Lecture 13: Unsupervised Learning: Clustering and Mixture Models

Readings: ISL (Ch. 12), PRML[1], ESL (Ch. 14); code

Topics: Unsupervised learning: overview, clustering and dissimilarity measures, K-means clustering, hierarchical clustering, Gaussian mixture models, Expectation-Maximization (EM) for GMMs, relationship between GMMs and K-means

1 Unsupervised Learning: Overview

Given dataset $\{x_i\}_{i=1}^N$ without labels, where $x_i \in \mathbb{R}^d$ (equivalently, given data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$), the main goal is to discover structure or learn properties of the underlying data distribution.

Key differences from supervised learning:

- No labels or responses are available to guide learning.
- We are often not directly interested in prediction.
- The goal is more subjective than in supervised learning.
- Success depends on interpretability and downstream utility.
- Unsupervised learning is often used for exploratory data analysis, visualization, and preprocessing.

Main tasks in unsupervised learning (we focus on the first one today):

1. **Clustering**: partition data into groups with similar characteristics.
2. **Dimensionality Reduction**: find a low-dimensional representation that retains key information.
3. **Density Estimation**: model the distribution from which data are sampled.
4. **Generative Modeling**: build models that can produce new samples from the learned distribution.

2 Clustering

Definition 2.1 (Clustering)**.** Partition N observations into K groups (clusters) such that observations within each cluster are more similar to each other than to observations in other clusters.

Clustering requires specification of:

- a **dissimilarity measure** $d(x_i, x_{i'})$ between observations,

[1] In addition to ISL and ESL, we use Bishop's PRML as our reference for unsupervised learning, which provides more in-depth treatment of core methods with a unified probabilistic framework. ESL Ch. 14 covers a broader range of topics but with less depth on the fundamentals we emphasize. Also, note that Ch. 15 in a modern version of PRML covers similar material; see `https://www.bishopbook.com/`.

- a **clustering algorithm** defining how dissimilarities are used,
- the **number of clusters** K, which is often unknown.

Common dissimilarity measures:

- Euclidean (ℓ_2) distance:
$$d(x_i, x_{i'}) = \|x_i - x_{i'}\|_2$$
- Manhattan (ℓ_1) distance:
$$d(x_i, x_{i'}) = \|x_i - x_{i'}\|_1$$
- Correlation-based dissimilarity:
$$d(x_i, x_{i'}) = 1 - \text{corr}(x_i, x_{i'})$$

There are many different clustering methods. We focus on K-means clustering and Gaussian mixture models.

3 K-Means Clustering

3.1 Intuitive Formulation

Definition 3.1 (K-Means Clustering)**.** Given N observations $\{x_i\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$, the goal of **K-means clustering** is to partition the data into K distinct, non-overlapping clusters $C_1, \ldots, C_K$ that minimize the total within-cluster variation:

$$W(C) = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - m_k\|_2^2,$$

where

$$m_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

is the mean (centroid) of cluster k.

💡 Each cluster has a "center," and assignments are chosen to minimize within-cluster variance.

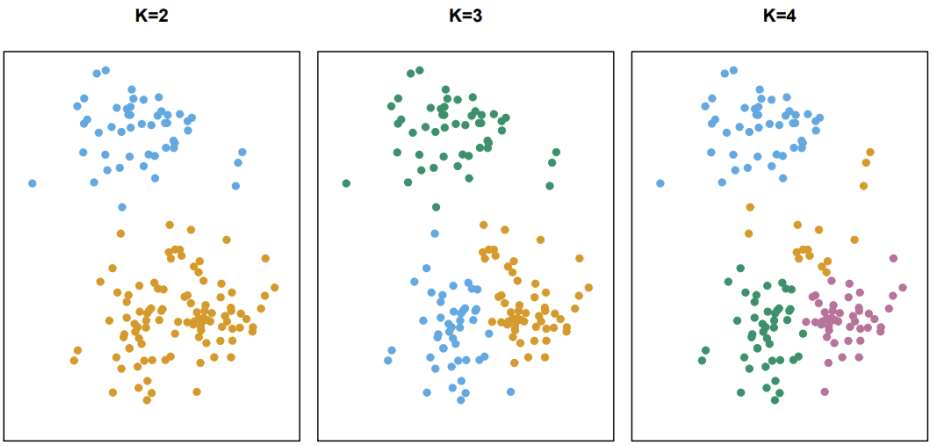

FIGURE 12.7. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.*

3.2 Assignment Variables and Objective

This is a difficult problem to solve globally since there are almost K^N possible assignments. A simple iterative algorithm, however, can be shown to provide a local optimum.

