

Machine learning

Clustering¹

Hamid Beigy

Sharif University of Technology

December 25, 2021



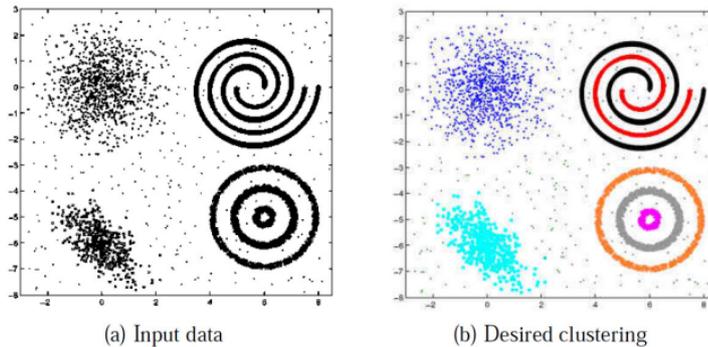
¹Some slides are taken from P. Rai slides



1. Introduction
2. K-means clustering
3. Hierarchical Clustering
4. Model-based clustering
5. Cluster validation and assessment
6. Reading

Introduction

- ▶ Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters.
- ▶ Dissimilarities and similarities are assessed based on the feature values describing the objects and often involve distance measures.
- ▶ Clustering is usually **an unsupervised learning** problem.
- ▶ Consider a dataset $X = \{x_1, \dots, x_N\}$, $x_i \in R^D$.
- ▶ Assume there are K clusters C_1, \dots, C_K .
- ▶ The goal is to **group** the examples into K **homogeneous** partitions.



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

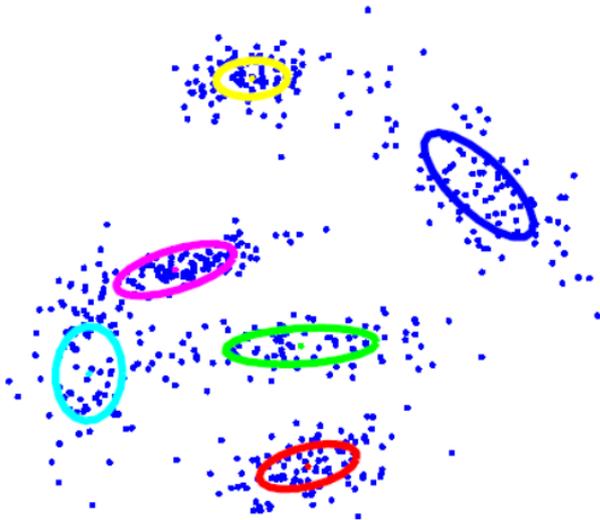


- ▶ A good clustering is one that achieves:
 - ▶ High within-cluster similarity
 - ▶ Low inter-cluster similarity
- ▶ Applications of clustering
 - ▶ Document/Image/Webpage Clustering
 - ▶ Image Segmentation
 - ▶ Clustering web-search results
 - ▶ Clustering (people) nodes in (social) networks/graphs
 - ▶ Pre-processing phase

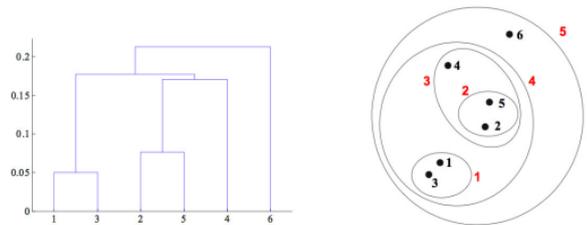


- ▶ The clustering methods can be compared using the following aspects:
 - ▶ **The partitioning criteria** : In some methods, all the objects are partitioned so that no hierarchy exists among the clusters.
 - ▶ **Separation of clusters** : In some methods, data partitioned into mutually exclusive clusters while in some other methods, the clusters may not be exclusive, that is, a data object may belong to more than one cluster.
 - ▶ **Similarity measure** : Some methods determine the similarity between two objects by the distance between them; while in other methods, the similarity may be defined by connectivity based on density or contiguity.
 - ▶ **Clustering space** : Many clustering methods search for clusters within the entire data space. These methods are useful for low-dimensionality data sets. With high- dimensional data, however, there can be many irrelevant attributes, which can make similarity measurements unreliable. Consequently, clusters found in the full space are often meaningless. It's often better to instead search for clusters within different subspaces of the same data set.

Flat or Partitional clustering (Partitions are independent of each other)



Hierarchical clustering (Partitions can be visualized using a tree structure - a dendrogram)



Possible to view partitions at different levels of granularities (i.e., can refine/coarsen clusters) using different K .

K-means clustering



- ▶ Associate a **prototype** μ_k , $k = 1, \dots, K$ with each cluster.
- ▶ Let r_{nk} be the **indicator** of $x_n \in C_k$.
- ▶ The goal is to minimize

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

- ▶ Goal is to find $\{r_{nk}\}$ and $\{\mu_k\}$.
- ▶ Optimization is performed by alternating minimization
 - ▶ Optimize over $\{r_{nk}\}$ for a fixed $\{\mu_k\}$.
 - ▶ Optimize over $\{\mu_k\}$ for a fixed $\{r_{nk}\}$.
- ▶ Initialize with arbitrary choices of $\{\mu_k\}$
- ▶ Iterative updates continue till convergence
- ▶ Guaranteed to converge, as objective is monotonic decreasing



- ▶ **Input:** N examples $\{x_1, \dots, x_N\}$; $x_n \in \mathbb{R}^D$; the number of partitions K
- ▶ **Initialize:** K cluster means μ_1, \dots, μ_K , each $\mu_k \in \mathbb{R}^D$.

