COMP 182: Algorithmic Thinking
Markov Chains and Hidden Markov Models

Luay Nakhleh

1 Independence First

Consider the following experiment: A fair coin is tossed $10,000$ times, where each toss is independent of all the tosses the precede it. Let $X = X_1X_2 \ldots X_{10,000}$ be the sequence of the sides that show in one run of this experiment, such that $X_i \in \{H, T\}$ for $1 \leq i \leq 10,000$. One question of interest in this case is: what is $p(X)$? Notice that this experiment can be viewed as 10,000 independent Bernoulli trials with success probability 0.5. Making use of the independence assumption, we have

$$p(X) = \prod_{i=1}^{10,000} p(X_i).$$

Making use of the fact that the success probability is 0.5, we have

$$p(X) = \prod_{i=1}^{10,000} 0.5 = (0.5)^{10,000}.$$

A very important practical issue. Notice that in computing $p(x)$, we are raising a number between 0 and 1 to a very big power. Whatever representation is used in your programming language of choice, underflow is a very likely outcome here, and it wouldn't be surprising if get answer 0.0 for $p(X)$ above. A solution for this problem is to work with $\log$ (base 2, for example) probabilities. So, if we take the log of both sides above, we get

$$\log p(X) = \log((0.5)^{10,000}) = 10,000 \cdot \log 0.5 = -10,000.$$

In this case, your computer is computing a summation, and underflow does not occur.

You might be asking now: But, if I'm interested in $p(X)$ itself, I'd still need to compute $2^{-10,000}$, and we're back to the same problem! You're right about that, but there are two issues here:

- If we directly compute $(0.5)^{10,000}$, we would not be able to report results. If, on the other hand, we compute $10,000 \cdot \log 0.5$, we'd get $-10,000$, and report the answer as "It is 2 to the power $-10,000$." (If you're not convinced by this example, because it's too trivial, imagine the product is taken over $10,000$ numbers between 0 and 1, no two of which are equal.)
- More importantly, in real applications, we often compare quantities (e.g., if we are doing hill-climbing to find the optimal value of a parameter and in every step we evaluate a function like $p(X)$). That is, we're interested in knowing whether $p(X) < p(X')$. In this case, working with $\log$ probabilities would be equivalent to working with the probabilities themselves, yet without running into underflow situations.

2 Dependence and No Hidden States: Markov Chains

Let us now consider a "strange" coin that tends to come up H most often in the very toss (a larger number of experiments were done and it was found that 90% of the time, the first toss in each experiment was an H), and whenever a toss is an H, the probability that the next toss shows H is 0.6, and whenever a toss is a T, the probability that the next toss shows H is 0.3. In this case, we cannot treat the tosses as independent, since each toss depends on the toss

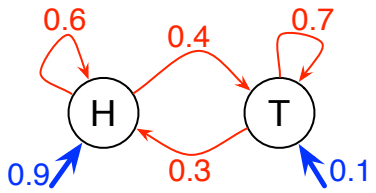

Figure 1: A Markov chain with two states.

the immediately precedes it. We can describe the probability distribution on sequences generated by this coin with the diagram shown in Fig. 1, which is what we call a *Markov chain.*

Consider now an experiment where this strange coin is tossed 5 times giving rise to the sequences $X = THHHT$. What is $p(X)$ given this coin? To compute this probability, we have to account of two types of probabilities:

- The probability that each of the given letters is preceded by the letter to its left in the sequence?
- Since the first letter has no other letter to its left, what is the probability that the sequence has T as the first letter?

The first set of probabilities come from the *transition probabilities* of the Markov chain: These are the red numbers in Fig. 1. The second set of probabilities come from the *initial probabilities* of the Markov chain: These are the blue numbers in Fig. 1. Using the values in Fig. 1, we have

$$p(X) = p(T)p(H|T)p(H|H)p(H|H)p(T|H) = 0.1 \times 0.3 \times 0.6 \times 0.6 \times 0.4.$$

We can formalize Markov chains as follows.

Definition 1 *A Markov chain M is a tuple (Q, A, π) where $Q = \{1, \ldots, K\}$ is a set of states, $A_{K\times K}$ is a stochastic matrix (that is, the sum of all elements in each row is 1), and π_K is a stochastic vector (that is, the sum of its elements is 1).*

Given a sequence of observations $X = X_1 X_2 \ldots X_L$, where $X_i \in \{1, 2, \ldots, K\}$, the probability of X under Markov model M is

$$p(X) = \pi_{X_1} \prod_{i=1}^{L-1} A[X_i, X_{i+1}].$$

In other words, π_j denotes the probability of starting in state j, and $A[j, \ell]$ denotes the probability that an observation is ℓ given that the observation that immediately precedes it is j. Sometimes we write $A[j, \ell]$ as $A_{j,\ell}$.

