

# Handout on the Finite Mixture Models and Expectation Maximization Algorithm

Lecturer &amp; author: Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

## Aim

Motivation, and implementation of Finite Mixture Models and the standard Expectation Maximization Algorithm

## Reading list:

- McLachlan, G.J. & Peel, D. (2000). Finite Mixture Models. Wiley
- McLachlan, G. J., & Krishnan, T. (2007). The EM algorithm and extensions (Vol. 382). John Wiley & Sons.
- Benaglia, T., Chauveau, D., Hunter, D., & Young, D. (2009). mixtools: An R package for analyzing finite mixture models. Journal of Statistical Software, 32(6), 1-29.
- Wu, C. J. (1983). On the convergence properties of the EM algorithm. The Annals of statistics, 95-103.

## 1 Finite Mixture Models

**Definition 1.** Let  $\{f_j(y|\theta_j); j = 1, \dots, K\}$  be a collection of probability distributions where  $\{\theta_j\}_{j=1}^K$  are parameters of the  $j$ -th component  $f_j(y|\theta_j)$ . Let  $\{\varpi_j\}_{j=1}^K$  be a set of weights where  $\varpi_j > 0$  and  $\sum_{j=1}^K \varpi_j = 1$ . The mixture distribution derived from the aforementioned collections is

$$f(y|\varpi, \theta) = \sum_{j=1}^K \varpi_j f_j(y|\theta_j), \quad y \in \mathcal{Y} \quad (1)$$

where  $\theta := (\theta_j, j = 1, \dots, K)$  and  $\varpi := (\varpi_j, j = 1, \dots, K)$ .  $f_j(y|\theta_j)$  is called  $j$ -th mixture component with mixture weight  $\varpi_j$ .

**Definition 2.** A finite mixture model is called parametric mixture model if its components are members of the same parametric family of distributions eg.  $\{f(y|\theta_j); j = 1, \dots, K\}$ , and hence

$$f(y|\varpi, \theta) = \sum_{j=1}^K \varpi_j f(y|\theta_j), \quad y \in \mathcal{Y}$$

**Example 3.** A (multivariate) Normal Mixture model has density

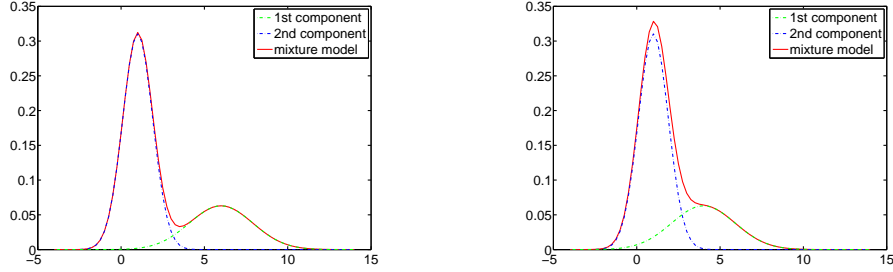
$$f(y|\varpi, \mu, \Sigma) = \sum_{j=1}^K \varpi_j \mathcal{N}(y|\mu_j, \Sigma_j)$$

**Note 4.** Finite mixture models provide semi-parametric modeling approach to model distribution density/mass functions with unknown shapes. This is accomplished because they can be considered as weighted averages of densities; see (1). In other words, observations  $y_i$ , are realizations such that

$$y_i \stackrel{\text{iid}}{\sim} f(y|\varpi, \theta) = \sum_{j=1}^K \varpi_j f_j(y|\theta_j)$$

Eg, the density of a multimodal, or a asymmetric distribution can be approximated by a suitable mixture model of symmetric and unimodal distributions.

**Example 5.** A Finite Mixture of (unimodal and symmetric) Normal distributions with different parameter values can describe a population with two groups Normally distributed with different parameters.



