Expectation Maximization (EM) and Gaussian Mixture Models (GMM)

A Gaussian mixture model can be written in the form:

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

Where $\sum \pi_k = 1$, and each Gaussian in the mixture has unique means/covariances. This is equivalent, in practice, to a 'complete' dataset consisting of both the measured x values and a set of latent z variables that describe which Gaussian in the mixture a single data point was drawn from. So, for each x, we have a paired unknown (latent) z. These can be drawn from a categorical distribution such that:

$$p(z_k = 1) = \pi_k \tag{2}$$

Where $\vec{z} = (z_1, z_2, ..., z_k)$, where k is the number of Gaussians/clusters in our feature space. Thus each z vector is a one-hot encoding of the mixtures, represented by a categorical distribution:

$$p(\vec{z}) = \prod_{k=1}^{K} \pi_k^{z_k} \tag{3}$$

Thus we can condition x on z:

$$p(\vec{x}|\vec{z}) = \prod_{k=1}^{K} \mathcal{N}(\vec{x}|\mu_k, \Sigma_k)^{z_k} \tag{4}$$

If we marginalize over the full join probability p(x,z), we will arrive back at the original mixture model we started with for p(x).

Another useful distribution is the probability for z, conditioned on x. Here we can use Bayes rule to find:

$$p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{p(x)} \tag{5}$$

This is an important value, called the responsibility of model k for example x, that indicates how likely one of the models in the mixture is to be responsible for producing x. It is given by:

$$\gamma(z_{nk}) \equiv \frac{\pi_k \mathcal{N}(\vec{x}_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\vec{x}_n|\mu_j, \Sigma_j)} \tag{6}$$

We wish to maximize the log-likelihood for p(x). For a set of N measured data points, the full probability becomes, for iid variables:

$$p(X|\pi,\mu,\Sigma) = \prod_{i=1}^{N}\sum_{k=1}^{K}\pi_k\mathcal{N}(\vec{x}_n|\mu_k\Sigma_k) \tag{7}$$

We can analytically solve for coupled-algebraic equations that solve the maximization of ln p(x). Taking derivatives with respect to μ_k lead to:

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\vec{x}_n \tag{8}$$

Where $N_k = \sum_{n=1}^{N}\gamma(z_{nk})$, and this is essentially a weighted average for μ, weighted by the proportion that each x appears to be represented by that particular mixture. Σ_k derivatives give us:

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(\vec{x}_n - \vec{\mu}_k)((\vec{x}_n - \vec{\mu}_k)^T \tag{9}$$

(a weighted covariance), while derivatives over π_k (constrained to $\sum\pi_k = 1$) lead to:

$$\pi_k = \frac{N_k}{N} \tag{10}$$

Although these are solutions for μ, Σ, π, they are all interconnected non-linearly through γ, and so we cannot write down a closed form solution as we can in, for example, OLS. We can, however, do an iterative approximation, as outlined in the problem.

Algorithm 1: EM Algorithm for GMM

Result: Optimized π, μ, Σ
initialize π, μ, Σ (possibly with k-means);
initialize iter = 0;
initialize converged = False;
while *iter* < *max_iter or not converged:* **do**
  E step: update γ;
  M step: update μ, Σ, π;
  if *converged:* **then**
    converged = True
  end
  iter+ = 1;
end

We may check for convergence using the log-likelihood value, or follow the convergence of the model parameters as well.

Part A: Implement EM Algorithm for GMMs

This algorithm is implemented for the Old Faithful dataset in the accompanying notebook. The GMM class tracks the evolution of all the parameters, but terminates if the log-likelihood stops evolving. We show plots for the evolution of π and μ, as well as the parameters of μ and Σ. We assume (with some foresight) that we have 2 clusters.

First we show the evolution of the Gaussian mixture in Fig. 1. We see that it fairly quickly converges on what looks like a reasonable solution for the final mixture, with the two clusters being properly identified.

