ECE 6123
Advanced Signal Processing:
EM, HMMs and BP

Peter Willett

Fall 2017

1 Expectation-Maximization (EM)

1.1 The Algorithm and Why it Works

Suppose we have a problem in which the variables can be divided as follows:

Z: The observation – known, of course.

X: The unknown parameters that are desired.

K: Some hidden random variables.

In fact we could have $\mathbf{X}$ as an unknown random variable (i.e., with a prior) but for the present discussion let's assume it is a parameter. Our goal is to find the maximum-likelihood estimator (MLE) of $\mathbf{X}$ based on $\mathbf{Z}$. Now, if you can easily write $p_{\mathbf{X}}(\mathbf{Z})$ by all means maximize it and skip this whole section. What we are interested in are cases in which $p_{\mathbf{X}}(\mathbf{Z}|\mathbf{K})$ and $p_{\mathbf{X}}(\mathbf{K})$ can both be written, but (with integration in the most general sense, possibly meaning a sum)

$$p_{\mathbf{X}}(\mathbf{Z}) \;=\; \int p_{\mathbf{X}}(\mathbf{Z}|\mathbf{K}) p_{\mathbf{X}}(\mathbf{K}) d\mathbf{K} \tag{1}$$

is irritating and complicated to evaluate, let alone maximize.

The EM approach has two steps. We begin with some sort of guess – and, yes, it can matter a lot – as to $\mathbf{X}^0$, and set $n = 0$.

***E-step*:** Here we form the "Q-function"

$$Q(\mathbf{X}; \mathbf{X}^{(n)}) \;\equiv\; \int \log\left(p_{\mathbf{X}}(\mathbf{Z}, \mathbf{K})\right) p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z}) d\mathbf{K} \tag{2}$$

and the reason this is called the E-step should be obvious: it involves an expectation, albeit of an *un*expected function.

***M-step*:** We maximize (yes, that's why it's called the M-step) and form

$$\mathbf{X}^{(n+1)} \;=\; \arg\max_{\mathbf{X}} \left\{ Q(\mathbf{X}; \mathbf{X}^{(n)}) \right\} \tag{3}$$

We then increment n and return to the E-step.

The reason this works is actually pretty simple. We have

$$\begin{aligned} Q(\mathbf{X}; \mathbf{X}^{(n)}) &= \int \log\left(p_{\mathbf{X}}(\mathbf{Z}, \mathbf{K})\right) p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z}) d\mathbf{K} && (4) \\ &= \int \left(\log\left(p_{\mathbf{X}}(\mathbf{K}|\mathbf{Z})\right) + \log\left(p_{\mathbf{X}}(\mathbf{Z})\right)\right) p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z}) d\mathbf{K} && (5) \\ &= \log\left(p_{\mathbf{X}}(\mathbf{Z})\right) + \int \log\left(p_{\mathbf{X}}(\mathbf{K}|\mathbf{Z})\right) p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z}) d\mathbf{K} && (6) \end{aligned}$$

Now consider any two pmf's or pdf's q_1 & q_2. Noting that

$$\ln(x) \;\leq\; x - 1 \tag{7}$$

with equality if and only if $x = 1$ (draw the graph!), we have

$$\begin{aligned} \int q_1 \log(q_2) - \int q_1 \log(q_1) &= \int q_1 \log\left(\frac{q_2}{q_1}\right) && (8) \\ &\leq \log(2) \int q_1 \left(\frac{q_2}{q_1} - 1\right) && (9) \\ &\leq \log(2) \int q_1 \left(\frac{q_2}{q_1} - 1\right) && (10) \\ &= 0 && (11) \end{aligned}$$