(a) Bimodal PDF:  
 $f(y) = 0.7N(y|1, 0.9^2) + 0.3N(y|6, 1.9^2)$

(b) Right skewed PDF:  
 $f(y) = 0.7N(y|1, 0.9^2) + 0.3N(y|4, 1.9^2)$

Figure 1: Normal mixture models.  $N(x|\mu, \sigma^2)$  denotes the Normal distribution density at value  $x$  with mean  $\mu$  and variance  $\sigma^2$ .

**Note 6.** Finite mixture models provide a natural framework to model heterogeneity; eg, cluster analysis.

- Consider a sample of observables  $\{y_1, \dots, y_n\}$  of size  $n$  drawn from a heterogeneous population with groups  $\{G_1, \dots, G_K\}$  with sizes proportional to  $\{\varpi_1, \dots, \varpi_K\}$ .
- Let  $z_i \in \{1, \dots, K\}$  be latent allocation variable which acts as a label, such that the event  $\{z_i = j\}$  means that observation  $y_i$  belongs to group  $j$ . Consider that  $\{z_1, \dots, z_n\}$  are not observed or missing part of the data. Assume that  $z_i$  are independent random variables with distribution such that

$$\{z_i = j\} \sim \text{pr}(\{z_i = j\}) = \varpi_j$$

for all  $j = 1, \dots, K$ .

- If  $\{z_i\}$  were known, and given that  $z_i = j$ , it is

$$y_i|z_i \stackrel{\text{ind}}{\sim} f_{z_i}(y|\theta_{z_i})$$

So the joint distribution for the complete data  $\{y_i, z_i\}$  is such that

$$y_i|z_i \stackrel{\text{ind}}{\sim} f_{z_i}(y|\theta_{z_i})$$

$$z_i \stackrel{\text{ind}}{\sim} \text{pr}(z_i) = \varpi_{z_i}$$

So

$$f(y_i, z_i) = \varpi_{z_i} f_{z_i}(y_i|\theta_{z_i})$$

and

$$f(y_i|\varpi, \theta) = \sum_{z_i=1}^K f(y_i, z_i) = \sum_{z_i=1}^K \varpi_{z_i} f_{z_i}(y_i|\theta_{z_i}) = \sum_{j=1}^K \varpi_j f_j(y|\theta_j)$$

**Example 7.** The Normal Mixture Model which has marginal sampling distribution

$$y_i \stackrel{\text{iid}}{\sim} f(y|\varpi, \mu, \Sigma) = \sum_{j=1}^K \varpi_j N(y_i|\mu_j, \Sigma_j)$$

can be written in a hierarchical form

$$y_i | z_i \stackrel{\text{ind}}{\sim} \mathcal{N}(y_i | \mu_{z_i}, \Sigma_{z_i})$$

$$z_i \stackrel{\text{ind}}{\sim} \text{pr}(z_i) = \varpi_{z_i}$$

by considering the allocation latent variables  $z_i$  indicating the group to which observation  $y_i$  belongs.

## Inferential tools

*Note 8.* Given we learn MLEs  $\hat{\varpi}$  and  $\hat{\theta}$  of  $\varpi$  and  $\theta$ , density estimation (alternatively to the histogram) can be performed as

$$f(y' | \varpi, \mu, \Sigma) = \sum_{j=1}^K \hat{\varpi}_j f_j(y | \hat{\theta}_j)$$

*Note 9.* Given we learn MLEs  $\hat{\varpi}$  and  $\hat{\theta}$  of  $\varpi$  and  $\theta$ , clustering of a new observable  $y'$  in one of the groups  $\{G_1, \dots, G_K\}$  can be performed as follows:

1. By using Bayes theorem, compute

$$\text{pr}(\{y' \in G_j\} | y) = \frac{\varpi_j f_j(y | \theta_j)}{\sum_{j=1}^K \varpi_j f_j(y | \theta_j)} \Big|_{(\varpi, \theta) = (\hat{\varpi}, \hat{\theta})}$$

for all  $j = 1, \dots, K$ , as  $\text{pr}(y' \in G_j) = \varpi_j$ .

