19.2.2.2 An Example

We start with a simple example to clarify the concepts. Consider the simple network shown in figure 19.6. In the fully observable case, our maximum likelihood parameter estimate for the parameter \( \hat{\theta}_{d^1|C^0} \) is:

\[
\hat{\theta}_{d^1|C^0} = \frac{M[d^1, c^0]}{M[c^0]} = \frac{\sum_{m=1}^{M} I(\xi[m](D, C) = \langle d^1, c^0 \rangle)}{\sum_{m=1}^{M} I(\xi[m](C) = c^0)},
\]

where \( \xi[m] \) is the \( m \)th training example. In the fully observable case, we knew exactly whether the indicator variables were 0 or 1. Now, however, we do not have complete data cases, so we no longer know the value of the indicator variables.

Consider a partially specified data case \( \sigma = \langle a^1, \tau, \tau, d^0 \rangle \). There are four possible instantiations to the missing variables \( B, C \) which could have given rise to this partial data case: \( \langle b^1, c^1 \rangle \), \( \langle b^1, c^0 \rangle \), \( \langle b^0, c^1 \rangle \), \( \langle b^0, c^0 \rangle \). We do not know which of them is true, or even which of them is more likely.

However, assume that we have some estimate \( \theta \) of the values of the parameters in the model. In this case, we can compute how likely each of these completions is, given our distribution. That is, we can define a distribution \( Q(B, C) = P(B, C \mid \sigma, \theta) \) that induces a distribution over the four data cases. For example, if our parameters \( \theta \) are:

\[
\begin{align*}
\theta_{a^1} & = 0.3 & \theta_{b^1} & = 0.9 \\
\theta_{d^1|a^1} & = 0.1 & \theta_{d^1|b^1} & = 0.8 \\
\theta_{c^1|a^0,b^0} & = 0.83 & \theta_{c^1|a^1,b^0} & = 0.6 \\
\theta_{c^1|a^0,b^1} & = 0.09 & \theta_{c^1|a^1,b^1} & = 0.2,
\end{align*}
\]

then \( Q(B, C) = P(B, C \mid a^1, d^0, \theta) \) is defined as:

\[
\begin{align*}
Q(\langle b^1, c^1 \rangle) & = 0.3 \cdot 0.9 \cdot 0.2 \cdot 0.2 / 0.2196 = 0.0492 \\
Q(\langle b^1, c^0 \rangle) & = 0.3 \cdot 0.9 \cdot 0.8 \cdot 0.9 / 0.2196 = 0.8852 \\
Q(\langle b^0, c^1 \rangle) & = 0.3 \cdot 0.1 \cdot 0.6 \cdot 0.2 / 0.2196 = 0.0164 \\
Q(\langle b^0, c^0 \rangle) & = 0.3 \cdot 0.1 \cdot 0.4 \cdot 0.9 / 0.2196 = 0.0492,
\end{align*}
\]

where 0.2196 is a normalizing factor, equal to \( P(a^1, d^0 \mid \theta) \).

If we have another example \( \sigma' = \langle \tau, b^1, \tau, d^1 \rangle \). Then \( Q'(A, C) = P(A, C \mid b^1, d^1, \theta) \) is defined as:

\[
\begin{align*}
Q'(\langle a^1, c^1 \rangle) & = 0.3 \cdot 0.9 \cdot 0.2 \cdot 0.8 / 0.1675 = 0.2579 \\
Q'(\langle a^1, c^0 \rangle) & = 0.3 \cdot 0.9 \cdot 0.8 \cdot 0.1 / 0.1675 = 0.1290 \\
Q'(\langle a^0, c^1 \rangle) & = 0.7 \cdot 0.9 \cdot 0.9 \cdot 0.8 / 0.1675 = 0.2708 \\
Q'(\langle a^0, c^0 \rangle) & = 0.7 \cdot 0.9 \cdot 0.91 \cdot 0.1 / 0.1675 = 0.3423.
\end{align*}
\]

Intuitively, now that we have estimates for how likely each of the cases is, we can treat these estimates as truth. That is, we view our partially observed data case \( \langle a^1, \tau, \tau, d^0 \rangle \) as consisting of four complete data cases, each of which has some weight lower than 1. The weights correspond to our estimate, based on our current parameters, on how likely is this particular completion of the partial instance. (This approach is somewhat reminiscent of the weighted particles in the likelihood weighting algorithm.) Importantly, as we will discuss, we do
not usually explicitly generate these completed data cases; however, this perspective is the basis for the more sophisticated methods.

