

HMM II: Some theory for HMMs

Introduction

In the last lecture, we derived some of the basic properties of a Markov chain. Here we generalize the models to allow a sequence of hidden states evolving according to a Markov chain, together with a set of observable emitted symbols. As earlier, the Markov chain has state space \mathcal{S} , and the alphabet of emitted symbols is denoted by \mathcal{A} .

The unobservable (or hidden) sequence of states of length n is denoted by $\mathbf{X} = (X_1, X_2, \dots, X_n)$. The observed sequence of symbols is denoted by $\mathbf{A} = (A_1, A_2, \dots, A_n)$. The rules of evolution are

- $X_t, t = 1, 2, \dots$ evolves as a Markov chain with transition matrix $P = (p(i, j), i, j \in \mathcal{S})$, and X_1 has distribution $\pi(i), i \in \mathcal{S}$.
- The probability that state x emits letter a is $e(x, a)$.
- Conditional on \mathbf{X} , the emitted symbols A_1, A_2, \dots, A_n are independent, with

$$\mathbb{P}(A_j = a_j, j = 1, \dots, n | \mathbf{X} = (x_1, \dots, x_n)) = \prod_{j=1}^n e(x_j, a_j).$$

Example. A simple model for generating DNA sequences has two hidden states; $\mathcal{S} = \{1, 2\}$. The transition matrix is

$$P = \begin{pmatrix} 0.5 & 0.5 \\ 0.75 & 0.25 \end{pmatrix}$$

The alphabet is $\mathcal{A} = \{A, C, G, T\}$ and the emission laws are $(0.25, 0.25, 0.25, 0.25)$ from state 1 and $(0.1, 0.1, 0.1, 0.7)$ from state 2. You should think of how you can simulate such a process.

If you were given the sequence $\mathbf{A} = AATTTTTCGCGTTGG$ with $n = 15$, there are at least three important questions you could ask:

1. *Likelihood:* What is $\mathbb{P}(\mathbf{A})$ as a function of the model parameters?
2. *Decoding:* What is the most likely sequence of states \mathbf{X} that gave rise to \mathbf{A} ?

3. *Estimation:* How can the underlying parameters of the chain be estimated? This is sometimes called *learning* in the computer science literature.

Likelihoods. Let $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{a} = (a_1, \dots, a_n)$. Our rules show that

$$\begin{aligned} \mathbb{P}(\mathbf{X} = \mathbf{x}, \mathbf{A} = \mathbf{a}) &= \mathbb{P}(\mathbf{X} = \mathbf{x})\mathbb{P}(\mathbf{A} = \mathbf{a}|\mathbf{X} = \mathbf{x}) \\ &= \pi(x_1) \prod_{j=2}^n p(x_{j-1}, x_j) \cdot \prod_{j=1}^n e(x_j; a_j). \end{aligned} \quad (1)$$

From this we can calculate

$$\mathbb{P}(\mathbf{A} = \mathbf{a}) = \sum_{\mathbf{x}} \mathbb{P}(\mathbf{X} = \mathbf{x}, \mathbf{A} = \mathbf{a}), \quad (2)$$

the sum being taken over all $|\mathcal{S}|^n$ possible patterns. Thus for our example with $|\mathcal{S}| = 2$ and $n = 1000$, we have a sum with $2^{1000} \approx 10^{301}$ states in it, so the calculation in (2) takes about $2 \cdot 1000 \cdot 10^{301}$ operations. We obviously need something more clever than this!

