Gaussian mixture model

Zhe Feng

July 3, 2020

1 Introduction

Gaussian mixture model (GMM) is a very interesting model and itself has many applications, though outshined more advanced models recently, it still serve as a good base model for clustering and serve as good stepping stone to understand more complicated models such as hidden markoe model and it is also tightly related to expectation maximisation algorithm (EM-algorithm), a family of algorithms that is behand many statistical models.

2 Problem description

2.1 Model form

GMM can be described by the following formula:

$$\begin{aligned} P(x; \boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_{k=1}^{K} \tau_k p_k(x; \mu_k, \Sigma_k) \\ p_k(x; \mu_k, \Sigma_k) &= (2\pi)^{-\frac{n}{2}} \det |\Sigma_k|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\right] \end{aligned} \tag{1}$$

where

- $x \in \mathbb{R}^n$ is the n dimensional random variable observed
- $P : \mathbb{R}^n \to [0, 1]$ is the GMM probability density function (i.e. our model)
- $\boldsymbol{\mu} = \{\mu_1, \mu_2 \cdots, \mu_K\}$ and $\mu_k \in \mathbb{R}^n$ is the mean of the kth Gaussian component
- $\boldsymbol{\Sigma} = \{\Sigma_1, \Sigma_3 \cdots, \Sigma_K\}$ and $\Sigma_k \in \mathbb{R}^{n\times n}$ is the covariance matrix of the ith Gaussian component

- $\boldsymbol{\tau} = \{\tau_1, \tau_2, \cdots, \tau_K\}$ and $\tau_k \in \mathbb{R}$ is the weight of the kth component and $\sum_k^K \tau_k = 1$, it is also called the mixing parameter. The item τ can be considered as the prior probability of a hidden state z_k

With this formulation, similar to the 3 fundamental problems of hidden Markov model (HMM), we are interested in solving the following problems:

1. **Learning problem:** given a set of N observations $X = \{x_1, x_2, \cdots, x_N\}$, what is the most likely model $P(\cdot; \boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$
2. **Prediction problem:** given a model $P(\cdot; \boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and an observation x, what is the probability of x is generated by the kth component?
3. **Evaluation problem:** given a set of N observations $X = \{x_1, x_2, \cdots, x_N\}$ and a model $P(\cdot; \boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, what is the probability X is generated by the model P.

The 3rd problem is probably less interested by people and the the 2nd problem is in general trivial to solve for GMM. We mainly care about the first learning problem.

2.2 The learning problem and the optimisation problem under the hood

In terms of learning problem, as stated in the last section, we are given a set of observations $X = \{x_1, x_2 \cdots x_N\}$ and we want to find a set of model parameters $\boldsymbol{\theta} = \{\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ such that the likelihood of observing the data set X is maximised. Mathematically:

$$\max_{\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}} L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; X) = \prod_{i=1}^{N} \sum_{k=1}^{K} \tau_k p_k(x_i; \mu_k, \Sigma_k) \tag{2}$$

3 Solve the optimisation problem

We can of course try to solve the nonlinear optimisation problem with some generic numerical nonlinear programming solvers such as interior point method, but it would be relatively slow and less robust. Instead, people use Expectation-Maximisation (EM) algorithm to solve problem like this.

3.1 EM-algorithm

To solve this problem with EM algorithm, we need to reformat the problem (1) a bit. Assume GMM is a generative model with a latent variable $z = \{1, 2 \cdots K\}$ indicates which gaussian component is 'activated' and the probability of a data point x

is generated by the kth component is $P(z = k) = \tau_k$, similart to X, we can define $Z = \{z_1, z_2 \cdots z_N\}$ then the likelihood function can be written as:

$$L(\boldsymbol{\mu}, \boldsymbol{\Sigma}; X, Z) = \prod_{i=1}^{N} \prod_{k=1}^{K} [\tau_k p_k(x_i; \mu_k, \Sigma_k)]^{\mathbb{I}(z_i=k)} \tag{3}$$

where $\mathbb{I}(z_i = k)$ is the indicator function which equals 1 if $z_i = k$ and 0 otherwise. Note the above likelihood can also be expressed as $\prod_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}(z_i = k) [\tau_k p_k(x_i; \mu_k, \Sigma_k)]$, but in this way, it will cause trouble in the log-likelihood function, which is:

$$\log L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; X, Z) = \log \prod_{i=1}^{N} \prod_{k=1}^{K} [\tau_k p_k(x_i; \mu_k, \Sigma_k)]^{\mathbb{I}(z_i=k)} \tag{4}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}(z_i = k) [\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)] \tag{5}$$

