CS 383: Machine Learning

Prof Adam Poliak

Fall 2024

12/03/2024

Lecture 29

Announcements – Remaining Assignments

HW07: due Wednesday 11/27

HW08: due Thursday 12/12

Just the Pytorch & Word2Vec notebook

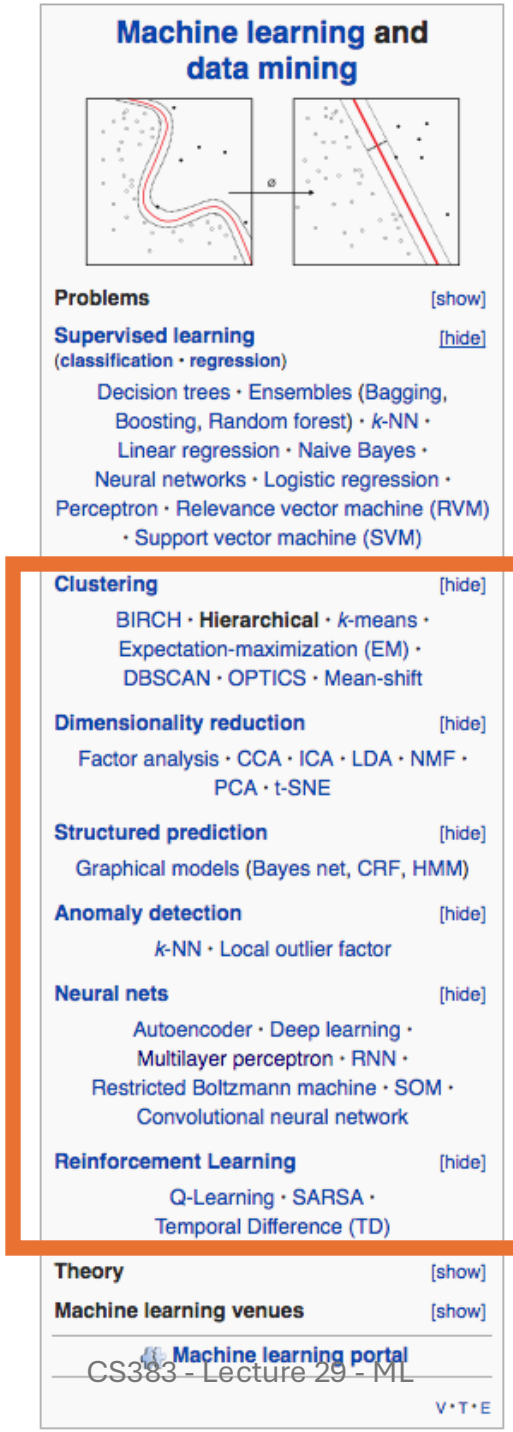
No Project Presentations – due end of finals period

Midterm: Wednesday 12/11

Outline

Unsupervised learning

Supervised Learning:
makes use of examples where we know the underlying "truth" (label/output)



Unsupervised Learning:
Learn underlying structure or features without labeled training data

Image: wikipedia

Unsupervised learning: 3 main areas

1) Clustering: group data points into clusters based on features only

2) Dimensionality reduction: remove feature correlation, compress data, visualize data

3) Structured prediction: model latent variables (example: Hidden Markov Models)

Unsupervised Algorithms

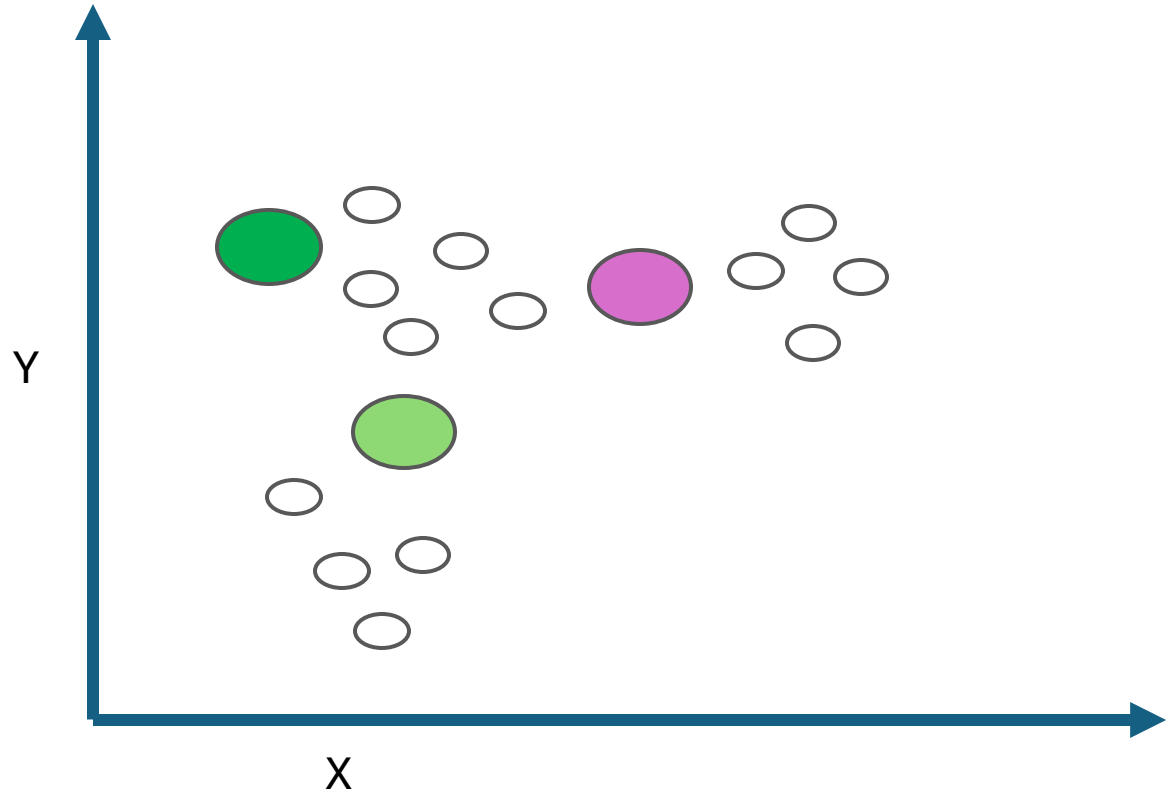
- K-means
- Gaussian Mixture Models (GMM)
- Principle Component Analysis (PCA)

