UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Fall 2024)

Machine Learning

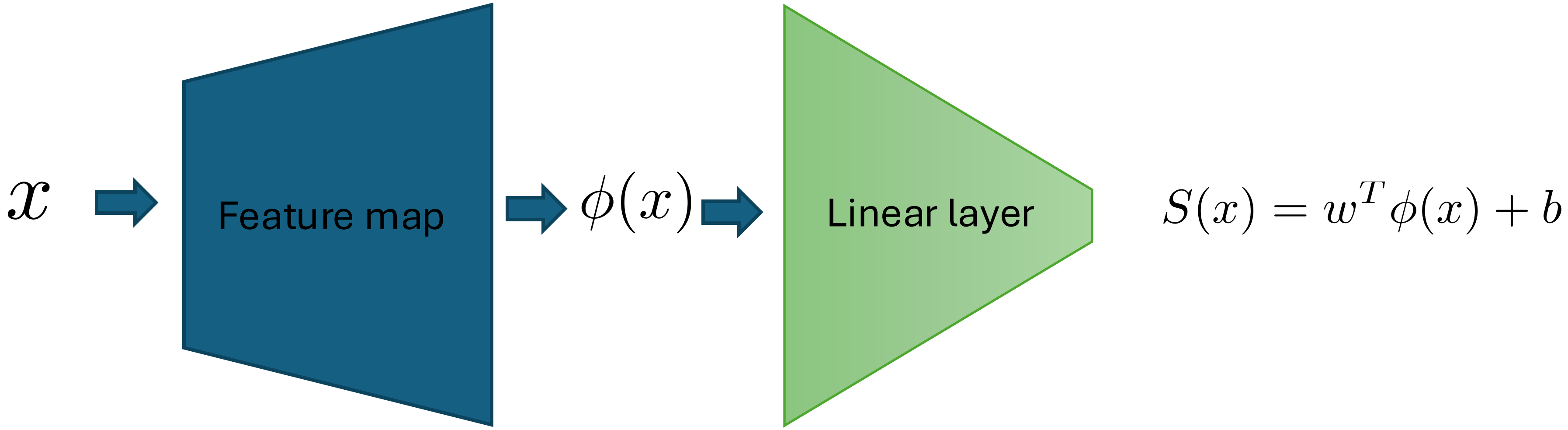
Lecture 18: Clustering

Chong Liu

Assistant Professor of Computer Science

Nov 14, 2024

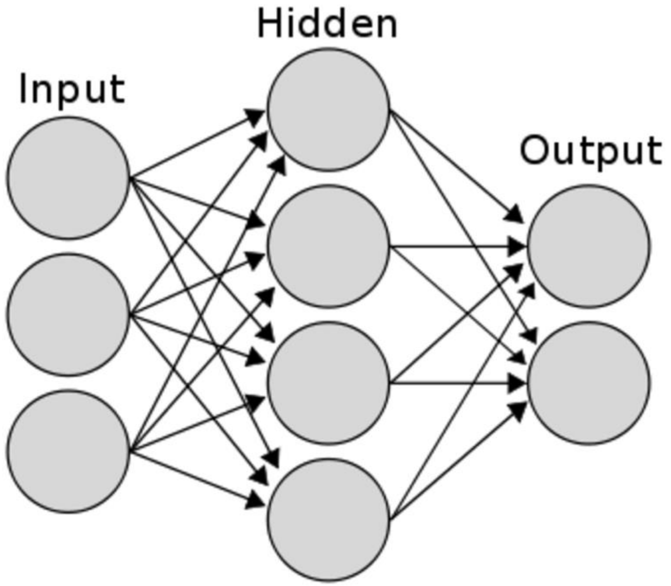
Recap: From kernels to neural networks



Recap: Two-layer neural networks

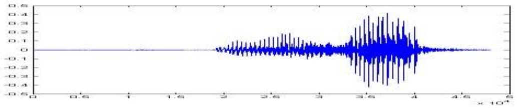
- Neural network: $S(x) \; = \; w_2^T(W_1 x + \boldsymbol{b}_1) + b_2$
  - Still a linear model at the end of the day, so let's add a nonlinearity σ!

- Two-layer MLP: $S(x) \; = \; w_2^T \sigma(W_1 x + \boldsymbol{b}_1) + b_2$
  - Linear model w.r.t. to a learnable feature map



Recap: Learning ≈ Configuring the learnable function so it behaves as instructed.

- Speech Recognition

$f\Big($  $\Big) =$ “Hello!”

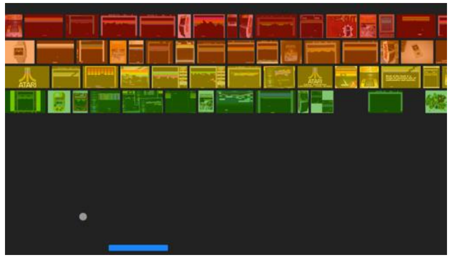
- Handwritten Recognition

$f\Big($  $\Big) =$ “2”

- Weather forecast

$f\Big($ Thursday $\Big) =$ “  Saturday”

- Play video games

$f\Big($  $\Big) =$ “move left”

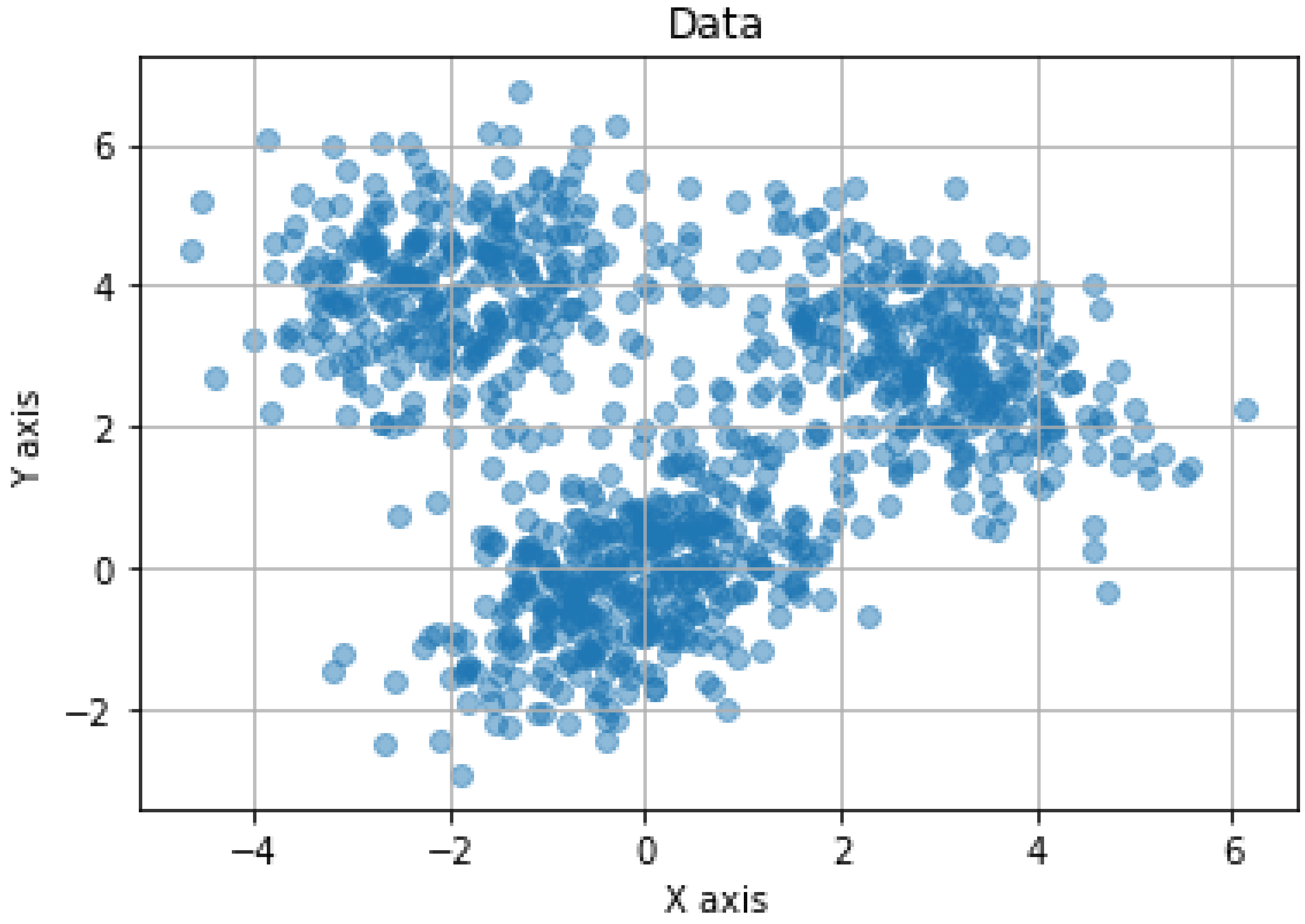
Unsupervised learning

- Input space: $\mathcal{X}$ Images, videos, text, graphs, proteins, programs, etc…
- Output space: None.
- Hypothesis space: $\mathcal{H}$
- Each hypothesis h is a particular way to summarize the data
- Loss function $\ell : \mathcal{H} \times \mathcal{X} \to \mathbb{R}$
- Goal:
  - Discover data structure
  - Often achieved by minimizing the loss

