Expectation-Maximisation Algorithm
Derivation and a cookbook for further recepies

Sven Laur
University of Tartu
email swen@math.ut.ee

May 8, 2019

1 Mixture model

To simplify the treatment, we consider here only how to derive and apply the expectation-maximisation algorithm in the context of soft clustering. The treatment of other applications is analogous. The soft-clustering is based on a probabilistic model, which combines k different data sources. More precisely, we assume that individual observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are generated by k different data sources $\mathcal{D}_1, \ldots, \mathcal{D}_k$. Each data source $\mathcal{D}_j$ is modelled by a probability distribution with parameters $\boldsymbol{\Theta}_j$ and thus we can always compute the likelihood $\mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j]$. Furthermore, we assume that all data points are independently sampled and the probability that $\boldsymbol{x}_i$ is generated by the data source $\mathcal{D}_j$ is λ_j. These parameters are also known as mixture proportions. Now let

$$\boldsymbol{\Theta} = (\lambda_1, \ldots, \lambda_k, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_k)$$

denote the set of all parameters. Then it is straightforward to estimate the likelihood of an individual observation

$$\mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}] = \sum_{j=1}^{k} \lambda_j \cdot \mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j] \ .$$

As a concrete example, consider a two-dimensional mixture model where both components are a two-dimensional normal distributions with parameters:

$$\Sigma_1 = \begin{pmatrix} 4.125 & -3.875 \\ -3.875 & 4.125 \end{pmatrix} \qquad \boldsymbol{\mu}_1 = \begin{pmatrix} 4 \\ -4 \end{pmatrix}$$
$$\Sigma_2 = \begin{pmatrix} 4.125 & 3.875 \\ 3.875 & 4.125 \end{pmatrix} \qquad \boldsymbol{\mu}_2 = \begin{pmatrix} -4 \\ 4 \end{pmatrix}$$

and equal mixture proportions:

$$\lambda_1 = 0.5 \qquad \lambda_2 = 0.5 \ .$$

The corresponding model is depicted in Figure 1.

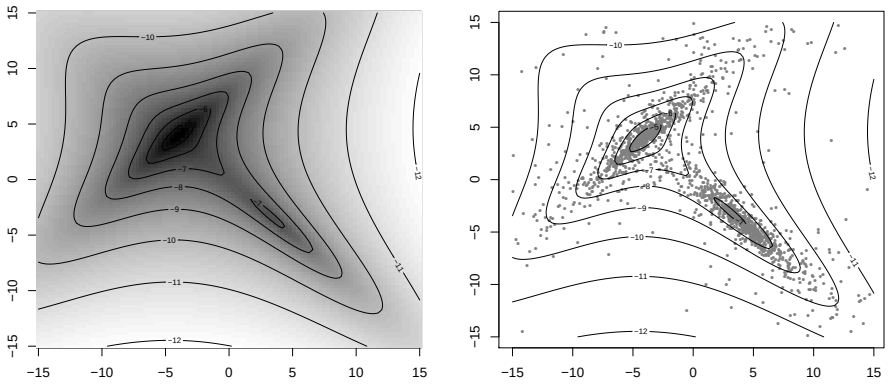

Figure 1: Two-dimesnional Gaussian mixture model. Probability density function is depicted on the left and 2000 element sample is depicted on the right.

2 Approximation of log-likelihood

Let $\boldsymbol{z} = (z_1, \ldots, z_n)$ denote a potential labelling of points where $z_i \in \{1, \ldots, k\}$ shows from which data source $\mathcal{D}_j$ the datapoint $\boldsymbol{x}_i$ is generated. For brevity, let

$$\mathcal{Z} = \{1, \ldots, k\}^n$$

denote the set of all possible labelings. Then the posterior distribution of model parameters

$$\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots \boldsymbol{x}_n] = \sum_{\boldsymbol{z} \in \mathcal{Z}} \mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \tag{1}$$

can be further decomposed by applying product rule to individual terms

$$\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right] \cdot \mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$$

where the first term assigns probabilities to individual labelings and the second term is the term we want to express[1]. Let us assume that we have somehow gotten a rather good estimate $q : \mathcal{Z} \to \mathbb{R}$ on the labelling probabilities, i.e.,

$$\forall z \in \mathcal{Z} : \qquad q(\boldsymbol{z}) \approx \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]$$

and we want to use and improve this estimation further. Without loss of generality, we can assume that the approximation is a proper probability distribution:

$$\sum_{\boldsymbol{z} \in \mathcal{Z}} q(\boldsymbol{z}) = 1 \ . \tag{2}$$

[1] Although it is formally correct we have gotten nowhere, yet

If $q(\boldsymbol{z})$ is not a probability distribution, we can always renormalise it. The latter does not change the values much, since

$$\sum_{\boldsymbol{z}\in\mathcal{Z}} \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{x_1}, \ldots, \boldsymbol{x}_n\right] = 1 \ .$$

