Notes on Expectation-Maximization

Paul F. Roysdon, Ph.D.

I. Introduction

Expectation-Maximization (EM) [1], [2] is an iterative optimization algorithm (unsupervised learning) to estimate some unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^n$, given measurements $\mathbf{u} \in \mathbb{R}^n$, and "hidden" (nuisance) variables $\boldsymbol{\nu} \in \mathbb{R}^n$, that need to be integrated out. Unlike Maximum Likelihood Estimation (MLE) where all data must be present to estimate the parameter $\boldsymbol{\theta}$, the EM algorithm makes a guess of the parameter – accounting for missing data – then adjusts the model to fit the parameter and the data.

To maximize the posterior probability of the parameters $\boldsymbol{\theta}$, given the data $\mathbf{u}$, marginalizing over $\boldsymbol{\nu}$, then

$$\hat{\boldsymbol{\theta}}^* = \underset{\boldsymbol{\theta}}{arg\,max} \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} p(\boldsymbol{\theta}, \boldsymbol{\nu}|\mathbf{u}) \tag{1}$$

where $\hat{\boldsymbol{\theta}}^*$ is the optimal estimate of true parameter $\boldsymbol{\theta}$, and $\mathcal{V}^n$ is the space of hidden variables $\boldsymbol{\nu}$.

Because the optimization is a function of two variables $(\boldsymbol{\theta}, \boldsymbol{\nu})$, the optimization *could* alternate between estimating the unknowns $\boldsymbol{\theta}$ and the hidden variables $\boldsymbol{\nu}$. Instead of finding the best $\boldsymbol{\nu} \in \mathcal{V}^n$ given an estimate $\hat{\boldsymbol{\theta}}$ at each iteration, EM computes a *distribution* over the space $\mathcal{V}^n$. Furthermore, given a random initialization, EM is guaranteed to converge to a local maximum (proof provided in [1]).

II. Derivation

The EM algorithm can be derived in many ways. The authors of [3] provide an intuitive explanation whereby the Expectation-step (E-step) can be interpreted as constructing a local lower-bound to the posterior distribution, whereas the Maximization-step (M-step) optimizes the bound, thereby improving the estimate for the unknowns.

We will first consider a *lower-bound* for the optimal estimate $\hat{\boldsymbol{\theta}}^*$, then an *optimal-bound* for which only a unique solution of $\hat{\boldsymbol{\theta}}$ exists, and finally maximize that bound.

A. Lower-Bound

From eqn. 1, maximize the logarithm of the joint distribution, which is proportional to the posterior,

$$\begin{aligned}\hat{\boldsymbol{\theta}}^* &= \underset{\boldsymbol{\theta}}{arg\,max} \;\; \log p(\mathbf{u}, \boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{arg\,max} \;\; \log \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} p(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}).\end{aligned} \tag{2}$$

Starting with an initial estimate (guess) $\hat{\boldsymbol{\theta}}^{(i)}$ for the parameters $\boldsymbol{\theta}$, and $(\cdot)^{(i)}$ denotes the iteration number, compute a lower bound $B(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(i)})$ to the function $p(\boldsymbol{\theta}, \mathbf{u})$, and maximize that bound. If iterated for $i = 1, \ldots, \tau$, eqn. 2 will converge to a local maximum.

The lower bound $B(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(i)})$ can be re-written as a sum of logarithms,

$$\begin{aligned}\log p(\mathbf{u}, \boldsymbol{\theta}) &= \log \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} p(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}) \\ &= \log \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu}) \frac{p(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta})}{f^{(i)}(\boldsymbol{\nu})},\end{aligned}$$

where $f^{(i)}(\boldsymbol{\nu})$ is an arbitrary probability distribution over the space $\mathcal{V}^n$. By Jensen's inequality [1], we have

$$\begin{aligned}\log B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)}) &\triangleq \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu}) \;\; \log \frac{p(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta})}{f^{(i)}(\boldsymbol{\nu})} \\ &\leq \log \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu}) \frac{p(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta})}{f^{(i)}(\boldsymbol{\nu})},\end{aligned}$$

transforming the log of sums into a sum of logs.

