Gaussian Mixture Models (GMMs)

Short Video

1 Introduction to Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are widely used for **density estimation**, **clustering**, and **classification**. They provide a flexible way to model complex distributions by assuming that data is generated from a mixture of multiple Gaussian distributions.

A **Gaussian Mixture Model (GMM)** is represented mathematically as:

$$f_X(x) = \sum_{c=1}^{K} \pi_c N(x|\mu_c, \Sigma_c) \tag{1}$$

where:

- $N(x|\mu_c, \Sigma_c)$ is the Gaussian distribution with mean μ_c and covariance matrix Σ_c.
- π_c are the mixture weights, such that $\sum_{c=1}^{K} \pi_c = 1$.

Pay attention that c means component in this case, not class.

In a Gaussian Mixture Model (GMM), the number of components K is a hyperparameter. Selecting an appropriate K is crucial, as a small K may lead to underfitting, while a large K can cause overfitting. GMMs allow approximating any sufficiently regular distribution (given enough components), making them useful for various applications, including:

- **Density Estimation**: Modeling the probability distribution of a dataset.
- **Clustering**: Identifying underlying subgroups within data (alternative to K-means).
- **Classification**: Assigning labels to data points based on learned distributions.

2 Definition, Interpretation, and Latent Variable Formulation

A GMM can be viewed as a **latent variable model**, where each data point is associated with an unobserved cluster label C.

3 Hard Assignment vs. Soft Assignment in GMM

A Gaussian Mixture Model (GMM) assigns data points to Gaussian components in two different ways: **hard assignment** and **soft assignment**. These approaches describe how each data point is associated with the mixture components. The cluster label C_i (which represents which Gaussian component generated x_i) is a latent variable—it exists but is not observed directly.

3.1 Hard Assignment

In the hard assignment approach, each data point x_i is assigned to exactly one of the K Gaussian distributions. This is similar to k-means clustering, where each point belongs exclusively to a single cluster. The assignment is determined by selecting the Gaussian component with the highest posterior probability:

$$C_i = \arg\max_k P(C_i = k|x_i) \tag{2}$$

Once the component $C_i = k$ is chosen, the data point is assumed to be generated solely by that Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$. This approach simplifies the clustering but does not account for uncertainty in the assignment.

3.2 Soft Assignment

In contrast, soft assignment allows each data point x_i to belong to multiple Gaussian components with different probabilities. Instead of choosing a single component, we compute **responsibilities** that represent the probability that x_i belongs to each Gaussian:

$$\gamma_{ik} = P(C_i = k|x_i) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{3}$$

3.3 Comparison of Hard and Soft Assignment

Hard assignment forces each data point into a single cluster, which is useful in discrete clustering but may ignore the uncertainty in membership. Soft assignment, on the other hand, allows for probabilistic cluster membership, making it more flexible for modeling real-world data distributions. The Expectation-Maximization (EM) algorithm utilizes soft assignment to iteratively refine cluster parameters and improve mixture component estimation.

4 Estimation of Model Parameters

The parameters of a GMM (π_c, μ_c, Σ_c) are typically estimated using the **Expectation-Maximization (EM) algorithm**.

4.1 Expectation-Maximization (EM) Algorithm

1. **Initialization**: Choose initial values for π_c, μ_c, Σ_c.
2. **E-Step**: Compute the responsibilities:

$$\gamma_{c,i} = \frac{\pi_c N(x_i|\mu_c, \Sigma_c)}{\sum_{c'=1}^{K} \pi_{c'} N(x_i|\mu_{c'}, \Sigma_{c'})} \tag{4}$$

3. **M-Step**: Update the parameters using weighted averages:
   Think of $\sum_{i=1}^{N} \gamma_{c,i}$ as the effective number of points assigned to cluster c
   .

$$\mu_c = \frac{\sum_{i=1}^{N} \gamma_{c,i} x_i}{\sum_{i=1}^{N} \gamma_{c,i}} \tag{5}$$

$$\Sigma_c = \frac{\sum_{i=1}^{N} \gamma_{c,i} (x_i - \mu_c)(x_i - \mu_c)^T}{\sum_{i=1}^{N} \gamma_{c,i}} \tag{6}$$

$$\pi_c = \frac{\sum_{i=1}^{N} \gamma_{c,i}}{N} \tag{7}$$

4. **Repeat Steps 2 and 3** until convergence.

