EM-algorithm

Max Welling

California Institute of Technology 136-93
Pasadena, CA 91125
welling@vision.caltech.edu

1 Introduction

In the previous class we already mentioned that many of the most powerfull probabilistic models contain hidden variables. We will denote these variables with $\mathbf{y}$. It is usually also the case that these models are most easily written in terms of their joint density,

$$p(\mathbf{d}, \mathbf{y}, \boldsymbol{\theta}) = p(\mathbf{d}|\mathbf{y}, \boldsymbol{\theta})\ p(\mathbf{y}|\boldsymbol{\theta})\ p(\boldsymbol{\theta}) \tag{1}$$

Remember also that the objective function we want to maximize is the log-likelihood (possibly including the prior term like in MAP-estimation) given by,

$$\begin{aligned} L(\mathbf{d}, \boldsymbol{\theta}) &= \log[p(\mathbf{d}|\boldsymbol{\theta})] + \log[p(\boldsymbol{\theta})] & (2)\\ &= \log[\int d\mathbf{y}\ p(\mathbf{d}, \mathbf{y}|\boldsymbol{\theta})] + \log[p(\boldsymbol{\theta})] & (3) \end{aligned}$$

Notice that maximum likelihood estimation is a special case, by neglecting the prior term (i.e. set $p(\boldsymbol{\theta}) = 1$). In the following we will include the $p(\boldsymbol{\theta})$, because it does not complicate the derivation and treats a slightly more general case. It is then very easy to switch to ML estimation by changing $p(..., \boldsymbol{\theta}) \rightarrow p(...|\boldsymbol{\theta})$. For unsupervised learning we can simply set $\mathbf{d} = \mathbf{x}^N$,

$$L(\mathbf{x}^N, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log[p(\mathbf{x}_n|\boldsymbol{\theta})] + \log[p(\boldsymbol{\theta})], \tag{4}$$

while for supervised learning we may put $\mathbf{d} = \{\mathbf{x}^N, \mathbf{t}^N\}$ and decompose,

$$L(\mathbf{t}^N, \mathbf{x}^N, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log[p(\mathbf{t}_n|\mathbf{x}_n, \boldsymbol{\theta})] + \sum_{n=1}^{N} \log[p(\mathbf{x}_n|\boldsymbol{\theta})] + \log[p(\boldsymbol{\theta})]. \tag{5}$$

In the case of supervised learning we may decide that we are not interested in modelling the input distribution $p(\mathbf{x}^N|\boldsymbol{\theta})$ and simply omit it (i.e. set it equal to one).

We could now directly take derivatives with respect to $\boldsymbol{\theta}$ of this likelihood functions and use them to find the maximum (for instance through gradient descent). It turns out that these equations can become quite hairy, and an easier method exists, called expectation maximization (EM). The main idea is that it is much easier to optimize the joint density $\log[p(\mathbf{d}, \mathbf{y}, \boldsymbol{\theta})]$ if we had known the values for $\mathbf{y}$. Unfortunately we don't know them and that was the reason we have integrated them out in the likelihood. Instead we will optimize the function,

$$Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \mathbf{E}[\log[p(\mathbf{d}, \mathbf{y}, \boldsymbol{\theta}_t)]\ |\mathbf{d}, \boldsymbol{\theta}_{t-1}], \tag{6}$$

i.e. given the data and the parameter values from a previous iteration. We therefore iterate,

E-step Calculate $Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$, given some parameter estimates from the previous iteration $\boldsymbol{\theta}_{t-1}$.

M-step Maximize $Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ over $\boldsymbol{\theta}_t$.

This procedure is guaranteed to improve the log-likelihood at every iteration.

2 Example

Let's consider the case where we have a random variable distributed according to a multonomial distribution,

$$p(n_1, n_2, n_3) = \frac{N!}{n_1!\ n_2!\ n_3!} p_1^{n_1} p_2^{n_2} p_3^{n_3} \tag{7}$$

with $n_1 + n_2 + n_3 = N$ and $p_1 + p_2 + p_3 = 1$. For instance, think of a jar with three different colours of balls. The probability of drawing a red ball is p_1, a green ball p_2, and a blue ball p_3. After we picked a ball we return it in the jar (i.e. the draws are independent). If we decide on picking N balls, the probability of a particular draw of n_1 red balls, n_2 green balls and n_3 blue balls is then described by the above distribution. The combinatorial factor adds the different possibilities to arrive at this draw. Now, let us assume that we have the following parametrized model for the probabilities of drawing the different colours,

