EM Algorithm applied to Hidden Markov Trees (HMTs)

Brendan Law

9 January, 2019

??

A document showing my formal(-ish) derivation for the Expectation Maximisation (EM) algorithm as applied to HMTs. The derivation of this borrows heavily from the EM algorithm as applied to Hidden Markov Models (HMMs), but with a tree structure governing the latent states and dependencies, instead of a standard HMM (usually sequential states indexed by time). Rather than being markovian on the state underlying the previous time-indexed observation, the HMT is markovian on the state underlying the parent of an observation in the tree. I have borrowed heavily from the lecture notes presented in Li (2008), but according to Crouse et al. (1998), there is extra material to be found in Ronen et al. (1995) for trees, and in Rabiner (1989) - the original paper on EM parameterisation of a HMM.

The derivation here applies to generic data, which has a distribution governed by the parameters related to its underlying state, and where the state relationships are tied together by a generic tree structure. The idea is to apply this further to wavelet coefficients with related by a HMT structure, as mentioned in Crouse et al. (1998). This paper suggests one of two states (high variance vs low variance), imposing one of two distributions onto each observation, as well as requiring the markov tree to be a binary tree (each node having two children, except the leaf nodes).

1 Preamble

Is there any better way to define it instead of using a pair?? Nodes?

- Data-state pairs, $\{x_i, z_i\}$, for each $i \in \{1, \ldots, n\}$:
  - Data (continuous or discrete): x_i
  - Latent (hidden) states : $z_i = k,\ k \in \{1, \ldots, m\}$
- Generic density of data, conditional on state: $f_{X_i|Z_i}(x_i \mid z_i = k, \boldsymbol{\theta}) \sim$ Distribution governed by state k
- $\boldsymbol{\theta}$ is a vector of parameters. It includes:
  - Parameters relating to each of the k distributions
  - The probability of each state, k, for the root ~~data~~-state ~~pair~~ only $P(Z_1 = k \mid \boldsymbol{\theta}) \equiv P(Z_1 = k) = \pi_k$
  - The transition probabilities between parent and child states $P(Z_i = k \mid Z_{p(i)} = l, \boldsymbol{\theta}) \equiv P(Z_i = k \mid Z_{p(i)} = l) = \epsilon_{i,p(i)}^{kl}$ for each pair $i \in \{2, \ldots, n\}$, and each pair of states $(k, l) \in \{1, \ldots, m\}^2$
  - Also note that:
    - $\sum_{k=1}^{m} \pi_k = 1$
    - $\sum_{l=1}^{m} P(Z_i = k, Z_{p(i)} = l) = P(Z_i = k)$
  - The EM algorithm is such that there is an initial estimate of $\boldsymbol{\theta}$, which is then updated at each iteration until some sort of convergence is achieved.
- Vector of observations, $\mathbf{X} = (x_1, \ldots, x_n)$, and corresponding states, $\mathbf{Z} = (z_1, \ldots, z_n)$

Different definition of pairs, but it can be confusing.

Some extra notation and assumptions regarding the tree structure:

- For a given pair, i, the parent of that pair is denoted $p(i)$, and the set of children (there may be multiple) of that pair are contained ~~in the~~ set denoted $c(i)$.
- The root of the tree is denoted to start at the first pair; $\{x_1, z_1\}$
- There are no dependencies imposed on the data ~~points,~~ $\mathbf{X}$. These only depend on each other through the dependencies imposed on the latent states underlying the data, $\mathbf{Z}$

directly

Observations are independent conditional on their states

- ~~Observations are *conditionally independent based only on their states*~~, with a few key principles:
  1. Conditional on the state of a data-state pair, a wavelet coefficient is **independent of all other random variables** (all other states and coefficients). That is, a pair's state holds all the information required to determine the probability distribution of its data, regardless of the other states or data.

$$f\big(x_i \mid \{X_j\}_{j\neq i},\, \{Z_j = z_j\}_{j\neq i},\, Z_i = z_i\big) = f_{X_i|Z_i}(x_i \mid Z_i = z_i) \tag{1}$$

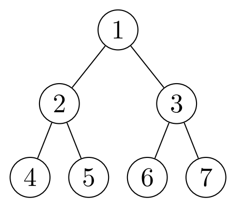
$X_{p(i)}$

  2. **Given the parent state**, $Z_{p(i)}$, $\{X_i, Z_i\}$ is independent of the entire tree except the dependents of data point i
  3. **Given the child state**, Z_j, $j \in c(i)$, $\{X_i, Z_i\}$ is independent of all of data point i's dependents