K-means Algorithms

1. Initialize: Randomly pick K points as cluster centers

Randomly pick K points as centers

- Example: 2D point patterns



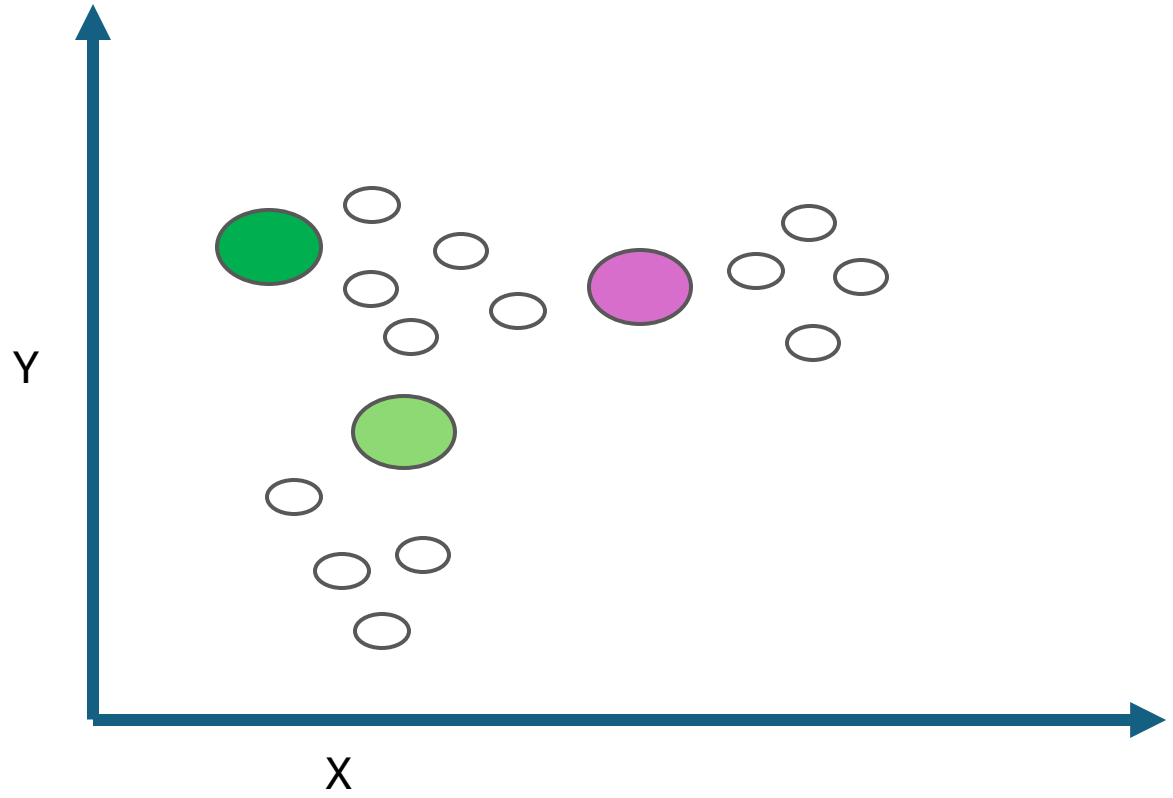
K-means Algorithms

1. Initialize: Randomly pick K points as cluster centers

2. Assign data points to each cluster
    1. Based on distance between point and cluster's center

