Statistical Methods III (MATH3051) *Epiphany term, 2021*

Handout on the Finite Mixture Models and Expectation Maximization Algorithm

Lecturer & 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 $\{f_j(y|\theta_j);\ j = 1, ..., K\}$ 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

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

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

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

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

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

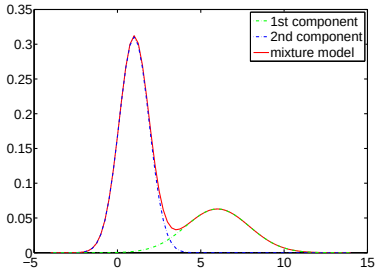
$$f(y|\varpi, \mu, \Sigma) = \sum_{j=1}^{K} \varpi_j \mathcal{N}(y|\mu_j, \Sigma_j)$$

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

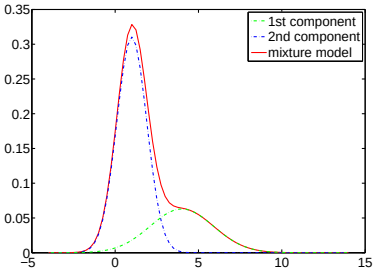
$$y_i \overset{iid}{\sim} f(y|\varpi, \theta) = \sum_{j=1}^{K} \varpi_j f_j(y|\theta_j)$$

Eg, the density of a multimodel, 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:
$f(y) = 0.7\mathcal{N}(y|1, 0.9^2) + 0.3\mathcal{N}(y|6, 1.9^2)$



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

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

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

- *Consider a sample of observables $\{y_1, ..., y_n\}$ of size n drawn from a heterogeneous population with groups $\{G_1, ..., G_K\}$ with sizes proportional to $\{\varpi_1, ..., \varpi_K\}$.*
- *Let $z_i \in \{1, ..., K\}$ 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 $\{z_1, ..., z_n\}$ are not observed or missing part of the data. Assume that z_i are independent random variables with distribution such that*

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

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

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

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

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

$$y_i|z_i \overset{ind}{\sim} f_{z_i}(y_i|\theta_{z_i})$$
$$z_i \overset{ind}{\sim} pr(z_i) = \varpi_{z_i}$$

  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

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

can be written in a hierarchical form

$$y_i|z_i \overset{ind}{\sim} \mathcal{N}(y_i|\mu_{z_i}, \Sigma_{z_i})$$
$$z_i \overset{ind}{\sim} pr(z_i) = \varpi_{z_i}$$

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 ϖ and θ, 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 ϖ and θ, clustering of a new observable y' in one of the groups $\{G_1, ..., G_K\}$ can be performed as follows:

1. *By using Bayes theorem, compute*

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

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

2. *Find*

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

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

But how to train a finite mixture model?

Note 10. Assume

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

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

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

Note 11. However, I cannot get an explicit solution by directly differentiating wrt (ϖ, μ, Σ) and setting equal to zero. More over the resulting likelihood equations are too complicated to by solves, eg.:

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

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* $y_1, ...y_n$ *such that*

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

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

- *Assuming I wish to find the MLE* $\hat{\theta}$ *of* θ*, 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 Θ *of the unknown quantities* θ*. This technique is known as known as demarginalization, data imputation, data augmentation, etc...*

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

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

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

$$y_i|z_i \overset{ind}{\sim} \mathcal{N}(y_i|\mu_{z_i}, \Sigma_{z_i})$$
$$z_i \overset{ind}{\sim} pr(z_i) = \varpi_{z_i}$$

Hence, for $i = 1, ..., n$*, it is*

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

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

The Algorithm

Note 17. Observe that for any θ^*

$$\log\left(L\left(\theta|y\right)\right) = \mathcal{E}_{z\sim q(z|y,\theta^*)}\left(\log\left(L\left(\theta|y,z\right)\right)\right) - \mathcal{E}_{z\sim q(z|y,\theta^*)}\left(\log\left(q\left(z|y,\theta\right)\right)\right) \tag{2}$$

and define

$$Q\left(\theta|\theta^*,y\right) = \mathcal{E}_{z\sim q(z|y,\theta^*)}\left(\log\left(L\left(\theta|y,z\right)\right)\right) \tag{3}$$

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

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

see the pseudo-algorithm below.

Algorithm 1 Expectation Maximization algorithm.

Set a seed $\theta^{(0)}$.
Iterate, for $t = 1, 2, 3, ...$:

1. *E-step (Expectation step) Compute:*

$$Q\left(\theta|\tilde{\theta}^{(t-1)},y\right) = \mathcal{E}_{z\sim q\left(z|y,\tilde{\theta}^{(t-1)}\right)}\left(\log\left(L\left(\theta|y,z\right)\right)\right)$$

2. *M-step (Maximization step) Find:*

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

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 $\epsilon > 0$.