The forward algorithm. Write

$$\alpha_t(i) = \mathbb{P}(X_t = i, A_1 = a_1, \dots, A_t = a_t),$$

and initialize with

$$\begin{aligned} \alpha_1(i) &= \mathbb{P}(X_1 = i, A_1 = a_1) \\ &= \pi(i)e(i, a_1). \end{aligned}$$

We then note that

$$\begin{aligned} \alpha_{t+1}(j) &= \mathbb{P}(X_{t+1} = j, A_1 = a_1, \dots, A_t = a_t, A_{t+1} = a_{t+1}) \\ &= \sum_l \mathbb{P}(X_{t+1} = j, X_t = l, A_1 = a_1, \dots, A_t = a_t, A_{t+1} = a_{t+1}) \\ &= \sum_l \mathbb{P}(X_{t+1} = j, A_{t+1} = a_{t+1} \mid X_t = l, A_t = a_t, A_1 = a_1, \dots, A_{t-1} = a_{t-1}) \\ &\quad \times \mathbb{P}(X_t = l, A_t = a_t, A_1 = a_1, \dots, A_{t-1} = a_{t-1}) \\ &= \sum_l \mathbb{P}(X_{t+1} = j, A_{t+1} = a_{t+1} \mid X_t = l) \\ &\quad \times \mathbb{P}(X_t = l, A_t = a_t, A_1 = a_1, \dots, A_{t-1} = a_{t-1}) \\ &= \sum_l p(l, j)e(j, a_{t+1})\alpha_t(l) \\ &= e(j, a_{t+1}) \sum_l \alpha_t(l)p(l, j), \end{aligned} \quad (3)$$

for $j \in \mathcal{S}, 1 \leq t \leq n - 1$. To finish the algorithm, note that

$$\begin{aligned} \mathbb{P}(A_1 = a_1, \dots, A_n = a_n) &= \sum_{l \in \mathcal{S}} \mathbb{P}(X_n = l, A_1 = a_1, \dots, A_n = a_n) \\ &= \sum_{l \in \mathcal{S}} \alpha_n(l). \end{aligned} \quad (4)$$

How many calculations is this? It is of order $|\mathcal{S}|^2 n$, which translates to 4,000 for our example (in contrast to 2×10^{304} !)

Which states are most likely? In order to answer this question, we need to derive another algorithm for computing probabilities in HMMs.

The backward algorithm. Write

$$\beta_t(i) = \mathbb{P}(A_{t+1} = a_{t+1}, \dots, A_n = a_n \mid X_t = i),$$

with

$$\beta_n(i) = 1 \text{ for all } i \in \mathcal{S}.$$

The backward algorithm is derived as follows:

$$\begin{aligned} \beta_t(j) &= \mathbb{P}(A_{t+1} = a_{t+1}, \dots, A_n = a_n \mid X_t = j) \\ &= \sum_l \mathbb{P}(A_{t+1}, X_{t+1} = l, A_{t+2}, \dots, A_n \mid X_t = j) \\ &= \sum_l \mathbb{P}(A_{t+2}, \dots, A_n \mid A_{t+1}, X_{t+1} = l, X_t = j) \\ &\quad \times \mathbb{P}(A_{t+1}, X_{t+1} = l \mid X_t = j) \\ &= \sum_l \mathbb{P}(A_{t+2}, \dots, A_n \mid X_{t+1} = l) \mathbb{P}(A_{t+1} \mid X_{t+1} = l) p(j, l) \\ &= \sum_l \beta_{t+1}(l) e(l, a_{t+1}) p(j, l), \end{aligned} \quad (5)$$

for $t = n - 1, n - 2, \dots, 1; j \in \mathcal{S}$. Again, this is of order $n|\mathcal{S}|^2$ operations.

There are several answers to the question ‘Which states are most likely?’ For example, define

$$\gamma_t(i) = \mathbb{P}(X_t = i \mid A_1, \dots, A_n).$$

We can write this in terms of $\alpha_t(i)$ and $\beta_t(i)$, since

$$\begin{aligned} \gamma_t(i) &= \mathbb{P}(X_t = i \mid A_1, \dots, A_n) \\ &\propto \mathbb{P}(X_t = i, A_1, \dots, A_n) \\ &= \mathbb{P}(A_{t+1}, \dots, A_n \mid X_t = i, A_1, \dots, A_t) \mathbb{P}(X_t = i, A_1, \dots, A_t) \\ &= \alpha_t(i) \beta_t(i). \end{aligned}$$

It follows that

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_l \alpha_t(l)\beta_t(l)}. \quad (6)$$

We could use this to reconstruct the state sequence via

$$\hat{X}_t = \arg \max_{j \in \mathcal{S}} \gamma_t(j). \quad (7)$$

While this maximizes the expected number of correct states, the reconstructed process might in fact be impossible. Remember that this is not a joint reconstruction.

