Introduction to Machine Learning

Clustering Algorithms:

K-Means and Hierarchical Clustering

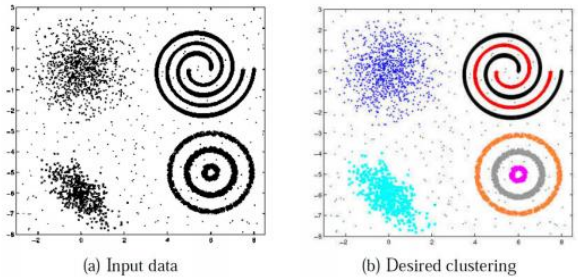
Prof. Subrahmanya Swamy Peruru
Department of Electrical Engineering
IIT Kanpur

Clustering

In some cases, we may not know the right number of clusters in the data and may want to learn from data itself

Given: N unlabeled examples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$; desired no. of partitions K

Goal: Group the examples into K "homogeneous" partitions



(a) Input data (b) Desired clustering

Picture courtesy: "Data Clustering: 50 Years Beyond K-Means", A.K. Jain (2008)

Loosely speaking, it is classification without ground truth labels

A good clustering is one that achieves

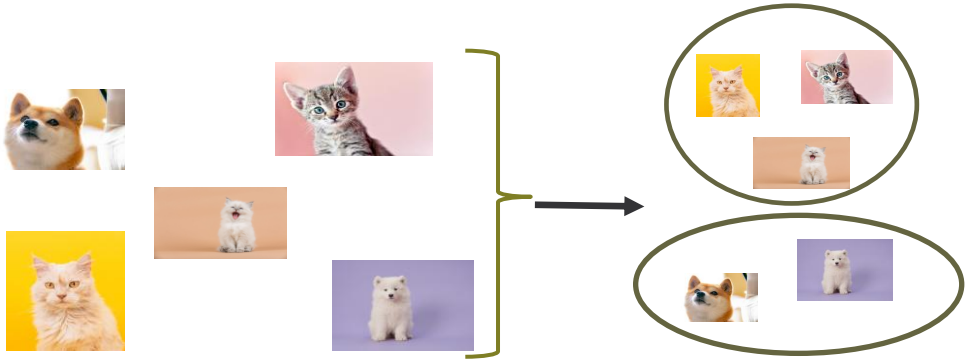
- High within-cluster similarity
- Low inter-cluster similarity

Similarity is subjective: Clustering output depends on choice of distance or similarity metric

Slides are adapted from Prof. Piyush Rai lectures

Clustering: Some Examples

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

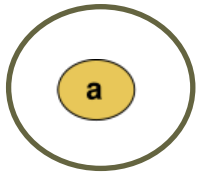


- Clustering web-search results
- Clustering (people) nodes in (social) networks/graphs
- .. and many more..

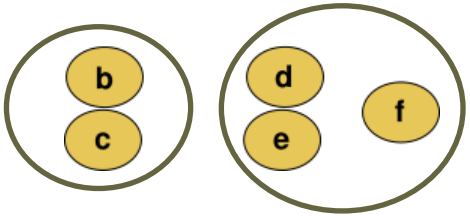
Picture courtesy: http://people.cs.uchicago.edu/~pff/segment/