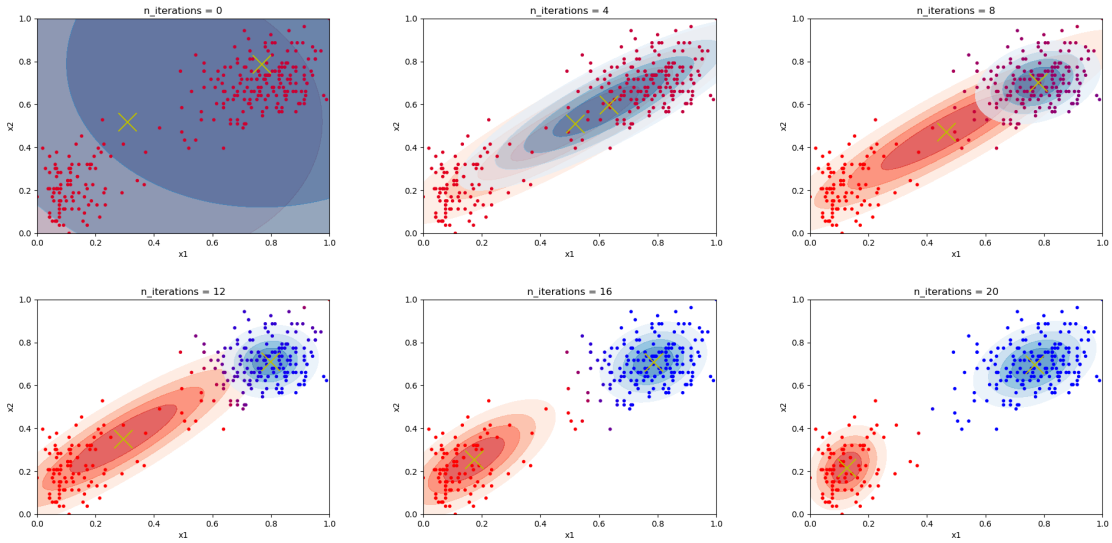

Figure 1: Convergence of the GMM algorithm for the Old Faithful dataset. Shown are 6 different iterations over the data, with the last being nearly fully converged. The shaded contours correspond to the underlying Gaussian mixtures, while the color of the data points corresponds to the calculated responsibilities.

Next we see convergence of both the log-likelihood and average cluster memberships (π_k) in Fig. 2, as well as convergence of the cluster means and covariances in Fig. 3.

Part B: Model Extensions

1. K-Means Equivalency

We can show that the K-Means model is an extreme case of a GMM, where instead of 'soft assignments' of the responsibilities (so that $\gamma(z_{nk}) \leq 1$), we specifically require $\gamma \to 1$. To see how this is possible, we remove the covariance as a floating parameter that we wish to

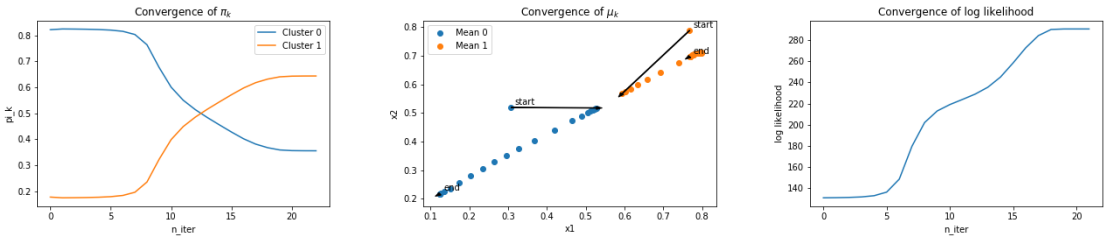

Figure 2: Convergence of the GMM algorithm for the Old Faithful dataset. Shown are plots showing convergence of π_k (left) and the log-likelihood (right) over 22 iterations (enough for the log-likelihood to stop changing appreciably). The middle plot shows the path of each of the two Gaussian means, eventually slowing down and converging on the final positions.

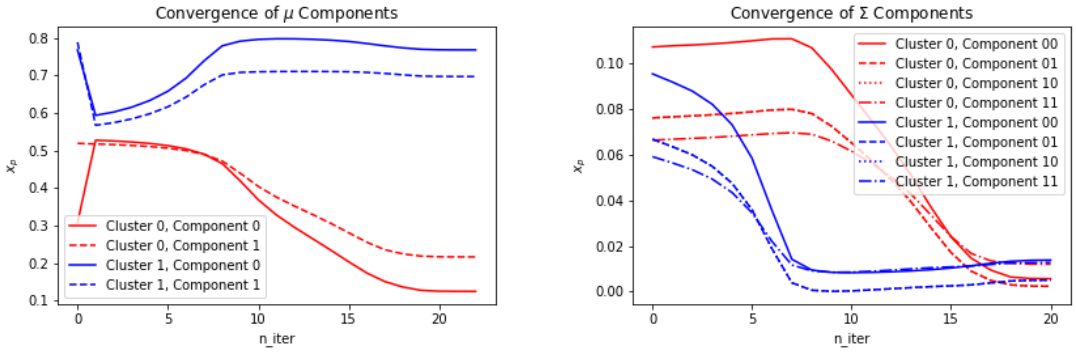

Figure 3: Convergence of the GMM algorithm for the Old Faithful dataset. Shown are plots showing convergence of μ_k (left) and Σ_k (right) over 22 iterations (enough for the log-likelihood to stop changing appreciably).