To simiplify notation a bit (and also makes it a bit more general), we define model parameters $\theta := \{\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

Now our optimisation problem become:

$$\max_{\theta} \log L(\theta; X, Z) = \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}(z_i = k) [\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)] \tag{6}$$

3.1.1 The E-step

In EM algorithm, the E-step, or the expectation step is to take the expectation of the log-likelihood function over hiddden variable Z, i.e. find $Q(\theta; X) := E_{Z|X}[\log L(\theta; X, Z)]$. Note that the expected log-likelihood function $Q(\theta; X)$ is a function of model parameter θ parameterised by X, not Z as it is averaged over Z.

$$Q(\theta; X) := E_{Z|X}[\log L(\theta; X, Z)] \tag{7}$$

$$= E_{Z|X} \left[\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}(z_i = k) [\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)] \right] \tag{8}$$

$$= \sum_{i=1}^{N} E_{z_i|x_i} \left[\sum_{k=1}^{K} \mathbb{I}(z_i = k) [\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)] \right] \tag{9}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} P(z_i = k|x_i) [\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)] \tag{10}$$

Now we need to calculate the posterior of z_i:

$$T_{k,i} := P(z_i = k|x_i) = \frac{P(x_i|z_i = k)P(z_i = k)}{P(x_i)} \tag{11}$$

$$= \frac{P(x_i|z_i = k)P(z_i = k)}{\sum_{k=1}^{K} P(x_i|z_i = k)P(z_i = k)} \tag{12}$$

substitute $P(x_i|z_i = k) = p_k(x_i; \mu_k, \Sigma_k)$ and $P(z_i = k) = \tau_k$, we have:

$$\boxed{T_{k,i} = \frac{p_k(x_i; \mu_k, \Sigma_k)\tau_k}{\sum_{k=1}^{K} p_k(x_i; \mu_k, \Sigma_k)\tau_k}} \tag{13}$$

With the posterior $P(z_i = k|x_i) = T_{k,i}$ calculated, which is the main output of the E-step algorithmically, we can evaluate the expected log-likelihood $Q(\theta|X)$, and this is the objective function in the M-step for maximisation.

3.1.2 The M-step

The M-step, or the maximisation step, is to maximize the expected log-likelihood function $Q(\theta; X)$ w.r.t. θ:

$$\theta^* = \arg\max_{\theta} Q(\theta; X) = \sum_{i=1}^{N}\sum_{k=1}^{K} T_{k,i} \left[\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)\right] \tag{14}$$

recall that $\theta = \{\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and

$$p_k(x_i; \mu_k, \Sigma_k) = (2\pi)^{-\frac{n}{2}} \det |\Sigma_k|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(x - \mu_k)^\top \Sigma_k^{-1}(x - \mu_k)\right]$$

which is the Gaussain distribution. This optimisation problem can be solved analytically by setting the derivative to 0. We first substitute the gaussian pdf in the above equation and we have

$$\begin{aligned} Q(\theta|X) = \sum_{i=1}^{N}\sum_{k=1}^{K} T_{k,i}[&\log \tau_k + \log(2\pi)^{-\frac{n}{2}} + \log\det|\Sigma_k|^{-\frac{1}{2}} \\ &- \frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)] \end{aligned}$$

Calculate $\boldsymbol{\tau}^*$

To calculate τ^* we take derivative against Q, however with one twist: we need to have the constraint $\sum_k^K \tau_k = 1$, so equivaliently we are solving

$$\boldsymbol{\tau}^*, \lambda^* = \arg\max_{\boldsymbol{\tau},\lambda} \hat{Q}(\boldsymbol{\tau}, \lambda) := \sum_{i=1}^{N}\sum_{k=1}^{K} T_{k,i} \log \tau_k + \lambda(1 - \sum_{k=1}^{K} \tau_k) \tag{15}$$

where λ is the Lagrange multiplier. Note that technically speaking we also need the constraint $\tau_k \geq 0$, but this constraint is satisfied automatically since we have $T_{k,i} > 0$ and the above equation is linear combination. The above is also not 100% rigourous because the multiplier shoudl also applied to the original objective function, the only reason we can do this is because other decision variables does not concern λ.