2. Find

$$j^* = \arg \max_{\forall j} \{\text{pr}(\{y' \in G_j\} | y)\} = \arg \max_{\forall j} \left\{ \frac{\varpi_j f_j(y | \theta_j)}{\sum_{j=1}^K \varpi_j f_j(y | \theta_j)} \Big|_{(\varpi, \theta) = (\hat{\varpi}, \hat{\theta})} \right\}$$

where we plug  $\hat{\varpi}$  and  $\hat{\theta}$  in  $\varpi$  and  $\theta$  in the formulas above.

## But how to train a finite mixture model?

*Note 10.* Assume

$$y_i \stackrel{\text{iid}}{\sim} f(y | \varpi, \mu, \Sigma) = \sum_{j=1}^K \varpi_j \mathcal{N}(y_i | \mu_j, \Sigma_j)$$

to find MLEs  $\hat{\varpi}$ ,  $\hat{\mu}$  and  $\hat{\Sigma}$  of  $\varpi$ ,  $\mu$  and  $\Sigma$

$$(\hat{\varpi}, \hat{\mu}, \hat{\Sigma}) = \arg \max_{\varpi, \mu, \Sigma} \left( \sum_{i=1}^n \log \sum_{j=1}^K \varpi_j \mathcal{N}(y_i | \mu_j, \Sigma_j) \right)$$

*Note 11.* However, I cannot get an explicit solution by directly differentiating wrt  $(\varpi, \mu, \Sigma)$  and setting equal to zero. More over the resulting likelihood equations are too complicated to be solved, eg.:

$$0 = \sum_{i=1}^n \frac{\mathcal{N}(y_i | \mu_j, \Sigma_j) - \mathcal{N}(y_i | \mu_K, \Sigma_K)}{\sum_{j=1}^K \varpi_j \mathcal{N}(y_i | \mu_j, \Sigma_j)} \Big|_{(\varpi, \mu, \Sigma) = (\hat{\varpi}, \hat{\mu}, \hat{\Sigma})}$$

etc...

*Note 12.* Alternatively, I could possibly resort to other computational procedures that can produce these MLEs in a more computationally convenient manner (see, the Expectation Maximization algorithm).

## 2 Expectation Maximization (EM) algorithm

### Motivation

- Assume observables  $y_1, \dots, y_n$  such that

$$y_i \stackrel{\text{iid}}{\sim} f(\cdot|\theta)$$

where  $y = (y_1, \dots, y_n)$ , and  $\theta \in \Theta \subseteq \mathbb{R}^d$  is unknown. I wish to learn  $\theta$ .

- Assuming I wish to find the MLE  $\hat{\theta}$  of  $\theta$ , such that

$$\hat{\theta} = \arg \max_{\forall \theta} (L(y|\theta)) = \arg \max_{\forall \theta} (\log L(y|\theta))$$

however it is difficult (or undesirable) to directly perform the underline optimization.—How do I find  $\hat{\theta}$ ?

### An idea

*Note 13.* It can be done by extending the space  $\Theta$  of the unknown quantities  $\theta$ . This technique is known as demarginalization, data imputation, data augmentation, etc...

*Note 14.* Impute (or augment) the observed data  $y = (y_1, \dots, y_n)$  with unknown quantities  $z = (z_1, \dots, z_m)$  which can be considered as missing data of a complete data set  $(y, z)$ , or as latent parameters by extending the parameter array into  $(\theta, z)$ . Assume that  $z|y, \theta \sim q(z|y, \theta)$ , where

$$g(y, z|\theta) = q(z|y, \theta) f(y|\theta)$$

or equivalently that  $y, z|\theta \sim g(y, z|\theta)$  where

$$q(z|y, \theta) = \frac{g(y, z|\theta)}{f(y|\theta)}$$

where  $q(z|y, \theta)$  and  $g(y, z|\theta)$  are specified by the researcher. Note that  $\int g(y, z|\theta) dz = f(y|\theta)$ .

