Learning with Missing Data

Guy Van den Broeck

November 13, 2014

For many applications, one cannot expect to observe every variable of every example in the data. There can be many reasons why data is missing. In surveys, some people will choose not to disclose private information. For medical data, it is not feasible to run every test on every patient. Historically, some data may have been forgotten, and can never be recovered. Sensory equipment might fail or produce faulty measurements. In all those cases, we are still interested in learning Bayesian networks and their parameters from missing data.

1 Missing Data Mechanisms

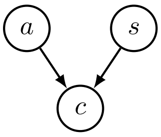
To learn from missing data, it is important to first understand the mechanisms that cause data to go missing. By assuming certain mechanisms, we will be able to derive specific algorithms for missing data and prove their properties.

There are three types of variables in missing data problems:

- *fully-observed* variables are always observed,
- *hidden* or *latent* variables are never observed, and
- *partially-observed* variables are observed in some examples but missing in others.

Latent variables are an interesting case: why would we want to learn with variables that are never observed? Latent variable can capture properties of the examples that we know exist, but that we cannot observe. For example, different users react differently to advertisement. We can model this by adding a latent "personality" variable to our model. Even though we cannot know the personality of any specific person (it is not even clear what the personality values would be), we can learn a model that clusters people according to their personality and uses this information to explain their behavior. Other examples of latent variables are genetic properties of patients, and topics of text documents. This are useful concepts to include into a machine learning model.

We begin by introducing some notation specific to missing data. As an example, consider Figure 1, depicting a Bayesian network, and a dataset with missing values, modeling the relationship between exposure to asbestos (a), being a smoker (s) and the incidence of lung cancer (c). Here, the value for variable c is always observed in the data, while the value for variables a and s can be missing. An *augmented dataset* extends a standard dataset with missing values as follows. For every partially-observed variable x, the augmented dataset has an additional random variable m_x denoting the missingness of x. When x is observed in the data, then $m_x = \mathsf{ob}$, and otherwise $m_x = \mathsf{unob}$. Note that the missingness variable m_x itself is always observed. For our example dataset, the augmented dataset has two additional missingness variables: m_a and m_s. This augmented dataset, and its augmented probability distribution $p(a, s, c, m_a, m_s)$, can serve as a useful tool for analyzing missing data problems.



(a) Bayesian network

a	s	c
?	1	1
1	0	0
0	?	1
0	?	0
1	1	1
?	0	0
1	0	1

(b) Dataset

a	s	c	m_a	m_s
?	1	1	unob	ob
1	0	0	ob	ob
0	?	1	ob	unob
0	?	0	ob	unob
1	1	1	ob	ob
?	0	0	unob	ob
1	0	1	ob	ob

(c) Augmented dataset

Figure 1: Data with missing values and Bayesian network

1.1 Missing Completely at Random (MCAR)

The most basic missing data mechanism is called *missing completely at random* (MCAR). This type of missing data occurs when the missingness of the data is independent of any of the random variables in the original Bayesian network. In other words, whether or not part of a data example is observed does not depend on the values in that example at all. For our example, an MCAR mechanism would be that random errors in the data collection process make that for 10% of the information about a, and 15% of the information about s, goes missing, but which values go missing is completely at random and does not depend on the values of a, s, or even c.

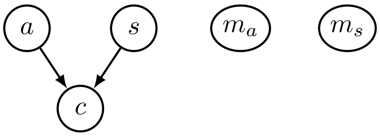


Figure 2: Missingness graph (augmented Bayesian network) for the MCAR assumption

More formally, the MCAR assumption says that the variables $\mathbf{x}$ in the Bayesian network, and the missingness variables $\mathbf{m}$, are independent, that is, $\mathbf{x} \perp\!\!\!\perp \mathbf{m}$. We can also depict such assumptions about $p(\mathbf{x}, \mathbf{m})$ graphically, in a Bayesian network for $p(\mathbf{x}, \mathbf{m})$ called a missingness graph. The MCAR assumption holds in a missingness graph where the $\mathbf{m}$ nodes are disconnected from the x nodes. Figure 2 depicts an MCAR missingness graph for our example.

