Estimation by the EM algorithm

Yun-Hsiang Chan

June 2021

A commonly used method of finding maximum-likelihood estimates of HMMs is the EM algorithm. The tools we need to do so are the forward and the backward probabilities.

I. Forward and backwward probabilities

Recall the row vector α_t, for $t = 1, 2, ..., T$ as follows:

$$\alpha_t = \delta P(x_1)\Gamma P(x_2)...\Gamma P(x_t) = \delta P(x_1)\Pi_{s=2}^{t}\Gamma P(x_s)$$

with δ denoting the initial distribution of the Markov chain. We have referred to the elements of α_t as **forward probabilities**, but we have not yet justified their description as probabilities.

$\alpha_t(j)$, the jth component of α_t, is indeed a probability, the joint probability $Pr(X_1 = x_1, X_2 = x_2, ..., X_t = x_t, C_t = j)$.

We shall also need the vector of **backward probabilities** β_t which , for $t = 1, 2, ..., T$ is defined by

$$\beta_t' = \Gamma P(x_{t+1})\Gamma P_{x_{t+2}}...\Gamma P(x_T)1' = (\Pi_{s=t+1}^{T}\Gamma P(x_s))1'$$

with the convention that an empty product is the identity matrix; the case $t = T$ therefore yields $\beta_T = 1$.

$\beta_t(j)$, the jth component of β_t, can be identified as the conditional probability $Pr(X_{t+1} = x_{t+1}, ..., X_T = x_T|C_t = j)$.

It will then follow that, for $t = 1, ..., T$

$$\alpha_t(j)\beta_t(j) = Pr(X^{(T)} = x^{(T)}, C_t = j)$$

Proof. Since $\boldsymbol{\alpha}_1 = \boldsymbol{\delta}\mathbf{P}(x_1)$, we have

$$\alpha_1(j) = \delta_j\, p_j(x_1) = \Pr(C_1 = j)\Pr(X_1 = x_1 \mid C_1 = j),$$

hence $\alpha_1(j) = \Pr(X_1 = x_1, C_1 = j)$; that is, the proposition holds for $t = 1$. We now show that, if the proposition holds for some $t \in \mathbb{N}$, then it also holds for $t+1$:

$$\begin{aligned}
\alpha_{t+1}(j) &= \sum_{i=1}^{m} \alpha_t(i)\gamma_{ij}p_j(x_{t+1}) \qquad \text{(see (4.3))} \\
&= \sum_i \Pr(\mathbf{X}^{(t)} = \mathbf{x}^{(t)}, C_t = i)\Pr(C_{t+1} = j \mid C_t = i) \\
&\qquad\qquad \times \Pr(X_{t+1} = x_{t+1} \mid C_{t+1} = j) \\
&= \sum_i \Pr(\mathbf{X}^{(t+1)} = \mathbf{x}^{(t+1)}, C_t = i, C_{t+1} = j) \qquad (4.4) \\
&= \Pr(\mathbf{X}^{(t+1)} = \mathbf{x}^{(t+1)}, C_{t+1} = j),
\end{aligned}$$

as required. The crux is the line numbered (4.4); equation (B.1) provides the justification thereof. □

Figure 1: Proof of Proposition 1

1. Forward probabilities

It follows immediately from the definition of α_t that , for $t = 1, ..., T-1$, $\alpha_{t+1} = \alpha_t \Gamma P(x_{t+1})$ or, in scalar form,

$$\alpha_{t+1}(j) = (\sum_{i=1}^{m} \alpha_t(i)\gamma_{ij})p_j(x+1)$$

Proposition 1
For $t = 1, ..., T$ and $j = 1, ..., m$

$$\alpha_t(j) = Pr(X^{(t)} = x^{(t)}, C_t = j)$$

The proof is in Figure 1.

2. Backward probabilities

It follows immediately from the definition of β_t that $\beta_t' = \Gamma P(x_{t+1})\beta_{t+1}'$, for $t = 1, ..., T-1$

Proposition 2
For $t = 1, ..., T-1$, and $i = 1, 2, ..., m$

$$\beta_t(i) = Pr(X_{t+1} = x_{t+1}, X_{t+2} = x_{t+2}, ..., X_T = x_T | C_t = i)$$

providied that $Pr(C_t = i) > 0$. In a more compact notation,

$$\beta_t(i) = Pr(X_{t+1}^T = x_{t+1}^T | C_t = i)$$

where X_a^b denotes the vector $(X_a, X_{a+1}, ..., X_b)$

This proposition identifies $\beta_t(i)$ as a conditional probability: the probability of the observations being $x_{t+1}, ..., x_T$, given that the Markov chain is in state i at time t.

The entire proof is in the textbook.

3. Properties of forward and backward probabilities

We now establish a result relating the forward and backward probabilities $\alpha_t(i)$ and $\beta_t(i)$ to the probabilities $Pr(X^{(T)} = x^{(T)}, C_t = i)$. This we shall use in applying the EM algorithm to HMMs, and in local decoding.

Proposition 3
For $t = 1, ..., T$ and $i = 1, ..., m$

$$\alpha_t(i)\beta_t(i) = Pr(X^{(T)} = x^{(T)}, C_t = i)$$

and consequently $\alpha_t\beta_t' = Pr(X^{(T)} = x^{(T)} = L_T)$, for each such t.

Proposition 5
Firstly, for $t = 1, ..., T$

$$Pr(C_t = j|X^{(T)} = x^{(T)}) = \alpha_t(j)\beta_t(j)/L_T$$

and secondly, for $t = 2, ..., T$

$$Pr(C_{t-1} = j, C_t = k|X^{(T)} = x^{(T)}) = \alpha_{t-1}(j)\gamma_{jk}p_k(x_t)\beta_t(k)/L_T$$

The EM algorithm