$Z_{p(i)}$ X_i Z_i $j \in c(i)$

2 Derivation of joint distribution of states

Consider a binary tree of 7 data-state pairs, as per the diagram below (which shows the states only):



To derive the joint distribution of states, we use the conditional probability definition, as well as the property of state independence conditional on the parent, repeatedly. The derivation is as follows:

$$\begin{aligned}
P(\mathbf{Z}) &= P(Z_1, \ldots, Z_7)\\
&= P(Z_2, \ldots, Z_7 \mid Z_1)P(Z_1)\\
&= P(Z_2, Z_4, Z_5 \mid Z_1)P(Z_3, Z_6, Z_7 \mid Z_1)P(Z_1)\\
&\qquad\qquad\text{(conditional independence of subtrees rooted at children of } Z_1)\\
&= P(Z_4, Z_5 \mid Z_1, Z_2)P(Z_2 \mid Z_1)P(Z_6, Z_7 \mid Z_1, Z_3)P(Z_3 \mid Z_1)P(Z_1)\\
&= P(Z_4, Z_5 \mid Z_2)P(Z_2 \mid Z_1)P(Z_6, Z_7 \mid Z_3)P(Z_3 \mid Z_1)P(Z_1)\\
&\qquad\qquad\text{(conditional on parent's state, independent of } Z_1)\\
&= P(Z_4 \mid Z_2)P(Z_5 \mid Z_2)P(Z_2 \mid Z_1)P(Z_6 \mid Z_3)P(Z_7 \mid Z_3)P(Z_3 \mid Z_1)P(Z_1)\\
&\qquad\qquad\text{(conditional independence, given parent)}
\end{aligned}$$

Alternatively, in reverse:

$$\begin{aligned}
P(\mathbf{Z}) &= P(Z_1, \ldots, Z_7)\\
&= P(Z_4 \mid Z_1, \ldots, Z_3, Z_5, \ldots, Z_7)P(Z_1, \ldots, Z_3, Z_5, \ldots, Z_7)\\
&= P(Z_4 \mid Z_2)P(Z_1, \ldots, Z_3, Z_5, \ldots, Z_7) &&\text{(conditional independence given parent state)}\\
&= P(Z_4 \mid Z_2)P(Z_5 \mid Z_2)P(Z_6 \mid Z_3)P(Z_7 \mid Z_3)P(Z_1, Z_2, Z_3) &&\text{(same for other states, given parents)}\\
&= P(Z_4 \mid Z_2)P(Z_5 \mid Z_2)P(Z_6 \mid Z_3)P(Z_7 \mid Z_3)P(Z_2 \mid Z_1)P(Z_3 \mid Z_1)P(Z_1)
\end{aligned}$$

Therefore, we can generalise it as such:

$$P(\mathbf{Z}) = P(Z_1)\prod_{i=2}^{n} P(Z_i \mid Z_{p(i)}) \tag{2}$$

This is almost identical to the HMM, where we have states indexed by time, Z_t, $t \in \{1, \ldots, T\}$, whereby the states are Markov on the previous state (the state in time t is only dependent on that in time $t-1$). The joint distribution of the states in a HMM would be:

$$\begin{aligned}
P(\mathbf{Z}) &= P(Z_1, \ldots, Z_T) \\
&= P(Z_T \mid Z_1, \ldots, Z_{T-1}) P(Z_1, \ldots, Z_{T-1}) \\
&= P(Z_T \mid Z_{T-1}) P(Z_1, \ldots, Z_{T-1}) \\
&= \ldots \\
&= P(Z_1) \prod_{t=2}^{T} P(Z_t \mid Z_{t-1})
\end{aligned}$$

Almost identical to a HMT, but where the states are Markov on the parent's state. This is known as the *ordered Markov property*; the theories here relate to a broader area of study surrounding Directed Graphical Models (DGM) - see Murphy (2012).