solve for, and instead fix it to be: $\Sigma_k = \epsilon\mathcal{I}, \forall k$. The responsibilities then become:

$$\gamma(z_{nk}) = \frac{\pi_k exp\{-\frac{1}{2\epsilon}||x_n - \mu_k||^2\}}{\sum_j \pi_j exp\{-\frac{1}{2\epsilon}||x_n - \mu_j||^2\}} \tag{11}$$

Consider the limit where $\epsilon \to 0$. Also consider all of the distances in the exponent: let d_{ns} be the shortest possible squared distance between a given x_n and every μ_k. Then, by construction:

$$\frac{exp(-\frac{1}{2\epsilon}d_{nk})}{exp(-\frac{1}{2\epsilon}d_{ns})} = exp\left(-\frac{1}{2\epsilon}(d_{nk} - d_{ns})\right) \to \delta_s^k \tag{12}$$

Where this goes to 0 for $k \neq s$ because $d_{nk} > d_{ns}\ \forall\ k$. If k = s, then of course the ratio goes

to 1. Thus, if we factor out the shortest distance exponent from our responsibilities, we find:

$$\begin{aligned}\gamma(z_{nk}) &= \frac{\pi_k exp\{-\frac{1}{2\epsilon}d_{nk}\}}{exp\{-\frac{1}{2\epsilon}d_{ns}\}\sum_j \pi_j d_s^j} \\ &= exp\left(-\frac{1}{2\epsilon}(d_{nk}-d_{ns})\right)\frac{\pi_k}{\pi_s} \\ &= \delta_s^k \end{aligned} \tag{13}$$

Thus $\gamma(z_{nk})$ goes to 1 for the closest centroid, and 0 otherwise, exactly like in K-means. Furthermore, this reduces the calculation of the mean to:

$$\begin{aligned}\mu_k &= \frac{1}{N_k}\sum_n \gamma(x_{nk})x_n \\ &= \frac{1}{N_k}\sum_n \delta_s^k x_n \\ &= \frac{1}{N_k}\sum_{n\in closest} x_n \end{aligned} \tag{14}$$

And

$$N_k = \sum_n \gamma(z_{nk}) = \sum_n \delta_s^k = \#\text{ assigned to cluster} \tag{15}$$

This is identical to the K-means formulation, where you take the average of those assigned to the cluster as the new centroids. Thus, we see that K-means is in fact a limiting case of the more general GMM-EM approximation.

2. T-Distribution Mixture Models

The EM algorithm can be applied to more than just Gaussian distributions of data. One example is the t-distribution, which is similar to the Gaussian and has the form:

$$f(x|v,\Sigma,\mu) = \frac{\Gamma(\frac{v+p}{2})}{\Gamma(\frac{v}{2})}(\pi v)^{-p/2}\det|\Sigma|^{-1/2}\left(1+\frac{t^2}{v}\right)^{-\frac{v+p}{2}} \tag{16}$$

Where $t^2 = (x-\mu)^T\Sigma^{-1}(x-\mu)$, v is the degrees of freedom of the distribution, p is the number of parameters (or dimensions) of the data, μ is the centre of the distribution, and Σ is related to the variance (while not exactly the variance, it is still positive semi-definite, which we will make use of later on). Γ are the standard gamma functions. In the limit that $v \to \infty$, this reduces exactly to a Gaussian distribution. For future reference, we also note

that log of this distribution is:

$$\begin{aligned}\log f(x|v,\Sigma,\mu) = &+ \log\Gamma\left(\frac{v+p}{2}\right) - \log\Gamma\left(\frac{v}{2}\right) - \frac{p}{2}\log\pi \\ &- \frac{p}{2}\log v - \frac{1}{2}\log\det|\Sigma| \\ &- \frac{v+p}{2}\log\left(1+\frac{t^2}{v}\right) \end{aligned} \tag{17}$$

To build up our EM method, we will need to know a few different probability distributions. Following the Gaussian Mixture Model methods, we use a latent variable z_i to tell us which t-distribution in the mixture observation x_i came from. Note that we will typically use i to indicate observations (and it ranges from 1 to N), while k indicates mixture components (and ranges from 1 to K). Thus we can write down:

$$p(x_i|z_i = k) = f(x_i|\theta_k) \tag{18}$$

Where for convenience we roll all of the t-distribution parameters into θ_k. Our latent variables z_i come from a categorical distribution:

$$p(z_i = k) = \pi_k \tag{19}$$

$$p(\vec{z_i}) = \prod_k \pi_k^{z_{ik}} \tag{20}$$

Where in the second line we have used a one-hot-encoding of z_i to create indicator variables for each z. That is, $z_{ik} = 1$ if $z_i = k$, and 0 otherwise. With these two probabilities, we can write down the posterior probability for both x and z:

$$\begin{aligned} p(x_i, z_i) &= p(x_i|z_i)p(z_i) \\ &= \prod_k \left(\pi_k f(x_i|\theta_k)\right)^{z_{ik}} \end{aligned} \tag{21}$$