*Note 15.* The specification of  $z$  and  $q(\cdot|\cdot, \cdot)$  or  $g(\cdot, \cdot|\cdot)$  is made so that they lead to convenient computations (we will see later). Among the possible options where  $z$ ;  $q(\cdot|\cdot, \cdot)$ ;  $g(\cdot, \cdot|\cdot)$  lead to convenient computations, it is preferable to use  $z$ ;  $q(\cdot|\cdot, \cdot)$ ;  $g(\cdot, \cdot|\cdot)$  which has a pretty interesting interpretation (if any).

**Example 16.** For the Finite mixture model with  $\theta = (\varpi, \mu, \Sigma)$  and

$$f(y_i|\varpi, \mu, \Sigma) = \sum_{j=1}^K \varpi_j \mathbf{N}(y_i|\mu_j, \Sigma_j), \quad y \in \mathbb{R}$$

we can choose  $z = (z_1, \dots, z_n) \in \{1, \dots, K\}^n$  where  $z_i$  is the label indicating to which group the observable  $y_i$  belongs

$$y_i|z_i \stackrel{\text{ind}}{\sim} \mathbf{N}(y_i|\mu_{z_i}, \Sigma_{z_i})$$

$$z_i \stackrel{\text{ind}}{\sim} pr(z_i) = \varpi_{z_i}$$

Hence, for  $i = 1, \dots, n$ , it is

$$g(y_i, z_i|\varpi, \mu, \Sigma) = \varpi_{z_i} \mathbf{N}(y_i|\mu_{z_i}, \Sigma_{z_i})$$

$$f(y_i|\varpi, \mu, \Sigma) = \sum_{j=1}^K \varpi_j \mathbf{N}(y_i|\mu_j, \Sigma_j)$$

$$q(z_i|y_i, \varpi, \mu, \Sigma) = \frac{\varpi_{z_i} \mathbf{N}(y_i|\mu_{z_i}, \Sigma_{z_i})}{\sum_{j=1}^K \varpi_j \mathbf{N}(y_i|\mu_j, \Sigma_j)}$$

100 Here the latent variables  $\{z_i\}$  can be nicely interpreted as missing data, and their interpretation is clear.

## 101 The Algorithm

102 *Note 17.* Observe that for any  $\theta^*$

$$103 \log(L(\theta|y)) = E_{z \sim q(z|y, \theta^*)}(\log(L(\theta|y, z))) - E_{z \sim q(z|y, \theta^*)}(\log(q(z|y, \theta))) \quad (2)$$

104 and define

$$105 Q(\theta|\theta^*, y) = E_{z \sim q(z|y, \theta^*)}(\log(L(\theta|y, z))) \quad (3)$$

106 *Note 18.* EM is an iterative procedure which generates a sequence of  $\{\tilde{\theta}^{(t)}\}_{t \geq 1}$  by recursively maximizing (3) as

$$107 \tilde{\theta}^{(t)} = \arg \max_{\theta \in \Theta} \left( Q(\theta|\tilde{\theta}^{(t-1)}, y) \right) = \arg \max_{\theta \in \Theta} \left( E_{z \sim q(z|y, \tilde{\theta}^{(t-1)})}(\log(L(\theta|y, z))) \right)$$

108 see the pseudo-algorithm below.

---

### Algorithm 1 Expectation Maximization algorithm.

---

Set a seed  $\theta^{(0)}$ .

Iterate, for  $t = 1, 2, 3, \dots$ :

1. E-step (Expectation step) Compute:

$$Q(\theta|\tilde{\theta}^{(t-1)}, y) = E_{z \sim q(z|y, \tilde{\theta}^{(t-1)})}(\log(L(\theta|y, z)))$$

2. M-step (Maximization step) Find:

$$\tilde{\theta}^{(t)} = \arg \max_{\theta \in \Theta} \left( Q(\theta|\tilde{\theta}^{(t-1)}, y) \right)$$

Terminate when a termination criterion is satisfied, eg.,

$$\left\| \tilde{\theta}^{(t)} - \tilde{\theta}^{(t-1)} \right\| < \epsilon \left\| \tilde{\theta}^{(t-1)} \right\|$$

for some small  $\epsilon > 0$ .

