Lecture 08

Soft Clustering

Soft clustering (also known as fuzzy clustering) is a type of clustering technique where data points can belong to multiple clusters at once, each with a certain degree of membership (between 0 and 1), rather than belonging to just one cluster as in hard clustering (like K-Means).

Example: A film like "Avatar" could belong 70% to the "Sci-Fi" cluster and 30% to the "Adventure" cluster. Unlike hard clustering (which would force it into only one cluster), soft clustering recognizes overlaps

Each data point x_i has a *membership value* u_{ij} for each cluster j, satisfying:

$$\sum_{j=1}^{k} u_{ij} = 1$$

where,

$u_{ij} \in [0,1]$ → degree of belonging of x_i to cluster j

k = number of clusters

Gaussian Mixture Model (GMM)

A Gaussian Mixture Model assumes that your data is generated from a mixture (combination) of several Gaussian (normal) distributions, each representing a cluster.

So instead of saying "each point belongs to exactly one cluster" (like K-Means), GMM says: *Each point belongs to all clusters with different probabilities*. That's exactly what makes it soft clustering.

GMM assumes data is generated as

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

Where,

- K : number of clusters
- π_k : weight (mixing coefficient) of cluster k, with $\Sigma_k \pi_k = 1$
- $\mathcal{N}(x \mid \mu_k, \Sigma_k)$: Gaussian distribution with mean μ_k and covariance Σ_k

Each cluster has:

- A **mean vector** $\mu_k \rightarrow center.$
- A **covariance matrix** $\Sigma_k \rightarrow shape/orientation.$
- A **weight** $\pi_k \rightarrow how\ large\ that\ cluster\ is.$

For each data point x_i, GMM computes responsibilities:

$$r_{ik} = P(cluster\ k \mid x_i)$$

These are probabilities that tell you how much cluster k "owns" data point x_i.

Training is done via the **Expectation-Maximization (EM)** algorithm:

1. **E-step:** Compute responsibilities $r_{ik} = P(k \mid x_i)$
2. **M-step:** Update parameters π_k, μ_k, Σ_k based on those responsibilities.

Repeat until convergence.

Question: *How many parameters does GMM have? and what are they?*
Answer: has three types of parameters:

1. Mixing coefficient (weight): π_k
2. Mean vector: μ_k
3. Covariance matrix: Σ_k

Question: *Number of free parameters?*
Answer: Total number of free parameters are:

$$(K-1) + KD + \frac{KD(D+1)}{2}$$

Where K = clusters, D = number of features.

Expectation-Maximization (EM)

If we take log of likelihood in normal MLE That log of a sum makes the expression nonlinear and not directly differentiable with respect to each parameter, so we can't solve it analytically. The EM algorithm is an iterative method that finds the MLE parameters when there are hidden variables here, the latent cluster assignments.

Step 1: Introduce hidden variables

Define a latent variable z_i for each data point, indicating which Gaussian generated it:

- $z_i = k$ means "point i came from cluster k"

Since we don't know z_i, EM alternates between estimating z_i (E-step) and updating parameters (M-step).

Step 2: The E-step (Expectation)

Compute responsibilities:

$$r_{ik} = P(z_i = k \mid x_i, \theta^{old})$$

using Bayes' rule:

$$r_{ik} = \frac{\pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}$$

Each r_{ik} is the probability that cluster k generated x_i.

Step 3: The M-step (Maximization)

Update parameters to maximize the expected complete log-likelihood using the responsibilities as soft counts.

- Updating mixing coefficients:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} r_{ik}$$

- Update means:

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

- Update covariances:

$$\Sigma_k = \frac{\sum_{i=1}^{N} r_{ik} (x_i - \mu_i)(x_i - \mu_i)^T}{\sum_{i=1}^{N} r_{ik}}$$

Then recompute responsibilities (E-step), and repeat until the log-likelihood converges.