Under this assumptions, we can approximate the sought posterior probability using individual labellings $\boldsymbol{z}$:

$$\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = \frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{\mathrm{p}[\boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]} \approx \frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{q(\boldsymbol{z})} \tag{3}$$

By substituting both approximations into the equation (1), we get another circular statement which is obvious:

$$\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots \boldsymbol{x}_n] = \sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) \cdot \frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{q(\boldsymbol{z})} \ .$$

To maximise this, we can take logarithms form both sides and maximise the right hand side of the following statement

$$\log \mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots \boldsymbol{x}_n] = \log \left(\sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) \cdot \frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{q(\boldsymbol{z})} \right) \ .$$

Since logarithm is a convex-cap function, the Jensen inequality allows us to push the logarithm through the sum to get a lower bound

$$F(q, \boldsymbol{\Theta}) := \sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) \cdot \log \left(\frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{q(\boldsymbol{z})} \right) \leq \log \mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots \boldsymbol{x}_n] \ . \tag{4}$$

As $q(\boldsymbol{z})$ is a good approximation of $\mathrm{p}[\boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, we can use the approximation (3) to show that the lower bound $F(q, \boldsymbol{\Theta})$ is close to to the actual value. Indeed, note that

$$\sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) \cdot \log \left(\frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{q(\boldsymbol{z})} \right) \approx \sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) \cdot \log\big(\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big)$$

which together with the equation (2) itself implies

$$\sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) \cdot \log \left(\frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{q(\boldsymbol{z})} \right) \approx \log\big(\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big)$$

To summarise, we have shown the following important claim.

Claim 1. *Assume that $q(\boldsymbol{z})$ is a good approximation of a posterior probability $\Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]$. Then the lower bound $F(q, \boldsymbol{\Theta}) \approx \log\big(\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big)$.*

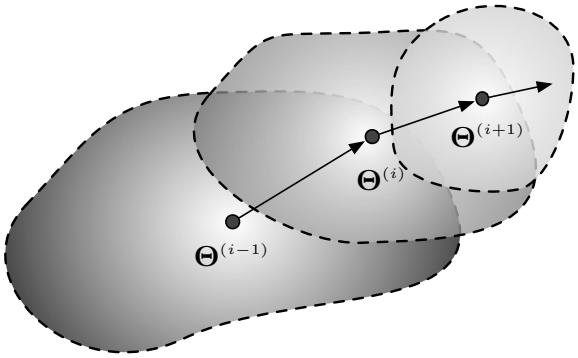

Figure 2: Iterative approximate search for maximal a posteriori estimate. Given a current solution $\boldsymbol{\Theta}^{(i)}$, approximate the log-likelihood and find the next optimum $\boldsymbol{\Theta}^{(i+1)}$. The algorithm is guaranteed to work if the new solution $\boldsymbol{\Theta}^{(i+1)}$ is inside the grey region where the approximation is good. In each step, we have to recalibrate the approximation by setting $q(\boldsymbol{z}) = \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}^{(i)}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]$.

3 Iterative minimisation algorithm

In most cases, exact minimisation of the log-likelihood $\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ is intractable and even the exact computation of partial derivatives for the iterative minimisation is a challenging task. Hence, the iterative minimisation of $F(q, \boldsymbol{\Theta})$ together with the recalibration of the approximation $q(\boldsymbol{z})$ on the posterior probability $\Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]$ is a compelling alternative, see Figure 2.