For the example in Fig. 1, imagine that H corresponds to state 1 and T corresponds to state 2. Then, we have

$$\pi_1 = 0.9 \quad \text{and} \quad \pi_2 = 0.1$$

and

$$A[1,1] = 0.6 \quad A[1,2] = 0.4 \quad A[2,1] = 0.3 \quad A[2,2] = 0.7.$$

It is sometimes convenient to add a start state, s, and assume that the Markov chain always starts in that state. In this case, we transfer the π values to the transitions from s to each of the K states:

$$A[\text{“}s\text{”}, i] = \pi_i \quad \forall i \in \{1, 2, \ldots, K\}.$$

Notice that by adding this start state we do not actually add any more information nor do we change the probability distribution.

Furthermore, in some applications, it is important that we add an end state, e, and assume the Markov chain must always end in that state. In this case, we have entries $A[i, \text{“}e\text{”}]$ in the stochastic matrix, for every state $i \in \{1, 2, \ldots, K\}$. For example, see Fig. 2 for a slightly modified Markov chain from Fig. 1.

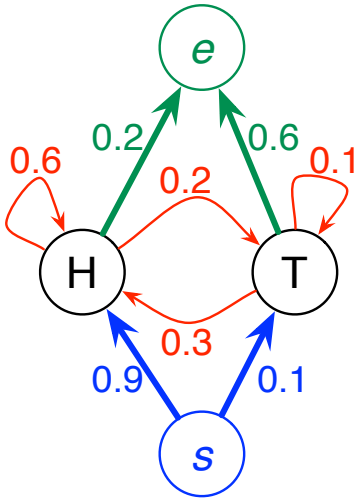

Figure 2: A Markov chain with two states and two explicit start and end states.

In this case, given a sequence of observations $X = X_1 X_2 \dots X_L$, and assuming that s and e are not explicitly in the sequence (that is, $X_1 \neq s$ and $X_2 \neq e$), then

$$p(X) = A[\text{“}s\text{”}, X_1] \cdot \left(\prod_{i=1}^{L-1} A[X_i, X_{i+1}] \right) \cdot A[X_L, \text{“}e\text{”}].$$

Suppose that $A[T, \text{“}e\text{”}] = 0$ for the Markov chain in Fig. 2, then the probability of a sequence that ends with a T is 0. For example, in this case, $p(HHT) = 0$. Similarly for the start state. If, say, $A[\text{“}s\text{”}, H] = 0$, then the sequence cannot start with an H, and we'd have, for example, $p(HHH) = 0$.

3 Dependence and Hidden States: Hidden Markov Models

Imagine now a different type of experiments. Suppose we have two coins, C1 and C2. Coin C1 is fair; that is, $p(H) = p(T) = 0.5$. Coin C2 is biased (or, loaded), with $p(H) = 0.8$ and $p(T) = 0.2$. In the experiment, we choose one of the two coins (we choose coin C1 with probability 0.3 and coin C2 with probability 0.7), and toss it. Then, we repeat the following procedure 99 times (recording the outcomes of the coin tosses, but not the coin selections):

- If the last coin tossed was C1, choose for the next toss C1 with probability 0.4 and C2 with probability 0.6, and toss it.
- If the last coin tossed was C2, choose for the next toss C1 with probability 0.5 and C2 with probability 0.5, and toss it.

Notice that the sequence of the coin choices (e.g., the sequence, C1C1C1C2C2C1), without tossing them, is generated by a Markov chain, shown in Fig. 3. Therefore, if we label the sequence of coin choices as $Z = Z_1 Z_2 \dots Z_L$, where $Z_i \in \{C1, C2\}$ for $1 \leq i \leq L$, then the probability of this sequence is given by

$$p(Z) = \pi_{Z_1} \prod_{i=1}^{L-1} A[Z_i, Z_{i+1}].$$

However, based on the experiment we just described, what we observe are not the sequence of coins that were chosen, but rather that the outcomes of the tosses of the chosen coins. For example, we might observe the sequence $X =$

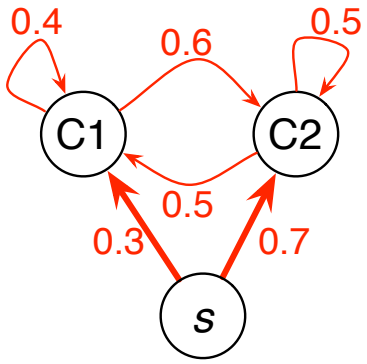

Figure 3: A Markov chain that captures a sequence of coin choices from a set of two coins $\{C1, C2\}$.

