Gaussian Mixture Models (GMM)

1. Introduction

Gaussian Mixture Model

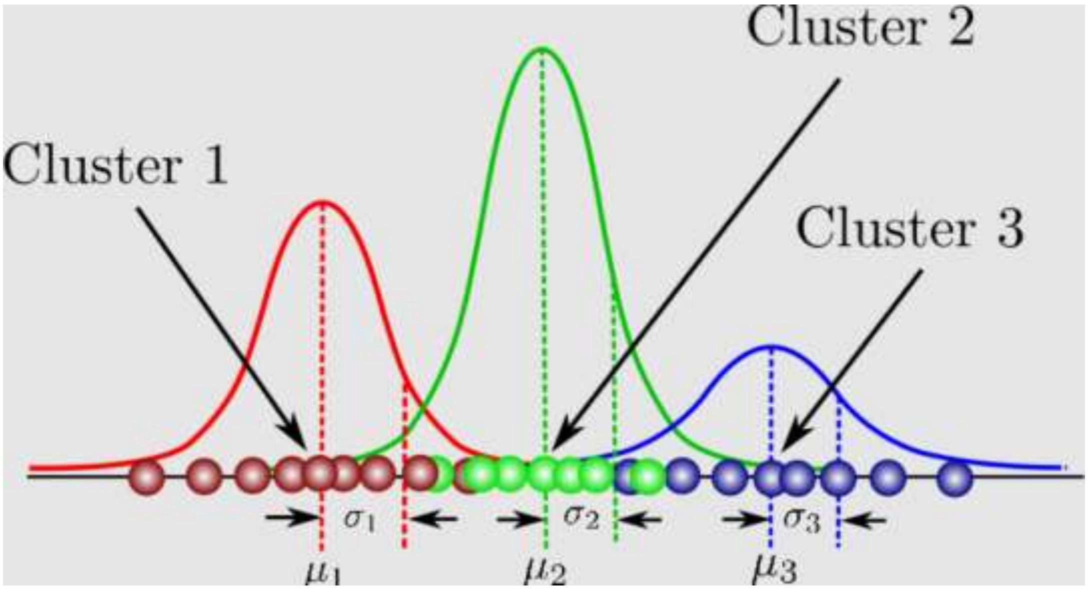
- A **Gaussian Mixture Model (GMM)** is a **probabilistic model** that assumes data is generated from a **mixture of several Gaussian distributions** with unknown parameters.
- It is used for **unsupervised learning**, **clustering**, and **density estimation**.

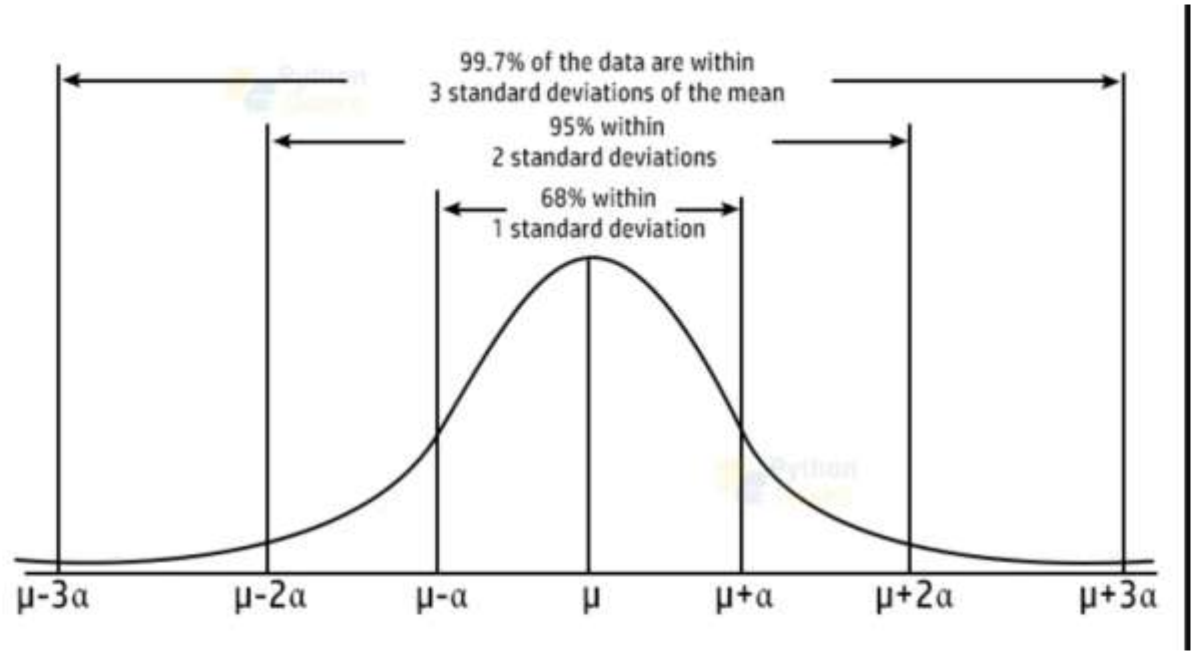
Why GMM?