The derivative w.r.t. τ_k is:

$$\frac{\partial}{\partial \tau_k} \hat{Q}(\tau_k, \lambda) = \frac{\sum_{i=1}^{N} T_{k,i}}{\tau_k} + \lambda \tag{16}$$

$$\frac{\partial}{\partial \lambda} \hat{Q}(\tau_k, \lambda) = 1 - \sum_{k=1}^{K} \tau_k \tag{17}$$

The optimal solution is obtained at where gradient vanished, thus τ_k^* and λ^*

$$0 = \frac{\sum_{i=1}^{N} T_{k,i}}{\tau_k}^* - \lambda^* \tag{18}$$

$$0 = 1 - \sum_{j \neq k}^{K} \tau_j^* + \tau_k^* \tag{19}$$

Note k here is not used for indexing, but represent a specific index (1, 2 etc.) and j is used for indexing sums. From the above we have:

$$\tau_k^* = \frac{\sum_{i=1}^{N} T_{k,i}}{\lambda^*} \; \forall k \tag{20}$$

Now we need a separate equation to determine λ^*, this can be done by substitue equation (20) into (19):

$$0 = 1 - \sum_{k=0}^{K} \frac{\sum_{i=1}^{N} T_{k,i}}{\lambda^*} \tag{21}$$

$$\therefore \lambda^* = \sum_{i=1}^{N} \sum_{k=0}^{K} T_{k,i} = N \tag{22}$$

note we used the fact $\sum_{k=0}^{K} T_{k,i} = 1$ because $T_{k,i}$ is the posterior for z_i and it sums to 1 along k. Eventually, we have our optimal $\boldsymbol{\tau}^*$:

$$\boxed{\boldsymbol{\tau}^* = [\tau_k^*]_{k=1:K} = \frac{1}{N} \sum_{i=1}^{N} T_{k,i} \; \forall k} \tag{23}$$

Interestingly, if we compare the unconstrained solution, which is $\sum_i^N T_{k,i}$, the constrained solution is the normalised version of it. We can also see, that the optimal value

(in terms of maximum likelihood estimator) of prior of z_i, τ_k is the expected value of the posterior of z_i, $T_{k,i}$ averaged over all observations (This is also essentially the MLE solution for binomial distribution).

Calculate $\boldsymbol{\mu}^*$
Following similar principle, taking derivative of Q w.r.t. $\boldsymbol{\mu}$ we have:

$$\frac{\partial}{\partial \mu_k} Q(\mu_k) = \frac{\partial}{\partial \mu_k} \sum_{i=1}^{N} \sum_{j=1}^{K} -T_{j,i} \frac{1}{2} (x_i - \mu_j)^\top \Sigma_k^{-1} (x_i - \mu_j) \tag{24}$$

$$= \sum_{i=1}^{N} -T_{k,i} \Sigma_k^{-1} (x_i - \mu_k) \tag{25}$$

agian, we use k here only for a specific component and use j as the sum index. By setting derivative to 0 we have:

$$0 = \sum_{i=1}^{N} -T_{k,i} \Sigma_k^{-1} (x - \mu_k) \tag{26}$$

$$\Sigma_k^{-1} \sum_{i=1}^{N} T_{k,i} \mu_k = \Sigma_k^{-1} \sum_{i=1}^{N} T_{k,i} x_i \tag{27}$$

therefore by rearange the above equation, we have our optimal $\boldsymbol{\mu}^*$:

$$\boxed{\boldsymbol{\mu}^* = [\mu_k^*]_{k=1:K} = \frac{\sum_{i=1}^{N} T_{k,i} x_i}{\sum_{i=1}^{N} T_{k,i}} \ \forall k} \tag{28}$$

Compare to the maximum likelihood estimation of mean for gaussian distribution $\frac{1}{N} \sum_i^N x_i$, the above estimate is equivalent of a 'soft count' or a 'weighted count' w.r.t. to the posterior probability of z_i version of it.

Calculate $\boldsymbol{\Sigma}^*$
The estimation of the optimal covariance matrix $\boldsymbol{\Sigma}^*$ is the trickest part. First we will need some property about matrices, the following equalities are from fundamental linear algebra and we will not derive them here, but use them as they are:

- property of trace (cyclic product's trace are the same):

$$\text{tr}[ABC] = \text{tr}[BCA] = \text{tr}[CAB] \tag{29}$$

- derivative of trace (similar to derivative of linear combination):

$$\frac{\partial}{\partial A} \text{tr}[BA] = B^\top \tag{30}$$

- derivative of quadratic form (used the above to properties):

$$\frac{\partial}{\partial A} x^\top A x = \frac{\partial}{\partial A} \text{tr}[x^\top A x] = \frac{\partial}{\partial A} \text{tr}[x x^\top A] = [x x^\top]^\top = x x^\top \tag{31}$$

