CSC 246 Machine Learning | Lecture 10

Clustering

Lecturer: Jiaming Liang | *October 27, 2026*

Clustering is useful when we want to discover structure without knowing the correct groups in advance. For instance, a retailer might group customers by purchasing behavior, or a scientist might group measurements with similar patterns. Unlike classification, clustering has no observed class labels y_i to guide the learning process. The notion of similarity must come from the features and the model we choose.

1 K-means

Suppose we seek K clusters. We introduce a cluster assignment $c_i \in \{1, \dots, K\}$ for each observation and a center $\boldsymbol{\mu}_j \in \mathbb{R}^d$ for each cluster. K-means tries to make points close to their assigned centers by minimizing the sum of squared distances:

$$J(\mathbf{c}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K) = \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu}_{c_i}\|_2^2. \tag{1}$$

The number K is chosen in advance. The centers and assignments are unknown and must be learned together. This is not an easy joint optimization problem: the assignments are discrete, and different choices of clusters can lead to different solutions.

Rather than solve the joint problem directly, K-means alternates between two simple operations. Starting from K initial centers, it repeats:

1. *Assignment step:* with the centers fixed, assign each observation to its nearest center,

$$c_i \leftarrow \arg\min_{1 \le j \le K} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2.$$

2. *Center update:* with the assignments fixed, replace each nonempty cluster center by the average of its assigned points,

$$\boldsymbol{\mu}_j \leftarrow \frac{1}{|C_j|} \sum_{i \in C_j} \mathbf{x}_i, \qquad C_j = \{i : c_i = j\}.$$

We repeat until the assignments or centers stop changing (or until the change in the objective is sufficiently small). An empty cluster requires a separate convention, such as reinitializing its center.

In fact, K-means is an alternating minimization method for minimizing $J(\mathbf{c}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K)$ in (1). With fixed centers $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$,the assignment step minimizes each individual term $\|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2$

by choosing the nearest center. Then, with updated assignment c_i's, the contribution of cluster j to the objective $J(\mathbf{c}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$ is

$$f_j(\boldsymbol{\mu}_j) = \sum_{i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2.$$

Minimizing this objective for cluster j boils down to taking a gradient with respect to the center and setting it to zero

$$\nabla f_j(\boldsymbol{\mu}_j) = 2 \sum_{i \in C_j} (\boldsymbol{\mu}_j - \mathbf{x}_i) = 0, \qquad \boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{i \in C_j} \mathbf{x}_i.$$

Thus, neither step increases J. This guarantees that the sequence of objective values is nonincreasing, but it does not guarantee finding the globally best clustering.

Example 1. *Consider the one-dimensional observations* $0, 2, 7, 9$*, with* $K = 2$ *and initial centers* $\mu_1 = 0$ *and* $\mu_2 = 7$*. The nearest-center rule assigns* $\{0, 2\}$ *to cluster 1 and* $\{7, 9\}$ *to cluster 2. The updated centers are* $\mu_1 = 1$ *and* $\mu_2 = 8$*. Keeping these assignments, the squared-error objective decreases from* $J = 8$ *to* $J = 4$*. In this example the assignments then remain unchanged, so the algorithm has reached a fixed point.*

A practical limitation is that K-means makes hard assignments: each observation is placed in exactly one cluster, even if it is almost equidistant from two centers. Moreover, the squared Euclidean distance favors clusters with roughly round, similarly scaled shapes. To handle ambiguous assignments and more flexible cluster shapes, we turn to a probabilistic model.

2 Gaussian Mixture Models

A Gaussian mixture model (GMM) assumes that observations arise from a combination of K Gaussian distributions. The j-th component has a mean $\boldsymbol{\mu}_j \in \mathbb{R}^d$, a positive definite covariance matrix $\Sigma_j \in \mathbb{R}^{d \times d}$, and a mixing weight $\pi_j \geq 0$. The mixing weights satisfy $\sum_{j=1}^K \pi_j = 1$. The resulting density is

$$p(\mathbf{x}; \theta) = \sum_{j=1}^K \pi_j \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_j, \Sigma_j), \qquad \theta = \{(\pi_j, \boldsymbol{\mu}_j, \Sigma_j)\}_{j=1}^K.$$

Here $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma)$ is the multivariate Gaussian density

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma) = \frac{\exp\big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\big)}{(2\pi)^{d/2} (\det \Sigma)^{1/2}}.$$

The mean describes a component's location, while the covariance describes its spread and shape.