$HHTHHT$. If we knew the sequence of coin choices, e.g., Z=C1C1C1C2C2C1, we would be able to compute the joint probability of X and Z by

$$p(X, Z) = p(Z)p(X|Z) = \left(\pi_{Z_1} \prod_{i=1}^{L-1} A[Z_i, Z_{i+1}]\right)\left(\prod_{i=1}^{L} E_{Z_i}(X_i)\right). \quad (1)$$

Here, we use $E_{Z_i}(X_i)$ to denote the probability that coin Z_i generates X_i when tossed. For example, based on our description above, $E_{C1}(H) = E_{C1}(T) = 0.5$, $E_{C2}(H) = 0.8$, and $E_{C2}(T) = 0.2$.

Notice that, unlike in the case of Markov chains, we here make a distinction between the states (C1 and C2 in our example) and the alphabet symbols (H and T in our example).[1] Therefore, to complete the description of the model, we have to also describe the E probabilities; see Fig. 4.

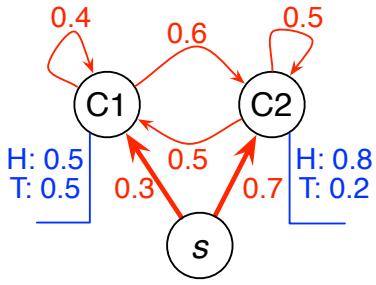

Figure 4: A hidden Markov model that corresponds to the experiment with two coins $\{C1, C2\}$. The transition probabilities are shown in red, and the emission probabilities are shown in blue.

We can now define hidden Markov models (HMMs) formally.

Definition 2 *A hidden Markov model, or HMM, M is a tuple (Q, Σ, A, E, π) where $Q = \{1, \ldots, K\}$ is a set of states, Σ is an alphabet, $A_{K\times K}$ is a stochastic matrix (that is, the sum of all elements in each row is 1), $E_{K\times|\Sigma|}$ is a stochastic matrix, and π_K is a stochastic vector (that is, the sum of its elements is 1).*

In this definition, $A[j, \ell]$ is, as before, the probability that the HMM transitions to state ℓ from state j, $E_j(\sigma)$ is the probability that the HMM emits alphabet symbol σ from state j, and π is as defined for Markov chains. So, for the example of Fig. 4, we have:

[1] In the case of Markov chains, there is a 1-1 correspondence between the states and the alphabet letters, so we usually call each state by the (only) letter it corresponds to. This is why in Fig. 1, for example, we called the states H and T; it would not have made sense to call them after the coin being tossed, since there is a single coin being tossed.

- $Q = \{\text{“}s\text{”}, C1, C2\}$;
- $\Sigma = \{H, T\}$;
- $A[\text{“}s\text{”}, C1] = 0.3$, $A[\text{“}s\text{”}, C2] = 0.7$, $A[C1, C1] = 0.4$, $A[C1, C2] = 0.6$, $A[C2, C1] = 0.5$, and $A[C2, C2] = 0.5$;
- $E_{C1}(H) = E_{C1}(T) = 0.5$, $E_{C2}(H) = 0.8$, and $E_{C2}(T) = 0.2$; and,
- $\pi_{\text{“}s\text{”}} = 1$.

Eq. (1) above gives the joint probability of a sequence of observations X and a (known) path of states Z. If the HMM also has an explicit “e” state (end), then the equation must account of that as well.

A major question of interest in the context of HMMs is: Given a sequence X of observations and an HMM M, what is the path of states Z that gives rise to X with the highest probability. More precisely, we are interested in

$$Z^* \leftarrow \text{argmax}_Z p(X, Z),$$

where Z is taken over all possible paths of states. As we saw in class, Z^* can be computed efficiently using Algorithm **Viterbi**.

Algorithm 1: Viterbi.

Input: A first-order HMM M with states $Q = \{1, 2, \ldots, K\}$ given by its transition matrix A, emission probability matrix E (alphabet Σ), and probability distribution π on the (initial) states; a sequence X of length L (indexed from 0 to $L-1$).

Output: A sequence Z, with $|Z| = |X|$, that maximizes $p(Z, X)$.

$v[\ell, 0] \leftarrow (\pi_\ell \cdot E_\ell(X_0))$ for every $\ell \in Q$;
for $i \leftarrow 1$ **to** $L-1$ **do**
  foreach $\ell \in Q$ **do**
    $v[\ell, i] \leftarrow E_\ell(X_i) \cdot \max_{\ell' \in Q}(v[\ell', i-1] \cdot A_{\ell', \ell})$;
    $bp[\ell, i] \leftarrow \text{argmax}_{\ell' \in Q}(v[\ell', i-1] \cdot A_{\ell', \ell})$;
$Z_{L-1} \leftarrow \text{argmax}_{\ell' \in Q} v[\ell', L-1]$;
for $i \leftarrow L-2$ **downto** 0 **do**
  $Z_i \leftarrow bp[Z_{i+1}, i+1]$;
return Z

The line in blue must be replaced by

$$Z_{L-1} \leftarrow \text{argmax}_{\ell' \in Q}(v[\ell', L-1] \cdot A_{\ell', \text{“}e\text{”}})$$

if an end state “e” exists.