Unsupervised Maximum Likelihood (re-)Estimation

Hidden variable variant

Suppose you no longer see the outcome of $Z$; you still see the tosses of the chosen coin, but you can’t tell which it is.

The data now looks like this

<table>
<thead>
<tr>
<th>d</th>
<th>Z</th>
<th>$X$: tosses of chosen coin</th>
<th>H counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?</td>
<td>H H H H H H H T T</td>
<td>(8H)</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>T T H H T T T H T T</td>
<td>(2H)</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>H T H H T H H H H T</td>
<td>(7H)</td>
</tr>
<tr>
<td>4</td>
<td>?</td>
<td>H T H H H T H H H H</td>
<td>(8H)</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>T T T T T T H T T T</td>
<td>(1H)</td>
</tr>
<tr>
<td>6</td>
<td>?</td>
<td>H H T H H H H H H H</td>
<td>(9H)</td>
</tr>
<tr>
<td>7</td>
<td>?</td>
<td>T H H T H H H H H H</td>
<td>(7H)</td>
</tr>
<tr>
<td>8</td>
<td>?</td>
<td>H H H H H T H H H</td>
<td>(9H)</td>
</tr>
<tr>
<td>9</td>
<td>?</td>
<td>H H T T T T T T H T</td>
<td>(3H)</td>
</tr>
</tbody>
</table>

$Z$ is so-called hidden variable in each case.
We still have the probability model for combinations \((Z, X)\), with the same parameters \(\theta_s, \theta_{h|a}\) and \(\theta_{h|b}\).

We would still like to find values for \(\theta_s, \theta_{h|a}\) and \(\theta_{h|b}\) which again maximise the probability of the observed data.

For each \(d\) we just know the coin-tosses \(X^d\). Their probability is now a sum

\[
p(X^d) = p(Z = a)p(X^d|Z = a) + p(Z = b)p(X^d|Z = b) = \theta_s \theta_{h|a} (1 - \theta_{h|a})^{\#(d,t)} + (1 - \theta_s) \theta_{h|b} (1 - \theta_{h|b})^{\#(d,t)},
\]

and the entire data set’s probability, \(p(d)\) is the product:

\[
p(d) = \prod_d p(X^d)
\]

\[
= \prod_d \left[ \theta_s \theta_{h|a} (1 - \theta_{h|a})^{\#(d,t)} + (1 - \theta_s) \theta_{h|b} (1 - \theta_{h|b})^{\#(d,t)} \right] (11)
\]

\[
= \prod_d \left[ \theta_s \theta_{h|a} (1 - \theta_{h|a})^{\#(d,t)} + (1 - \theta_s) \theta_{h|b} (1 - \theta_{h|b})^{\#(d,t)} \right] (12)
\]

The 'product of sums' problem

\[
p(d) = \prod_d \left[ \theta_s \theta_{h|a} (1 - \theta_{h|a})^{\#(d,t)} + (1 - \theta_s) \theta_{h|b} (1 - \theta_{h|b})^{\#(d,t)} \right]
\]

so can we maximise (12), repeated above?

the preceding procedure of taking logs runs into a dead-end, because \(p(d)\) is no longer all products, turning into sums. Instead the log is

\[
\sum_d \log \left[ \theta_s \theta_{h|a} (1 - \theta_{h|a})^{\#(d,t)} + (1 - \theta_s) \theta_{h|b} (1 - \theta_{h|b})^{\#(d,t)} \right]
\]

and there is no known way to cleverly break this down as there was before

this is essentially the problem we face if we want to do parameter estimation with hidden variables – this is done widely in eg. Machine Translation and Speech Recognition. The EM or 'Expectation Maximisation' algorithm will turn out to be the solution.

The general hidden variable set-up

before proceeding lets try to make clear the general case of a hidden variable problem

You have \(D\) data items

In the fully observed case, each data item \(d\) is represented by the values of a set of variables, which we’ll split into two sets \(\{z^d, x^d\}\)

and you have a probability model – ie. formula – spelling how likely any such fully observed case is \(P((z^d, x^d); \theta)\) where \(\theta\) are all the parameters of the model

In the hidden case, for each data item \(d\) you just have values on a subset of the variables \(x^d\); the other variables \(z^d\) are hidden.