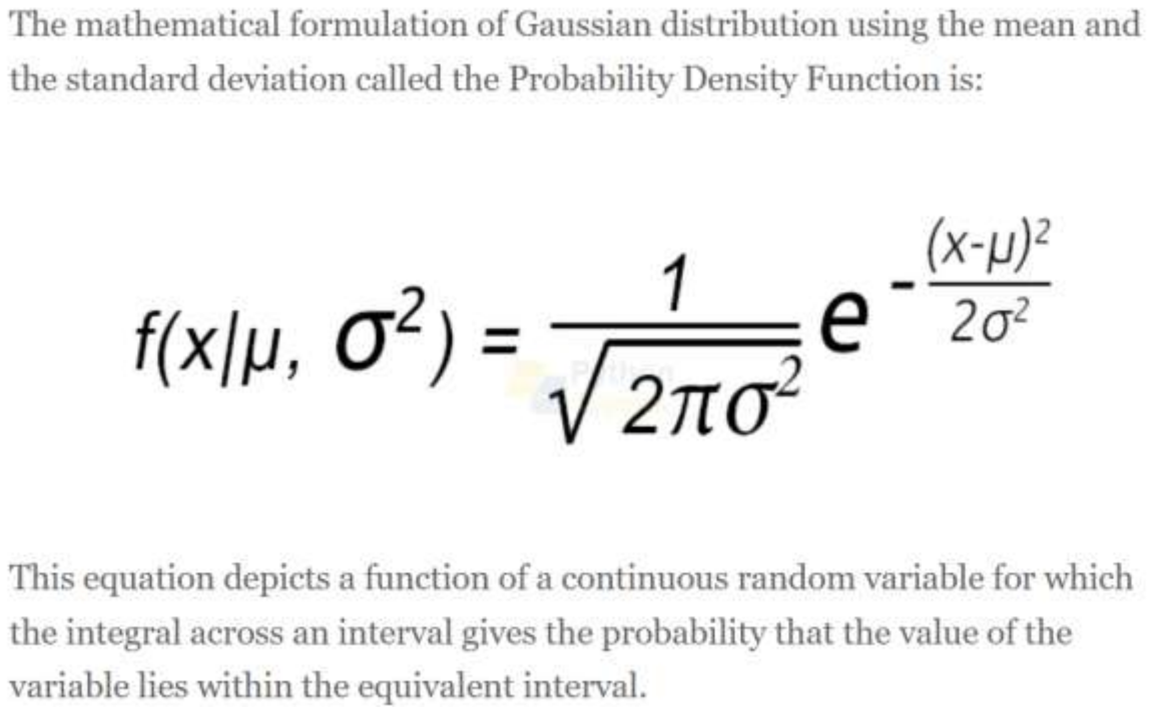
- Unlike k-means, which assigns points to clusters based on distance, GMM provides **soft clustering** by assigning probabilities of membership to each cluster.
- Can model **elliptical shapes**, not just spherical clusters.
- GMM is a powerful tool for modeling data from multiple Gaussian sources.
- It provides a probabilistic, soft clustering alternative to k-means.
- Trained using the EM algorithm.
- Important to assess model fit and choose the number of components wisely.

2. Theoretical Background

A Gaussian Mixture Model (GMM) is a probabilistic model in machine learning that assumes data is generated from a mixture of several Gaussian (normal) distributions. Each of these component Gaussian distributions represents a cluster within the data, and the model estimates the parameters (mean, covariance, and mixture weight) of each component.







The mathematical formulation of Gaussian distribution using the mean and the standard deviation called the Probability Density Function is:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

This equation depicts a function of a continuous random variable for which the integral across an interval gives the probability that the value of the variable lies within the equivalent interval.

- **Probabilistic Nature:**

  Unlike hard clustering methods like K-means, GMMs offer "soft clustering," assigning a probability to each data point of belonging to each cluster. This provides a richer understanding of cluster assignments.

- **Components:**

  A GMM is composed of multiple Gaussian distributions, each characterized by its own:

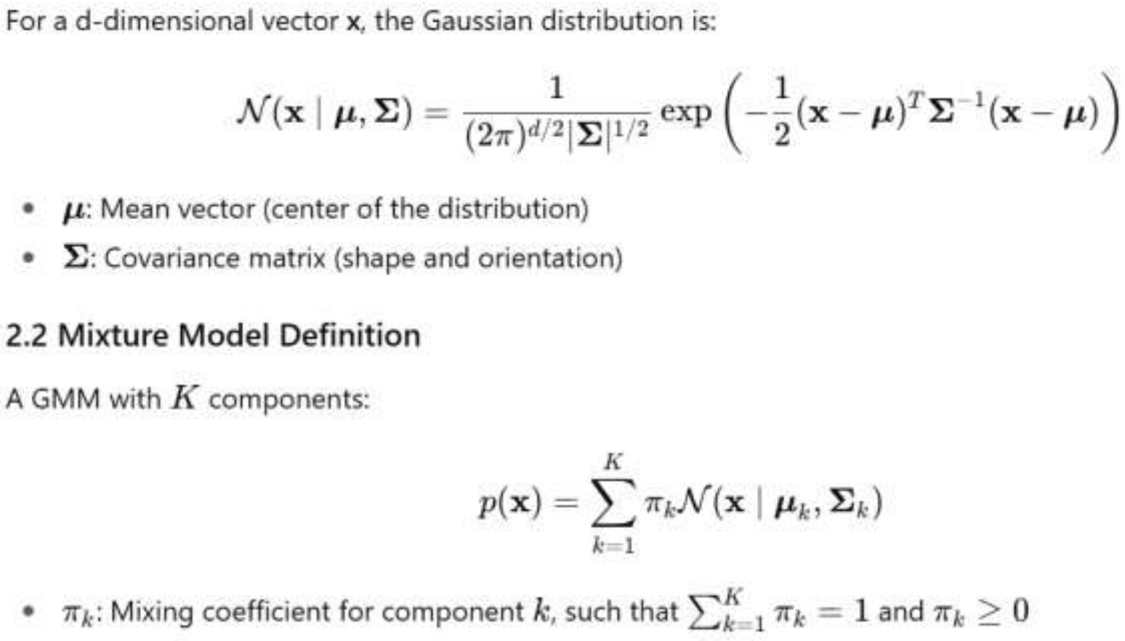
- **Mean:** The center of the cluster.
- **Covariance Matrix:** Describes the shape and orientation of the cluster (allowing for non-spherical clusters, unlike K-means).
- **Mixture Weight:** Represents the prior probability of a data point belonging to that specific component.
- **Expectation-Maximization (EM) Algorithm:**