3 Complete log likelihood derivation

$$\begin{aligned}
P(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) &= P(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta}) P(\mathbf{Z} \mid \boldsymbol{\theta}) && \text{(def'n of conditional probability)} \\
&= P(X_1, \ldots, X_n \mid Z_1, \ldots, Z_n, \boldsymbol{\theta}) P(Z_1, \ldots, Z_n \mid \boldsymbol{\theta}) && \text{(expanding the vectors)} \\
&= \left[\prod_{i=1}^{n} P(X_i \mid Z_1, \ldots, Z_n, \boldsymbol{\theta}) \right] P(Z_1, \ldots, Z_n \mid \boldsymbol{\theta}) \\
& && \text{(conditional independence of } X_i\text{'s given their states)} \\
&= \left[\prod_{i=1}^{n} P(X_i \mid Z_1, \ldots, Z_n, \boldsymbol{\theta}) \right] P(Z_1 \mid \boldsymbol{\theta}) \prod_{i=2}^{n} P(Z_i \mid Z_{p(i)}, \boldsymbol{\theta}) && \text{(using (2))} \\
&= \left[\prod_{i=1}^{n} P(X_i \mid Z_i, \boldsymbol{\theta}) \right] P(Z_1 \mid \boldsymbol{\theta}) \prod_{i=2}^{n} P(Z_i \mid Z_{p(i)}, \boldsymbol{\theta}) && \text{(distribution of } X_i \text{ determined by state, } Z_i) \\
&= \prod_{k=1}^{m} \pi_k^{\mathbb{1}\{Z_1 = k\}} \prod_{k=1}^{m} \prod_{l=1}^{m} \prod_{i=2}^{n} \epsilon_{i,p(i)}^{l,k\ \mathbb{1}\{Z_i = l\}\mathbb{1}\{Z_{p(i)} = k\}} \prod_{k=1}^{m} \prod_{i=1}^{n} P(X_i \mid Z_i = k, \boldsymbol{\theta})^{\mathbb{1}\{Z_i = k\}} \\
& && \text{(law of total probability; specific states for each obs, represented by indicator RVs)} \\
\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) &= \log(P(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})) \\
&= \log \left(\prod_{k=1}^{m} \pi_k^{\mathbb{1}\{Z_1 = k\}} \prod_{k=1}^{m} \prod_{l=1}^{m} \prod_{i=2}^{n} \epsilon_{i,p(i)}^{l,k\ \mathbb{1}\{Z_i = l\}\mathbb{1}\{Z_{p(i)} = k\}} \prod_{k=1}^{m} \prod_{i=1}^{n} P(X_i \mid Z_i = k, \boldsymbol{\theta})^{\mathbb{1}\{Z_i = k\}} \right) \\
&= \sum_{k=1}^{m} \mathbb{1}\{Z_1 = k\} \log \pi_k \ldots \\
&+ \sum_{k=1}^{m} \sum_{l=1}^{m} \sum_{i=2}^{n} \mathbb{1}\{Z_i = l\}\mathbb{1}\{Z_{p(i)} = k\} \log \epsilon_{i,p(i)}^{l,k} \ldots \\
&+ \sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{1}\{Z_i = k\} \log P(X_i \mid Z_i = k, \boldsymbol{\theta})
\end{aligned}$$

$\mathbb{1}\{Z_i = l, Z_{p(i)} = k\}$

Fix things based on suggestion above. Summation orders and

Note that:

$$\mathbb{1}\{Z_i = k\} = \begin{cases} 1 & Z_i = k \\ 0 & Z_i \neq k \end{cases}$$

This remains a random variable, as we don't know what value the latent state variable, Z_i, will take, but that it may take on any of $k \in \{1, \ldots, m\}$ with some probability. As we will see, after setting an initial guess of $\boldsymbol{\theta}$, both of the log terms are known (parameters given in θ), and are no longer random variables.

Log pi_k and log epsilon blahblah were never random variables. It's just unknown for unknown theta. After the initial guess, it's just known.

??

using an EM algorithm that iterates ....

4 EM algorithm

We will now compute the MLE of the parameters ~~in~~ $\boldsymbol{\theta}$ by ~~iterating~~ through the following two steps, and updating $\boldsymbol{\theta}$ at the end of each step.

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})}) = \mathbb{E}_{(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(\mathrm{t})})}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})] \qquad \text{(finding unknowns in this statement is the E-step)}$$

$$\boldsymbol{\theta}^{(\mathrm{t}+1)} := \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})}) \qquad \text{(this is the M-step)}$$

Perhaps define theta_(t)?

5 E-step: derivation

$$\begin{aligned}
Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(\mathrm{t})}) &= \mathbb{E}_{(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(\mathrm{t})})}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})] \\
&= \mathbb{E}_{(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(\mathrm{t})})}\Big[\\
&\sum_{k=1}^{m} \mathbb{1}\{Z_1 = k\} \log \pi_k \dots \\
&+ \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=2}^{n} \mathbb{1}\{Z_i = l\}\mathbb{1}\{Z_{p(i)} = k\} \log \epsilon_{i,p(i)}^{l,k} \dots \\
&+ \sum_{k=1}^{m}\sum_{i=1}^{n} \mathbb{1}\{Z_i = k\} \log P(X_i \mid Z_i = k, \boldsymbol{\theta})\Big] \\
&= \sum_{k=1}^{m} P(Z_1 = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})}) \log \pi_k \dots \\
&+ \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=2}^{n} P(Z_i = l,\, Z_{p(i)} = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})}) \log \epsilon_{i,p(i)}^{l,k} \dots \\
&+ \sum_{k=1}^{m}\sum_{i=1}^{n} P(Z_i = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})}) \log P(X_i \mid Z_i = k, \boldsymbol{\theta})\Big]
\end{aligned}$$

Change order of summations!