Usually initialized **randomly**, but good initialization is crucial; **many smarter initialization heuristics** exist (e.g., **K-means++**, Arthur & Vassilvitskii, 2007)

- ▶ **Repeat:**

- ▶ (Re)-Assign each example x_n to its closest cluster center (based on the smallest Euclidean distance)

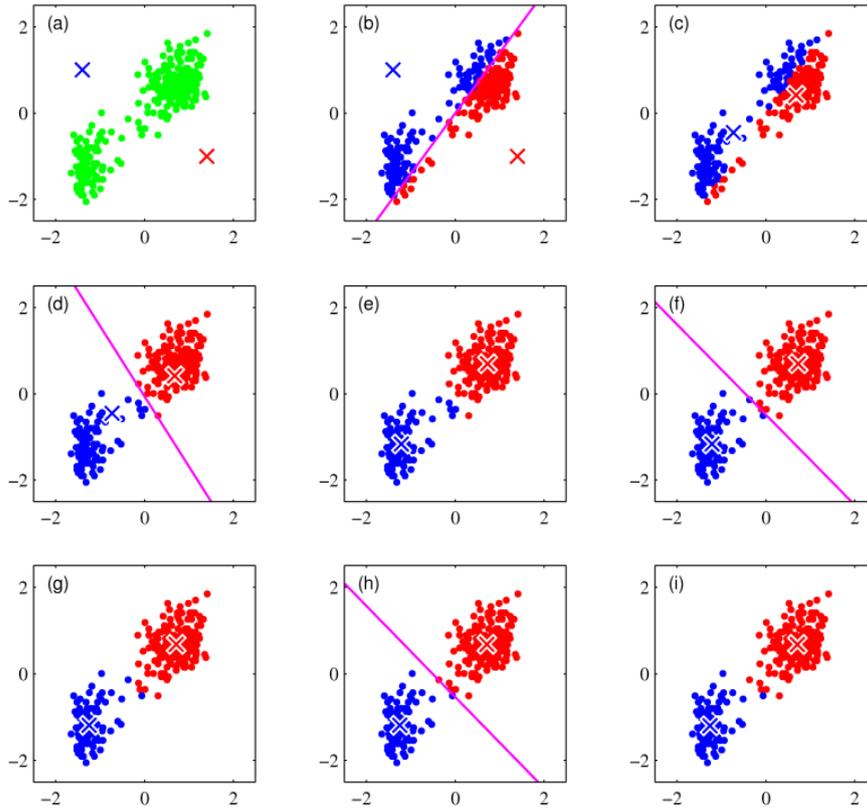
$$r_{nk} = \begin{cases} 1 & \text{if } k = \underset{j}{\operatorname{argmin}} \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

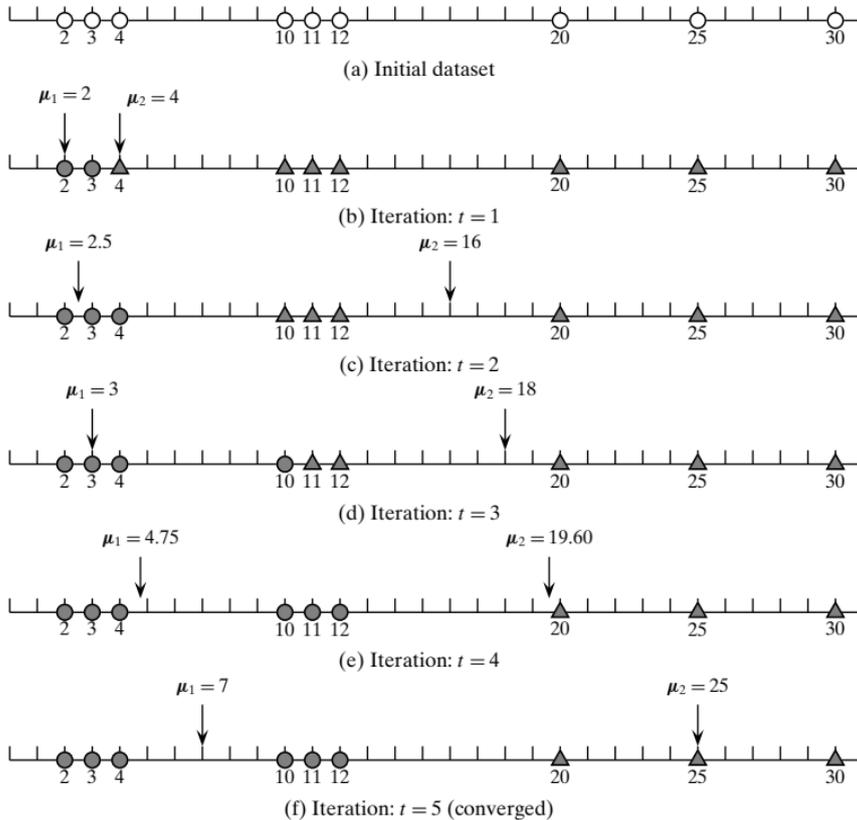
Let C_k is the set of examples assigned to cluster k with center μ_k .