Finally, we are also interested in the posterior probability for z, given that we actually know what the observations are:

$$p(z_i = k|x_i) = \frac{p(z_i = k)p(x_i|z_i)}{\sum_j p(z_i = j)p(x_i|z_i)} \tag{22}$$

Using our indicator variables, we write this as:

$$\gamma_{ik} \equiv p(z_{ik} = 1|x_i) = \frac{\pi_k f(x_i|\theta_k)}{\sum_k \pi_j f(x_i|\theta_j)} \tag{23}$$

These are the same form as the Gaussian Model, in that they are the responsibilities of each distribution for contribution to each observation (how probable was it that observation

x_i came from component distribution k?). If we include every observation in our posterior probability for x and z, we find (assume independent and identically distributed observations):

$$p(X, Z) = \prod_i \prod_k [\pi_k f(x_i|\theta_k)]^{z_{ik}} \tag{24}$$

Or in more useful logarithm form:

$$\log(P(X, Z)) = \sum_i \sum_k z_{ik} \log\left(\pi_k f(x_i|\theta_k)\right) \tag{25}$$

Now that we have the full posterior probability, we turn to the EM method. First we need to find the expectation of the log of $P(X, Z)$ with respect to the latent variables. That is the first step is to determine:

$$E_{Z|X}\left[\log(P(X, Z))\right] \tag{26}$$

To do this, we use the known, or old values of θ_k and π_k to calculate $P(Z|X)$ in the 'E' step. This is then used in the expectation for the log-likelihood (averaged over unknown latent variables), which we maximize with respect to θ_k and π_k to update our parameter values in the 'M' step. So let's deal with the expected value first, where we explicitly include θ now to keep track which ones are 'old' and which ones are 'new':

$$\begin{aligned} E_{Z|X,\theta^{old}}\left[\log(P(X, Z|\theta^{new}))\right] &= E_{Z|X,\theta^{old}}\left[\sum_i \sum_k z_{ik} \log\left(\pi_k^{new} f(x_i|\theta_k^{new})\right)\right] \\ &= \sum_i \sum_k E_{Z|X,\theta^{old}}\left[z_{ik} \log\left(\pi_k^{new} f(x_i|\theta_k^{new})\right)\right] \\ &= \sum_i \sum_k E_{Z|X,\theta^{old}}\left[z_{ik}\right] \log\left(\pi_k^{new} f(x_i|\theta_k^{new})\right) \end{aligned} \tag{27}$$

Where we have used the linearity of expectation values to reduce this to only requiring the expected value of z_{ik}. However, because this is an indicator variable, its expected valuable is simply equal to its probability, and so:

$$E_{Z|X,\theta^{old}}\left[z_{ik}\right] = \gamma_{ik}^{old} \tag{28}$$

As we defined above, using the old versions of π_k and θ_k (although moving forward I'll drop the old from γ, as it should always be understood to be calculated first before continuing further). Thus, our expected value for the log-likelihood becomes:

$$E_{Z|X,\theta^{old}}\left[\log(P(X, Z|\theta^{new}))\right] = \sum_i \sum_k \gamma_{ik}\left[\log\left(\pi_k^{new}\right) + \log\left(f(x_i|\theta_k^{new})\right)\right] \tag{29}$$

Now, we wish to maximize this with respect to the parameters of the model. (At this point, I'm also dropping the new off the parameters, as it should be understood that we are maximizing with respect to these. I'm also just going to call the entire expecation value E for convenience). Let's start off simple. If we take derivatives with respect to π_k (where we include the constraint that $\sum_k \pi_k = 1$ via a Lagrange multiplier), we find:

$$\frac{\partial(E + \lambda(\sum_j \pi_j - 1))}{\partial \pi_k} = \sum_i \frac{\gamma_{ik}}{\pi_k} + \lambda = 0 \tag{30}$$

Multiplying by π_k and then summing over k leads to:

$$\sum_i \sum_k \gamma_{ik} + \lambda \sum_k \pi_k = 0$$

$$\lambda = -\sum_i \sum_k \frac{\pi_k f(x_i|\theta_k)}{\sum_k \pi_j f(x_i|\theta_j)} = -\sum_i 1 = -N \tag{31}$$

And so $\lambda = -N$. We can substitute this back in to find π_k:

$$\pi_k = \frac{\sum_i \gamma_{ik}}{N} = \frac{N_k}{N} \tag{32}$$

Where we define N_k the same as before.