B. Optimal-Bound

Since we know $B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)})$ to be the lower bound, the optimal bound at $\boldsymbol{\theta}^{(i)}$ is found by maximizing

$$B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)}) = \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu}) \;\; \log \frac{p(\mathbf{u}, \boldsymbol{\nu}, \hat{\boldsymbol{\theta}}^{(i)})}{f^{(i)}(\boldsymbol{\nu})} \tag{3}$$

w.r.t. the distribution $f^{(i)}(\boldsymbol{\nu})$. To enforce the constraint $\sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu}) = 1$, introduce a Lagrange multiplier λ, i.e. solve the constrained optimization problem. The objective function $\mathcal{J}(f^{(i)})$ is

$$\begin{aligned}\mathcal{J}(f^{(i)}) = \lambda &\left[1 - \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu})\right] \\ &+ \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu}) \;\; \log p(\mathbf{u}, \boldsymbol{\nu}, \hat{\boldsymbol{\theta}}^{(i)}) \\ &- \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} f^{(i)}(\boldsymbol{\nu}) \;\; \log f^{(i)}(\boldsymbol{\nu}).\end{aligned} \tag{4}$$

The derivative of eqn. 4 w.r.t. $f^{(i)}(\boldsymbol{\nu})$ is

$$\frac{\partial \mathcal{J}}{\partial f^{(i)}(\boldsymbol{\nu})} = -\lambda + \log p(\mathbf{u}, \boldsymbol{\nu}, \hat{\boldsymbol{\theta}}^{(i)}) - \log f^{(i)}(\boldsymbol{\nu}) - 1.$$

Solving for $f^{(i)}(\boldsymbol{\nu})$

$$\begin{aligned}f^{(i)}(\boldsymbol{\nu}) &= \frac{p(\mathbf{u}, \boldsymbol{\nu}, \hat{\boldsymbol{\theta}}^{(i)})}{\sum_{\boldsymbol{\nu} \in \mathcal{V}^n} p(\mathbf{u}, \boldsymbol{\nu}, \hat{\boldsymbol{\theta}}^{(i)})} \\ &= p(\boldsymbol{\nu}|\mathbf{u}, \hat{\boldsymbol{\theta}}^{(i)}).\end{aligned} \tag{5}$$

Evaluating eqn. 3 using the result in eqn. 5

$$B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)}) = \sum_{\boldsymbol{\nu} \in \mathcal{V}^n} p(\boldsymbol{\nu}|\mathbf{u}, \hat{\boldsymbol{\theta}}^{(i)}) \ \log \frac{p(\mathbf{u}, \boldsymbol{\nu}, \hat{\boldsymbol{\theta}}^{(i)})}{p(\boldsymbol{\nu}|\mathbf{u}, \hat{\boldsymbol{\theta}}^{(i)})}$$
$$= \log p(\mathbf{u}, \hat{\boldsymbol{\theta}}^{(i)}),$$

and therefore the optimal bound $\hat{\boldsymbol{\theta}}^{(i)}$ indeed touches the objective function.

C. Maximizing the Bound

To maximize $B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)})$ w.r.t. $\boldsymbol{\theta}$, compute the expected value, $E\langle\cdot\rangle$, w.r.t. $f^{(i)}(\boldsymbol{\nu}) \triangleq p(\boldsymbol{\nu}|\mathbf{u}, \hat{\boldsymbol{\theta}}^{(i)})$, such that

$$B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)}) \triangleq E\langle \log p(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta})\rangle + \mathcal{E}$$
$$= E\langle \log p(\mathbf{u}, \boldsymbol{\nu}|\boldsymbol{\theta})\rangle + \log p(\boldsymbol{\theta}) + \mathcal{E}$$
$$= \mathcal{Q}^{(i)}(\boldsymbol{\nu}) + \log p(\boldsymbol{\theta}) + \mathcal{E},$$

where $\mathcal{Q}^{(i)}(\boldsymbol{\nu})$ is the expected *complete* log-likelihood, $p(\boldsymbol{\theta})$ is the prior on the parameters $\boldsymbol{\theta}$, and $\mathcal{E} = -E\left\langle \log f^{(i)}(\boldsymbol{\nu})\right\rangle$ is the entropy of the distribution $f^{(i)}(\boldsymbol{\nu})$. Because $\mathcal{E}$ is independent of $\boldsymbol{\theta}$, the maximization simplifies to

$$\hat{\boldsymbol{\theta}}^{(i+1)} = \underset{\boldsymbol{\theta}}{arg\,max}\, B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)})$$
$$= \underset{\boldsymbol{\theta}}{arg\,max} \left[\mathcal{Q}^{(i)}(\boldsymbol{\nu}) + \log p(\boldsymbol{\theta})\right]. \quad (6)$$