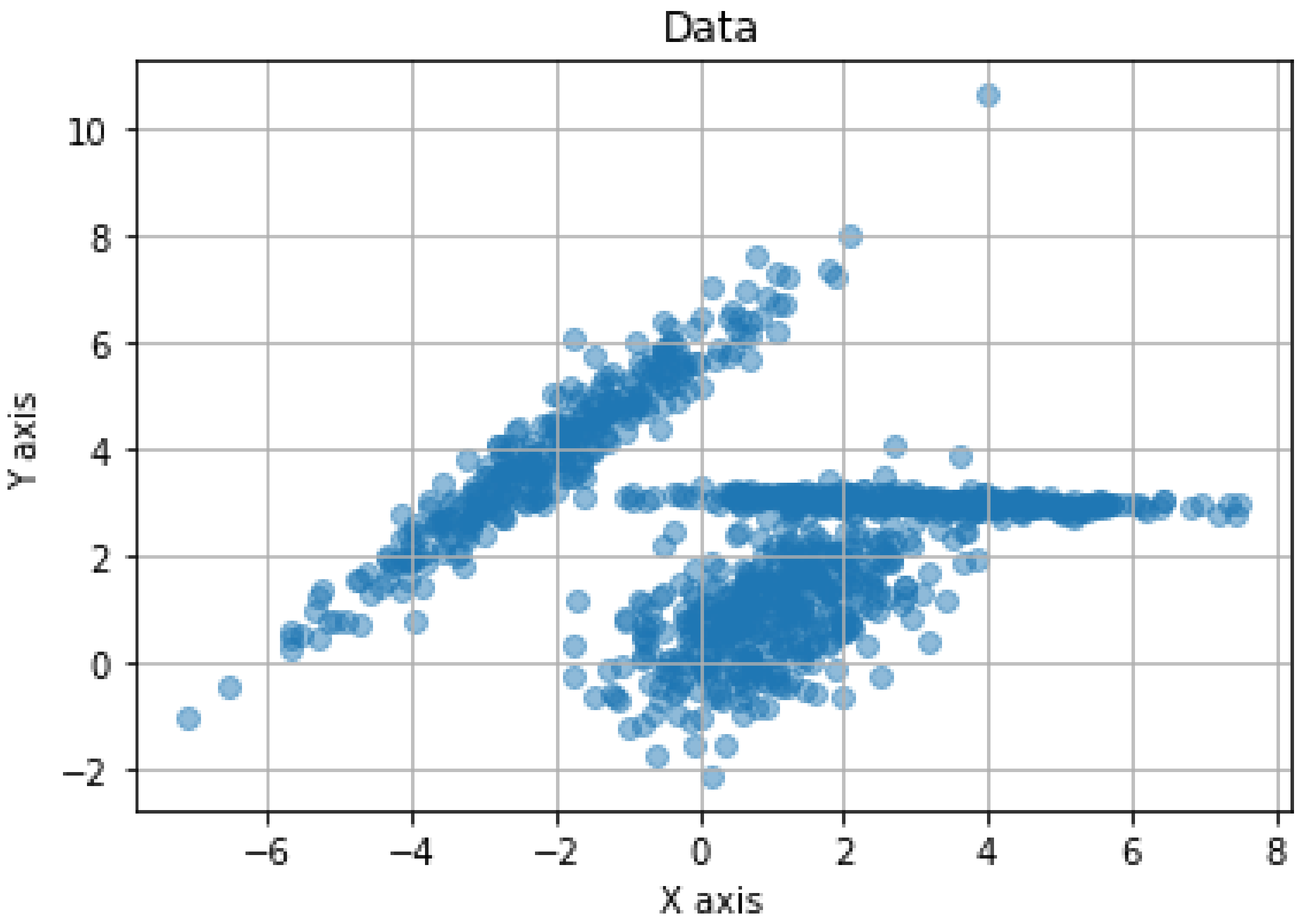
Goal of unsupervised learning is to **learn data structures** without labels

- Discussion: What kind of structures can you see?

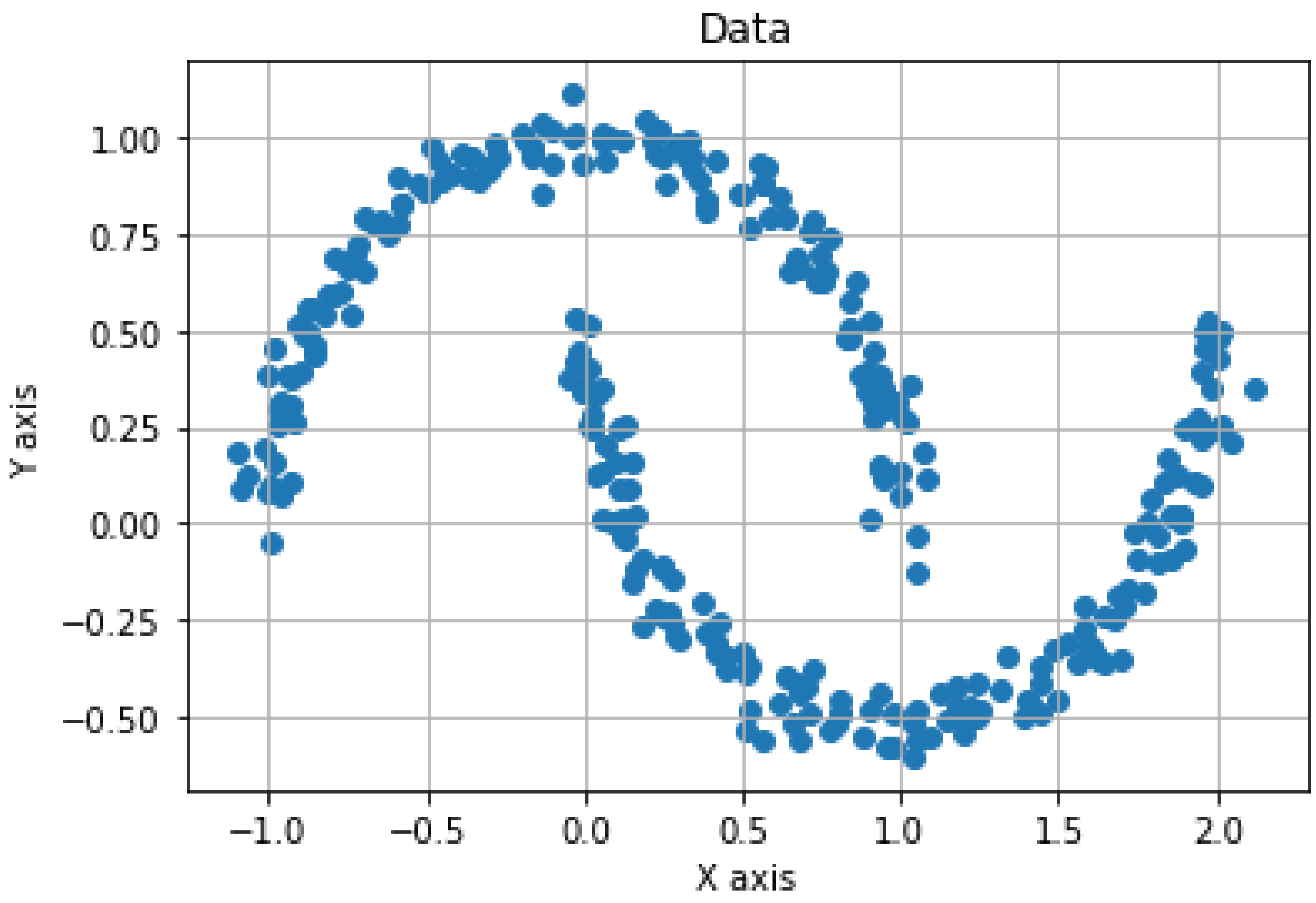
What kind of structures can you see?



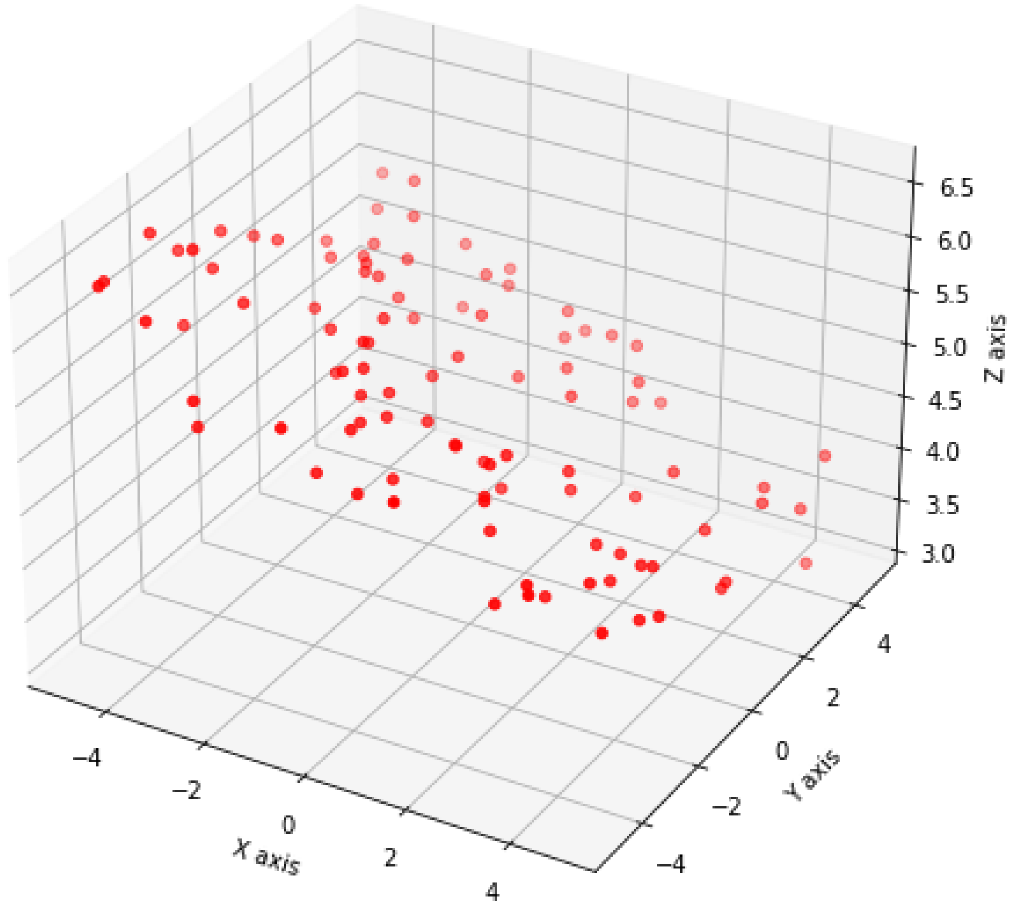
What kind of structures can you see?



