Chapter 7

HMMs

In this chapter I treat *Hidden Markov Models* (HMMs). These are intimately associated with n-gram models and widely used as a computational model for language processing.

7.1 Markov Chains

To understand Hidden Markov Models, we must first understand *Markov chains*. These are basically deterministic finite state automata with associated probabilities. Each arc is associated with a probability value and all arcs leaving any particular node must exhibit a probability distribution, i.e. their values must total 1. In addition, one node is designated as the "starting" node. A simple example is given in figure 7.1.

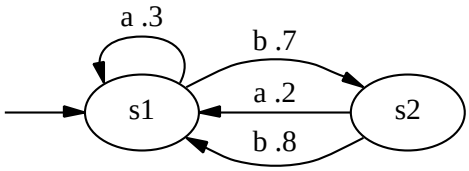

Figure 7.1: A simple Markov chain

As with a FSA, the machine moves from state to state following the arcs given. The sequence of arc symbols denotes the string generated—or accepted—by the machine. The difference between a Markov chain and a FSA is that in the former case there are probabilities associated with each arc. These probabilities are multiplied together to produce the probability that the machine might follow any particular sequence of arcs/states. For example, the probability of producing a single a and returning to s_1 is .3; the probability of going from s_1 to s_2 and emitting a b is .7. Hence the probability of ab is $.3 \times .7 = .14$. The probability of producing the sequence ba, however, is $.7 \times .2 = .14$. In the first case we go from s_1 to s_1 to s_2; in the second case from s_1 to s_2 to s_1.

There are several key facts to note about a Markov chain. First, as we stated above, the probabilities associated with the arcs from any state exhibit a probability distribution. Second, Markov chains are analogous to a *deterministic* finite state automaton: there are no choices at any point either in terms of start state or in terms of what arcs to follow. Thus there is precisely one and only one arc labelled with each symbol in the alphabet from each state in the chain. Third, it follows that any symbol string uniquely determines a state sequence. That is, for any particular string, there is one and only one corresponding sequence of states through the chain.

7.2 Hidden Markov Models

It's also possible to imagine a *non-deterministic Markov chain*; these are referred to as *Hidden Markov Models* (HMMs). Once we've introduced indeterminacy anywhere in the model, we can't uniquely identify a state sequence for all strings. Hence the state sequence is "hidden". Given the model above, we can introduce indeterminacy in several ways. First, we can allow for multiple start states.

The example in figure 7.2 is an example of this sort. Here each state is associated with a "start" probability. (Those must, of course, exhibit a probability distribution and sum to 1.) This means, that for any particular string, one must factor in all possible start probabilities. For example, a string a could be generated/accepted by starting in s_1 and then following the arc back to s_1 ($.4 \times .3 = .12$). We could also start in s_2 and then follow the arc back to s_1 ($.6 \times .2 = .12$).[1]

[1] The fact that these two paths have the same overall probability is an accident.

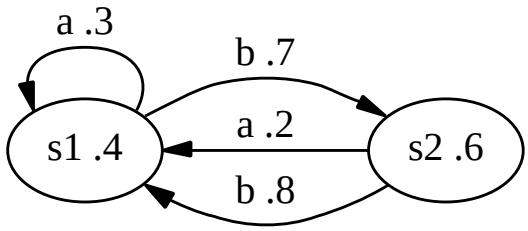

Figure 7.2: Multiple start states

The *overall* probability of the string a is the sum of the probabilities of all possible paths through the HMM: $.12 + .12 = .24$. Notice then that we cannot really be sure which path may have been taken to get to a, though if the paths have different probabilities then we can calculate the most likely probability.[2]

Indeterminacy can also be introduced by adding multiple arcs from the same state for the same symbol. For example, the HMM in figure 7.3 is a minimal modification of the Markov chain in figure 7.1. Consider how this HMM deals with a string ab. Here only s_1 is a legal start state. We can generate/accept a by either following the arc back to s_1 (.3) or by following the arc to s_2 (.2). In the former case, we can get b by following the arc from s_1 to s_2. In the latter case, we can get b by following the arc from s_2 back to s_1. This gives the following total probabilities for the two state sequences given.

$$\begin{aligned} s_1 \rightarrow s_1 \rightarrow s_2 &= .3 \times .5 = .15 \\ s_1 \rightarrow s_2 \rightarrow s_1 &= .2 \times .8 = .16 \end{aligned} \tag{7.1}$$

This results in an overall probability of .31 for ab. The second state sequence is of course the more likely one since it exhibits a higher overall probability.