Let us introduce some notation:

- **Cluster representatives (centers):**

$$M = [m_1 \mid \cdots \mid m_K] \in \mathbb{R}^{d \times K},$$

  where $m_k \in \mathbb{R}^d$ is the center of cluster k.

- **Assignment variables:**

$$r_{ik} = \begin{cases} 1, & \text{if } x_i \text{ is assigned to cluster } k, \\ 0, & \text{otherwise,} \end{cases} \qquad \text{with } \sum_{k=1}^{K} r_{ik} = 1.$$

The matrix $R = (r_{ik}) \in \{0,1\}^{N \times K}$ encodes cluster assignments using a **1-of-K** coding scheme.

The K-means objective, also called the **distortion function**, is

$$J(R, M) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|x_i - m_k\|_2^2.$$

The optimization problem is

$$\min_{R,M} J(R, M) \quad \text{s.t. } r_{ik} \in \{0,1\}, \quad \sum_{k=1}^{K} r_{ik} = 1.$$

3.3 Coordinate Descent Interpretation

The optimization of $J(R, M)$ is intractable globally, but it can be minimized iteratively by alternating two steps.

(1) Assignment step (update R). Suppose M is fixed. Then each point is assigned to the nearest center:

$$r_{ik} = \begin{cases} 1, & \text{if } k = \arg\min_j \|x_i - m_j\|_2^2, \\ 0, & \text{otherwise.} \end{cases}$$

Ties are broken arbitrarily.

(2) Update step (update M). Given fixed assignments R, minimize $J(R, M)$ with respect to M:

$$\nabla_{m_k} J(R, M) = 2 \sum_{i=1}^{N} r_{ik}(m_k - x_i) = 0,$$

which yields

$$m_k = \frac{\sum_{i=1}^{N} r_{ik} x_i}{\sum_{i=1}^{N} r_{ik}}.$$

Thus each cluster center is the mean of all points assigned to that cluster.

3.4 Lloyd's Algorithm

K-Means (Lloyd's Algorithm)

1. **Input:** Data $\{x_i\}_{i=1}^N$, number of clusters K, stopping criterion.
2. **Initialize:** Choose initial centers $M \in \mathbb{R}^{d \times K}$, for example by sampling K data points.
3. Repeat until convergence:
   (a) **Assignment step:** For each $i = 1, \ldots, N$, assign x_i to its nearest center:

$$r_{ik} = \begin{cases} 1, & \text{if } k = \arg\min_j \|x_i - m_j\|_2^2, \\ 0, & \text{otherwise.} \end{cases}$$

   (b) **Update step:** For each $k = 1, \ldots, K$, recompute the cluster mean:

$$m_k = \frac{\sum_{i=1}^{N} r_{ik} x_i}{\sum_{i=1}^{N} r_{ik}}.$$

4. **Return:** centers M and assignments R.

The complexity is $O(NKdT)$, where T is the number of iterations, which is typically small.

3.5 Connecting the Two Formulations

The intuitive and matrix formulations are equivalent.

Define the cluster set representation:

$$C_k = \{\, i : r_{ik} = 1 \,\} \subseteq \{1, \ldots, N\}.$$

Then

$$m_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i, \qquad J(R, M) = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - m_k\|_2^2.$$

Summary:

- The intuitive view uses the cluster sets C_k.
- The matrix view uses the binary assignment matrix R and center matrix M.
- Both lead to the same objective and the same iterative algorithm.

3.6 Properties and Initialization

Properties:

- K-means converges to a local minimum.
- It is not guaranteed to find the global minimum.
- It is sensitive to initialization.

Initialization strategies:

1. **Random initialization:** select K data points at random.
2. **K-means++:** choose the first center randomly, then select each subsequent center with probability proportional to its squared distance from the nearest existing center.
3. **Multiple restarts:** run K-means multiple times and keep the best solution.

3.7 Choosing K

Common approaches:

- **Elbow method:** plot $W(C)$ versus K; choose the elbow point.
- **Gap statistic:** compare $\log W(C)$ with the value obtained from reference data.
- **Silhouette analysis:** choose K that maximizes the average silhouette score.
- **Domain knowledge:** often the most meaningful guide.

⚠ No universally best method exists. In practice, try several values of K and compare results for consistency and interpretability.

3.8 Hierarchical Clustering

Unlike K-means, hierarchical clustering does not require us to commit to a single value of K in advance. Instead, it produces a hierarchy of nested clusters, visualized as a **dendrogram**, which lets us inspect clusterings at multiple resolutions.

On the choice of dissimilarity measure:

- **Euclidean distance**: compares magnitudes.
- **Correlation distance**: compares shapes or patterns.