(linearity of expectations, expectation of indicator RVs)

It's just E(I{A and B}) = P(A and B).

Note that, in the above, we have used the fact that, for two events A, B:

$$\begin{aligned}
\mathbb{E}[\mathbb{1}\{A\}\mathbb{1}\{B\}] &= P(A \cap B) \\
&= P(A, B)
\end{aligned}$$

We now hit the expected hurdle in our expectation step - that for each observation i, and for each possible pairing of states k and l, we need to evaluate both $P(Z_i = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})})$ and $P(Z_i = l, Z_{p(i)} = k \mid \mathbf{X}, \boldsymbol{\theta}^{(\mathrm{t})})$. Calculating these two values in these setting is not as straightforward as the application of Bayes' rule in the GMM case. This is where the *forward-backward algorithm* comes in handy (in HMM cases; see Li (2008), Rabiner (1989)) - it allows for efficient evaluation of these two quantities. (Note that it is also known as the *upward-downward algorithm* in artificial intelligence literature - as we are adapting our workings to fit those in Crouse et al. (1998), we will use its notation and terminology to guide our steps).

haha I prefer the forward-backward algorithm

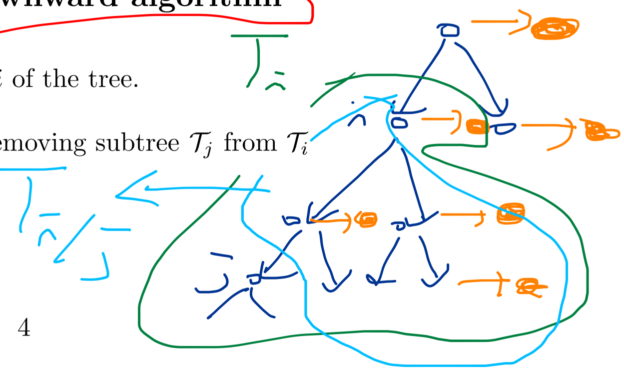
5.1 Evaluating probabilities - upward-downward algorithm

Notation (as per Crouse et al. (1998)):

$\mathcal{T}_i$ represents the subtree of data rooted at node i of the tree.

$\mathcal{T}_1 = (X_1, \dots, X_n)$, ie all the data in the tree.

$\mathcal{T}_{i\setminus j}$ represents the subtree of data obtained by removing subtree $\mathcal{T}_j$ from $\mathcal{T}_i$



Conditional likelihoods:

Probability of all data rooted at node i when the state for node it is given as k?

$$\beta_i(k) := P(\mathcal{T}_i \mid Z_i = k, \boldsymbol{\theta})$$
$$\beta_{i,p(i)}(k) := P(\mathcal{T}_i \mid Z_{p(i)} = k, \boldsymbol{\theta})$$
$$\beta_{p(i)\setminus i}(k) := P(\mathcal{T}_{p(i)\setminus i} \mid Z_{p(i)} = k, \boldsymbol{\theta})$$

Probability of all data rooted at node i when the state for i-th node's parent is given as k?

When the state for i-th node's parent is given as k, probability of all data rooted at the i-th node's parent minus all data rooted at the i-th node [or data at the i-th parent node + it's the other child's (not node it) descendants].

Joint probability functions:

$$\alpha_i(k) := P(Z_i = k, \mathcal{T}_{1\setminus i} \mid \boldsymbol{\theta})$$

??? P(T_p(i) l Z_p(i) = k) = P(T_i T_p(i)\i l Z_p(i) = k) <- can be expressed using beta_i,p(i) (k) and beta_p(i)\i(k)??