$$p_1 = \frac{1}{4} \tag{8}$$

$$p_2 = \frac{1}{4} + \frac{p}{4} \tag{9}$$

$$p_3 = \frac{1}{2} - \frac{p}{4} \tag{10}$$

where p is thus the only parameter. Now assume that the man who is doing the experiment is actually colourblind and cannot discern the red from the green balls. He draws N balls, but only sees $m_1 = n_1 + n_2$ red/green balls and $m_2 = n_3$ blue balls. The question is, can the man still estimate the parameter p and with that in hand calculate his best guess for the number of red and green balls (obviously, he knows the number of blue balls). Fortunately the man is smart and maximizes the likelihood of his observations,

$$L(m_1, m_2) = \frac{N!}{m_1!\ m_2!}(p_1 + p_2)^{m_1} p_3^{m_2} \tag{11}$$

$$L(m_1, m_2) = \frac{N!}{m_1!\ m_2!}(\frac{1}{2} + \frac{p}{4})^{m_1}(\frac{1}{4} - \frac{p}{4})^{m_2} \tag{12}$$

which is now a binomial distribution, where the probability of drawing a red or a green ball is given by $p_1 + p_2$. Taking the logarithm and maximizing with respect to p gives,

$$p = 2\frac{m_1 - m_2}{m_1 + m_2} \tag{13}$$

and therefore he estimates his total number of red and green balls to be,

$$\mathbf{E}[n_1|m_1] = \frac{p_1}{p_1 + p_2}\ m_1 = \frac{1}{4}(m_1 + m_2) \tag{14}$$

$$\mathbf{E}[n_2|m_1] = \frac{p_2}{p_1 + p_2}\ m_1 = \frac{1}{4}(3\ m_1 - m_2) \tag{15}$$

Now let's see what the EM procedure would give us in this case. First we have to compute the average of the log of the conplete data pdf over the unobserved variables, given the observed variables. Therefore we need,

$$p(n_1, n_2|m_1) = \frac{m_1!}{n_1!\ n_2!}(\frac{p_1}{p_1 + p_2})^{n_1}(\frac{p_2}{p_1 + p_2})^{n_2} \tag{16}$$

First notice that n_3 is determined if m_2 is given (they are equal). For the red and green balls we know that the total must be given by m_2, and that each ball is drawn with a relative probability $\frac{p_1}{p_1+p_2}$ and $\frac{p_2}{p_1+p_2}$ respectively, hence they are distributed according to the above binomial distribution. Now let's look at the complete data likelihood. In our formalism we need to look at,

$$p(n_1, n_2, n_3, m_1, m_2) = \delta(m_1 - n_1 - n_2)\ \delta(m_2 - n_3)\ p(n_1, n_2, n_3). \tag{17}$$

The delta functions reflect the fact that the random variables m_i are deterministic functions of n_i. Although logarithms of delta functions have to be defined properly, for instance by describing them as the limit of a Gaussian, we may now reason that the logarithm changes the product into a sum of three terms, of which the terms which contain delta functions do not depend on any parameters to be optimized and can be omitted for that reason. The same holds for some terms in the expression for $p(n)$. Putting everything together we find,

$$L \propto \mathbf{E}[n_2|m_1] \log(\frac{1}{4} + \frac{\tilde{p}}{4}) + m_2 \log(\frac{1}{2} - \frac{\tilde{p}}{4}) \tag{18}$$

where we have

$$\mathbf{E}[n_2|m_1] = m_1 \; \frac{p_2}{p_1 + p_2} = m_1 \; \frac{1+p}{2+p} \tag{19}$$

This calculation is formally the E-step. Notice that $\tilde{p}$ denotes the "new" p which is going to be updated while p denotes the value inserted from the previous iteration. Now taking derivatives and equating to zero gives,

$$\tilde{p} = \frac{2\mathbf{E}[n_2|m_1] - m_2}{\mathbf{E}[n_2|m_1] + m_2} \tag{20}$$

which formally comprises the M-step. Iterating these equations will lead to the same answer as the analytical expression (see demo-EM).