Next up, let's consider the mean values, μ_k:

$$\begin{aligned}
\frac{\partial E}{\partial \mu_k} &= \sum_i \gamma_{ik} \frac{\partial \log f}{\partial \mu_k} \\
&= \sum_i \gamma_{ik} \frac{-(p+v)}{2} \frac{v}{v+t^2} \frac{1}{v} \frac{\partial t^2}{\partial \mu_k} \\
&= \sum_i \gamma_{ik} \frac{-(p+v)}{2} \frac{1}{v+t^2} (-2\Sigma_k^{-1}(x_i - \mu_k)) \\
&= \sum_i \gamma_{ik} \frac{(p+v)}{v+t^2} (\Sigma_k^{-1}(x_i - \mu_k)) \\
&= \sum_i \gamma_{ik} u_{ik} (\Sigma_k^{-1}(x_i - \mu_k)) = 0
\end{aligned} \tag{33}$$

Where we have used the vector derivative:

$$\frac{\partial x^T a}{\partial x} = \frac{\partial a^T x}{\partial x} = a \tag{34}$$

To do the t^2 derivative. Here we have also introduced another new parameter, u_{ij} that will show up more than once:

$$u_{ik} = \frac{p + v_k}{v_k + t_{ik}^2} \tag{35}$$

We can multiply from the left by Σ_k to remove the variance, leaving us with:

$$\sum_i \gamma_{ik} u_{ik} (x_i - \mu_k) = 0 \tag{36}$$

Note that u_{ik} technically also depends on μ_k, but this is a much weaker dependence than anywhere else in the equation (similar to how the equation for Σ in the GMM technically depends on μ_k as well, but we can iteratively solve for μ_k first, and then Σ second. Here we just do u_{ik} first, and then μ_k, similar to calculating γ_{ik}.) [1]. Thus to first order, we can write:

$$\mu_k = \frac{\sum_i \gamma_{ik} u_{ik} x_i}{\sum_i \gamma_{ik} u_{ik}} \tag{37}$$

This makes sense, as we have a weighted average for μ_k with updated weights $\gamma_{ik} u_{ik}$.

Now let's do Σ:

$$\begin{aligned}
\frac{\partial E}{\partial \Sigma_k} &= \sum_i \gamma_{ik} \frac{\partial \log f}{\partial \Sigma_k} \\
&= \sum_i \gamma_{ik} \left(-\frac{1}{2} \frac{1}{\det \Sigma_k} \frac{\partial \det \Sigma_k}{\partial \Sigma_k} - \frac{(p+v)}{2} \frac{v}{v+t^2} \frac{1}{v} \frac{\partial t^2}{\partial \Sigma_k} \right) \\
&= \sum_i \gamma_{ik} \frac{1}{2} \left(-\frac{1}{\det \Sigma_k} (\det \Sigma_k \cdot \Sigma_k^{-T}) + \frac{(p+v)}{v+t^2} \Sigma_k^{-T} (x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-T} \right) \\
&= \sum_i \gamma_{ik} \frac{1}{2} \left(-\Sigma_k^{-1} + u_{ik} \Sigma_k^{-1} (x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-1} \right) = 0
\end{aligned} \tag{38}$$

Here we have used the following matrix derivative properties:

$$\frac{\partial \det X}{\partial X} = \det X \cdot X^{-1T} \tag{39}$$

$$\frac{\partial a^T X^{-1} b}{\partial X} = -X^{-T} a b^T X^{-T} \tag{40}$$

We have also used the fact that Σ_k is symmetric (and will remain symmetric upon each update, as we shall see) to remove the transpose from Σ.

Now, we multiply from the left and right by Σ_k and factor out the 2 to find:

$$\sum_i \gamma_{ik} \left(-\Sigma_k + u_{ik} (x_i - \mu_k)(x_i - \mu_k)^T \right) = 0 \tag{41}$$

[1] In fact, if you use Newtons method to calculate the first update to $\mu_k = \mu_k^{old} - f'(\mu_k)/f(\mu_k)$, you will find the exact same equation as the above if you assume the changes due to u_{ik} are small relative to everywhere else

Or, solving for Σ, we get:

$$\Sigma_k = \frac{\sum_i \gamma_{ik} u_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i \gamma_{ik}} \tag{42}$$

This is not quite a weighted average, and might lead to underestimates of the variance (which we shall see later on). It might be worth playing with making this a full weighted average, and including u_{ik} in the denominator as well to avoid biasing this parameter.

