18 Hidden Markov Models

In Markov models, we directly observe the state of the process (for example, in a random walk, the state is completely described by the position of the random walker). We can easily imagine a system where we don't directly observe the state but observe some outcome which depends on the state. Call the observed outcomes *symbols*. We thus make a distinction between the **sequence of states** and the **sequence of symbols**. We formally model these systems as hidden Markov models (HMMs).

Example: Weather. Suppose we live in a place that gets winds either from the south or north. On days the wind is from the south, it is rainy and cold 75% of the days, and warm and sunny the other 25% of the days. When the wind is from the north, is is rainy and cold 20% of days, and warm and sunny 80% of the days. If it is northerly today, it will be northerly tomorrow with probability 0.7. If it is southerly today it will be south with prob. 0.6. □

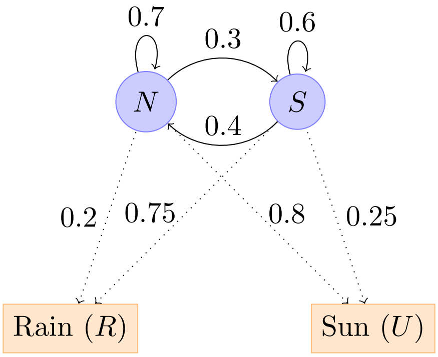

Figure 5: The chance of Sunny or Rainy weather on a day is determined by the wind direction. When it is Northerly, Sun is more likely, when it is Southerly, Rain is more likely.

The idea in the HMM is that we only observe the sequence of symbols but to understand what is going on, we need to know what the underlying state is.

Example cont.: A state sequence for the weather example might be $NNNSSNS$ while the emissions sequence looks like (using R for rain and U for sun) $UUUURUR$. We might observe the weather (rainy or sunny) and wonder what is causing the patterns we see. The pattern is best understood by knowing the sequence of states (direction of wind). □

Notation for HMMs

The ith state of the state sequence is π_i and transitions (of the states) is given by $a_{kl} = P(\pi_i = l|\pi_{i-1} = k)$. We use b's to represent the symbols so we get a new set of parameters called emission probabilities, $e_\pi(b) = P(x_i = b|\pi_i = \pi)$ is the prob of emitting symbol b given that we are in state π.

Example cont.: In the weather example $a_{NN} = 0.7$, $a_{NS} = 0.3$, $a_{SS} = 0.6$ and $a_{NS} = 0.4$. The emission probabilities are $e_N(R) = 0.2$, $e_N(U) = 0.8$, $e_S(R) = 0.75$ and $e_S(U) = 0.25$. □

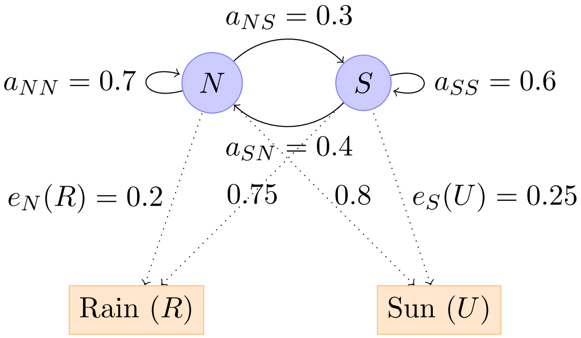

Figure 6: The chance of Sunny or Rainy weather on a day is determined by the wind direction. When it is Northerly, Sun is more likely, when it is Southerly, Rain is more likely.

Example: Cheating Casino. A casino tips things in its favour at a dice table by occasionally switching from a fair die to a loaded one. The loaded die produces a 6 50% of the time, and each of 1-5 just 10% of the time. Call the fair die F and the loaded die L. We have transitions $a_{FL} = 0.05$ and $a_{LF} = 0.1$. $a_{FF} = 1 - a_{FL}$ and $a_{LL} = 1 - a_{LF}$. Emissions are $e_F(b) = 1/6$ for $b = 1, \dots, 6$ while $e_L(b) = 1/10$ for $b = 1, \dots, 5$ and $e_L(6) = 0.5$.

