12 The EM Algorithm

As we saw before, many estimation problems require maximization of the probability distribution with respect to an unknown parameter, for example when computing ML estimates of the parameters or MAP estimates of the hidden random variables. For many interesting problems, differentiating the probability distribution with respect to the parameter of interest and setting the derivative to zero results in a nonlinear equation that does not have a closed-form solution. In such cases, we have to resort to numerical optimization.

Example 1. Let w be a (hidden) Bernoulli random variable with parameter δ, $\mathbb{P}(\mathsf{w}=1)=\delta$. Let y be a noisy observation of w obtained through a binary symmetric measurement mechanism:

$$p_{\mathsf{y}|\mathsf{w}}(y|w)=\begin{cases}\epsilon, & y\neq w\\ 1-\epsilon, & y=w\end{cases}. \tag{1}$$

Let $\mathbf{w}=[w_1,\ldots,w_n]^{\mathrm{T}}$ be a binary string of N i.i.d. samples of w. We do not get to observe $\mathbf{w}$ directly, but instead we get access to the corresponding string of measurements $\mathbf{y}=[y_1,\ldots,y_n]^{\mathrm{T}}$. Our goal is to estimate the system parameters δ and ϵ.

To compute the ML estimates of the parameters, we form the likelihood model:

$$\begin{aligned}p_{\mathbf{y}}(\mathbf{y};\delta,\epsilon)&=\prod_{i=1}^{N}p_{\mathsf{y}_i}(y_i;\delta,\epsilon)\\&=\prod_{i=1}^{N}\left[p_{\mathsf{y}_i|\mathsf{w}_i}(y_i|0;\delta,\epsilon)p_{\mathsf{w}_i}(0;\delta,\epsilon)+p_{\mathsf{y}_i|\mathsf{w}_i}(y_i|1;\delta,\epsilon)p_{\mathsf{w}_i}(1;\delta,\epsilon)\right]\\&=\prod_{i=1}^{N}\left[\epsilon^{y_i}(1-\epsilon)^{1-y_i}(1-\delta)+\epsilon^{1-y_i}(1-\epsilon)^{y_i}\delta\right].\end{aligned} \tag{2}$$

We can attempt to find δ and ϵ that maximize the likelihood directly by setting its partial derivatives with respect to δ and ϵ to zero. This leads to a pair of polynomial equations whose roots represent the extrema points of the likelihood function. In general, such equations do not have a closed-form solution.

While various generic hill-climbing methods exist for maximizing functions, we should be able to take advantage of the special properties of probability distributions to develop specialized, more effective algorithms for this problem. The EM algorithm is an example of such a method. It exploits the structure in the probability distribution to efficiently search for its local maximum. The EM algorithm is so named because it iterates between an Expectation (E) step, and a Maximization (M) step.

12.1 Problem Setup And Motivation

The EM algorithm assumes that *complete data* $\mathbf{z}$ is generated by the probability distribution $p_{\mathbf{z}}(\cdot;\mathbf{x})$ but is not is not fully observable. Instead, we get access to *observed data* $\mathbf{y}$ that is related to the complete data $\mathbf{z}$. Our goal is to compute the ML estimate of the parameter $\mathbf{x}$ from the observed data $\mathbf{y}$. Here we assume that $\mathbf{y}$ is a deterministic function of $\mathbf{z}$, i.e., $\mathbf{y} = \mathbf{g}(\mathbf{z})$. This setup naturally arises in many problems, such as the one presented in Example 1, where the complete data $\mathbf{z}$ consists of the hidden binary string $\mathbf{w}$ and the observed binary string $\mathbf{y}$. The formulation can also be extended to handle probabilistic relationships between the complete and observed data.