Let us study this iterative maximisation procedure in more detail. In particular, let $q_i(\boldsymbol{z}) = \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}^{(i)}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]$ and $q_{i+1}(\boldsymbol{z}) = \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}^{(i+1)}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]$ be the approximations of the labelling distributions used in steps i and $i+1$. Then the next value of parameters is computed by maximising the lower bound:

$$\boldsymbol{\Theta}^{(i+1)} = \operatorname*{argmax}_{\boldsymbol{\Theta}} F(q_i, \boldsymbol{\Theta}) \ . \tag{5}$$

Let us assume further that $\boldsymbol{\Theta}^{(i+1)}$ is so close to $\boldsymbol{\Theta}^{(i)}$ that

$$q_i(\boldsymbol{z}) \approx q_{i+1}(\boldsymbol{z}) \ .$$

Then Claim 1 assures that the lower bound is rather close to the actual log-likelihood:

$$F(q_i, \boldsymbol{\Theta}^{(i+1)}) \approx \log\bigl(\mathrm{p}[\boldsymbol{\Theta}^{(i+1)}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\bigr) \ .$$

Moreover, the approximation becomes more and more accurate if the difference between individual steps converges to zero:

$$\left\|\boldsymbol{\Theta}^{(i)} - \boldsymbol{\Theta}^{(i+1)}\right\| \to 0 \ .$$

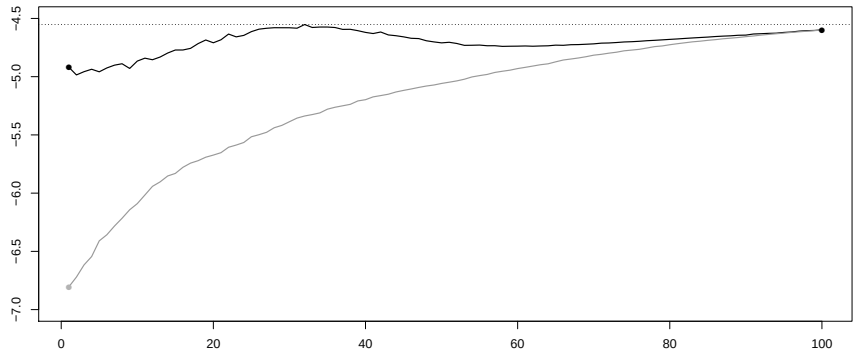

Figure 3: Convergence of the EM-algorithm. The x-axis depicts iterations. The y-axis depicts the values of the log-likelihood and its lower bound. The grey line shows how the lower bound increases during iterative minimisation. The black line shows the maximal value of the log-likelihood in a fixed size surrounding of $\boldsymbol{\Theta}^{(i)}$. The iterative maximisation quickly converges to a local maxima. However, it also misses a another maximum, since this maximum is rather far away from the search path and thus the approximation is not good enough to detect it.

As a direct consequence note that if we manage to show that the iterative maximisation procedure converges

$$\boldsymbol{\Theta}^{(i)} \to \boldsymbol{\Theta}^*$$

then also the lower bound converges to the log-likelihood:

$$F(q_i, \boldsymbol{\Theta}^{(i)}) \to \log\big(\mathrm{p}[\boldsymbol{\Theta}^* | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big)$$

Moreover, as $\boldsymbol{\Theta}^*$ is the local maximum of the lower bound $F(q_*, \boldsymbol{\Theta})$ then it must be also a local maximum of $\log\big(\mathrm{p}[\boldsymbol{\Theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big)$. Informal reasoning is the following. In a small surrounding of $\boldsymbol{\Theta}^*$, the approximation

$$F(q_*, \boldsymbol{\Theta}) \approx \log\big(\mathrm{p}[\boldsymbol{\Theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big)$$

is so good that the maximum condition

$$F(q_*, \boldsymbol{\Theta}) \leq F(q_*, \boldsymbol{\Theta}^*)$$

also implies the same for log-likelihoods

$$\log\big(\mathrm{p}[\boldsymbol{\Theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big) \leq \log\big(\mathrm{p}[\boldsymbol{\Theta}^* | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\big) \ .$$

These basic concepts behind the algorithm are illustrated in Figure 3.