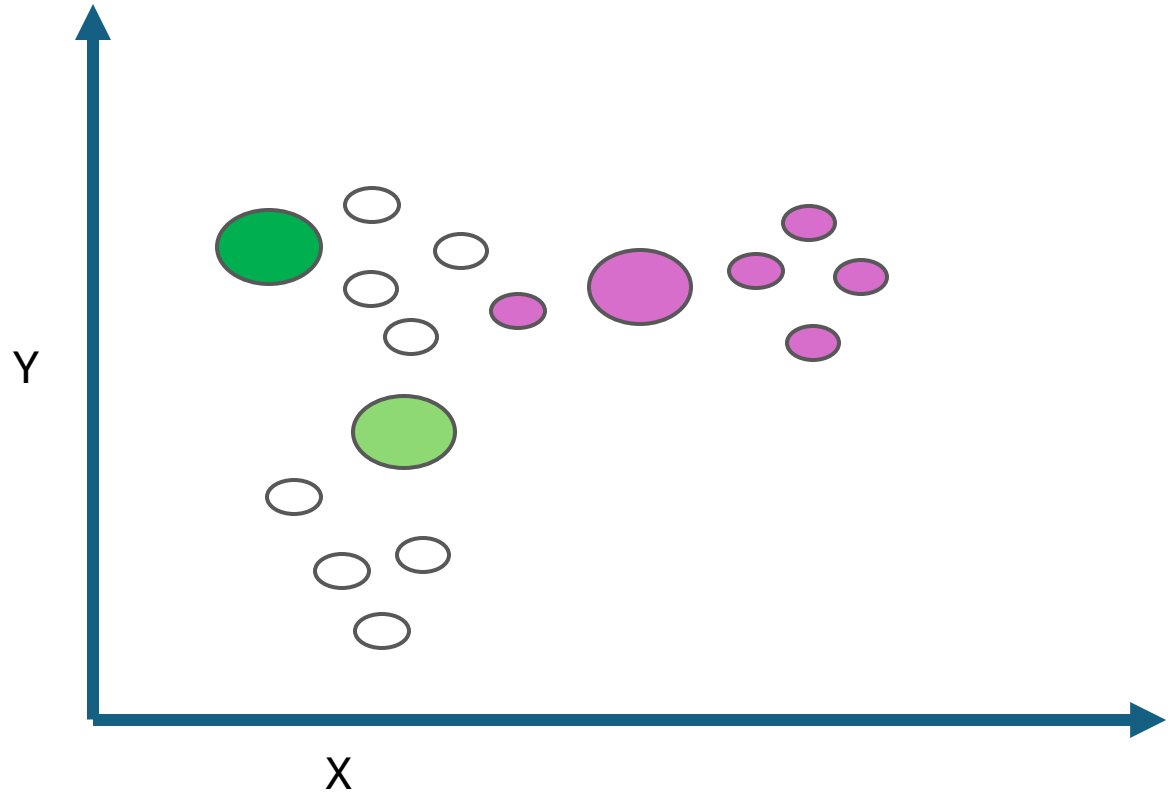
Assign data points to each cluster

- Example: 2D point patterns



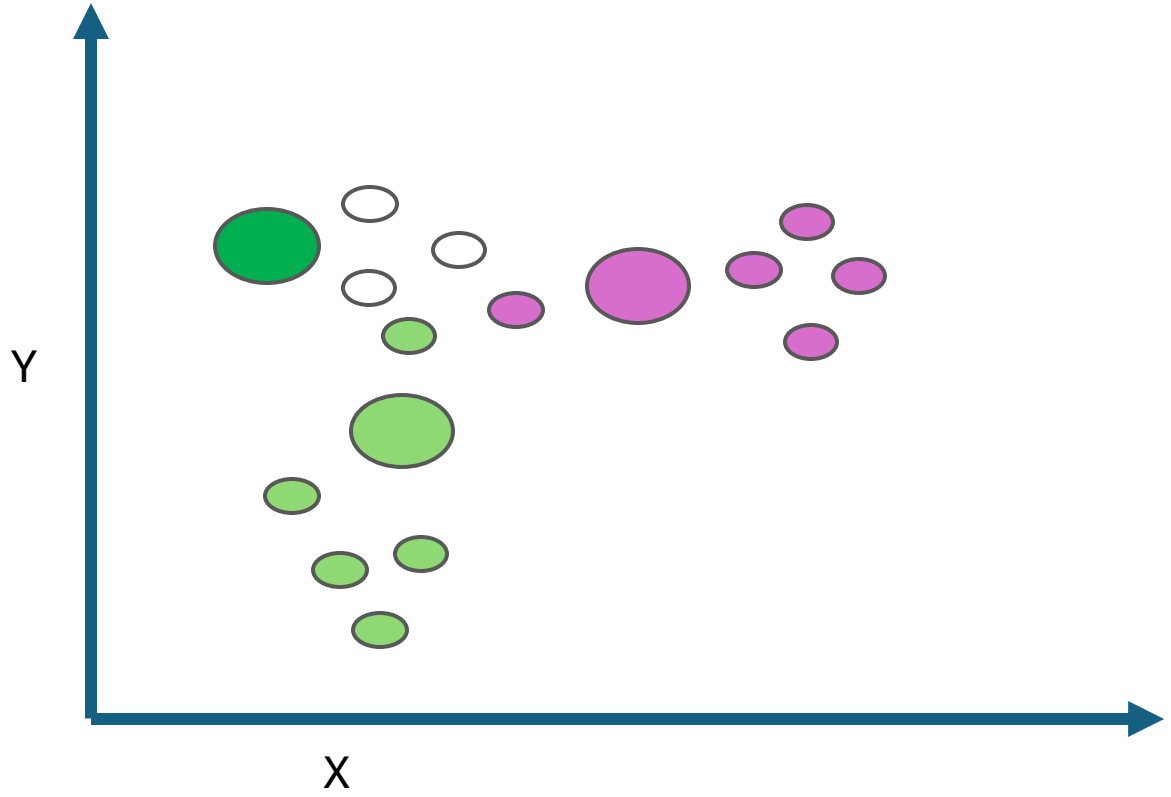
Assign data points to each cluster

- Example: 2D point patterns



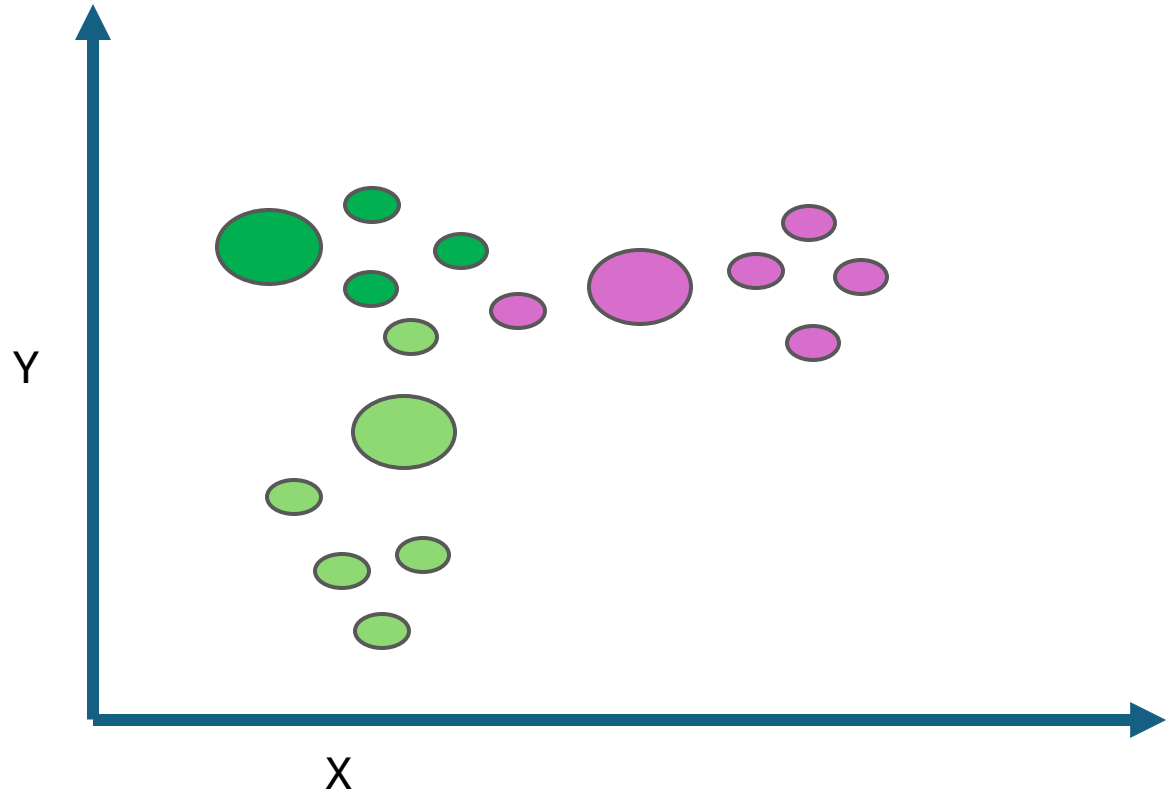
Assign data points to each cluster

- Example: 2D point patterns