If \(\mathcal{A}(z)\) represents the space of all possible values for the variables \(z\), then the probability of each partial data item is

\[
P(x^d; \theta) = \sum_{k \in \mathcal{A}(z)} P(z = k|x^d; \theta)
\]

Taking stock: what kinds of thing can we calculate?

- **parameters given visible data**: we have seen illustrations where \(z\) is known for each datum, and where finding parameter values maximising the data likelihood was easy: its relative frequencies all the way (scenario 1: 1 vis var; scenario 2: 2 vis vars, one for coin-choice, and one for the coin-tosses on whatever coin was chosen). In fact to do the parameter estimation we really just needed numbers about how often types of outcomes occurred.

- **posterior probs on hidden vars**: if we have all the parameters \(\theta\), for datum \(d\) we can ‘easily’ work out \(P(z = k|x^d; \theta)\). In our third scenario where the coin choice was hidden, for \(Z = a\) the formula is

\[
P(Z = a|X^d; \theta_s, \theta_{h|a}, \theta_{h|b}) = \frac{\theta_s \theta_{h|a} (1 - \theta_{h|a})^{\#(d,t)} + (1 - \theta_s) \theta_{h|b} (1 - \theta_{h|b})^{\#(d,t)}}{	heta_s \theta_{h|a} (1 - \theta_{h|a})^{\#(d,t)} + (1 - \theta_s) \theta_{h|b} (1 - \theta_{h|b})^{\#(d,t)}}
\]

- EM methods put those two abilities to use in iterative procedures to re-estimate parameters.
The EM Algorithm

The EM algorithm is a parameter (re-)estimation procedure, which starting from some original setting of parameters $\theta^0$, generates a converging sequence of re-estimates:

$$\theta^0 \rightarrow \ldots \rightarrow \theta^n \rightarrow \theta^{n+1} \rightarrow \ldots \rightarrow \theta^{\text{final}}$$

where each $\theta^n$ goes to $\theta^{n+1}$ by a so-called E-step, followed by a M step:

**E step**

generate a virtual complete data corpus by treating each incomplete data item $(x^d)$ as standing for all possible completions with values for $z$, $(z = k, x^d)$, weighting each by its conditional probability $P(z = k| x^d; \theta^n)$, under current parameters $\theta^n$: often this quantity is called the responsibility. Use $\gamma_d(k)$ for $P(z = k| x^d)$.

**M step**

treating the ‘responsibilities’ $\gamma_d(k)$ as if they were counts, apply maximum likelihood estimation to the virtual corpus to derive new estimates $\theta^{n+1}$.

**EM sketch**

Let’s use the notation $\gamma_d(k)$ for $P(z = k| x^d)$ – which is something of a convention in EM methods

we will describe EM for the moment as just a kind of procedure or recipe.

Later we will consider how to show that the procedure does something sensible.

- **Viterbi EM**: (i) using some values for $\theta$, for each $d$ work out $\gamma_d(k)$ for each value $k \in A(z)$; (ii) pick the best $z = k$ and ‘complete’ $d$ with this value for $z$ making a virtual complete corpus; (iii) re-estimate $\theta$ on this virtual data. If you go back to (i) and do this over and over again you would be doing what is called Viterbi EM

- **real EM**: (i) using some values for $\theta$, for each $d$ work out $\gamma_d(k)$ for each value $k \in A(z)$; (ii) pretend these $\gamma_d(k)$ are counts in a virtual corpus of completions of $d$; (iii) re-estimate $\theta$ on this virtual data. If you go back to (i) and do this over and over again you would be doing what is called EM

**EM sketch specific for scenario 3 (hidden coin choice)**

- **Viterbi EM**: (i) using some values for $\theta, \theta_{hi}, \theta_{hl}$, for each $d$ work out $\gamma_d(k)$ for each value $k \in \{a, b\}$; (ii) pick the best $Z = k$ and ‘complete’ $d$ with this value for $Z$ making a virtual complete corpus; (iii) re-estimate $\theta, \theta_{hi}, \theta_{hl}$ on this virtual data. If you go back to (i) and do this over and over again you would be doing what is called Viterbi EM