This means that

$$\int q_1 \log(q_2) \;\leq\; \int q_1 \log(q_1) \tag{12}$$

with equality if and only if $q_1 = q_2$. Returning to (6) we see from (12) that if

$$Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) \;>\; Q(\mathbf{X}^{(n)}; \mathbf{X}^{(n)}) \tag{13}$$

then

$$\log\left(p_{\mathbf{X}^{(n+1)}}(\mathbf{Z})\right) \;>\; \log\left(p_{\mathbf{X}^{(n)}}(\mathbf{Z})\right) \tag{14}$$

since the second term must have decreased. This means that the change from $\mathbf{X}^{(n)}$ to $\mathbf{X}^{(n+1)}$ must have increased the likelihood that we are aiming to maximize. Note that although the M-step by tradition requires a maximization, the M could also stand for *majorization*: all that is really required

for (14) is an increase in Q. There is no real need for a *maximization* if that turns out to be difficult or expensive. Note also that (14) tells us clearly that this is *hill-climbing* approach: there is no guarantee that a global maximum likelihood be found.

As a final note, many authors refer to $\mathbf{K} \bigcup \mathbf{Z}$ as the *complete data* and $\mathbf{Z}$ as the *in*complete data. I don't care for the nomenclature; but there it is.

1.2 The Gaussian Mixture Example

Sometimes you need to manufacture the $\mathbf{K}$'s yourself. Consider that you have N independent z_i's from the same Gaussian mixture pdf

$$p(z) = \sum_{m=1}^{M} \frac{p_m}{\sqrt{|2\pi\mathbf{R}|}} e^{-\frac{1}{2}(z-\mu_m)^T\mathbf{R}^{-1}(z-\mu_m)} \tag{15}$$

where the mixture priors $\{p_m\}$ and the means $\{\mu_m\}$ are both unknown. You could insert[1] $\mathbf{K} = \{k_i\}$ such that $k_i \in \{1, M\}$ and

$$Pr(k_i = m) = p_m \tag{16}$$

$$\{k_i\} \sim \text{independent and identically distributed} \tag{17}$$

$$p(z_i|k_i) = \frac{1}{\sqrt{|2\pi\mathbf{R}|}} e^{-\frac{1}{2}(z_i-\mu_{k_i})^T\mathbf{R}^{-1}(z_i-\mu_{k_i})} \tag{18}$$

In the EM formalism the first thing we need is $p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z})$. This is relatively easy:

$$p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z}) = \prod_{i=1}^{N} p(k_i|\mathbf{Z}) \tag{19}$$

$$= \prod_{i=1}^{N} p_{\mathbf{X}^{(n)}}(k_i|z_i) \tag{20}$$

$$\equiv \prod_{i=1}^{N} w_i(k_i) \tag{21}$$

$$w_i(m) = \frac{p_m^{(n)} p_{\mathbf{X}^{(n)}}(z_i|k_i = m)}{\sum_{l=1}^{M} p_l^{(n)} p_{\mathbf{X}^{(n)}}(z_i|k_i = l)} \tag{22}$$

$$= \frac{p_m^{(n)} \frac{1}{\sqrt{|2\pi\mathbf{R}|}} e^{-\frac{1}{2}(z_i-\mu_m^{(n)})^T\mathbf{R}^{-1}(z_i-\mu_m^{(n)})}}{\sum_{l=1}^{M} p_l^{(n)} \sqrt{|2\pi\mathbf{R}|} e^{-\frac{1}{2}(z_i-\mu_l^{(n)})^T\mathbf{R}^{-1}(z_i-\mu_l^{(n)})}} \tag{23}$$

[1] Actually this is the way you would generate such random variables.

