Indian Institute of Information Technology, Allahabad

CVBL IIIT Allahabad

Introduction to Machine Learning

k-Means Clustering

By

Dr. Shiv Ram Dubey
Associate Professor
Computer Vision and Biometrics Lab
Department of Information Technology
Indian Institute of Information Technology, Allahabad
Web: https://profile.iiita.ac.in/srdubey/

DISCLAIMER

The content (text, image, and graphics) used in this slide are adopted from many sources for academic purposes. Broadly, the sources have been given due credit appropriately. However, there is a chance of missing out some original primary sources. The authors of this material do not claim any copyright of such material.

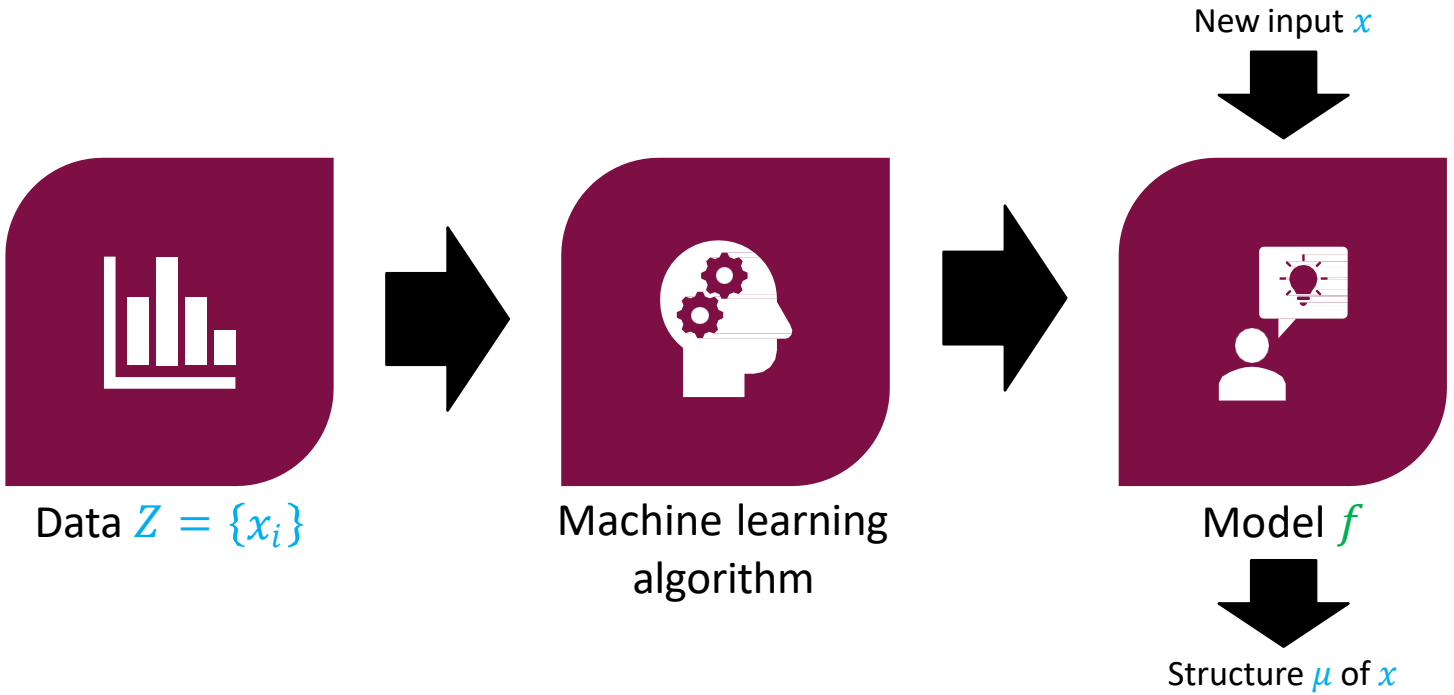
Types of Learning

- **Supervised learning**
  - **Input:** Examples of inputs and desired outputs
  - **Output:** Model that predicts output given a new input

- **Unsupervised learning**
  - **Input:** Examples of some data (no “outputs”)
  - **Output:** Representation of structure in the data

- **Reinforcement learning**
  - **Input:** Sequence of interactions with an environment
  - **Output:** Policy that performs a desired task

Unsupervised Learning



Applications of Unsupervised Learning

- **Visualization**
  - Exploring a dataset, or a machine learning model's outputs
- **Feature Learning**
  - Automatically construct lower-dimensional features
  - Especially useful with a lot of unlabeled data and just a few labeled examples
- **Image Compression**
  - E.g., JPEG is adopting unsupervised machine learning approaches

Applications of Unsupervised Learning

- Visualize the data, find clusters
- Identify interesting supervised learning problems within your dataset
- Generate new data
- Identify important features in the dataset

Loss Minimization Framework

- **To design an unsupervised learning algorithm:**
  - **Model family:** Choose a model family $F = \{f_w\}_w$, where $\mu = f_w(x)$ encodes the structure of x
  - **Loss function:** Choose a loss function $L(w; Z)$

- **Resulting algorithm:**

$$\hat{w}(Z) = \arg\min_w L(w; Z)$$

Types of Unsupervised Learning

- **Clustering**
  - Map samples $x \in \mathbb{R}^d$ to $f(x) \in \mathbb{N}$
  - Each $k \in \mathbb{N}$ is associated with a representative example $x_k \in \mathbb{R}^d$
  - **Examples:** K-means clustering, greedy hierarchical clustering