- ▶ **Update** the cluster means

$$\mu_k = \frac{1}{|C_k|} \sum_{x_n \in C_k} x_n = \frac{\sum_{n=1}^N r_{nk} x_n}{\sum_{n=1}^N r_{nk}}$$

- ▶ **Stop:** when cluster means or the “loss” (defined later) doesn’t change by much





$K = 2$



$K = 3$

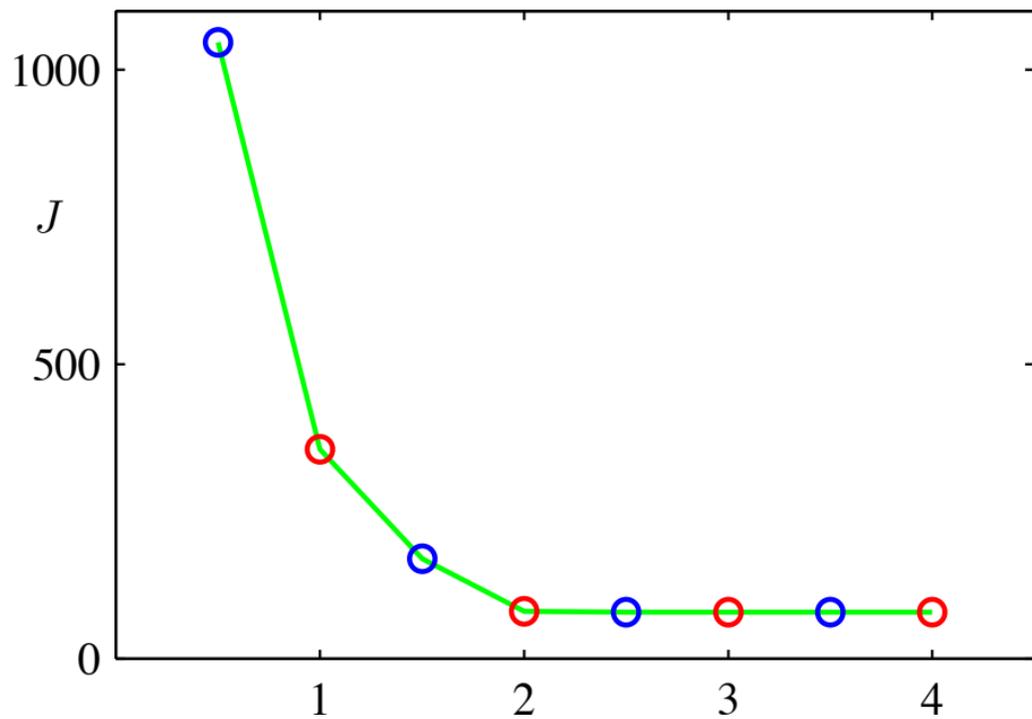


$K = 10$



Original image





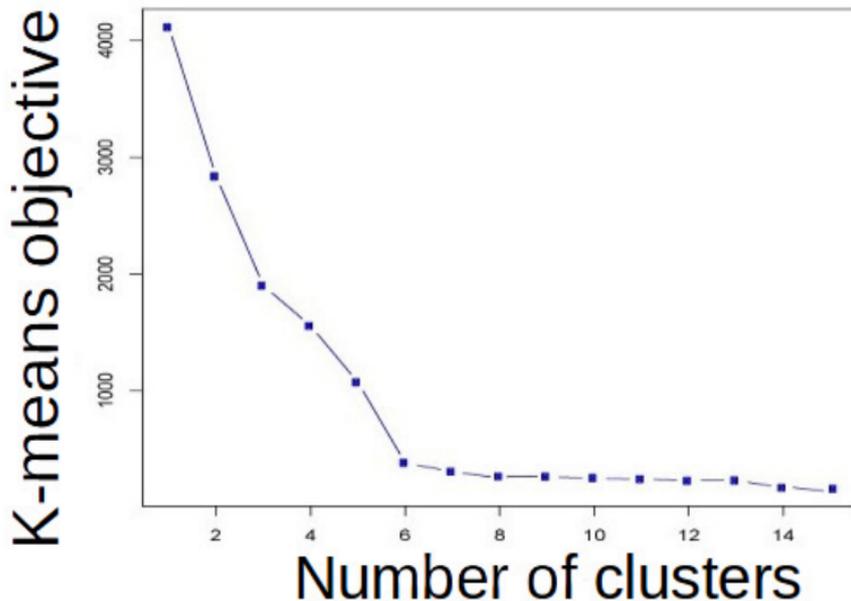


- ▶ Consider the K-means objective function

$$J(X, \mu, r) = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

- ▶ It is a **non-convex** objective function, so may have **many local minima**.
- ▶ Also **NP-hard** to minimize in general (note that r is discrete)
- ▶ The K-means algorithm is a heuristic to optimize this function
- ▶ K-means algorithm alternated between the following two steps
 - ▶ assign points to closest centers
 - ▶ recompute the center means
- ▶ The algorithm usually converges to a local minima. Multiple runs with different initializations are usually tried to find a good solution.