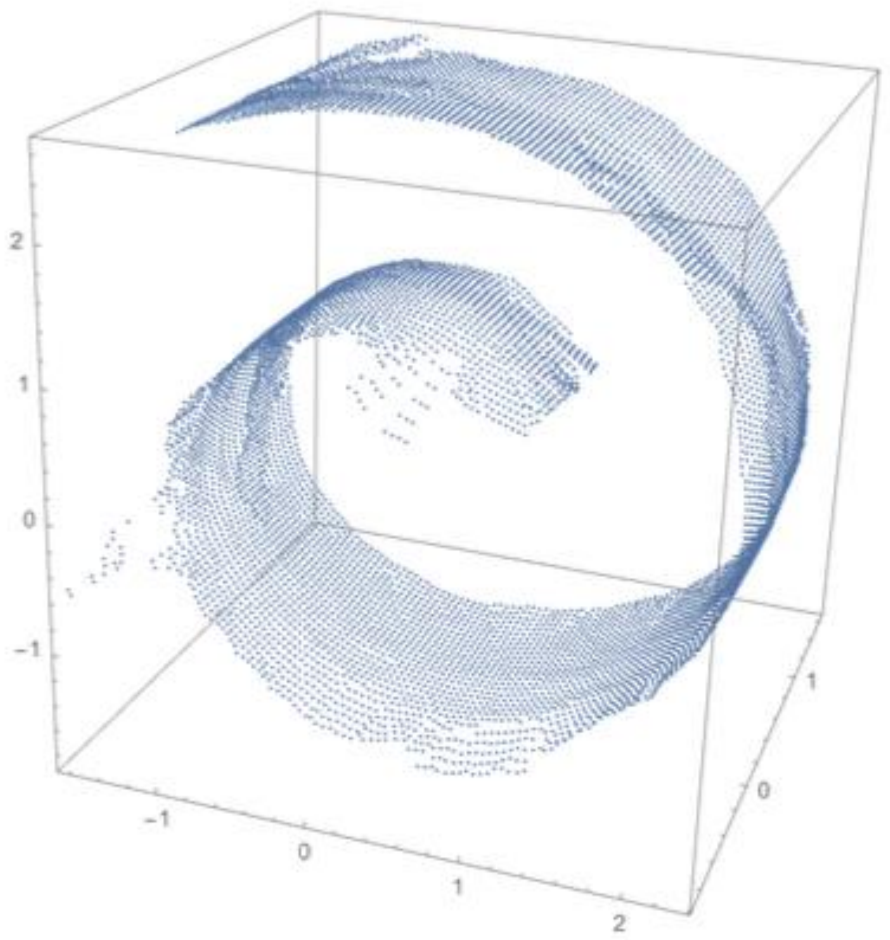
What kind of structures can you see?



What kind of structures can you see?

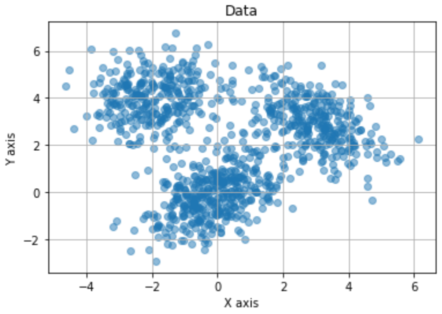


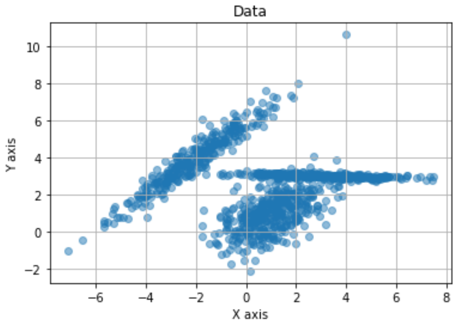
What kind of structures can you see?

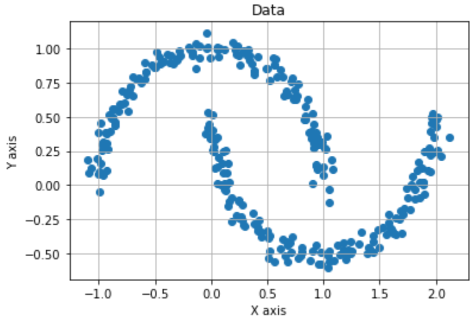


Two broad categories of unsupervised learning (1) Clustering (2) Dimension reduction

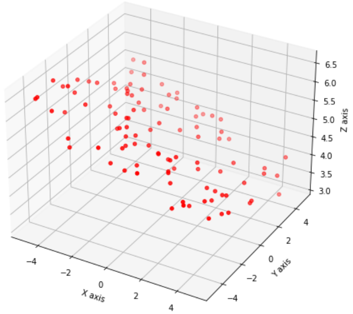
- Clustering:
  - finding a partition of the data that makes sense.

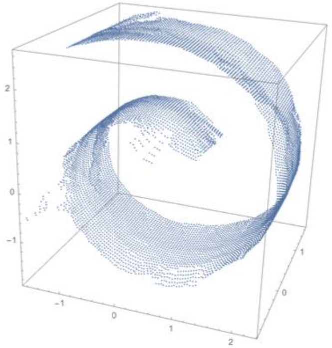




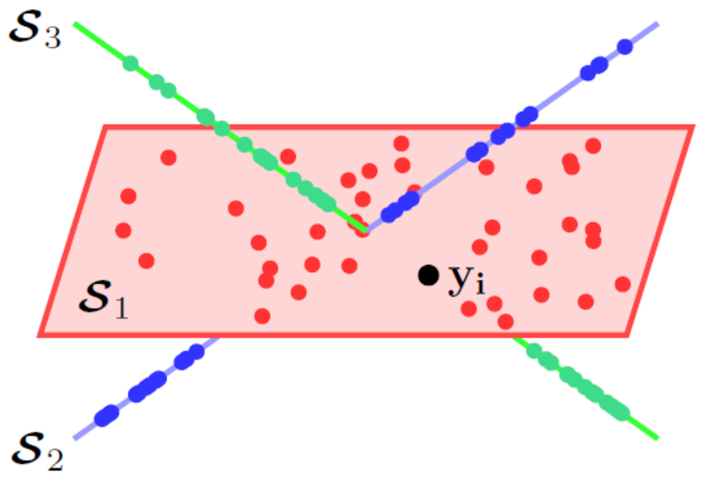


- Dimension reduction:
  - identifying a more compact representation (low-dimension) of data



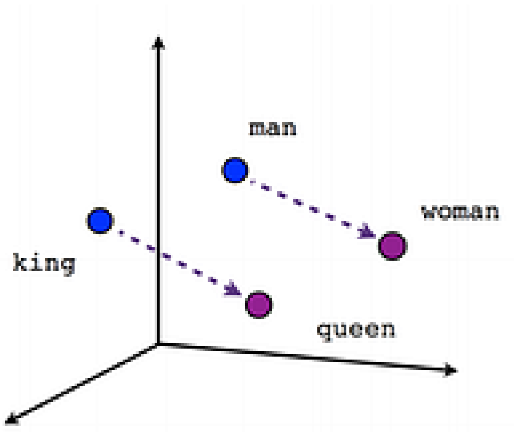


Application: Motion segmentation and subspace clustering

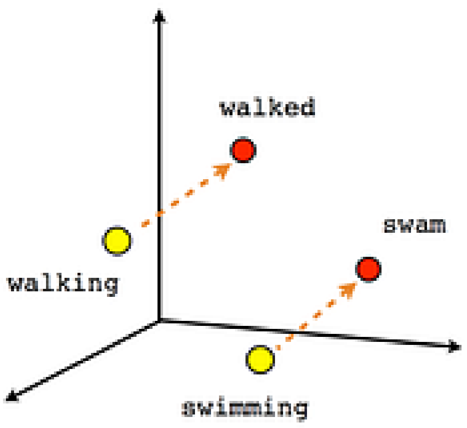


Applications: learn useful vector space representation of language

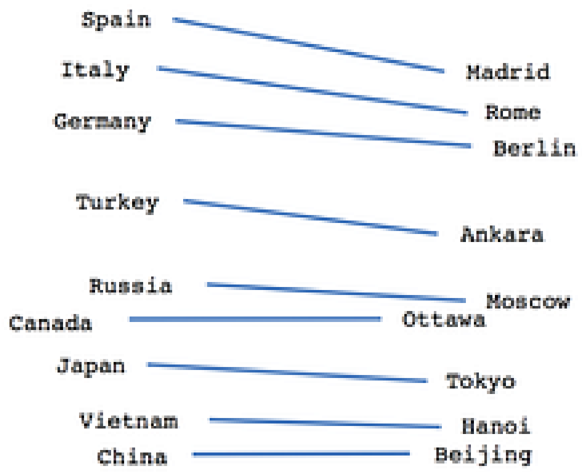
- So you can do algebra on them..