The nomenclature involving w's is fairly common for the posterior probabilities. Now we have

$$
\begin{aligned}
Q(\mathbf{X};\mathbf{X}^{(n)}) &= \int \log\left(p_{\mathbf{X}}(\mathbf{Z},\mathbf{K})\right) p_{\mathbf{X}^{(n)}}(\mathbf{K}) d\mathbf{K} && (24)\\
&= \sum_{\mathbf{K}} \left(\sum_{i=1}^{N} \left(\left(\log(p_{k_i}) \; - \; \frac{1}{2}\log\left(|2\pi\mathbf{R}|\right) \right.\right.\right. \\
&\qquad \left.\left.\left. - \frac{1}{2}\left(z_i - \mu_{k_i}\right)^T \mathbf{R}^{-1}\left(z_i - \mu_{k_i}\right) \right) \prod_{i=1}^{M} w_i(k_i) \right) \right) && (25)\\
&= \sum_{m=1}^{M} \sum_{i=1}^{N} \left(\left(\log(p_m) \; - \; \frac{1}{2}\log\left(|2\pi\mathbf{R}|\right) \right.\right. \\
&\qquad \left.\left. - \frac{1}{2}\left(z_i - \mu_m\right)^T \mathbf{R}^{-1}\left(z_i - \mu_m\right) \right) w_i(m) \right) && (26)
\end{aligned}
$$

Maximizing (26) over p_m subject to the constraint that these prior probabilities add to unity yields

$$
\frac{\partial}{\partial p_m} \left(\sum_{i=1}^{N} \log(p_m) w_i(m) - \lambda p_m \right) \; = \; 0 \qquad (27)
$$

or

$$
p_m \; = \; \frac{\sum_{i=1}^{N} w_i(m)}{\sum_{i=1}^{N}\sum_{l=1}^{M} w_i(l)} \qquad (28)
$$

where of course the denominator is most easily found by normalization. As for the μ_m's we take the gradient

$$
\begin{aligned}
\nabla \left(\sum_{i=1}^{N} \left(\frac{1}{2}\left(z_i - \mu_m\right)^T \mathbf{R}^{-1}\left(z_i - \mu_m\right) \right) w_i(m) \right) &= 0 && (29)\\
\sum_{i=1}^{N} \left(\mathbf{R}^{-1}\left(z_i - \mu_m\right) \right) w_i(m) &= 0 && (30)
\end{aligned}
$$

or

$$
\mu_m \; = \; \frac{\sum_{i=1}^{N} w_i(m) z_i}{\sum_{i=1}^{N} w_i(m)} \qquad (31)
$$

This is a nice easy recursion: Start with a guess about the μ_m's and p_m's. Then calculate the w's according to (23). Then update the parameters according to (28) & (31); and go back to getting new w's. Stop when you get tired of it – or more likely when the estimates stop moving. Note that

this is a "soft" version of the celebrated k-means algorithm for clustering. It is also interesting to note that it *is* possible to estimate the covariance as well, and also to allow the covariances to be different across the various modes.

2 The Hidden Markov Model

2.1 Modeling for EM

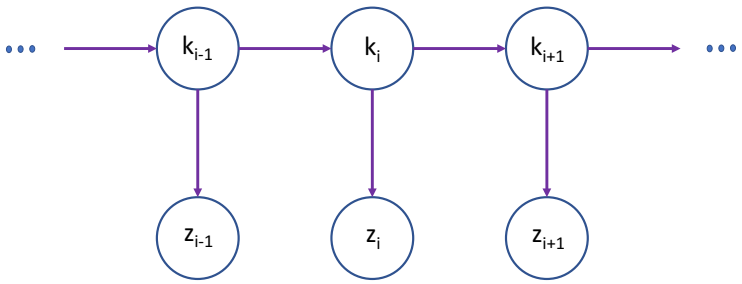
A Markov model has

$$p(\mathbf{Z}) \;=\; p(z_1)\prod_{i=2}^{N} p(z_i|z_{i-1}) \tag{32}$$

whereas a *hidden* Markov model (HMM) does not give direct access to the Markov process:

$$p(\mathbf{Z},\mathbf{K}) \;=\; \left(p(k_1)\prod_{i=2}^{N} p(k_i|k_{i-1})\right)\left(\prod_{i=1}^{N} p(z_i|k_i)\right) \tag{33}$$

A fragment of an HMM is pictured below.