Finally, we come to the most convoluted parameter, the degrees of freedom of each distribution. Let's take derivatives with respect to v:

$$\begin{aligned}
\frac{\partial E}{\partial v_k} &= \sum_i \gamma_{ik} \frac{\partial \log f}{\partial v_k} \\
&= \sum_i \gamma_{ik} \frac{1}{2} \left(\psi\left(\frac{v+p}{2}\right) - \psi\left(\frac{v}{2}\right) - \frac{p}{v} - \log\left(\frac{v+t^2}{v}\right) + \frac{v+p}{v+t^2}\frac{t^2}{v} \right)
\end{aligned} \tag{43}$$

Where $\psi(x) = \partial \log \Gamma(x)/\partial x$ is the digamma function. Let's clean this up a little bit. Consider the polynomial pieces:

$$\begin{aligned}
-\frac{p}{v} + \frac{v+p}{v+t^2}\frac{t^2}{v} &= \frac{(v+p)t^2 - p(v+t^2)}{(v+t^2)v} - 1 + 1 \\
&= \frac{(v+p)t^2 - p(v+t^2) - v(v+t^2)}{(v+t^2)v} + 1 \\
&= \frac{(v+p)(t^2 - (v+t^2))}{(v+t^2)v} + 1 \\
&= -\frac{v+p}{v+t^2} + 1 \\
&= -u_{ik} + 1
\end{aligned} \tag{44}$$

Also consider the log piece:

$$\begin{aligned}
\log\left(\frac{v+t^2}{v}\right) &= \log\left(\frac{v+t^2}{v}\right) + log(v+p) - log(v+p) \\
&= \log\left(\frac{v+t^2}{v+p}\right) + log(v+p) - log(v) \\
&= -\log u_{ik} + log\frac{v+p}{2} - log\frac{v}{2}
\end{aligned} \tag{45}$$

Putting this back together we get:

$$\frac{\partial E}{\partial v_k} = \sum_i \gamma_{ik} \frac{1}{2} \left(\psi\left(\frac{v+p}{2}\right) - \log\frac{v+p}{2} - \psi\left(\frac{v}{2}\right) + \log\frac{v}{2} + 1 + \log u_{ik} - u_{ik} \right) \tag{46}$$

Setting this to zero and factoring out N_k for terms that don't depend on i we find the final equation for v_k:

$$\psi\left(\frac{v_k+p}{2}\right) - \log\frac{v_k+p}{2} - \psi\left(\frac{v_k}{2}\right) + \log\frac{v_k}{2} + \frac{1}{N_k}\sum_i \gamma_{ik}\left(\log u_{ik} - u_{ik}\right) = 0 \tag{47}$$

This is highly non-analytic, and the behaviour of the digamma function makes it difficult to solve for v_k. There is no linearly available v_k to approximate, as in the case of the means previously. Some papers (see for example, Peel et. al, 2000 [2]) suggest setting v_k beforehand, as then the M step exists in completely closed form, as we have analytic solutions for μ_k, Σ_k and π_k. They also note that the search for v is time-consuming and does not necessarily lead to better results. Thus for now, I make note of the solution for v_k, but will work with the algorithm holding v fixed (and as I am testing it on t-distributions I have set up myself, then I already know v in advance and can simply set it to the correct value from the beginning).

Thus we finally have an algorithm for the EM method for t-distribution mixture models (TMM):

Algorithm 2: EM Algorithm for TMM

Result: Optimized π, μ, Σ
initialize π, μ, Σ, v (possibly with k-means);
initialize iter = 0;
initialize converged = False;
while *iter* < *max_iter or not converged:* **do**
  E step: update γ_{ik}, u_{ik};
    $\gamma_{ik} = \frac{\pi_k f(x_i|\theta_k)}{\sum_k \pi_j f(x_i|\theta_j)}$;
    $u_{ik} = \frac{p+v_k}{v_k+t_{ik}^2}$;
  M step: update μ, Σ, π;
    $\pi_k = N_k/N$;
    $\mu_k = \frac{\sum_i \gamma_{ik}u_{ik}x_i}{\sum_i \gamma_{ik}u_{ik}}$;
    $\Sigma_k = \frac{\sum_i \gamma_{ik}u_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_i \gamma_{ik}}$;
  if *converged:* **then**
    converged = True
  end
  iter+ = 1;
end

This algorithm is implemented in the accompanying notebook. The TMM class follows the same methodology as the GMM class implemented previously. Here instead, we test it on some data that was randomly drawn from a 1-d mixture of 5 t-distributions. The initial, randomly drawn sample was created by first randomly choosing 1 of the t-distributions

[2] Robust Mixture Modelling using the t-distribution, Peel et. al, 2000.

according to a distribution of π_k (chosen arbitrarily). Then from the t-distribution, a sample is randomly drawn. This is repeated until we have $n_{samples}$. A reasonably large sample of 1000 was chosen. A value of v=5 was arbitrarily chosen for each distribution (hopefully small enough so that they don't look identical to Gaussian models). The results are shown as a histogram in Fig. 4.