The first and most basic algorithm to learn from missing data follows directly from the MCAR definition. Suppose we want the learn the parameters for the conditional probability $p(x|\mathbf{y})$. Under the MCAR assumption, we have that $x, \mathbf{y} \perp\!\!\!\perp \mathbf{m}$, which means that

$$p(x, \mathbf{y}, \mathbf{m}) = p(x, \mathbf{y}) \cdot p(\mathbf{m}) \tag{1}$$

$$p(x, \mathbf{y}, \mathbf{m} = \mathsf{ob}) = p(x, \mathbf{y}) \cdot p(\mathbf{m} = \mathsf{ob}) \tag{2}$$

$$p(x, \mathbf{y}) = p(x, \mathbf{y}|\mathbf{m} = \mathsf{ob}) \tag{3}$$

$$p(x|\mathbf{y}) = p(x|\mathbf{y}, \mathbf{m} = \mathsf{ob}). \tag{4}$$

Therefore, the probability $p(x|\mathbf{y})$ can be estimated from $p(x|\mathbf{y}, \mathbf{m} = \mathsf{ob})$. We already know how to do this: it is the probability $p(x|\mathbf{y})$ for to the cases where everything is fully observed ($\mathbf{m} = \mathsf{ob}$). Estimating this quantity, as in the complete-data case, can be done by simple counting. This algorithm is called *listwise deletion* or *complete-case analysis*.

When there a few examples in the data, or when many values are missing, listwise deletion becomes impractical. The *pairwise deletion* or *available-case analysis* algorithms avoids the cost of deleting all missing values. When estimating $p(x|\mathbf{y})$, we only need the value for x and the values for $\mathbf{y}$ to be observed. The fact that variables outside of x or $\mathbf{y}$ are unobserved does not matter. Hence, we can learn parameters as

$$p(x|\mathbf{y}) = p(x|\mathbf{y}, \mathbf{m}_x = \mathsf{ob}, \mathbf{m}_\mathbf{y} = \mathsf{ob}). \tag{5}$$

where again the problem is reduced to learning from complete data.

a	s	c	m_a	m_s
1	0	0	ob	ob
1	1	1	ob	ob
1	0	1	ob	ob

(a) Dataset for listwise deletion ($\mathbf{m} = \mathsf{ob}$)

a	s	c	m_a	m_s
?	1	1	unob	ob
1	0	0	ob	ob
1	1	1	ob	ob
?	0	0	unob	ob
1	0	1	ob	ob

(b) Dataset for $p(s)$ pairwise deletion ($m_s = \mathsf{ob}$)

Figure 3: Subsets of the dataset in Figure 1 used by deletion algorithms.

For our example, the MCAR assumption says that $a, s, c \perp\!\!\!\perp m_a, m_s$. Figure 3 shows the dataset used by listwise deletion. Indeed, this dataset has no missing values, at the cost of only having 3 examples. Listwise deletion estimates

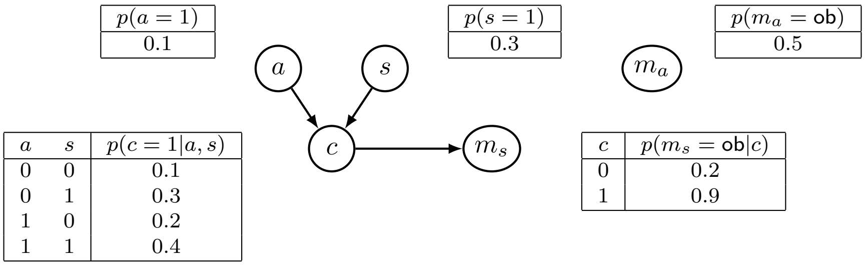
$$p(s) = p(s|m_a = \mathsf{ob}, m_s = \mathsf{ob}) \approx 1/3. \tag{6}$$

Pairwise deletion also learns from data where a is missing and estimates from 5 examples that

$$p(s) = p(s|m_s = \mathsf{ob}) \approx 2/5. \tag{7}$$

1.2 Missing at Random (MAR)

The MCAR assumption is strong and does not often hold in real-world data. It is more common to assume that the data is *missing at random* (MAR). This assumption holds if the missingness variables are conditionally independent of the partially-observed variables x_m given the fully-observed variables x_o, i.e., if $x_m \perp\!\!\!\perp m \mid x_o$. Graphically, this corresponds to a missingness graph where the $\mathbf{m}$ are allowed to have parents that are fully observed. For example, suppose that



$p(a = 1)$
0.1

$p(s = 1)$
0.3

$p(m_a = \mathsf{ob})$
0.5

a	s	$p(c = 1|a, s)$
0	0	0.1
0	1	0.3
1	0	0.2
1	1	0.4

c	$p(m_s = \mathsf{ob}|c)$
0	0.2
1	0.9

Figure 4: An MAR missingness graph

people diagnosed with cancer are more likely to divulge private information, and tell doctors whether they smoked or not. The missingness of s now depends on the value of c. In our example missingness graph, this means that there is an edge from c to m_s (see Figure 4). Whether s is missing depends on all other variables, including the value of s. However, given that we know c, the missingness of s becomes independent of its value. The variable a remains MCAR.

