Pre-class progress check

By now, you should have

1. Started thinking about your group project proposals (due 11.06 by 11.59pm)
2. Reviewed slides and code from classes 1-6
3. Finished quiz 7 (due at 8:59 AM today)

1. Reviewed slides and code from class 7
2. Started working on assignment 3

1. Started working on group project
2. Reviewed slides from class 8 (today)

University of Pittsburgh | School of Public Health

Unsupervised learning (I): Clustering (II)
- Gaussian mixture models
- EM algorithm

BIOST 2155

Class 8
October 23rd, 2026

CLASS 1

Mapping what we have learnt so far…

	Features (input)	Targets/ labels (output)	Task type	ML type	Metric
Prediction of brain volume from gestational age	✅	✅ (Numeric)	Regression	**Supervised learning**	*RMSE, R-squared*
Classification of heart failure using ultrasound or magnetic resonance imaging	✅	✅ (Categorical)	Classification		*Accuracy, Precision, Recall, AUC*
Patient health profiles using wearable device data	✅	🚫	Clustering	**Unsupervised learning**	*Silhouette score*
Compressing information from 30 features to study similarities/differences in benign and malignant breast tumors	✅	🚫	Dimensionality reduction		*Explained Variance*

CLASS 1

Patient health profile study using wearable devices

Use ML to identify patient health profiles based on their routine clinical measurements.

Context: Chronic conditions related to obesity and high blood pressure are a massive public health burden, leading to diseases like heart attack, stroke, and type 2 diabetes.
Challenge: Patients often have multiple risk factors. A manual review of every patient's data is inefficient for large-scale population health management.

By using a dataset of common patient measurements, we can group individuals and associate their combined metrics with different levels of health risk.

Body Mass Index (BMI): A measure of body fat calculated from height and weight.
A higher BMI is a well-established indicator for a range of chronic health issues, serving as a foundational metric for general health.

Systolic Blood Pressure (SBP): Measures the pressure in your arteries when your heart beats.
Elevated SBP is a direct and critical risk factor for cardiovascular events.

Why do we care?

Risk Stratification.
Targeted Interventions.
Proactive Healthcare.

CLASS 1

Patient health profile study using wearable devices



Group 1 (**High-Risk**): High BMI (e.g., 33-38) and high SBP (e.g., 150-165).
Group 2 (**Healthy**): Lower BMI (e.g., 20-24) and normal SBP (e.g., 110-125).
Group 3 (**At-Risk**): Moderate BMI (e.g., 27-31) and elevated SBP (e.g., 130-145).

Beyond generative models

No labels? No problem

Like classification, we want to find hidden common structures within our data.

Unlike classification, we don’t have any pre-existing labels to train our models.

We still want to identify (**hidden**) labels. Since we don't have labels, we'll treat them as latent (**hidden**) variables.

Roadmap:

1. Assume data is from a mixture of several Gaussian (normal) distributions, where each distribution represents a cluster.
2. Use the EM algorithm - an iterative method that (a) figures out which points belong to which cluster and (b) what each cluster looks like.

The deliverables

What do we get from GMMs?

1. **Cluster Assignments:** A final label for each data point, indicating which group it most likely belongs to.

2. **Probabilistic Information:** The probability of each point belonging to every cluster (e.g., Point X is 90% likely to be in group A, 10% likely to be in group B).

3. **Cluster Descriptions:** The characteristics (center, shape, and size) of each discovered group.