One way to interpret the model is to imagine a two-stage procedure for generating one observation. First, draw a hidden label $z \in \{1, \dots, K\}$ with $\Pr(z = j) = \pi_j$. Next, conditional on $z = j$, draw $\mathbf{x}$ from $\mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$. Because the label z is not observed, it is called a latent variable. Summing over its possible values recovers the mixture density:

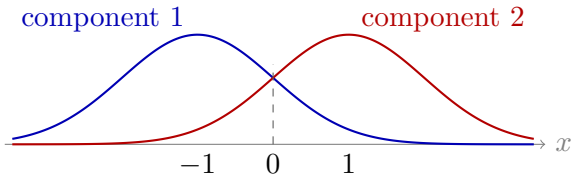
$$p(\mathbf{x}; \theta) = \sum_{j=1}^{K} \Pr(z = j)\, p(\mathbf{x} \mid z = j; \theta).$$

Clustering with a GMM means inferring the latent labels as well as learning the component parameters.

Suppose the parameters θ are currently known. For an observation $\mathbf{x}_i$, we want the posterior probability that it belongs to component j. Bayes' rule gives

$$\gamma_{ij} := \Pr(z_i = j \mid \mathbf{x}_i; \theta) = \frac{\pi_j \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_\ell, \Sigma_\ell)}.$$

For each point, $0 \le \gamma_{ij} \le 1$ and $\sum_{j=1}^{K} \gamma_{ij} = 1$. Unlike K-means, a point does not have to belong exclusively to a single cluster.



Two overlapping Gaussian components. A point near $x = 0$ has ambiguous membership.

Example 2. *Suppose $K = 2$, $\pi_1 = \pi_2 = 1/2$, and both components have variance 1, with means $\mu_1 = -1$ and $\mu_2 = 1$. A point at $x = 0$ has responsibilities $\gamma_1 = \gamma_2 = 1/2$ by symmetry. For a point at $x = 1$, the second component has greater density, and Bayes' rule gives*

$$\Pr(z = 2 \mid x = 1) = \frac{e^{0}}{e^{-2} + e^{0}} = \frac{1}{1 + e^{-2}} \approx 0.881.$$

This is strong evidence for component 2, but it is not absolute certainty.

We do not know the means, covariances, or mixing weights in advance. As in logistic regression, one natural approach is maximum likelihood estimation (MLE). Given independent observations $\mathbf{x}_1, \dots, \mathbf{x}_n$, the log-likelihood is

$$\ell(\theta) = \sum_{i=1}^{n} \log p(\mathbf{x}_i; \theta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{K} \pi_j \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j) \right).$$

The main difficulty is the logarithm of a sum. We cannot simply separate it into one likelihood term for each Gaussian. If the hidden labels z_i were actually observed, the log-likelihood would instead be

$$\sum_{i=1}^{n}\sum_{j=1}^{K}\mathbf{1}\{z_i=j\}\left[\log\pi_j+\log\mathcal{N}(\mathbf{x}_i\mid\boldsymbol{\mu}_j,\Sigma_j)\right],$$

which separates across components and would be much easier to optimize. This observation motivates the expectation-maximization (EM) method: use probabilities for the unknown labels, and then fit the parameters using those probabilities.

3 EM Method

Motivated by the observation above, we are interested in maximizing the following objective

$$F(q,\theta)=\sum_{i=1}^{n}\sum_{j=1}^{K}q_{ij}\log\frac{\pi_j\mathcal{N}(\mathbf{x}_i\mid\boldsymbol{\mu}_j,\Sigma_j)}{q_{ij}},$$

where $q_{ij}>0$ and $\sum_j q_{ij}=1$. Note that the concavity of the logarithm (Jensen's inequality) implies that for any q,

$$\ell(\theta)=\sum_{i=1}^{n}\log\left(\sum_{j=1}^{K}q_{ij}\frac{\pi_j\mathcal{N}(\mathbf{x}_i\mid\boldsymbol{\mu}_j,\Sigma_j)}{q_{ij}}\right)\geq\sum_{i=1}^{n}\sum_{j=1}^{K}q_{ij}\log\frac{\pi_j\mathcal{N}(\mathbf{x}_i\mid\boldsymbol{\mu}_j,\Sigma_j)}{q_{ij}}=F(q,\theta).\qquad(2)$$

So, $F(q,\theta)$ is a lower approximate of $\ell(\theta)$ that is easier to handle than $\ell(\theta)$.

EM is a method that maximizes $F(q,\theta)$ over q and θ alternatively as a way to approximately solving the original MLE $\max_\theta\ell(\theta)$. Specifically, EM alternates between estimating the latent labels and updating the parameters. Starting from initial parameters $\theta^{(0)}$, each iteration consists of an E-step and an M-step.