More generally, let $H[m]$ denote the variables whose values are missing in the data instance $o[m]$. We now have a data set $D^+$ consisting of

$$\cup_m \{\langle o[m], h[m] \rangle : h[m] \in \text{Val}(H[m])\},$$

where each data case $\langle o[m], h[m] \rangle$ has weight $Q(h[m]) = P(h[m] | o[m], \theta)$.

We can now do standard maximum likelihood estimation using these completed data cases. We compute the expected sufficient statistics:

$$\bar{M}_\theta[y] = \sum_{m=1}^{M} \sum_{h[m] \in \text{Val}(H[m])} Q(h[m]) I(\xi(Y) = y).$$

We then use these expected sufficient statistics as if they were real in the MLE formula. For example:

$$\bar{\theta}_{d^1|\circ^0} = \frac{\bar{M}_\theta[d^1, c^0]}{\bar{M}_\theta[c^0]}.$$

In our example, suppose the data consist of the two instances $o = \langle a^1, ?^1, ?, c^0 \rangle$ and $o' = \langle ?, b^1, ?, d^1 \rangle$. Then, using the calculated $Q$ and $Q'$ from above, we have that

$$\bar{M}_\theta[d^1, c^0] = Q(\langle a^1, c^0 \rangle) + Q'(\langle a^0, c^0 \rangle) = 0.1290 + 0.3423 = 0.4713,$$

$$\bar{M}_\theta[c^0] = Q(\langle b^1, c^0 \rangle) + Q'(\langle b^0, c^0 \rangle) + Q'(\langle a^1, c^0 \rangle) + Q'(\langle a^0, c^0 \rangle) = 0.8852 + 0.0492 + 0.1290 + 0.3423 = 1.4057.$$

Thus, in this example, using these particular parameters to compute expected sufficient statistics, we get

$$\bar{\theta}_{d^1|\circ^0} = \frac{0.4713}{1.4057} = 0.3353.$$

Note that this estimate is quite different from the parameter $\theta_{d^1|\circ^0} = 0.1$ that we used in our estimate of the expected counts. The initial parameter and the estimate are different due to the incorporation of the observations in the data.

This intuition seems nice. However, it may require an unreasonable amount of computation. To compute the expected sufficient statistics, we must sum over all the completed data cases. The number of these completed data cases is much larger than the original data set. For each $o[m]$, the number of completions is exponential in the number of missing values. Thus, if we have more than few missing values in an instance, an implementation of this approach will not be able to finish computing the expected sufficient statistics.

Fortunately, it turns out that there is a better approach to computing the expected sufficient statistic than simply summing over all possible completions. Let us reexamine the formula for an expected sufficient statistic, for example, $\bar{M}_\theta[c^1]$. We have that

$$\bar{M}_\theta[c^1] = \sum_{m=1}^{M} \sum_{h[m] \in \text{Val}(H[m])} Q(h[m]) I(\xi(C) = c^1).$$
Let us consider the internal summation, say for a data case \( o = (a^1, b^1, c^1) \). We have four possible completions, as before, but we are only summing over the two that are consistent with \( c^1 \), that is, \( Q(b^1, c^1) + Q(b^0, c^1) \). This expression is equal to \( Q(c^1) = P(c^1 \mid a^1, b^0, \theta) = P(c^1 \mid o[1], \theta) \). This idea clearly generalizes to our other data cases. Thus, we have that

\[
\tilde{M}_\theta[c^1] = \sum_{m=1}^{M} P(c^1 \mid o[m], \theta).
\]

Now, recall our formula for sufficient statistics in the fully observable case:

\[
M[c^1] = \sum_{m=1}^{M} I(\xi[m](C) = c^1).
\]

Our new formula is identical, except that we have substituted our indicator variable — either 0 or 1 — with a probability that is somewhere between 0 and 1. Clearly, if in a certain data case we get to observe \( C \), the indicator variable and the probability are the same. Thus, we can view the expected sufficient statistics as filling in soft estimates for hard data when the hard data are not available.

