HMM example

Matthew Stephens

University of Chicago

Feburary 5, 2026

See here for a PDF version of this vignette.

Background

This vignette illustrates the computation of the "forwards" and "backwards" probabilities in a Hidden Markov Model (HMM), and their use to infer the (marginal) posterior distribution of the latent state at each location.

Simulate from an HMM

To illustrate we simulate a simple HMM with two states, $Z_t \in \{1, 2\}$, and with the emission distributions in state k being normal with mean k. The transition matrix for the Markov chain is symmetric, with probability 0.9 of staying in the same state, and 0.1 of switching at each step.

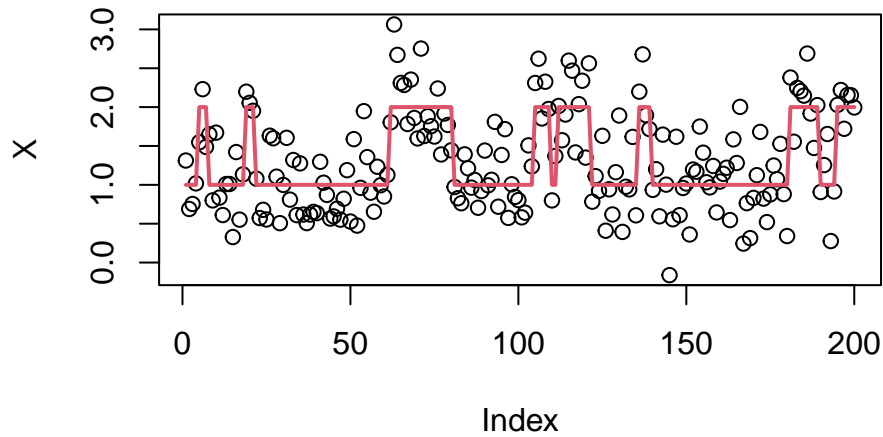Here is some code to simulate from this:

```
set.seed(1)
T = 200
K = 2
sd= 0.4
P = cbind(c(0.9,0.1),c(0.1,0.9))

# Simulate the latent (Hidden) Markov states
Z = rep(0,T)
Z[1] = 1
for(t in 1:(T-1)){
  Z[t+1] = sample(K, size=1, prob=P[Z[t],])
}

# Simulate the emitted/observed values
X= rnorm(T,mean=Z,sd=sd)

plot(X, main="Realization of HMM; latent states shown in red")
lines(Z,col=2,lwd=2)
```

Realization of HMM; latent states shown in red

Compute Forwards Probabilities

We define the forwards probabilities as

$$\alpha_{tk} := p(X_1, \ldots, X_t; Z_n = k) = p(X_1, \ldots, X_t | Z_n = k) Pr(Z_n = k).$$

So

$$\alpha_{1k} = \pi_k p(X_1 | Z_1 = k),$$

where $\pi_k = Pr(Z_1 = k)$. (Here we assume $\pi = (0.5, 0.5)$, which is the stationary distribution of P.)

Further, we can compute α_{t+1} from α_t using the forward algorithm:

$$
\begin{aligned}
\alpha_{(t+1)k} &= \sum_j p(X_1, \ldots, X_t, X_{t+1}, Z_t = j, Z_{t+1} = k) \\
&= \sum_j p(Z_t = j) p(Z_{t+1} = k | Z_t = j) p(X_1, \ldots, X_t, X_{t+1} | Z_t = j, Z_{t+1} = k) \\
&= \sum_j p(Z_t = j) p(Z_{t+1} = k | Z_t = j) p(X_1, \ldots, X_t | Z_t = j, Z_{t+1} = k) p(X_{t+1} | Z_t = j, Z_{t+1} = k, X_1, \ldots, X_t) \\
&= \sum_j p(Z_t = j) p(Z_{t+1} = k | Z_t = j) p(X_1, \ldots, X_t | Z_t = j) p(X_{t+1} | Z_{t+1} = k) \\
&= \sum_j p(X_1, \ldots, X_t, Z_t = j) p(Z_{t+1} = k | Z_t = j) p(X_{t+1} | Z_{t+1} = k) \\
&= \sum_j \alpha_{tj} P_{jk} \, p(X_{t+1} | Z_{t+1} = k) \\
&= (\alpha_t.P)_k \, p(X_{t+1} | Z_{t+1} = k).
\end{aligned}
$$