Male-Female



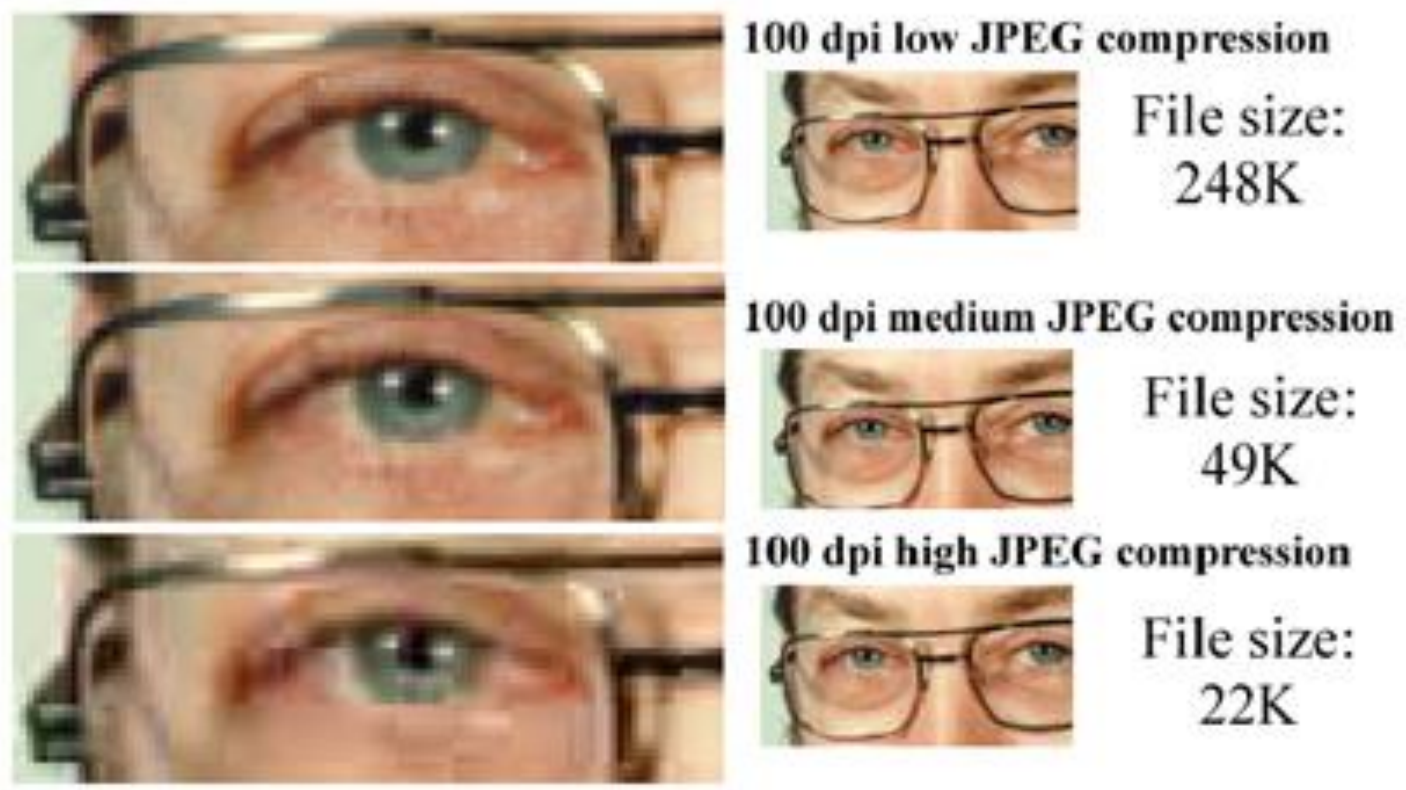
Verb tense



Country-Capital

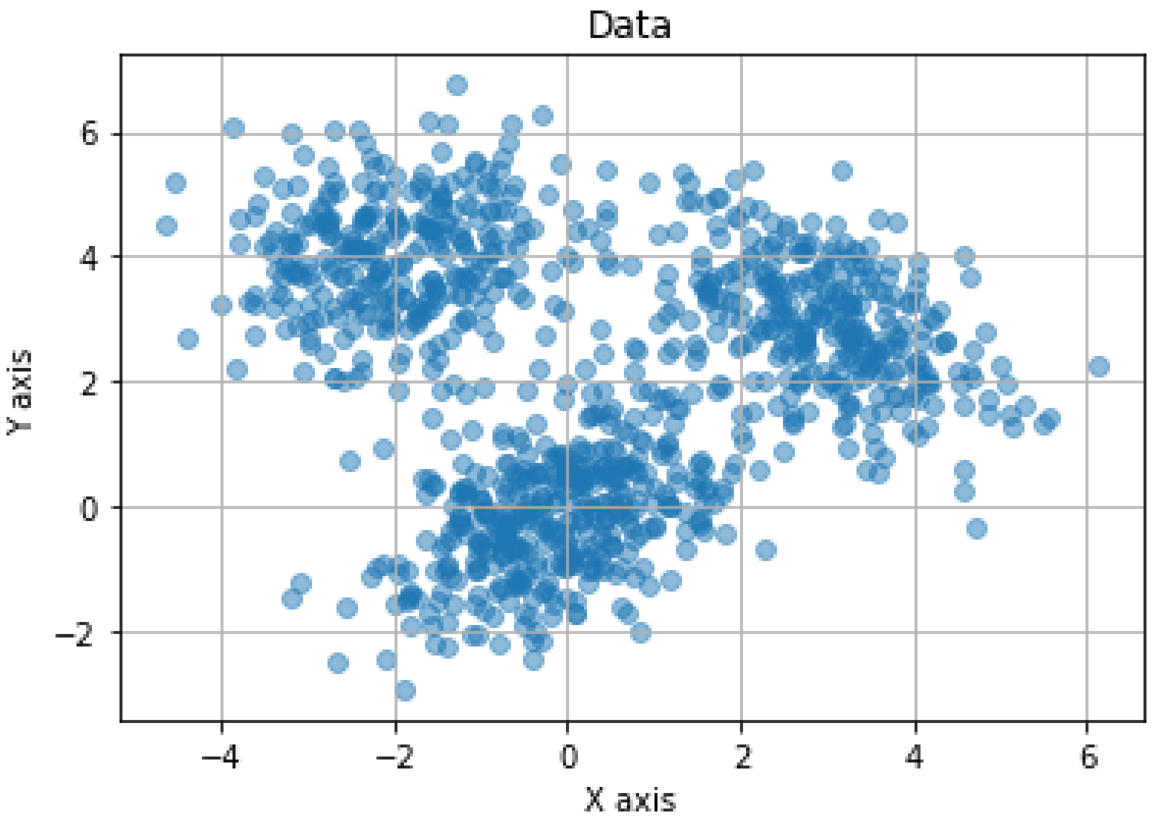
Source: https://towardsdatascience.com/creating-word-embeddings-coding-the-word2vec-algorithm-in-python-using-deep-learning-b337d0ba17a8

Application: Image / video compression



Source: http://preservationtutorial.library.cornell.edu/intro/intro-07.html

How do you learn the structure you see?



- Come up with a loss function to minimize?
- Come up a probabilistic model that generates the data?

The problem of k-means clustering

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

- Where $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ is a partition of the datase $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$,
- And $\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x},$ is called a cluster center (centroid) of S_i

The above optimization problem is equivalent to the following loss minimization

$$\min_{\mu_1, \ldots, \mu_k \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \min_{j \in [k]} \|x_i - \mu_j\|^2$$

- Once we find the **centroids**, finding the partition of the data is easy.
- If we have the partition, finding the corresponding centroids is also easy.
- **Idea:** Alternating minimizing the centroids and cluster assignments.

K-means clustering with Lloyd's algorithm

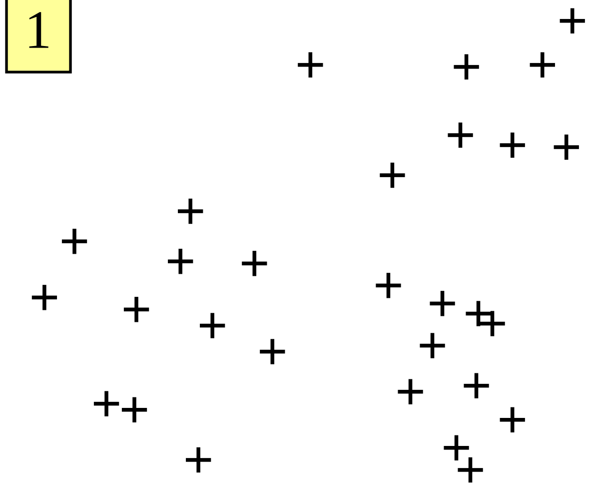
K is a hyperparameter

Algorithm KMeans(D, K) – K-means clustering using Euclidean distance Dis_2.