---

109 *Note 19.* The maximization required in M-step of Algorithm 1 can be performed either analytically, or numerically.  
 110 Numerical optimization can be performed for instance with the use of Newton method or Broyden method. Introduc-  
 111 tory notes about Newton method can found in Appendix A in the hand out of Topics in Statistics III/IV <sup>1</sup>.

## 112 Properties / comments

113 *Note 20.* The following Theorem says that EM tends to increase the likelihood at each iteration, but it does not say  
 114 anything about convergence.

115 **Theorem 21.** *The sequence  $\{\tilde{\theta}^{(t)}\}$  generated from EM satisfies*

$$116 L(y|\tilde{\theta}^{(t+1)}) \geq L(y|\tilde{\theta}^{(t)})$$

117 *with equality satisfied if and only if*

$$118 Q(\tilde{\theta}^{(t+1)}|\tilde{\theta}^{(t)}, y) = Q(\tilde{\theta}^{(t)}|\tilde{\theta}^{(t)}, y)$$

---

<sup>1</sup>[https://github.com/georgios-stats/Topics\\_in\\_Statistics\\_Michaelmas\\_2020/blob/master/Contingency\\_Tables/Handouts\\_LogLinearModel.pdf](https://github.com/georgios-stats/Topics_in_Statistics_Michaelmas_2020/blob/master/Contingency_Tables/Handouts_LogLinearModel.pdf)

119 *Proof.* From 2 it is

$$120 \log(L(\theta|y)) = Q(\theta|\theta^*, y) - \mathbb{E}_{z \sim q(z|y, \theta^*)} (\log(q(z|y, \theta)))$$

121 It is

$$122 Q(\tilde{\theta}^{(t+1)}|\tilde{\theta}^{(t)}, y) \geq Q(\tilde{\theta}^{(t)}|\tilde{\theta}^{(t)}, y) \quad (4)$$

123 because

$$124 \tilde{\theta}^{(t+1)} = \arg \max_{\theta \in \Theta} (Q(\theta|\tilde{\theta}^{(t)}, y))$$

125 It is

$$126 \mathbb{E}_{z \sim q(z|y, \tilde{\theta}^{(t)})} (\log(q(z|y, \tilde{\theta}^{(t+1)}))) \leq \mathbb{E}_{z \sim q(z|y, \tilde{\theta}^{(t)})} (\log(q(z|y, \tilde{\theta}^{(t)})))$$

127 because

$$128 \mathbb{E}_{z \sim q(z|y, \tilde{\theta}^{(t)})} (\log(q(z|y, \tilde{\theta}^{(t+1)}))) \leq \mathbb{E}_{z \sim q(z|y, \tilde{\theta}^{(t)})} (\log(q(z|y, \tilde{\theta}^{(t)})))$$

$$129 \iff \mathbb{E}_{z \sim q(z|y, \tilde{\theta}^{(t)})} \left( \log \left( \frac{q(z|y, \tilde{\theta}^{(t+1)})}{q(z|y, \tilde{\theta}^{(t)})} \right) \right) \leq 0$$

130 which is true because by Jensen's inequality

$$131 \mathbb{E}_{z \sim q(z|y, \tilde{\theta}^{(t)})} \left( \log \left( \frac{q(z|y, \tilde{\theta}^{(t+1)})}{q(z|y, \tilde{\theta}^{(t)})} \right) \right) \leq \log \left( \mathbb{E}_{z \sim q(z|y, \tilde{\theta}^{(t)})} \left( \frac{q(z|y, \tilde{\theta}^{(t+1)})}{q(z|y, \tilde{\theta}^{(t)})} \right) \right) = \log \left( \int q(z|y, \tilde{\theta}^{(t+1)}) dz \right) = \log(1) = 0$$

132 □

133 *Note 22.* Regularity conditions that guaranty that EM converges to the MLE or the 'real parameter' value are presented

134 in (Wu, C. J., 1983). For instance see, the following Theorem.

