

# 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

$$\mathbf{z} = [1, 0, 0, 0, 0]^T. \quad (1)$$

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  $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]^T$ , 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

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

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  $p(z_k = 1|\boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{z_k} = 1 \times \pi_k \times 1 \times \dots \times 1$ . 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

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

which can also be written as

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

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

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

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

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

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

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

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. This 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

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

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

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

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 responsible 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  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  and we wish to model this data using a GMM. We represent this dataset in a matrix  $\mathbf{X}$  where  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ . 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

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

where  $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$  and  $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K\}$ . Before we continue, however, there are three core issues with the GMM log-likelihood 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  $\frac{1}{|\boldsymbol{\Sigma}_k|}$ , 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  $\theta = \{\pi, \mu, \Sigma\}$  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  $\theta$ .

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  $K$  Gaussian 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 the  $n^{th}$  row represents  $\mathbf{z}_n^T$ . We can write the log-likelihood function in Equation (10) as

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

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 the joint distribution  $p(\mathbf{x}, \mathbf{z})$ . This idea requires a complete dataset  $\{\mathbf{X}, \mathbf{Z}\}$ , and in turn, provides a complete log-likelihood function that we can maximise. This log-likelihood function is given as

$$\ln p(\mathbf{X}, \mathbf{Z}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)], \quad (12)$$

where  $z_{nk}$  is a vector in the one-of- $K$  representation discussed earlier. For convenience, we can replace this with the indicator function  $\mathbf{1}_k(\mathbf{z}_n)$ , where

$$\mathbf{1}_k(\mathbf{z}_n) = \begin{cases} 1 & \text{if } z_{nk} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

Thus, Equation (12) can be written as

$$\ln p(\mathbf{X}, \mathbf{Z}) = \sum_{n=1}^N \sum_{k=1}^K \mathbf{1}_k(\mathbf{z}_n) [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)], \quad (14)$$

The maximisation of the complete log-likelihood function in Equation (12) 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  $p(\mathbf{z}|\mathbf{x}, \theta)$ , and we can use this posterior distribution to infer information about  $\mathbf{Z}$ . In this benefit, we can leverage on the mathematical benefits of the complete data log-likelihood, while still updating the model parameters to optimise the model to fit to the available data.

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 generative model, evaluate the posterior distribution  $p(\mathbf{Z}|\mathbf{X})$  for each of the  $n$  samples, and then collate these samples into a  $\mathbf{Z}$  matrix. In doing so, we have an estimation of  $\mathbf{Z}$ , that is a function of the current model parameter state  $\theta_{current}$ . As this inferred state depends on  $\theta$ , and we have at no stage made any assumptions as to whether this state is optimal, we also need to perform a maximisation of Equation (12) with respect to  $\theta$ . This idea is referred to as expectation maximisation, whereby we first determine the expectation of the complete data log-likelihood (the E-step) and then maximise this expectation (the M-step) with respect to the model parameters  $\theta$ .

More formally, we use the current model parameters  $\theta_{old}$  to compute  $p(\mathbf{Z}|\mathbf{X}, \theta_{old})$ , and use this posterior distribution to compute the expectation of the complete data log-likelihood for  $\theta$ .

This process is simple enough to describe, but I believe that the devil is within the details of the process. As such, let us try to compute the expectation of the complete data log-likelihood. The expected value of the Equation (14) can be written generally as

$$\begin{aligned}\mathbb{E}_{\mathbf{Z} \sim p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_{old})} \{\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})\} &= \int_{\mathbf{Z}} \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_{old}) d\mathbf{Z} \\ \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}_{old}) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_{old}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}),\end{aligned}\tag{15}$$

where the posterior distribution is clearly isolated, and showcases how we perform an iterative update scheme when using the EM algorithm. First, we find the posterior distribution  $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_{old})$  using the current model parameters  $\boldsymbol{\theta}_{old}$ . Then, we maximise the expectation of the complete data log-likelihood  $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}_{old})$  to update  $\boldsymbol{\theta}$ . However, this derivation simply shows how the posterior and joint distribution interact within the complete data log-likelihood. If we substitute and expand on the terms within the complete data likelihood, we obtain

$$\begin{aligned}\mathbb{E}_{\mathbf{Z} \sim p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})} \{\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})\} &= \mathbb{E}_{\mathbf{Z} \sim p(\mathbf{Z}|\mathbf{X})} \left\{ \sum_{n=1}^N \sum_{k=1}^K \mathbf{1}_k(\mathbf{z}_n) [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \right\} \\ &= \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}_{\mathbf{Z} \sim p(\mathbf{Z}|\mathbf{X})} \{ \mathbf{1}_k(\mathbf{z}_n) [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \} \\ &= \sum_{n=1}^N \sum_{k=1}^K [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \mathbb{E}_{\mathbf{Z} \sim p(\mathbf{Z}|\mathbf{X})} \{ \mathbf{1}_k(\mathbf{z}_n) \},\end{aligned}\tag{16}$$