Types of Clustering

Partitional clustering

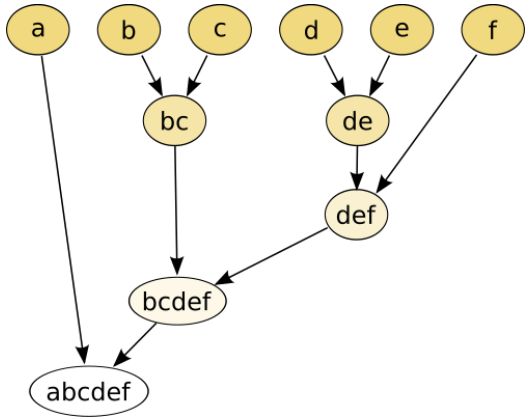


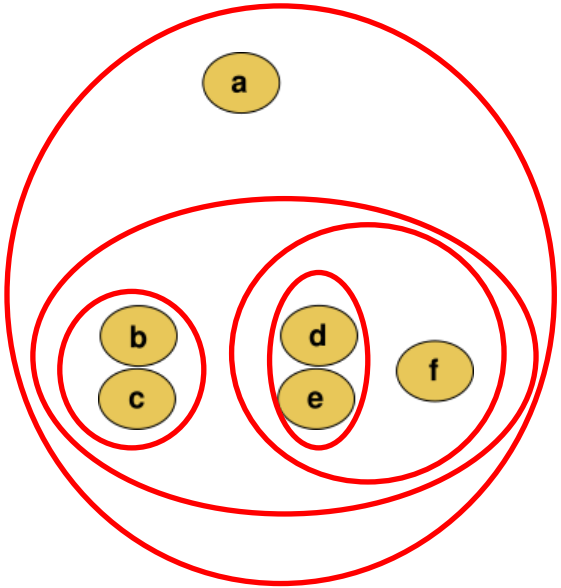
$K = 3$



No need to choose K

Hierarchical clustering



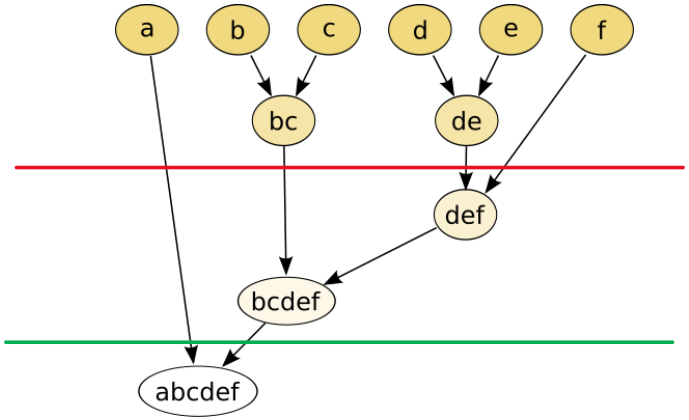


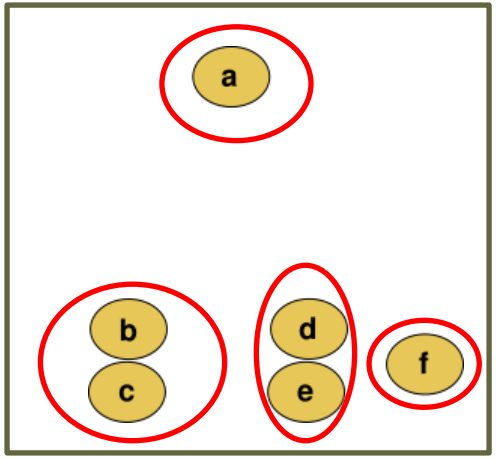
How to measure distance between 2 clusters?

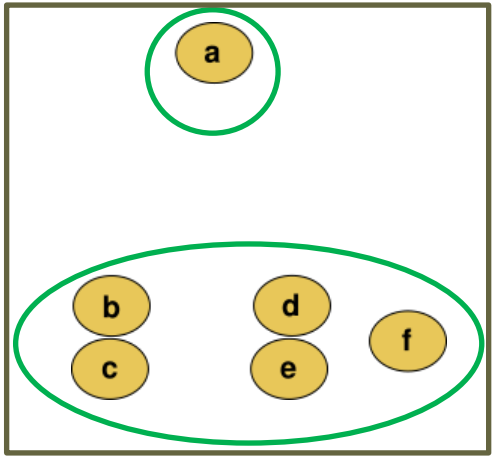
Single linkage: distance = closest pair of points across clusters
Complete linkage: distance = farthest pair of points across clusters
Average linkage: distance = average of all cross-cluster pairwise distances