A HMM can naturally include both extra arcs and multiple start states. The HMM in figure 7.4 exemplifies. This naturally results in even more

[2]This can be done efficiently with the Viterbi algorithm, which is presented in the next chapter.

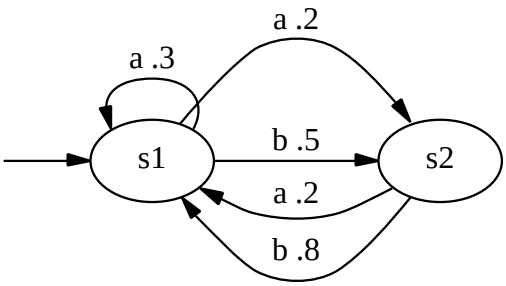

Figure 7.3: Multiple arcs

choices for any particular string. For example, the string *ab* can be produced with all the following sequences:

$$
\begin{aligned}
& s_1 \to s_1 \to s_2 = .1 \times .3 \times .5 = .015 \\
& s_1 \to s_2 \to s_1 = .1 \times .2 \times .8 = .016 \\
& s_2 \to s_1 \to s_2 = .9 \times .2 \times .5 = .09
\end{aligned} \tag{7.2}
$$

The overall probability is then .4.

7.3 Two kinds of HMMs

So far we have viewed HMMs as non-deterministic finite state automata with associated probabilities. However, there is an equivalent formalism that is often used to describe HMMs. Under this alternate view, we separate arc probabilities from emission probabilities so that effectively there is one and only one arc from each state to every other state. Some of these arcs may have a probability value of zero.[3] In addition, each state has an emission probability for every symbol, and again, some of these may have a zero probability value. An example is given in figure 7.5.

It is possible to show formally that the two characterizations are equivalent, but the construction is rather complex and so I leave that to the

[3]Equivalently, there is at most one arc from each state to any other state.

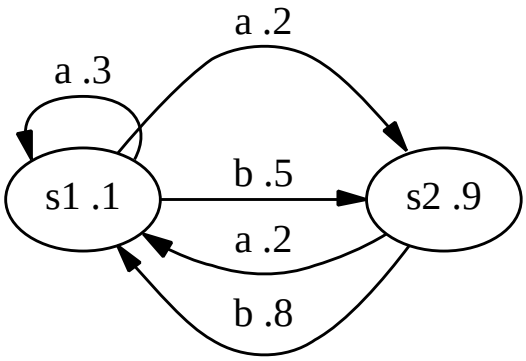

Figure 7.4: Multiple arcs & start states

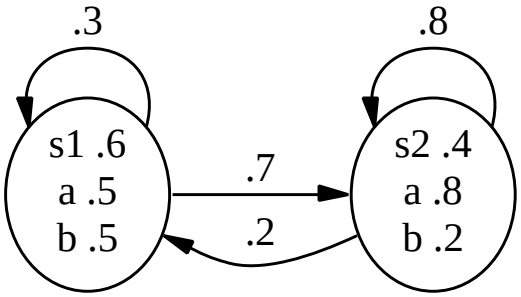

Figure 7.5: Another HMM formalism

interested reader. For our purposes, we will use both formalizations as convenient. The first is generally more useful when we wish to consider the ways in which a HMM is similar to a non-deterministic FSA. The latter formalism is more convenient when it is useful to treat arcs and symbol emissions separately. I will refer to the former as the *nondeterministic* characterization and the latter as the *separate emissions* characterization.