Scaling matters for clustering methods. Variables with larger variance can dominate. For standardized data, squared Euclidean distance and correlation distance can be shown to be equivalent; see Exercise 13.1.

💡 The choice of distance and scaling should match what "similarity" means in your data.

3.9 Practical Issues in Clustering

- Small design choices can have a major impact: standardization, distance measure, number of clusters.
- Validation is challenging; there is no universally accepted criterion.
- Clustering is sensitive to perturbations in the data.
- Raw and scaled variables can lead to very different results.
- A good practice is to try multiple parameter choices and compare consistent patterns.
- Clustering results should be viewed as exploratory, not absolute truth.

⚠ Since all observations are forced into clusters, outliers can distort the result. Mixture models provide an attractive probabilistic alternative.

4 Gaussian Mixture Models

4.1 Definition

A Gaussian mixture model (GMM) is a probabilistic soft alternative to K-means. Instead of assigning each point to exactly one cluster, it assigns each point a probability distribution over clusters.

Definition 4.1 (Gaussian Mixture Model)**.** Model data as coming from a mixture of K Gaussians:

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

where $\pi_k \geq 0$, $\sum_{k=1}^{K} \pi_k = 1$, and

- $\mu_k \in \mathbb{R}^d$ are component means,
- $\Sigma_k \in \mathbb{R}^{d \times d}$ are positive definite covariance matrices,
- $$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$
  is the Gaussian density.

GMMs are more flexible than K-means because the covariance matrices allow for elliptical clusters of varying size and orientation.

4.2 Latent Variable Interpretation

For each data point x_i, introduce a discrete latent variable $z_i \in \{1, \ldots, K\}$ indicating which component generated it.

- The prior probability of selecting component k is

$$p(z_i = k) = \pi_k.$$

- The conditional density of observing x_i given $z_i = k$ is

$$p(x_i \mid z_i = k) = \mathcal{N}(x_i; \mu_k, \Sigma_k).$$

- Hence the marginal density is

$$p(x_i) = \sum_{k=1}^{K} p(z_i = k)p(x_i \mid z_i = k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k).$$

Equivalently, represent z_i by a one-hot vector $r_i \in \{0,1\}^K$, where $\sum_{k=1}^{K} r_{ik} = 1$. Then

$$p(r_i) = \prod_{j=1}^{K} \pi_j^{r_{ij}}, \qquad p(x_i \mid r_i) = \prod_{j=1}^{K} \mathcal{N}(x_i; \mu_j, \Sigma_j)^{r_{ij}}.$$

4.3 Posterior Probabilities and Responsibilities

Using Bayes' rule, the posterior probability that point x_i came from component k is

$$\gamma_{ik} := p(z_i = k \mid x_i) = \frac{\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}.$$

These are called the **responsibilities**.

What γ_{ik} means:

- It is a soft assignment: $\gamma_{ik} \in [0,1]$ and $\sum_{k=1}^{K} \gamma_{ik} = 1$.
- Mathematically, $\gamma_{ik} = \mathbb{E}[r_{ik} \mid x_i]$.
- In contrast, K-means uses hard assignments $r_{ik} \in \{0,1\}$.
- If cluster k has high density at x_i and a large mixing weight π_k, then γ_{ik} is large.

Side remark. GMMs can approximate any continuous density arbitrarily well given enough mixture components. In this sense they are universal approximators of densities.

5 Expectation-Maximization for GMMs

5.1 Likelihood and Optimization Problem

Given i.i.d. data $\{x_i\}_{i=1}^{N}$, we seek parameters

$$\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$$

that maximize the log-likelihood

$$\log p(\mathbf{X} \mid \theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k) \right).$$

Challenge: The latent assignments z_i are unknown. Moreover, the optimal parameters and the responsibilities are interdependent:

- to compute γ_{ik}, we need θ,
- to update θ, we need γ_{ik}.

5.2 EM Idea

The Expectation-Maximization (EM) algorithm breaks the optimization into two alternating steps:

- **E-step (Expectation):** Given current parameters θ^{old}, compute the expected latent assignments:
$$\gamma_{ik} = p(z_i = k \mid x_i, \theta^{\text{old}}).$$
- **M-step (Maximization):** Given these soft assignments, update parameters by maximizing the expected complete-data log-likelihood.

This is analogous in spirit to K-means, except assignments are soft rather than hard.