Hierarchical Clustering

Cutting the tree at different levels results in different partitions

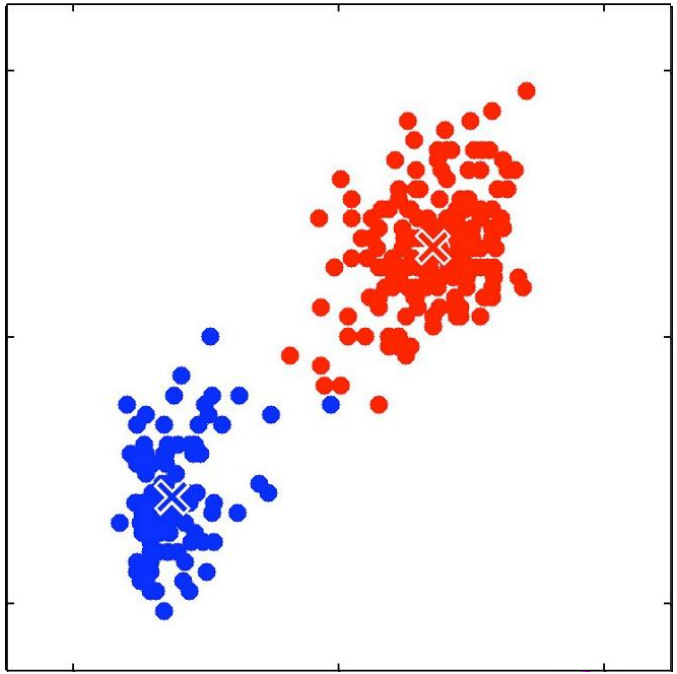






K-Means Algorithm

K-means: An Illustration



K-means algorithm

- **Input:** N unlabeled examples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$; $\mathbf{x}_n \in \mathbb{R}^D$; desired no. of partitions K
- **Desired Output:**
  - The cluster assignments of the data samples $z_1, z_2, z_3, \dots, z_N$; $z_n \in \{1,2, \dots, K\}$
  - The K cluster means denoted by $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$

K-means Algorithm

1. Initialize K cluster means $\mu_1, \dots, \mu_K$
2. For $n = 1, \dots, N$, assign each point $\boldsymbol{x}_n$ to the closest cluster

$$z_n = \arg\min_{k \in \{1,\dots,K\}} ||\boldsymbol{x}_n - \mu_k||^2$$

3. Suppose $\mathcal{C}_k = \{\boldsymbol{x}_n : z_n = k\}$. Re-compute the means

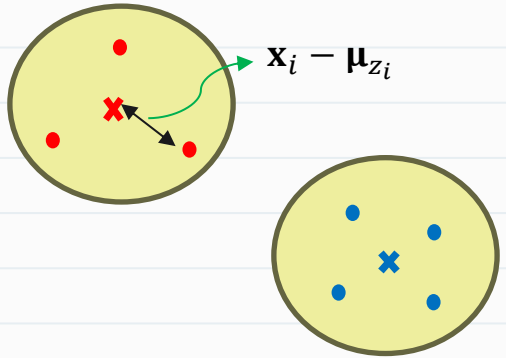
$$\mu_k = \text{mean}(\mathcal{C}_k), \quad k = 1, \dots, K$$

4. Go to step 2 if not yet converged

Stopping criterion: Stop when cluster assignments (equivalently, centroids) no longer change.

K-Means Derivation

K-means Objective



Minimize within-cluster sum of squared distances

$$\min_{\{z_i\},\{\mu_k\}} \sum_{i=1}^{n} \| \mathbf{x}_i - \boldsymbol{\mu}_{z_i} \|_2^2.$$

z_i is the cluster ID of sample $\mathbf{x}_i$

μ_k is the centroid of cluster k.

Key point:

Exact optimization is computationally intractable in general.

K-means is a greedy alternating minimization algorithm that converges to a local minimum.