- **real EM**: (i) using some values for $\theta, \theta_{hi}, \theta_{hl}$, for each $d$ work out $\gamma_d(k)$ for each value $k \in \{a, b\}$; (ii) pretend these $\gamma_d(k)$ are counts in a virtual corpus of completions of $d$; (iii) re-estimate $\theta, \theta_{hi}, \theta_{hl}$ on this virtual data. If you go back to (i) and do this over and over again you would be doing what is called EM
The EM Algorithm

In the **E**-step you should picture each data point \( X \) as split into virtual populations of \( Z = a \) and \( Z = b \) versions, with \( \gamma_d(Z) \) as the virtual counts:

\[
\begin{align*}
X^1 : (8H, 2T) & \quad \{ (z = a, X^1) \gamma_1(a) = 0.88 \\ (z = b, X^1) \gamma_1(b) = 0.12 \} \\
X^6 : (9H, 1T) & \quad \{ (z = a, X^6) \gamma_6(a) = 0.92 \\ (z = b, X^6) \gamma_6(b) = 0.08 \}
\end{align*}
\]

\[
\begin{align*}
X^2 : (2H, 8T) & \quad \{ (z = a, X^2) \gamma_2(a) = 0.34 \\ (z = b, X^2) \gamma_2(b) = 0.66 \} \\
X^7 : (7H, 3T) & \quad \{ (z = a, X^7) \gamma_7(a) = 0.83 \\ (z = b, X^7) \gamma_7(b) = 0.17 \}
\end{align*}
\]

\[
\begin{align*}
X^3 : (7H, 3T) & \quad \{ (z = a, X^3) \gamma_3(a) = 0.83 \\ (z = b, X^3) \gamma_3(b) = 0.17 \} \\
X^8 : (9H, 1T) & \quad \{ (z = a, X^8) \gamma_8(a) = 0.92 \\ (z = b, X^8) \gamma_8(b) = 0.08 \}
\end{align*}
\]

\[
\begin{align*}
X^4 : (8H, 2T) & \quad \{ (z = a, X^4) \gamma_4(a) = 0.88 \\ (z = b, X^4) \gamma_4(b) = 0.12 \} \\
X^9 : (3H, 7T) & \quad \{ (z = a, X^9) \gamma_9(a) = 0.22 \\ (z = b, X^9) \gamma_9(b) = 0.78 \}
\end{align*}
\]

\[
\begin{align*}
X^5 : (1H, 9T) & \quad \{ (z = a, X^5) \gamma_5(a) = 0.45 \\ (z = b, X^5) \gamma_5(b) = 0.55 \}
\end{align*}
\]

\[1\text{the } \gamma_d(Z) \text{ numbers above assume } \theta_a = 0.5, \theta_b|a = 0.4, \theta_b|b = 0.3\]

---

**Example calc of }\gamma_1(Z)\text{**

\[
d = 1 : p(Z = a, HHHHHHHTT) = 0.5 \times (0.4)^8 \times (0.6)^2 = 1.17965 \times 10^{-4}
\]

\[
d = 1 : p(Z = b, HHHHHHHTT) = 0.5 \times (0.3)^8 \times (0.7)^2 = 1.60744 \times 10^{-5}
\]

\[
d = 1 : \text{sum} = 0.000134039
\]

\[
\gamma_1(a) = \frac{p(Z = a, HHHHHHHTT)}{\sum_z p(Z = z, HHHHHHHTT)} = \frac{1.17965 \times 10^{-4}}{0.000134039} = 0.880077
\]

\[
\gamma_1(b) = \frac{p(Z = b, HHHHHHHTT)}{\sum_z p(Z = z, HHHHHHHTT)} = \frac{1.60744 \times 10^{-5}}{0.000134039} = 0.119923
\]

---

**M step for coin example**

In the **M** step you treat the \( \gamma_d(Z) \) values as if they were genuine counts and re-estimate parameters in the usual common-sense fashion based on relative frequencies.