D. Expectation-Maximization

At each iteration, the EM algorithm seeks an optimal lower bound $B(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(i)})$ at the current guess $\hat{\boldsymbol{\theta}}^{(i)}$, eqn. 3, then maximizes this bound to obtain an improved estimate $\hat{\boldsymbol{\theta}}^{(i+1)}$, eqn. 6. In summary:

- E-step: calculate $f^{(i)}(\boldsymbol{\nu}) \triangleq p(\boldsymbol{\nu}|\mathbf{u}, \hat{\boldsymbol{\theta}}^{(i)})$.
- M-step: $\hat{\boldsymbol{\theta}}^{(i+1)} = \underset{\boldsymbol{\theta}}{arg\,max} \left[\mathcal{Q}^{(i)}(\boldsymbol{\nu}) + \log p(\boldsymbol{\theta})\right]$.

Note that the E-step $\mathcal{Q}^{(i)}(\boldsymbol{\nu})$ is calculated by evaluating $f^{(i)}(\boldsymbol{\nu})$ at the current estimate $\hat{\boldsymbol{\theta}}^{(i)}$. However, the M-step optimizes $\mathcal{Q}^{(i)}(\boldsymbol{\nu})$ w.r.t. the *free variable* $\boldsymbol{\nu}$ to obtain the new estimate $\hat{\boldsymbol{\theta}}^{(i+1)}$.

III. Example

A. Gaussian Mixture Model

To demonstrate the EM algorithm, we use the example of a Gaussian Mixture Model (GMM). A GMM is a model consisting of m-unique Gaussian distributions. Recall, given the random variable $x \sim \mathcal{N}(\mu, \sigma)$, the probability distribution function is

$$p(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean, σ is the standard deviation, and $\mathcal{N}(\cdot)$ denotes a Normal or Gaussian distribution.

Let the sum of the values expected by all m Gaussians be defined as

$$f(\mathbf{x}) \sum_{i=1}^{m} \alpha_i \cdot \phi(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

where $\phi(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a Gaussian function with mean $\boldsymbol{\mu}_i \in \mathbb{R}^{n\times 1}$, covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^n$, and weights α_i, where $\sum_{i=1}^m \alpha_i = 1$.

Let the probability that input $\mathbf{x}_j$ belongs to class, or distribution, c_i be defined

$$p(\mathbf{x}_j \in c_i) = \frac{\hat{\alpha}_i \cdot \phi(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)}{\sum_{k=1}^m \hat{\alpha}_i \cdot \phi(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}. \quad (7)$$

Given a set of points, we seek the estimate of the weights $\hat{\alpha}_i$. The solution is found iteratively by EM.

Assume that two Gaussian distributions are randomly generated,

$$G_1 = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2)$$
$$G_2 = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2^2).$$

Given a random variable from one of the two distributions, the probability of selecting the correct distribution is p, such that the total probability is

$$y = pG_1 + (1-p)G_2.$$

If the probability *distribution* of p is denoted as $\boldsymbol{\gamma}$, then the probability *density* is

$$P(\mathbf{y}) = \boldsymbol{\gamma} \cdot \phi(\mathbf{y}; \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1) + (1-\boldsymbol{\gamma}) \cdot \phi(\mathbf{y}; \boldsymbol{\mu}_2, \boldsymbol{\sigma}_2)$$

Introducing a new variable f enables us to determine the correct distribution, such that $f = 0$ for data from distribution one, i.e. G_1, and $f = 1$ for G_2.