1. Choose a suitable initial estimate for model parameters $\boldsymbol{\Theta}^{(0)}$.
2. E-STEP. Given an estimate $\boldsymbol{\Theta}^{(t)}$ update labelling probability:

$$\forall \boldsymbol{z} \in \mathcal{Z}: \qquad q_t(\boldsymbol{z}) = \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}^{(t)}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right] \ .$$

3. M-STEP. Find the next estimate $\boldsymbol{\Theta}^{(t+1)}$ by minimising the lower bound:

$$\boldsymbol{\Theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\Theta}} F(q_t, \boldsymbol{\Theta}) \ .$$

4. Repeat from the step 2 until convergence.

Figure 4: Expectation-maximisation algorithm

4 Expectation-Maximisation algorithm

Our treatment of the iterative minimisation algorithm described above was rather informal. In particular, we did not probe that the sequence of iterative updates leads to the sequence of estimates

$$\boldsymbol{\Theta}^{(0)}, \boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(i)}, \boldsymbol{\Theta}^{(i+1)} \ldots$$

that converges to a limiting value $\boldsymbol{\Theta}^*$. It is straightforward to see that

$$F(q_t, \boldsymbol{\Theta}^{(i)}) \leq F(q_t, \boldsymbol{\Theta}^{(t+1)}) \ .$$

since the update step (5) maximises $F(q_t, \boldsymbol{\Theta})$. However, the recalibration of the labelling probabilities:

$$q_{t+1}(\boldsymbol{z}) = \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}^{(t+1)}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]$$

is completely detached form $F(q, \boldsymbol{\Theta})$ and thus we must do further analysis to show

$$F(q_t, \boldsymbol{\Theta}^{(t+1)}) \leq F(q_{t+1}, \boldsymbol{\Theta}^{(t+1)}) \ ,$$

which would assure the convergence since the log-likelihood of the model must be bounded and thus also its lower bound $F(q, \boldsymbol{\Theta})$. The corresponding proof can be obtained through straightforward manipulations

$$\begin{aligned} F(q_{t+1}, \boldsymbol{\Theta}^{(t+1)}) &= \sum_{\boldsymbol{z} \in \mathcal{Z}} q_{t+1}(\boldsymbol{z}) \cdot \log\left(\frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}^{(t+1)}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{\Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}^{(t+1)}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]}\right) \\ &= \sum_{\boldsymbol{z} \in \mathcal{Z}} q(\boldsymbol{z}) \cdot \log\left(\mathrm{p}[\boldsymbol{\Theta}^{(t+1)}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\right) \\ &= \log\left(\mathrm{p}[\boldsymbol{\Theta}^{(t+1)}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\right) \end{aligned}$$

where the second equation from the Bayes equation

$$\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta}^{(t+1)}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = \Pr\left[\boldsymbol{z}|\boldsymbol{\Theta}^{(t+1)}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right] \cdot \mathrm{p}[\boldsymbol{\Theta}^{(t+1)}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \ .$$

Hence we have shown that after the recalibration step the lower bound matches the value it bounds:

$$F(q_{i+1}, \boldsymbol{\Theta}^{(t+1)}) = \log(\Pr\left[\boldsymbol{\Theta}^{(i+1)}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right])$$

and thus also

$$F(q_t, \boldsymbol{\Theta}^{(t+1)}) \leq F(q_{t+1}, \boldsymbol{\Theta}^{(t+1)}) \ .$$

Hence, we have shown the following claim.

Theorem 1. *The expectation-maximisation algorithm is guaranteed to converge if the log-likelihood function* $\log(\mathrm{p}[\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n])$ *is bounded.*

Proof sketch. Since both steps of the algorithm increase the lower bound and the log-likelihood is bounded the sequence of values

$$F(q_1, \boldsymbol{\Theta}^{(1)}), \ldots, F(q_i, \boldsymbol{\Theta}^{(i)}), \ldots$$

must converge to a limiting value. Hence the algorithm will stop. It is much more difficult to formally show that the convergence also takes place in terms of parameters $\boldsymbol{\Theta}^{(0)}, \boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(i)}, \ldots$, as large change in arguments might lead to small change in the output. To guarantee convergence in parameters, we must assume regularity from the function, i.e., there are no plateaus of constant values, where the algorithm could get stuck. □

5 Semantics behind the maximisation step

The general description of the M-step is rather void of any meaning. In the following, we show how one can give a fractional semantics to this step. The main difference between the EM-algorithm and the hard clustering algorithm is in the E-step. The hard clustering algorithm chooses the most probable label z_i for the datapoint $\boldsymbol{x}_i$ and the proceeds to find optimal parameters for reach cluster. The soft-clustering algorithm assigns weights

$$w_{ij} = \Pr\left[z_i = j|\boldsymbol{x}_i, \boldsymbol{\Theta}^{(t)}\right]$$

to each data point and uses all points with appropriate weighs to re-estimate cluster parameters $\boldsymbol{\Theta}$. Next, we will show that the steps in the soft-clustering algorithm correspond to the E-step and the M-step.