Assign data points to each cluster

- Example: 2D point patterns



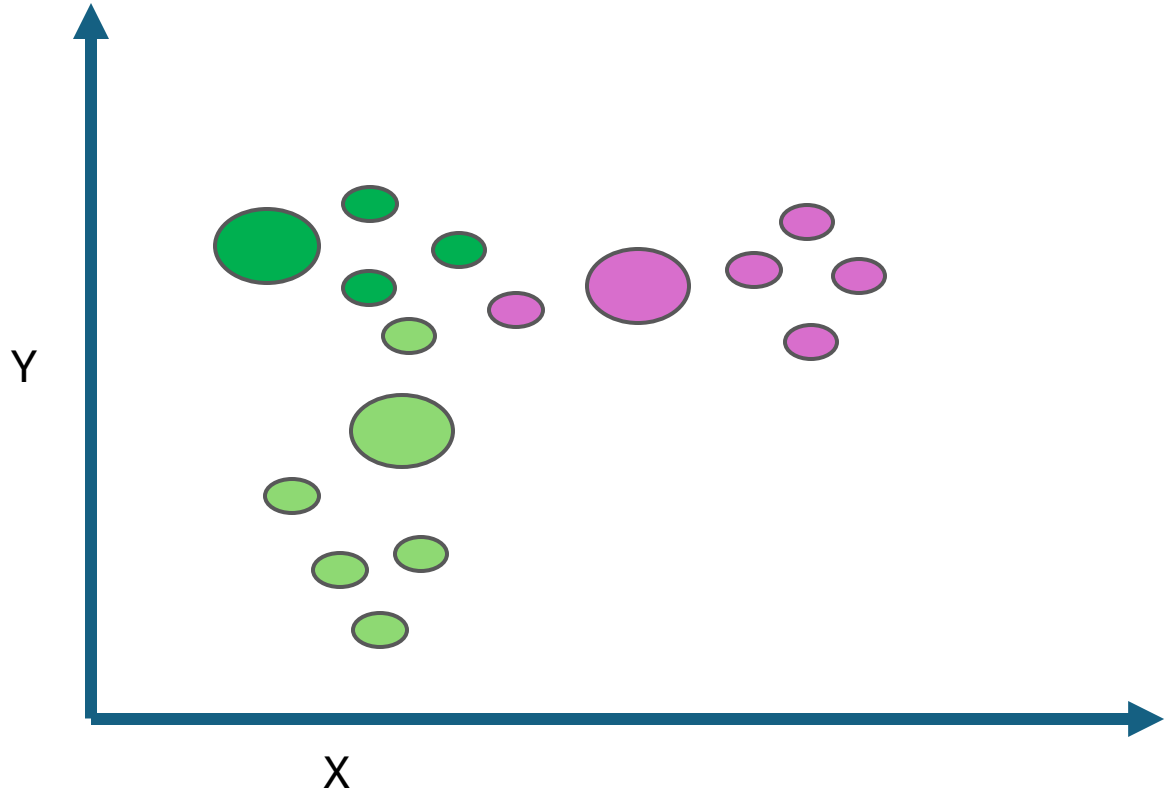
K-means Algorithms

1. Initialize: Randomly pick K points as cluster centers

2. Assign data points to each cluster
    1. Based on distance between point and cluster's center
3. Update the center of each cluster
    1. The average of its assigned points