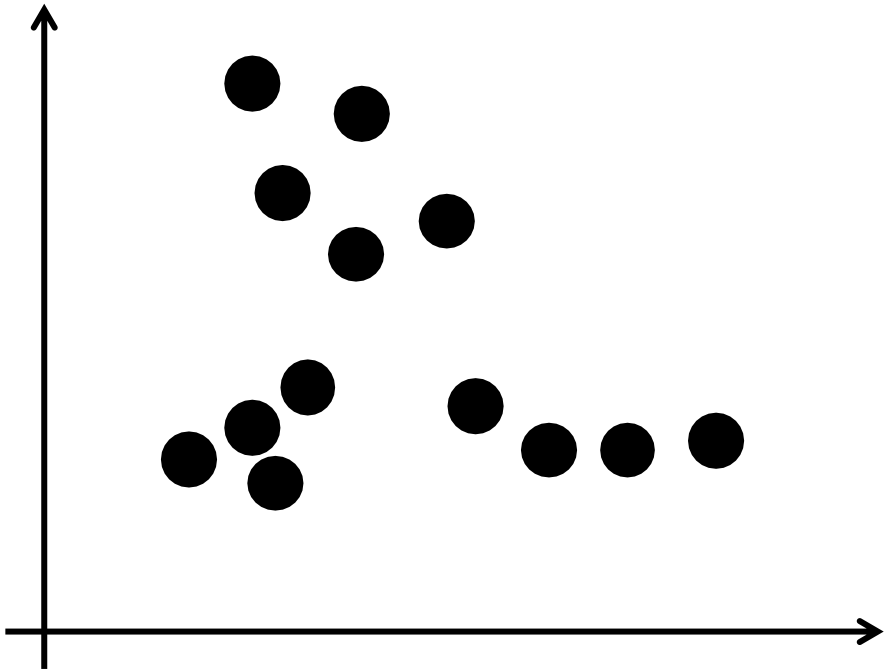
- **Dimensionality reduction**
  - Map samples $x \in \mathbb{R}^d$ to $f(x) \in \mathbb{R}^{d'}$, where $d' \ll d$
  - **Example:** Principal components analysis (PCA)
  - Modern deep learning is based on this idea

The Clustering Problem

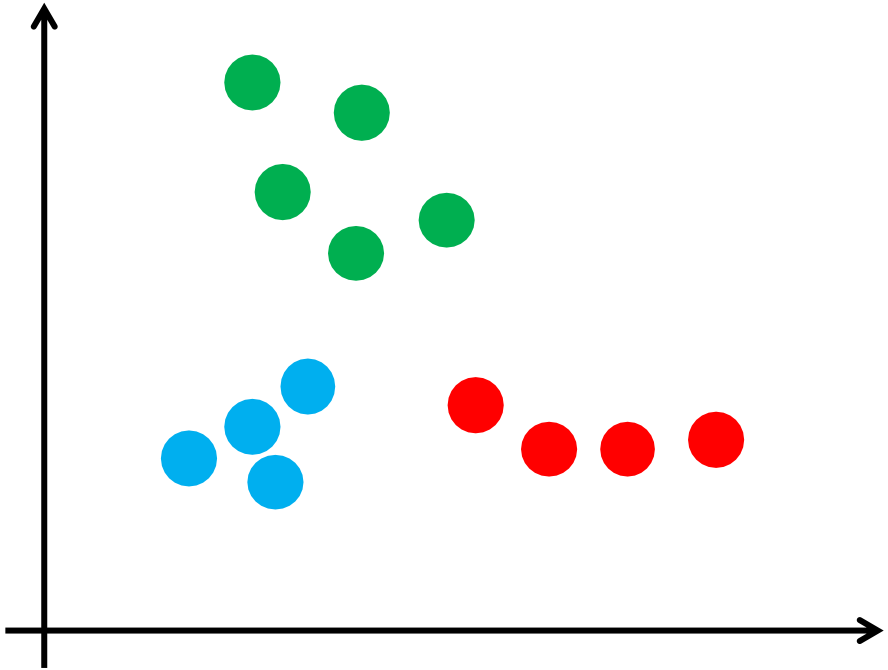
- **Input:** Dataset $Z = \{x_i\}_{i=1}^{n}$

- **Output:** Model $f(x) \in \{1, \dots, K\}$
  - **Intuition:** Predictions should encode "natural" clusters in the data
  - Here, $K \in \mathbb{N}$ is a hyperparameter