135 **Theorem 23.** Let  $\{\tilde{\theta}^{(t)}\}$  be a sequence  $\{\tilde{\theta}^{(t)}\}$  generated from EM. If  $Q(\theta|\theta^*, y)$  is continuous on  $\theta$  and  $\theta^*$  then

- 136 1.  $\{\tilde{\theta}^{(t)}\}$  converges to  $\hat{\theta}$
- 137 2.  $L(\tilde{\theta}^{(t)}|y)$  converges to  $L(\hat{\theta}|y)$

138 where  $\hat{\theta}$  be a stationary point of the likelihood  $L(\theta|y)$  –not the MLE necessarily.

139 *Note 24.* EM is prone to local trapping (eg,  $\tilde{\theta}^{(t)}$  may converge to a local optimum). Yet the limit point of  $\tilde{\theta}^{(t)}$  often

140 depends on the initial seed  $\theta^{(0)}$ . An easy remedy to mitigate the this issue is to run EM multiple times by initiating it

141 from different seeds, and at the end get the  $\tilde{\theta}^{(\infty)}$  with the largest likelihood  $L(\tilde{\theta}^{(\infty)}|y)$ .

## 142 Variations

143 *Note 25.* Integration in the E-step in EM (Algorithm 1) may not be tractable. Instead, we can approximate the integral

144 by Monte Carlo integration, as

145 E'-step (Expectation step)

146 1. Simulate:

$$147 z^{(\xi)} \sim q(\cdot|y, \tilde{\theta}^{(t-1)}), \quad \xi = 1, \dots, N$$

148 2. Compute:

$$149 Q(\theta|\tilde{\theta}^{(t-1)}, y) \approx \frac{1}{N} \sum_{\xi=1}^N (\log(L(\theta|y, z^{(\xi)})))$$

150 the above This is a reasonable approximation based on LLN arguments. See below

---

**Algorithm 2** Monte Carlo Expectation Maximization (MCEM) algorithm

---

Set a seed  $\theta^{(0)}$ .

Iterate, for  $t = 1, 2, 3, \dots$ :

1. E'-step (Expectation step)

(a) Simulate:

$$z^{(\xi)} \sim q(\cdot | y, \tilde{\theta}^{(t-1)}), \quad \xi = 1, \dots, N$$

(b) Compute:

$$Q(\theta | \tilde{\theta}^{(t-1)}, y) \approx \frac{1}{N} \sum_{\xi=1}^N \left( \log \left( L(\theta | y, z^{(\xi)}) \right) \right)$$

2. M-step (Maximization step) Find:

$$\tilde{\theta}^{(t)} = \arg \max_{\theta \in \Theta} \left( Q(\tilde{\theta}^{(t)} | \tilde{\theta}^{(t-1)}, y) \right)$$

Terminate when a termination criterion is satisfied, eg.,

$$\left\| \tilde{\theta}^{(t)} - \tilde{\theta}^{(t-1)} \right\| < \epsilon \left\| \tilde{\theta}^{(t-1)} \right\|$$

for some small  $\epsilon > 0$ .

---

151 **Implementation: EM for Finite Normal Mixture models**

152 **Example 26.** (Continue) Assume observables  $\{y_i\}_{i=1}^n$  such that  $y_i \sim f(y_i | \varpi, \mu, \Sigma)$  with

153 
$$f(y_i | \varpi, \mu, \Sigma) = \sum_{j=1}^K \varpi_j \mathbf{N}(y_i | \mu_j, \Sigma_j)$$

154 where  $\varpi, \mu, \Sigma$  are unknown. Find MLE  $(\hat{\varpi}, \hat{\mu}, \hat{\Sigma})$  for  $(\varpi, \mu, \Sigma)$ .