Deletion methods for MCAR data lead to *biased* parameter estimates on MAR data. Suppose we are interested in learning the parameter for $p(c = 1|a = 0, s = 1)$ whose true value is 0.3. Listwise or pairwise deletion (they are identical in this case) estimate this parameter as

$$\begin{aligned} p(c = 1|a = 0, s = 1, \mathbf{m} = \mathsf{ob}) &= p(c = 1|a = 0, s = 1, m_a = \mathsf{ob}, m_s = \mathsf{ob}) && (8) \\ &= p(c = 1|a = 0, s = 1, m_s = \mathsf{ob}) && (9) \\ &= \frac{p(c = 1, m_s = \mathsf{ob}|a = 0, s = 1)}{p(m_s = \mathsf{ob}|a = 0, s = 1)} && (10) \\ &= \frac{p(c = 1|a = 0, s = 1)p(m_s = \mathsf{ob}|c = 1)}{\sum_c p(c|a = 0, s = 1)p(m_s = \mathsf{ob}|c)} && (11) \\ &= \frac{0.3 \cdot 0.9}{0.3 \cdot 0.9 + 0.7 \cdot 0.2} = 0.66. && (12) \end{aligned}$$

This means that, even when we have plenty of data, only looking at the part of the data that is complete will result in biased probabilities. The probability of cancer for patients that smoke but were not exposed to asbestos is overestimated by 36%. Indeed, the missingness mechanism is more likely to show us cases where smoking and cancer co-occur, and more likely to hide the other cases. We need more sophisticated algorithms to correctly learn from MAR data.

One can show that MAR missingness graphs are always *identifiable*, assuming that all parameters are non-zero. This means that, given sufficient data, and two possible missingness graphs, we can tell with high confidence which missingness graph the data was generated from.

1.3 Missing Not at Random (MNAR)

All missing data mechanisms that are not MAR (and therefore also not MNAR) are categorized as *missing not at random* (MNAR). In these mechanisms, there is a dependency between a variable and its missingness, even when given the fully observed variables. Figure 5 depicts an MNAR missingness graph.

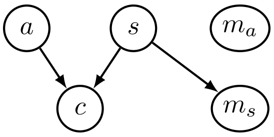
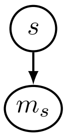


Figure 5: MNAR missingness graph

MNAR missingness mechanisms are in general *non-identifiable*. This means that there exist two different Bayesian networks and missingness mechanisms that generate the same data. The distribution over the observed values is identical. Consider a distribution with a single variable s that is missing not at random, and the two distributions, $p(s, m_s)$ and $p'(s, m_s)$ in Figure 6. The second and fourth row of the table both result in data where s is missing. These two distributions therefore generate the same data when

$$\theta_1 = \theta'_1, \qquad \theta_3 = \theta'_3, \qquad \text{and} \quad \theta_2 + \theta_4 = \theta'_2 + \theta'_4. \qquad (13)$$



(a) MNAR missingness graph

s	m_s	$p(s, m_s)$	$p'(s, m_s)$
0	ob	θ_1	θ'_1
0	unob	θ_2	θ'_2
1	ob	θ_3	θ'_3
1	unob	θ_4	θ'_4

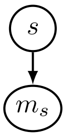
(b) Joint probability distributions

Figure 6: Two distributions for a simple MNAR missingness graph

Two concrete examples of such indistinguishable distributions are show in Figure 7. In both cases, the data will have 30% value 0, 10% value 1, and the remaining 60% of values missing. In the underlying distribution, however, $p(s)$ is very different.

s	m_s	$p(s, m_s)$	$p'(s, m_s)$
0	ob	0.3	0.3
0	unob	0.1	0.5
1	ob	0.1	0.1
1	unob	0.5	0.1

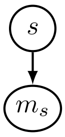
(a) Distributions



$p(s=1)$
0.6

s	$p(m_s = \text{ob}\|s)$
0	0.750
1	0.166

(b) Parameters p



$p(s=1)$
0.2

s	$p(m_s = \text{ob}\|s)$
0	0.375
1	0.500

(c) Parameters p'