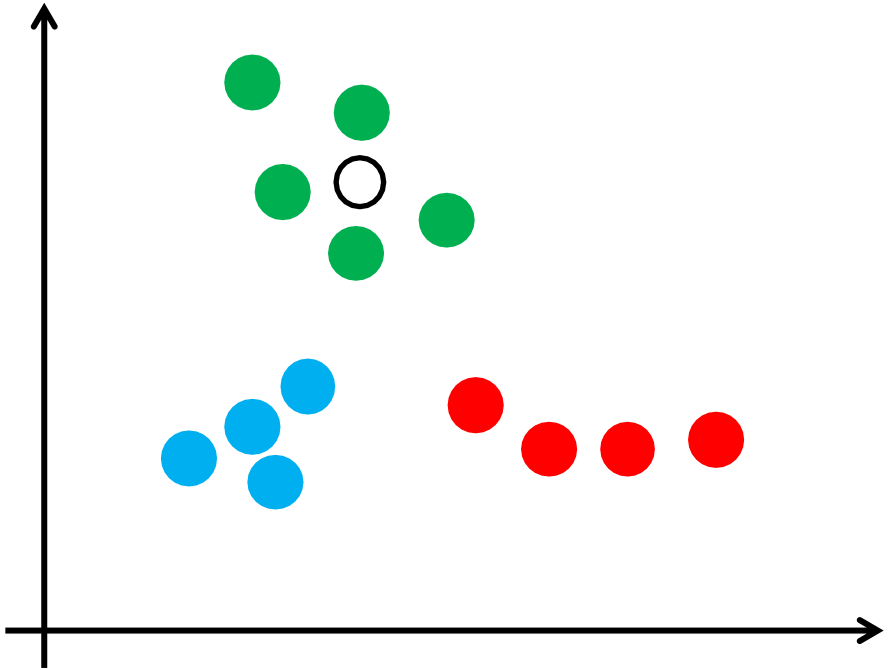
The Clustering Problem



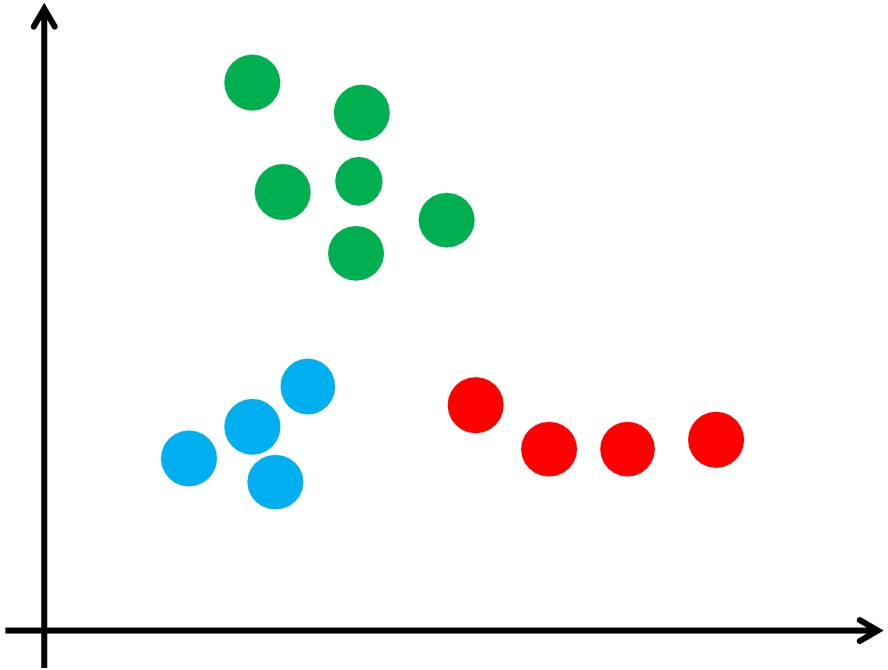
The Clustering Problem



The Clustering Problem

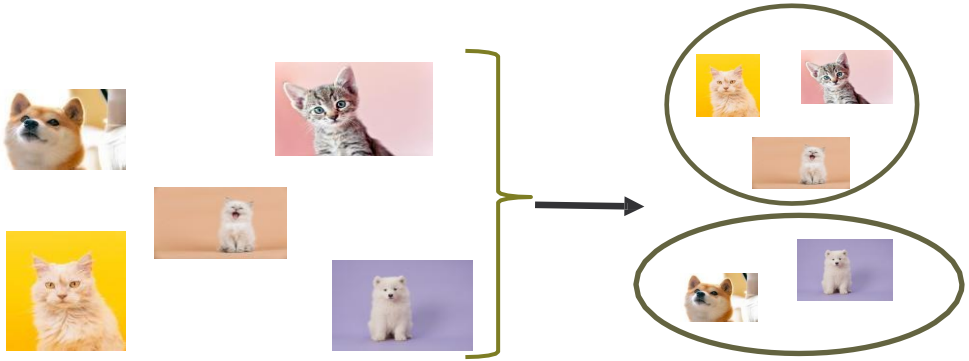


The Clustering Problem



Clustering: Some Examples

- Document/Image/Webpage Clustering
- Image Segmentation (clustering pixels)





- Clustering web-search results
- Clustering (people) nodes in (social) networks/graphs
- .. and many more..

Picture courtesy: http://people.cs.uchicago.edu/~pff/segment/

The Clustering Problem

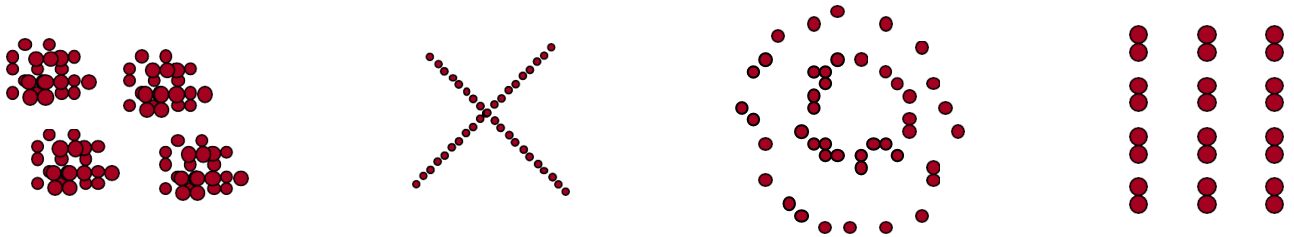
- **Input:** Dataset $Z = \{x_i\}_{i=1}^{n}$

- **Output:** Model $f(x) \in \{1, \dots, K\}$
  - **Intuition:** Predictions should encode "natural" clusters in the data
  - Here, $K \in \mathbb{N}$ is a hyperparameter

- How to formalize "naturalness"?
  - Using a loss function!