K-means Objective

Let $r_{ik} \in \{0,1\}$ indicate if point i is assigned to cluster k:

$$r_{ik} = \begin{cases} 1 & \text{if } z_i = k, \\ 0 & \text{otherwise.} \end{cases} \quad \text{with} \sum_{k=1}^{K} r_{ik} = 1.$$

Example:

Suppose $K = 3$, and if $\mathbf{x}_5$ is assigned to cluster 2, i.e., $z_5 = 2$. Then $r_{51} = 0,\ r_{52} = 1,\ r_{53} = 0$

Then the loss $||\mathbf{x}_5 - \boldsymbol{\mu}_{z_5}||^2 = ||\mathbf{x}_5 - \boldsymbol{\mu}_2||^2 = r_{51} \times ||\mathbf{x}_5 - \boldsymbol{\mu}_1||^2 + r_{52} \times ||\mathbf{x}_5 - \boldsymbol{\mu}_2||^2 + r_{53} \times ||\mathbf{x}_5 - \boldsymbol{\mu}_3||^2$

i.e., $||\mathbf{x}_i - \boldsymbol{\mu}_{z_i}||^2 = \sum_{k=1}^{K} r_{ik} \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel_2^2$

Then the K-means objective becomes

$$\min_{\{r_{ik}\},\{\boldsymbol{\mu}_k\}} \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel_2^2$$

Iterative Algorithm:

Fix centroids and optimize assignments

Fix assignments and optimize centroid locations.

Step 1: Fix centroids and optimize cluster assignments

Fix centroids $\{\boldsymbol{\mu}_k\}$ and solve for the optimal cluster assignments $\{r_{ik}\}$

$$\min_{\{r_{ik}\}} \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel_2^2.$$

Since the objective is separable across data points, for each i we solve:

$$\min_{r_{i1},\ldots,r_{iK}} \sum_{k=1}^{K} r_{ik} \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel_2^2 \quad \text{s.t. } r_{ik} \in \{0,1\}, \sum_{k} r_{ik} = 1.$$

Observation: only one r_{ik} can be 1, so choose the smallest distance:

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{j \in \{1,\ldots,K\}} \parallel \mathbf{x}_i - \boldsymbol{\mu}_j \parallel_2^2, \\ 0 & \text{otherwise.} \end{cases}$$

So: assignment update = assign to the cluster with the closest centroid

Step 2: Fix Assignments and optimize centroids

Fix assignments $\{r_{ik}\}$ .Optimize w.r.t. $\boldsymbol{\mu}_k$:

$$J = \sum_{i=1}^{n}\sum_{k=1}^{K} r_{ik} \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel_2^2 = \sum_{k=1}^{K}\sum_{i=1}^{n} r_{ik} \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel_2^2.$$

The terms for different k separate. For a fixed k, define

$$J_k(\boldsymbol{\mu}_k) = \sum_{i=1}^{n} r_{ik} \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel_2^2.$$

Differentiate and set to zero:

$$\nabla_{\boldsymbol{\mu}_k} J_k = \sum_{i=1}^{n} r_{ik}\, 2(\boldsymbol{\mu}_k - \mathbf{x}_i) = 0 \quad \Rightarrow \quad \boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} r_{ik}\, \mathbf{x}_i}{\sum_{i=1}^{n} r_{ik}}.$$

So: centroid update = mean of points in that cluster.