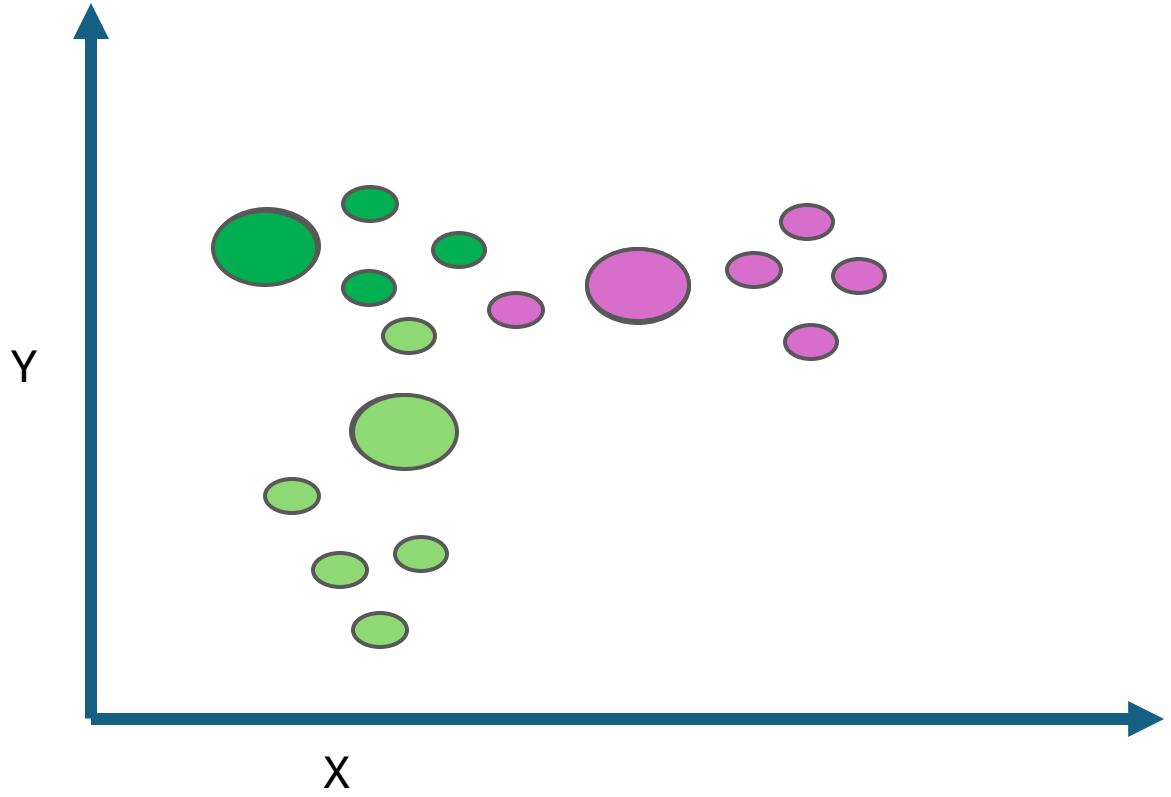
Update Centers

- Example: 2D point patterns



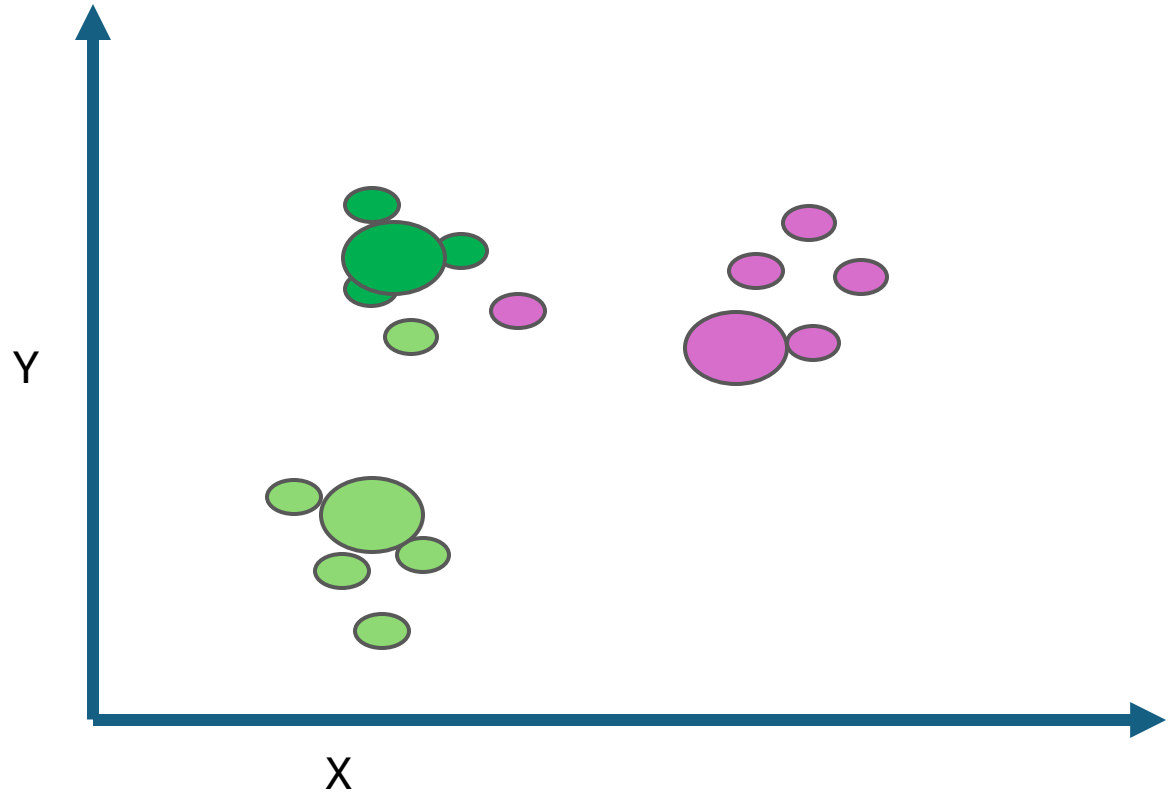
Update Centers

- Example: 2D point patterns



Updated Centers

- Example: 2D point patterns



K-means Algorithms