Clustering Loss

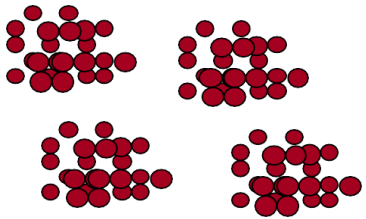
- Loss depends on the structure of the data we are trying to capture



- K-Means clustering aims to minimize specific loss over a specific model family

K-Means Clustering Model Family

- **Parameters:**
- **Centroids** μ_j, for $j \in \{1, \ldots, K\}$
  - One for each cluster (K is a hyperparameter)
  - **Intuition:** μ_j is the "center" of cluster j



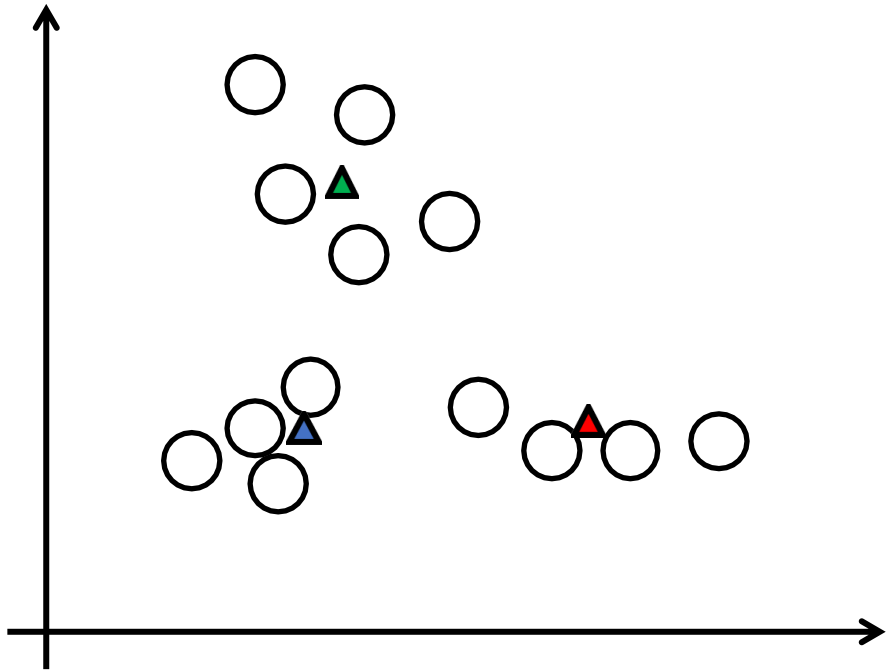
- **Assignment:** assign each data point x it to the nearest cluster:

$$f_\mu(x) = \arg\min_j \left\| x - \mu_j \right\|_2^2$$

  - Can use other distance functions

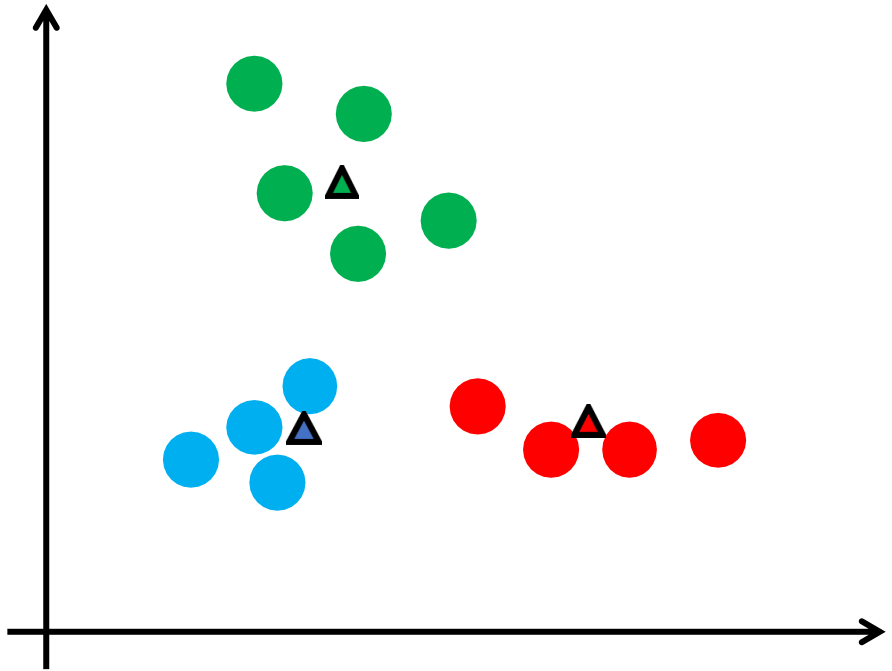
K-Means Clustering Loss

- Compute Squared Error of each point in the training data to its centroid



K-Means Clustering Loss

- Compute Squared Error of each point in the training data to its centroid



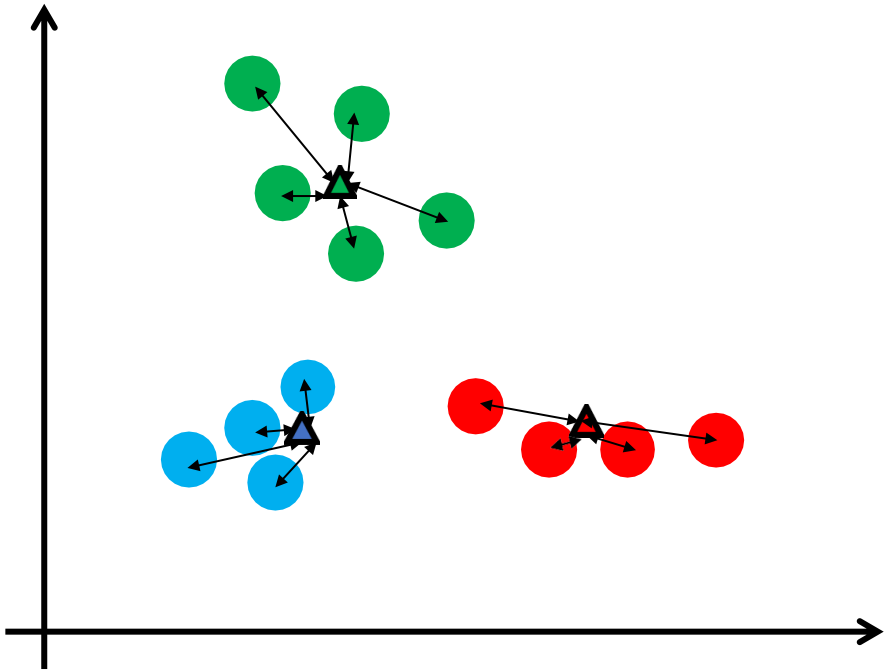
K-Means Clustering Loss

- K-means clustering chooses centroids that minimize loss of training examples Z

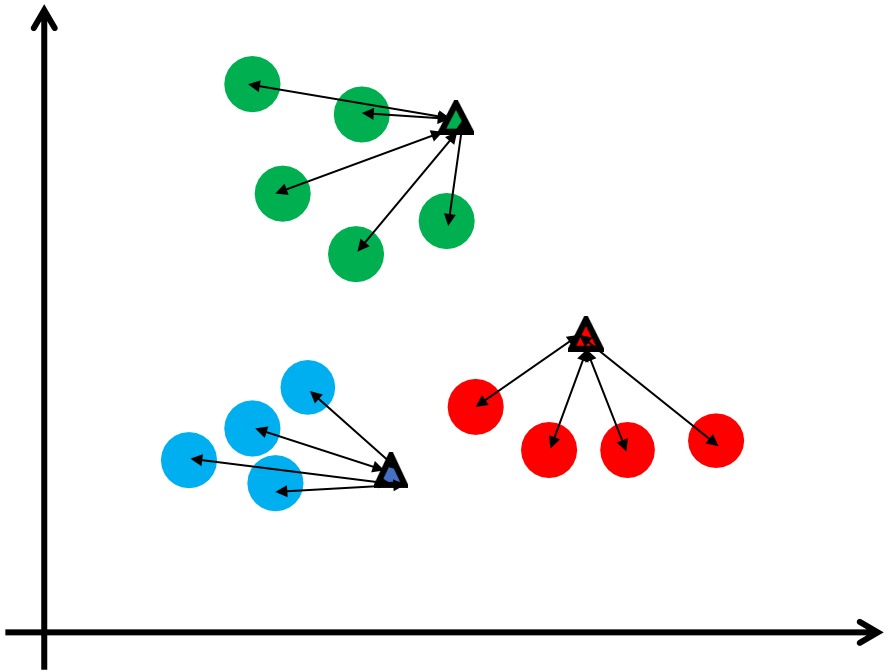
- Compute MSE of each point in the training data to its nearest centroid:

$$L(\mu; Z) = \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - \mu_{f_\mu(x_i)} \right\|_2^2$$

K-Means Clustering Loss



K-Means Clustering Loss



Optimization

- If we know the assignment of points to clusters:
  - **Mean of point per cluster** is the vector that minimizes the squared loss!
  - Without knowledge of true assignments, this optimization is non-convex and has many local optimums

- If we know the centroids:
  - Assignment for a point can be done by finding its closet centroid

K-Means Clustering Algorithm

- An iterative clustering algorithm
- **Initialize**: Pick *K* random points as cluster centers
- **Alternate**:
  - Assign data points to closest cluster center
  - Change the cluster center to the average of its assigned points
- **Stop** when no points' assignments change

K-Means Clustering Algorithm

Kmeans(Z):

for $j \in \{1, \dots, k\}$:

$\mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1,2, \dots\}$:

for $i \in \{1, \dots, n\}$:

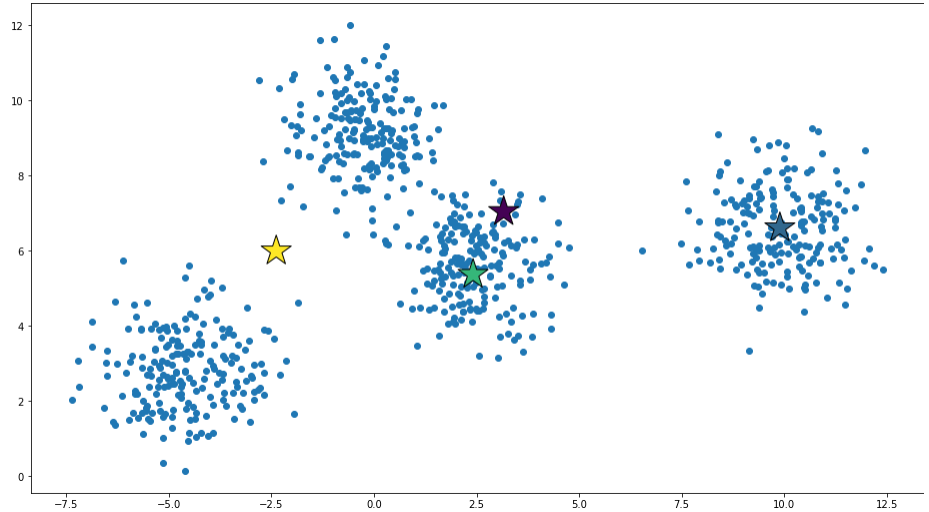
$j_{t,i} \leftarrow f_{\mu_t}(x_i)$

for $j \in \{1, \dots, k\}$:

$\mu_{t,j} \leftarrow \text{mean}(\{ x_i \mid j_{t,i} = j \})$

if $\mu_t = \mu_{t-1}$:

return μ_t



K-Means Clustering Algorithm

Kmeans(Z):

for $j \in \{1, \dots, k\}$:

$\mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1,2, \dots\}$:

for $i \in \{1, \dots, n\}$:

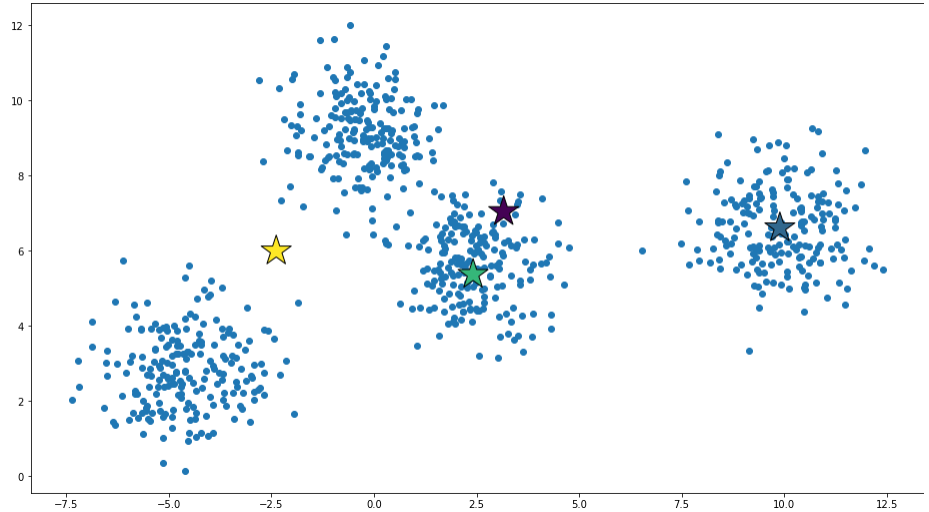
$j_{t,i} \leftarrow f_{\mu_t}(x_i)$

for $j \in \{1, \dots, k\}$:

$\mu_{t,j} \leftarrow \text{mean}(\{ x_i \mid j_{t,i} = j \})$

if $\mu_t = \mu_{t-1}$:

return μ_t



K-Means Clustering Algorithm

Kmeans(Z):

for $j \in \{1, \ldots, k\}$:

$\mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1,2, \ldots\}$:

for $i \in \{1, \ldots, n\}$:

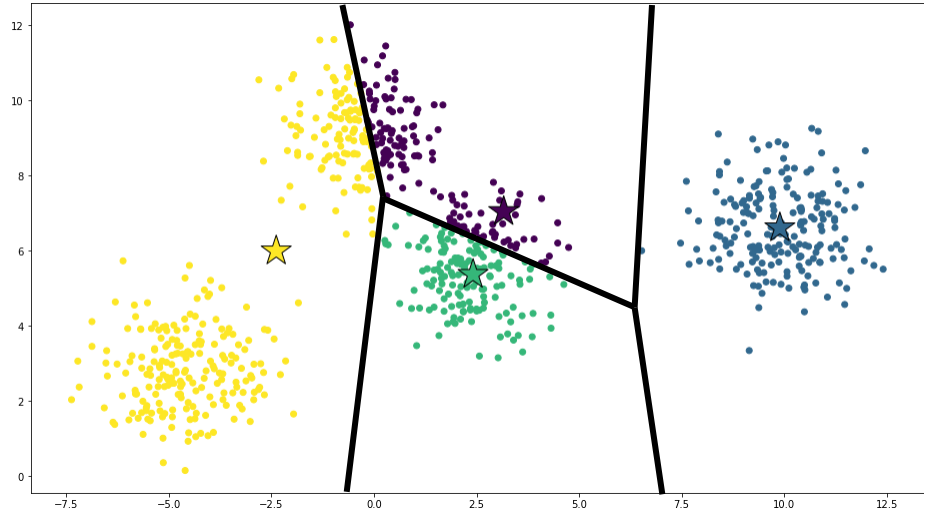
$j_{t,i} \leftarrow f_{\mu_t}(x_i)$

for $j \in \{1, \ldots, k\}$:

$\mu_{t,j} \leftarrow \text{mean}(\{ x_i \mid j_{t,i} = j \})$

if $\mu_t = \mu_{t-1}$:

return μ_t



K-Means Clustering Algorithm

Kmeans(Z):

for $j \in \{1, \dots, k\}$:

$\mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1,2, \dots\}$:

for $i \in \{1, \dots, n\}$:

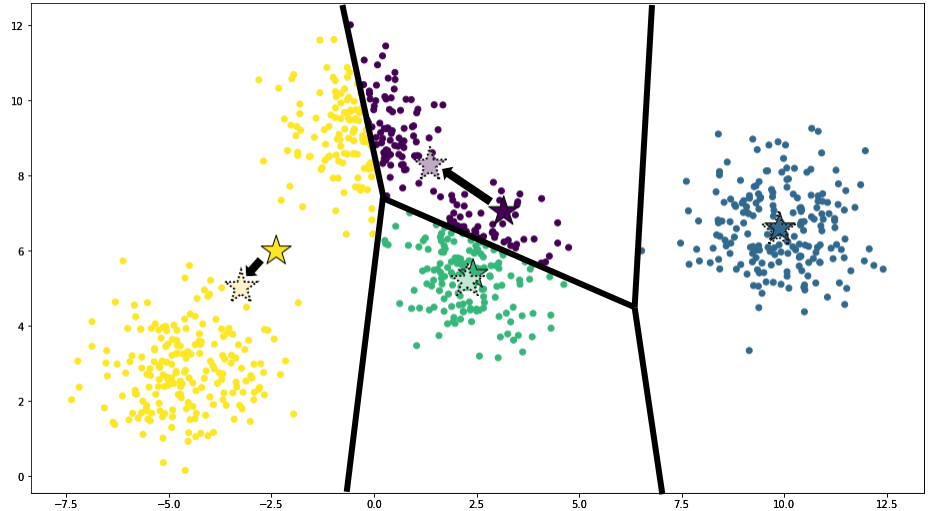
$j_{t,i} \leftarrow f_{\mu_t}(x_i)$

for $j \in \{1, \dots, k\}$:

$\mu_{t,j} \leftarrow \text{mean}(\{ x_i \mid j_{t,i} = j \})$

if $\mu_t = \mu_{t-1}$:

return μ_t



Example: K-Means for Image Segmentation

Original





Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.