We can show some properties of the above three conditional likelihoods:

$$\begin{aligned}
\beta_{i,p(i)}(k) &= P(\mathcal{T}_i \mid Z_{p(i)} = k, \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(\mathcal{T}_i, Z_i = l \mid Z_{p(i)} = k, \boldsymbol{\theta}) && \text{(law of total probability)} \\
&= \sum_{l=1}^{m} P(\mathcal{T}_i \mid Z_i = l, Z_{p(i)} = k, \boldsymbol{\theta}) P(Z_i = l \mid Z_{p(i)} = k, \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(\mathcal{T}_i \mid Z_i = l, \boldsymbol{\theta}) P(Z_i = l \mid Z_{p(i)} = k, \boldsymbol{\theta}) \\
& && \text{(subtree rooted at i conditionally independent of parent's state given its state)} \\
&= \sum_{l=1}^{m} \beta_i(l) \epsilon_{i,p(i)}^{lk} && \text{(equation (21) in Crouse et al. (1998))}
\end{aligned}$$

$$\begin{aligned}
\beta_i(k) &= P(\mathcal{T}_i \mid Z_i = k, \boldsymbol{\theta}) \\
&= P(X_i, \mathcal{T}_{c(i)_{c_1}}, \ldots, \mathcal{T}_{c(i)_{c_n}} \mid Z_i = k, \boldsymbol{\theta}) && \text{(where $c(i)_{c_j}$ represents the jth child of i)} \\
&= \left[\prod_{j \in c(i)} P(\mathcal{T}_j \mid Z_i = k, \boldsymbol{\theta}) \right] P(X_i \mid Z_i = k, \boldsymbol{\theta})
\end{aligned}$$

Perhaps c(i)[j] might be better notation?

Inconsistent notations

Expressed as emission prob

(conditional independence between trees rooted at $c(i)$, and to X_i due to conditioning on Z_i. Eqn (22) in Crouse)

Expressed as beta_j,i(k)???

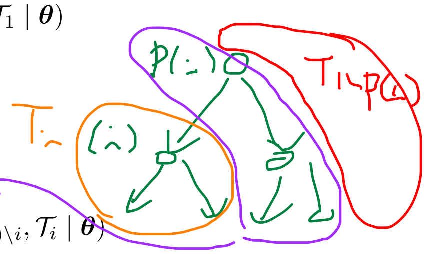
$$\begin{aligned}
\beta_{p(i)\backslash i}(k) &= P(\mathcal{T}_{p(i)\backslash i} \mid Z_{p(i)} = k, \boldsymbol{\theta}) \\
&= \cancel{\frac{\beta_{p(i)}(k)}{P(\mathcal{T}_i \mid S_{p(i)} = k, \boldsymbol{\theta})}} && \text{(can be derived from conditional independence as used above)} \\
&= \frac{\beta_{p(i)}(k)}{\beta_{i,p(i)}(k)} && \text{(equation (23) in Crouse et al. (1998))}
\end{aligned}$$

Beta_p(i)(k) = P(T_p(i) I Z_p(i) = k)
= P(T_p(i)\i, T_i I Z_p(i) = k)
= P(T_p(i)\i I Z_p(i) = k) P(T_i I Z_p(i) = k)
= Beta_p(i)\i(k) Beta_i,p(i)(k)

$$\begin{aligned}
\alpha_i(k) &= P(Z_i = k, \mathcal{T}_{1\backslash i} \mid \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(Z_i = k, Z_{p(i)} = l, \mathcal{T}_{1\backslash i} \mid \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(Z_i = k, Z_{p(i)} = l, \mathcal{T}_{1\backslash p(i)}, \mathcal{T}_{p(i)\backslash i} \mid \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i}, Z_i = k \mid \mathcal{T}_{1\backslash p(i)}, Z_{p(i)} = l, \boldsymbol{\theta}) P(\mathcal{T}_{1\backslash p(i)}, Z_{p(i)} = l \mid \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i}, Z_i = k \mid Z_{p(i)} = l, \boldsymbol{\theta}) \alpha_{p(i)}(l) \\
& && \text{(conditional independence on the two sets of wavelets due to conditioning on parent state)} \\
&= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i} \mid Z_i = k, Z_{p(i)} = l, \boldsymbol{\theta}) P(Z_i = k \mid Z_{p(i)} = l, \boldsymbol{\theta}) \alpha_{p(i)}(l) \\
&= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i} \mid Z_{p(i)} = l, \boldsymbol{\theta}) \epsilon_{i,p(i)}^{kl} \alpha_{p(i)}(l) && \text{(conditional independence given $Z_{p(i)}$)} \\
&= \sum_{l=1}^{m} \beta_{p(i)\backslash i}(l) \epsilon_{i,p(i)}^{kl} \alpha_{p(i)}(l) && \text{(equation (25) in Crouse et al. (1998))}
\end{aligned}$$

