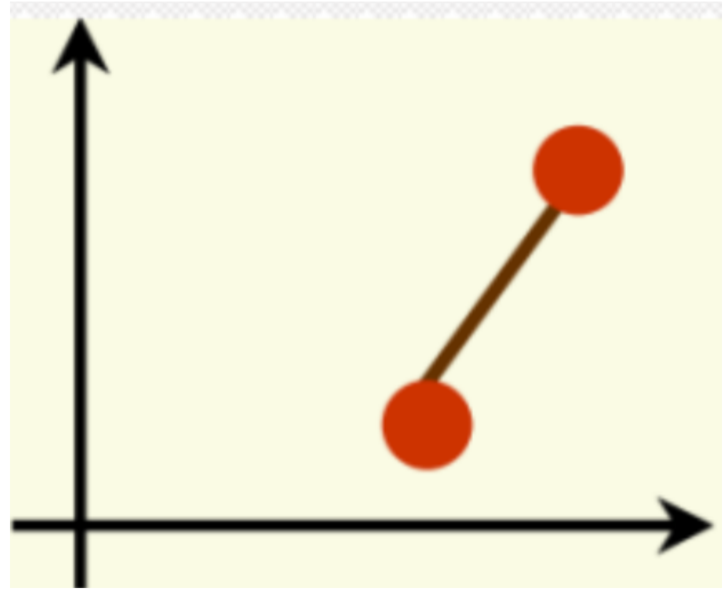
- A "good proximity measure" refers to a reliable method of gauging the closeness or similarity between objects or data points.
- Clusters should remain the same even if we make changes to the data that are typical or expected for the specific problem we're working on.
- For example for object recognition, should have invariance to rotation



Distance (dissimilarity) Measures

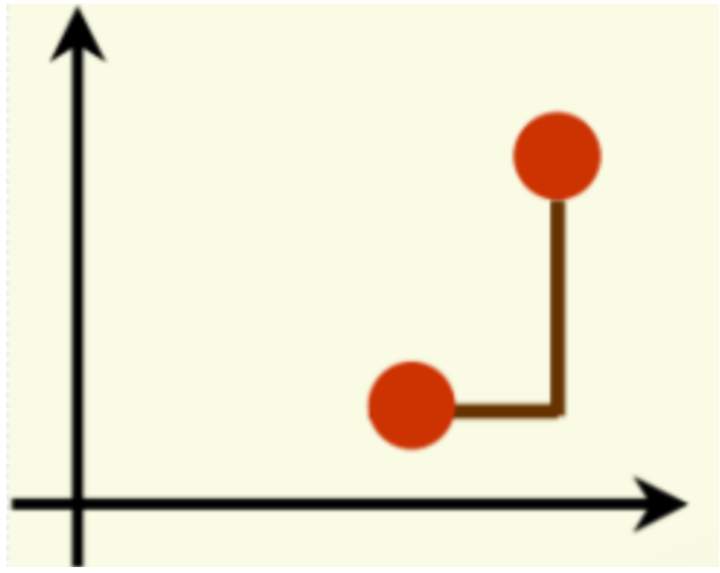
- **Euclidean distance**: translation invariant

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} \left(x_i^{(k)} - x_j^{(k)}\right)^2}$$



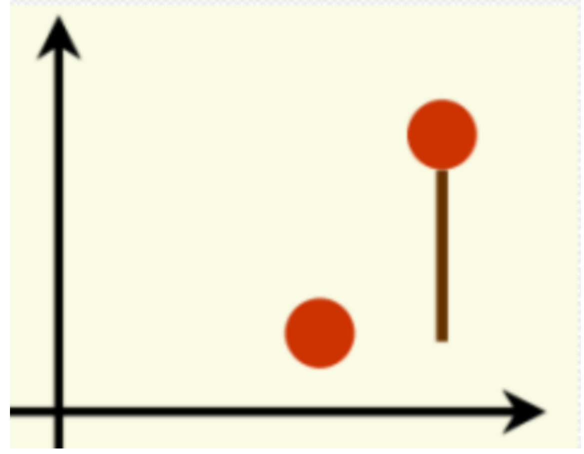
- **Manhattan (city block) distance**: approximation to Euclidean distance, cheaper to compute.