I have gone out of my way to use non-standard HMM nomenclature (especially the k_i's) to emphasize the relationship to the EM algorithmic tools we have developed. We will define $\mathbf{X} = \{\mathbf{A}, \mathbf{B}, \mathbf{p}\}$ in which

$$A(m|n) \;=\; Pr(k_i = m|k_{i-1} = n) \tag{34}$$

$$B(l|n) \;=\; Pr(z_i = l|k_i = n) \tag{35}$$

$$p(m) \;=\; Pr(k_1 = m) \tag{36}$$

These are what we seek: the $M \times M$ matrix $\mathbf{A}$ and the $M \times L$ matrix $\mathbf{B}$, meaning that there are M "hidden" states and L kinds[2] of outputs. And of

[2]There is a perfectly good formulation of the HMM that allows for continuous-valued outputs; for simplicity of notation we will assume discreteness.

course this is a prime example of a problem that is absolutely panting for EM to come solve it.

2.2 The Forward-Backward Algorithm

In this section we seek an expression for the posterior probabilities of the state sequence. Define

$$\alpha(\mathbf{Z}_1^{i+1}, m) \equiv p(\mathbf{Z}_1^i, k_i = m) \tag{37}$$

We can write

$$\begin{aligned}
\alpha(\mathbf{Z}_1^{i+1}, m) &= \sum_{n=1}^{M} p(z_{i+1}, k_{i+1} = m, k_i = n, \mathbf{Z}_1^i) && (38)\\
&= \sum_{n=1}^{M} p(z_{i+1}, k_{i+1} = m | k_i = n, \mathbf{Z}_1^i) \\
&\qquad \times Pr(k_i = n, \mathbf{Z}_1^i) && (39)\\
&= \sum_{n=1}^{M} p(z_{i+1} | k_{i+1} = m, \mathbf{Z}_1^i, k_i = n) \\
&\qquad \times Pr(k_{i+1} = m | \mathbf{Z}_1^i, k_i = n) Pr(k_i = n, \mathbf{Z}_1^i) && (40)\\
&= \sum_{n=1}^{M} p(z_{i+1} | k_{i+1} = m) \\
&\qquad \times Pr(k_{i+1} = m | k_i = n) Pr(k_i = n, \mathbf{Z}_1^i) && (41)\\
&= \sum_{n=1}^{M} B(z_{i+1}|m) A(m|n) \alpha(\mathbf{Z}_1^i, n) && (42)
\end{aligned}$$

which is a nice recursive formula for $\alpha(\cdot|\cdot)$ when initialized with

$$\begin{aligned}
\alpha(\mathbf{Z}_1^1, m) &= Pr(z_1, k_i = m) && (43)\\
&= \frac{B(z_1|m)p(m)}{\sum_{l=1}^{M} B(z_1|l)p(l)} && (44)
\end{aligned}$$

This is the *forward* part of the forward-backward algorithm. Notice that if all we wanted was what amounted to a *filter* – that is, we want the posterior probability $p(k_i = m|\mathbf{Z}_1^i)$ – then all we need to do is one single "forward" pass and normalize the sum over m of the $\alpha(\mathbf{Z}_1^N, m)$'s to be unity. Similarly, If what we wanted was just $p(\mathbf{Z}_1^i)$ – that would give us the *likelihood* that we might use to test if the model is correct – then all we need do is sum

$\alpha(\mathbf{Z}_1^N, m)$ over m to marginalize that out. That is, in either of these cases we would be done. However, in order to estimate the model we require detailed information about $p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z})$; for that we need the *backward* pass. However, similar to the Kalman Smoother[3] all that is needed is one forward and one backward sweep [4] per iteration.