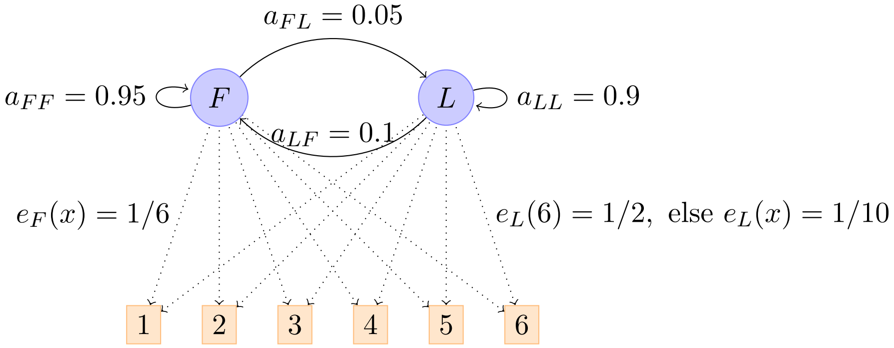

Figure 7:

A run of the chain might look like:

```
F F F F L F F F L L L L L L L L F F F F F F F F F F F F F F L L L L F F F
```

```
F L L L L L L F F F F F F F F
```

producing rolls

```
2 5 1 6 6 2 4 2 4 5 6 6 3 6 2 4 2 2 3 4 6 3 6 5 3 4 5 2 3 5 1 6 6 6 6 2 4
6 6 2 6 6 6 6 6 1 5 1 6 4 1 2.
```
□

Think of HMMs as a generative process where the first state π_1 is chosen according to probability a_{0i}. From that state, a symbol is emitted according to the probabilities e_{π_1}. State 2 is then chosen according to transition probs $a_{\pi_1 i}$ which emits a symbol and so on.

The joint probability of states and symbols is then,

$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

18.1 The Viterbi algorithm for finding the most probable state path

Multiple underlying states could produce the same path. Clearly, though, some states are more likely than others given the emissions. If we see lots of very good weather, for example, we may guess that northerlies are predominating in our city. If we see long strings of 6s, we might guess that the casino has switched to the loaded die, while sequences of apparently evenly distributed throws suggest a fair die is being used.

If we want to work with just one state path, it can be argued that the state path with the highest probability is the one to choose. Let

$$\pi^* = \arg\max_{\pi} P(x, \pi).$$

We can find π^* using a recursive algorithm due to Andrew Viterbi (an Jewish Italian refugee from the fascists who grew up in the USA, wrote this algorithm while at MIT and then started Qualcomm, a now huge technology company).

Let $v_k(i)$ be the probability of the most probable state path ending in state k at observation i (that is, considering only the first i observations). Suppose $v_k(i)$ is known for all states k. Then the next probability $v_l(i+1)$ can be calculated for observation x_{i+1} as

$$v_l(i+1) = e_l(x_{i+1}) \max_k (v_k(i) a_{kl}).$$

If we insist that all sequences start in a begin state called 0, we have $v_0(0) = 1$ and we can use a backtracking procedure like we saw in the pairwise alignment algorithm to find the most probable path.

The Viterbi algorithm thus proceeds as follows:

Initialisation $i = 0$: $v_0(0) = 1$, $v_k(0) = 0$ for $k > 0$.

Recursion $i = 1, \ldots, L$: $v_l(i) = e_l(x_i) \max_k (v_k(i-1) a_{kl})$.
$\quad\quad Pointer_i(l) = \arg\max_k (v_k(i-1) a_{kl})$

Termination: $P(x, \pi^*) = \max_k(v_k(L)a_{k0})$. $\pi^*_L = \arg\max_k(v_k(L)a_{k0})$.

Traceback $i = L, \dots, 1$**:** $\pi^*_{i-1} = Pointer_i(\pi^*_i)$.

Here we assume an end state is modeled, as a_{k0}. If the end is not modeled, this term disappears.

Even the most probable path typically has a very low likelihood, so it is important to work in log units. The recursion in log units is

$$V_l(i+1) = \log(e_l(x_{i+1})) + \max_k(V_k(i) + \log(a_{kl}))$$

where $V_k(i) = \log(v_k(i))$.