$$d(x_i, x_j) = \sum_{k=1}^{d} \left| x_i^{(k)} - x_j^{(k)} \right|$$



- **Chebyshev distance**: approximation to Euclidean distance, cheapest to compute

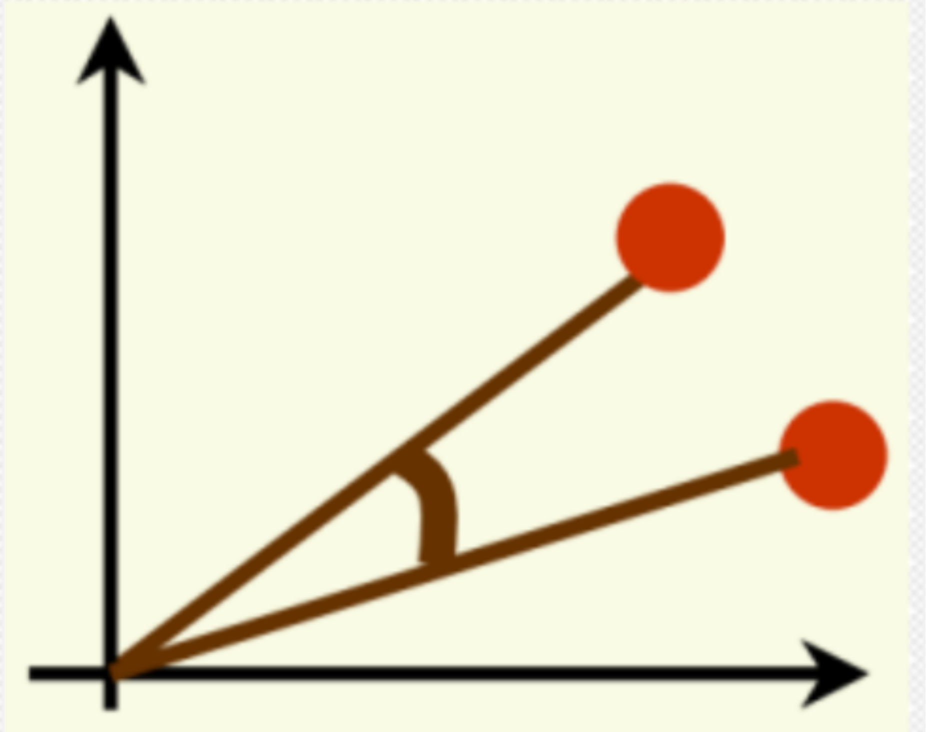
$$d(x_i, x_j) = \max_{1 \le k \le d} \left| x_i^{(k)} - x_j^{(k)} \right|$$