The Viterbi algorithm The Viterbi algorithm provides a way to find the most likely state sequence, $\arg \max_{\mathbf{x}} \mathbb{P}(\mathbf{X} = \mathbf{x}, \mathbf{A} = \mathbf{a})$. To do this, define

$$\delta_t(i) = \max_{x_1, \dots, x_{t-1}} \mathbb{P}(X_1 = x_1, \dots, X_{t-1} = x_{t-1}, X_t = i, A_1, \dots, A_t).$$

Then

$$\begin{aligned} \delta_{t+1}(j) &= \max_{x_1, \dots, x_t} \mathbb{P}(X_1 = x_1, \dots, X_{t-1} = x_{t-1}, X_t = x_t, X_{t+1} = j, A_1, \dots, A_{t+1}) \\ &= \max_{x_1, \dots, x_t} \mathbb{P}(X_{t+1} = j, A_{t+1} \mid X_t = x_t, A_t, X_{t-1}, A_{t-1}, \dots, X_1, A_1) \\ &\quad \times \mathbb{P}(X_1 = x_1, \dots, X_{t-1} = x_{t-1}, X_t = x_t, A_1, \dots, A_t) \\ &= \max_{x_t} p(x_t, j) e(j, A_{t+1}) \max_{x_1, \dots, x_{t-1}} \mathbb{P}(X_1 = x_1, \dots, A_t) \\ &= e(j, A_{t+1}) \max_i p(i, j) \delta_t(i). \end{aligned} \quad (8)$$

To produce the algorithm, we need to keep track of the argument that maximizes the term on the right of (8) for each t and j . Here is the final algorithm:

Initialize: Set

$$\delta_1(i) = \mathbb{P}(X_1 = i, A_1) = \pi(i)e(i, A_1), \quad \psi_1(i) = 0, i \in \mathcal{S}.$$

Recursion: For $2 \leq t \leq n, j \in \mathcal{S}$,

$$\begin{aligned} \delta_t(j) &= e(j, A_t) \max_{i \in \mathcal{S}} \delta_{t-1}(i) p(i, j) \\ \psi_t(j) &= \arg \max_{i \in \mathcal{S}} \delta_{t-1}(i) p(i, j) \end{aligned}$$

Termination:

$$P^* = \max_{i \in \mathcal{S}} \delta_n(i), \quad x_n^* = \arg \max_{i \in \mathcal{S}} \delta_n(i)$$

Path backtracking:

$$x_t^* = \psi_{t+1}(x_{t+1}^*), \quad t = n - 1, n - 2, \dots, 1.$$

References

- Baldi P, Brunak S (1998) *Bioinformatics. The machine learning approach*. MIT Press, Chapter 7.
- Durbin J, Eddy S, Krogh A, Mitchison G (1998) *Biological Sequence Analysis*. Cambridge University Press.
- Koski T (2001) Hidden Markov models for bioinformatics. Kluwer Academic Publishers.
- Mount DW (2001) *Bioinformatics. Sequence and genome analysis*. Cold Spring Harbor Laboratory Press.
- Rabiner LR (1989) A tutorial on hidden Markov chains and selected applications in speech recognition. *Proceedings of the IEEE* **77**: 257–286.
- MacDonald IL, Zucchini W (1997) *Hidden Markov and other models for discrete-valued time series*. Chapman and Hall.