

# Gaussian Mixture Model

He He

CDS, NYU

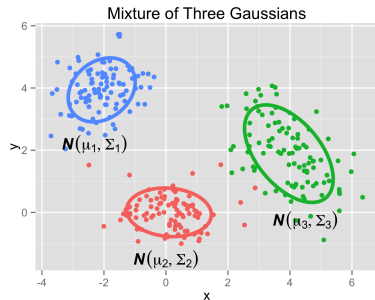
April 27, 2021

# Probabilistic Model for Clustering

- Problem setup:
  - There are  $k$  clusters (or **mixture components**).
  - We have a probability distribution for each cluster.
- Generative story of a **mixture distribution**:
  - 1 Choose a random cluster  $z \in \{1, 2, \dots, k\}$ .
  - 2 Choose a point from the distribution for cluster  $z$ .

Example:

- 1 Choose  $z \in \{1, 2, 3\}$  with  $p(1) = p(2) = p(3) = \frac{1}{3}$ .
- 2 Choose  $x \mid z \sim \mathcal{N}(X \mid \mu_z, \Sigma_z)$ .



# Gaussian mixture model (GMM)

Generative story of GMM with  $k$  mixture components:

- 1 Choose cluster  $z \sim \text{Categorical}(\pi_1, \dots, \pi_k)$ .
- 2 Choose  $x \mid z \sim \mathcal{N}(\mu_z, \Sigma_z)$ .

Probability density of  $x$ :

- Sum over (marginalize) the **latent variable**  $z$ .

$$p(x) = \sum_z p(x, z) \tag{1}$$

$$= \sum_z p(x \mid z) p(z) \tag{2}$$

$$= \sum_k \pi_k \mathcal{N}(\mu_k, \Sigma_k) \tag{3}$$

# Learning GMMs

How to learn the parameters  $\pi_k, \mu_k, \Sigma_k$ ?

- MLE (also called maximize marginal likelihood).
- Log likelihood of data:

$$L(\theta) = \sum_{i=1}^n \log p(x_i; \theta) \quad (4)$$

$$= \sum_{i=1}^n \log \sum_z p(x, z; \theta) \quad (5)$$

- Cannot push log into the sum...  $z$  and  $x$  are coupled.
- No closed-form solution for GMM—try to compute the gradient yourself!

## Learning GMMs: observable case

Suppose we observe cluster assignments  $z$ . Then MLE is easy:

$$n_z = \sum_{i=1}^n 1(z_i = z) \quad \# \text{ examples in each cluster} \quad (6)$$

$$\hat{\pi}(z) = \frac{n_z}{n} \quad \text{fraction of examples in each cluster} \quad (7)$$

$$\hat{\mu}_z = \frac{1}{n_z} \sum_{i: z_i = z} x_i \quad \text{empirical cluster mean} \quad (8)$$

$$\hat{\Sigma}_z = \frac{1}{n_z} \sum_{i: z_i = z} (x_i - \hat{\mu}_z)(x_i - \hat{\mu}_z)^T. \quad \text{empirical cluster covariance} \quad (9)$$

The inference problem: observe  $x$ , want to know  $z$ .

$$p(z = j | x_i) = p(x, z = j) / p(x) \quad (10)$$

$$= \frac{p(x | z = j) p(z = j)}{\sum_k p(x | z = k) p(z = k)} \quad (11)$$

$$= \frac{\pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}{\sum_k \pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k)} \quad (12)$$

- $p(z | x)$  is a *soft assignment*.
- If we know the parameters  $\mu, \Sigma, \pi$ , this would be easy to compute.

Let's compute the cluster assignments and the parameters iteratively.

The expectation-minimization (EM) algorithm:

- ① Initialize parameters  $\mu, \Sigma, \pi$  randomly.
- ② Run until convergence:
  - ① E-step: fill in latent variables by inference.
    - compute soft assignments  $p(z | x_i)$  for all  $i$ .
  - ② M-step: standard MLE for  $\mu, \Sigma, \pi$  given “observed” variables.
    - Equivalent to MLE in the observable case on data weighted by  $p(z | x_i)$ .