We code this as follows:

```
# this is the function Pr(X_t | Z_t=k) for our example
emit = function(k,x){
```

```
    dnorm(x,mean=k,sd=sd)
}

pi = c(0.5,0.5) #Assumed prior distribution on Z_1

alpha = matrix(nrow = T,ncol=K)

# Initialize alpha[1,]
for(k in 1:K){
  alpha[1,k] = pi[k] * emit(k,X[1])
}

# Forward algorithm
for(t in 1:(T-1)){
  m = alpha[t,] %*% P
  for(k in 1:K){
    alpha[t+1,k] = m[k]*emit(k,X[t+1])
  }
}
```

Likelihood calculation

Note that the forwards algorithm also allows us to compute the likelihood, $p(X_1, \ldots, X_T)$. Indeed, by definition of α we have

$$p(X_1, \ldots, X_T) = \sum_k \alpha_{Tk}.$$

So the likelihood is:

```
sum(alpha[T,])
# [1] 1.535e-65
```

Notice that these alpha numbers can get very small! This can cause numerical issues if T were larger and we should really be more careful to avoid this! A common strategy is to "renormalize" the αs at each iteration: that is, for each t divide the α_{tk} by $\sum_k \alpha_{tk}$, and then store the value of this sum separately. Maybe you can work out the details!

Compute Backwards Probabilities

We define the backwards probabilities as

$$\beta_{tk} := p(X_{t+1}, \ldots, X_T | Z_t = k).$$

with the "boundary condition" $\beta_{Tk} = 1$.

By this definition

$$\beta_{tk} = \sum_j Pr(X_{t+1}, \ldots, X_L, Z_{t+1} = j | Z_t = k)$$

$$= \sum_j Pr(X_{t+2}, \ldots, X_L, Z_{t+1} = j | Z_t = k) Pr(X_{t+1} | Z_t = k, X_{t+2}, \ldots, X_L, Z_{t+1} = j)$$

$$= \sum_j Pr(X_{t+2}, \ldots, X_L, Z_{t+1} = j | Z_t = k) Pr(X_{t+1} | Z_{t+1} = j)$$

$$= \sum_j Pr(X_{t+2}, \ldots, X_L | Z_t = k, Z_{t+1} = j) P(Z_{t+1} = j | Z_t = k) Pr(X_{t+1} | Z_{t+1} = j)$$

$$= \sum_j Pr(X_{t+2}, \ldots, X_L | Z_{t+1} = j) P(Z_{t+1} = j | Z_t = k) Pr(X_{t+1} | Z_{t+1} = j)$$

$$= \sum_j \beta_{(t+1)j} \, P_{kj} \, p(X_{t+1} | Z_{t+1} = j).$$

Here is code to compute these values iteratively:

```
beta = matrix(nrow = T,ncol=K)

# Initialize beta
for(k in 1:K){
  beta[T,k] = 1
}

# Backwards algorithm
for(t in (T-1):1){
  for(k in 1:K){
    beta[t,k] = sum(beta[t+1,]*P[k,]*emit(1:K,X[t+1]))
  }
}
```

Posterior distribution of each state

We are now in a position to compute the posterior distribution on each state.

By the definitions of α and β we have:

$$p(X_1, \ldots, X_T; Z_t = k) = p(X_1, \ldots, X_t; Z_n = k) p(X_{t+1}, \ldots, X_T | X_1, \ldots, X_t, Z_n = k)$$
$$= p(X_1, \ldots, X_t; Z_n = k) p(X_{t+1}, \ldots, X_T | Z_t = k)$$
$$= \alpha_{tk} \, \beta_{tk}.$$

Thus we can now compute the posterior distribution for each Z_t given the data $X_1, \ldots, X_T$:

$$\Pr(Z_t = k | X_1, \ldots, X_T) = \alpha_{tk} \beta_{tk} / \sum_k \alpha_{tk} \beta_{tk}$$

Here we compute this and plot the posterior on top of the "truth" that we simulated

```r
ab = alpha*beta
prob = ab/rowSums(ab)
plot(prob[,2],type="l",ylim=c(0,1), main="Posterior probability of state 2 (black);\n Ind(true
lines(Z==2,col=2,lwd=2)
```

**Posterior probability of state 2 (black);
Ind(true state is 2) superposed (red)**