Recap: Soft Clustering with Gaussian Mixture Models

DS701 Session 7 — Mon Sep 28, 2026

Today’s plan

Knowledge-Check Review

KC 1: The generative story

A Gaussian mixture model is a generative story. Describe, in two steps, how the model says a single data point x_i is produced, and write down the resulting density $p(x_i \mid \theta)$. What are the parameters θ?

KC 2: Why EM?

In lecture 05 we found the MLE for a single Gaussian by taking the log-likelihood, differentiating, and solving in closed form (sample mean, sample variance). Why does the same approach not work for a mixture, and how do the two EM steps get around it? Say specifically what the E-step and the M-step each compute.

KC 3: "Just k-means with probabilities"?

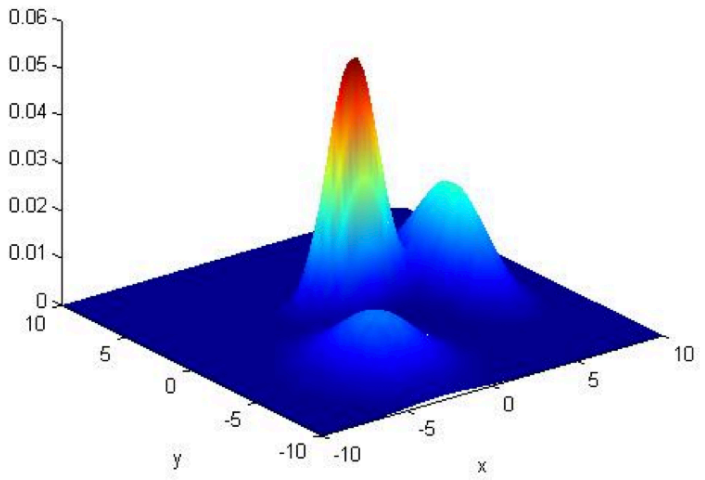
*A classmate says "a GMM is just k-means with probabilities attached." Name **two** ways that description is incomplete, and give one dataset the lecture showed where the difference visibly matters.*

Highlights

The model is a recipe for making data

For each of the n points, independently:

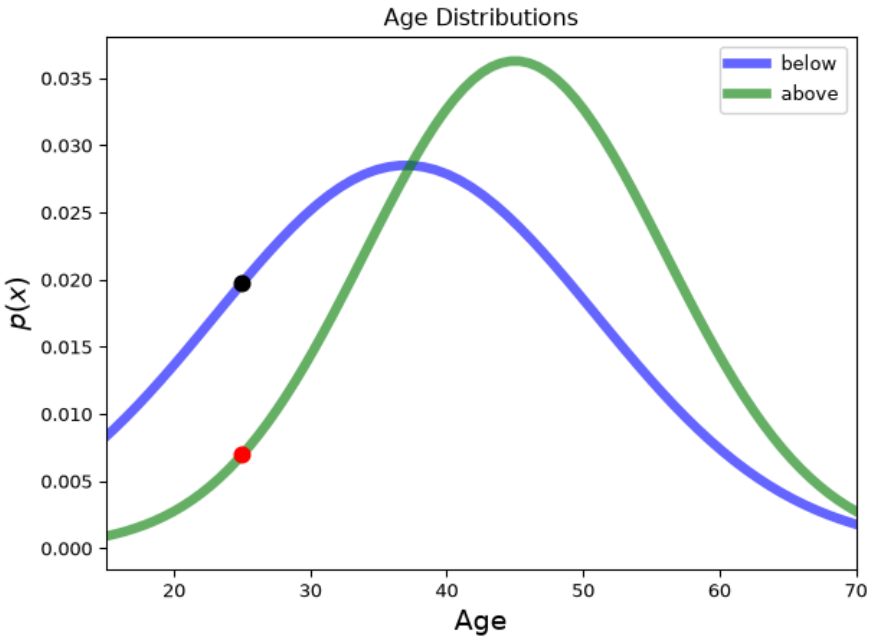
1. roll a K-sided die with face probabilities $\pi_1, \ldots, \pi_K$ — that is the **latent** label z_i;
2. draw x_i from the Gaussian for that face, $\mathcal{N}(\mu_{z_i}, \Sigma_{z_i})$.