We stress that we use posterior probabilities in computing expected sufficient statistics. Thus, although our choice of \( \theta \) clearly influences the result, the data also play a central role. This is in contrast to the probabilistic completion we discussed earlier that used a prior probability to fill in values, regardless of the evidence on the other variables in the same instances.

### 19.2.2.3 The EM Algorithm for Bayesian Networks

We now present the basic EM algorithm and describe the guarantees that it provides.

**Networks with Table-CPDs** Consider the application of the EM algorithm to a general Bayesian network with table-CPDs. Assume that the algorithm begins with some initial parameter assignment \( \theta^0 \), which can be chosen either randomly or using some other approach. (The case where we begin with some assignment to the missing data is analogous.) The algorithm then repeatedly executes the following phases, for \( t = 0, 1, \ldots \).

**Expectation (E-step):** The algorithm uses the current parameters \( \theta^t \) to compute the expected sufficient statistics.

- For each data case \( o[m] \) and each family \( X, U \), compute the joint distribution \( P(X, U \mid o[m], \theta^t) \).
- Compute the expected sufficient statistics for each \( x, u \) as:

\[
\tilde{M}_{\theta^t}[x, u] = \sum_{m} P(x, u \mid o[m], \theta^t).
\]

This phase is called the E-step (expectation step) because the counts used in the formula are the expected sufficient statistics, where the expectation is with respect to the current set of parameters.
Algorithm 19.2 Expectation-maximization algorithm for BN with table-CPDs

Procedure Compute-ESS ( 
  \( G \), // Bayesian network structure over \( X_1, \ldots, X_n \) 
  \( \theta \), // Set of parameters for \( G \) 
  \( D \) // Partially observed data set 
 )

\[
1. \quad // \text{Initialize data structures}
2. \quad \text{for each } i = 1, \ldots, n 
3. \quad \text{for each } x_i, u_i \in \text{Val}(X_i, \text{Pa}_{X_i}^G) 
4. \quad M[x_i, u_i] \leftarrow 0 
5. \quad // \text{Collect probabilities from all instances} 
6. \quad \text{for each } m = 1 \ldots M 
7. \quad \text{Run inference on } (G, \theta) \text{ using evidence } o[m] 
8. \quad \text{for each } i = 1, \ldots, n 
9. \quad \text{for each } x_i, u_i \in \text{Val}(X_i, \text{Pa}_{X_i}^G) 
10. \quad M[x_i, u_i] \leftarrow M[x_i, u_i] + P(x_i | o[m]) 
11. \quad \text{return } \{ M[x_i, u_i] : \forall i = 1, \ldots, n, \forall x_i, u_i \in \text{Val}(X_i, \text{Pa}_{X_i}^G) \}
\]

Procedure Expectation-Maximization ( 
  \( G \), // Bayesian network structure over \( X_1, \ldots, X_n \) 
  \( \theta^0 \), // Initial set of parameters for \( G \) 
  \( D \) // Partially observed data set 
 )

\[
1. \quad // \text{E-step} 
2. \quad \{ M_t[x_i, u_i] \} \leftarrow \text{Compute-ESS}(G, \theta^t, D) 
3. \quad // \text{M-step} 
4. \quad \text{for each } i = 1, \ldots, n 
5. \quad \text{for each } x_i, u_i \in \text{Val}(X_i, \text{Pa}_{X_i}^G) 
6. \quad \theta_{x_i | u_i}^{t+1} \leftarrow \frac{M_t[x_i, u_i]}{M_t[u_i]} 
7. \quad \text{return } \theta^{t+1} 
\]

Maximization (M-step): Treat the expected sufficient statistics as observed, and perform maximum likelihood estimation, with respect to them, to derive a new set of parameters. In other words, set

\[
\theta_{x_i | u_i}^{t+1} = \frac{M_{t[x_i, u_i]}}{M_{t[u_i]}}.
\]

M-step

This phase is called the M-step (maximization step), because we are maximizing the likelihood relative to the expected sufficient statistics.

A formal version of the algorithm is shown fully in algorithm 19.2.