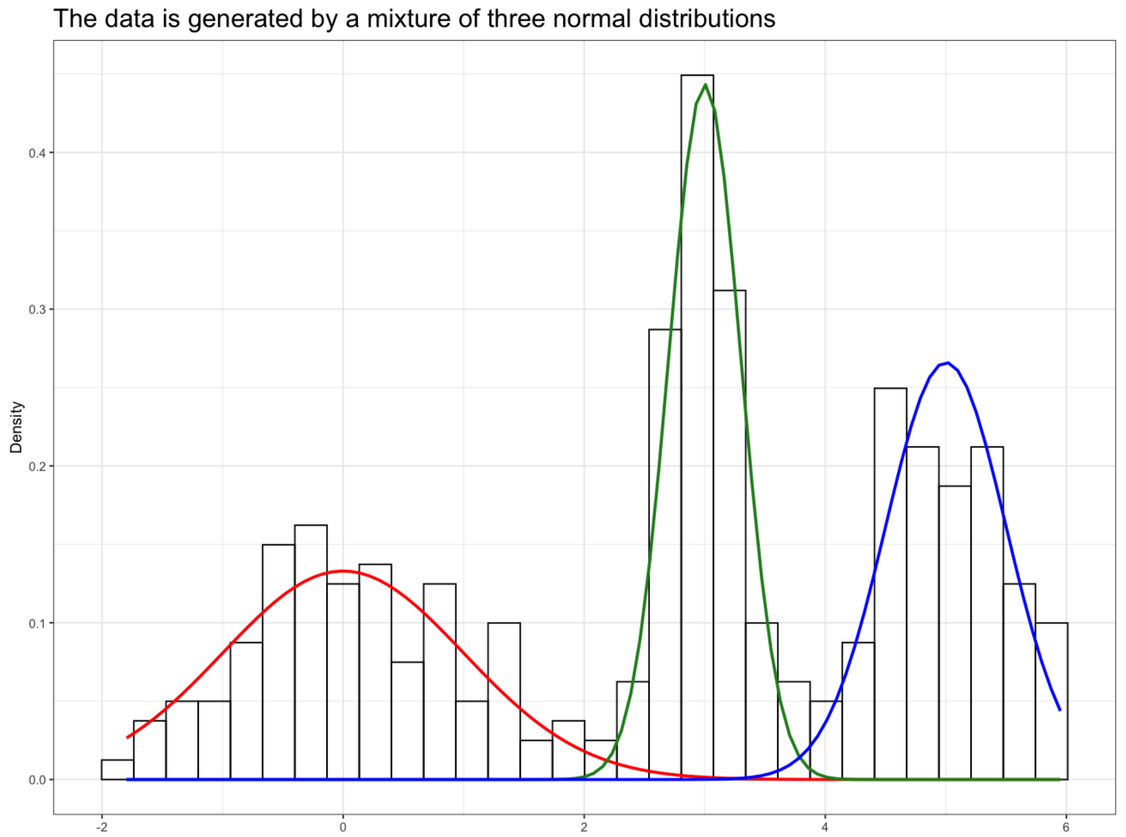
The Core Idea of GMMs

Visualizing GMMs for $p = 1$

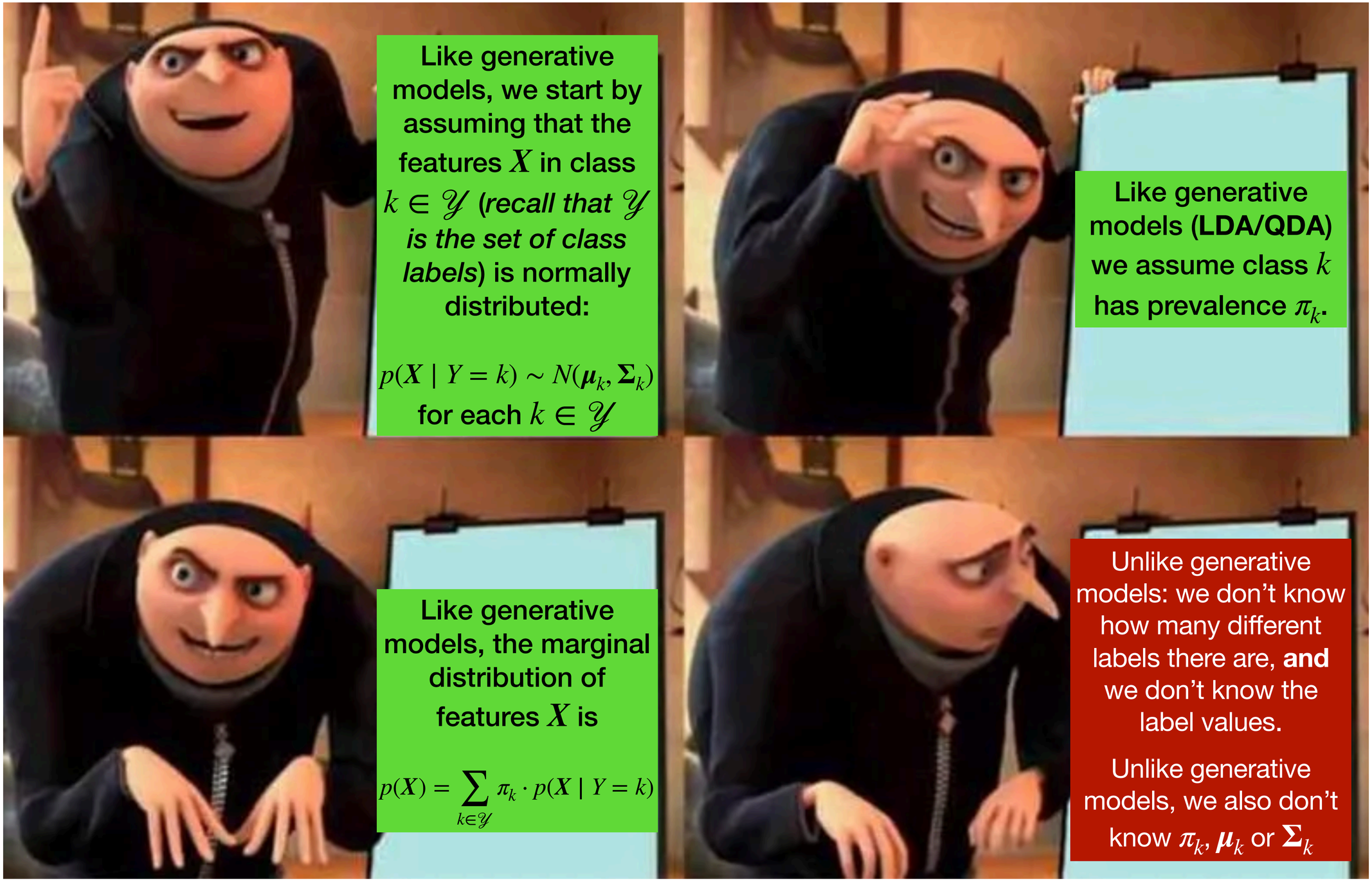
Core assumption: data is a mixture of several Gaussian (normal) distributions

Imagine a painter mixing a few primary colors to create all the colors on a canvas.

GMMs find the "primary" distributions (colors) that, when mixed together, create the overall distribution of our data.



Using GMMs, we try and identify the three curves and figure out which color (label) to assign to each observation


Like generative models, we start by assuming that the features X in class $k \in \mathcal{Y}$ (*recall that $\mathcal{Y}$ is the set of class labels*) is normally distributed:
$p(X \mid Y = k) \sim N(\mu_k, \Sigma_k)$
for each $k \in \mathcal{Y}$
Like generative models (**LDA/QDA**) we assume class k has prevalence π_k.
Like generative models, the marginal distribution of features X is
$p(X) = \sum_{k \in \mathcal{Y}} \pi_k \cdot p(X \mid Y = k)$
Unlike generative models: we don't know how many different labels there are, **and** we don't know the label values.
Unlike generative models, we also don't know π_k, μ_k or Σ_k

Plan: Assume Latent Labels & Use EM

We don't have labels, we'll treat them as latent (hidden) variables

- Our goal is to find hidden structure in data. We believe there are distinct groups, but we don't have any pre-existing labels to tell us which point belongs to which group.
- We assume the data is a mixture of several Gaussian distributions, where each distribution represents a cluster.
- We use the Expectation-Maximization (**EM**) algorithm to figure out:
  - $\hat{P}(Y = k \mid X)$: **which points belong to which cluster**
  - $\hat{\pi}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k$: **what each cluster looks like**



The Expectation-Maximization (EM) Algorithm

Iterative process to estimate the GMM parameters from data

EM solves a "chicken-and-egg" problem:

1. If we knew which cluster a point belonged to, we could easily calculate the cluster's parameters: $\hat{\pi}_k$, $\hat{\boldsymbol{\mu}}_k$ or $\hat{\boldsymbol{\Sigma}}_k$
2. If we knew the cluster's parameters, we could easily figure out the probability of a point belonging to it: $\hat{P}(Y = k \mid \boldsymbol{X})$

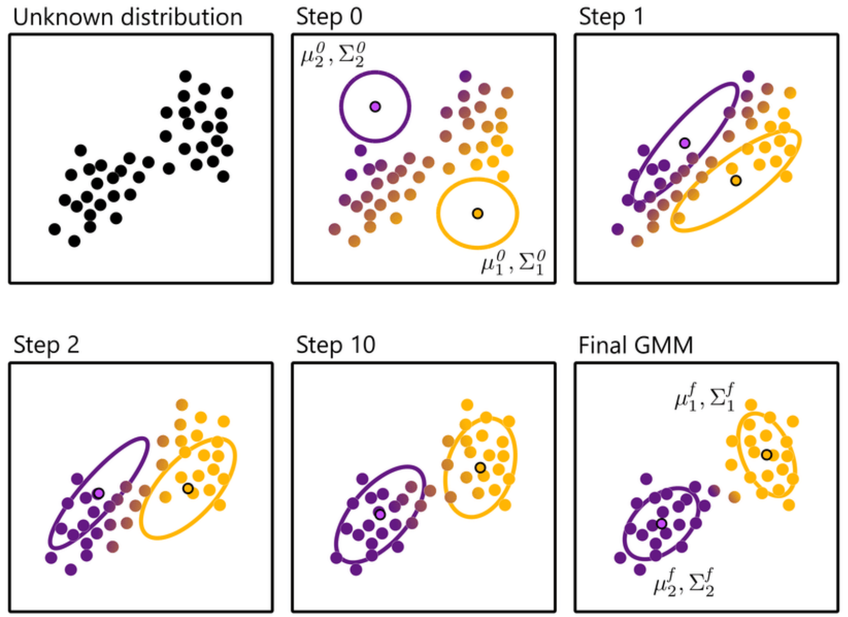
We know neither; EM helps us do both.

Maximum Likelihood from Incomplete Data Via the *EM* Algorithm

AP Dempster, NM Laird… - Journal of the royal … 1977 - Wiley Online Library

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

☆ Save Cite Cited by 76915 Related articles All 55 versions »



Applying the EM algorithm to GMMs

EM iterates between two steps until the parameters stop changing

- **E-Step (Expectation)**: Guess the membership probabilities: $\hat{P}(Y = k \mid \boldsymbol{X})$.
    - Given the current **cluster parameters**, calculate the probability of each data point belonging to each cluster.
- **M-Step (Maximization)**: Given membership probabilities, re-calculate the **cluster parameters** ($\hat{\pi}_k$, $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$) for each cluster.

Points with a high probability of belonging to a cluster will have more influence on its new parameters.

EM algorithm: tailored for GMMs

Two-class clustering problem: $\mathcal{Y} = \{1,2\}$

General EM algorithm is beyond the scope of this class; to learn more, refer to PRML (432 - 442)

Challenge: try writing out the algorithm below for general k-class membership problem.

Recall (from generative model approach):

$$P(Y = 1 \mid \boldsymbol{X}) = \frac{\pi_1 P(\boldsymbol{X} \mid Y = 1)}{\pi_1 P(\boldsymbol{X} \mid Y = 1) + \pi_2 P(\boldsymbol{X} \mid Y = 2)}$$

Where

- $P(\boldsymbol{X} \mid Y = 1) \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$
- $P(\boldsymbol{X} \mid Y = 2) \sim N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$
- $\pi_1 + \pi_2 = 1$