3 EM-Algorithm

Consider a function $L_\theta(x)$ that needs to be maximized over the parameters θ, and assume that it can be written as follows,

$$L_\theta(x) = F_\theta[x,q] + R_\theta[x,q], \tag{21}$$

where $R_\theta[x,q]$ is a postive restterm, depending on the variable x and the *function* q, with the following properties,

$$R[x,q] \geq 0 \qquad \forall\, q \tag{22}$$

$$\exists \;\, p \;\; \Rightarrow \;\; R[x,p] = 0 \tag{23}$$

With these properties it is easy to prove that the following iterative scheme increases $L(x)$ at every iteration,

$$\begin{aligned} L_{\theta_t}(x) &= F_{\theta_t}[x, p_t] \\ &\leq F_{\theta_{t+1}}[x, p_t] \quad \text{maximize over } \theta_t \\ &= L_{\theta_{t+1}}(x) - R_{\theta_{t+1}}[x, p_t] \quad \text{using } p_t = q_{t+1} \\ &\leq L_{\theta_{t+1}}(x) \quad \text{since } R[x,q] \geq 0 \;\; \forall\, q. \end{aligned} \tag{24}$$

The mysterious part about this is of course where we get the function $R[x,q]$. In the following we will show that if we take L to be the log-likelihood, R can be easily derived, and the iterative scheme above can be used to maximize the log-likelihood. In the following $q(\mathbf{y})$ will represent an arbitrary probability distribution.

$$\begin{aligned} L(\mathbf{d}, \boldsymbol{\theta}) &= \log[p(\mathbf{d}, \boldsymbol{\theta})] \\ &= \int d\mathbf{y} \; q(\mathbf{y}) \; \log[p(\mathbf{d}, \boldsymbol{\theta})] \\ &= \int d\mathbf{y} \; q(\mathbf{y}) \; \log\left[\frac{p(\mathbf{d}, \mathbf{y}, \boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{d}, \boldsymbol{\theta})} \times \frac{q(\mathbf{y})}{q(\mathbf{y})}\right] \\ &= Q(q||p_{\mathbf{joint}}) + H(q||q) + KL(q||p_{\mathbf{post}}) \end{aligned} \tag{25}$$

where we used $\int d\mathbf{y} \; q(\mathbf{y}) = 1$ and we have defined

$$Q(q||p_{\mathbf{joint}}) = \int d\mathbf{y} \; q(\mathbf{y}) \log[p(\mathbf{d}, \mathbf{y}, \boldsymbol{\theta})] \tag{26}$$

$$H(q||q) = -\int d\mathbf{y} \; q(\mathbf{y}) \log[q(\mathbf{y})] \tag{27}$$

$$KL(q||p_{\mathbf{post}}) = \int d\mathbf{y} \; q(\mathbf{y}) \log[\frac{q(\mathbf{y})}{p(\mathbf{y}|\mathbf{d}, \boldsymbol{\theta})}] \tag{28}$$

The last term $KL(q||p_{\mathbf{post}})$ is the Kullback-Leibler distance between the arbitrary probability density $q(\mathbf{y})$ and the posterior density $p(\mathbf{y}|\mathbf{d},\boldsymbol{\theta})$. If we look back at our original optimization problem we see that we need to identify $F = Q + H$ and $KL = R$. Finally, we need to prove that the two properties of R hold. Both are direct consequences of the fact that the KL-term computes the distance between the distribution $q(\mathbf{y})$ and $p(\mathbf{y}|\mathbf{d},\boldsymbol{\theta})$, which is always larger than zero and equal to zero if $q(\mathbf{y}) = p(\mathbf{y}|\mathbf{d},\boldsymbol{\theta})$. The proof goes as follows.

It is easy to see that if we set

$$q(\mathbf{y}) = p(\mathbf{y}|\mathbf{d},\boldsymbol{\theta}), \quad \text{then} \quad R = 0, \tag{29}$$

confirming the first property. The second property is a direct concequence of Jensen's inequality,

$$\mathbf{E}[f(x)] \geq f(\mathbf{E}[x]) \tag{30}$$

for convex functions f. Using that $f(x) = -\log(x)$ is convex we write,

$$\begin{aligned} KL(q||p) &= \int dx\ q \left[\log(\frac{q}{p})\right] \\ &= \int dx\ q \left[-\log(\frac{p}{q})\right] \\ &\geq -\log\left[\int dx\ q\frac{p}{q}\right] \\ &= -\log \int dx\ p \\ &= -\log(1) \\ &= 0 \\ &\Rightarrow KL(q||p) \geq 0, \end{aligned} \tag{31}$$

confirming the second property.