5.3 EM Algorithm for GMMs

EM for GMM

1. Initialize $\pi_k = 1/K$ and choose initial $\{\mu_k, \Sigma_k\}$.
2. Repeat until convergence:
   (a) **E-step:** Compute responsibilities
   $$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}.$$
   (b) **M-step:** Define
   $$N_k = \sum_{i=1}^{N} \gamma_{ik}.$$
   Then update
   $$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_i,$$
   $$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T,$$
   $$\pi_k = \frac{N_k}{N}.$$
3. Return the means, covariances, mixture weights, and responsibilities.

EM guarantees a monotonic increase in the log-likelihood at each iteration.

5.4 Initialization Strategies

Different ways to initialize the EM algorithm:

- **Random**: choose K data points randomly as μ_k, and set $\Sigma_k = \text{Cov}(\mathbf{X})$.
- **K-means**: run K-means first, then use the resulting centers and within-cluster covariances.
- **Multiple restarts**: run EM several times and keep the solution with the highest log-likelihood.

Remark 5.1. More generally, EM is a broad class of algorithms for maximum likelihood or MAP estimation in latent-variable models.

5.5 Derivation of M-Step Updates

Mean update. Setting the derivative with respect to μ_k to zero yields

$$\sum_{i=1}^{N} \gamma_{ik} \Sigma_k^{-1}(x_i - \mu_k) = 0,$$

so

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} x_i}{\sum_{i=1}^{N} \gamma_{ik}} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_i.$$

Thus μ_k is a weighted average of the data, where the weights are the responsibilities.

Covariance update. Maximizing with respect to Σ_k gives the weighted sample covariance:

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T.$$

Mixing weight update. To enforce $\sum_k \pi_k = 1$, introduce a Lagrange multiplier and maximize

$$\mathcal{L}(\pi, \lambda) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}_{ik} \right) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right),$$

where $\mathcal{N}_{ik} = \mathcal{N}(x_i; \mu_k, \Sigma_k)$.

Solving yields

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

So the mixture weight of component k is the proportion of the effective number of points assigned to that cluster.

The update equations are interdependent, which is why the EM procedure must iterate.

5.6 EM vs SGD for GMM Parameter Estimation

The EM algorithm performs coordinate ascent on the log-likelihood

$$L(\theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k) \right).$$

An alternative is to use stochastic gradient ascent:

$$\theta \leftarrow \theta + \eta \nabla_\theta L(\theta).$$

Comparison:

- EM uses closed-form updates and ensures monotonic likelihood increase.
- SGD scales better to very large or streaming datasets.

- EM is typically faster for classical GMM clustering.
- SGD is useful when EM-style updates are intractable.

💡 EM is analytical and monotonic, but full-batch-based. SGD is scalable and flexible, but approximate.

6 Relationship Between GMM and K-Means

K-means is a special case of GMM where assignments become hard.

Suppose all covariance matrices are fixed and spherical:

$$\Sigma_k = \sigma^2 I.$$

Then

$$\gamma_{ik} = \frac{\pi_k \exp\left(-\frac{1}{2\sigma^2}\|x_i - \mu_k\|^2\right)}{\sum_{j=1}^{K} \pi_j \exp\left(-\frac{1}{2\sigma^2}\|x_i - \mu_j\|^2\right)}.$$

As $\sigma^2 \to 0$, the exponential term corresponding to the smallest squared distance dominates. Hence

$$\gamma_{ik} \to \begin{cases} 1, & \text{if } k = \arg\min_j \|x_i - \mu_j\|^2, \\ 0, & \text{otherwise.} \end{cases}$$

So the soft assignments become hard assignments, and the mean update becomes the K-means centroid update.

7 GMM vs K-Means for Clustering

⚠ In a standard GMM, if a component collapses onto a single data point, its covariance can go to zero and the likelihood can become arbitrarily large. This is a failure mode not present in K-means.

Summary comparison:

Property	**K-means**	**GMM**
Assignment	Hard (0/1)	Soft (probabilities)
Cluster shape	Spherical	Elliptical (general)
Cluster size	Similar	Can vary
Probabilistic	No	Yes
Complexity[2]	$O(NKdT)$	$O(NKd^2T)$
Speed	Fast	Slower
Uncertainty	No	Yes (γ_{ik})

💡 Try the experimental exercises to better understand how these two clustering methods differ in practice.

8 Exercises

1. Correlation vs. Euclidean Distance for Standardized Data.

For centered data where each feature has zero mean, define feature-wise standardized observations:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sigma_j}, \qquad \sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2.$$

(a) Show that

$$\frac{1}{N}\sum_{i=1}^{N} \tilde{x}_{ij}^2 = 1 \quad \text{for all } j.$$

(b) Define the sample correlation

$$\rho(x_i, x_k) = \frac{1}{d}\sum_{j=1}^{d} \tilde{x}_{ij}\tilde{x}_{kj}.$$

Show that

$$\|\tilde{x}_i - \tilde{x}_k\|^2 = 2d(1 - \rho(x_i, x_k))$$

when $\|\tilde{x}_i\|^2 = \|\tilde{x}_k\|^2 = d$.