It is easy to see that

$$p_{\mathbf{z}}(\mathbf{z};\mathbf{x}) = \sum_{\mathbf{y}} p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x}) p_{\mathbf{y}}(\mathbf{y};\mathbf{x}) = p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|g(\mathbf{z});\mathbf{x}) p_{\mathbf{y}}(g(\mathbf{z});\mathbf{x}). \tag{3}$$

Therefore, for any value $\mathbf{z}$ that satisfies $\mathbf{y} = g(\mathbf{z})$, we obtain

$$\log p_{\mathbf{y}}(\mathbf{y};\mathbf{x}) = \log p_{\mathbf{z}}(\mathbf{z};\mathbf{x}) - \log p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x}). \tag{4}$$

Since the left-hand side of (4) does not depend on $\mathbf{z}$, computing expectation of both sides of this equation with respect to $p_{\mathbf{z}|\mathbf{y}}(\cdot|\mathbf{y};\mathbf{x}')$ for the observed data $\mathbf{y}$ and any value of the parameter $\mathbf{x}'$ yields

$$\log p_{\mathbf{y}}(\mathbf{y};\mathbf{x}) = \mathbb{E}\left[\log p_{\mathbf{z}}(\mathbf{z};\mathbf{x})|\mathbf{y}=\mathbf{y};\mathbf{x}'\right] - \mathbb{E}\left[\log p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x})|\mathbf{y}=\mathbf{y};\mathbf{x}'\right]. \tag{5}$$

Using the notation

$$U(\mathbf{x};\mathbf{x}') = \mathbb{E}\left[\log p_{\mathbf{z}}(\mathbf{z};\mathbf{x})|\mathbf{y}=\mathbf{y};\mathbf{x}'\right] \tag{6}$$

and

$$V(\mathbf{x};\mathbf{x}') = -\mathbb{E}\left[\log p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x})|\mathbf{y}=\mathbf{y};\mathbf{x}'\right], \tag{7}$$

we arrive at the following relationship:

$$\log p_{\mathbf{y}}(\mathbf{y};\mathbf{x}) = U(\mathbf{x};\mathbf{x}') + V(\mathbf{x};\mathbf{x}') \quad \text{for any } \mathbf{x}'. \tag{8}$$

The Gibbs' inequality immediately implies that for any $\mathbf{x}'$

$$\begin{aligned} V(\mathbf{x};\mathbf{x}') &= -\mathbb{E}\left[\log p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x})|\mathbf{y}=\mathbf{y};\mathbf{x}'\right] \\ &\geq -\mathbb{E}\left[\log p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x}')|\mathbf{y}=\mathbf{y};\mathbf{x}'\right] = V(\mathbf{x}';\mathbf{x}') \end{aligned} \tag{9}$$

Since the Gibbs' inequality is satisfied with equality if and only if the two distributions are identical, choosing $\mathbf{x} \neq \mathbf{x}'$ such that $U(\mathbf{x},\mathbf{x}') \geq U(\mathbf{x}',\mathbf{x}')$ guarantees that

$$\log p_{\mathbf{y}}(\mathbf{y};\mathbf{x}) = U(\mathbf{x},\mathbf{x}') + V(\mathbf{x},\mathbf{x}') > U(\mathbf{x}',\mathbf{x}') + V(\mathbf{x}',\mathbf{x}') = \log p_{\mathbf{y}}(\mathbf{y};\mathbf{x}'). \tag{10}$$

This relationship is at the core of the EM algorithm that iteratively refines parameter estimates by maximizing $U(\mathbf{x},\mathbf{x}')$ with respect to $\mathbf{x}$ at each step.

12.2 The EM Algorithm

Given the observed data $\mathbf{y}$, the EM algorithm generates a set of successive parameter estimates $\hat{\mathbf{x}}^{(0)}, \ldots, \hat{\mathbf{x}}^{(n)} \ldots$ as follows:

- Initialize (guess) a parameter estimate $\hat{\mathbf{x}}^{(0)}$.
- Repeat until convergence

  E-step Given the previous parameter estimate $\hat{\mathbf{x}}^{(n)}$, form

  $$U(\mathbf{x}; \hat{\mathbf{x}}^{(n)}) = \mathbb{E}_{p_{\mathbf{z}|\mathbf{y}}(\cdot|\mathbf{y};\hat{\mathbf{x}}^{(n)})} \left[\log p_{\mathbf{z}}(\mathbf{z}; \mathbf{x}) | \mathbf{y} = \mathbf{y}; \hat{\mathbf{x}}^{(n)} \right]. \tag{11a}$$