- If we knew $\hat{\boldsymbol{\theta}} = (\hat{\pi}_1, \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}_1, \hat{\boldsymbol{\Sigma}}_2)$ [**cluster parameters**], we could calculate $\hat{P}(Y = k \mid \boldsymbol{X})$ [cluster membership].
- If we knew $\hat{P}(Y = k \mid \boldsymbol{X})$ [cluster membership], we could calculate $\hat{\boldsymbol{\theta}} = (\hat{\pi}_1, \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}_1, \hat{\boldsymbol{\Sigma}}_2)$ [**cluster parameters**]

$$N_k = \sum_{n=1}^{N} \hat{P}(Y = k \mid \boldsymbol{X}_n) \qquad \hat{\boldsymbol{\mu}}_k = N_k^{-1} \sum_{n=1}^{N} \hat{P}(Y = k \mid \boldsymbol{X}_n) \cdot \boldsymbol{X}_n$$

$$\hat{\boldsymbol{\Sigma}}_k = N_k^{-1} \sum_{n=1}^{N} \hat{P}(Y = k \mid \boldsymbol{X}_n) \cdot (\boldsymbol{X}_n - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{X}_n - \hat{\boldsymbol{\mu}}_k)^t \qquad \hat{\pi}_k = N_k / N$$

Notice: there is a slight change from LDA/QDA!!

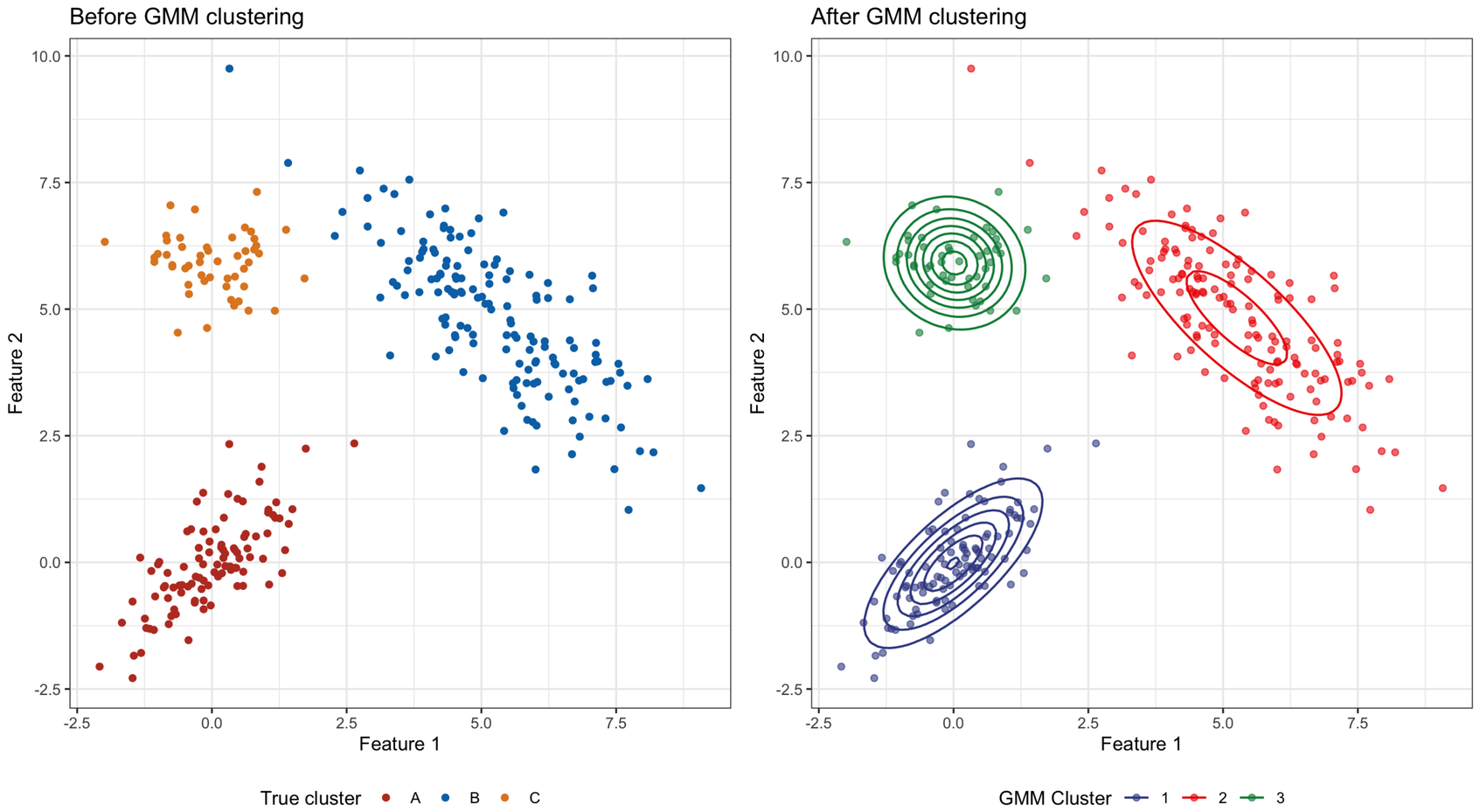
- Start with *initial guess* of **cluster parameters**: set $i = 0$ and call it $\hat{\boldsymbol{\theta}}_i$
- **E-step**: Use $\hat{\boldsymbol{\theta}}_i$ to calculate cluster membership: call it $\hat{P}_i(Y = k \mid \boldsymbol{X})$
- **M-step**: Use cluster membership to update **cluster parameters**: $\hat{\boldsymbol{\theta}}_{i+1}$
- *Convergence check*: if $\left| \hat{\boldsymbol{\theta}}_{i+1} - \hat{\boldsymbol{\theta}}_i \right|_2 > \epsilon$, go to **E-step**; else, stop.