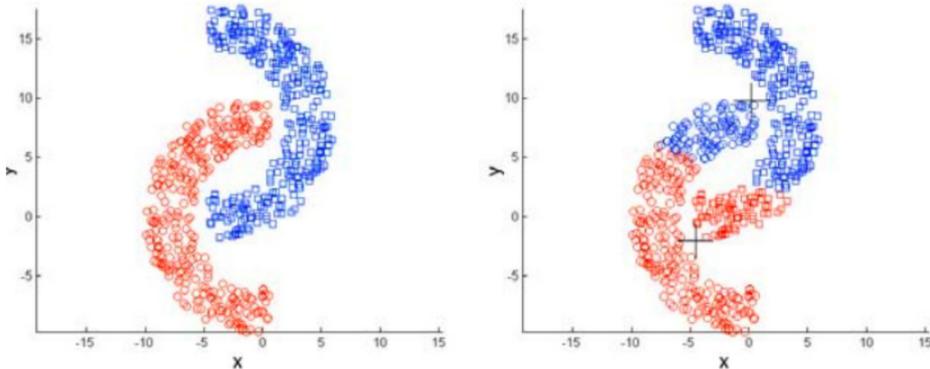
- ▶ Try different values of K , plot the K-means objective versus K .



- ▶ We can also use information criterion such as **AIC** (Akaike Information Criterion) or **BIC** (Bayesian Information Criterion) and choose K that gives smallest **AIC/BIC** (both penalize large K values)



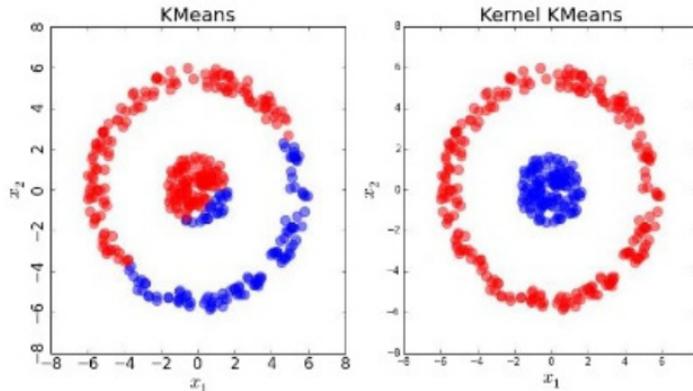
- ▶ Makes **hard assignments of points** to clusters
 - ▶ A point either completely belongs to a cluster or doesn't belong at all
 - ▶ No notion of a soft assignment.
- ▶ Works well only if the clusters are **roughly of equal sizes**.
- ▶ Probabilistic clustering methods such as Gaussian mixture models can handle both these issues
- ▶ K-means also works well only when the clusters are round-shaped and does badly **if the clusters have non-convex shapes**



- ▶ **Kernel K-means** or **Spectral clustering** can handle non-convex

- ▶ The idea is to replace the **Euclidean distance/similarity** computations in K-means by the **kernelized versions**

$$\begin{aligned}d^2(x_n, \mu_k) &= \|\phi(x_n) - \phi(\mu_k)\|^2 \\ &= K(x_n, x_n) + K(\mu_k, \mu_k) - 2K(\mu_k, x_n)\end{aligned}$$





- ▶ Computing $K(x_n, x_n)$ is easy: Simply compute this kernel function.
- ▶ Compute $K(\mu_k, \mu_k)$ as follows (assume N_k is the no. of points in cluster k)

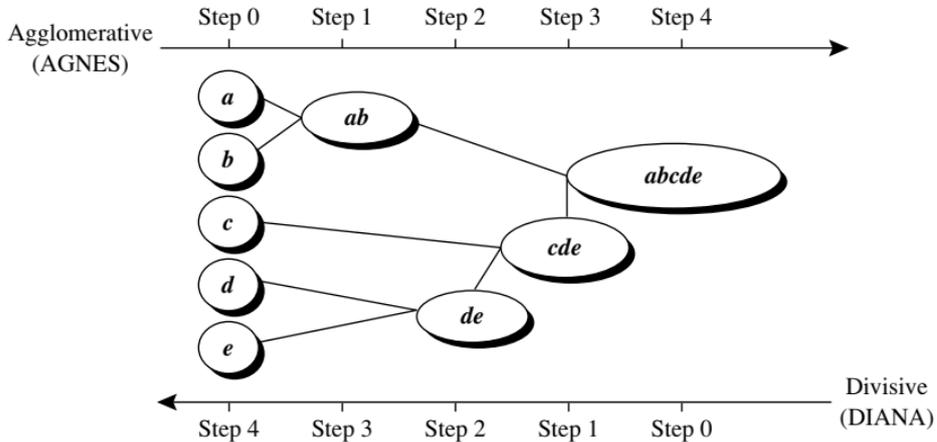
$$\begin{aligned}
 K(\mu_k, \mu_k) &= \phi^T(\mu_k)\phi(\mu_k) = \left[\frac{1}{N_k} \sum_{n=1}^{N_k} \phi(x_n) \right]^T \left[\frac{1}{N_k} \sum_{n=1}^{N_k} \phi(x_n) \right] \\
 &= \frac{1}{N_k^2} \sum_{n=1}^{N_k} \sum_{m=1}^{N_k} \phi^T(x_n)\phi(x_m) \\
 &= \frac{1}{N_k^2} \sum_{n=1}^{N_k} \sum_{m=1}^{N_k} K(x_n, x_m)
 \end{aligned}$$

- ▶ $K(\mu_k, x_n)$ can be computed as

$$\begin{aligned}
 K(\mu_k, x_n) &= \phi^T(\mu_k)\phi(x_n) = \left[\frac{1}{N_k} \sum_{n=1}^{N_k} \phi(x_n) \right]^T \phi(x_n) \\
 &= \frac{1}{N_k} \sum_{m=1}^{N_k} \sum_{m=1}^{N_k} \phi^T(x_n)\phi(x_m) = \frac{1}{N_k} \sum_{m=1}^{N_k} K(x_n, x_m)
 \end{aligned}$$

Hierarchical Clustering

- ▶ A hierarchical clustering method works by grouping data objects into a hierarchy or “tree” of clusters.