As a mental trick to help visualize you might consider all the preceding \( \gamma_d(Z) \) as multiplied by 100 – effectively each single \( d \) is being treated as split out into 100 virtual versions, with \( \gamma_d(Z) \times 100 \) for each \( Z \) alternative

the 'common-sense' re-estimation of the parameters obtained this way represent a maximum likelihood estimate for any complete corpus that exhibits the same ratios as the obtained virtual corpus.
M step for coin example

For the coin scenario we can write down formulae for what the new round of estimates will be

In (9), (9), (9) we had the estimation formulae for the fully observed case, making use of an indicator functions \( \delta(d,.) \) – which for any given \( d \) are 1 for just one value of \( Z \). The re-estimation formula for an M step are just these with the indicator function \( \delta(d,.) \) replaced throughout by \( \gamma_d(.) \)

\[
est(\theta_a) = \frac{\sum_d \gamma_d(A)}{D} \tag{13}
est(\theta_{h|a}) = \frac{\sum_d \gamma_d(A) \#(d,h)}{\sum_d \gamma_d(A)10} \tag{14}
est(\theta_{h|b}) = \frac{\sum_d \gamma_d(B) \#(d,h)}{\sum_d \gamma_d(B)10} \tag{15}
\]

Properties of EM re-estimation

EM starts with some setting \( \theta^0 \) of the parameters and one E-M cycle takes one setting \( \theta^n \) into another \( \theta^{n+1} \).

the data gets likelier over the iterations

\[ P(d;\theta^n) \leq P(d;\theta^{n+1}) \]

because the data cannot just get likelier and likelier, the procedure converges to a final setting \( \theta^{\text{final}} \)

so whatever values \( \theta^0 \) you start with, running the algorithm will give you better values \( \theta^{\text{final}} \)

'common sense' M-step for \( \theta_a, \theta_{h|a} \) and \( \theta_{h|b} \)

in case that did not persuade, here's how to get to these re-estimation formulae by 'common sense' based on the virtual corpus

for \( \theta_a \), need \( (\text{cnt of virtual } Z = A \text{ cases})/(\text{cnt of all virtual } Z \text{ cases}) \), ie.

\[
est(\theta_a) = \frac{\sum_d \gamma_d(a)}{\sum_d \gamma_d(a) + \sum_d \gamma_d(b)} = \frac{\sum_d \gamma_d(a)}{\sum_d (\gamma_d(a) + \gamma_d(b))} = \frac{\sum_d \gamma_d(A)}{D} \tag{16}
\]

for \( \theta_{h|a} \), need

\( (\text{cnt of } H \text{ in virtual } Z = a \text{ cases})/(\text{cnt of all tosses in virtual } Z = a \text{ cases}) \), ie.

\[
est(\theta_{h|a}) = \frac{\sum_d \gamma_d(a) \#(d,h)}{\sum_d \gamma_d(a) \#(d,h) + \#(d,t)} = \frac{\sum_d \gamma_d(a) \#(d,h)}{\sum_d \gamma_d(a)10} \tag{17}
\]

for \( \theta_{h|b} \), need

\( (\text{cnt of } H \text{ in virtual } Z = b \text{ cases})/(\text{cnt of all tosses in virtual } Z = b \text{ cases}) \), ie.

\[
est(\theta_{h|b}) = \frac{\sum_d \gamma_d(b) \#(d,h)}{\sum_d \gamma_d(b) \#(d,h) + \#(d,t)} = \frac{\sum_d \gamma_d(b) \#(d,h)}{\sum_d \gamma_d(b)10} \tag{18}
\]

some provisos though ...

- Caveat One: there may be many local maxima, so there is no guarantee that the re-estimation will converge to the best values
- Caveat Two: if the data set \( d \) is rather small the derived parameters may fit fresh data only poorly – this the classic over-fitting problem.
- Caveat Three: it will be prohibitively expensive to calculate all \( \gamma_d(k) \) if the set \( A(z) \) of the possible values of \( z \) is exponentially big. This does not apply to our hidden coin choice scenario – size of \( A(z) \) is 2 – but definitely applies to applications we are going to look at (eg. in Machine Translation and Speech Recognition) and requires algorithmic ingenuity to make it still work.
A numerically worked example