  M-step Find the next parameter estimate $\hat{\mathbf{x}}^{(n+1)}$ by maximizing $U(\cdot; \hat{\mathbf{x}}^{(n)})$:

  $$\hat{\mathbf{x}}^{(n+1)} = \arg\max_{\mathbf{x}} U(\mathbf{x}; \hat{\mathbf{x}}^{(n)}). \tag{11b}$$

The original ML formulation seeks to maximize the likelihood of the observed data, which is obtained through *marginalization* of the complete data distribution. The EM algorithm maximizes a cost function that is equal to the *expectation* of the logarithm of the complete data, i.e., it replaces a marginal with a geometric mean. The algorithm offers a computational advantage if finding the maximum of the function $U(\cdot; \hat{\mathbf{x}}^{(n)})$ in each step is simpler than maximizing the likelihood function $p_{\mathbf{y}}(\mathbf{y}; \cdot)$.

Our earlier analysis implies that any iteration of the algorithm that changes the value of the parameter is guaranteed to strictly increase the likelihood value. In practice, we can relax the M-step of the algorithm to seek $\hat{\mathbf{x}}^{(n+1)}$ that increases the value of U, rather than maximizes it. As long as $\hat{\mathbf{x}}^{(n+1)} \neq \hat{\mathbf{x}}^{(n)}$ and $U(\hat{\mathbf{x}}^{(n+1)}; \hat{\mathbf{x}}^{(n)}) \geq U(\hat{\mathbf{x}}^{(n)}; \hat{\mathbf{x}}^{(n)})$, the likelihood value increases in each iteration. This variant of the algorithm is called Generalized EM (GEM) and is often used if the maximum of the function $U(\cdot; \hat{\mathbf{x}}^{(n)})$ cannot be computed in a closed form.

We can also prove that under mild conditions, the algorithm converges to a stationary point of the likelihood function, i.e., a point at which its derivative is zero. Consequently, if the likelihood function has a single maximum, the algorithm is guaranteed to converge to it. Rather than prove the convergence for the most general case, we will provide the proof for the exponential families in the next section.

The convergence rate and computational complexity of the algorithm depend crucially on the choice of the complete data $\mathbf{z}$. The physical setup of the problem will often suggest a natural choice of the complete data, but it is still a design choice that requires some thought for each new problem.

The algorithm can also be adapted for MAP estimation by replacing the likelihood function with the posterior probability distribution:

$$\log p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = \log \frac{p_{\mathbf{z},\mathbf{x},\mathbf{y}}(\mathbf{z}, \mathbf{x}, \mathbf{y}) p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})}{p_{\mathbf{z},\mathbf{x},\mathbf{y}}(\mathbf{z}, \mathbf{x}, \mathbf{y}) p_{\mathbf{y}}(\mathbf{y})} = \log p_{\mathbf{z},\mathbf{x}|\mathbf{y}}(\mathbf{z}, \mathbf{x}|\mathbf{y}) - \log p_{\mathbf{z}|\mathbf{x},\mathbf{y}}(\mathbf{z}|\mathbf{x}, \mathbf{y}) \tag{12}$$

and taking expectation of both sides with respect to the distribution $p_{\mathbf{z}|\mathbf{x},\mathbf{y}}(\cdot|\mathbf{x}', \mathbf{y})$.

12.3 The EM Algorithm for Exponential Families

Here we analyze the algorithm behavior for the case when the complete data is generated by a member of a linear exponential family. The analysis can also be extended to general exponential families.