We can also derive the joint distributions of the two quantities we are after (rather than the quantities conditional on the data):

$$
\begin{aligned}
P(Z_i = k, \mathcal{T}_1 \mid \boldsymbol{\theta}) &= P(Z_i = k, \mathcal{T}_{1\setminus i}, \mathcal{T}_i \mid \boldsymbol{\theta}) \\
&= P(\mathcal{T}_i \mid Z_i = k, \mathcal{T}_{1\setminus i}, \boldsymbol{\theta}) P(Z_i = k, \mathcal{T}_{1\setminus i} \mid \boldsymbol{\theta}) \\
&= P(\mathcal{T}_i \mid Z_i = k, \boldsymbol{\theta}) P(Z_i = k, \mathcal{T}_{1\setminus i} \mid \boldsymbol{\theta}) \\
&= \beta_i(k)\alpha_i(k) \\
\therefore P(\mathbf{X} \mid \boldsymbol{\theta}) &= P(\mathcal{T}_1 \mid \boldsymbol{\theta}) \\
&= \sum_{k=1}^{m} P(Z_i = k, \mathcal{T}_1 \mid \boldsymbol{\theta}) \\
&= \sum_{k=1}^{m} \beta_i(k)\alpha_i(k)
\end{aligned}
$$



$$
\begin{aligned}
P(Z_i = k, Z_{p(i)} = l, \mathcal{T}_1 \mid \boldsymbol{\theta}) &= P(Z_i = k, Z_{p(i)} = l, \mathcal{T}_{1\setminus p(i)}, \mathcal{T}_{p(i)\setminus i}, \mathcal{T}_i \mid \boldsymbol{\theta}) \\
&= P(Z_{p(i)} = l, \mathcal{T}_{1\setminus p(i)} \mid \boldsymbol{\theta}) P(Z_i = k, \mathcal{T}_{p(i)\setminus i}, \mathcal{T}_i \mid Z_{p(i)} = l, \mathcal{T}_{1\setminus p(i)}, \boldsymbol{\theta}) \\
&= \alpha_{p(i)}(l) P(Z_i = k, \mathcal{T}_{p(i)\setminus i}, \mathcal{T}_i \mid Z_{p(i)} = l, \boldsymbol{\theta}) && \text{(conditional independence)} \\
&= \alpha_{p(i)}(l) P(\mathcal{T}_{p(i)\setminus i}, \mathcal{T}_i \mid Z_i = k, Z_{p(i)} = l, \boldsymbol{\theta}) P(Z_i = k \mid Z_{p(i)} = l, \boldsymbol{\theta}) \\
&= \alpha_{p(i)}(l) P(\mathcal{T}_{p(i)\setminus i}, \mathcal{T}_i \mid Z_i = k, Z_{p(i)} = l, \boldsymbol{\theta}) \epsilon_{i,p(i)}^{kl} \\
&= \alpha_{p(i)}(l) P(\mathcal{T}_{p(i)\setminus i} \mid Z_i = k, Z_{p(i)} = l, \boldsymbol{\theta}) P(\mathcal{T}_i \mid Z_i = k, Z_{p(i)} = l, \boldsymbol{\theta}) \epsilon_{i,p(i)}^{kl} \\
& && \text{(conditional independence)} \\
&= \alpha_{p(i)}(l) P(\mathcal{T}_{p(i)\setminus i} \mid Z_{p(i)} = l, \boldsymbol{\theta}) P(\mathcal{T}_i \mid Z_i = k, \boldsymbol{\theta}) \epsilon_{i,p(i)}^{kl} && \text{(conditional independence)} \\
&= \alpha_{p(i)}(l) \beta_{p(i)\setminus i}(l) \beta_i(k) \epsilon_{i,p(i)}^{kl} && \text{(conditional independence)}
\end{aligned}
$$

Hence we can write down formulae required for the quantities in the E-step:

$$
P(Z_i = k \mid \mathbf{X}, \boldsymbol{\theta}) = \frac{P(Z_i = k, \mathbf{X} \mid \boldsymbol{\theta})}{P(\mathbf{X} \mid \boldsymbol{\theta})} = \frac{\beta_i(k)\alpha_i(k)}{\sum_{l=1}^{m} \beta_i(l)\alpha_i(l)} \tag{3}
$$

$$
P(Z_i = k, Z_{p(i)} = l \mid \mathbf{X}, \boldsymbol{\theta}) = \frac{P(Z_i = k, Z_{p(i)} = l, \mathbf{X} \mid \boldsymbol{\theta})}{P(\mathbf{X} \mid \boldsymbol{\theta})} = \frac{\alpha_{p(i)}(l)\beta_{p(i)\setminus i}(l)\beta_i(k)\epsilon_{i,p(i)}^{kl}}{\sum_{l=1}^{m} \beta_i(l)\alpha_i(l)} \tag{4}
$$

To execute the algorithm the following steps are required (as per Crouse et al. (1998)):

Up-step

0. Initialise at **finest (lowest) scale**, $J = 1$: $\beta_i(k) = f(X_i \mid Z_i = k, \boldsymbol{\theta})$ for each $k \in \{1, \ldots, m\}$
1. $\forall Z_i$ at scale J, $\forall k \in \{1, \ldots, m\}$, calculate each of the following three quantities:
   (a) $\beta_{i,p(i)}(k)$
   (b) $\beta_{p(i)}(k)$
   (c) $\beta_{p(i)\setminus i}(k)$
2. $J := J + 1$
3. If $J = L$ (coarsest/highest level), then stop, else return to step 1.