Similarity Measures

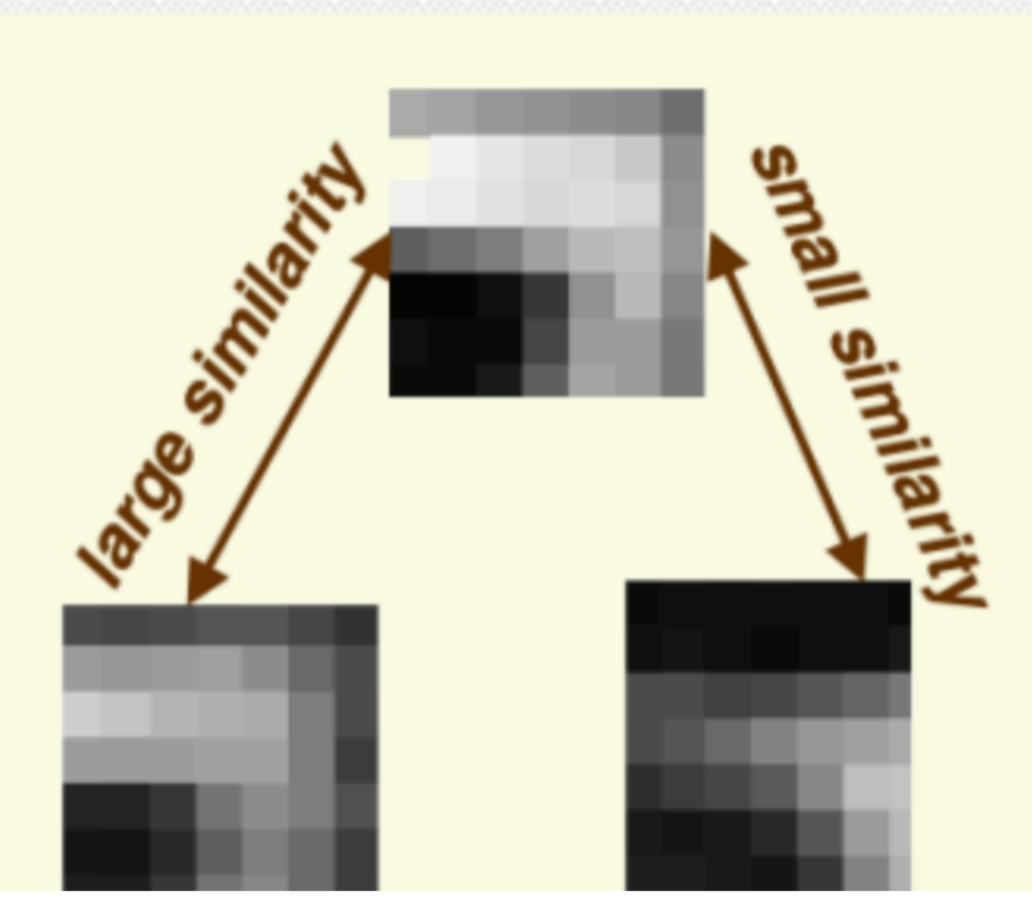
- **Cosine similarity:**
  - the smaller the angle, the larger
  - the similarity
  - scale in variant measure
  - popular in text retrieval

$$s(x_i, x_j) = \frac{\sum_{k=1}^{d}\left(x_i^{(k)} - \bar{x}_i\right)\left(x_i^{(k)} - \bar{x}_i\right)}{\left[\sum_{k=1}^{d}\left(x_i^{(k)} - \bar{x}_i\right)^2 \sum_{k=1}^{d}\left(x_j^{(k)} - \bar{x}_j\right)^2\right]^{1/2}}$$



- **Correlation coefficient:**
  - popular in image processing

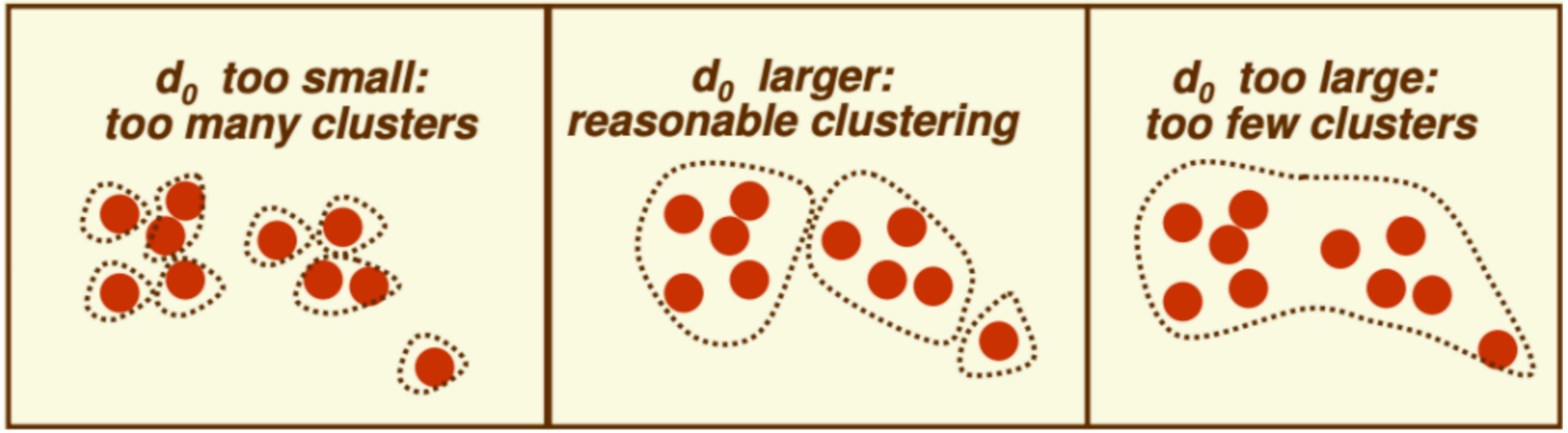
$$s(x_i, x_j) = \frac{\sum_{k=1}^{d}\left(x_i^{(k)} - \bar{x}_i\right)\left(x_i^{(k)} - \bar{x}_i\right)}{\left[\sum_{k=1}^{d}\left(x_i^{(k)} - \bar{x}_i\right)^2 \sum_{k=1}^{d}\left(x_j^{(k)} - \bar{x}_j\right)^2\right]^{1/2}}$$