To keep things manageable on slides lets suppose a minute data set
\[
\begin{array}{ccc}
1 & ? & H \\
2 & ? & T \\
\end{array}
\]

looks like having \( A \) be entirely biased one way, and \( B \) entirely the other will give maximum prob to this. The outcomes when EM is run from start \( \theta_a = 0.5, \theta_{b|a} = 0.75 \) and \( \theta_{b|b} = 0.5 \) is:

\[
\begin{array}{cc}
\theta_a & \theta_{b|a} & \theta_{b|b} & \text{logprob} & \text{prob} \\
0.5 & 0.75 & 0.5 & -3.97763 & 0.0634766 \\
0.446154 & 0.775862 & 0.277778 & -3.36722 & 0.0969094 \\
0.467361 & 0.922972 & 0.128866 & -2.59395 & 0.165632 \\
0.49254 & 0.993083 & 0.0214144 & -2.08205 & 0.236179 \\
0.5 & 1 & 0 & -2 & 0.25 \\
\end{array}
\]

so EM finds the intuitive solution. Next few slides trace the first iteration of the algorithm.

On the particular data set at hand the joint probability formulae are particularly simple

\[
\begin{align*}
P(Z = a, \mathbf{X}^d) &= \theta_a \times (\theta_{h|a})^2 \\
P(Z = b, \mathbf{X}^d) &= \theta_b \times (\theta_{h|b})^2 \\
P(Z = a, \mathbf{X}^d) &= \theta_a \times (\theta_{h|a})^2 \\
P(Z = b, \mathbf{X}^d) &= \theta_b \times (\theta_{h|b})^2 \\
\end{align*}
\]

and thus the formulae for \( \gamma(d|Z) \) are:

\[
\begin{align*}
\gamma_1(a) &= \frac{\theta_a \times (\theta_{h|a})^2}{\theta_a \times (\theta_{h|a})^2 + \theta_b \times (\theta_{h|b})^2} \\
\gamma_1(b) &= \frac{\theta_b \times (\theta_{h|b})^2}{\theta_a \times (\theta_{h|a})^2 + \theta_b \times (\theta_{h|b})^2} \\
\gamma_2(a) &= \frac{\theta_a \times (\theta_{t|a})^2}{\theta_a \times (\theta_{t|a})^2 + \theta_b \times (\theta_{t|b})^2} \\
\gamma_2(b) &= \frac{\theta_b \times (\theta_{t|b})^2}{\theta_a \times (\theta_{t|a})^2 + \theta_b \times (\theta_{t|b})^2} \\
\end{align*}
\]

To carry out an EM estimation of the parameters given the data we need some initial setting of the parameters. We will suppose this is:

\[
\begin{align*}
\theta_a &= \frac{1}{2}, \theta_b = \frac{1}{2}, \theta_{h|a} = \frac{3}{4}, \theta_{t|a} = \frac{1}{4}, \theta_{h|b} = \frac{1}{2}, \theta_{t|b} = \frac{1}{2}
\end{align*}
\]

ITERATION 1

For each piece of data have to first compute the conditional probabilities of the hidden variable given the data:

\[
\begin{align*}
d = 1 &: p(Z = A, HH) = 0.5 \times 0.75 \times 0.75 = 0.28125 \\
d = 1 &: p(Z = B, HH) = 0.5 \times 0.5 \times 0.5 = 0.125 \\
d = 1 &: \Rightarrow \text{sum} = 0.40625 \\
d = 1 &: \Rightarrow \gamma_1(A) = 0.692308 \\
d = 1 &: \Rightarrow \gamma_1(B) = 0.307692 \\
d = 2 &: p(Z = A, TT) = 0.5 \times 0.25 \times 0.25 = 0.03125 \\
d = 2 &: p(Z = B, TT) = 0.5 \times 0.5 \times 0.5 = 0.125 \\
d = 2 &: \Rightarrow \text{sum} = 0.15625 \\
d = 2 &: \Rightarrow \gamma_2(A) = 0.2 \\
d = 2 &: \Rightarrow \gamma_2(B) = 0.8
\end{align*}
\]
Armed with these \( \gamma \) values we now treat each data item \( X^d \) as if it splits into two versions, one filling out \( Z \) as \( a \), and with `count` \( \gamma_d(a) \), and one filling out \( Z \) as \( b \), and with `count` \( \gamma_d(b) \).

\[
X^1 : (2H, 0T) \begin{cases} (z = a, X^1) \ \gamma_1(a) = 0.692308 \\ (z = b, X^1) \ \gamma_1(b) = 0.307692 \end{cases}
\]

\[
X^2 : (0H, 2T) \begin{cases} (z = a, X^2) \ \gamma_2(a) = 0.2 \\ (z = b, X^2) \ \gamma_2(b) = 0.8 \end{cases}
\]