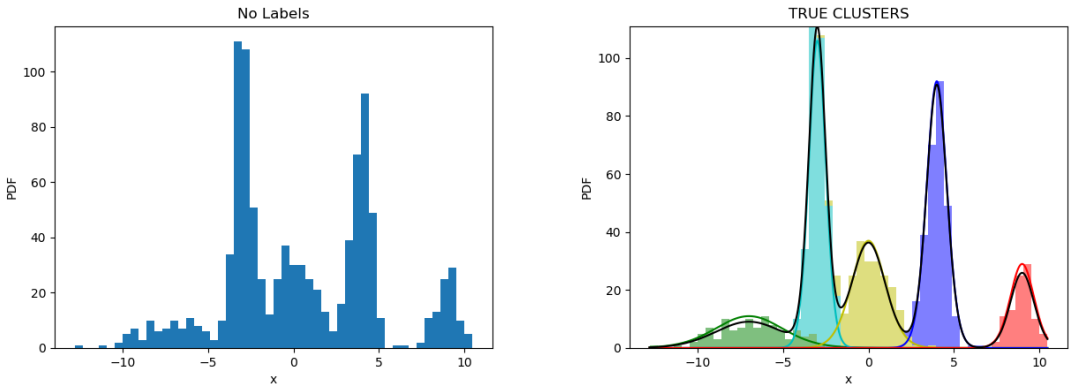

Figure 4: Sampled data to produce a mixture of t-distributions. On the left, the data is shown without labels, while on the right the clusters from which each sample were drawn are coloured as well. The black line is the overall probability distribution (scaled to match the histogram of the samples).

Next we apply the TMM algorithm. The algorithm is shown at various iterations during the sequence in Fig. 5. It's interesting to note that it appears to miss the central peak through the first few iterations, but catches it later on. However, this seems to be a problem with the EM model in general: if the peak had been slightly smaller, the nearby distribution could easily mask the smaller signal, and a fit could be found that just includes both bumps under one broader (increased Σ) and more probable (increased π) distribution. This almost happened as we can see in iteration 10, before it found the smaller centre. Poorer initializations of the parameters may often lead to missed peaks, highlighting the importance of ensembling even in simple scenarios[3].

As before, we can look at how well all of the parameters are converging. In Fig. 6, we see that the log-likelihood has plateaued, as have the π values, spread roughly between 10 and 40% per distribution. The parameters of the t-distributions are also converging, as we see in Fig. 7.

Finally, the moment of truth: how well do the results stack up with the input parameters? The results are shown in Fig. 8. We see that the mean positions of the distributions match fairly well, with the largest mean difference being 5% (not counting the mean that was initialized at 0). The weights do reasonably as well, with all the differences being under 15%. However, we see that this method tends to underestimate the variances (like we mentioned

[3]Yes, I had to run this algorithm more than once before it actually latched on to all 5 distributions.

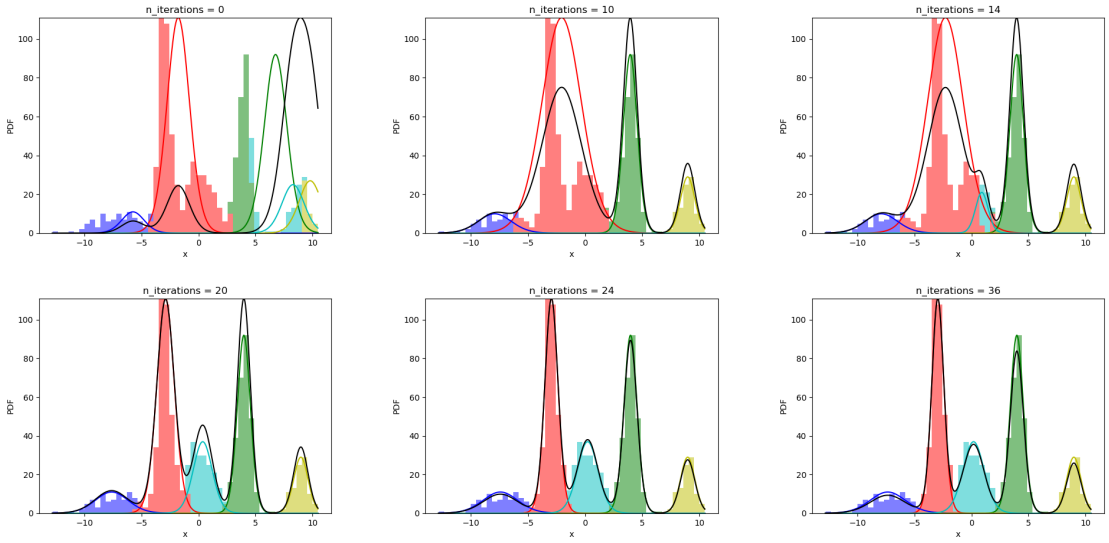

Figure 5: Convergence of the TMM algorithm for the t-distribution mixture model. Shown are 6 different iterations over the data, with the last being nearly fully converged. The coloured lines correspond to the underlying t-distribution mixtures (scaled to be able to see them), while the black line shows the current overall distribution model (also scaled to fit the underlying histograms). The colours of the underlying histograms correspond to the t-distribution with the largest responsibility.

earlier), with some of them being up to 50% smaller than the true values. This seems reasonable however, as at each step, the model is essentially attempting to shrink itself around each peak, and so naturally the widths of the models will shrink to accommodate this.