155 **Solution.** It is  $\theta = (\varpi, \mu, \Sigma)$ . Also

156 
$$L(\theta | y) = \prod_{i=1}^n \sum_{j=1}^k \varpi_j \mathbf{N}(y_i | \mu_j, \Sigma_j)$$

157 
$$L(\theta | y, z) = \prod_{i=1}^n \prod_{j=1}^k [\mathbf{N}(y_i | \mu_{z_i}, \Sigma_{z_i}) \varpi_{z_i}]^{1(z_i=j)}$$

158 
$$\log(L(\theta | y, z)) = \sum_{i=1}^n \sum_{j=1}^k 1(z_i = j) \log(\varpi_j \mathbf{N}(y_i | \mu_j, \Sigma_j))$$

159 
$$= \sum_{i=1}^n \sum_{j=1}^k 1(z_i = j) \left( \log(\varpi_j^{(t)}) - \frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\det(\Sigma_j)) - \frac{1}{2} (y_i - \mu_j)^\top \Sigma_j^{-1} (y_i - \mu_j) \right)$$

160 E-step:

$$\begin{aligned}
161 \quad Q\left(\theta|\tilde{\theta}^{(t)}, y\right) &= \mathbb{E}_{z \sim q(z|y, \theta^{(t)})} (\log(L(\theta|y, z))) \\
162 \quad &= \sum_{i=1}^n \sum_{j=1}^k \mathbb{E}_{z \sim q(z|y, \theta^{(t)})} (1(z_i = j)) \log\left(\varpi_j^{(t)} \mathbf{N}(y_i|\mu_j^{(t)}, \Sigma_j^{(t)})\right) \\
163 \quad &= \sum_{i=1}^n \sum_{j=1}^k \mathbb{E}_{z \sim q(z|y, \theta^{(t)})} (1(z_i = j)) \left( -\frac{1}{2} \log(2\pi) - \frac{1}{2} \log\left(\det\left(\Sigma_j^{(t)}\right)\right) - \frac{1}{2} \left(y_i - \mu_j^{(t)}\right)^\top \left(\Sigma_j^{(t)}\right)^{-1} \left(y_i - \mu_j^{(t)}\right) \right)
\end{aligned}$$

164 where

$$\begin{aligned}
165 \quad \mathbb{E}_{z \sim q(z|y, \theta^{(t)})} (1(z_i = j)) &= \text{pr}\left(z_i = j|y, \theta^{(t)}\right) = q\left(z_i = j|y_i, \varpi^{(t)}, \mu^{(t)}, \Sigma^{(t)}\right) \\
166 \quad &= \frac{\varpi_j^{(t)} \mathbf{N}(y_i|\mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_{j=1}^K \varpi_j^{(t)} \mathbf{N}(y_i|\mu_j^{(t)}, \Sigma_j^{(t)})}
\end{aligned}$$

167 Let denote

$$168 \quad \Pi_{i,j}^{(t)} = \frac{\varpi_j^{(t)} \mathbf{N}(y_i|\mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_{j=1}^K \varpi_j^{(t)} \mathbf{N}(y_i|\mu_j^{(t)}, \Sigma_j^{(t)})}$$

169 So

$$\begin{aligned}
170 \quad Q\left(\theta|\tilde{\theta}^{(t-1)}, y\right) &= \sum_{i=1}^n \sum_{j=1}^k \Pi_{i,j}^{(t)} \log\left(\varpi_j^{(t)} \mathbf{N}(y_i|\mu_j^{(t)}, \Sigma_j^{(t)})\right) \\
171 \quad &= \sum_{i=1}^n \sum_{j=1}^k \Pi_{i,j}^{(t)} \left( \log\left(\varpi_j^{(t)}\right) - \frac{1}{2} \log(2\pi) - \frac{1}{2} \log\left(\det\left(\Sigma_j^{(t)}\right)\right) - \frac{1}{2} \left(y_i - \mu_j^{(t)}\right)^\top \left(\Sigma_j^{(t)}\right)^{-1} \left(y_i - \mu_j^{(t)}\right) \right)
\end{aligned}$$