We start with the general form of the likelihood

$$p_{\mathbf{z}}(\mathbf{z};\mathbf{x}) = \exp\left\{\mathbf{x}^{\mathrm{T}}\mathbf{t}(\mathbf{z}) - \alpha(\mathbf{x}) + \beta(\mathbf{z})\right\}, \tag{13}$$

where $\mathbf{t}(\cdot)$ is the (vector) natural statistic and $\beta(\cdot)$ is the log base distribution. Substituting into (11a), we obtain

$$\begin{aligned} U(\mathbf{x},\hat{\mathbf{x}}^{(n)}) &= \mathbb{E}\left[\log p_{\mathbf{z}}(\mathbf{z};\mathbf{x})|\mathbf{y}=\mathbf{y};\hat{\mathbf{x}}^{(n)}\right] \\ &= \mathbf{x}^{\mathrm{T}}\,\mathbb{E}\left[\mathbf{t}(\mathbf{z})|\mathbf{y}=\mathbf{y};\hat{\mathbf{x}}^{(n)}\right] - \alpha(\mathbf{x}) + \mathbb{E}\left[\beta(\mathbf{z})|\mathbf{y}=\mathbf{y};\hat{\mathbf{x}}^{(n)}\right]. \end{aligned} \tag{14}$$

To maximize $U(\mathbf{x},\hat{\mathbf{x}}^{(n)})$, we set its partial derivatives to zero,

$$\frac{\partial}{\partial x_i}U(\mathbf{x},\hat{\mathbf{x}}^{(n)}) = \mathbb{E}\left[t_i(\mathbf{z})|\mathbf{y}=\mathbf{y};\hat{\mathbf{x}}^{(n)}\right] - \frac{\partial\alpha(\mathbf{x})}{\partial x_i} = 0. \tag{15}$$

This yields a set of implicit equations in $\mathbf{x}$ that define the update.

For the linear exponential families,

$$\frac{\partial\alpha(\mathbf{x})}{\partial x_i} = \mathbb{E}\left[t_i(\mathbf{z});\mathbf{x}\right]. \tag{16}$$

In this case, the M-step of the algorithm seeks the next parameter estimate $\hat{\mathbf{x}}^{(n+1)}$ that produces the first order moments of the natural statistics of the complete data that are equal to the corresponding conditional moments for the previous parameter estimate $\hat{\mathbf{x}}^{(n)}$:

$$\mathbb{E}\left[t_i(\mathbf{z});\hat{\mathbf{x}}^{(n+1)}\right] = \mathbb{E}\left[t_i(\mathbf{z})|\mathbf{y}=\mathbf{y};\hat{\mathbf{x}}^{(n)}\right] \quad \text{for all } i. \tag{17}$$

At a fixed point $\hat{\mathbf{x}}^*$ of the algorithm,

$$\mathbb{E}\left[t_i(\mathbf{z});\hat{\mathbf{x}}^*\right] = \mathbb{E}\left[t_i(\mathbf{z})|\mathbf{y}=\mathbf{y};\hat{\mathbf{x}}^*\right] \quad \text{for all } i. \tag{18}$$

Since $\mathbf{y}=\mathbf{g}(\mathbf{z})$, the conditional distribution $p_{\mathbf{z}|\mathbf{y}}(\cdot|\mathbf{y};\mathbf{x})$ is also a member of an exponential family, characterized by the same natural statistics but a different log partition function. Specifically, for $\mathbf{y}=\mathbf{g}(\mathbf{z})$,

$$\begin{aligned} p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x}) &= \frac{p_{\mathbf{y}|\mathbf{z}}(\mathbf{y}|\mathbf{z};\mathbf{x})\,p_{\mathbf{z}}(\mathbf{z};\mathbf{x})}{p_{\mathbf{y}}(\mathbf{y};\mathbf{x})} \\ &= \frac{p_{\mathbf{z}}(\mathbf{z};\mathbf{x})}{p_{\mathbf{y}}(\mathbf{y};\mathbf{x})} \\ &= \exp\left\{\mathbf{x}^{\mathrm{T}}\mathbf{t}(\mathbf{z}) - (\alpha(\mathbf{x}) + \ln p_{\mathbf{y}}(\mathbf{y};\mathbf{x})) + \beta(\mathbf{z})\right\}. \end{aligned} \tag{19}$$

We note that since $\mathbf{y}$ is the observed data, $\ln p_{\mathbf{y}}(\mathbf{y};\mathbf{x})$ in (19) is a constant and can be grouped with the log partition function $\alpha(\mathbf{x})$.