Then from these `expected` counts we re-estimate parameters

\[
est(\theta_a) = E(A)/2 = 0.892308/2 = 0.446154
\]
\[
est(\theta_b) = E(B)/2 = 1.107692/2 = 0.553846
\]

Note the denominator 2 in the re-estimation formula for \( \theta_a \). We could have written the denominator as \( E(A) + E(B) \), but this is

\[
\sum_d \gamma_d(a) + \sum_d \gamma_d(b) = \sum_d [\gamma_d(a) + \gamma_d(b)] = \sum_d [1] = 2
\]

Then from these `expected` counts we re-estimate parameters

\[
est(\theta_{h|a}) = E(A, H)/\sum_X [E(A, X)] = 1.38462/(1.38462 + 0.4) = 1.38462/1.78462 = 0.775862
\]
\[
est(\theta_{t|a}) = E(A, T)/\sum_X [E(A, X)] = 0.4/(1.38462 + 0.4) = 0.4/1.78462 = 0.224138
\]
re-estimating $\theta_{h|b}$

$$est(\theta_{h|b}) = E(B, H) / \sum_x [E(B, X)] = 0.615385 / (0.615385 + 1.6)$$

$$= 0.615385 / 2.21538$$

$$= 0.277778$$

$$est(\theta_{t|b}) = E(B, T) / \sum_x [E(B, X)] = 1.6 / (0.615385 + 1.6)$$

$$= 1.6 / 2.21538$$

$$= 0.722222$$

More realistic run of EM

recall the data we had for our 2nd scenario, with the coin-choice observed:

<table>
<thead>
<tr>
<th>$d$</th>
<th>$Z$</th>
<th>$X$: tosses of chosen coin</th>
<th>$H$ counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A H H H H H H H T T</td>
<td>(8H)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>B T T H H T H T T T T</td>
<td>(2H)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A H T H H T H H H H</td>
<td>(7H)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A H T H H H T H H H</td>
<td>(8H)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>B T T T T T T T T T</td>
<td>(1H)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>A H H T H H H H H H</td>
<td>(9H)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A T H H H H H H H</td>
<td>(7H)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A H H H H H T H H</td>
<td>(9H)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>B H H T T T T T T</td>
<td>(3H)</td>
<td></td>
</tr>
</tbody>
</table>

recall supervised estimation gave: $\theta_a = 0.66, \theta_{h|a} = 0.8, \theta_{h|b} = 0.2$

the above traced through how the 2nd row of the table below comes from the first.

| $\theta_a$ | $\theta_{h|a}$ | $\theta_{h|b}$ | logprob  | prob     |
|------------|---------------|---------------|----------|----------|
| 0.5        | 0.75          | 0.5           | -3.97763 | 0.0634766|
| 0.446154   | 0.755862      | 0.277778      | -3.36722 | 0.0969094|
| 0.467361   | 0.922972      | 0.128866      | -2.59395 | 0.165632 |
| 0.49254    | 0.993083      | 0.0214144     | -2.08205 | 0.236179 |
| 0.5        | 1.0           | 0.0           | -2.0     | 0.25     |

In the end it converges to $\theta_a = 0.5, \theta_{h|a} = 1, \theta_{h|b} = 0$.

also tracked in the table is the increasing prob of the data, and log-prob

More realistic run of EM continued

here’s an outcome of running EM treating $Z$ as hidden

| $\theta_a$ | $\theta_{h|a}$ | $\theta_{h|b}$ | logprob  | prob     |
|------------|---------------|---------------|----------|----------|
| 0.5        | 0.4           | 0.3           | -101.033 | 3.85587e-31|
| 0.698501   | 0.70713       | 0.351806      | -77.3507 | 5.18952e-24|
| 0.666619   | 0.793432      | 0.213219      | -73.2502 | 8.90206e-23|
| 0.66705    | 0.799293      | 0.200725      | -73.2201 | 9.08992e-23|
| 0.667134   | 0.799354      | 0.200453      | -73.2201 | 9.08999e-23|

no further change

these are very close to the numbers obtained when $Z$ was not hidden.

On this data set also the final outcome is not very dependent on the initial values