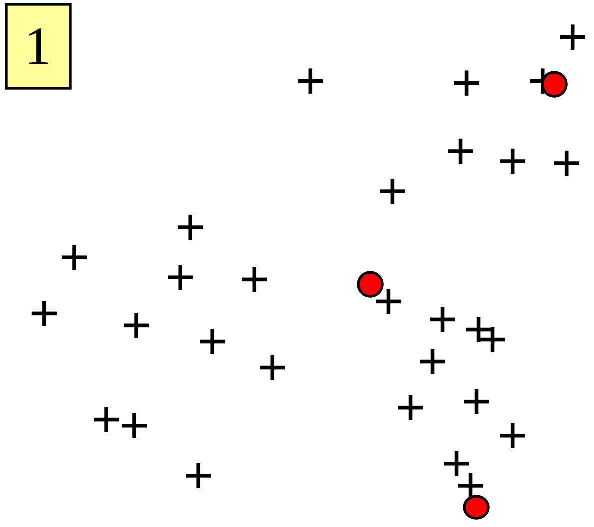
Input : data $D \subseteq \mathbb{R}^d$; number of clusters $K \in \mathbb{N}$.
Output : K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$.

randomly initialise K vectors $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$;
repeat
  assign each $\mathbf{x} \in D$ to $\arg\min_j \mathrm{Dis}_2(\mathbf{x}, \boldsymbol{\mu}_j)$; ⟵ *1-Nearest neighbor assignment*
  for $j = 1$ to K **do**
    $D_j \leftarrow \{\mathbf{x} \in D | \mathbf{x}$ assigned to cluster $j\}$; ⟵ *Partition defined by assignment*
    $\boldsymbol{\mu}_j = \frac{1}{|D_j|} \sum_{\mathbf{x} \in D_j} \mathbf{x}$; ⟵ *Re-compute the cluster mean*
  end
until no change in $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$;
return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$;

