EM Algorithm applied to Gaussian Mixture Models (GMM)

Brendan Law

4 January, 2019

A document showing my formal(-ish) derivation for the EM algorithm as applied to GMMs. Was a good exercise to understand the EM algorithm better before I moved onto deriving the same form for wavelets.

1 Preamble

- Observations (continuous data): $x_i, i \in \{1, \ldots, n\}$
- Latent (hidden) states : $z_k, k \in \{1, \ldots, m\}$
- Conditional density of data given the state - iid observations, normally distributed: $f(x_i \mid z_i = k, \boldsymbol{\theta}) \sim N(\mu_k, \sigma_k^2)$
- $\boldsymbol{\theta}$ is a vector of parameters. It includes:
  - μ_k, σ_k for each k
  - $P(z_i = k) \equiv P(z_i = k \mid \boldsymbol{\theta}) \equiv \pi_k$ is the same across all i for each k
  - Note that $\sum_{k=1}^{m} \pi_k = 1$
- Vector of observations, $\mathbf{X} = (x_1, \ldots, x_n)$, and corresponding states, $\mathbf{Z} = (z_1, \ldots, z_n)$
- The EM algorithm is such that there is an initial estimate of $\boldsymbol{\theta}$, which is then updated at each iteration until some sort of convergence is achieved.

2 Complete log likelihood derivation

$$
\begin{aligned}
P(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) &= P(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta}) P(\mathbf{Z} \mid \boldsymbol{\theta}) && \text{(def'n of conditional probability)} \\
&= P(x_1, \ldots, x_n \mid z_1, \ldots, z_n, \boldsymbol{\theta}) P(z_1, \ldots, z_n \mid \boldsymbol{\theta}) && \text{(expanding the vectors)} \\
&= P(x_i, \ldots, x_n \mid z_1, \ldots, z_n, \boldsymbol{\theta}) \prod_{i=1}^{n} P(z_i \mid \boldsymbol{\theta}) && \text{(independence of } z_i\text{'s)} \\
&= \prod_{i=1}^{n} P(x_i \mid z_i, \boldsymbol{\theta}) P(z_i \mid \boldsymbol{\theta}) && \text{(independence of } x_i\text{'s conditional on } z_i\text{)} \\
&= \prod_{i=1}^{n} \prod_{k=1}^{m} [P(x_i \mid z_i = k, \boldsymbol{\theta}) \pi_k]^{\mathbb{1}\{z_i = k\}} && \text{(law of total probability; specific states for each obs)} \\
&= \prod_{i=1}^{n} \prod_{k=1}^{m} [P(x_i \mid z_i = k, \boldsymbol{\theta}) \pi_k]^{\mathbb{1}\{z_i = k\}} && \text{(prior probability same across observations)} \\
\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) &= \log(P(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})) \\
&= \log(\prod_{i=1}^{n} \prod_{k=1}^{m} [P(x_i \mid z_i = k, \boldsymbol{\theta}) \pi_k]^{\mathbb{1}\{z_i = k\}}) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{m} (\mathbb{1}\{z_i = k\}) \log\{P(x_i \mid z_i = k, \boldsymbol{\theta}) \pi_k\} \\
&= \sum_{i=1}^{n} \sum_{k=1}^{m} (\mathbb{1}\{z_i = k\}) [\log P(x_i \mid z_i = k, \boldsymbol{\theta}) + \log \pi_k] && \text{(log laws)}
\end{aligned}
$$

Note that:

$$
\mathbb{1}\{z_i = k\} = \begin{cases} 1 & z_i = k \\ 0 & z_i \neq k \end{cases}
$$

This remains a random variable, as we don't know what value the latent state variable, z_i, will take, but that it may take on any of $k \in \{1, \ldots, m\}$ with some probability. As we will see, after setting an initial guess of $\boldsymbol{\theta}$, both of the log terms are known (parameters given in θ), and are no longer random variables.

3 EM algorithm

We will now compute the MLE of the parameters in $\boldsymbol{\theta}$ by iterating through the following two steps, and updating $\boldsymbol{\theta}$ at the end of each step.

$$
\begin{aligned}
Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})}) &= \mathbb{E}_{(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})})}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})] && \text{(finding unknowns in this statement is the E-step)} \\
\boldsymbol{\theta}^{(\mathrm{t}+1)} &:= \operatorname*{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})}) && \text{(this is the M-step)}
\end{aligned}
$$

4 E-step: derivation

$$
\begin{aligned}
Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})}) &= \mathbb{E}_{(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})})}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})] \\
&= \mathbb{E}_{(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})})}[\sum_{i=1}^{n} \sum_{k=1}^{m} (\mathbb{1}\{z_i = k\}) [\log P(x_i \mid z_i = k, \boldsymbol{\theta}) + \log \pi_k]] \\
&= \sum_{i=1}^{n} \sum_{k=1}^{m} P(z_i = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})}) [\log P(x_i \mid z_i = k, \boldsymbol{\theta}) + \log \pi_k] \\
& && \text{(linearity of expectations, expectation of indicator RVs)}
\end{aligned}
$$

We hit the expected hurdle in our expectation step - that we need to evaluate $P(z_i = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})})$ for each observation i and possible state k. We can calculate these values (sometimes referred to as 'responsibilities' in the literature, in Bishop (2006), for example) as such:

$$
\begin{aligned}
P(z_i = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})}) &= P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})}) && \text{(by IID of observations)} \\
&= \frac{P(x_i \mid z_i = k, \boldsymbol{\theta}^{(\mathrm{t})})\pi_k}{\displaystyle\sum_{k'=1}^{m} P(x_i \mid z_i = k', \boldsymbol{\theta}^{(\mathrm{t})})\pi_{k'}} && \text{(Bayes' rule)}
\end{aligned}
$$

where $\pi_k = P(z_i = k \mid \boldsymbol{\theta}^{(\mathrm{t})})$ (conditioned on the old θ estimate), in this case.