- derivative of log of determinant:

$$\frac{\partial}{\partial A} \log\det|A| = (A^\top)^{-1} = A^{-1} \tag{32}$$

where A is a symmetric matrix

Now if we take derivative of Q w.r.t. $\mathbf{\Sigma}$:

$$\frac{\partial}{\partial \Sigma_k} Q(\Sigma_k) = \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \frac{\partial}{\partial \Sigma_k^{-1}} \sum_{i=1}^{N} \sum_{j=1}^{K} T_{k,i}[\log\det|\Sigma_k|^{-\frac{1}{2}} - \frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)] \tag{33}$$

$$= \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \frac{\partial}{\partial \Sigma_k^{-1}} \sum_{i=1}^{N} T_{k,i}[\frac{1}{2}\log\det|\Sigma_k|^{-1} - \frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)] \tag{34}$$

$$\tag{35}$$

Now with equation (31) for quadratic form and equation (32) for log of determinant, we have:

$$\frac{\partial}{\partial \Sigma_k} Q(\Sigma_k) = \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \frac{\partial}{\partial \Sigma_k^{-1}} \sum_{i=1}^{N} T_{k,i}[\frac{1}{2}\log\det|\Sigma_k|^{-1} - \frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)] \tag{36}$$

$$= \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \sum_{i=1}^{N} T_{k,i}\frac{1}{2}\Sigma_k - T_{k,i}\frac{1}{2}(x_i - \mu_k)(x_i - \mu_k)^\top \tag{37}$$

Note that $(x_i - \mu_k)(x_i - \mu_k)^\top \in \mathbb{R}^{n\times n}$ is the outer product, also we only caculated $\frac{\partial Q(\Sigma_k)}{\partial \Sigma_k^{-1}}$ and left $\frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k}$ as it is, because when setting derivative to 0, this term will cancel out:

$$0 = \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \sum_{i=1}^{N} T_{k,i}\frac{1}{2}\Sigma_k - T_{k,i}\frac{1}{2}(x_i - \mu_k)(x_i - \mu_k)^\top \tag{38}$$

$$0 = \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \sum_{i=1}^{N} T_{k,i}\frac{1}{2}\Sigma_k - \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \sum_{i=1}^{N} T_{k,i}\frac{1}{2}(x_i - \mu_k)(x_i - \mu_k)^\top \tag{39}$$

$$\frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \sum_{i=1}^{N} T_{k,i}\frac{1}{2}\Sigma_k = \frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k} \sum_{i=1}^{N} T_{k,i}\frac{1}{2}(x_i - \mu_k)(x_i - \mu_k)^\top \tag{40}$$

$$\tag{41}$$

Now we can see $\frac{\partial \Sigma_k^{-1}}{\partial \Sigma_k}$ can be cancelled out on both sides. The optimal $\mathbf{\Sigma}$ is then:

$$\boxed{\mathbf{\Sigma}^* = [\Sigma_k]_{k=1:K} = \frac{\sum_{i=1}^{N} T_{k,i}(x_i - \mu_k)(x_i - \mu_k)^\top}{\sum_{i=1}^{N} T_{k,i}}} \tag{42}$$

3.2 EM-algorithm for computing Gaussiam Mixture Model

In summary, the algorithm can be summarised as the following:

1. Initialise with random $\boldsymbol{\tau}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$
2. take E-step, calculate $T_{k,i}$:

$$\boxed{T_{k,i} = \frac{p_k(x_i; \mu_k, \Sigma_k)\tau_k}{\sum_{k=1}^{K} p_k(x_i; \mu_k, \Sigma_k)\tau_k}} \tag{43}$$

3. take M-step, update $\boldsymbol{\tau}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$:

$$\boldsymbol{\tau}^* = [\tau_k^*]_{k=1:K} = \frac{1}{N}\sum_{i=1}^{N} T_{k,i} \; \forall k \tag{44}$$

$$\boldsymbol{\mu}^* = [\mu_k^*]_{k=1:K} = \frac{\sum_{i=1}^{N} T_{k,i} x_i}{\sum_{i=1}^{N} T_{k,i}} \; \forall k \tag{45}$$

$$\boldsymbol{\Sigma}^* = [\Sigma_k]_{k=1:K} = \frac{\sum_{i=1}^{N} T_{k,i}(x_i - \mu_k)(x_i - \mu_k)^\top}{\sum_{i=1}^{N} T_{k,i}} \tag{46}$$

4. repeat from step 2 and 3 until converge.