E-step: Bayes' rule, one point at a time

Age 25 in the income example. Black dot: density under *below*; red dot: density under *above*.

$$P(\text{above} \mid 25) = \frac{\text{red} \cdot P(\text{above}}{\text{red} \cdot P(\text{above}) + \text{black}}$$



M-step: lecture 05, weighted

One Gaussian (lecture 05) — the MLE:

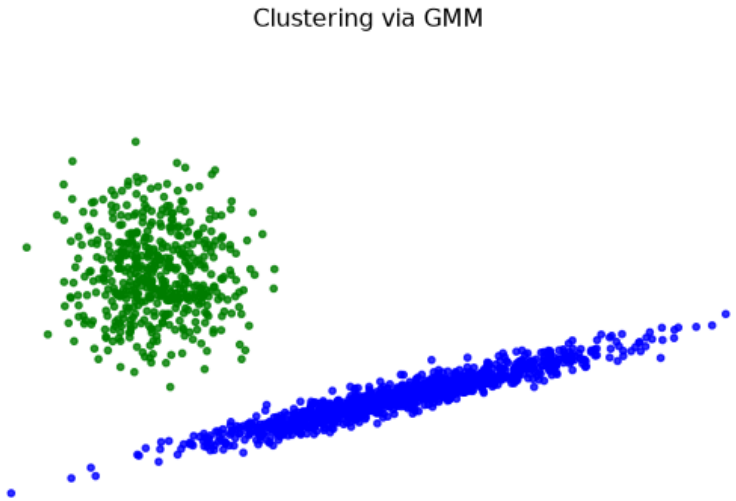
$$\bar{\mu} = \frac{1}{N}\sum_i x_i, \qquad \bar{\sigma}^2 = \frac{1}{N}\sum_i (x_i - \bar{\mu})^2$$

Component k of a mixture — with $N_k = \sum_i \gamma_{ik}$:

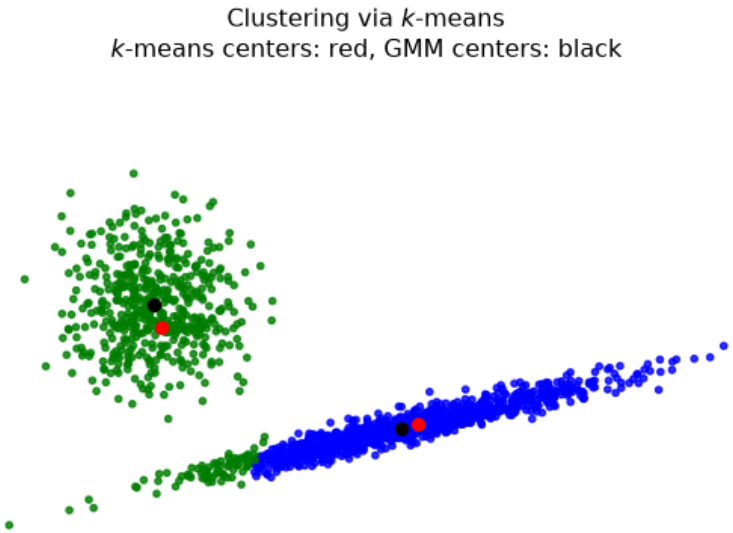
$$\mu_k = \frac{1}{N_k}\sum_i \gamma_{ik}\, x_i, \qquad \Sigma_k = \frac{1}{N_k}\sum_i \gamma_{ik}$$

Why iterate at all — and what you get for it

GMM vs. k-means, side by side



GMM, `covariance_type="full"`



k-means, $k = 2$

Choosing how much shape to allow

`covariance_type` is where the assumption becomes a dial (K components in d dimensions):

type	each Σ_k is	shape	covariance parameters
`spherical`	$\sigma_k^2 I$	circle, own radius	K
`diag`	diagonal	axis-aligned ellipse	Kd
`tied`	one shared Σ	same ellipse everywhere	$\sim d^2$
`full`	its own Σ_k	any ellipse	$\sim Kd^2$

In-Class Activity

Activity: EM by hand, then soft vs. hard — visualized

Goal: implement the E-step and M-step yourself on a 1-D two-component mixture, iterate while watching the log-likelihood climb, check against `sklearn.mixture.GaussianMixture` — then go to 2-D: responsibilities as color intensity next to k-means' straight boundaries, the four covariance types as fitted ellipses, and one point the GMM calls 50/50.

- Work in groups of 2–3.
- Parts marked (**autograded**) are submitted to Gradescope; the rest is participation.
- Staff will circulate — be ready to explain any part of your work.
- Dependencies: `numpy`, `scipy`, `matplotlib`, `scikit-learn` only.

Section A1: Open in Colab Open on GitHub

Section B1: Open in Colab Open on GitHub

Going deeper

Full lecture notes: Soft Clustering with Gaussian Mixture Models

- Soft Clustering — the income/age example and conditional probability
- MLE in One Slide — the lecture-05 callback the M-step is built on
- The Gaussian Mixture Model — the generative story and the mixture density
- The Log-Likelihood Challenge — the sum inside the log
- E-Step and M-Step — the two updates
- Iteration and Convergence and Convergence criteria
- k-means vs GMMs — soft vs. hard, explicit vs. implicit assumptions
- How many parameters are estimated? — `full` / `tied` / `diag` / `spherical`
- Appendices: MLE derivation, complete-data likelihood, deriving the E-step
- Next up: generalization — train/test discipline and overfitting (07).