We can evaluate this expression as we have all the required information stored in our parameter set, $\boldsymbol{\theta}^{(\mathrm{t})}$. Also note that in deriving the expression for $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})})$, we are using the expected log-likelihood as a proxy for the actual log-likelihood, as the actual hidden states are unknown (we don't know if $z_i = k$, and therefore, whether $\mathbb{1}\{z_i = k\}$ is 1 or 0). Thus, this algorithm will aim to find a $\boldsymbol{\theta}$ which maximises the expected log-likelihood, over the probability of each data point taking on a particular state. It can be shown that when this procedure converges, the resulting $\boldsymbol{\theta}$ will correspond to a local maxima of the actual log-likelihood function. Consequently, the resultant $\boldsymbol{\theta}$ is likely to vary depending on how it is initialised - different starting points may converge to different $\boldsymbol{\theta}$ and different log likelihood values.

5 M-step: maximisation

After calculating the responsibilities, we are free of RVs - we have an expression which we can numerically optimise with respect to $\boldsymbol{\theta}$. For the case of the GMM, we can split $\boldsymbol{\theta}$ into two 'groups' of parameters to optimise for:

1. Normal distribution parameters; μ_k, σ_k for $k \in \{1, \ldots, m\}$
2. Probabilities of latent states; π_k, across all i for each $k \in \{1, \ldots, m\}$

To optimise the parameters in 1, we note that the only part of the $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})})$ expression which is affected by these parameters is the part which contains $\log P(x_i \mid z_i = k, \boldsymbol{\theta})$. Furthermore, we can maximise for each state, $k \in \{1, \ldots, m\}$ as the expression is additive in terms of the states. Hence, finding the $\boldsymbol{\theta}$ which maximises

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})})$$

is equivalent to finding the $\boldsymbol{\theta}$ which maximises

$$\sum_{i=1}^{n} P(z_i = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})}) \log P(x_i \mid z_i = k, \boldsymbol{\theta}) \tag{1}$$

for each $k \in \{1, \ldots, m\}$. We note the similarity between this expression and the log-likelihood expression of a normal distribution

$$\sum_{i=1}^{n} \log P(x_i | \boldsymbol{\theta})$$

where $P(x_i | \boldsymbol{\theta}) = P(x_i | \mu, \sigma) \sim N(\mu, \sigma)$, and can see that the expression in 1 is the log-likelihood of a weighted normal distribution. Therefore, the parameters which maximise the log-likelihood of a weighted normal distribution will also maximise our expression, for each k. Hence, we have that:

$$\mu_k^{(t+1)} := \frac{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})\, x_i}{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})} \tag{2}$$

$$\sigma_k^{2(t+1)} := \frac{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})(x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})} \tag{3}$$

To optimise the parameters in 2, only the remaining log term in $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})})$ is affected by altering these parameters. Furthermore, as we are finding the maximising values of $P(z = k \mid \boldsymbol{\theta})$ subject to the constraint that $\sum_{k=1}^{m} P(z = k \mid \boldsymbol{\theta}) = 1$, this becomes a constrained optimisation problem, where we can use a method such as a Lagrangian in order to evaluate. Define

$$\pi_k := P(z = k \mid \boldsymbol{\theta})$$

Then,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\pi}, \lambda) &= \left[\sum_{i=1}^{n}\sum_{k=1}^{m} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})}) \log P(z = k \mid \boldsymbol{\theta})\right] + \lambda(1 - \sum_{k=1}^{m} P(z = k | \boldsymbol{\theta})) \\
&= \left[\sum_{i=1}^{n}\sum_{k=1}^{m} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})}) \log \pi_k\right] + \lambda(1 - \sum_{k=1}^{m} \pi_k)
\end{aligned}$$

For each $k \in \{1, \ldots, m\}$,

$$\begin{aligned}
\frac{\delta \mathcal{L}}{\delta \pi_k} &= \left[\sum_{i=1}^{n} \frac{P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})}{\pi_k}\right] - \lambda = 0 \qquad \text{(All } \pi_j,\ j \neq k \text{ disappear when differentiated with respect to } \pi_k\text{)} \\
\Rightarrow \pi_k &= \frac{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})}{\lambda}, \text{ for each } k \in \{1, \ldots, m\} \\
\frac{\delta \mathcal{L}}{\delta \lambda} &= 1 - \sum_{k=1}^{m} \pi_k = 0
\end{aligned}$$

Therefore, for a given λ:

$$\begin{aligned}
\pi_k &= \frac{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})}{\lambda} \\
\sum_{k=1}^{m} \pi_k &= \sum_{k=1}^{m} \left(\frac{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})}{\lambda}\right) = 1 \\
\Rightarrow \lambda &= \sum_{k=1}^{m}\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})}) \\
\Rightarrow \pi_k &= \frac{\sum_{i=1}^{n} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})}{\sum_{i=1}^{n}\sum_{k=1}^{m} P(z_i = k \mid x_i, \boldsymbol{\theta}^{(\mathrm{t})})}
\end{aligned}$$

References

C. M. Bishop. *Pattern recognition and machine learning.* Information science and statistics. New York : Springer, c2006., 2006. ISBN 9780387310732.

S. Borman. The expectation maximization algorithm-a short tutorial. *Submitted for publication*, pages 1–9, 2004.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

R. Grosse and N. Srivastava. Lecture 16: Mixture models. 2015.