- ▶ Hierarchical clustering methods
 - ▶ Agglomerative hierarchical clustering
 - ▶ Divisive hierarchical clustering



- ▶ How measure the distance between two clusters, where each cluster is generally a set.
- ▶ Four widely used measures for distance between clusters are as follows

- ▶ **Minimum distance**

$$d_{\min}(C_i, C_j) = \min_{p \in C_i, q \in C_j} \{|p - q|\}$$

- ▶ **Maximum distance**

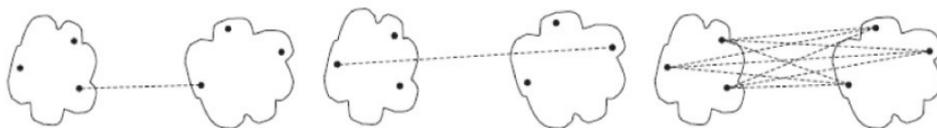
$$d_{\max}(C_i, C_j) = \max_{p \in C_i, q \in C_j} \{|p - q|\}$$

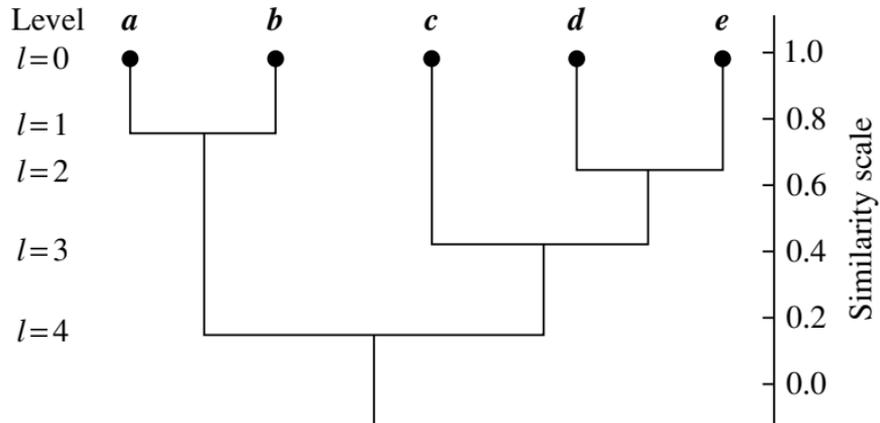
- ▶ **Mean distance**

$$d_{\text{mean}}(C_i, C_j) = |\mu_i - \mu_j|$$

- ▶ **Average distance**

$$d_{\text{avg}}(C_i, C_j) = \frac{1}{N_i N_j} \sum_{p \in C_i, q \in C_j} |p - q|$$





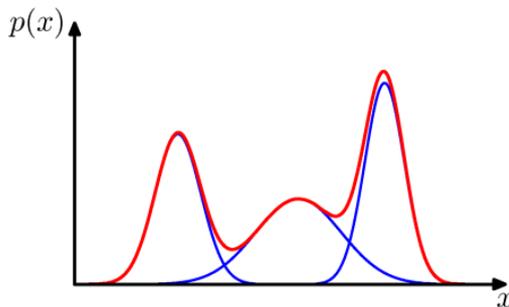
Model-based clustering



- ▶ K -means is closely related to a probabilistic model known as the **Gaussian mixture model**.

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \Sigma_k)$$

- ▶ π_k, μ_k, Σ_k are parameters. π_k are called mixing proportions and each Gaussian is called a **mixture component**.
- ▶ The model is simply a weighted sum of Gaussian. But it is much more powerful than a single Gaussian, because it can model multi-modal distributions.



- ▶ Note that for $p(x)$ to be a probability distribution, we require that $\sum_k \pi_k = 1$ and that for all k we have $\pi_k > 0$. Thus, we may interpret the π_k as probabilities themselves.
- ▶ Set of parameters $\theta = \{\{\pi_k\}, \{\mu_k\}, \{\Sigma_k\}\}$



- ▶ Let use a K -dimensional binary random variable z in which a particular element z_k equals to 1 and other elements are 0.
- ▶ The values of z_k therefore satisfy $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$
- ▶ We define the joint distribution $p(x, z)$ in terms of a marginal distribution $p(z)$ and a conditional distribution $p(x|z)$.
- ▶ The marginal distribution over z is specified in terms of π_k , such that

$$p(z_k = 1) = \pi_k$$

- ▶ We can write this distribution in the form of

$$p(z_k = 1) = \prod_{k=1}^K \pi_k^{z_k}$$

- ▶ The conditional distribution of x given a particular value for z is a Gaussian

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k)$$

- ▶ This can also be written in the form of

$$p(x|z_k = 1) = \prod_{k=1}^K \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}$$