  The parameters of a GMM are typically estimated using the EM algorithm. This iterative process alternates between two steps:
- **E-step (Expectation):** Calculates the "responsibilities" of each component for each data point, essentially estimating the posterior probability of a data point belonging to each cluster given the current parameter estimates.
- **M-step (Maximization):** Updates the parameters (mean, covariance, and mixture weights) of each component to maximize the likelihood of the observed data, based on the responsibilities calculated in the E-step.
- **Applications:**

  GMMs are widely used for:
- **Clustering:** Grouping data points into clusters based on their likelihood of belonging to different Gaussian components.
- **Density Estimation:** Estimating the probability density function of complex, multimodal data.
- **Anomaly Detection:** Identifying data points that have a low probability of belonging to any of the learned Gaussian components.
- **Advantages:**

  GMMs are flexible and can model complex data distributions, including those with non-spherical clusters. They provide probabilistic assignments, offering more insight than hard clustering.
- **Limitations:**
  GMMs can be computationally intensive, especially with large datasets and many components. The number of components often needs to be determined in advance or estimated using model selection techniques.

2.1 Multivariate Gaussian Distribution



For a d-dimensional vector **x**, the Gaussian distribution is:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

- $\boldsymbol{\mu}$: Mean vector (center of the distribution)
- $\boldsymbol{\Sigma}$: Covariance matrix (shape and orientation)

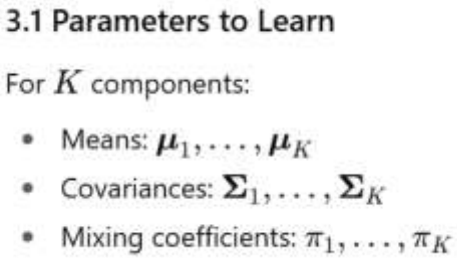
2.2 Mixture Model Definition

A GMM with K components:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- π_k: Mixing coefficient for component k, such that $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k \geq 0$

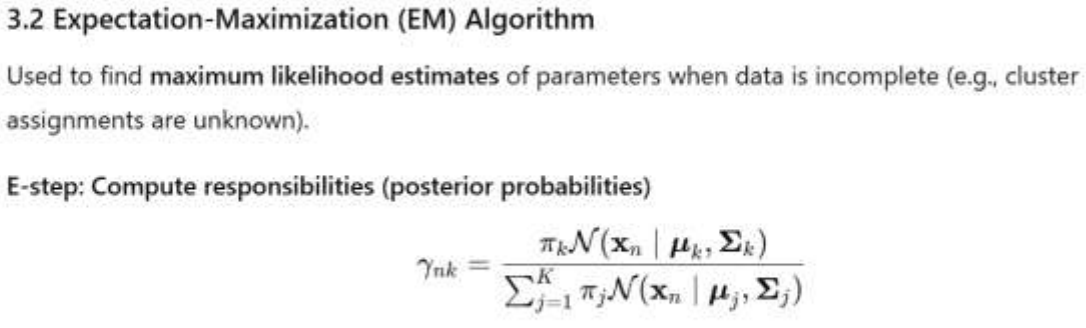
3. Learning GMM Parameters



3.1 Parameters to Learn

For K components:

- Means: $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
- Covariances: $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$
- Mixing coefficients: $\pi_1, \ldots, \pi_K$

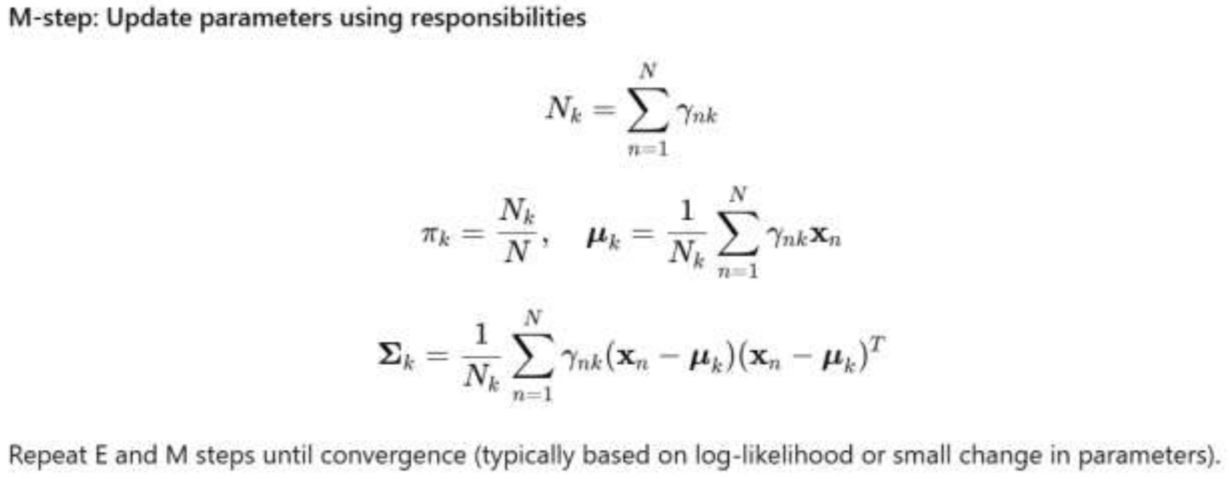


3.2 Expectation-Maximization (EM) Algorithm

Used to find **maximum likelihood estimates** of parameters when data is incomplete (e.g., cluster assignments are unknown).