5 Gaussian Mixture Models for Classification Problems

Gaussian Mixture Models (GMMs) can be utilized for classification by modeling class-conditional probability densities. Given a dataset with multiple classes, each class c is represented by a GMM characterized by its parameters:

$$\Theta_c = \{\mu_c, \Sigma_c, \pi_c\}, \tag{8}$$

where:

- μ_c represents the mean of each Gaussian component in the mixture,
- Σ_c is the covariance matrix,
- π_c denotes the mixing coefficients, satisfying $\sum_k \pi_{c,k} = 1$.

Classification using GMMs follows a **Bayesian approach**, where the probability of an input sample X belonging to class C is computed using Bayes' theorem:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}, \tag{9}$$

where:

- $P(X|C)$ is the likelihood, modeled using the Gaussian mixture,
- $P(C)$ is the prior probability of class C,
- $P(X)$ is the evidence, obtained by marginalizing over all classes.

5.1 Handling Open-Set Classification with GMMs

In a **closed-set classification** scenario, the model assumes that every test sample belongs to one of the known classes. However, in an **open-set classification** setting, a model must account for the possibility of **unknown** classes appearing during inference.

GMMs naturally lend themselves to **open-set classification** due to their ability to model class-conditional probability densities. This enables the classifier to:

1. **Identify Unknown Samples:** If $P(X|C)$ is significantly low for all known classes, the sample X is likely from an unknown class.
2. **Threshold-Based Rejection:** A decision threshold τ can be introduced such that if:

   $$\max_c P(C|X) < \tau, \tag{10}$$

   the sample is rejected as belonging to an **unknown** class.

   Still to keep in mind that the sum of

   $$P(C|X) \tag{11}$$

   for each C will sum up to 1

This approach allows for a **soft decision boundary**, making GMM-based classification more robust in real-world scenarios where new, unseen classes might be encountered. The effectiveness of this method depends on proper tuning of the mixture components, covariance matrices, and threshold values to balance classification accuracy with rejection performance.

6 Variations of Gaussian Mixture Models

Potential Issues of Gaussian Mixture Models (GMMs), Possible Ways to Address These Issues, and Possible Variations of the Model

1. Potential Issues of GMMs

GMMs are powerful for density estimation and classification, but they come with several limitations: • Curse of Dimensionality: In high-dimensional spaces, GMMs struggle because Gaussian distributions become inefficient at modeling complex data structures. • Overfitting: When the number of components K is too high, GMMs may overfit the data, capturing noise rather than meaningful patterns. • Slow Convergence in EM Algorithm: The Expectation-Maximization (EM) algorithm, commonly used for GMM training, can converge slowly and sometimes gets stuck in local optima. • Sensitivity to Initialization: Poor initialization of mixture components can lead to suboptimal clustering or slow convergence. • Assumption of Gaussian Components: GMMs assume that each cluster follows a Gaussian distribution, which may not always be true in real-world datasets.

2. Possible Ways to Address These Issues • Dimensionality Reduction: Apply PCA (Principal Component Analysis) before using GMMs to reduce the number of features and mitigate the curse of dimensionality. • Regularization: Use regularized covariance matrices (e.g., adding a small constant to the diagonal) to prevent singularity issues. • Better Initialization: Use K-Means clustering to initialize the GMM parameters before running EM, improving convergence.

Gaussian Mixture Models (GMMs) can be adapted in various ways by modifying the covariance structure of the Gaussian components. Below, we discuss three important variations: Tied GMMs, Diagonal GMMs, and Full GMMs.

6.1 Tied GMMs

Tied GMMs use a single covariance matrix for all mixture components rather than allowing each to have its own. This constraint reduces the number of parameters and prevents overfitting in cases where data is limited.

6.2 Diagonal GMMs

Diagonal GMMs impose a restriction that the covariance matrices of all Gaussian components are diagonal. This means that each feature's variance is modeled independently, ignoring feature correlations.