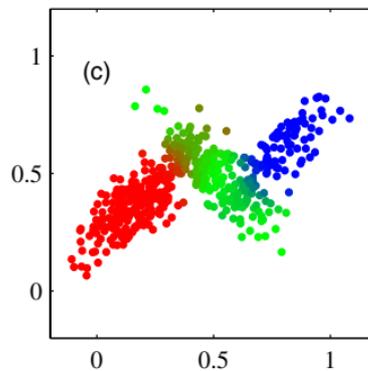
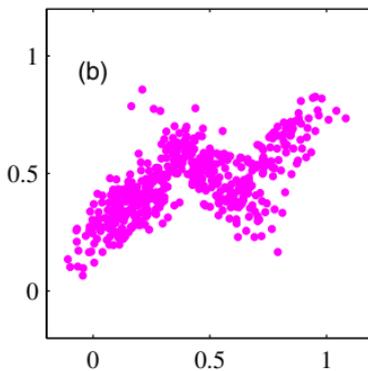
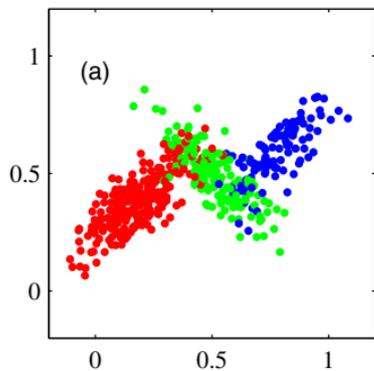
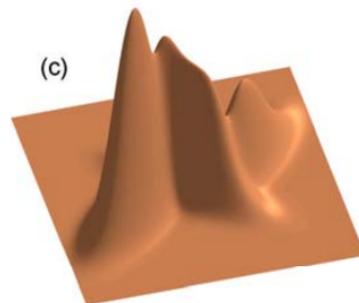
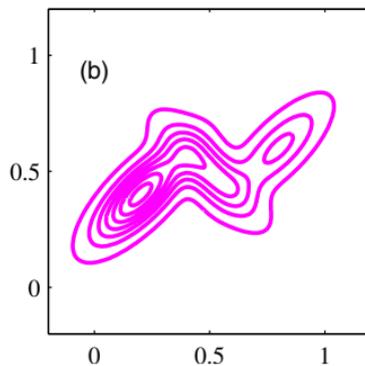
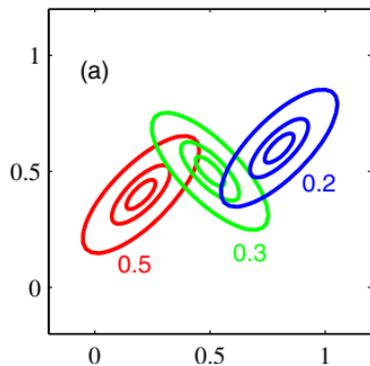
- ▶ The marginal distribution of x equals to

$$p(x) = \sum_z p(z)p(x|z) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

- ▶ We can write $p(z_k = 1|x)$ as

$$\begin{aligned} \gamma(z_k) = p(z_k = 1|x) &= \frac{p(z_k = 1)p(x|z_k = 1)}{p(x)} \\ &= \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(x|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)} \end{aligned}$$

- ▶ We shall view π_k as the prior probability of $z_k = 1$, and the quantity $\gamma(z_k)$ as the corresponding posterior probability once we have observed x .





- ▶ Let $X = \{x_1, \dots, x_N\}$ be drawn i.i.d. from mixture of Gaussian. The log-likelihood of the observations equals to

$$\ln p(x|\mu, \pi, \Sigma) = \sum_{n=1}^N \ln \left[\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right]$$

- ▶ Finding the derivatives of $\ln p(x|\mu, \pi, \Sigma)$ with respect to μ_k and setting it equal to zero, we obtain

$$0 = - \sum_{n=1}^N \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\underbrace{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}_{\gamma(z_{nk})}} \Sigma_k (x_n - \mu_k)$$

- ▶ Multiplying by Σ_k^{-1} and then simplifying, we obtain

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n$$

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$



- ▶ Finding the derivatives of $\ln p(x|\mu, \pi, \Sigma)$ with respect to Σ_k and setting it equal to zero, we obtain

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

- ▶ We maximize $\ln p(x|\mu, \pi, \Sigma)$ with respect to π_k with constraint $\sum_{k=1}^K \pi_k = 1$. This can be achieved using a Lagrange multiplier and maximizing the following quantity

$$\ln p(x|\mu, \pi, \Sigma) + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right).$$

which gives

$$\sum_{n=1}^N \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} + \lambda$$

- ▶ If we now multiply both sides by π_k and sum over k making use of the constraint $\sum_{k=1}^K \pi_k = 1$, we find $\lambda = -N$. Using this to eliminate λ and rearranging we obtain

$$\pi_k = \frac{N_k}{N}$$



1. Initialize μ_k , Σ_k , and π_k , and evaluate the initial value of the log likelihood.
2. **E step** Evaluate $\gamma(z_{nk})$ using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}$$

3. **M step** Re-estimate the parameters using the current value of $\gamma(z_{nk})$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n$$