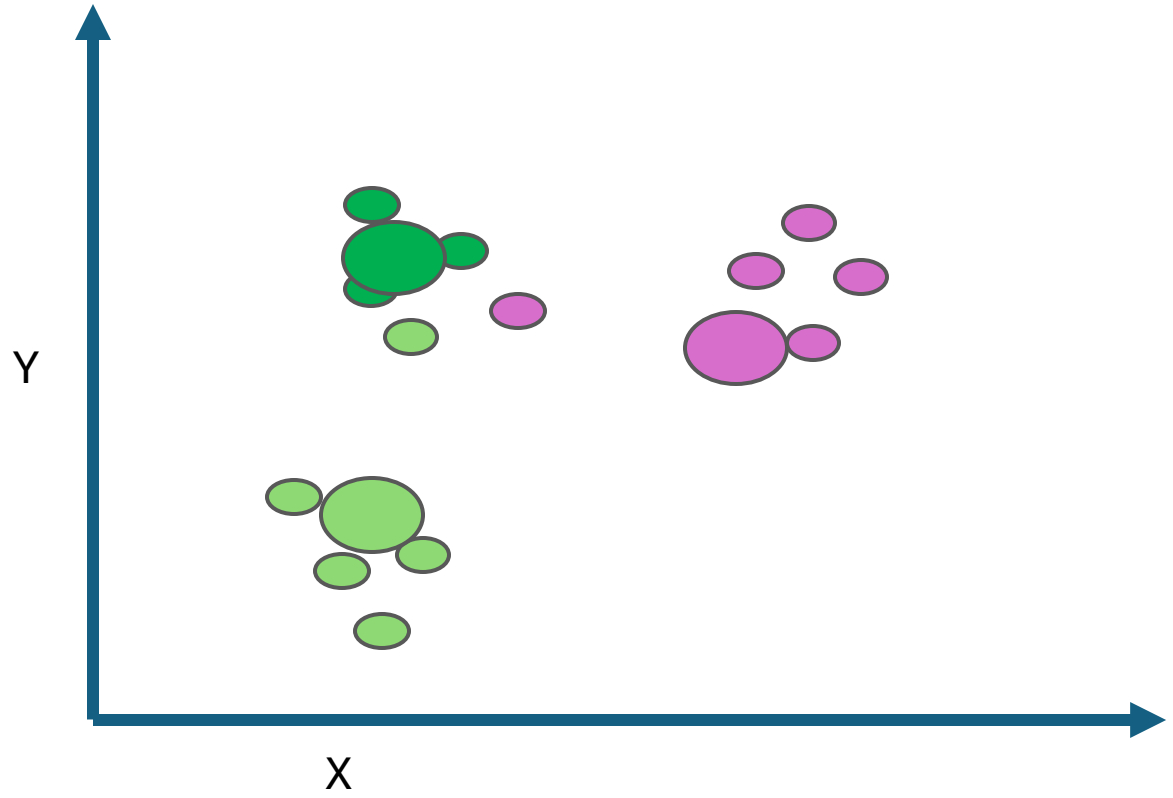
1. Initialize: Randomly pick K points as cluster centers

2. Assign data points to each cluster
    1. Based on distance between point and cluster's center
3. Update the center of each cluster
    1. The average of its assigned points