Since for linear exponential families the first moments of the natural statistics are equal to the partial derivatives of the log partition function, we obtain

$$\begin{aligned}\frac{\partial}{\partial x_i}\left[\ln p_{\mathbf{y}}(\mathbf{y};\mathbf{x})\right]_{\hat{\mathbf{x}}^*} &= \frac{\partial}{\partial x_i}\left[\log p_{\mathbf{z}}(\mathbf{z};\mathbf{x}) - \log p_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y};\mathbf{x})\right]_{\hat{\mathbf{x}}^*} \\ &= \frac{\partial}{\partial x_i}\left[\alpha(\mathbf{x}) + \ln p_{\mathbf{y}}(\mathbf{y};\mathbf{x})\right]_{\hat{\mathbf{x}}^*} - \frac{\partial}{\partial x_i}\left[\alpha(\mathbf{x})\right]_{\hat{\mathbf{x}}^*} \\ &= \mathbb{E}\left[t_i(\mathbf{z})|\mathbf{y}=\mathbf{y};\hat{\mathbf{x}}^*\right] - \mathbb{E}\left[t_i(\mathbf{z});\hat{\mathbf{x}}^*\right] = 0 \quad \text{for all } i, \end{aligned} \tag{20}$$

i.e., at the fixed point of the algorithm, the gradient of the likelihood function is equal to zero. In other words, the algorithm converges to the stationary point of the likelihood function.

12.4 Example EM Algorithm

Now we derive the EM algorithm for the problem in Example 1. First, we form the full data $\mathbf{z} = (\mathbf{w}, \mathbf{y})$ and construct the probability distribution

$$\begin{aligned} p_{\mathbf{z}}(\mathbf{z}=(\mathbf{w},\mathbf{y});\delta,\epsilon) &= p_{\mathbf{w},\mathbf{y}}(\mathbf{w},\mathbf{y};\delta,\epsilon) \\ &= \prod_{i=1}^{N} p_{w_i,y_i}(w_i,y_i;\delta,\epsilon) \\ &= \prod_{i=1}^{N} p_{w_i}(w_i;\delta)p_{y_i|w_i}(y_i|w_i;\epsilon). \end{aligned} \tag{21}$$

We define $\Delta(w,y)$ to be a function that is equal to 1 if the bits y and w are equal and 0 otherwise. Using this helper function, we obtain,

$$p_{\mathbf{z}}((\mathbf{w},\mathbf{y});\delta,\epsilon) = \prod_{i=1}^{N} \delta^{w_i}(1-\delta)^{1-w_i}\epsilon^{1-\Delta(y_i,w_i)}(1-\epsilon)^{\Delta(y_i,w_i)} \tag{22}$$

and

$$\begin{aligned}\log p_{\mathbf{z}}((\mathbf{w},\mathbf{y});\delta,\epsilon) = \log\delta\left(\sum_{i=1}^{N} w_i\right) + \log(1-\delta)\left(N - \sum_{i=1}^{N} w_i\right) \\ + \log\epsilon\left(N - \sum_{i=1}^{N}\Delta(y_i,w_i)\right) \\ + \log(1-\epsilon)\left(\sum_{i=1}^{N}\Delta(y_i,w_i)\right). \end{aligned} \tag{23}$$

Similarly, we derive the posterior probability distribution

$$\begin{aligned} p_{\mathbf{z}|\mathbf{y}}((\mathbf{w},\mathbf{y})|\mathbf{y};\delta,\epsilon) &= \frac{p_{\mathbf{z},\mathbf{y}}((\mathbf{w},\mathbf{y}),\mathbf{y};\delta,\epsilon)}{p_{\mathbf{y}}(\mathbf{y};\delta,\epsilon)} \\ &= \frac{p_{\mathbf{w},\mathbf{y}}(\mathbf{w},\mathbf{y};\delta,\epsilon)}{p_{\mathbf{y}}(\mathbf{y};\delta,\epsilon)} \\ &= \frac{\prod_{i=1}^{N} p_{w_i,y_i}(w_i,y_i;\delta,\epsilon)}{\prod_{i=1}^{N} p_{y_i}(y_i;\delta,\epsilon)} \\ &= \prod_{i=1}^{N} p_{w_i|y_i}(w_i|y_i;\delta,\epsilon). \end{aligned} \tag{24}$$