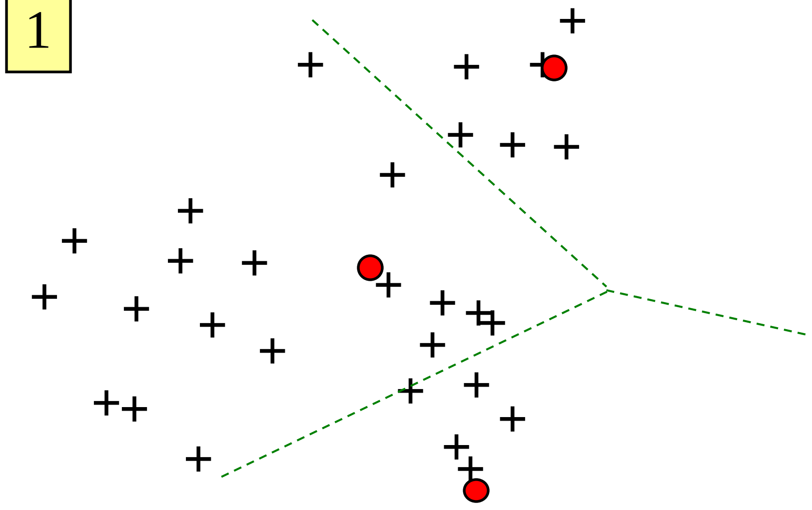
1

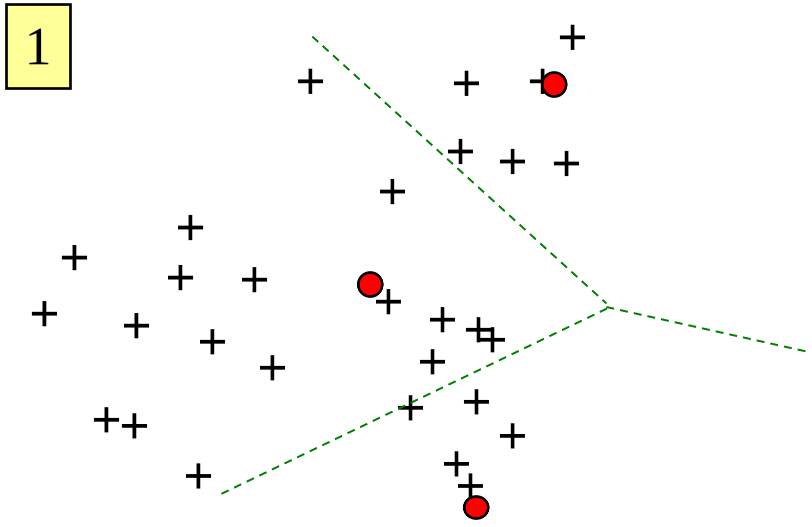


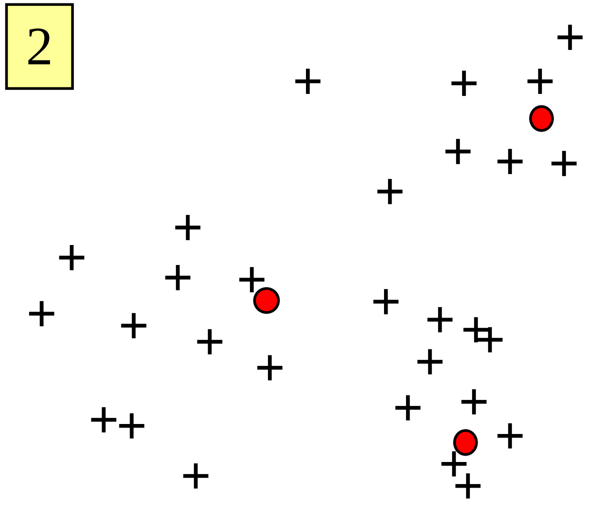
1

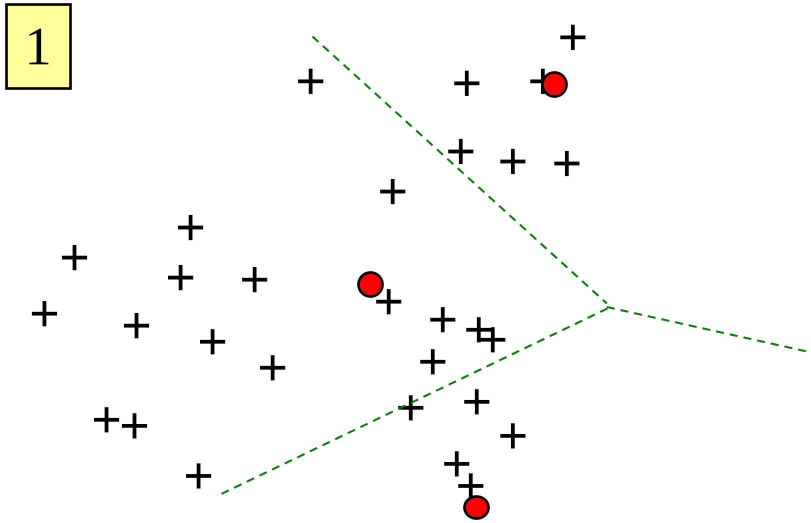


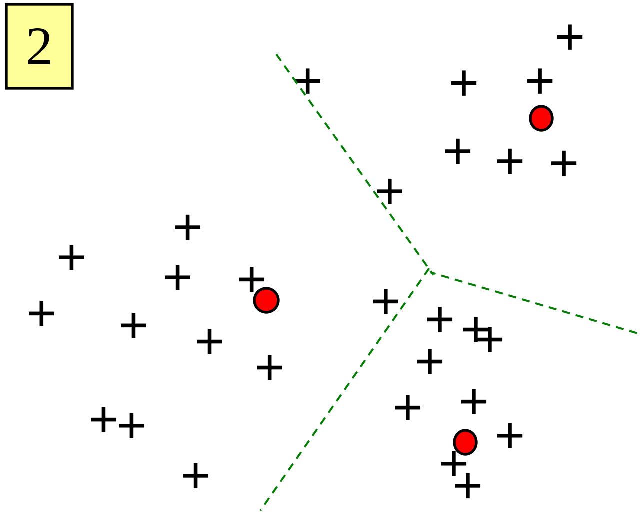
1

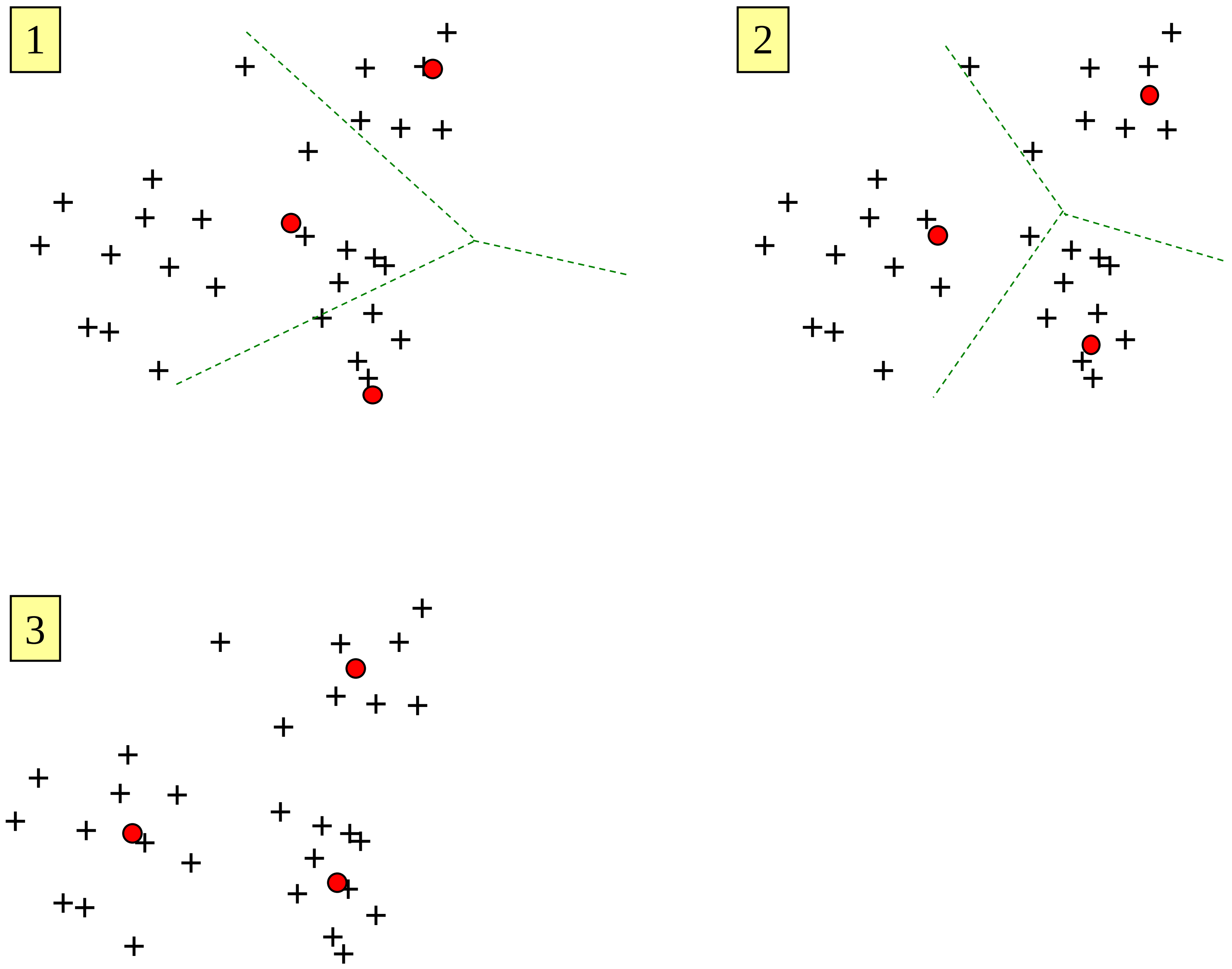


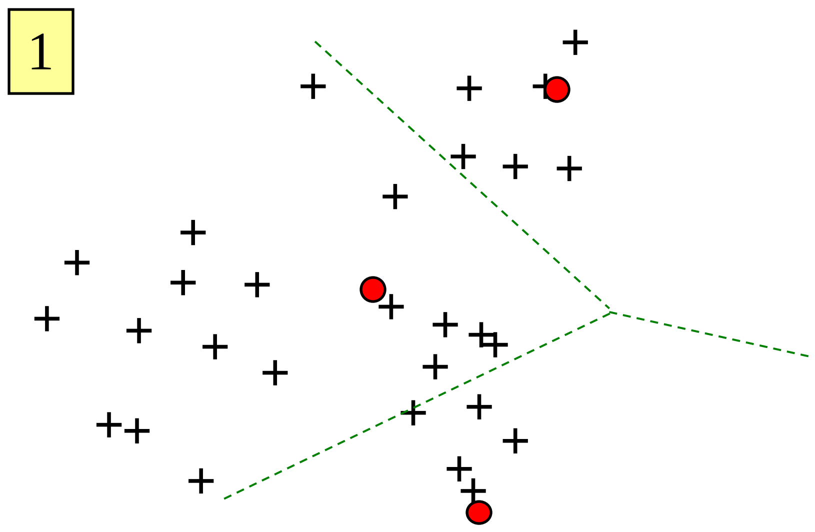

1

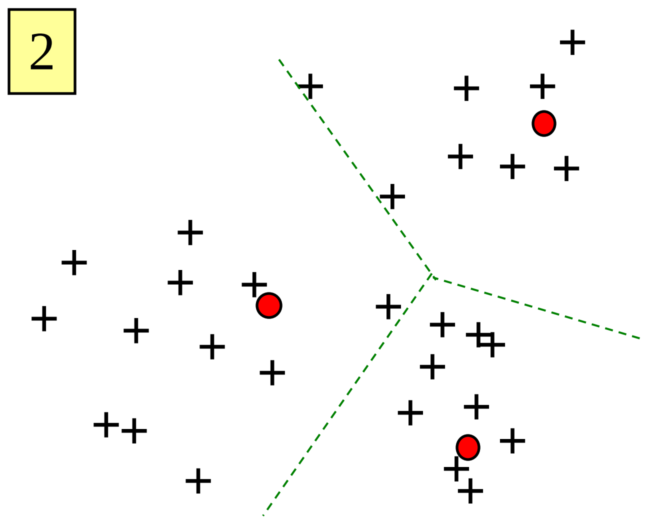

2

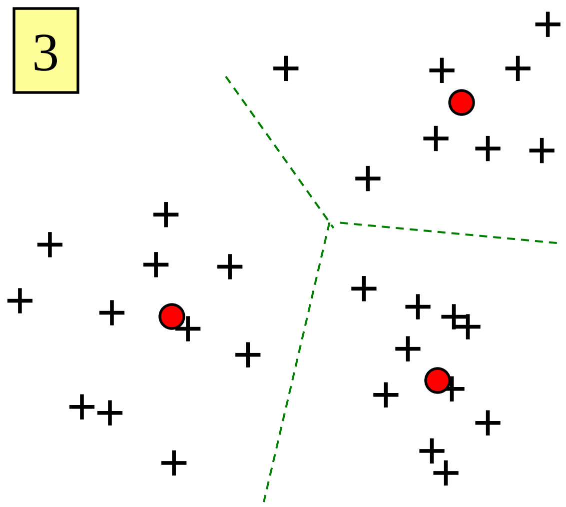

1

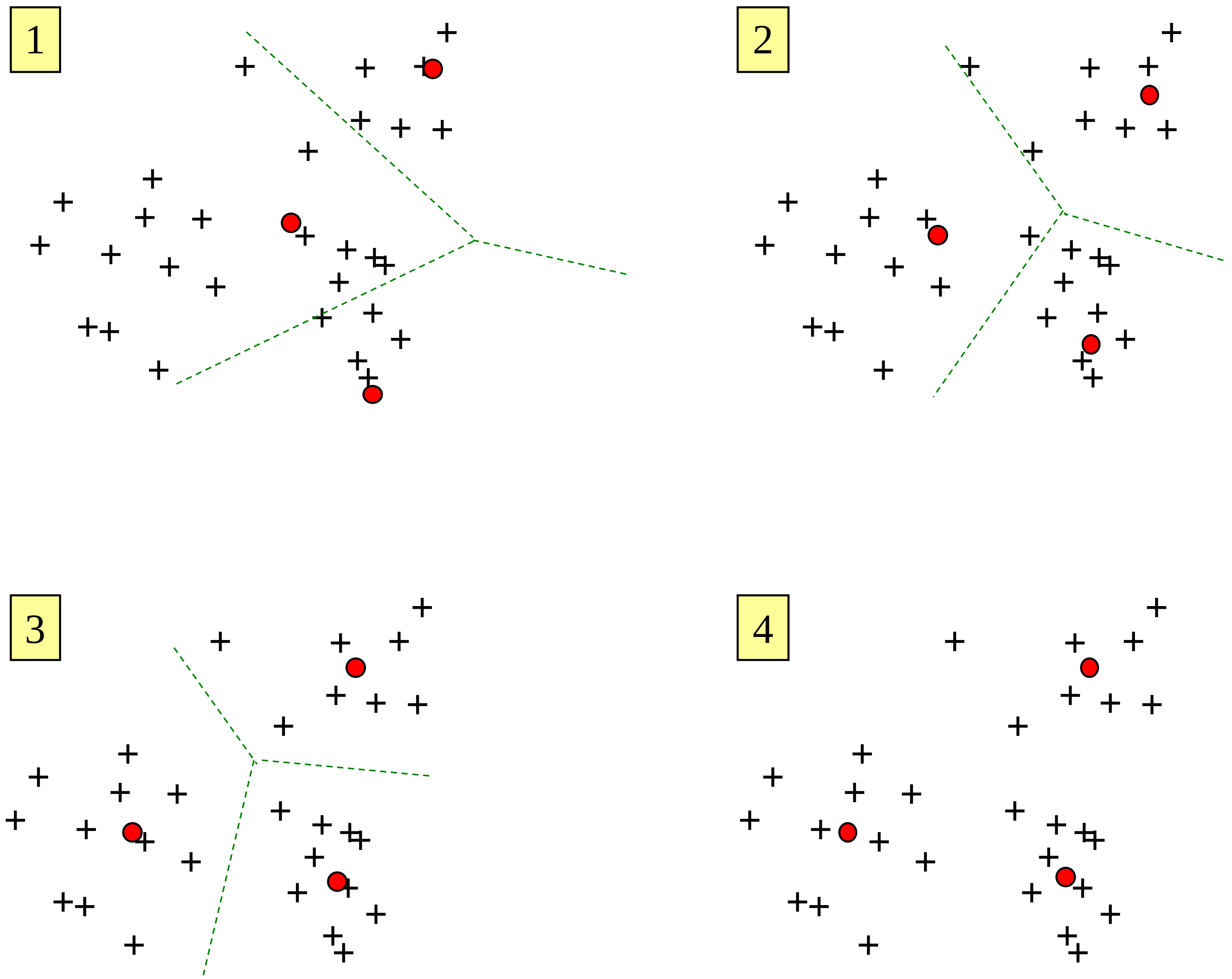

2

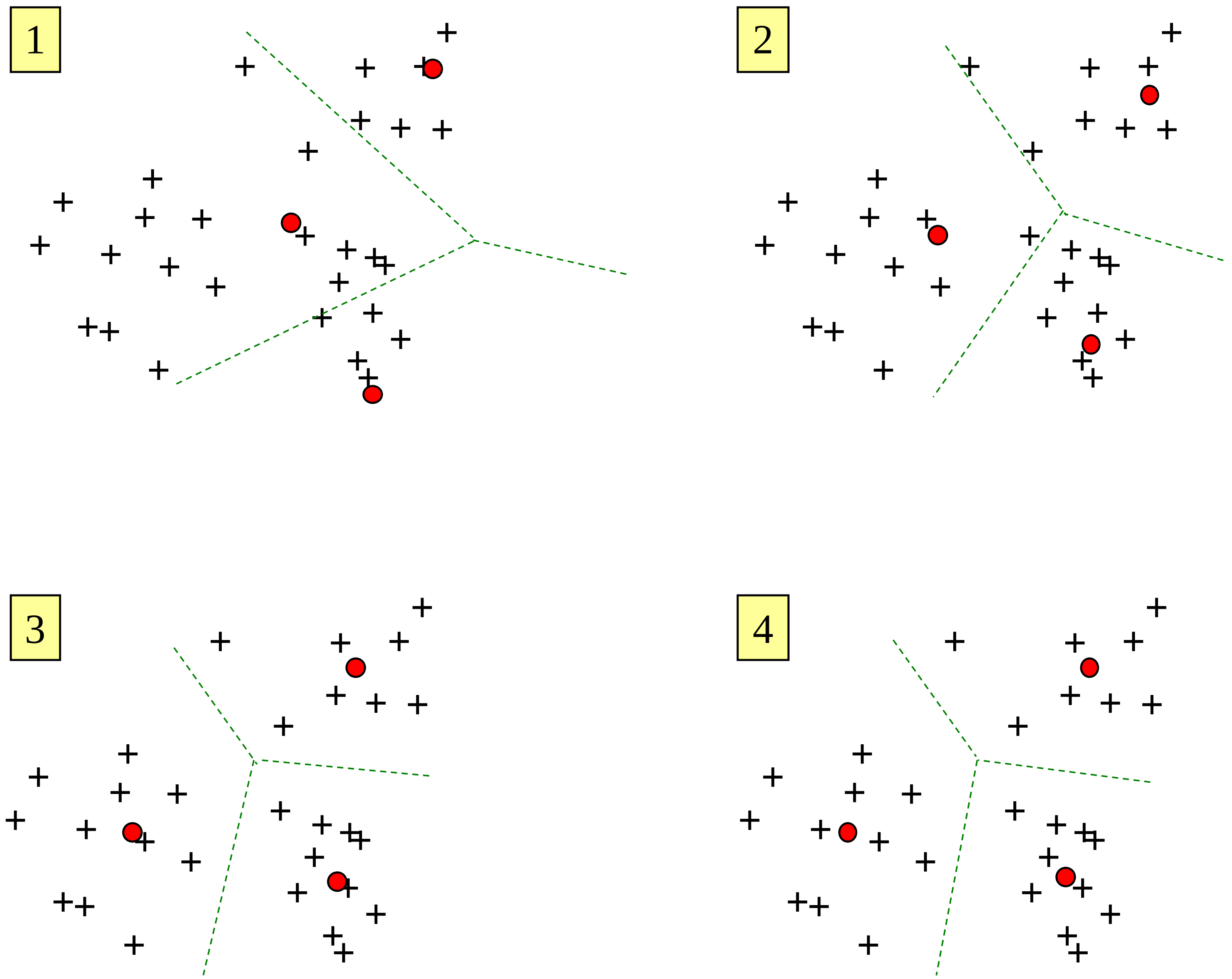

1
2
3

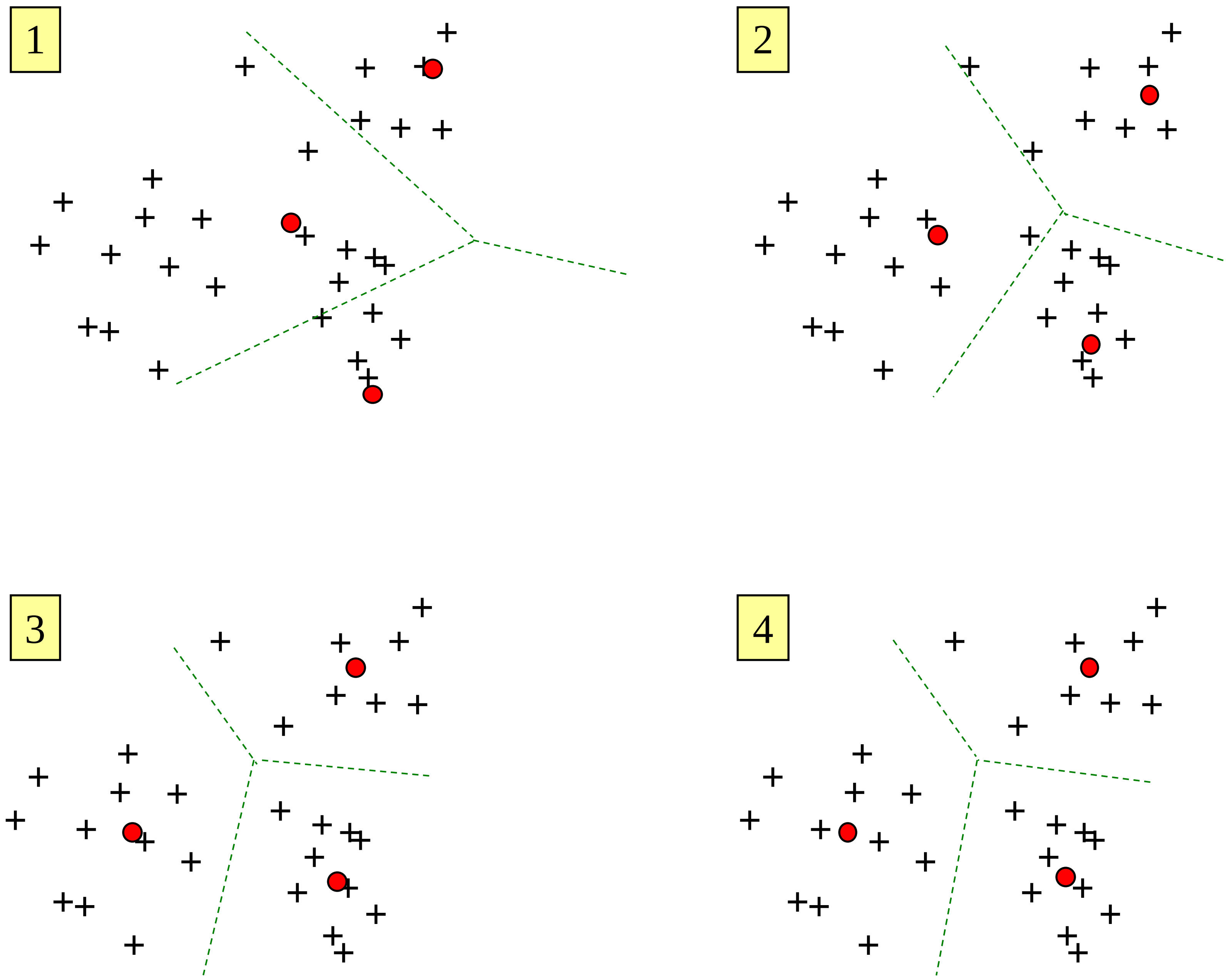

1