First of all note that the recalibration rule in the E-step can be decomposed further as all data points are assumed to be independently drawn and labelled:

$$q_t(\boldsymbol{z}) = \frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n|\boldsymbol{\Theta}^{(t)}]}{\mathrm{p}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n|\boldsymbol{\Theta}^{(t)}]} = \prod_{i=1}^{n} \frac{\mathrm{p}[z_i, \boldsymbol{x}_i|\boldsymbol{\Theta}^{(t)}]}{\mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}^{(t)}]} = \prod_{i=1}^{n} \Pr\left[z_i|\boldsymbol{x}_i, \boldsymbol{\Theta}^{(t)}\right] \ ,$$

which itself can be restated in terms of weights

$$q_t(\boldsymbol{z}) = \prod_{i=1}^{n} w_{iz_i} \quad .$$

Secondly, we will decompose also the lower bound minimised in the M-step

$$\begin{aligned} F(q_t, \boldsymbol{\Theta}) &= \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\frac{\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}{q_t(\boldsymbol{z})}\right) \\ &= \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\right) - \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log(q_t(\boldsymbol{z})) \quad . \end{aligned}$$

As the right term is independent of $\boldsymbol{\Theta}$, we can maximise only the first term

$$F_1(\boldsymbol{\Theta}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\mathrm{p}[\boldsymbol{\Theta}, \boldsymbol{z} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]\right) \quad .$$

By the Bayes rule

$$\mathrm{p}[\boldsymbol{z}, \boldsymbol{\Theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = \mathrm{p}[\boldsymbol{z}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n | \boldsymbol{\Theta}] \cdot \frac{\mathrm{p}[\boldsymbol{\Theta}]}{\mathrm{p}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}$$

where by our assumption the prior on the model parameters is constant. Thus, the second sum in the following expression

$$F_1(\boldsymbol{\Theta}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\mathrm{p}[\boldsymbol{z}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n | \boldsymbol{\Theta}]\right) + \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\frac{\mathrm{p}[\boldsymbol{\Theta}]}{\mathrm{p}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]}\right)$$

is constant during the maximisation and we ignore it. Hence, we can maximise the following function

$$F_2(\boldsymbol{\Theta}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\mathrm{p}[\boldsymbol{z}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n | \boldsymbol{\Theta}]\right)$$

instead. Now we can apply the independence of $\boldsymbol{x}_i, z_i$ pairs and further simplify

$$\begin{aligned} F_2(\boldsymbol{\Theta}) &= \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\prod_{i=1}^{n} \mathrm{p}[\boldsymbol{x}_i, z_i | \boldsymbol{\Theta}]\right) \\ &= \sum_{i=1}^{n} \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\mathrm{p}[\boldsymbol{x}_i, z_i | \boldsymbol{\Theta}]\right) \\ &= \sum_{i=1}^{n} \sum_{\boldsymbol{z} \in \mathcal{Z}} q_t(\boldsymbol{z}) \cdot \log\left(\mathrm{p}[\boldsymbol{x}_i, z_i | \boldsymbol{\Theta}]\right) \\ &= \sum_{i=1}^{n} \sum_{\boldsymbol{z} \in \mathcal{Z}} \prod_{\ell=1}^{n} w_{\ell z_\ell} \cdot \log\left(\mathrm{p}[\boldsymbol{x}_i, z_i | \boldsymbol{\Theta}]\right) \quad . \end{aligned}$$

For any fixed i value, the inner sum

$$\sum_{\boldsymbol{z}\in\mathcal{Z}}\prod_{\ell=1}^{n} w_{\ell z_\ell}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i|\boldsymbol{\Theta}]\right)$$

can be viewed as a standard sum over products

$$\sum_{\boldsymbol{z}\in\mathcal{Z}}\prod_{\ell=1}^{n} a_{\ell z_\ell} = \prod_{\ell=1}^{n}\sum_{j=1}^{k} a_{\ell j}$$

where

$$a_{\ell z_\ell} = \begin{cases} w_{\ell z_\ell} & \text{if } i \neq \ell \ , \\ w_{i z_i}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i|\boldsymbol{\Theta}]\right) & \text{if } i = \ell \ , \end{cases}$$

we get that

$$\sum_{\boldsymbol{z}\in\mathcal{Z}}\prod_{\ell=1}^{n} w_{\ell z_\ell}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i|\boldsymbol{\Theta}]\right) = \prod_{\substack{\ell=1\\ \ell\neq i}}^{n}\left(\sum_{j=1}^{k} w_{\ell j}\right)\cdot\left(\sum_{j=1}^{k} w_{ij}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i=j|\boldsymbol{\Theta}]\right)\right) \ .$$