Simplest Clustering Algorithm

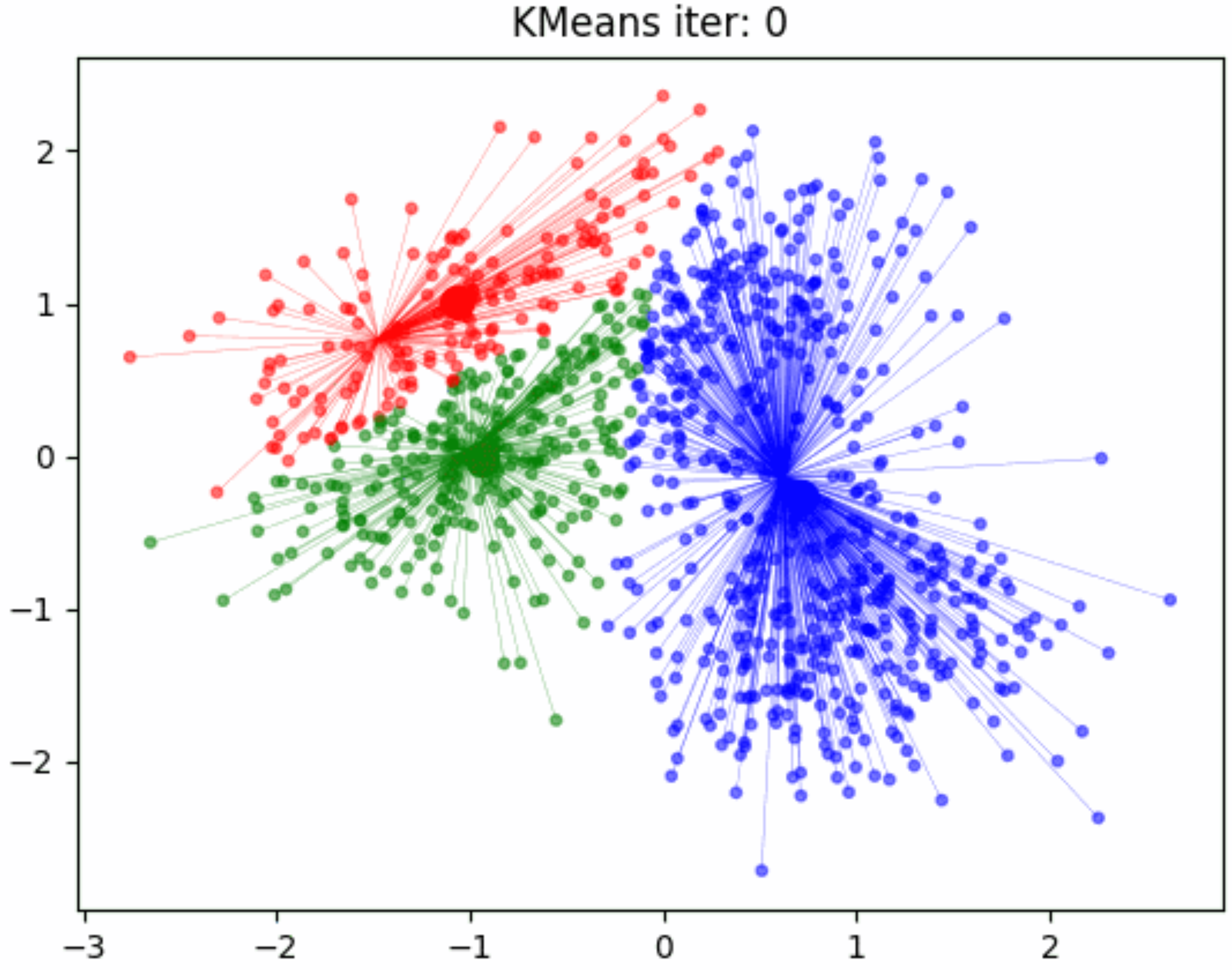
- Having defined a proximity function, can develop a simple clustering algorithm
- **go over all sample pairs**, and put them in the same cluster if the distance between them is less then some threshold distance d0 (or if similarity is larger than s0)