2


3

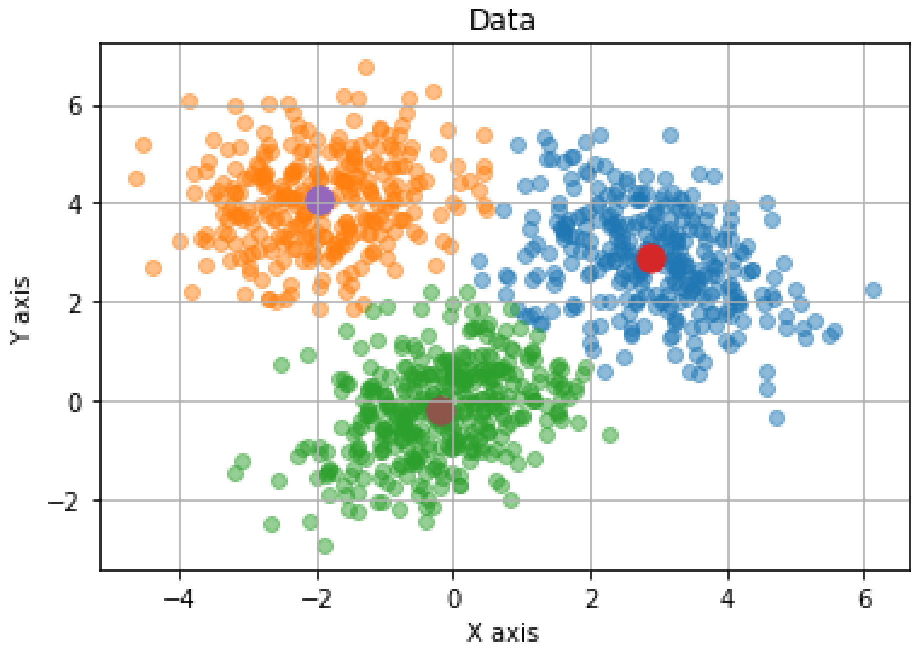

1
2
3
4

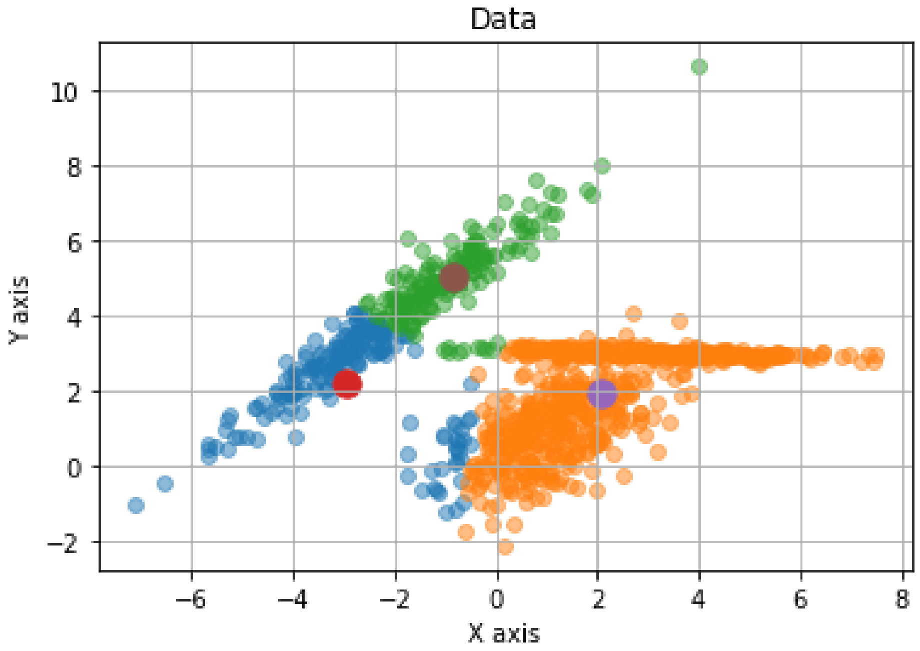

1
2
3
4



No change – finished!

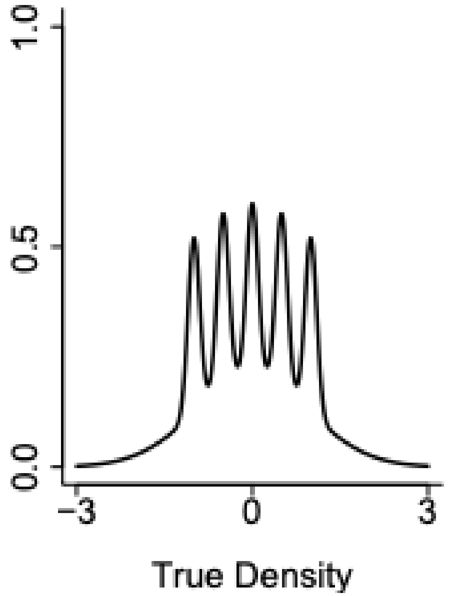
K-means on our previous examples

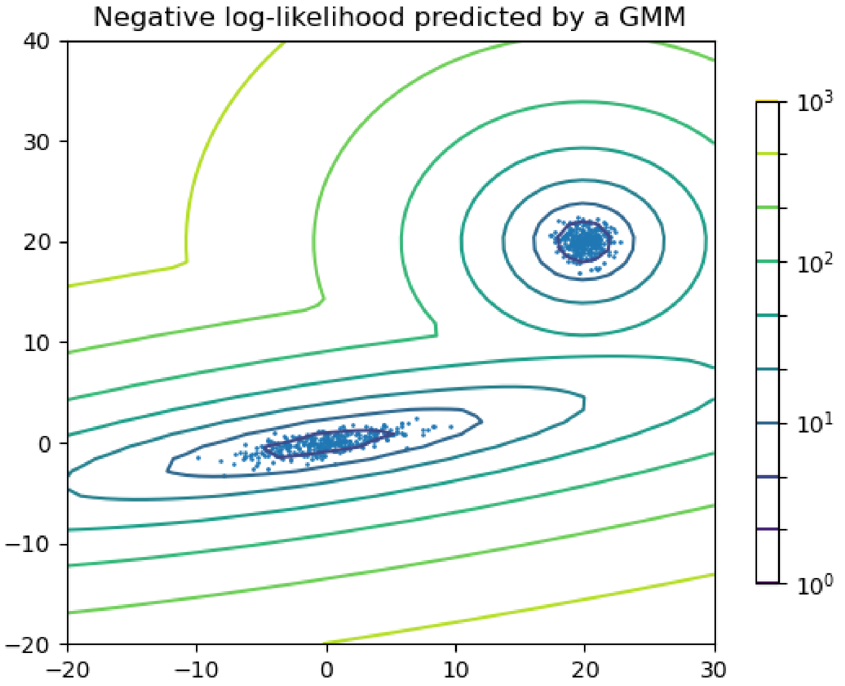




Gaussian mixture models

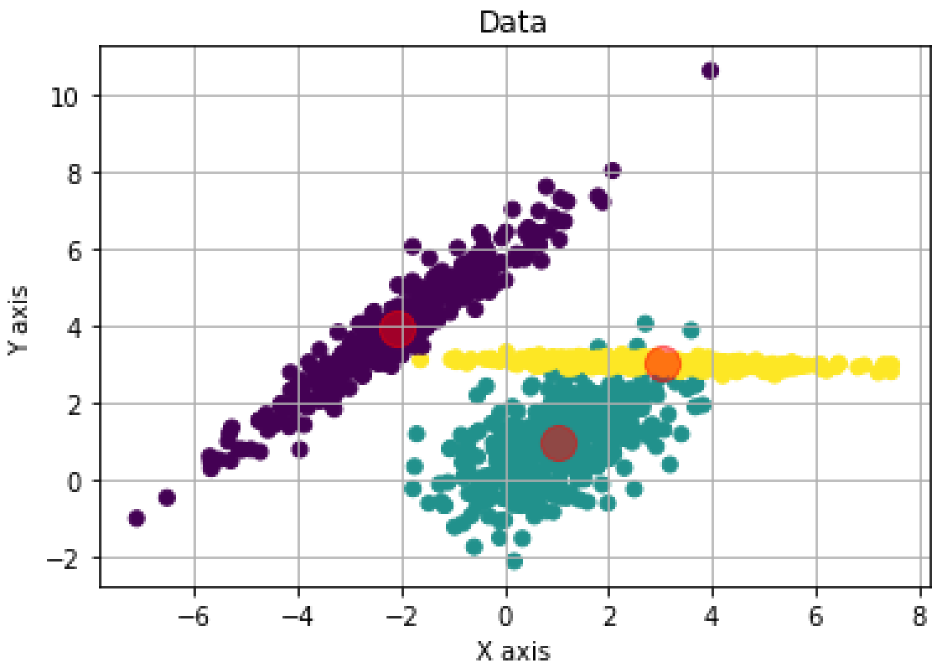
- Assume the data is generated from a mixture of Gaussian distribution

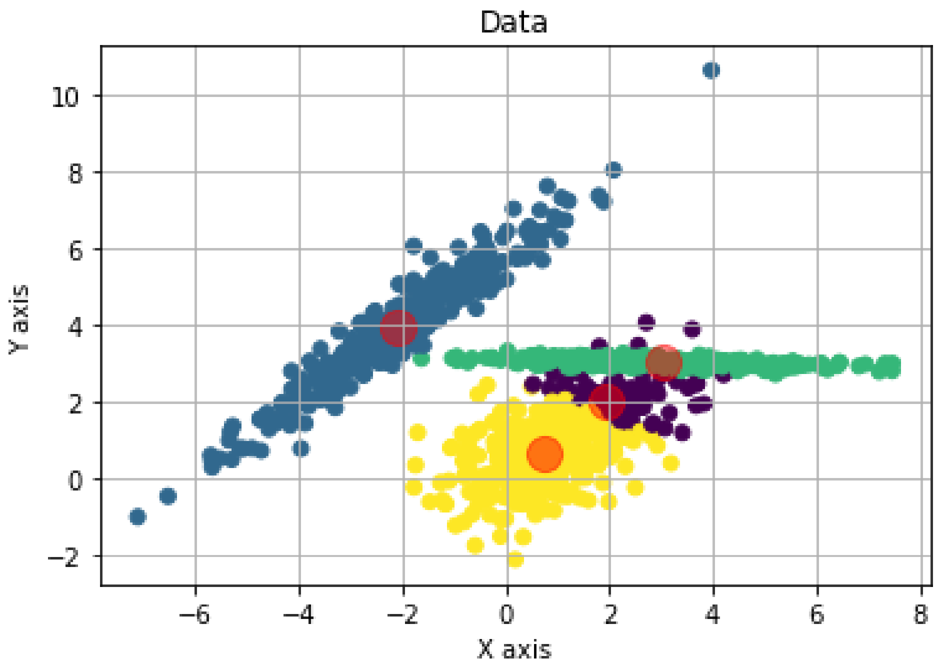




- Data generating process

Fitting mixture of Gaussian model





- Assign **soft** labels to each data point.
- Algorithms for fitting Gaussian mixture models?
  - Expectation-Maximization (not covered in this course… but also alternating making updates)

Summary: Unsupervised Learning

- K-means algorithm
  - Assign hard labels to data points
  - How does it work?
    - Alternating makes updates
    - Which distance function to use?
    - How many cluster centers (centroids) to choose?
    - How to initialize the centroids?

- Gaussian mixture models
  - Assign soft labels to data points
  - A probabilistic model for clustering