E-step: Compute responsibilities (posterior probabilities)

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$



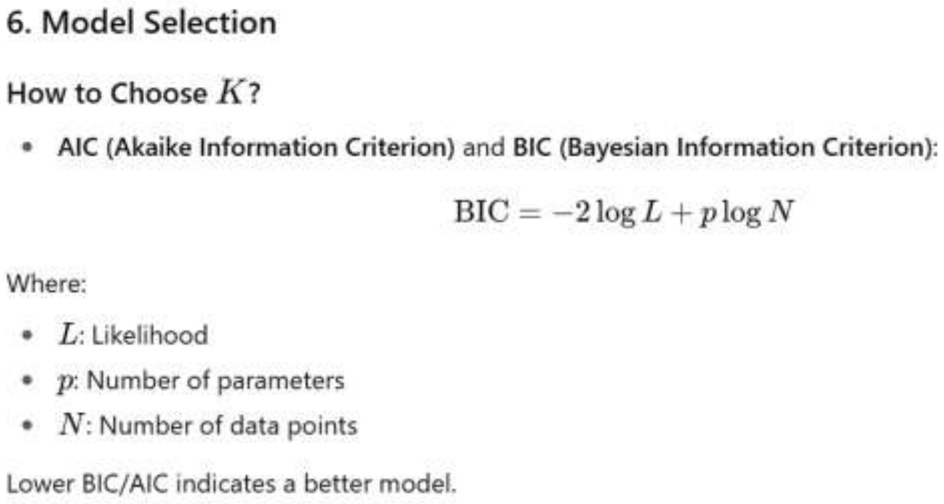
M-step: Update parameters using responsibilities

$$N_k = \sum_{n=1}^{N} \gamma_{nk}$$

$$\pi_k = \frac{N_k}{N}, \quad \boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

Repeat E and M steps until convergence (typically based on log-likelihood or small change in parameters).



6. Model Selection

How to Choose K?

- **AIC (Akaike Information Criterion)** and **BIC (Bayesian Information Criterion)**:

$$\text{BIC} = -2 \log L + p \log N$$

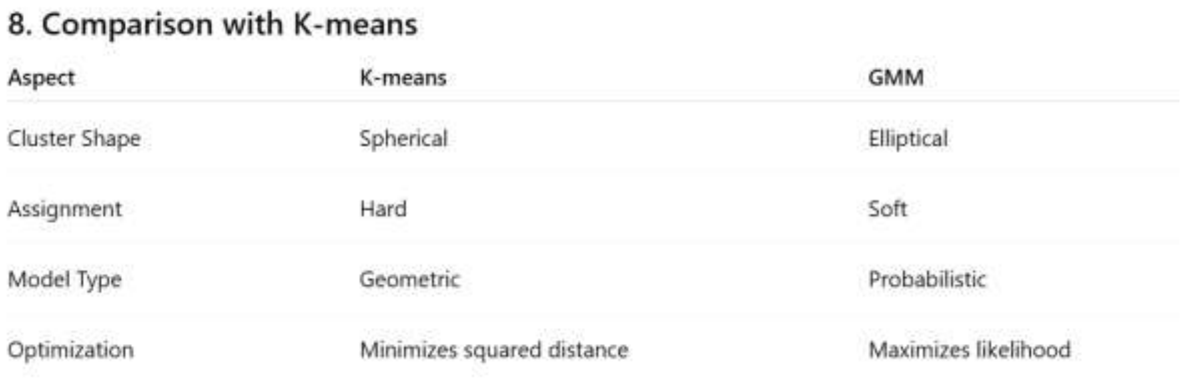
Where:

- L: Likelihood
- p: Number of parameters
- N: Number of data points

Lower BIC/AIC indicates a better model.

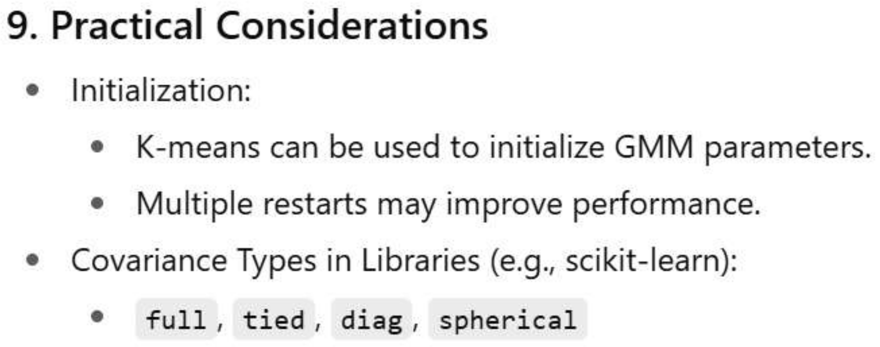
7. Extensions of GMM

- **Variational GMMs**: Bayesian version, incorporating priors over parameters.
- **Dirichlet Process GMMs**: Non-parametric model allowing infinite components (automatically finds number of clusters).
- **Tied covariance GMMs**: Shared covariance across components.



