# Gaussian Mixture models - explained

Ryan Balshaw

January 2022

## 1 Introduction

The purpose of this document is to write up all of the derivations I spent time going through related to GMMs. I will work on this over time, and make sure that it is simple to understand and that a new reader can understand how to optimise the hyper-parameters of a GMM.

Gaussian mixture models (GMMs) are an interesting and useful extension of the standard Gaussian model that incorporated multiple individual Gaussian distributions into one model through superposition. This is done to enrich and meliorate the model to better capture any potential data multi-modality, improve the model flexibility to better capture the data distribution of interest, and we can explicitly control important aspects of the GMM as its underlying formulation is that of a Gaussian distribution. Furthermore, as described in Bishop [CITE], if a sufficient number of Gaussians is used, almost any continuous distribution can be approximated to an arbitrary accuracy.

At its core, a GMM is a generative model, however its differentiating factor is that the latent variables are discrete, as opposed to the continuous case that can be found in other generative models such as probabilistic Principal Component Analysis (pPCA)[CITE], Variational Auto-Encoders (VAEs) [CITE], and Generative Adversarial Networks (GANs). A fully connected directed graphical model representation of a GMM is given in Figure [CITE], where $\mathbf{z}$ is the latent variable node that is a parent to $\mathbf{x}$, which is the data node. In this graphical formulation, we define a joint distribution $p(\mathbf{x}, \mathbf{z})$ in terms of a marginal distribution $p(\mathbf{z})$ and a conditional distribution $p(\mathbf{x}|\mathbf{z})$. As I mentioned previously, the distinction between the GMMs and other generative models is that the latent variables are discrete, and we assume that the variable has a 1-of-$K$ representation. This representation produces a vector where the $k^{th}$ index $z_k$ equals 1, and all other indices are equal to zero. This vector also satisfies $\sum_{k=1}^{K} z_k = 1$. As an example, assume that $K = 5$, and that $z_0 = 1$, then the latent sample $\mathbf{z}$, for will be represented by

$$\mathbf{z} = \begin{bmatrix} 1, & 0, & 0, & 0, & 0 \end{bmatrix}^T. \tag{1}$$

However, this latent representation does not describe a probability distribution, as the latent variable simply refers to the $k^{th}$ state, and this carries no information as to how likely the $k^{th}$ state is. We need another variable, which we denote $\boldsymbol{\pi}$, which is a $K$ dimensional vector $\boldsymbol{\pi} = \begin{bmatrix} \pi_1, & \cdots, & \pi_k \end{bmatrix}^T$, to describe how likely the $k^{th}$ state is. In this format, $\pi_k$ represents the probability of $z_k = 1$. The marginal distribution $p(\mathbf{z})$ can then be given as

$$p(\mathbf{z}|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_k}, \tag{2}$$

which is known as the generalised Bernoulli distribution. If you are a little confused about the effect of the product $\prod$ operator, remember that any scalar raised to the power of zero equals 1. Thus, for a given $z_k = 1$ state, $\pi_k$ is returned as $p(z_k = 1|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_k} = 1 \times \pi_k \times 1 \times \cdots \times 1$. It is important to note that $\boldsymbol{\pi}$ is constrained by $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$ to ensure that $p(\mathbf{z}|\boldsymbol{\pi})$ is a valid probability distribution. Now, the key ingredient of a GMM is that each of the $K$ discrete indices in $\mathbf{z}$ describes a Gaussian distribution. This is given as

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3}$$

which can also be written as

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}. \tag{4}$$

Importantly, as each conditional distribution $p(\mathbf{x}|z_k = 1)$ has parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, we can impose constraints on $\boldsymbol{\Sigma}_k$ to change how the model fits to the data. Examples of such constraints are *i)* the spherical covariance constraint, *ii)* the diagonal covariance constraint, *iii)* the tied diagonal covariance constraint, *iv)* the tied covariance constraint, and *v)* the full covariance which imposes no constraint on $\boldsymbol{\Sigma}_k$. I will elaborate on each of these constraints at a later stage.

We can write the joint distribution $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ as

$$p(\mathbf{x}, \mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}, \tag{5}$$

and we can obtain the marginal data distribution $p(\mathbf{x})$ through

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) d\mathbf{z}, \tag{6}$$

where the integral can be simplifed to a summation as $\mathbf{z}$ is discrete. The resulting marginal distribution can be given as

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{7}$$

I find the fact that the marginal distribution $p(\mathbf{x})$ can be tractably determined to be a useful result of the discrete latent variable assumption, as often in non-linear generative models the integral over the latent space cannot be tractably computed as the latent space is continuous. The is beneficial in two ways: *i)* we can directly estimate the data likelihood for any sample $\mathbf{x}_i$ using $p(\mathbf{x})$, and *ii)* we can estimate the posterior latent conditional probability for any sample $\mathbf{x}_i$. Recall that Bayes' theorem states that the relationship between conditional probabilities is

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z})p(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}, \tag{8}$$

and for $z_k = 1$, this can be written as

$$\begin{aligned} p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)}, \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \end{aligned} \tag{9}$$

As described in Bishop [CITE], $p(z_k = 1|\mathbf{x})$ is referred to as the responsibility that latent component $z_k$ accepts in being responsibile for the sample $\mathbf{x}_i$.