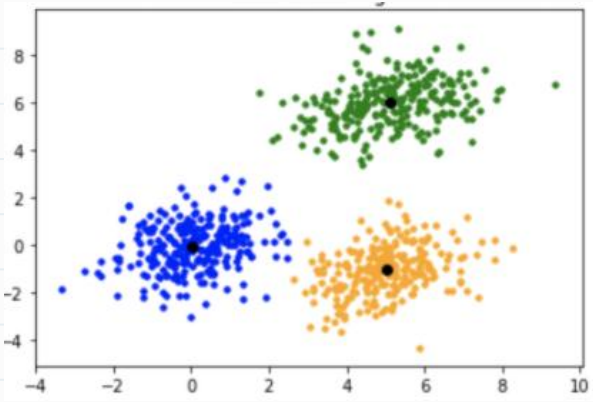
Practical Aspects of K-Means

Initialization matters: Run Multiple times

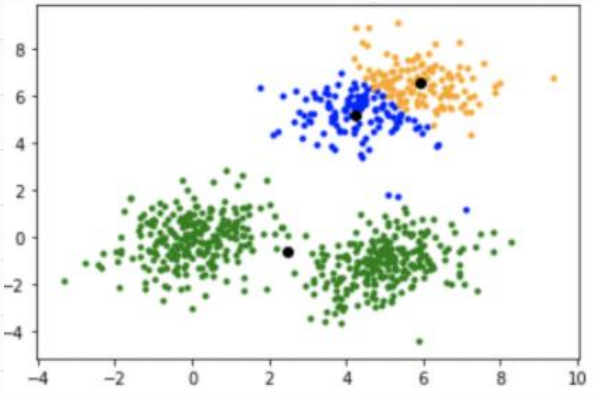
Non-convex objective ⇒ different initializations ⇒ different local minima.

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:

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

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

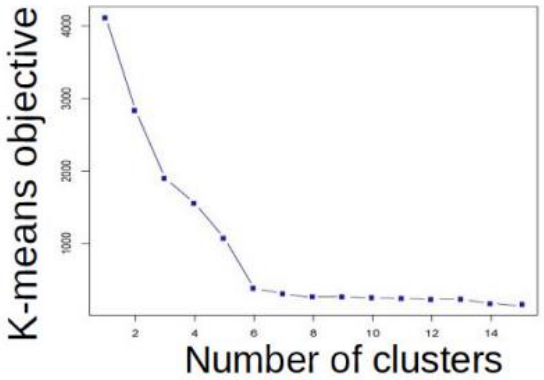
Choosing K: elbow method

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

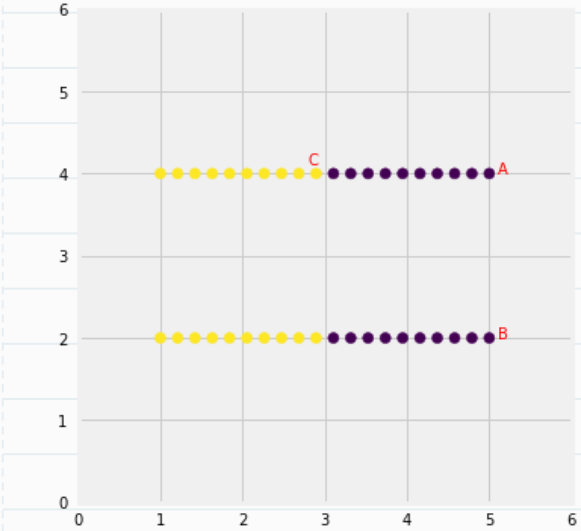
Elbow point: The value of K beyond which further increase in K yields only marginal improvement in the k-means objective..

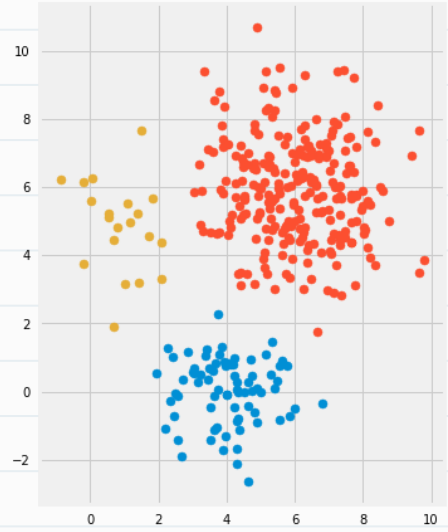
Silhouette score:

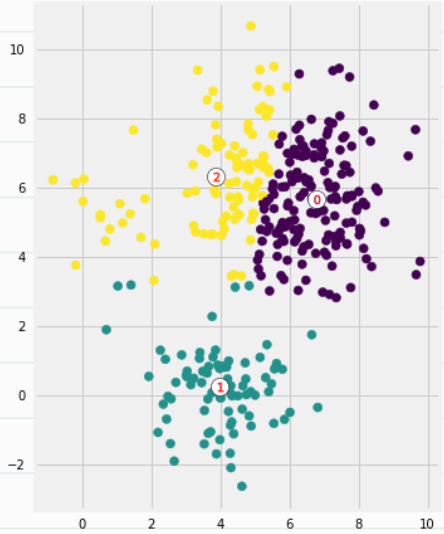
Another method to choose K

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

Common failure modes for K-Means



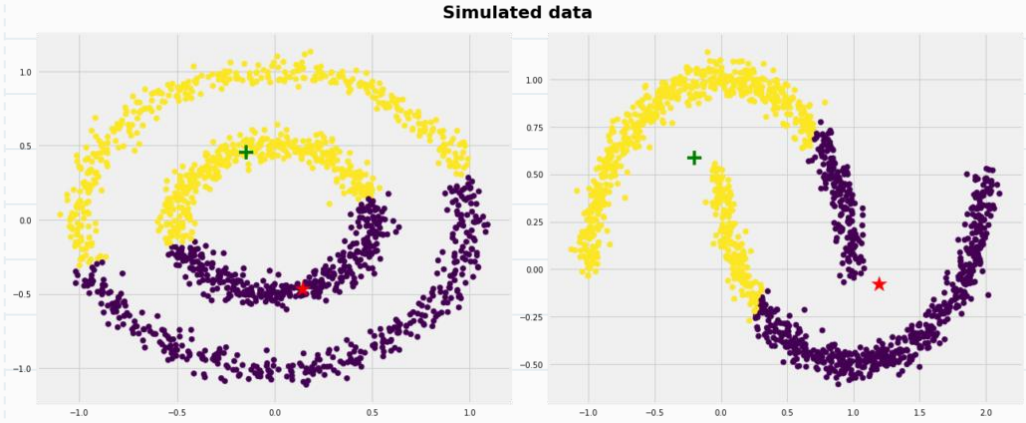




- k-means works best for roughly spherical clusters.
- It may fail for elongated or anisotropic clusters.
- May do badly if clusters are not of nearly equal sizes
- Could ignore small clusters and split a large cluster to minimize the within-cluster variance (loss function)

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)

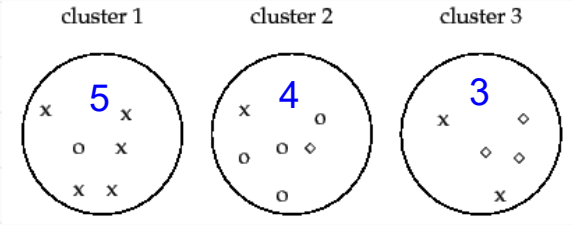
- Groups points based on density
- Handles arbitrary shapes and noise
- No need to specify K

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

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

Other Metrics

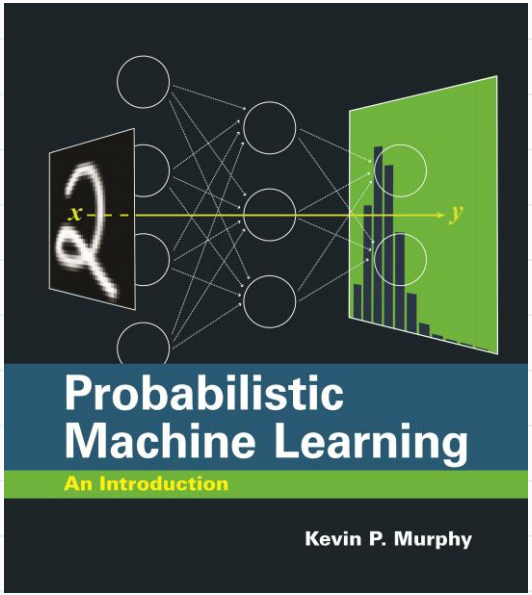
- **Rand Index (RI)**
- **Normalized Mutual Information (NMI)**

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

Reference

Chapter 21

of “Probabilistic Machine Learning: An Introduction” book by Kevin P. Murphy



Thank You