172 M-step: by differentiating  $Q\left(\theta|\tilde{\theta}^{(t-1)}, y\right)$  and setting equal to zero I get

$$\begin{aligned}
173 \quad \varpi^{(t+1)} &= \arg \max_{\forall \varpi_j} \left( Q\left(\theta|\tilde{\theta}^{(t)}, y\right) \right) \implies \varpi_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^n \Pi_{i,j}^{(t)} \\
174 \quad \left(\mu^{(t+1)}, \Sigma^{(t+1)}\right) &= \arg \max_{\forall \mu, \Sigma} \left( Q\left(\theta|\tilde{\theta}^{(t)}, y\right) \right) \\
175 \quad &\implies \begin{cases} \mu_j^{(t+1)} &= \sum_{i=1}^n \frac{\Pi_{i,j}^{(t)}}{\sum_{i=1}^n \Pi_{i,j}^{(t)}} y_i \\ \Sigma_j^{(t+1)} &= \sum_{i=1}^n \frac{\Pi_{i,j}^{(t)}}{\sum_{i=1}^n \Pi_{i,j}^{(t)}} \left(y_i - \mu_j^{(t+1)}\right)^\top \left(y_i - \mu_j^{(t+1)}\right) \end{cases}
\end{aligned}$$

### 176 3 Examples & code with Finite Normal Mixture Models

177 *Note 27.* An R implementation of the Expectation Maximization for training Finite Mixture Models is in the R package  
178 *mixtools*;

- 179 • <https://cran.r-project.org/web/packages/mixtools/index.html>
- 180 • <https://cran.r-project.org/web/packages/mixtools/vignettes/mixtools.pdf>

181 **Example 28.** Here are some R examples.



```

182 #faithful 1d
rm(list=ls())
#install.packages('mixtools')
library(mixtools)
data(faithful)
hist(faithful$waiting)
obj.mix <- normalmixEM(faithful$waiting,
arbvar = FALSE,k = 2)
plot(obj.mix, which=1)
plot(obj.mix, which=2)
lines(density(obj.mix$x), lty=2, lwd=0.8)

```

```

183 #faithful
rm(list=ls())
#install.packages('mixtools')
library('mixtools')
data("faithful")
X <- faithful[,1:2]
plot(X)
# 2 components
obj.mix.2<-mvnormalmixEM(X,k=2)
obj.mix.2
par(mfrow=c(2,2))
hist(X[,1])
hist(X[,2])
plot.mixEM(obj.mix.2,whichplots=1)
plot.mixEM(obj.mix.2,whichplots=2)
par(mfrow=c(1,1))
plot.mixEM(obj.mix.2,whichplots=2)
# 3 components
obj.mix.3 <- mvnormalmixEM(X,k=3)
obj.mix.3
par(mfrow=c(2,2))
hist(X[,1])
hist(X[,2])
plot.mixEM(obj.mix.3,whichplots=1)
plot.mixEM(obj.mix.3,whichplots=2)
par(mfrow=c(1,1))
plot.mixEM(obj.mix.3,whichplots=2)

```

```

# N0data
rm(list=ls())
#install.packages('mixtools')
library('mixtools')
data(N0data)
plot(N0data)
X <- N0data plot(X)
# 2 components
obj.mix.2 <- mvnormalmixEM(X,k=2)
obj.mix.2
par(mfrow=c(2,2))
hist(X[,1])
hist(X[,2])
plot.mixEM(obj.mix.2,whichplots=1)
plot.mixEM(obj.mix.2,whichplots=2)
par(mfrow=c(1,1))
plot.mixEM(obj.mix.2,whichplots=2)
# 3 components
obj.mix.3 <- mvnormalmixEM(X,k=3)
obj.mix.3
par(mfrow=c(2,2))
hist(X[,1])
hist(X[,2])
plot.mixEM(obj.mix.3,whichplots=1)
plot.mixEM(obj.mix.3,whichplots=2)
par(mfrow=c(1,1))
plot.mixEM(obj.mix.3,whichplots=2)

```

## 184 4 Practice

185 **Question 29.** *Try to write the R code for the Example 26.*