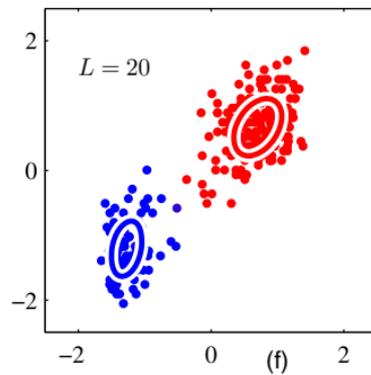
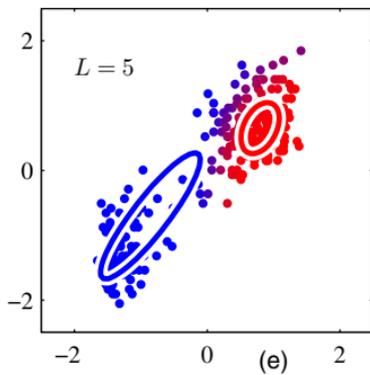
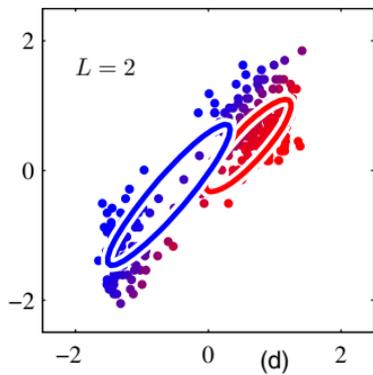
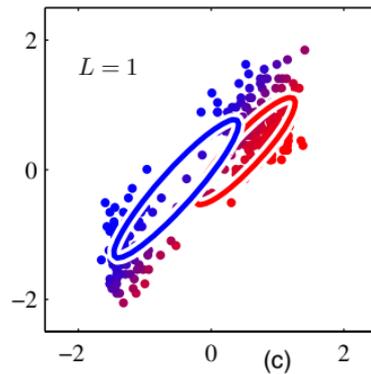
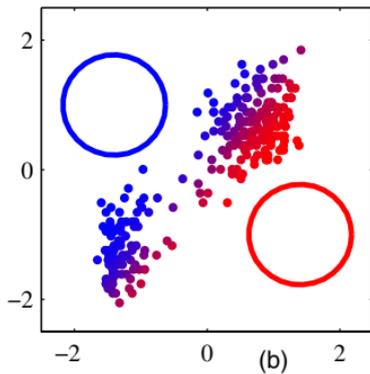
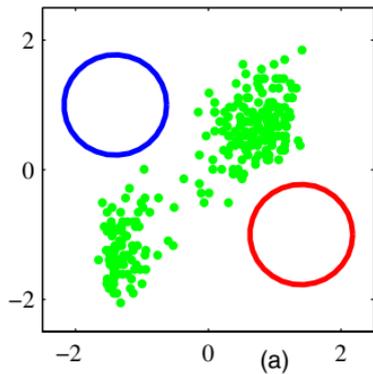
$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

where $N_k = \sum_{n=1}^N \gamma(z_{nk})$.

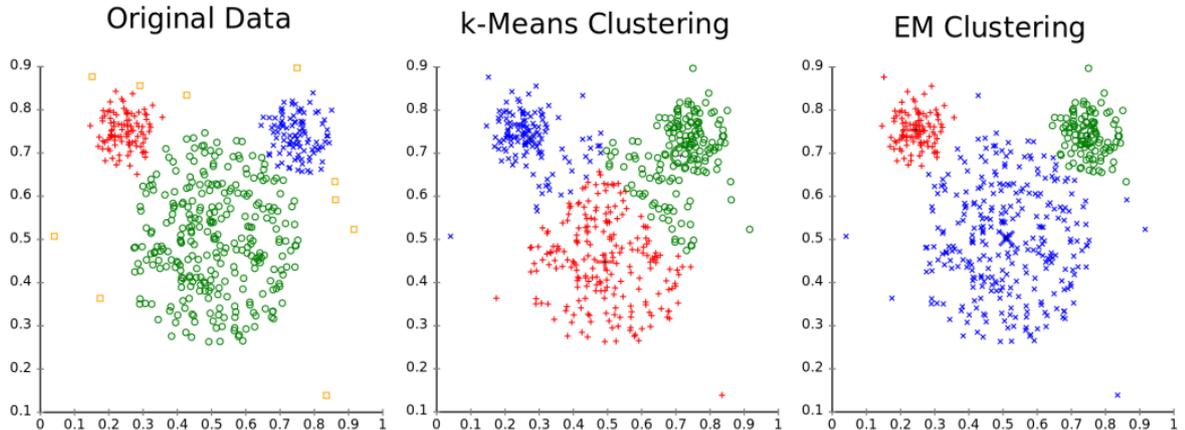
4. Evaluate the log likelihood $\ln p(x|\mu, \pi, \Sigma) = \sum_{n=1}^N \ln \left[\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right]$ and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

Please read section 9.2 of Bishop.





1. consider a dataset clustered by K -means and GMM..



2. For the GMM clustering, the most probable cluster for each point has been labeled.
3. K -means, unlike GMM, tends to learn equi-sized clusters.
4. In what situation, the results of GMM is equivalent to the results of K -means? (do it.)