We can now identify the E-step as substituting $q(\mathbf{y}) = p(\mathbf{y}|\mathbf{d},\boldsymbol{\theta}_{t-1})$ into Q and averaging $\log[p(\mathbf{d},\mathbf{y},\boldsymbol{\theta}_t)]$ over it, while the M-step consists maximizing Q with respect to $\boldsymbol{\theta}_t$. Notice that the M-step is equivalent to maximizing $F = Q + H$, since H does not depend $\boldsymbol{\theta}_t$.

We have shown that the above iterations will increase the log-likelihood at every iteration, but we haven't shown that the algorithm converges only when it has reached a local maximum of this likelihood. This proof is beyond the scope of this class.

We also notice that the above derivation is completely symmetric under the exchange $\mathbf{d} \leftrightarrow \boldsymbol{\theta}$. We could imagine a situation where we needed to maximize $p(\mathbf{x},\boldsymbol{\theta})$ over $\mathbf{x}$, given $\boldsymbol{\theta}$. EM could therefore also be employed to this problem, by using the same algorithm, but interchanging $\mathbf{d} = \mathbf{x}$ and $\boldsymbol{\theta}$.

4 Generalizations

From the above derivation it is also clear that we can perform partial M-steps. As long as each M-step improves Q, but not maximizes it, we are still guaranteed that the log-likelihood increases at every iteration. We could for instance use gradient ascent as a partial M-step. This algorithm is called "generalized EM" (GEM).

In the same spirit we can also perform partial E-steps. First notice that by substituting $q(\mathbf{y}) = p(\mathbf{y}|\mathbf{d},\boldsymbol{\theta}_{t-1})$ we have actually maximized F with respect to $q(\mathbf{y})$. This is easily seen by taking the derivative with respect to q and enforcing the fact that it integrates to 1 by a Lagrange multiplier term,

$$\begin{aligned} &\frac{\partial}{\partial q}\ F(q) + \lambda \int (q-1) \\ &= \frac{\partial}{\partial q}(\int\ q\ (\log[p_{\mathbf{joint}}] - \log[q] + \lambda) \\ &= \log[p_{\mathbf{joint}}] - \log[q] + \lambda - 1 \\ &\Rightarrow\ q \propto p_{\mathbf{joint}} \\ &\Rightarrow\ q(\mathbf{y}) = p(\mathbf{y}|\mathbf{d},\boldsymbol{\theta}) \end{aligned} \tag{32}$$

where the last line follows because q must be properly normalized. This derivation is slightly sloppy since we are taking derivatives of functionals with respect to functions, which would, when done properly, add a lot of notational burden while arriving at the same result. Thus, we see that a full E-step is done by maximizing F

with respect to q. This leads to the generalization that a partial E-step would involve improving F, without actually maximizing it. This can be done by modeling q as a function of parameters and possibly on the data, $q = q(\mathbf{y}|\mathbf{d}, \boldsymbol{\nu})$. Maximization over q is then performed by maximization over ν, for instance by gradient ascent. Only when the true posterior is included in this parameterized family, can we actually reach the true maximum likelihood solution. But, in case q is only an approximation to the true posterior at best, not all is lost. In that case we improve a bound on the log-likelihood since we have

$$F = L - KL \leq L \qquad \text{since} \quad KL \geq 0 \tag{33}$$

Maximizing F is thus equivalent to maximizing a lower bound on L. Maximizing F has two consequences. Firstly it results in maximizing L, and at the same time it will minimize KL, which is the KL-distance between the true posterior and our parametrized model of it, i.e. q is driven as close as possible (in the sense of Kl-distance) to the true posterior under the current parametrized model q. Naturally, this makes only sense when the true posterior is too difficult to calculate analytically. After averaging the joint density $p_{\mathbf{joint}}$ over the approximate posterior q, the M-step then maximizes F (or Q) over $\boldsymbol{\theta}_t$. Summarizing, the *variational* EM algorithm thus alternates the following steps

E-step Maximize F with respect to q.

M-step Maximize F (or Q) with respect to $\boldsymbol{\theta}$.

Notice that now we need to include the entropy term H in the E-step of the optimization, since it depends on q.

Apart from the above variational procedure there are alternative methods to approximate the E-step if it turns out too complicated. The most widely used methods are sampling techniques, which approximate the average by

$$\int d\mathbf{y}\ p(\mathbf{y})\ f(\mathbf{y}) \approx \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{y}_n) \tag{34}$$

Examples are "Markov Chain Monte Carlo" (MCMC) and "Gibbs" sampling. We will discuss these methods in more detail in later lectures.