Let us define

$$\begin{aligned} q_i\left(\delta,\epsilon\right) &\triangleq p_{w_i|y_i}(1|y_i;\delta,\epsilon) \\ &= \frac{\delta\epsilon^{1-\Delta(y_i,1)}(1-\epsilon)^{\Delta(y_i,1)}}{\delta\epsilon^{1-\Delta(y_i,1)}(1-\epsilon)^{\Delta(y_i,1)} + (1-\delta)\epsilon^{1-\Delta(y_i,0)}(1-\epsilon)^{\Delta(y_i,0)}} \\ &= \frac{\delta\epsilon^{1-y_i}(1-\epsilon)^{y_i}}{\delta\epsilon^{1-y_i}(1-\epsilon)^{y_i} + (1-\delta)\epsilon^{y_i}(1-\epsilon)^{1-y_i}}. \end{aligned} \tag{25}$$

Now we are ready to derive the lower bound:

$$\begin{aligned} U(\delta,\epsilon;\delta',\epsilon') &= \mathbb{E}\left[\log p_{\mathbf{z}}(\mathbf{z};\delta,\epsilon)|\mathbf{y}=\mathbf{y};\delta',\epsilon'\right] \\ &= \log\delta\left(\sum_{i=1}^{N}\mathbb{E}\left[w_i|\mathbf{y}=\mathbf{y};\delta',\epsilon'\right]\right) \\ &+ \log(1-\delta)\left(N-\sum_{i=1}^{N}\mathbb{E}\left[w_i|\mathbf{y}=\mathbf{y};\delta',\epsilon'\right]\right) \\ &+ \log\epsilon\left(N-\sum_{i=1}^{N}\mathbb{E}\left[\Delta(y_i,w_i)|\mathbf{y}=\mathbf{y};\delta',\epsilon'\right]\right) \\ &+ \log(1-\epsilon)\left(\sum_{i=1}^{N}\mathbb{E}\left[\Delta(y_i,w_i)|\mathbf{y}=\mathbf{y};\delta',\epsilon'\right]\right). \end{aligned} \tag{26}$$

It is easy to see that

$$\mathbb{E}\left[w_i|\mathbf{y}=\mathbf{y};\delta',\epsilon'\right] = q_i\left(\delta',\epsilon'\right), \tag{27}$$

and

$$\begin{aligned} \mathbb{E}\left[\Delta(y_i,w_i)|\mathbf{y}=\mathbf{y};\delta',\epsilon'\right] &= q_i\left(\delta',\epsilon'\right)\Delta(y_i,1) + \left(1-q_i\left(\delta',\epsilon'\right)\right)\Delta(y_i,0) \\ &= q_i\left(\delta',\epsilon'\right)y_i + \left(1-q_i\left(\delta',\epsilon'\right)\right)(1-y_i). \end{aligned} \tag{28}$$

Substituting these expressions into the lower bound function, we obtain

$$U(\delta,\epsilon;\delta',\epsilon') = \log\delta\left(\sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right)\right) + \log(1-\delta)\left(N - \sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right)\right) + \log\epsilon\left(N - \sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right) y_i + \left(1 - q_i\left(\delta',\epsilon'\right)\right)\left(1-y_i\right)\right) + \log(1-\epsilon)\left(\sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right) y_i + \left(1 - q_i\left(\delta',\epsilon'\right)\right)\left(1-y_i\right)\right). \quad (29)$$

We note that the parameters δ and ϵ are separated in the lower bound, i.e., the derivative of $U(\delta,\epsilon;\delta',\epsilon')$ with respect to one of the parameters does not depend on the other parameter. This is a desired property as it allows us to optimize the lower bound with respect to each parameter separately. We will typically seek full data that leads to such separability.