Similarly, define

$$\beta(\mathbf{Z}_{i+1}^N, m) \equiv p(\mathbf{Z}_{i+1}^N | k_i = m) \tag{45}$$

We can write

$$\begin{aligned}
\beta(\mathbf{Z}_i^N, m) &= p(\mathbf{Z}_i^N | k_{i-1} = m) && (46)\\
&= \sum_{n=1}^{M} p(\mathbf{Z}_{i+1}^N, z_i, k_i = n | k_{i-1} = m) && (47)\\
&= \sum_{n=1}^{M} p(\mathbf{Z}_{i+1}^N | z_i, k_i = n, k_{i-1} = m) \\
&\qquad \times p(z_i | k_i = n, k_{i-1} = m) \\
&\qquad \times Pr(k_i = n | k_{i-1} = m) && (48)\\
&= \sum_{n=1}^{M} Pr(\mathbf{Z}_{i+1}^N | k_i = n) p(z_i | k_i = n) \\
&\qquad \times Pr(k_i = n | k_{i-1} = m) && (49)\\
&= \sum_{n=1}^{M} \beta(\mathbf{Z}_{i+1}^N, n) B(z_i | n) A(n | m) && (50)
\end{aligned}$$

which is a nice recursive formula for $\beta(\cdot|\cdot)$ when initialized with

$$\beta(\mathbf{Z}_{N+1}^N, m) = \frac{1}{N} \tag{51}$$

This is the *backward* part of the forward-backward algorithm. It is typical to scale both forward and backward directions, since underflow often results.

One key fact that is especially interesting is that these quantities give us

[3] Actually there is a nice alternative derivation of the Kalman Smoother that is exactly Baum-Welch.

[4] In the EM (or Baum-Welch) algorithm the estimates for $\mathbf{X}^{(n)} = \{\mathbf{A}, \mathbf{B}, \mathbf{p}\}$ change each iteration. Naturally, therefore, each iteration requires a new forward and backward sweep to determine the requisite $p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z})$.

the marginal probabilities for state occupancies. That is,

$$\begin{aligned} Pr(k_i = m|\mathbf{Z}_1^N) &\propto p(k_i = m, \mathbf{Z}_1^N) && (52)\\ &= p(\mathbf{Z}_1^i, \mathbf{Z}_{i+1}^N, k_i = m) && (53)\\ &= p(\mathbf{Z}_{i+1}^N|\mathbf{Z}_1^i, k_i = m)p(\mathbf{Z}_1^i|k_i = m) && (54)\\ &= p(\mathbf{Z}_{i+1}^N|k_i = m)p(\mathbf{Z}_1^i|k_i = m) && (55)\\ &= \beta(\mathbf{Z}_{i+1}^N|k_i = m)\alpha(\mathbf{Z}_1^i|k_i = m) && (56) \end{aligned}$$

and normalizing (56) to sum to unity is obvious. We also see from (52) that we can write

$$\begin{aligned} p(\mathbf{Z}_1^N) &= \sum_{m=1}^{M} p(k_i = m, \mathbf{Z}_1^N) && (57)\\ &= \sum_{m=1}^{M} \beta(\mathbf{Z}_{i+1}^N|k_i = m)\alpha(\mathbf{Z}_1^i|k_i = m) && (58) \end{aligned}$$

which, interestingly, does not depend[5] on i. This is the likelihood of the whole sequence given the model, and can be useful for model testing.

Now we'll need

$$\begin{aligned} w_1(m) &\equiv Pr_{\mathbf{X}^{(n)}}(k_i = m|\mathbf{Z}) && (59)\\ &= \beta(\mathbf{Z}_2^N|k_i = m)\alpha(\mathbf{Z}_1^1|k_i = m) && (60) \end{aligned}$$