8. Comparison with K-means

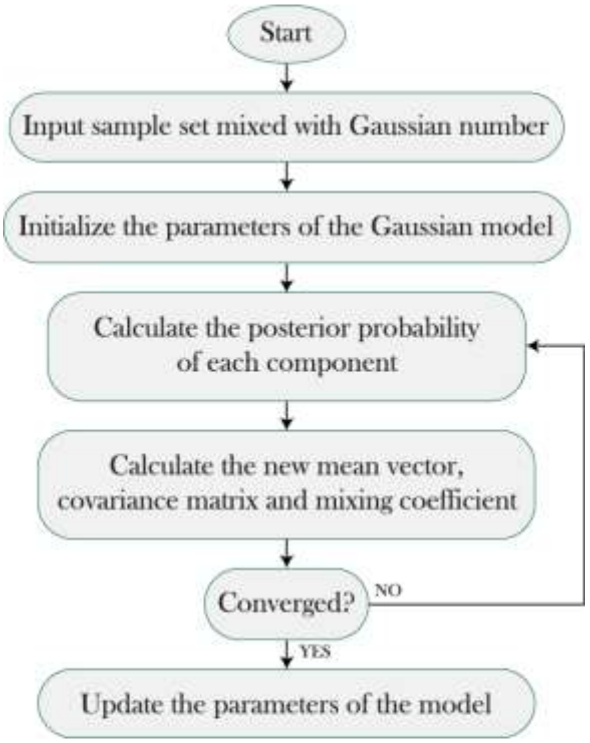
Aspect	K-means	GMM
Cluster Shape	Spherical	Elliptical
Assignment	Hard	Soft
Model Type	Geometric	Probabilistic
Optimization	Minimizes squared distance	Maximizes likelihood



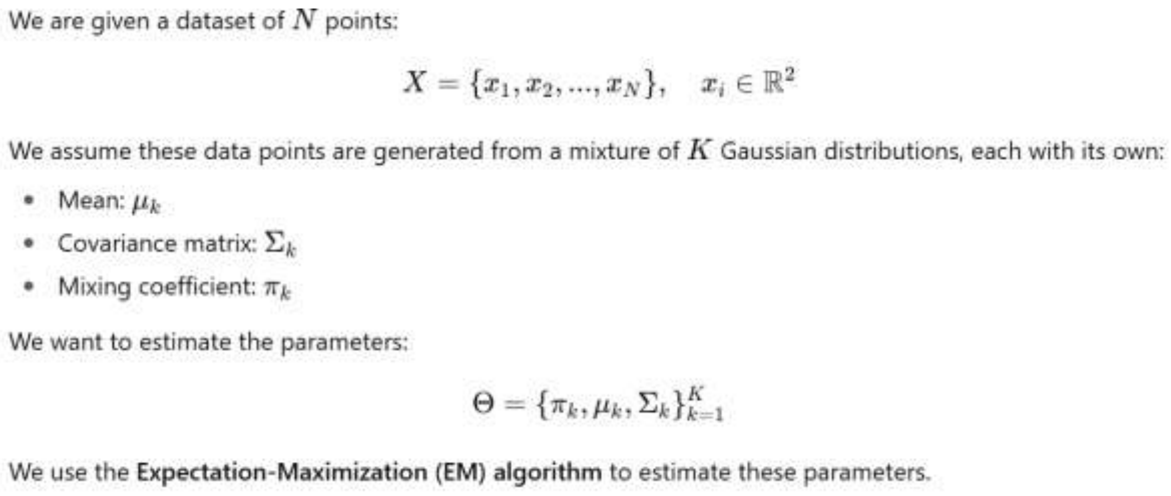
9. Practical Considerations

- Initialization:
  - K-means can be used to initialize GMM parameters.
  - Multiple restarts may improve performance.
- Covariance Types in Libraries (e.g., scikit-learn):
  - `full`, `tied`, `diag`, `spherical`

Flow Chart


Start
Input sample set mixed with Gaussian number
Initialize the parameters of the Gaussian model
Calculate the posterior probability of each component
Calculate the new mean vector, covariance matrix and mixing coefficient
Converged?
NO
YES
Update the parameters of the model

Step-by-Step Solution: Gaussian Mixture Model (GMM)



We are given a dataset of N points:

$$X = \{x_1, x_2, ..., x_N\}, \quad x_i \in \mathbb{R}^2$$

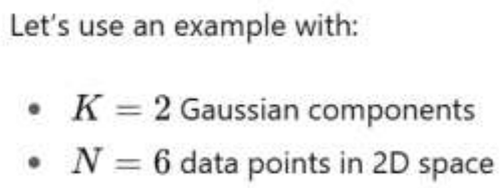
We assume these data points are generated from a mixture of K Gaussian distributions, each with its own:

- Mean: μ_k
- Covariance matrix: Σ_k
- Mixing coefficient: π_k

We want to estimate the parameters:

$$\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$$

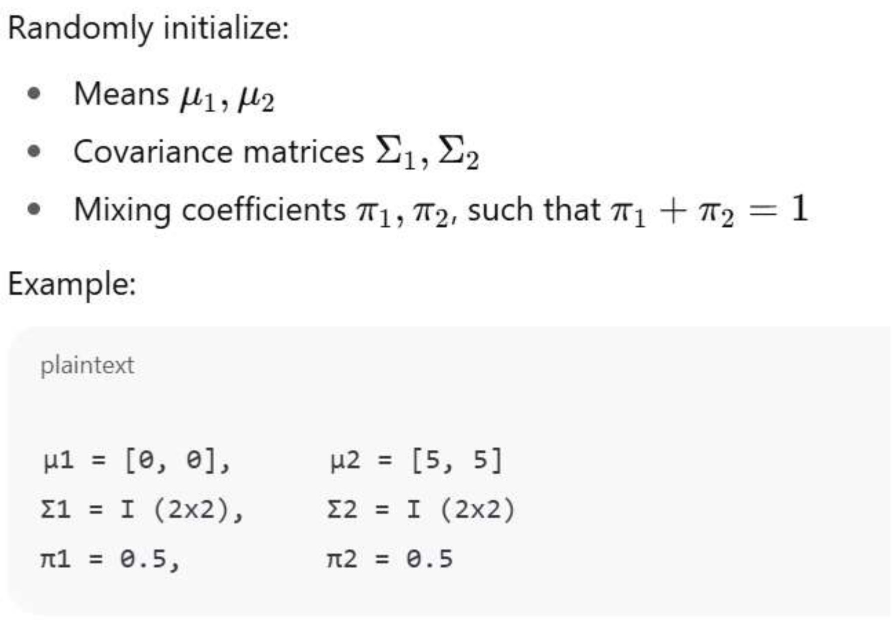
We use the **Expectation-Maximization (EM) algorithm** to estimate these parameters.



