Optimization Techniques for Machine Learning

AMLZC326 · #13 Machine Learning I

Anshid Aboobacker

Motivation

- K-means assigns each data point to *exactly one* cluster (hard assignment).
- But what if a point lies between two clusters, or clusters have different shapes?
- This lecture introduces a probabilistic approach — the **Gaussian Mixture Model (GMM)** — and the **EM algorithm** that fits it.

Learning Objectives

By the end of this lecture you should be able to:

- Explain when a single Gaussian is insufficient and motivate mixture models
- Define a GMM and interpret mixture weights, means, and covariances
- Derive the EM algorithm (E-step: compute responsibilities; M-step: update parameters)
- Describe the convergence guarantee and compare EM with K-means

Table of Contents

Why Not a Single Gaussian?

- Real data often has multiple clusters
- A single Gaussian cannot capture multiple peaks
- We need a more flexible model

Idea: Combine multiple simple distributions

Gaussian Mixture Model (GMM)

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

- μ_k: Mean of cluster k
- Σ_k: Covariance (spread)
- π_k: Weight of cluster ($\sum \pi_k = 1$)

Key Idea: Weighted combination of Gaussians

Running Example

Dataset:

$$\{-3, -2.5, -1, 0, 2, 4, 5\}$$

- Assume $K = 3$ Gaussians
- Initialize:
  - Means: $-4,\ 0,\ 8$
  - Equal weights: $\frac{1}{3}$

Question: Which Gaussian generated each point?

Table of Contents

The Optimization Problem

$$\log p(X|\theta) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right)$$

- Want to maximize likelihood
- Problem: log of sum $\neq$ sum of logs
- No closed-form solution

We need an iterative approach

Responsibilities

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

- Probability that cluster k generated x_n
- Soft assignment (not hard clustering)

Each point belongs to all clusters (with weights)

E-Step (Expectation)

Compute responsibilities using current parameters $\boldsymbol{\mu}_k, \Sigma_k, \pi_k$:

$$r_{nk} = \frac{\pi_k \mathcal{N}(x_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\boldsymbol{\mu}_j, \Sigma_j)}$$

- $r_{nk} \in [0, 1]$: probability that cluster k generated point x_n
- $\sum_{k=1}^{K} r_{nk} = 1$ for every n (soft partition of unity)
- Unlike K-means, every point is assigned to *all* clusters simultaneously with different weights

M-Step (Maximization)

Update parameters using responsibilities:

Mean:

$$\mu_k = \frac{\sum r_{nk} x_n}{\sum r_{nk}}$$

Covariance:

$$\Sigma_k = \frac{1}{N_k} \sum r_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

Weights:

$$\pi_k = \frac{N_k}{N}$$

Weighted averages

Convergence Guarantee

- The log-likelihood $\log p(X|\theta)$ is **non-decreasing** at every EM step
- EM converges to a **local maximum** (not necessarily global)
- Sensitive to initialisation — run multiple random starts and keep the best result

Practical tip: Initialise with K-means cluster centres for a good starting point.

EM Algorithm

1. Initialize μ_k, Σ_k, π_k
2. Repeat:
   - E-step: Compute r_{nk}
   - M-step: Update parameters
3. Until convergence

Each step improves likelihood

Key Takeaways

- GMM: $p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k)$ — a probabilistic, flexible clustering model with soft assignments
- EM alternates: **E-step** (compute responsibilities r_{nk}) and **M-step** (update $\boldsymbol{\mu}_k, \Sigma_k, \pi_k$ via weighted averages)
- The log-likelihood is **non-decreasing** at every EM step — convergence to a local maximum is guaranteed
- EM generalises K-means: K-means is EM with identity covariances and hard (0/1) assignments

Thank you :)