E-Step: estimate the latent labels. With the current parameters $\theta^{(t)}$, compute the posterior responsibilities for every observation and component:

$$\gamma_{ij}^{(t)}=\arg\max\left\{F(q,\theta^{(t)}):q_{ij}>0,\sum_j q_{ij}=1\right\}=\frac{\pi_j^{(t)}\,\mathcal{N}(\mathbf{x}_i\mid\boldsymbol{\mu}_j^{(t)},\Sigma_j^{(t)})}{\sum_{\ell=1}^{K}\pi_\ell^{(t)}\,\mathcal{N}(\mathbf{x}_i\mid\boldsymbol{\mu}_\ell^{(t)},\Sigma_\ell^{(t)})}.$$

Think of $\gamma_{ij}^{(t)}$ as a fractional assignment: x_i contributes weight $\gamma_{ij}^{(t)}$ to component j rather than a weight of exactly 0 or 1.

M-Step: update the mixture parameters θ. In the M-step, treat $\gamma_{ij}^{(t)}$ as fixed and update parameters $\theta^{(t+1)}=\arg\max_\theta F(\gamma^{(t)},\theta)$. Let $N_j^{(t)}=\sum_{i=1}^{n}\gamma_{ij}^{(t)}$ denote the effective number of

observations assigned to component j. The resulting updates are

$$\pi_j^{(t+1)} = \frac{N_j^{(t)}}{n},$$
$$\boldsymbol{\mu}_j^{(t+1)} = \frac{1}{N_j^{(t)}} \sum_{i=1}^{n} \gamma_{ij}^{(t)} \mathbf{x}_i,$$
$$\Sigma_j^{(t+1)} = \frac{1}{N_j^{(t)}} \sum_{i=1}^{n} \gamma_{ij}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_j^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_j^{(t+1)})^\top.$$

These are natural analogues of the ordinary sample proportions, means, and covariance matrices, but each observation has a fractional weight.

The algorithm then returns to the E-step using the new parameters and continues until the likelihood changes very little. EM is an iterative algorithm; in general, neither one iteration nor the final result is guaranteed to be the globally optimal mixture model.

We note that (2) is tight when $q = \gamma$. Indeed, denote

$$\gamma_{ij}(\theta) = \frac{p_{ij}(\theta)}{\sum_{k=1}^{K} p_{ik}(\theta)}.$$

Then, we have

$$F(\gamma(\theta), \theta) = \sum_{i,j} \gamma_{ij}(\theta) \log \frac{p_{ij}(\theta)}{\gamma_{ij}(\theta)} = \sum_{i,j} \gamma_{ij}(\theta) \log \left(\sum_k p_{ik}(\theta) \right) = \sum_i \log \left(\sum_k p_{ik}(\theta) \right) = \ell(\theta).$$

Now, recall that in the E-step, we have

$$F\left(\gamma^{(t)}, \theta^{(t)}\right) = \ell\left(\theta^{(t)}\right),$$

and in the M-step, we have

$$F\left(\gamma^{(t)}, \theta^{(t+1)}\right) \geq F\left(\gamma^{(t)}, \theta^{(t)}\right).$$

Putting together, we obtain

$$\ell\left(\theta^{(t+1)}\right) \geq F\left(\gamma^{(t)}, \theta^{(t+1)}\right) \geq F\left(\gamma^{(t)}, \theta^{(t)}\right) = \ell\left(\theta^{(t)}\right).$$

Hence, each EM iteration does not decrease the log-likelihood $\ell(\theta)$. Again, this monotonic improvement does not imply a global optimum.

4 Connection between K-means and EM

Both algorithms alternate between assigning observations to clusters and updating cluster parameters. The main distinction is that K-means uses hard assignments, while a GMM uses posterior

probabilities. In a particular case, the connection is even stronger. Suppose a GMM has equal mixing weights and all components share covariance $\sigma^2 I_d$. Then

$$\gamma_{ij} = \frac{\exp\bigl(-\|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2/(2\sigma^2)\bigr)}{\sum_{\ell=1}^{K} \exp\bigl(-\|\mathbf{x}_i - \boldsymbol{\mu}_\ell\|_2^2/(2\sigma^2)\bigr)}.$$

When σ^2 is small, a point assigns almost all its weight to its nearest center, provided the nearest center is unique. In the limit as $\sigma^2 \to 0$, the soft assignment becomes a hard nearest-center assignment, and the weighted mean update approaches the K-means center update. This explains why K-means is often described as a hard-clustering limit of Gaussian mixtures.