4. Repeat 2 & 3 until the assignments stop changing

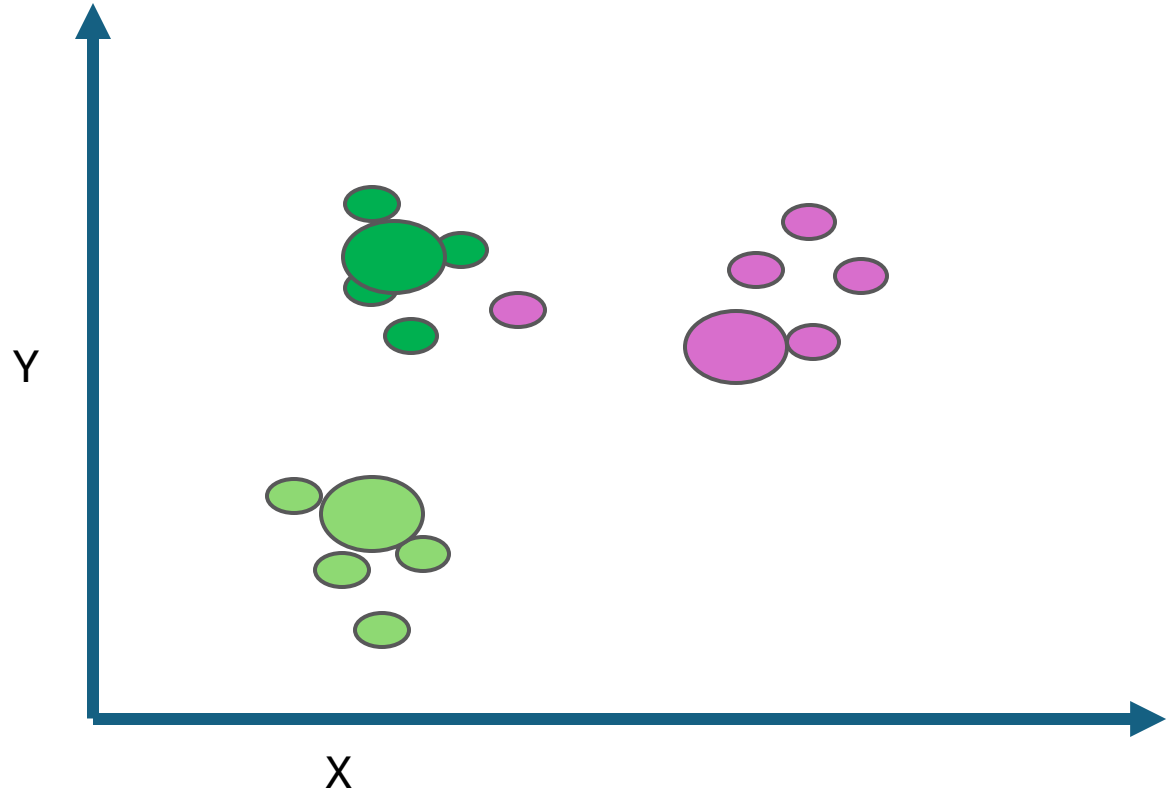
Reassign data points to each cluster

- Example: 2D point patterns



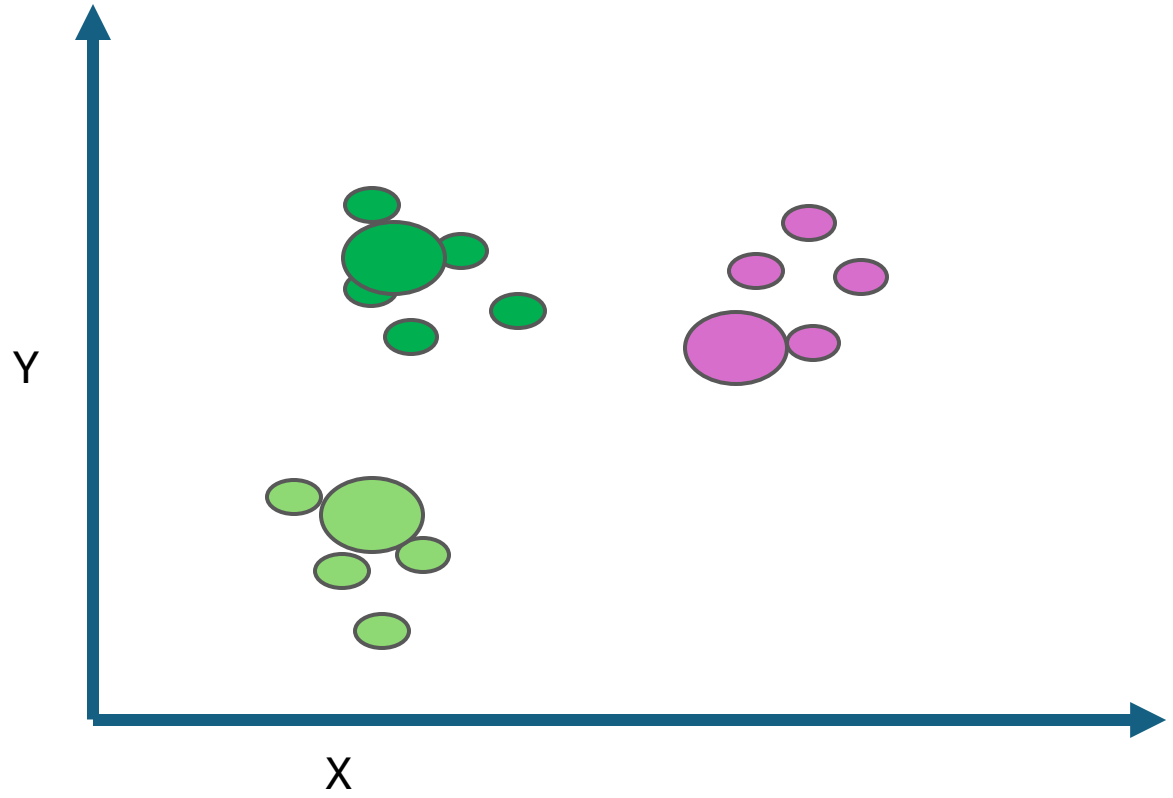
Reassign data points to each cluster

- Example: 2D point patterns



Reassign data points to each cluster

- Example: 2D point patterns



K-means Algorithms

1. Initialize: Randomly pick K points as cluster centers

2. Assign data points to each cluster
    1. Based on distance between point and cluster's center
3. Update the center of each cluster
    1. The average of its assigned points