## M-step for GMM

- Let  $p(z | x)$  be the soft assignments:

$$\gamma_i^j = \frac{\pi_j^{\text{old}} \mathcal{N}(x_i | \mu_j^{\text{old}}, \Sigma_j^{\text{old}})}{\sum_{c=1}^k \pi_c^{\text{old}} \mathcal{N}(x_i | \mu_c^{\text{old}}, \Sigma_c^{\text{old}})}.$$

- Exercise: show that

$$\mu_c^{\text{new}} = \frac{1}{n_c} \sum_{i=1}^n \gamma_i^c x_i$$

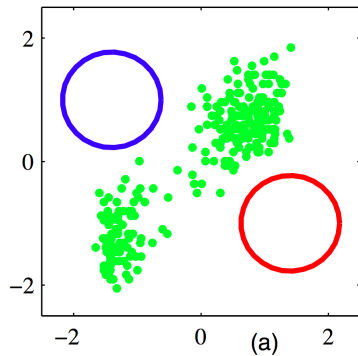
$$\Sigma_c^{\text{new}} = \frac{1}{n_c} \sum_{i=1}^n \gamma_i^c (x_i - \mu_c^{\text{new}}) (x_i - \mu_c^{\text{new}})^T$$

$$\pi_c^{\text{new}} = \frac{n_c}{n}.$$



# EM for GMM

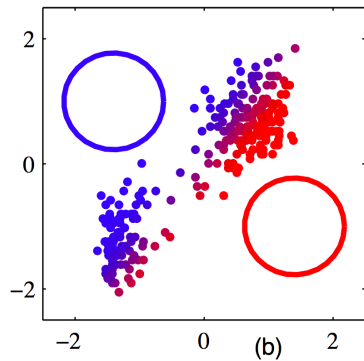
• Initialization



From Bishop's *Pattern recognition and machine learning*, Figure 9.8.

# EM for GMM

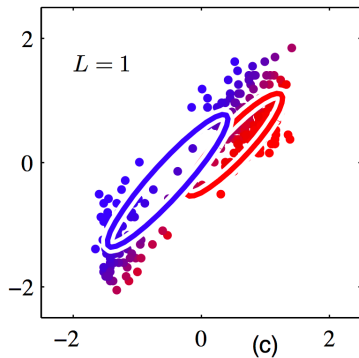
• First soft assignment:



From Bishop's *Pattern recognition and machine learning*, Figure 9.8.

# EM for GMM

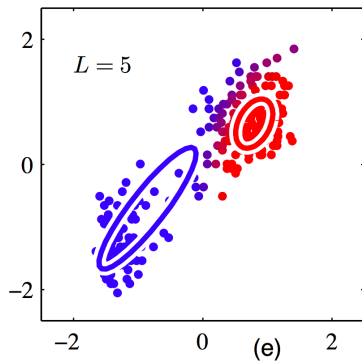
- First soft assignment:



From Bishop's *Pattern recognition and machine learning*, Figure 9.8.

# EM for GMM

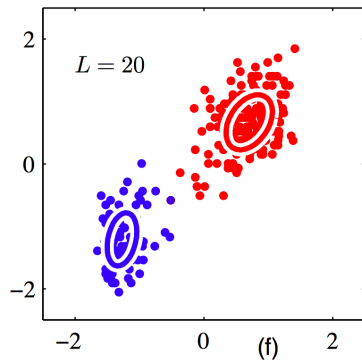
- After 5 rounds of EM:



From Bishop's *Pattern recognition and machine learning*, Figure 9.8.

# EM for GMM

- After 20 rounds of EM:



From Bishop's *Pattern recognition and machine learning*, Figure 9.8.

# EM for GMM: Summary

- EM is a general algorithm for learning latent variable models.
- *Key idea*: if data was fully observed, then MLE is easy.
  - E-step: fill in latent variables by computing  $p(z \mid x, \theta)$ .
  - M-step: standard MLE given fully observed data.
- Simpler and more efficient than gradient methods.
- Can prove that EM monotonically improves the likelihood and converges to a local minimum.
- $k$ -means is a special case of EM for GMM with *hard assignments*, also called hard-EM.