Down-step

0. Initialise state Z_1 at scale level $J = L$: $\alpha_1(k) = P(Z_1 = k, \mathcal{T}_{1\setminus 1} \mid \boldsymbol{\theta}) = P(Z_1 = k \mid \boldsymbol{\theta}) \cancel{= P(Z_1 = k)}$, for each $k \in \{1, \ldots, m\}$

1. $J := J - 1$

2. Calculate, $\forall Z_i$ at scale J, $\forall k \in \{1, \dots, m\}$, $\alpha_i(k)$

3. If $J = 1$ (finest/lowest level), then stop, else return to step 1.

? What is “this” in “this is complete” and “this” in “this expression”?

Once this is complete, we can evaluate this expression as we have all the required information stored in our parameter set, $\boldsymbol{\theta}^{(t)}$. Just as in the GMM case, in deriving the expression for $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$, we are using the expected log-likelihood as a proxy for the actual log-likelihood, as the actual hidden states are unknown (we don't know the true values of Z_i or $Z_{p(i)}$). Thus, this algorithm aims to find a $\boldsymbol{\theta}$ which maximises the expected log-likelihood, over the probabilities of each data point taking on particular states.

6 M-step: maximisation

Now to derive the maximisation step. Again, we can find the $\boldsymbol{\theta}$ which maximises the expression by considering each part separately, and maximising the parameters which affect those parts.

Note that all these expressions have been derived for one tree, with each i corresponding to separate nodes from the one tree. Therefore, to parameterise the probabilities of each node, we only have one data point in this case.

To make our analysis more meaningful, let's consider the case with $t \in \{1, \dots, T\}$ trees. With multiple trees, in the E-step, we calculate both quantities ($P(Z_i^t = k \mid \mathbf{X}^t, \boldsymbol{\theta}^{(t)})$ and $P(Z_i^t = k, Z_{p(i)}^t = l \mid \mathbf{X}^t, \boldsymbol{\theta}^{(t)})$) for each tree, t, for each node, i in that tree.

Better to use different index for tree t because you already used t to expression iteration as in theta(t).

We re-notate our vectors as such:

- $\mathbf{X} = [\mathbf{X}^1, \dots, \mathbf{X}^T]$, and $\mathbf{X}^t = [X_1^t, \dots, X_n^t]$ for the data
- $\mathbf{Z} = [\mathbf{Z}^1, \dots, \mathbf{Z}^T]$, and $\mathbf{Z}^t = [Z_1^t, \dots, Z_n^t]$ for the states

Considering the line with $\log \pi_k$, our objective is to optimise:

$$\sum_{k=1}^{m} \sum_{t=1}^{T} P(\mathbf{Z}_1^t = k \mid \mathbf{X}^t, \boldsymbol{\theta}^{(t)}) \log \pi_k$$

change order because interpretation makes more sense..

subject to:

$$\sum_{k=1}^{m} \pi_k = 1$$

Denote $A_i^t(k) := P(\mathbf{Z}_i^t = k \mid \mathbf{X}^t, \boldsymbol{\theta}^{(t)})$.

Using the Lagrangian, we get:

$$\begin{aligned}
\mathcal{L} &= \sum_{k=1}^{m} \sum_{t=1}^{T} A_1^t(k) \log \pi_k + \lambda(1 - \sum_{k=1}^{m} \pi_k) \\
\Rightarrow \frac{\delta \mathcal{L}}{\delta \pi_k} &= \sum_{t=1}^{T} \frac{A_1^t(k)}{\pi_k} - \lambda = 0, \text{ for each } k, \text{ and} \\
\frac{\delta \mathcal{L}}{\delta \lambda} &= \sum_{k=1}^{m} \pi_k = 1
\end{aligned}$$