4. Repeat 2 & 3 until the assignments stop changing

Goal of K-means

Find k clusters e $= \{e_1, e_2, e_3, \ldots, e_k\}$ s.t. we minimize $J(e)$

$$\sum_{k=1}^{K} \sum_{i \in e_k} ||\overrightarrow{x_i} - \overrightarrow{\mu_k}||^2 = J(e)$$

Expectation Maximization Algorithm

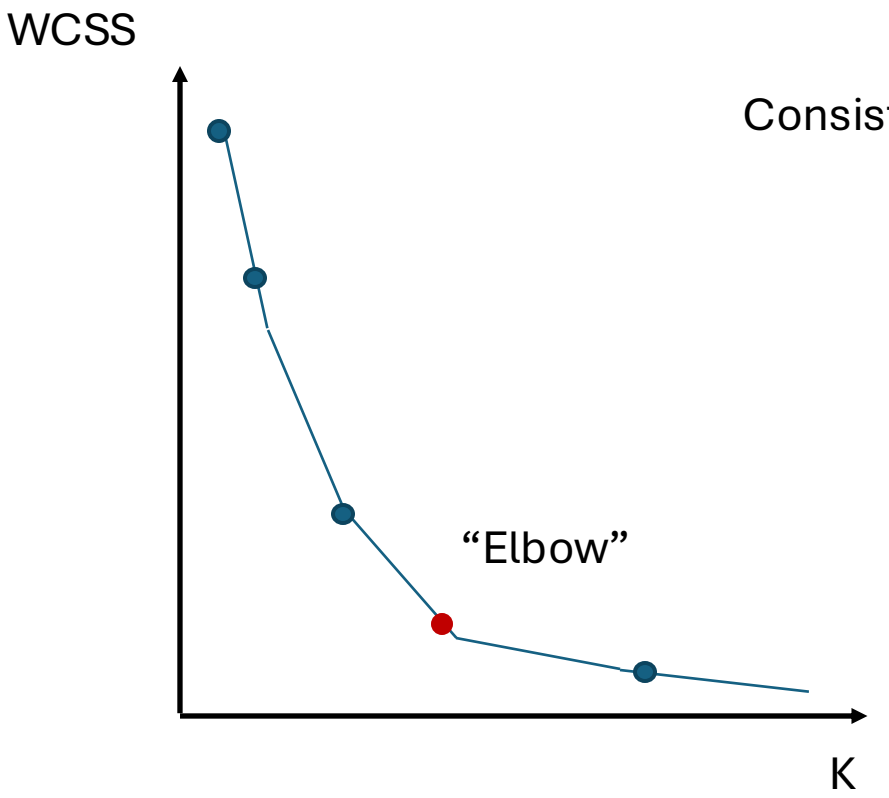
Expectation step:

- Assign members (data points) to each clusters

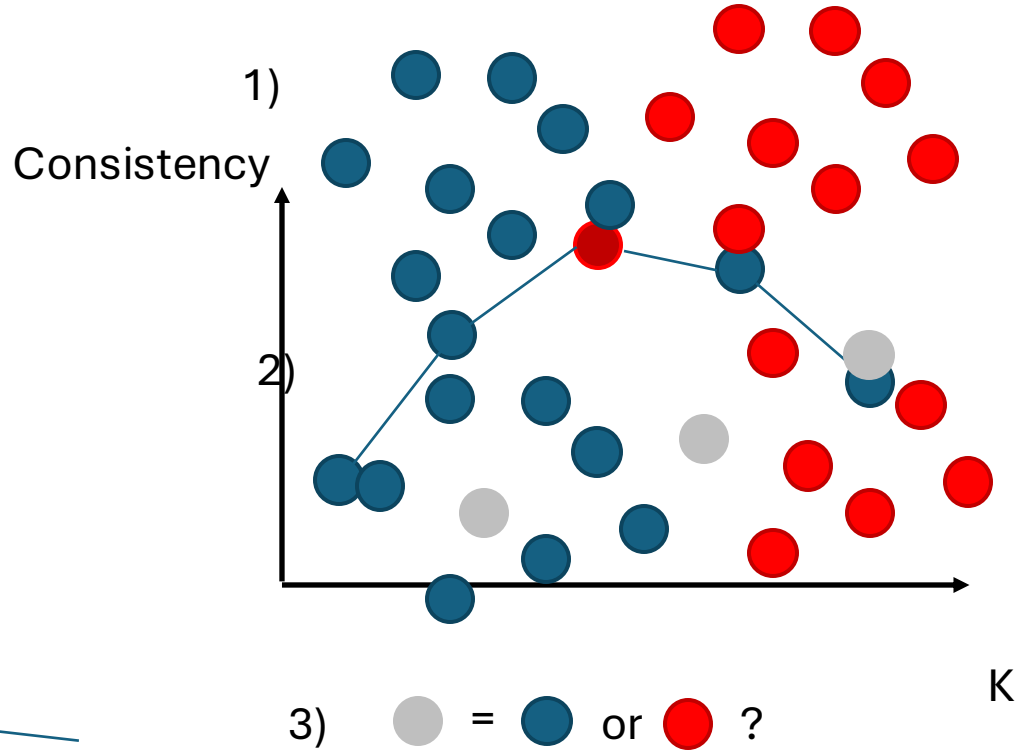
Maximization step:

- Recompute the means as the average of all members in a cluster

Choosing number of clusters



Elbow method



Cross-validation

Many other approaches: silhouette, gap statistics, Calinsky criterion etc…

Issues with k-means

Not generative:

- Cant use it to generate a new datapoint

Lack of uncertainty

- Doesn’t account for sizes of clusters or variance of clusters

Gaussian Mixture Models (GMM)