Let's use an example with:

- $K = 2$ Gaussian components
- $N = 6$ data points in 2D space

Step 1: Initialization



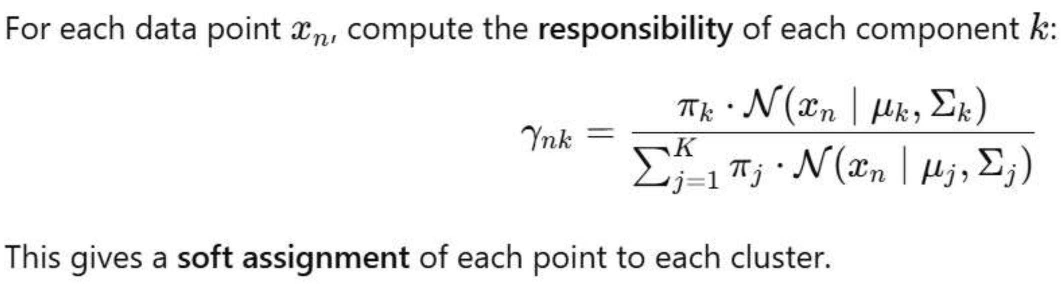
Randomly initialize:

- Means μ_1, μ_2
- Covariance matrices Σ_1, Σ_2
- Mixing coefficients π_1, π_2, such that $\pi_1 + \pi_2 = 1$

Example:

```plaintext
μ1 = [0, 0],       μ2 = [5, 5]
Σ1 = I (2x2),      Σ2 = I (2x2)
π1 = 0.5,          π2 = 0.5
```

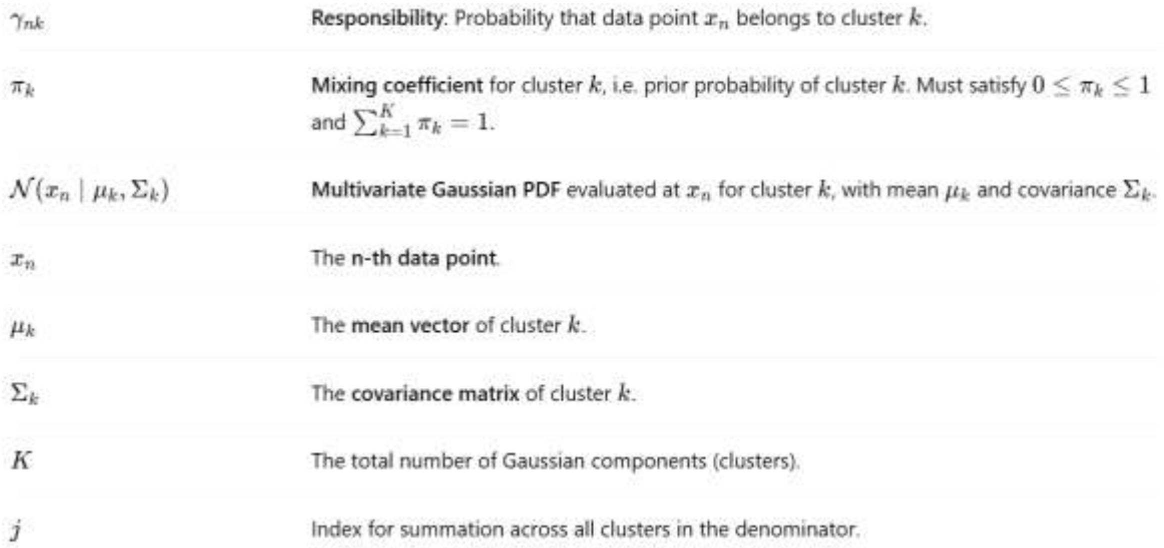
Step 2: Expectation (E-Step)



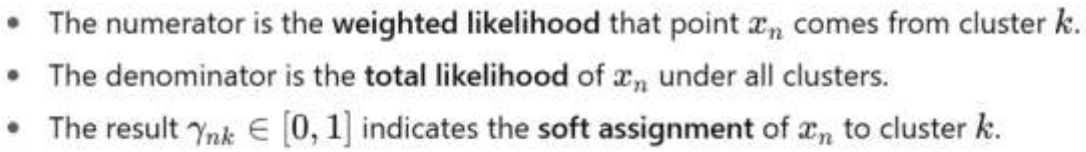
For each data point x_n, compute the **responsibility** of each component k:

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

This gives a **soft assignment** of each point to each cluster.