Finally, we notice that the derivation of the EM algorithm did not depend on the fact that $\log[p(\mathbf{d}, \boldsymbol{\theta})]$ is a probability density. Of, course it has to be postive everywhere, since otherwise the logarithm would be undefined, but it does not depend in $p(\mathbf{d})$ being normalized. Notice also that the derivation did crucially depend on the fact that the posterior $p(\mathbf{y}|\mathbf{d})$ is a normalized probability density, otherwise the KL term is not guaranteed to be postive. These facts suggest a generalization to positive unnormalized functions $g(\mathbf{d})$ which can be written as,

$$g(\mathbf{d}) = \int d\mathbf{y}\ f(\mathbf{d}, \mathbf{y}), \tag{35}$$

with postive $f(\mathbf{d}, \mathbf{y})$. In this case we can define,

$$p(\mathbf{y}|\mathbf{d}) = \frac{f(\mathbf{d}, \mathbf{y})}{g(\mathbf{d})} = \frac{f(\mathbf{d}, \mathbf{y})}{\int d\mathbf{y}\ f(\mathbf{d}, \mathbf{y})} \tag{36}$$

which is automatically normalized (and positive) with respect to $\mathbf{y}$, i.e. it defines a probability density. We can now repeat the whole derivation for the objective function,

$$L' = \log[g(\mathbf{d}|\boldsymbol{\theta})] \tag{37}$$

where the E-step consists of evaluating the posterior through (36) and using it to average $f(\mathbf{x}, \mathbf{y})$, while the M-step maximizes this resultant average with respect to some parameters in $f(\mathbf{x}, \mathbf{y})$.

5 EM-Example

As an example we will consider incomplete data generated from a K-dimensional Gaussian density, $p(\mathbf{x}) = \mathcal{G}_{\mathbf{x}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}]$. The data are incomplete because for every sample the values of some dimensions may be missing. One can imagine a malfunctioning measuring device. This example is therefore different in spirit than the latent variable models we will encouter in future classes, since there the models are build with hidden variables and their values cannot be observed in principle. In the present case, the model is such that every sample could in principle be observed, but we were unlucky and had a malfunctioning sensor.

For every datapoint we will divide the vector $\mathbf{x}_n$ into an observed part and a missing part: $\mathbf{x}_n = (\mathbf{x}_n^o, \mathbf{x}_n^m)$. This notation does not imply that always the last part of the vector is missing. Any dimension could be missing. The log likelihood that we aim to optimize is thus given by,

$$L(\mathbf{x}_N^o|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \int d\mathbf{x}_n^m \; \mathcal{G}_{\mathbf{x}_n}[\boldsymbol{\mu}, \boldsymbol{\Sigma}] \tag{38}$$

Because the missing dimensions are different for every datapoint, this sum of integrals becomes messy. Therefore we will do EM. First write down Q,

$$Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \sum_{n=1}^{N} \int d\mathbf{x}_n^m \; p(\mathbf{x}_n^m|\mathbf{x}_n^o, \boldsymbol{\theta}_{t-1}) \; \log\left[p(\mathbf{x}_n^o, \mathbf{x}_n^m|\boldsymbol{\theta}_t)\right] \tag{39}$$

The log of the joint is determined by the Gaussian density,

$$\log\left[p(\mathbf{x}_n^o, \mathbf{x}_n^m|\boldsymbol{\theta}_t)\right] = -\frac{K}{2}\log(2\pi) - \frac{1}{2}\log\det[\boldsymbol{\Sigma}] - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}) \tag{40}$$

where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For the posterior we will use the following lemma,
Lemma Let $\mathcal{G}_{\mathbf{x}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}]$ denote a normal density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. If we write $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2]$ and $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}$, then we have,

$$p(\mathbf{x}_2|\mathbf{x}_1) = \mathcal{G}_{\mathbf{x}_2}[\boldsymbol{\mu}_2 - \boldsymbol{\Sigma}_{21}(\boldsymbol{\Sigma}_{11})^{-1}(\boldsymbol{\mu}_1 - \mathbf{x}_1), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}(\boldsymbol{\Sigma}_{11})^{-1}\boldsymbol{\Sigma}_{12}] \tag{41}$$