Combining the two, we get that:

$$
\begin{aligned}
\pi_k &= \frac{\sum_{t=1}^{T} A_1^t(k)}{\lambda} \\
\sum_{k=1}^{m} \pi_k &= \sum_{k=1}^{m} \frac{\sum_{t=1}^{T} A_1^t(k)}{\lambda} = 1 \\
\Rightarrow \lambda &= \sum_{k=1}^{m} \sum_{t=1}^{T} A_1^t(k)
\end{aligned}
$$

Use different notation for (t+1)!

$$
\begin{aligned}
\therefore \pi_k^{\cancel{(t+1)}} &= \frac{\sum_{t=1}^{T} A_1^t(k)}{\sum_{k=1}^{m} \sum_{t=1}^{T} A_1^t(k)} \\
&= \frac{\sum_{t=1}^{T} A_1^t(k)}{T}
\end{aligned}
$$

In general, we get that, for all states, i,: (HOW?!?!?)

$$
P(Z_i = k \mid \boldsymbol{\theta})^{(t+1)} = \frac{\sum_{t=1}^{T} A_i^t(k)}{T}
$$

Now consider the line with $\log \epsilon_{i,p(i)}^{l,k}$, our objective is to optimise:

$$
\sum_{t=1}^{T} \sum_{k=1}^{m} \sum_{l=1}^{m} \sum_{i=2}^{n} P(\mathbf{Z}_i^t = l,\, \mathbf{Z}_{p(i)}^t = k \mid \mathbf{X}^t, \boldsymbol{\theta}^{\cancel{(t)}}) \log \epsilon_{i,p(i)}^{l,k}
$$

subject to:

$$
\sum_{l=1}^{m} \epsilon_{i,p(i)}^{l,k} = 1
$$

Denote $B_{i,p(i)}^t(l,k) := P(\mathbf{Z}_i^t = l,\, \mathbf{Z}_{p(i)}^t = k \mid \mathbf{X}^t, \boldsymbol{\theta}^{\cancel{(t)}})$.

Using the Lagrangian, we get:

$$
\begin{aligned}
\mathcal{L} &= \sum_{t=1}^{T} \sum_{k=1}^{m} \sum_{l=1}^{m} \sum_{i=2}^{n} B_{i,p(i)}^t(l,k) \log \epsilon_{i,p(i)}^{l,k} + \lambda\Big(\sum_{l=1}^{m} \epsilon_{i,p(i)}^{l,k} - 1\Big) \\
\Rightarrow \frac{\delta \mathcal{L}}{\delta \epsilon_{i,p(i)}^{l,k}} &= \sum_{t=1}^{T} \frac{B_{i,p(i)}^t(l,k)}{\epsilon_{i,p(i)}^{l,k}} = \lambda, \text{ for each } k,\ l \text{ and } i \text{ combination, and} \\
\frac{\delta \mathcal{L}}{\delta \lambda} &= \sum_{l=1}^{m} \epsilon_{i,p(i)}^{l,k} = 1
\end{aligned}
$$

Combining the two, we get that:

$$
\begin{aligned}
\epsilon_{i,p(i)}^{l,k} &= \frac{\sum_{t=1}^{T} B_{i,p(i)}^t(l,k)}{\lambda} \\
\sum_{l=1}^{m} \epsilon_{i,p(i)}^{l,k} &= \sum_{l=1}^{m} \frac{\sum_{t=1}^{T} B_{i,p(i)}^t(l,k)}{\lambda} = 1 \\
\Rightarrow \lambda &= \sum_{l=1}^{m} \sum_{t=1}^{T} B_{i,p(i)}^t(l,k) \\
\therefore {\epsilon_{i,p(i)}^{l,k}}^{(t+1)} &= \frac{\sum_{t=1}^{T} B_{i,p(i)}^t(l,k)}{\sum_{l=1}^{m} \sum_{t=1}^{T} B_{i,p(i)}^t(l,k)} \\
&= \frac{\sum_{t=1}^{T} B_{i,p(i)}^t(l,k)}{\sum_{t=1}^{T} A_{p(i)}^t(k)}
\end{aligned}
$$

These align with the results in Crouse et al. (1998).

For the remaining parameters in the conditional density, these can be solved separately. If it is a gaussian density, for example, the mean and standard deviation parameters are calculated in the same way as in the GMM case - by treating it as an MLE problem applied to a weighted gaussian distribution, with the appropriate weights.

References

M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *IEEE Transactions on signal processing*, 46(4):886–902, 1998.

J. Li. Hidden markov model. *Data Mining, Department of Statistics, Pennsylvania State University*, 2008.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 0262018020, 9780262018029.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

O. Ronen, J. Rohlicek, and M. Ostendorf. Parameter estimation of dependence tree models using the em algorithm. *IEEE Signal Processing Letters*, 2(8):157–159, 1995.