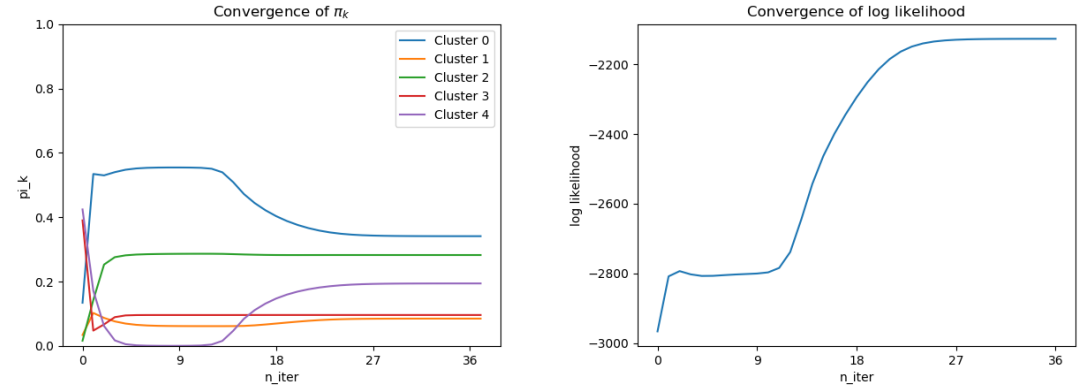

Figure 6: Convergence of the TMM algorithm for the simulated dataset. Shown are plots showing convergence of π_k (left) and the log-likelihood (right) over 36 iterations (enough for the log-likelihood to stop changing appreciably).

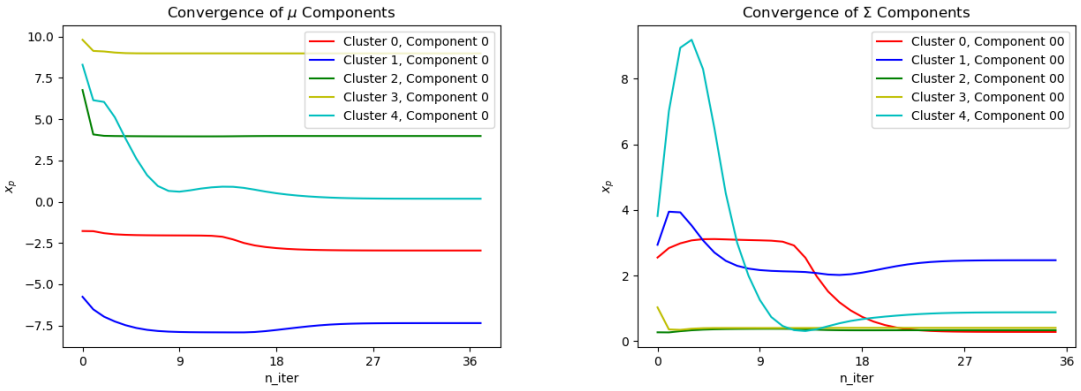

Figure 7: Convergence of the TMM algorithm for the simulated dataset. Shown are plots showing convergence of μ_k (left) and Σ_k (right) over 36 iterations (enough for the log-likelihood to stop changing appreciably).

```
-------------------------------------
Cluster 0
        True    Expected   % Difference
Mean    -7.00   -7.35      0.05
Sigma    2.00    2.47      0.24
Weight   0.10    0.09      0.15
-------------------------------------
Cluster 1
        True    Expected   % Difference
Mean    -3.00   -2.96      0.01
Sigma    0.50    0.28      0.43
Weight   0.30    0.34      0.14
-------------------------------------
Cluster 2
        True    Expected   % Difference
Mean     0.00    0.18      inf
Sigma    1.00    0.88      0.12
Weight   0.20    0.19      0.03
-------------------------------------
Cluster 3
        True    Expected   % Difference
Mean     4.00    3.98      0.01
Sigma    0.60    0.34      0.44
Weight   0.30    0.28      0.06
-------------------------------------
Cluster 4
        True    Expected   % Difference
Mean     9.00    8.98      0.00
Sigma    0.70    0.41      0.42
Weight   0.10    0.10      0.04
```

Figure 8: Comparison of the TMM algorithm results with the true underlying values for the simulated dataset. Weight corresponds to the parameter π, with the other two self-explanatory.