K=2





[Slide from David Sontag]

Example: K-Means for Image Segmentation



[Slide from David Sontag]

K-Means Convergence

- K-Means take an alternating optimization approach
- Each step is guaranteed to decrease the loss objective – thus guaranteed to converge

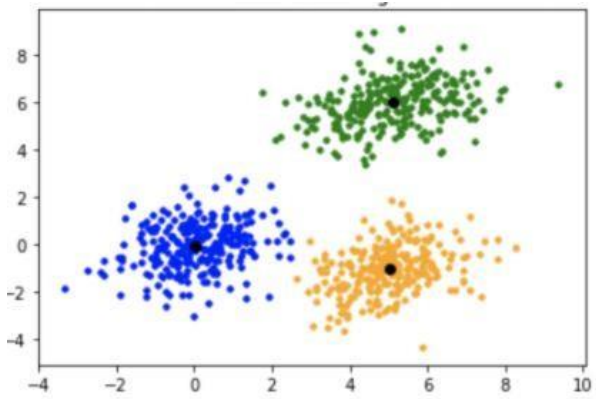
Initialization matters: Run Multiple times

Sensitive to initialization

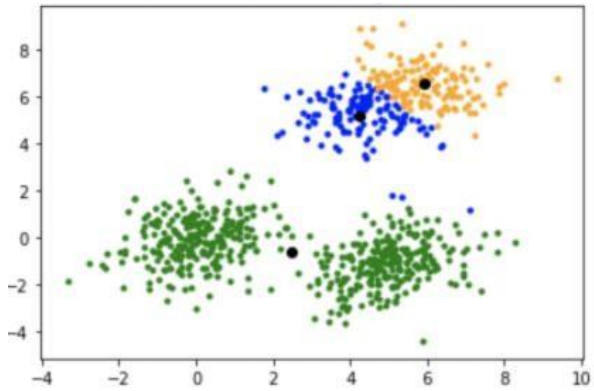
Non-convex objective ⇒ different initializations ⇒ different local minima.

Common approach: run K-means multiple times and pick the best solution (lowest loss).

Desired clustering



Poor initialization: bad clustering



K-means++ Algorithm: initialization: spreads initial centroids to reduce bad starts.

Feature scaling: K-means is scale-sensitive

Distance is Euclidean ⇒ a feature with large numeric scale dominates.

If features have different units (kg vs cm), standardize:

$$x_{ij} \leftarrow \frac{x_{ij} - \text{mean}(x_{.j})}{\text{std}(x_{.j})}$$

Rule of Thumb: preprocessing can change clusters more than any fancy trick inside K-means.

Number of Clusters

- As K becomes large
  - MSE becomes small
  - Many clusters → might be less useful

- Choice of K is subjective

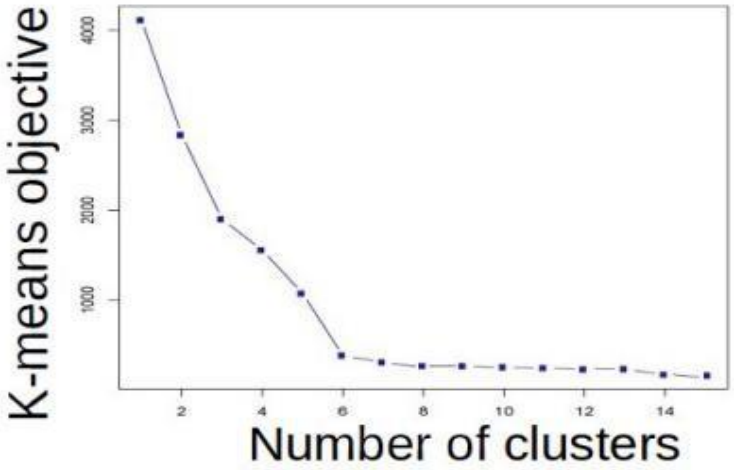
Choosing K: elbow method

Run the algorithm for various K; Plot the loss objective and look for diminishing returns.

Small *K***:** underfit (too few clusters).

Large *K***:** overfit (tiny clusters, less interpretability).

(E.g.: If $K = N$, each data point forms its own cluster and the loss becomes 0.)



K=6 is the elbow point

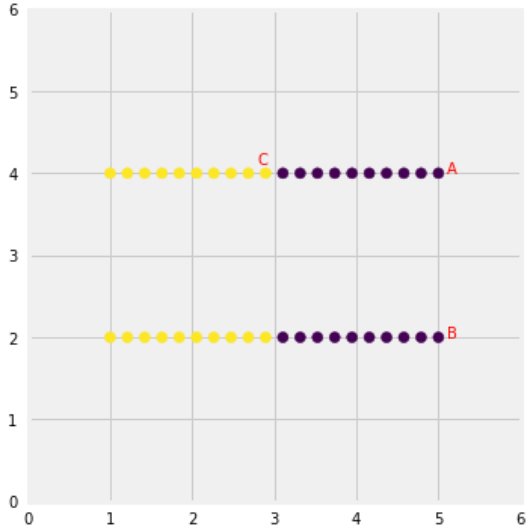
Elbow point: The value of K beyond which further increase in K yields only marginal improvement in the k-means objective..