ϵ is (pre-specified) *tolerance hyperparameter*.

$$\left| \boldsymbol{x} - \boldsymbol{y} \right|_2 = \sum_{i=1}^{n} (x_i - y_i)^2$$

GMMs in action

Simulated example: three well-separated classes



The clusters are well-separated. GMM clustering and true clustering are in agreement.

Challenge: modify the cluster parameters to see when GMM does poorly

GMMs in action: continued

Some other things to keep in mind

- Previously, the algorithm figured out there were 3 clusters. In the real world, how does it know?
  - If we choose $k = 1$, the model is too simple (and doesn't *actually* do any clustering)
  - If we choose $k = 100$, the model is (*probably*) too complex and overfit
- How do we find the "Goldilocks" number of clusters? We have two competing approaches: one is parametric, the other is non-parametric.

- **Parametric approach: Bayes Information Criteria (BIC)**. BIC balances model fit with model complexity.
  - It rewards models that fit the data well.
  - It penalizes models for having too many parameters (i.e., too many clusters).
  - Calculate the BIC for different values of k; the best model is the one with the highest BIC score.

- **Nonparametric approach: Silhouette Score.** It measures how well-separated clusters are. For each data point, the score computes $(b - a)/\max(a, b)$, where
  - Cohesion (a): average distance to all other points in the same cluster.
  - Separation (b): average distance to all points in the nearest neighboring cluster.
- **Near +1: The point is well-clustered. Near 0: The point is on the border between two clusters. Near -1: The point may be in the wrong cluster.**
- Calculate average silhouette score across all points for given k. The best model is the one with the highest average silhouette score.

Post-class debrief

Today we discussed…

In absence of class labels, we have to consider an unsupervised way of classification, a.k.a. clustering.

- GMM is a flexible, probabilistic clustering method.
- It models the data as a mixture of Gaussian distributions, allowing for probabilistic assignments.
- The EM algorithm is the engine that iteratively estimates the model parameters (mean, covariance, weight).
- GMM can find non-spherical clusters where other methods might fail.
- We use BIC or the Silhouette Score to select the optimal number of clusters and avoid overfitting.

- Strengths:
    - Flexible Shapes: Can model elliptical, non-spherical clusters.
    - Probabilistic Assignments: Provides probabilities (soft clustering), which can be more informative than hard clustering (k-means)
    - Principled Model Selection: BIC/Silhouette score provides a structured way to choose k.
- Weaknesses:
    - Assumption of Normality: Assumes clusters are Gaussian, which may not always be true.
    - Computationally Intensive: Can be slower than other methods, especially on large datasets.


That's all Folks!