Figure 7: Two non-identifiable MNAR missingness graphs

2 Maximum-Likelihood Learning with Missing Data

In the complete-data case, maximum-likelihood learning seeks to optimize the *likelihood* of the parameters given the data, which is defined as the probability of the data given the parameters:

$$\mathcal{L}(\theta|\mathfrak{D}) = p(\mathfrak{D}|\theta) = \prod_{i=1}^{N} p(\mathbf{d}_i|\theta) \tag{14}$$

where N is the dataset size and $\mathbf{d}_i$ represents the assignments made in the ith complete example of the dataset. We can define a similar likelihood function in the incomplete-data case. Now, a data example $\mathbf{d}_i$ may assign values to only a subset of the variables. The hidden variables in example i are denoted $\mathbf{h}_i$. The *marginal likelihood* is then

$$\mathcal{L}(\theta|\mathfrak{D}) = p(\mathfrak{D}|\theta) = \prod_{i=1}^{N} p(\mathbf{d}_i|\theta) = \prod_{i=1}^{N} \sum_{\mathbf{h}_i} p(\mathbf{d}_i, \mathbf{h}_i|\theta). \tag{15}$$

Suppose we know the missingness mechanism's structure and the parameters in the conditional probability tables for the missingness mechanisms $\mathbf{m}$. When we include this information into the likelihood equation, it becomes

$$\mathcal{L}_{\mathbf{m}}(\theta|\mathfrak{D}) = \prod_{i=1}^{N} p(\mathbf{d}_i, \mathbf{m}_{\mathbf{d}_i}|\theta) = \prod_{i=1}^{N} \sum_{\mathbf{h}_i} p(\mathbf{d}_i, \mathbf{h}_i, \mathbf{m}_{\mathbf{d}_i} = \text{ob}|\theta), \tag{16}$$

where $\mathbf{m}_{\mathbf{d}_i} = \mathsf{ob}$ denotes that the variables in $\mathbf{d}_i$ are observed. Now, the marginal likelihood function has a desirable property for MAR data: it decomposes as

$$\mathcal{L}_{\mathbf{m}}(\theta|\mathfrak{D}) = \prod_{i=1}^{N} \sum_{\mathbf{h}_i} p(\mathbf{d}_i, \mathbf{h}_i, \mathbf{m}_{\mathbf{d}_i} = \mathsf{ob}|\theta) \tag{17}$$

$$= \prod_{i=1}^{N} \sum_{\mathbf{h}_i} p(\mathbf{m}_{\mathbf{d}_i} = \mathsf{ob}|\mathbf{d}_i, \mathbf{h}_i, \theta) p(\mathbf{d}_i, \mathbf{h}_i|\theta) \tag{18}$$

$$= \prod_{i=1}^{N} \sum_{\mathbf{h}_i} p(\mathbf{m}_{\mathbf{d}_i} = \mathsf{ob}|\mathbf{d}_i) p(\mathbf{d}_i, \mathbf{h}_i|\theta) \tag{19}$$

$$= \prod_{i=1}^{N} p(\mathbf{m}_{\mathbf{d}_i} = \mathsf{ob}|\mathbf{d}_i) \sum_{\mathbf{h}_i} p(\mathbf{d}_i, \mathbf{h}_i|\theta) \tag{20}$$

$$= \left(\prod_{i=1}^{N} p(\mathbf{m}_{\mathbf{d}_i} = \mathsf{ob}|\mathbf{d}_i)\right) \cdot \left(\prod_{i=1}^{N} \sum_{\mathbf{h}_i} p(\mathbf{d}_i, \mathbf{h}_i|\theta)\right), \tag{21}$$

where we have used the MAR assumption in going from Equation 18 to 19. The parameters θ only appear in the second factor, which is independent of the missing data mechanisms $\mathbf{m}$. We can optimize the likelihood by only optimizing the second factor. This means that the maximum-likelihood parameters are the same for all mechanisms. It gives us a license to ignore the missing-data mechanism during parameter learning from MAR data.

The above property no longer holds MNAR data. As we saw in Section 1.3, two different sets of parameters can produce the exact same data under the MNAR assumption. In this case, knowing the missingness mechanism changes the maximum-likelihood parameters.

For complete data, optimizing the likelihood function was easy because it factorized in terms of the parameters. Unfortunately, this is no longer the case with missing data: the parameters are coupled, and the likelihood landscape is complex, with a large number of local optima.