Differentiating with respect to δ yields

$$\frac{\partial}{\partial\delta}U(\delta,\epsilon;\delta',\epsilon') = \frac{\sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right)}{\delta} - \frac{N - \sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right)}{1-\delta} = \frac{\sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right) - N\delta}{\delta(1-\delta)}. \quad (30)$$

Clearly, the maximum is achieved for

$$\hat{\delta} = \frac{1}{N}\sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right). \quad (31)$$

Similarly, we conclude that the value of ϵ that maximizes the lower bound is equal to

$$\hat{\epsilon} = 1 - \frac{1}{N}\sum_{i=1}^{N} q_i\left(\delta',\epsilon'\right) y_i + \left(1 - q_i\left(\delta',\epsilon'\right)\right)\left(1-y_i\right) = 1 - \frac{1}{N}\left(\sum_{y_i=1} q_i\left(\delta',\epsilon'\right) + \sum_{y_i=0}\left(1 - q_i\left(\delta',\epsilon'\right)\right)\right). \quad (32)$$

This concludes the derivation of the EM algorithm for this problem. Summarizing, the two steps of the algorithm are as follows:

E-step: Given the estimates of the parameters $\hat{\delta}^{(n)}$, $\hat{\epsilon}^{(n)}$, compute

$$q_i\left(\hat{\delta}^{(n)},\hat{\epsilon}^{(n)}\right) = \frac{\hat{\delta}^{(n)}\left(\hat{\epsilon}^{(n)}\right)^{1-y_i}\left(1-\hat{\epsilon}^{(n)}\right)^{y_i}}{\hat{\delta}^{(n)}\left(\hat{\epsilon}^{(n)}\right)^{1-y_i}\left(1-\hat{\epsilon}^{(n)}\right)^{y_i} + \left(1-\hat{\delta}^{(n)}\right)\left(\hat{\epsilon}^{(n)}\right)^{y_i}\left(1-\hat{\epsilon}^{(n)}\right)^{1-y_i}} \quad (33)$$

M-step: Update the parameters:

$$\hat{\delta}^{(n+1)} = \frac{1}{N} \sum_{i=1}^{N} q_i \left(\hat{\delta}^{(n)}, \hat{\epsilon}^{(n)} \right), \tag{34}$$

$$\hat{\epsilon}^{(n+1)} = 1 - \frac{1}{N} \left(\sum_{y_i=1} q_i \left(\hat{\delta}^{(n)}, \hat{\epsilon}^{(n)} \right) + \sum_{y_i=0} 1 - q_i \left(\hat{\delta}^{(n)}, \hat{\epsilon}^{(n)} \right) \right). \tag{35}$$

The update rules are quite intuitive (which is often the case with EM algorithms when the complete data is meaningfully chosen). The estimate $\hat{\delta}$ of the prior probability that a hidden bit w is equal to 1 is updated to be the average posterior probability that the hidden bits $w_1, \ldots, w_N$ are equal to 1. The estimate $\hat{\epsilon}$ of the probability that the observed bit is different from the hidden bit is updated using the corresponding average posterior probabilities as well. Note that we don't actually compute the lower bound $U(\delta, \epsilon; \delta\, \epsilon')$ explicitly when implementing the algorithm. Once we update the posterior probabilities $q_i(\hat{\delta}^{(n)}, \hat{\epsilon}^{(n)})$, the analytical form of the lower bound leads to the M-step above. Therefore, we can proceed directly to updating the parameters.

12.5 Further Reading

Several of the recommended textbooks discuss the EM algorithm.

Dempster, Laird and Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," J. Royal Stat. Society, vol. 39, no. 1, pp. 1-38, 1977, introduces the EM algorithm.

Wu and Jeff. "On the Convergence Properties of the EM Algorithm", analyzes the convergence of the algorithm.