# 2 Optimising the model

The next stage of this write-up is to detail how the model parameters are optimised. First, we assume that we have a dataset of $N$ observations $\{\mathbf{x}_i, \cdots, \mathbf{x}_N\}$ and we wish to model this data using a GMM. We represent this dataset in a matrix $\mathbf{X}$ where $\mathbf{X} = [\mathbf{x}_i, \cdots, \mathbf{x}_N]^T$. Note that $\mathbf{x}_i$ is a column vector $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{X}$ is a matrix where $\mathbf{X} \in \mathbb{R}^{N \times D}$ and the $n^{th}$ row of $\mathbf{X}$ is given by $\mathbf{x}_n^T$. The natural logarithm of the likelihood function $p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given as

$$\log_e p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \left[ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right], \tag{10}$$

where $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$ and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_K\}$. Before we continue, however, there are three core issues with the GMM log-likelihoood function that we wish to optimise. Firstly, if $K \geq 2$, there exists the case that the model may assign $\mu_j$ to any sample $\mathbf{x}_n$, and induce a singularity in the model. What is implied here is that the model may match the $k^{th}$ mean $\boldsymbol{\mu}_k$ with one of the samples such that the likelihood of $p(\mathbf{x}_n|z_k = 1)$ is proportional to $\frac{1}{|\boldsymbol{\Sigma}|}$, where $|\boldsymbol{\Sigma}_k| = \det(\boldsymbol{\Sigma}_k)$ is the determinant of $\boldsymbol{\Sigma}_k$. In this single sample matching process, the model can then reduce the covariance determinant (thereby shrinking the volume (?) of the distribution) around $\mathbf{x}_n$ such that the log-likelihood function tends towards infinity.

Secondly, as there are $K$ components, there are $K!$ equivalent solutions that correspond to the $K!$ ways in which the parameter set $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ can be assigned to $K$ components. This issue is central to interpreting the parameters of the model, however if we just wish to find a good generative model, the different equivalent solutions is irrelevant as they are simply index perturbations of $\boldsymbol{\theta}$.

Finally, the log-likelihood function in Equation (10) is complicated by the summation over the $K$ classes within the logarithm. If we take the derivative of the log-likelihood function, not only does the summation not disappear, but we are left with exponential terms in each of the $K$ Gausian distributions from $p(\mathbf{x}|\mathbf{z})$ which complicates the ability to obtain an analytical solution. To overcome this issue, we turn to the *expectation-maximisation* (EM) algorithm [CITE].

## 2.1 Expectation Maximisation

The objective of the EM algorithm is to find maximum likelihood solutions to models with latent variables. To initialise the EM algorithm framework, let $\mathbf{X}$ be a data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, where the $n^{th}$ row represents $\mathbf{x}_n^T$ and let $\mathbf{Z}$ be the latent matrix $\mathbf{Z} \in \mathbb{R}^{N \times K}$, where where the $n^{th}$ row represents $\mathbf{z}_n^T$. We can write the log-likelihood function in Equation (10) as

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \left[ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \right], \tag{11}$$

where we have replaced $\log_e(\cdot)$ with $\ln(\cdot)$. If we recall the steps in our derivation of Equation (10), we noted that the marginalisation of $p(\mathbf{x}, \mathbf{z})$ resulted in a sum over the $K$ latent states or classes, and that this summation term reflects in the log-likelihood function and ultimately complicates the procedure.

Imagine for a second that we had access to both the dataset $\mathbf{X}$ and the corresponding latent variables $\mathbf{Z}$. If we had access to this information, we would not have to maximise the log-likelihood function of $p(\mathbf{x})$, and we could rather use $p(\mathbf{x}, \mathbf{z})$. This idea defines a complete dataset $\{\mathbf{X}, \mathbf{Z}\}$, and in turn, a complete log-likelihood function that we can maximise. This maximisation of the complete log-likelihood function is straight-forward as the logarithm of Equation (5) is a sum of Gaussian logarithms, which is far easier to use. Naturally, we do not have access to $\mathbf{Z}$, otherwise we would not have tried to marginalise the variables out and used $p(\mathbf{x})$ in Equation (10). However, what we do have access to is the latent posterior conditional distribution $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$, and we can use this posterior distribution to our benefit.

As we do not have access to the complete data log-likelihood, we can consider its expected value under the posterior distribution of the latent variables. The notion here is that we take a model, evaluate the posterior distribution for each of the $n$ samples, and then collate these samples into a $\mathbf{Z}$ matrix.