3 Expectation Maximization

There are several algorithms to learn maximum-likelihood parameters from data, including gradient ascent and its extensions. By far the most popular technique, however, is the *expectation maximization* (EM) algorithm.

3.1 Algorithm

EM is an iterative algorithm that produces a sequence of parameters estimates $\theta^0, \theta^1, \theta^2, \ldots$. The initial parameters θ^0 are set arbitrarily. Next, EM iteratively performs two steps:

1. complete the missing data using parameters θ^{i-1}, and

2. estimate new parameters θ^i on the completed data.

These steps are repeated until either the θ^i, or the marginal likelihood $\mathcal{L}(\theta^i|\mathfrak{D})$ have converged.

The first step of EM is illustrated in Figure 8. The completed data set contains all possible ways of filling in the missing values in the incomplete data. Every complete data case gets assigned a weight, which is the probability of that completion according to the parameters θ^{i-1}. For example, completing the case $(?, 1, 1)$ to $(1, 1, 1)$ has weight $p_{\theta^{i-1}}(a = 1 \mid s = 1, c = 1)$.

a	s	c
?	1	1
1	0	0
0	?	1
0	?	0
1	1	1
?	0	0
1	?	?

(a) Incomplete data

a	s	c	Weight	For θ^{i-1} all 0.5
1	1	1	$p(a=1 \mid s=1, c=1, \theta^{i-1})$	$= 0.5$
0	1	1	$p_{\theta^{i-1}}(a=0 \mid s=1, c=1, \theta^{i-1})$	$= 0.5$
1	0	0	1	$= 1$
0	**1**	1	$p(b=1 \mid a=0, c=1, \theta^{i-1})$	$= 0.5$
0	**0**	1	$p(b=0 \mid a=0, c=1, \theta^{i-1})$	$= 0.5$
0	**1**	0	$p(b=1 \mid a=0, c=0, \theta^{i-1})$	$= 0.5$
0	**0**	0	$p(b=0 \mid a=0, c=0, \theta^{i-1})$	$= 0.5$
1	1	1	1	$= 1$
1	0	0	$p(a=1 \mid s=0, c=0, \theta^{i-1})$	$= 0.5$
0	0	0	$p(a=0 \mid s=0, c=0, \theta^{i-1})$	$= 0.5$
1	**0**	**1**	$p(b=0, c=1 \mid a=1, \theta^{i-1})$	$= 0.25$
1	**0**	**0**	$p(b=0, c=0 \mid a=1, \theta^{i-1})$	$= 0.25$
1	**1**	**1**	$p(b=1, c=1 \mid a=1, \theta^{i-1})$	$= 0.25$
1	**1**	**0**	$p(b=1, c=0 \mid a=1, \theta^{i-1})$	$= 0.25$

(b) Completed data

Figure 8: An incomplete dataset and its completion according to the first step of EM

Computing these weights requires running inference in the Bayesian network with parameters θ^{i-1}. Note that the weights for all the complete cases that come from the same incomplete case sum to one. The last column of Figure 8 shows the weights when θ^{i-1} sets all parameters in the network to 0.5.

The second step of EM estimates new parameters θ^i from the completed data. When dealing with standard complete data, we estimate the parameter $\theta_{x|y}$ as $\mathbb{I}_{\mathfrak{D}}(xy)/\mathbb{I}_{\mathfrak{D}}(y)$, where $\mathbb{I}_{\mathfrak{D}}(z)$ is the number of cases in $\mathfrak{D}$ where z holds. However, in the EM algorithm, our complete-data cases have weights. We can define the *expected count* $\mathbb{I}_{\mathfrak{D},i-1}(z)$ as the sum of weights of all complete cases where z holds, according to the parameters θ^{i-1}. Figure 9 shows the expected counts for completed data in Figure 8. From these expected counts, we estimate the new parameter $\theta^i_{x|y}$ as $\mathbb{I}_{\mathfrak{D},i-1}(xy)/\mathbb{I}_{\mathfrak{D},i-1}(y)$.