In the E-step we need to calculate the $\mathbf{E}[\log p(\mathbf{x}^o, \mathbf{x}^m|\boldsymbol{\theta})|\mathbf{x}_n^o]$, which reduces to the calculation of the sufficient statistics, $\mathbf{E}[\mathbf{x}_n^m|\mathbf{x}_n^o]$ and $\mathbf{E}[\mathbf{x}_n^m\mathbf{x}_n^m|\mathbf{x}_n^o]$, using (40). But we can use the above lemma for this calculation. Suppose we have a datapoint from which the last dimensions are missing. Again, I stress that this needs not be the case and the expectation below must be taken for every datapoint seperately over a different set of missing dimensions.

$$\mathbf{E}\begin{bmatrix} \mathbf{x}^o \\ \mathbf{x}^m \end{bmatrix} = \begin{pmatrix} \mathbf{x}^o \\ \mathbf{E}[\mathbf{x}^m] \end{pmatrix} \tag{42}$$

and

$$\mathbf{E}\begin{bmatrix} \mathbf{x}^o\mathbf{x}^{oT} & \mathbf{x}^o\mathbf{x}^{mT} \\ \mathbf{x}^m\mathbf{x}^{oT} & \mathbf{x}^m\mathbf{x}^{mT} \end{bmatrix} = \begin{pmatrix} \mathbf{x}^o\mathbf{x}^{oT} & \mathbf{x}^o\mathbf{E}[\mathbf{x}^m]^T \\ \mathbf{E}[\mathbf{x}^m]\mathbf{x}^{oT} & \mathbf{E}[\mathbf{x}^m\mathbf{x}^{mT}] \end{pmatrix} \tag{43}$$

where we substitute

$$\mathbf{E}[\mathbf{x}^m] = \boldsymbol{\mu}^m + \boldsymbol{\Sigma}^{mo}(\boldsymbol{\Sigma}^{oo})^{-1}(\boldsymbol{\mu}^o - \mathbf{x}^o) \tag{44}$$

$$\mathbf{E}[\mathbf{x}^m\mathbf{x}^{mT}] = \mathbf{E}[\mathbf{x}^m]\mathbf{E}[\mathbf{x}^m]^T + \boldsymbol{\Sigma}^{mm} - \boldsymbol{\Sigma}^{mo}(\boldsymbol{\Sigma}^{oo})^{-1}\boldsymbol{\Sigma}^{om} \tag{45}$$

Above we assumed for simplicitely that the first dimensions are missing, but in general the expectation appears at the locations of missing dimensions. This then is the solution to the E-step. For every datapoint, split the vector in observed and unobserved dimensions and calculate the expectations of the first and second moment with respect to the posterior which are given above.

The M-step then consists of taking derivatives of $Q = \mathbf{E}[\log p(\mathbf{x}^o, \mathbf{x}^m|\boldsymbol{\theta})|\mathbf{x}^o]$ with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and equating them to zero. Recall that we are maximizing with respect to the parameters in the joint density only; the parameters defining the posterior are from the previous iteration and considered constant. Taking the derivatives we find,

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\mu}} Q &= \sum_{n=1}^{N} \mathbf{E}\left[\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})\right] \Rightarrow \\ \boldsymbol{\mu}_{\mathbf{new}} &= \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}[\mathbf{x}_n] \end{aligned} \tag{46}$$

and

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\Sigma}^{-1}} Q &= \frac{1}{2}\sum_{n=1}^{N} \mathbf{E}\left[\boldsymbol{\Sigma} - (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T\right] \Rightarrow \\ \boldsymbol{\Sigma}_{\mathbf{new}} &= \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}[\mathbf{x}_n\mathbf{x}_n^T] - \boldsymbol{\mu}_{\mathbf{new}}\boldsymbol{\mu}_{\mathbf{new}}^T \end{aligned} \tag{47}$$

To calculate the expectations in these update rule we use (5) and (43) where the expectations are always taken over the missing dimensions only, which may be different from point to point. In the derivation we made use of the fact that,

$$\frac{\partial}{\partial \mathbf{A}}\, \mathbf{a}^T \mathbf{A} \mathbf{a} \quad = \quad = \mathbf{a}\mathbf{a}^T \tag{48}$$

$$\frac{\partial}{\partial \mathbf{A}}\, \log \det[\mathbf{A}] \quad = \quad \mathbf{A}^{-T} \tag{49}$$

The final algorithm thus consists of alternating (44) and (45) with (46) and (47). The result is easy to interpret since mean and covariance are updated by calculating the sample mean and covariance, but with missing dimensions replaced by their expectation, given the data.