B. Illustration

Using the method defined in Section III-A, we define three Gaussian mixture models varying from highly overlapped, to distinctly separate: see (red dots) Figs. 1, 3 and 5.

Applying the k-Means algorithm to the data, we can quickly identify the centroid of each distribution (cluster), however we know nothing about the standard deviation from that mean. The k-Means results (blue dots) are shown in Figs. 1, 3 and 5.

Alternatively, if we assume that the data is Normally distributed, we can treat the data as a GMM and solve the mean and standard deviation via the EM algorithm. During the execution of EM, the log-likelihood is computed. The algorithm terminates if either the change in log-likelihood is below a threshold, $\Delta\mathcal{L} < \epsilon$, or the maximum number of iterations is reached (here $iter_{max} = 500$).

The results (black dots and ellipses) of the EM are shown in Figs. 1, 3 and 5, producing estimates $(\hat{\mu}_1, \hat{\sigma}_1^2)$ and $(\hat{\mu}_2, \hat{\sigma}_2^2)$.

Notice when the Gaussians are highly overlapped, the EM algorithm requires more iterations (Fig. 2) than when they are separate (Fig. 6). Also notice that the k-Means algorithm computes a less accurate result (compared to EM) when the data is highly overlapped (Fig. 1), whereas the result of k-Means and EM are nearly identical when the distributions are distinct (Fig. 5). This is because k-Means computes a centroid of the distribution using a distance equation, versus estimating the actual parameters $\boldsymbol{\theta}$ that generated the distributions – a much more accurate estimate of the centroid.

Using the estimated parameters, $\hat{\theta}$, future predictions can be made based on historical data. Notice in Figs. 2, 4 and 6, the log-likelihood appears to "plateau", and the log-likelihood function is monotonically increasing with each iteration, this confirms our earlier claim.

A final note: the Gaussian assumption was made to use the derivations provided in Section III-A. If a different distribution, say a Poisson distribution, makes more sense for the data, then the EM algorithm must be modified accordingly.

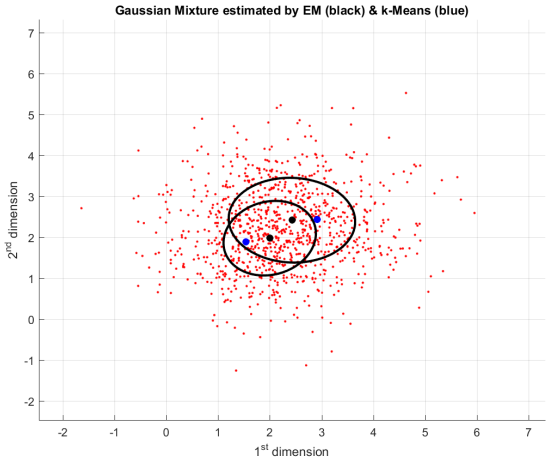

Fig. 1. Test 1: Gaussian mixture (red) estimated by k-Means (blue) and GMM (black).

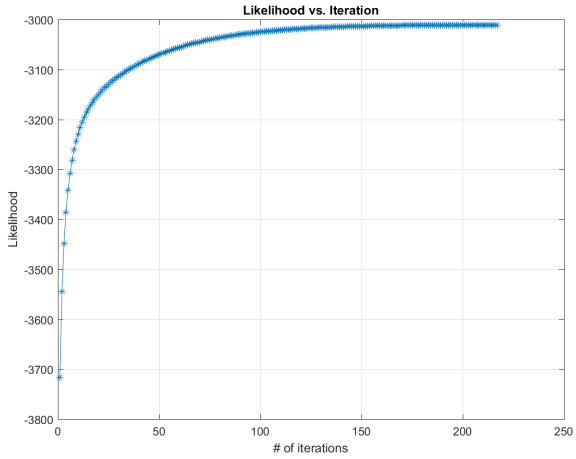

Fig. 2. Test 1: Log-Likelihood vs. EM iterations.