Since all weights $w_{\ell j}$ connected to the data point $\boldsymbol{x}_\ell$ sum up to one we get

$$\sum_{\boldsymbol{z}\in\mathcal{Z}}\prod_{\ell=1}^{n} w_{\ell z_\ell}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i=j|\boldsymbol{\Theta}]\right) = \sum_{j=1}^{k} w_{ij}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i=j|\boldsymbol{\Theta}]\right)$$

and thus the triple sum simplifies further

$$F_2(\boldsymbol{\Theta}) = \sum_{i=1}^{n}\sum_{\boldsymbol{z}\in\mathcal{Z}}\prod_{\ell=1}^{n} w_{\ell z_\ell}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i|\boldsymbol{\Theta}]\right) = \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i=j|\boldsymbol{\Theta}]\right) \ .$$

Hence, we have obtained the following claim.

Claim 2. *The maximisation task in the M-step is equivalent to the simplified maximisation task*

$$F_2(\boldsymbol{\Theta}) = \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}\cdot\log\left(\mathrm{p}[\boldsymbol{x}_i, z_i=j|\boldsymbol{\Theta}]\right) \to \max \ .$$

General recipe: The simplified minimisation task provides you a general way to derive the update formulae for the M-step.

In the clustering context, where the likelihood of $\boldsymbol{x}_i$ is determined completely by the cluster parameters $\boldsymbol{\Theta}_j$, which generate the point, the task can be further

simplified:

$$F_2(\boldsymbol{\Theta}) = \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij} \cdot \log\left(\mathrm{p}[\boldsymbol{x}_i|z_i = j, \boldsymbol{\Theta}] \cdot \Pr\left[z_i = j|\boldsymbol{\Theta}\right]\right)$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij} \cdot \log\left(\mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j]\right) + \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij} \cdot \log\left(\Pr\left[z_i = j|\lambda_j\right]\right) \quad .$$

Recipe for mixture proportions: As both terms can be independently maximised, we can directly find optimal update for the mixture proportions:

$$\lambda_j = \frac{\sum_{i=1}^{n} w_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}} \quad .$$

If mixture proportions are fixed by the model, then the update step is not possible and is ignored.

Recipe for cluster parameters: To fix the cluster parameters, we must minimise the following terms

$$G_j(\boldsymbol{\Theta}_j) = \sum_{i=1}^{n} w_{ij} \cdot \log\left(\mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j]\right) \quad .$$

Connection with factional counts: In many cases, direct derivation of optimal parameters is not analytical and there is only iterative minimisation algorithm for hard clustering problem:

$$H(\boldsymbol{\Theta}_j) = \sum_{i\in\mathcal{I}_j} \log \mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j] \to \min$$

where $\mathcal{I}_j$ is the set of points assigned to the cluster. Note that we can rewrite this as a double sum using indicator functions

$$H(\boldsymbol{\Theta}_j) = \sum_{i=1}^{n} [i \in \mathcal{I}_j] \cdot \log \mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j] \to \min \quad .$$

Since some points can coincide, i.e. $\boldsymbol{x}_i = \boldsymbol{x}_\ell$, then the procedure that can find a solution for $H(\boldsymbol{\Theta})$ can find a solution for

$$H_*(\boldsymbol{\Theta}_j) = \sum_{i=1}^{n} c_{ij} \cdot \log \mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j] \to \min \quad .$$

where c_{ij} are integer counts. We just have to duplicate $\boldsymbol{x}_i$ c_{ij} times to generate modified data set for which the original procedure knows how to find the maximising assignment. Obviously, we can approximate $G_j(\boldsymbol{\Theta}_j)$ by the following

sum with integer coefficients

$$G_j(\boldsymbol{\Theta}_j) \approx \frac{1}{\ell} \cdot \sum_{i=1}^{n} \sum_{j=1}^{k} \mathsf{round}(w_{ij} \cdot \ell) \cdot \mathrm{p}[\boldsymbol{x}_i | \boldsymbol{\Theta}_j]$$

where the size of ℓ determines the approximation precision. As a result, we can use the update step from the hard-clustering algorithm to maximise

$$H_*(\boldsymbol{\Theta}_j) = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathsf{round}(w_{ij} \cdot \ell) \cdot \mathrm{p}[\boldsymbol{x}_i | \boldsymbol{\Theta}_j] \ .$$