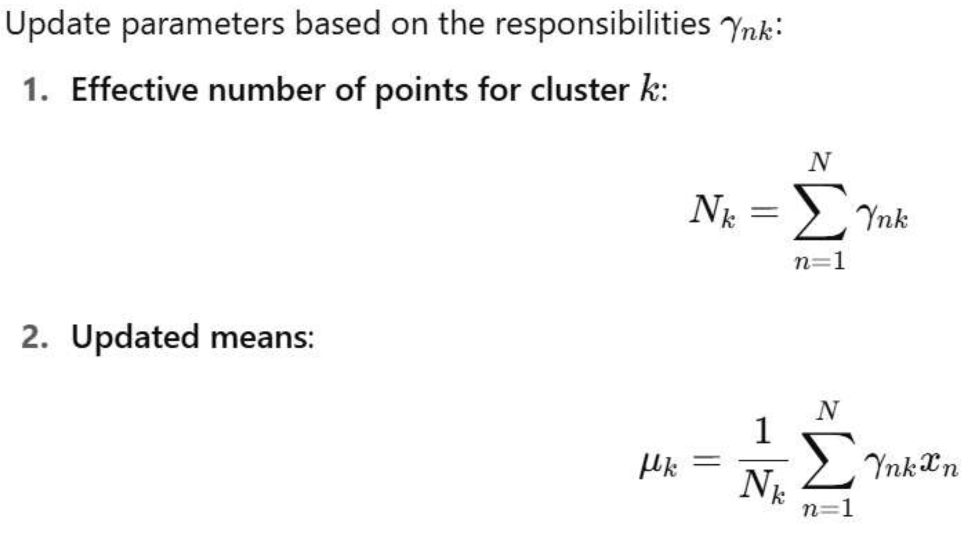


γ_{nk}	**Responsibility**: Probability that data point x_n belongs to cluster k.
π_k	**Mixing coefficient** for cluster k, i.e. prior probability of cluster k. Must satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.
$\mathcal{N}(x_n \mid \mu_k, \Sigma_k)$	**Multivariate Gaussian PDF** evaluated at x_n for cluster k, with mean μ_k and covariance Σ_k.
x_n	The **n-th data point**.
μ_k	The **mean vector** of cluster k.
Σ_k	The **covariance matrix** of cluster k.
K	The total number of Gaussian components (clusters).
j	Index for summation across all clusters in the denominator.



- The numerator is the **weighted likelihood** that point x_n comes from cluster k.
- The denominator is the **total likelihood** of x_n under all clusters.
- The result $\gamma_{nk} \in [0, 1]$ indicates the **soft assignment** of x_n to cluster k.

Step 3: Maximization (M-Step)



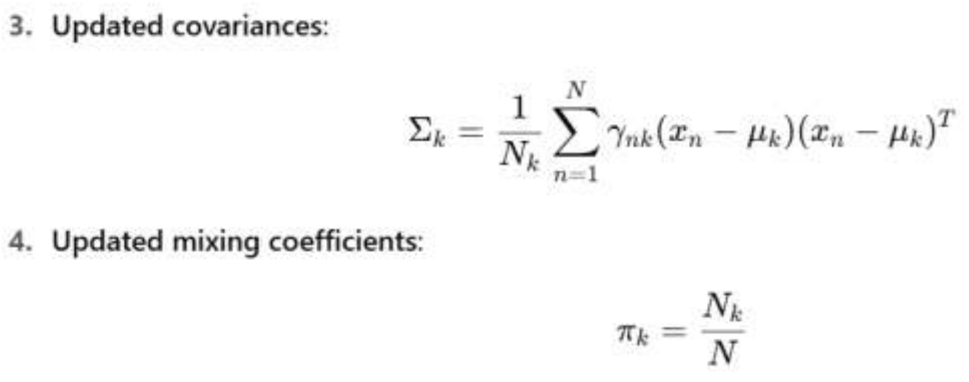
Update parameters based on the responsibilities γ_{nk}:

1. **Effective number of points for cluster k:**

$$N_k = \sum_{n=1}^{N} \gamma_{nk}$$

2. **Updated means:**

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n$$



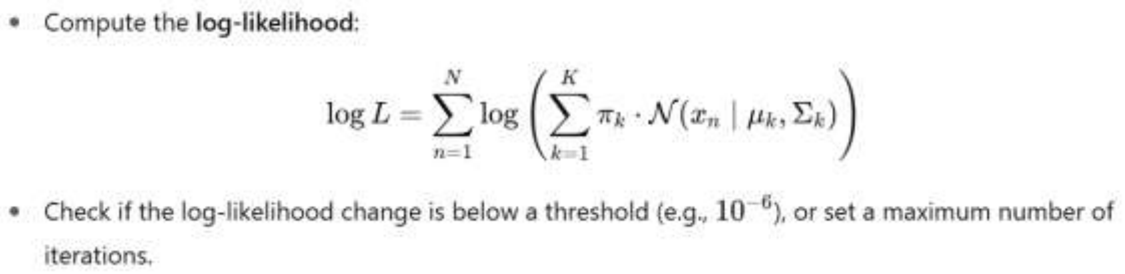
3. **Updated covariances:**

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

4. **Updated mixing coefficients:**

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

Step 4: Convergence Check



- Compute the **log-likelihood**:

$$\log L = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right)$$

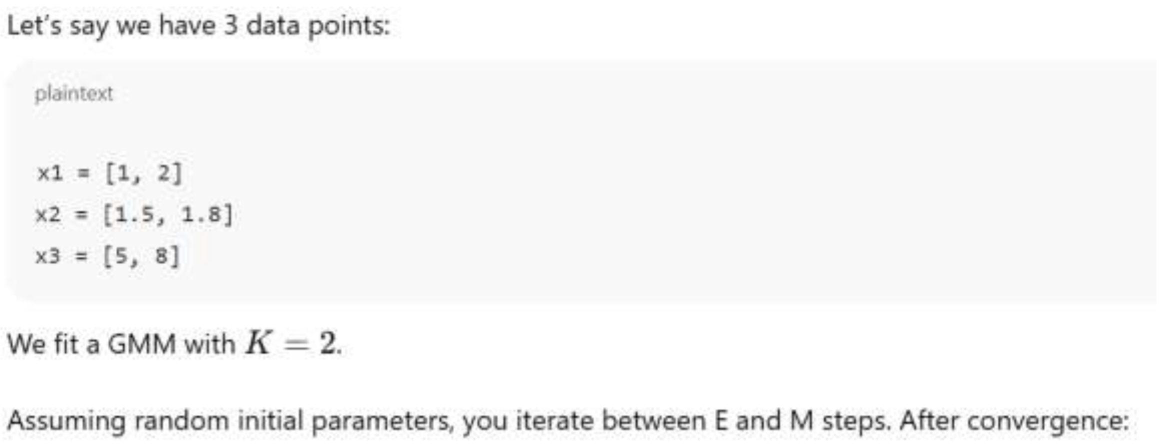
- Check if the log-likelihood change is below a threshold (e.g., 10^{-6}), or set a maximum number of iterations.

Step 5: Final Results

After convergence:

- Each data point has a **probability of belonging** to each cluster.
- You can assign points to the **most probable cluster** or use the probabilities directly.

Example

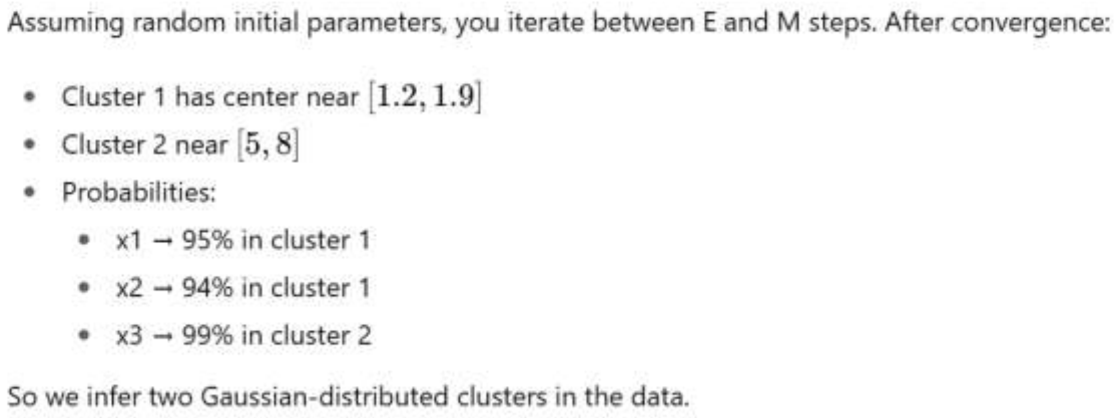


Let's say we have 3 data points:

```plaintext
x1 = [1, 2]
x2 = [1.5, 1.8]
x3 = [5, 8]
```

We fit a GMM with $K = 2$.

Assuming random initial parameters, you iterate between E and M steps. After convergence:

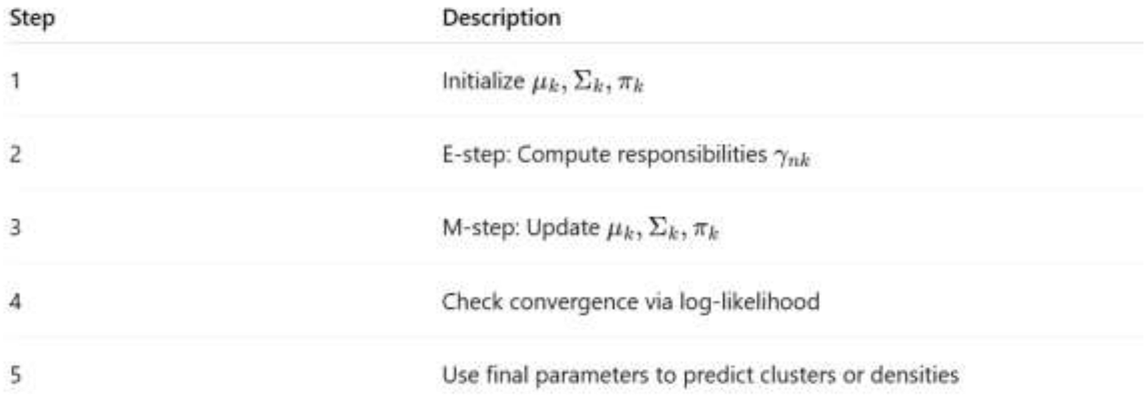


Assuming random initial parameters, you iterate between E and M steps. After convergence:

- Cluster 1 has center near $[1.2, 1.9]$
- Cluster 2 near $[5, 8]$
- Probabilities:
  - x1 → 95% in cluster 1
  - x2 → 94% in cluster 1
  - x3 → 99% in cluster 2

So we infer two Gaussian-distributed clusters in the data.

Summary of GMM Steps



Step	Description
1	Initialize μ_k, Σ_k, π_k
2	E-step: Compute responsibilities γ_{nk}
3	M-step: Update μ_k, Σ_k, π_k
4	Check convergence via log-likelihood
5	Use final parameters to predict clusters or densities

4. Applications

- **Clustering**: Soft assignment of data to clusters.
- **Anomaly Detection**: Low likelihood points can be flagged as anomalies.
- **Image Processing**: Segmentation based on pixel distributions.
- **Speech Recognition**: Acoustic modeling.

5. Advantages and Limitations

Advantages

- More flexible than k-means (elliptical clusters)
- Probabilistic foundation (soft assignments)
- Handles overlapping clusters

Limitations

- Requires choosing KKK
- Sensitive to initialization
- Can converge to local maxima
- Computationally expensive for high dimensions