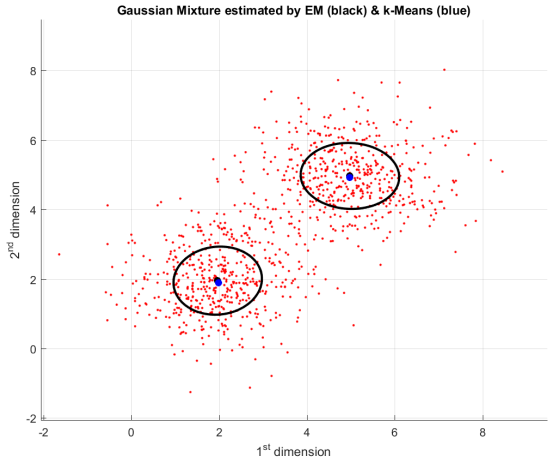

Fig. 3. Test 2: Gaussian mixture (red) estimated by k-Means (blue) and GMM (black).

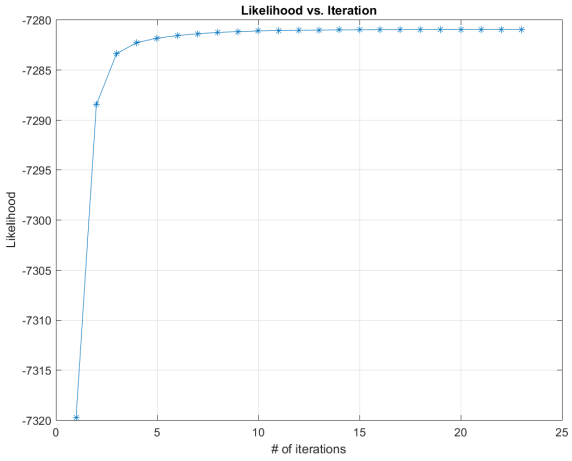

Fig. 4. Test 2: Log-Likelihood vs. EM iterations.

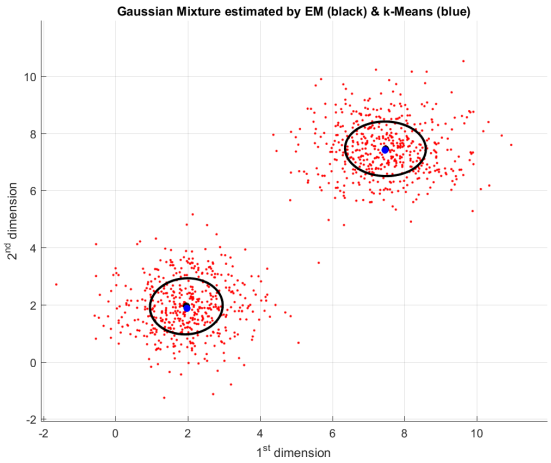

Fig. 5. Test 3: Gaussian mixture (red) estimated by k-Means (blue) and GMM (black).

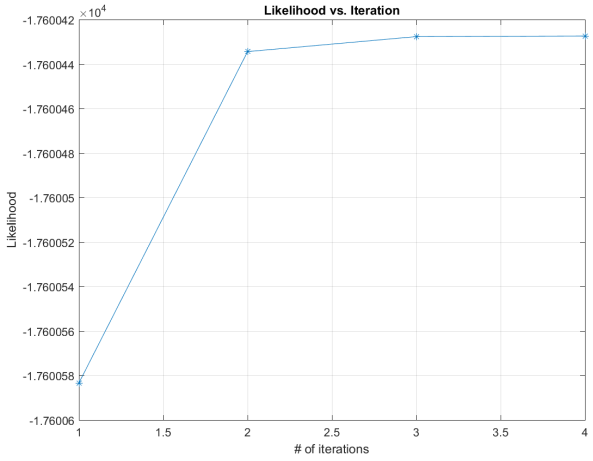

Fig. 6. Test 3: Log-Likelihood vs. EM iterations.

REFERENCES

[1] L. N. Dempster, A. and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society, Series B*, vol. 39(1), pp. 1–38, 1977.

[2] G. McLachlan and T. Krishnan, *The EM algorithm and extensions.* Wiley, Series in Probability and Statistics., 1997.

[3] R. Neal and G. Hinton, *A view of the EM algorithm that justifies incremental, sparse, and other variants.* Learning in Graphical Models. Kluwer Academic Press., 1997.