6 Examples

6.1 Gaussian mixture model

Recall that the closed form solution for updating the parameters of the Gaussian cluster are

$$\boldsymbol{\mu_j} = \frac{1}{|\mathcal{I}_j|} \cdot \sum_{i \in \mathcal{I}_j} \boldsymbol{x}_i$$

$$\Sigma_j = \frac{1}{|\mathcal{I}_j|} \cdot X_c^T X_c$$

where X_c is matrix consisting of centred row vectors $\boldsymbol{x}_i - \boldsymbol{\mu}_j$ belonging to the cluster. Now it is straightforward to verify that with integral multiplicity c_{ij} the formula for the cluster centres becomes

$$n_j = \sum_{i=1}^{n} c_{ij}$$

$$\boldsymbol{\mu_j} = \frac{1}{n_j} \cdot \sum_{i=1}^{n} c_{ij} \boldsymbol{x}_i$$

and thus the optimal solution for soft-clustering task is

$$n_j = \sum_{i=1}^{n} w_{ij}$$

$$\boldsymbol{\mu_j} = \frac{1}{n_j} \cdot \sum_{i=1}^{n} w_{ij} \boldsymbol{x}_i$$

The analysis of the shape parameter is a bit more involved, because we must understand how replicated rows in the centred matrix X_c affect Σ_j. Let us do this by the example. Assume that the original centred matrix is 2×5 matrix

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{pmatrix}$$

Then the corresponding covariance matrix is computed as

$$\Sigma = \frac{1}{2} \cdot \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \\ a_5 & b_5 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{pmatrix}$$

If the first row (data point) has has multiplicity 2 and the second multiplicity 3 then the corresponding covariance matrix is computed

$$\Sigma = \frac{1}{5} \cdot \begin{pmatrix} a_1 & a_1 & b_1 & b_1 & b_1 \\ a_2 & a_2 & b_2 & b_2 & b_2 \\ a_3 & a_3 & b_3 & b_3 & b_3 \\ a_4 & a_4 & b_4 & b_4 & b_4 \\ a_5 & a_5 & b_5 & b_5 & b_5 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{pmatrix}$$

and thus the element

$$\Sigma_{ij} = \frac{1}{5} \cdot (a_i a_j + a_i a_j + b_i b_j + b_i b_j + b_i b_j) = \frac{1}{5} \cdot (2a_i a_j + 3b_i b_j)$$

The same result can be obtained if we multiply

$$\Sigma = \frac{1}{5} \cdot \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \\ a_5 & b_5 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{pmatrix}$$

Hence we have empirically derived the following update rule

$$\Sigma_j = \frac{1}{n_j} \cdot X_c^T \mathsf{diag}(\boldsymbol{w}_{*,j}) X_c$$

where $\mathsf{diag}(\boldsymbol{w}_{*,j})$ is a diagonal matrix with elements $w_{1j}, \ldots, w_{nj}$.

6.2 Laplacian distribution

For this cluster model we do not have closed form solution for the shape parameters but we do have a closed form solution for the cluster centres. Namely,

$$\mu_{j\ell} = \mathsf{median}(x_{1k}, \ldots, x_{nk})$$

Hence, we can quickly derive what is the corresponding update step if the data points have multiple copies. In this case, we have the following scheme

$$\underbrace{a_1, \ldots, a_1}_{c_{1j}}, \underbrace{a_2, \ldots, a_2}_{c_{2j}}, \ldots, \underbrace{a_n, \ldots, a_n}_{c_{nj}}$$

where $a_1, \ldots, a_n$ are ordered increasingly and we have to choose the middle element. This lead to the following curious update step

1. Order the elements increasingly $a_1, \ldots, a_n$

2. Let $w_1, \ldots, w_n$ be the corresponding weights. Compute cumulative sums $d_i = w_1 + \ldots w_i$ and choose the element a_k for which $d_{k-1} < 0.5$ and $d_k > 0.5$. If there is a tie $d_k = 0.5$ then return $\frac{1}{2} \cdot (a_{k-1} + a_k)$.