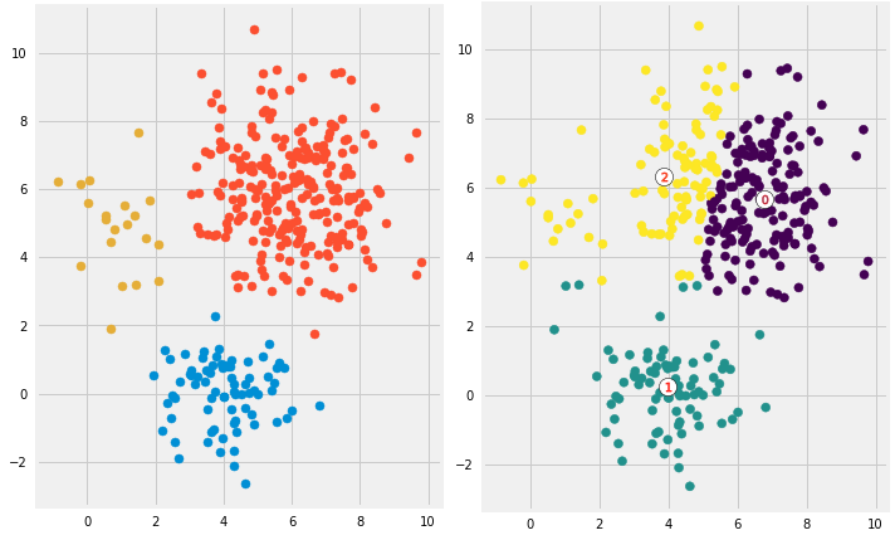
Silhouette score:

Another method to choose K

compares how close a point is to its own cluster versus the nearest other cluster

Common failure modes for K-Means



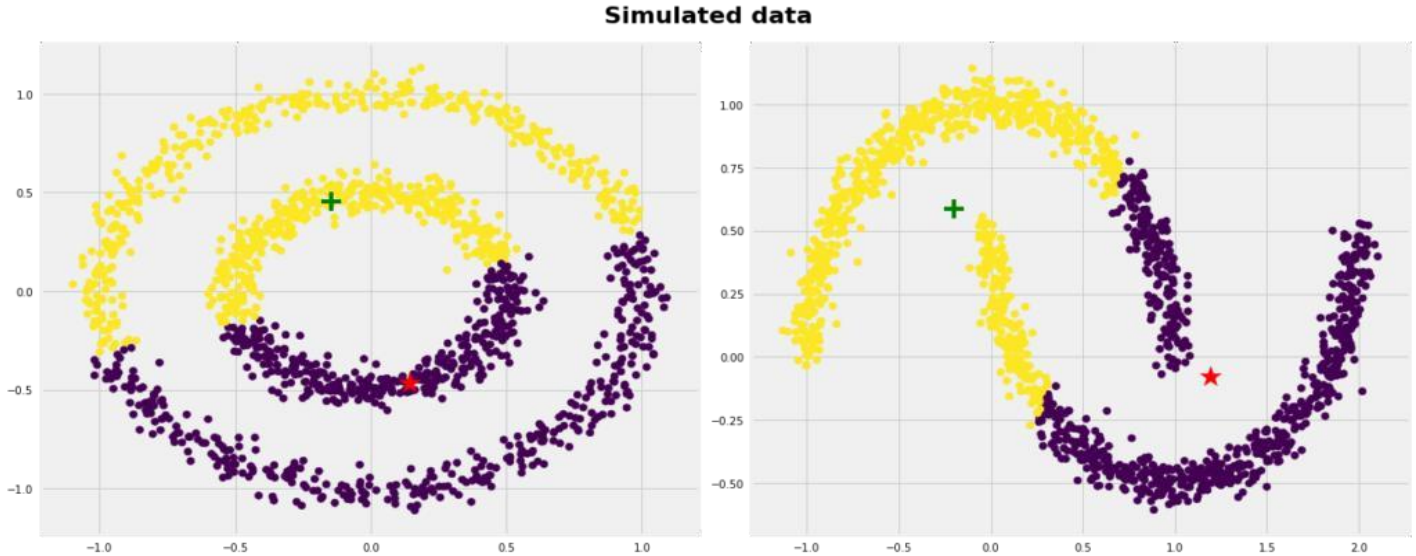


- k-means works best for roughly spherical clusters.
- It may fail for elongated clusters.

- May do badly if clusters are not of nearly equal sizes
- Could ignore small clusters and split a large cluster to minimize the within-cluster variance (loss function)

Fig Src: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a/

Common failure modes for K-Means



- K-mean assumes that the decision boundary between any two clusters is linear
- Since these clusters can't be separated by linear boundaries, it fails.

Fig Src: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a/

Beyond k-means: Alternative Approaches

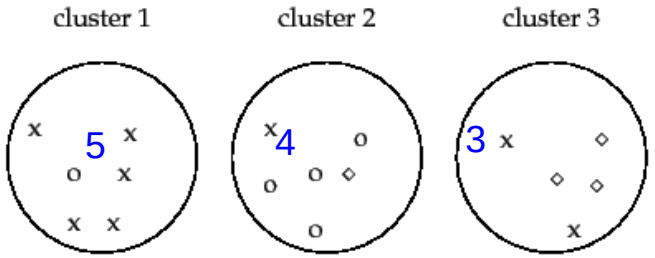
- When clusters are non-spherical or non-convex, consider:

  Kernel k-means
  - Implicitly maps data to a higher-dimensional feature space
  - Can capture non-linear cluster boundaries
- Spectral Clustering
  - Uses similarity graph +eigenvectors
  - Effective for complex, non-convex structures
- DBSCAN (Density-Based)
  - Groups points based on density
  - Handles arbitrary shapes and noise
  - No need to specify K
- Key Idea: Different clustering methods encode different geometric assumptions.

Evaluating Clustering (with Ground Truth)

When true labels are available, compare clustering with ground truth.

Purity: Looks at how many points in each cluster belong to the majority class in that cluster



Purity = (5+4+3)/17 ≈ 0.71

Other Metrics

- **Rand Index (RI)**
- **Normalized Mutual Information (NMI)**

3 classes (x, o , △, assuming known ground truth labels)

Key Idea:
Clustering evaluation must account for **label permutation** and **pairwise consistency**.

Acknowledgement

Most of the slides are based on the ML course of:

- Mingmin Zhao and Jiatao Gu, University of Pennsylvania
- Subrahmanya Swamy Peruru, IIT Kanpur

THANK YOU