a	s	c	Expected count $\mathbb{I}_{\mathfrak{D},i-1}(world)$	For θ^{i-1} all 0.5
0	0	0	$p(b=0 \mid a=0, c=0, \theta^{i-1}) + p(a=0 \mid s=0, c=0, \theta^{i-1})$	$= 1$
0	0	1	$p(b=0 \mid a=0, c=1, \theta^{i-1})$	$= 0.5$
0	1	0	$p(b=1 \mid a=0, c=0, \theta^{i-1})$	$= 0.5$
0	1	1	$p(b=1 \mid a=0, c=1, \theta^{i-1}) + p(a=0 \mid s=1, c=1, \theta^{i-1})$	$= 1$
1	0	0	$1 + p(a=1 \mid s=0, c=0, \theta^{i-1}) + p(b=0, c=0 \mid a=1, \theta^{i-1})$	$= 1.75$
1	0	1	$p(b=0, c=1 \mid a=1, \theta^{i-1})$	$= 0.25$
1	1	0	$p(b=1, c=0 \mid a=1, \theta^{i-1})$	$= 0.25$
1	1	1	$p(a=1 \mid s=1, c=1, \theta^{i-1}) + 1 + p(b=1, c=1 \mid a=1, \theta^{i-1})$	$= 1.75$

Figure 9: Expected counts for all complete-data cases

Suppose that we have seeded θ^0 such that all parameters are 0.5. The last column of Figure 9 shows the expected counts under θ^0 for every complete data case. In the first iteration of EM, we estimate the parameters θ^1 from this table. For example, $\theta^1_{a=1} = (1.75 + 0.25 + 0.25 + 1.75)/7 = 4/7$ and $\theta^1_{c=1|a=0,s=0} = (0.5)/(1 + 0.5) = 1/3$. Thus, in the first iteration of EM, the parameter $\theta_{a=1}$ increases, and the parameter $\theta_{c=1|a=0,s=0}$ decreases. These new parameters will induce new weights for the completed data in the next iteration. In turn, this will change the expected counts and lead to new parameter estimates.

3.2 Analysis

A key property of EM is that it *monotonically increases the marginal likelihood* of the parameters, that is, $\mathcal{L}(\theta^i|\mathfrak{D}) \geq \mathcal{L}(\theta^{i-1}|\mathfrak{D})$. Moreover, the fixpoints of EM, where $\theta^i = \theta^{i-1}$, are exactly the stationary points of the marginal (log-)likelihood function, where its gradient is zero. These properties make that EM effectively finds parameters that locally maximize the marginal likelihood. There is no guarantee, though, that EM will find the parameters that globally maximize the marginal likelihood. Which optimum EM ends up in depends strongly on the initial parameters θ^0. For this reason, one often runs EM multiple times with different θ^0.

These key properties follow from a characterization of EM as optimizing another type of likelihood, called the *expected log-likelihood*, at each iteration. For each data example i, we know its observed values $\mathbf{d}_i$. Suppose for now that we also know the values $\mathbf{h}_i$ of the hidden variables. Then, recall that the log-likelihood of that complete data is

$$\mathcal{LL}(\theta|\mathfrak{D}) = \sum_{i=1}^{N} \log p(\mathbf{d}_i, \mathbf{h}_i|\theta) \tag{22}$$

However, we do not know the true values of $\mathbf{h}_i$. Instead, we can assume certain parameters θ' and compute the expected value of $\log p(\mathbf{d}, \mathbf{h}|\theta)$ according to the distribution induced by θ'. This expected value is $\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{d}, \theta') \cdot \log p(\mathbf{d}, \mathbf{h}|\theta)$. The total expected log-likelihood then becomes

$$\mathcal{ELL}(\theta|\mathfrak{D}, \theta') = \sum_{i=1}^{N} \sum_{\mathbf{h}_i} p(\mathbf{h}_i|\mathbf{d}_i, \theta') \cdot \log p(\mathbf{d}_i, \mathbf{h}_i|\theta). \tag{23}$$

Note that this function involves two sets of parameters. The first parameters θ are the ones we want to compute the log-likelihood of. We can only do so if we assume a distribution over the hidden values. This assumed distribution is captured by the second parameters θ'.

The expected log-likelihood relates to the likelihood as follows. Let $\theta^\star$ be the parameters that maximize the expected log-likelihood:

$$\theta^\star = \arg\max_{\theta} \mathcal{ELL}(\theta|\mathfrak{D}, \theta'). \tag{24}$$

For these parameters, one can show that $\mathcal{L}(\theta^\star|\mathfrak{D}) \geq \mathcal{L}(\theta'|\mathfrak{D})$. Thus, maximizing the expected log-likelihood monotonically increases the marginal likelihood. Moreover, one can show that the parameters θ^i computed by the EM algorithm at iteration i are exactly the parameters that maximize $\mathcal{ELL}(\theta|\mathfrak{D}, \theta^{i-1})$. This proves that EM monotonically increases the marginal likelihood.