from (56). We'll also need

$$\begin{aligned} p(k_{i-1} = n, k_i = m|\mathbf{Z}_1^N) &\propto p(k_{i-1} = n, k_i = m, \mathbf{Z}_1^N) && (61)\\ &= p(\mathbf{Z}_1^{i-1}, z_i, \mathbf{Z}_{i+1}^N, k_{i-1} = n, k_i = m) && (62)\\ &= p(\mathbf{Z}_{i+1}^N|\mathbf{Z}_1^{i-1}, z_i, k_{i-1} = n, k_i = m)\\ &\qquad \times p(z_i|\mathbf{Z}_1^{i-1}, k_{i-1} = n, k_i = m)\\ &\qquad \times p(k_i = m|\mathbf{Z}_1^{i-1}, k_{i-1} = n)\\ &\qquad \times p(\mathbf{Z}_1^{i-1}, k_{i-1} = n) && (63)\\ &= p(\mathbf{Z}_{i+1}^N|k_i = m)p(z_i|, k_i = m)\\ &\qquad \times p(k_i = m|k_{i-1} = n)\\ &\qquad \times p(\mathbf{Z}_1^{i-1}, k_{i-1} = n) && (64)\\ &= \beta(\mathbf{Z}_{i+1}^N|m)B(z_i|m)A(m|n)\alpha(\mathbf{Z}_1^{i-1}, n) && (65) \end{aligned}$$

[5] In fact the solution is the same for every i.

Finally, we'll need

$$\begin{aligned} p(k_i = m, z_i = n|\mathbf{Z}_1^N) &\propto p(k_i = m, \mathbf{Z}_1^N)\mathcal{I}(z_i = n) && (66)\\ &= p(\mathbf{Z}_{i+1}^N|k_i = m)p(k_i = m, \mathbf{Z}_1^i) \\ &\quad \times \mathcal{I}(z_i = n) && (67)\\ &= \beta(\mathbf{Z}_{i+1}^N|m)\alpha(\mathbf{Z}_1^i, m)\mathcal{I}(z_i = n) && (68) \end{aligned}$$

And now we are ready to apply EM.

2.3 The Baum-Welch Algorithm

The Baum-Welch "re-estimation" algorithm was designed to estimate the parameters of an HMM based on one or preferably many sets of observations. The notation below assumes a single time series observation, but the extension to multiple observation sequences is obvious – and indeed estimation of the initial probability will be pretty poor if there is only one times-series observation, since there is only one exemplar. Note that there is no stipulation that the underlying Markov model be in "steady-state" – Baum-Welch works fine for non-stationary HMMs. The Baum-Welch procedure was discovered independently of EM; but it was later noted that it was exactly the EM algorithm applied to an HMM.

We begin by inserting (33) to (2). We have

$$\begin{aligned} Q(\mathbf{X}; \mathbf{X}^{(n)}) &= \int \log\left(p_{\mathbf{X}}(\mathbf{Z}, \mathbf{K})\right) p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z}) d\mathbf{K} && (69)\\ &= \sum_{\mathbf{K}} \left(\left(\log(p(k_1)) + \sum_{i=2}^{N} \log\left(A(k_i|k_{i-1})\right) \right. \right. \\ &\quad \left. \left. + \sum_{i=1}^{N} \log\left(B(z_i|k_i)\right) \right) p_{\mathbf{X}^{(n)}}(\mathbf{K}|\mathbf{Z}) \right) && (70) \end{aligned}$$

Maximizing the Q-function over all these is quite simple; the only "subtlety" (and it isn't very subtle, really) is that we have to apply the Lagrange constraint that all probabilities sum to unity. We get

$$p(m) = w_1(m) \tag{71}$$

using (60). We also have

$$A(m|n) = \kappa_{A(\cdot|n)} \sum_{i=2}^{N} p(k_{i-1} = n, k_i = m|\mathbf{Z}_1^N) \tag{72}$$

where the probabilities are from (65) and $\kappa_{A(\cdot|n)}$ is such that

$$\sum_{m=1}^{M} A(m|n) = 1 \tag{73}$$

is normalized. Finally, we get

$$B(l|m) \;=\; \kappa_{B(\cdot|m)} \sum_{i=1}^{N} p(k_i = m, z_i = n|\mathbf{Z}_1^N) \tag{74}$$

where the probabilities are from (68) and $\kappa_{B(\cdot|m)}$ is such that

$$\sum_{l=1}^{L} B(l|m) = 1 \tag{75}$$

is normalized. Baum-Welch says keep doing this iteration until convergence.