- **Advantages**: simple to understand and implement
- **Disadvantages**: very dependent on d0 (or s0), automatic choice of d0 (or s0) is not an easily solved issue.



K-means Clustering

K-means clustering is a popular unsupervised machine learning algorithm used for partitioning data into clusters. The goal is to divide a set of data points into clusters such that points within the same cluster are more similar to each other than they are to points in other clusters



K-means Clustering

- **Initialization**: Choose K initial centroids randomly from the data points. These centroids represent the center of each cluster.
- **Assignment**: Assign each data point to the nearest centroid, creating K clusters.
- **Update**: Recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster
- **Repeat**: Repeat steps 2 and 3 until the centroids no longer change significantly or a maximum number of iterations is reached.

K-means Clustering - Example

- Suppose we have the following dataset: {1,2,3,6,7,8}
- **Initialization**: We randomly choose two points from the dataset as initial centroids, let's say 2 and 7.
- **Assignment**: We calculate the distance of each point to the centroids and assign each point to the nearest centroid. For example, 2 is closer to 1 than 8, so it belongs to the first cluster. Similarly, 2 belongs to the first cluster, 3 belongs to the first cluster, 6 belongs to the second cluster, 7 belongs to the second cluster, and 8 belongs to the second cluster.
- **Update**: We recalculate the centroids of each cluster by taking the mean of the points in each cluster. The new centroids might be 2 for the first cluster and 7 for the second cluster.
- **Repeat**: We repeat the assignment and update steps until convergence.

K-means Clustering - Example

Step 1: Initialize centroids randomly. Let's say we choose:

$C1=2$

$C2=7$

Calculate the distance from each point to each centroid:

- For $C1=2$
  - Distance to 1 = 1
  - Distance to 2 = 0
  - Distance to 3 = 1
  - Distance to 6 = 4
  - Distance to 7 = 5
  - Distance to 8 = 6

K-means Clustering - Example

- C2=7
  - Distance to 1 = 6
  - Distance to 2 = 5
  - Distance to 3 = 4
  - Distance to 6 = 1
  - Distance to 7 = 0
  - Distance to 8 = 1

Assign each point to the nearest centroid:

- Cluster 1: {1, 2, 3}
- Cluster 2: {6, 7, 8}

K-means Clustering - Example

Step 2: Update centroids by taking the mean of all points assigned to each centroid:

$$C_1 = \frac{1+2+3}{3} = 2$$

$$C_2 = \frac{6+7+8}{3} = 7$$

K-means Clustering - Exercise

Perform K-means clustering manually on the following dataset:

{(1 , 2), (2 , 1), (2 , 3), (4 , 5), (5 , 4), (5 , 6)} Set the number of clusters (K=2).

Steps:

- Begin by randomly selecting two data points from the dataset as initial centroids.
- Assign each data point to the nearest centroid to form initial clusters.
- Calculate the mean of the points in each cluster to update the centroids.
- Repeat steps 2 and 3 until the centroids no longer change significantly.
- Present the final clusters, their centroids, and the data points assigned to each cluster.

Cluster2={(1 , 2), (2 , 1), (2 , 3)},

Cluster1={ (4 , 5), (5 , 4), (5 , 6)}

Thank You