which requires us to evaluate the expectation of the indicator variable with respect to the posterior distribution of the latent variables. If we expand this term by making use of Equation (8) the expectation of interest becomes

$$\begin{aligned}\mathbb{E}_{\mathbf{Z} \sim p(\mathbf{Z}|\mathbf{X})} \{ \mathbf{1}_k(\mathbf{z}_n) \} &= \int_{\mathbf{z}} \mathbf{z}_n p(\mathbf{z}_n | \mathbf{x}_n) d\mathbf{z} \\ &= \int_{\mathbf{z}} \mathbf{z}_n \frac{p(\mathbf{z}_n) p(\mathbf{x}_n | \mathbf{z}_n)}{p(\mathbf{x}_n)} d\mathbf{z} \\ &= \sum_{k=1}^K z_{nk} \frac{p(\mathbf{x}_n, \mathbf{z}_n)}{p(\mathbf{x}_n)} \\ &= \sum_{k=1}^K z_{nk} \frac{\prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)^{z_{nj}}} \\ &\text{as } \mathbf{z}_n \text{ has a 1-of-K representation with } z_k = 1, \text{ the result becomes} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \\ &= \gamma(z_{nk}).\end{aligned}\tag{17}$$

In this derivation I reversed some terms in the numerator of Bayes' theorem, to make it clear that Equation (5) was used, and introduced the  $\gamma(\cdot)$  variable to refer to the latent component responsibilities. There are numerous ways in which the term can be expanded, but I found this way to be the simplest to follow. Thus, the expectation of the complete data log-likelihood can be written as

$$\begin{aligned}\mathbb{E}_{\mathbf{Z} \sim p(\mathbf{Z}|\mathbf{X})} \{\ln p(\mathbf{X}, \mathbf{Z})\} &= \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \\ \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_{old}) &= \sum_{n=1}^N \sum_{k=1}^K p(z_k = 1 | \mathbf{x}_n, \boldsymbol{\theta}_{old}) [\ln p(\mathbf{x}_n, z_k = 1 | \boldsymbol{\theta})].\end{aligned}\tag{18}$$

Thus, we now have a tractable methodology to optimise a GMM for a given dataset. To make the steps clear, in case someone reading this might be struggling to place the full picture together, I have broken down this EM algorithm into two steps, namely:

**i) E step:** Evaluate the expectation of the complete data log-likelihood as given in Equation (18). This requires us to compute the responsibilities  $\gamma(z_{nk})$  using the current parameter set  $\theta_{old}$ .

**ii) M step:** Update the parameters by maximising Equation (18). This can be written formally as

$$\theta_{new} = \max_{\theta} \mathcal{L}(\theta, \theta_{old}). \quad (19)$$

We can then repeat these steps until we reach convergence. Convergence is measured by calculating the data log-likelihood given in Equation (10) after each parameter update and measuring the change from one iteration to the next.

### 3 Updating the model parameters

In this section, I will discuss how we optimise  $\mathcal{L}(\theta, \theta_{old})$  to obtain new parameters analytically. Naturally, there are a number of methods that we can try to optimise the parameters, such as numerical optimisation techniques, however it would be great if we could have analytical expressions for each of the terms in  $\theta \{\pi, \mu, \Sigma\}$ . As such, let us see what we can do with  $\mathcal{L}(\theta, \theta_{old})$ .

#### 3.1 Updating $\pi$

#### 3.2 Updating $\mu$

#### 3.3 Updating $\Sigma$

### 4 An alternative view on the data likelihood

In Equation (10), the log-likelihood function of the GMM model is given. It was shown and discussed that this objective function is complicated to optimise, and induces a number of problems to the extent where we needed to use the EM algorithm to help us.

However, there is an alternative derivation that one can use, that I found appear in a number of sources when I went through the derivation [CITE]. This derivation comes about by an alternative expansion of the joint distribution  $p(\mathbf{x}, \mathbf{z})$  when calculating  $p(\mathbf{x})$ . This expansion is as follows:

$$\begin{aligned} p(\mathbf{x}) &= \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \\ &= \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}) p(\mathbf{x}) d\mathbf{z} \end{aligned} \quad (20)$$

If we stop here, it is clear that we have just used an alternative <https://towardsdatascience.com/gaussian-mixture-models-and-expectation-maximization-a-full-explanation-50fa94111ddd>  
<https://wjchen.net/post/en/gmm-em-en.html>