Note 19. The maximization required in M-step of Algorithm 1 can be performed either analytically, or numerically. Numerical optimization can be performed for instance with the use of Newton method or Broyden method. Introductory notes about Newton method can found in Appendix A in the hand out of Topics in Statistics III/IV[1].

Properties / comments

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

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

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

with equality satisfied if and only if

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

Proof. From 2 it is

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

[1] https://github.com/georgios-stats/Topics_in_Statistics_Michaelmas_2020/blob/master/Contigency_Tables/Handouts_LogLinearModel.pdf

It is

$$Q\left(\tilde{\theta}^{(t+1)}|\tilde{\theta}^{(t)},y\right)\geq Q\left(\tilde{\theta}^{(t)}|\tilde{\theta}^{(t)},y\right) \tag{4}$$

because

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

It is

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

because

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

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

which is true because by Jensen's inequality

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

□

Note 22. Regularity conditions that guaranty that EM converges to the MLE or the 'real parameter' value are presented in (Wu, C. J., 1983). For instance see, the following Theorem.

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

1. *$\left\{\tilde{\theta}^{(t)}\right\}$ converges to $\hat{\theta}$*
2. *$L\left(\tilde{\theta}^{(t)}|y\right)$ converges to $L\left(\hat{\theta}|y\right)$*

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

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 depends on the initial seed $\theta^{(0)}$. An easy remedy to mitigate the this issue is to run EM multiple times by initiating it from different seeds, and at the end get the $\tilde{\theta}^{(\infty)}$ with the largest likelihood $L\left(\tilde{\theta}^{(\infty)}|y\right)$.

Variations

Note 25. Integration in the E-step in EM (Algorithm 1) may not be tractable. Instead, we can approximate the integral by Monte Carlo integration, as

E'-step (Expectation step)

1. *Simulate:*

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

2. *Compute:*

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

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 $t = 1, 2, 3, ...$:

1. *E'-step (Expectation step)*

   (a) *Simulate:*

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

   (b) *Compute:*

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

2. *M-step (Maximization step) Find:*

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

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 $\epsilon > 0$.

Implementation: EM for Finite Normal Mixture models

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

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

where ϖ, μ, Σ *are unknown. Find MLE* $(\hat{\varpi}, \hat{\mu}, \hat{\Sigma})$ *for* (ϖ, μ, Σ).

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

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

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

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

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

E-step:

$$Q\left(\theta|\tilde{\theta}^{(t)}, y\right) = \mathcal{E}_{z\sim q\left(z|y,\theta^{(t)}\right)}\left(\log\left(L(\theta|y, z)\right)\right)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{k}\mathcal{E}_{z\sim q\left(z|y,\theta^{(t)}\right)}\left(1(z_i = j)\right)\log\left(\varpi_j^{(t)}\mathcal{N}(y_i|\mu_j^{(t)}, \Sigma_j^{(t)})\right)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{k}\mathcal{E}_{z\sim q\left(z|y,\theta^{(t)}\right)}\left(1(z_i = j)\right)\left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\left(\det\left(\Sigma_j^{(t)}\right)\right) - \frac{1}{2}\left(y_i - \mu_j^{(t)}\right)^\top\left(\Sigma_j^{(t)}\right)^{-1}\left(y_i - \mu_j^{(t)}\right)\right)$$

where

$$\mathcal{E}_{z\sim q\left(z|y,\theta^{(t)}\right)}\left(1(z_i=j)\right)=pr\Big(z_i=j|y,\theta^{(t)}\Big)=q\left(z_i=j|y_i,\varpi^{(t)},\mu^{(t)},\Sigma^{(t)}\right)$$
$$=\frac{\varpi_j^{(t)}\mathcal{N}(y_i|\mu_j^{(t)},\Sigma_j^{(t)})}{\sum_{j=1}^K\varpi_j^{(t)}\mathcal{N}(y_i|\mu_j^{(t)},\Sigma_j^{(t)})}$$

Let denote

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

So

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

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

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

3 Examples & code with Finite Normal Mixture Models

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

- `https://cran.r-project.org/web/packages/mixtools/index.html`
- `https://cran.r-project.org/web/packages/mixtools/vignettes/mixtools.pdf`

Example 28. Here are some R examples.

```
#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)
```

```
#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)
```

```
# NOdata
rm(list=ls())
#install.packages('mixtools')
library('mixtools')
data(NOdata)
plot(NOdata)
X <- NOdata 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)
```

4 Practice

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