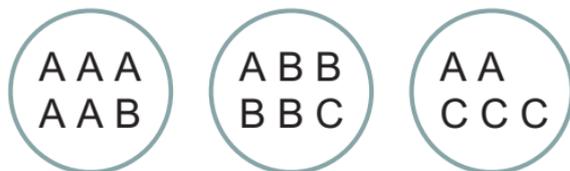
Cluster validation and assessment



- ▶ How good is the clustering generated by a method?
- ▶ How can we compare the clusterings generated by different methods?
- ▶ Clustering is an unsupervised learning technique and it is hard to evaluate the quality of the output of any given method.
- ▶ If we use probabilistic models, we can always evaluate the likelihood of a test set, but this has two drawbacks:
 1. It does not directly assess any clustering that is discovered by the model.
 2. It does not apply to non-probabilistic methods.
- ▶ We discuss some performance measures not based on likelihood.
- ▶ The goal of clustering is to assign points that are similar to the same cluster, and to ensure that points that are dissimilar are in different clusters.
- ▶ There are several ways of measuring these quantities
 1. **Internal criterion** : Typical objective functions in clustering formalize the goal of attaining high intra-cluster similarity and low inter-cluster similarity. But good scores on an internal criterion do not necessarily translate into good effectiveness in an application. An alternative to internal criteria is direct evaluation in the application of interest.
 2. **External criterion** : Suppose we have labels for each object. Then we can compare the clustering with the labels using various metrics. We will use some of these metrics later, when we compare clustering methods.



- ▶ Purity is a simple and transparent evaluation measure. Consider the following clustering.



- ▶ Let N_{ij} be the number of objects in cluster i that belongs to class j and $N_i = \sum_{j=1}^C N_{ij}$ be the total number of objects in cluster i .
- ▶ We define purity of cluster i as $p_i \triangleq \max_j \left(\frac{N_{ij}}{N_i} \right)$, and the overall purity of a clustering as

$$purity \triangleq \sum_i \frac{N_i}{N} p_i.$$

- ▶ For the above figure, the purity is

$$\frac{6}{17} \frac{5}{6} + \frac{6}{17} \frac{4}{6} + \frac{5}{17} \frac{3}{5} = \frac{5 + 4 + 3}{17} = 0.71$$

- ▶ Bad clusterings have purity values close to **0**, a perfect clustering has a purity of **1**.
- ▶ High purity is easy to achieve when the number of clusters is large. In particular, purity is **1** if each point gets its own cluster. Thus, we cannot use purity to trade off the quality of clustering against the number of clusters.



- ▶ Let $U = \{u_1, \dots, u_R\}$ and $V = \{v_1, \dots, v_C\}$ be two different (**flat**) clustering of N data points.
- ▶ For example, U might be the **estimated clustering** and V is **reference clustering** derived from the class labels.
- ▶ Define a 2×2 contingency table, containing the following numbers:
 1. **TP** is the number of pairs that are in the same cluster in both U and V (true positives);
 2. **TN** is the number of pairs that are in different clusters in both U and V (true negatives);
 3. **FN** is the number of pairs that are in different clusters in U but the same cluster in V (false negatives);
 4. **FP** is the number of pairs that are in the same cluster in U but different clusters in V (false positives).

- ▶ Rand index is defined as

$$RI \triangleq \frac{TP + TN}{TP + FP + FN + TN}$$

Rand index can be interpreted as the fraction of clustering decisions that are correct. Clearly $RI \in [0, 1]$.



- ▶ Consider the following clustering



- ▶ The three clusters contain 6, 6 and 5 points, so we have

$$TP + FP = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40.$$

- ▶ The number of true positives

$$TP = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20.$$

- ▶ Then $FP = 40 - 20 = 20$. Similarly, $FN = 24$ and $TN = 72$.

- ▶ Hence Rand index

$$RI = \frac{20 + 72}{20 + 20 + 24 + 72} = 0.68.$$

- ▶ Rand index only achieves its lower bound of 0 if $TP = TN = 0$, which is a rare event. We can define an adjusted Rand index

$$ARI \triangleq \frac{\text{index} - \mathbb{E}[\text{index}]}{\max \text{index} - \mathbb{E}[\text{index}]}.$$



- ▶ Another measure of cluster quality is computing mutual information between U and V .
- ▶ $P_{UV}(i, j) = \frac{|u_i \cap v_j|}{N}$ is the probability that a randomly chosen object belongs to cluster u_i in U and v_j in V .
- ▶ $P_U(i) = \frac{|u_i|}{N}$ is the probability that a randomly chosen object belongs to cluster u_i in U .
- ▶ $P_V(j) = \frac{|v_j|}{N}$ is the probability that a randomly chosen object belongs to cluster v_j in V .
- ▶ Then mutual information is defined

$$\mathbb{I}(U, V) \triangleq \sum_{i=1}^R \sum_{j=1}^C P_{UV}(i, j) \log \frac{P_{UV}(i, j)}{P_U(i)P_V(j)}$$

- ▶ This lies between 0 and $\min\{\mathbb{H}(U), \mathbb{H}(V)\}$.
- ▶ The maximum value can be achieved by using a lots of small clusters, which have low entropy.
- ▶ To compensate this, we can use **normalized mutual information (NMI)**

$$NMI(U, V) \triangleq \frac{\mathbb{I}(U, V)}{\frac{1}{2}[\mathbb{H}(U) + \mathbb{H}(V)]}$$

- ▶ This lies between 0 and 1.

Please read section 25.1 of Murphy.

Reading



1. Chapter 9 of [Pattern Recognition and Machine Learning Book](#) (Bishop 2006).
2. Sections 11.2.3 & 1.4 & 25.1 & 25.5 of [Machine Learning: A probabilistic perspective](#) (Murphy 2012).
3. Chapter 21 of [Probabilistic Machine Learning: An introduction](#) (Murphy 2022).



-  Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.
-  Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
-  – (2022). *Probabilistic Machine Learning: An introduction*. The MIT Press.