(c) What does this imply for clustering based on correlation versus squared Euclidean distance?

(d) Would a similar relationship hold for ℓ_1 distance? Explain briefly.

2. K-Means Algorithm Convergence.

Consider K-means clustering with objective

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\|x_i - m_k\|^2.$$

(a) Show that for fixed assignments $\{r_{ik}\}$, the objective is minimized by

$$m_k^* = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}.$$

(b) Argue that for fixed centroids $\{m_k\}$, the reassignment step always produces a non-increasing value of J.

(c) Using (a) and (b), together with the fact that there are only finitely many assignments, prove that K-means converges in finitely many iterations.

3. EM for Bayesian Linear Regression Hyperparameters.

Consider Bayesian linear regression with prior

$$p(w \mid \alpha) = \mathcal{N}(0, \alpha^{-1} I_d)$$

and Gaussian noise precision β. The E-step computes the posterior

$$q(w) = p(w \mid y, \alpha^{\text{old}}, \beta^{\text{old}}) = \mathcal{N}(m_N, S_N).$$

The expected complete-data log likelihood is

$$\begin{aligned}\mathbb{E}_q[\log p(y, w \mid \alpha, \beta)] =& \frac{d}{2}\log\left(\frac{\alpha}{2\pi}\right) - \frac{\alpha}{2}\mathbb{E}_q[w^T w] \\ &+ \frac{N}{2}\log\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2}\sum_{n=1}^{N}\mathbb{E}_q[(y_n - w^T\phi_n)^2].\end{aligned}$$

(a) Show that

$$\mathbb{E}_q[w^T w] = m_N^T m_N + \text{tr}(S_N).$$

(b) Show that maximizing with respect to α gives

$$\alpha^{\text{new}} = \frac{d}{m_N^T m_N + \text{tr}(S_N)}.$$

(c) Derive the analogous re-estimation equation for β.

4. Deriving the Covariance Update in the EM Algorithm.

In a Gaussian mixture model, the expected complete-data log-likelihood is

$$Q(\{\Sigma_k\}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\left[\log|\Sigma_k| + (x_i - \mu_k)^T\Sigma_k^{-1}(x_i - \mu_k)\right].$$

(a) In the scalar case $\Sigma_k = \sigma_k^2$, show that

$$\sigma_k^{2,\text{new}} = \frac{1}{N_k}\sum_i \gamma_{ik}(x_i - \mu_k)^2, \qquad N_k = \sum_i \gamma_{ik}.$$

(b) Using matrix derivative identities, derive

$$\Sigma_k^{\text{new}} = \frac{1}{N_k}\sum_{i=1}^{N}\gamma_{ik}(x_i - \mu_k)(x_i - \mu_k)^T.$$

(c) Interpret the role of γ_{ik} and contrast it with K-means.

[Experiment] For this entire exercise, use a single simulated dataset.

- Generate $N = 60$ observations in $\mathbb{R}^2$, with 20 points per class.
- Use means

  $$\mu_1 = (\delta, 0), \qquad \mu_2 = (0, \delta), \qquad \mu_3 = (-\delta, 0),$$

  with $\delta = 6$.

- Use covariance matrices

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}, \quad \Sigma_3 = \begin{bmatrix} 4 & 1.5 \\ 1.5 & 1 \end{bmatrix}.$$

- Store the data matrix $\mathbf{X}$ and the true labels $y \in \{1, 2, 3\}^{60}$.

Provide plots to visualize your results.

(a) Run K-means with $K = 3$. Compare the assignments to the true labels.

(b) Fit a 3-component GMM with full covariances. Compare the hard assignments induced by the responsibilities to the K-means result.

(c) Run K-means with $K = 2$ and $K = 4$. Describe what gets merged or split.

(d) Repeat (c) using a GMM. Does the pattern differ?

(e) Standardize each feature of $\mathbf{X}$ to obtain $\tilde{\mathbf{X}}$. Run K-means with $K = 3$ on $\tilde{\mathbf{X}}$. Compare with the unscaled result.

(f) Fit a 3-component GMM to the standardized data. Compare with the unscaled GMM result.

(g) Demonstrate the GMM covariance-collapse failure mode:

- Generate 15 points approximately on a line.
- Fit a GMM with $K = 3$.
- Inspect the eigenvalues of the fitted covariances.
- Regularize by adding λI to each covariance matrix, refit, and compare.