7.4 Formal HMM properties

There are a number of formal properties of Markov chains and HMMs that are useful. One extremely important property is *Limited Horizon*:

$$\mathrm{P}(X_{t+1} = s_k | X_1, \ldots, X_t) = \mathrm{P}(X_{t+1} = s_k | X_t) \tag{7.3}$$

This says that the probability of some state s_k given the set of states that have occurred before it is the same as the probability of that state given the *single* state that occurs just before it.

We've already said that a Markov chain or HMM can be defined in terms of a stochastic transition matrix A:

$$a_{ij} = \mathrm{P}(X_{t+1} = s_j | X_t = s_i) \tag{7.4}$$

Here, $a_{ij} \geq 0, \forall i, j$ and $\sum_{j=1}^{N} a_{ij} = 1, \forall i$.

One also has to set the probability of any initial state:

$$\pi_i = \mathrm{P}(X_1 = s_i) \tag{7.5}$$

Here, $\sum_{i=1}^{N} \pi_i = 1$.

The probability of any sequence of states is computed as follows:

$$\begin{aligned} \mathrm{P}(X_i, \ldots, X_T) &= \mathrm{P}(X_1)\mathrm{P}(X_2|X_1)\mathrm{P}(X_3|X_1, X_2)\ldots \\ &\quad \mathrm{P}(X_T|X_1, \ldots, X_{T-1}) \\ &= \mathrm{P}(X_1)\mathrm{P}(X_2|X_1)\mathrm{P}(X_3|X_2)\ldots\mathrm{P}(X_T|X_{T-1}) \\ &= \pi_{X_1} \prod_{T=1}^{T-1} a_{X_t X_{t+1}} \end{aligned} \tag{7.6}$$

Under our second formalization, a HMM additionally has *emission probabilities*:

$$\mathrm{P}(O_t = k | X_t = s_i, X_{t+1} = s_j) = b_{ijk} \tag{7.7}$$

Set of states	$S = s_1, \ldots, s_N$
Output alphabet	$K = \{k_1, \ldots, k_M\} = \{1, \ldots, M\}$
Initial state probabilities	$\Pi = \{\pi_i\}, i \in S$
State transition probabilities	$A = \{a_{ij}\}, i, j \in S$
Symbol emission probabilities	$B = \{b_{ijk}\}, i, j \in S, k \in K$
State sequence	$X = (X_1, \ldots, X_{T+1})$
	$X_t : S \rightarrow \{1, \ldots, N\}$

Figure 7.6: Terminology for HMMs

The terminology for a HMM is given in figure 7.6.

As a consequence of the structure of a HMM, there is a probability distribution over strings of any particular length:

$$\forall n \sum_{w_{1n}} \mathrm{P}(w_{1n}) = 1 \tag{7.8}$$

What this means is that when we sum the probabilities of all possible strings of any length n, their total is 1.

We calculate the probability of some string of symbols as follows. For any state sequence $X = (X_1, \ldots, X_{T+1})$:

$$\begin{aligned} \mathrm{P}(O|X, \mu) &= \textstyle\prod_{t=1}^{T} \mathrm{P}(o_t|X_t, X_{t+1}, \mu) \\ &= b_{X_1 X_2 o_1} b_{X_2 X_3 o_2} \ldots b_{X_T X_{T+1} o_T} \end{aligned} \tag{7.9}$$

and,

$$\mathrm{P}(X|\mu) = \pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \ldots a_{X_T X_{T+1}} \tag{7.10}$$

Since

$$\mathrm{P}(O, X|\mu) = \mathrm{P}(O|X, \mu)\mathrm{P}(X|\mu) \tag{7.11}$$

it follows that:

$$\begin{aligned} \mathrm{P}(O|\mu) &= \textstyle\sum_X \mathrm{P}(O|X, \mu)\mathrm{P}(X|\mu) \\ &= \textstyle\sum_{X_1 \cdots X_{T+1}} \pi_{X_1} \prod_{t=1}^{T} a_{X_t X_{t+1}} b_{X_t X_{t+1} o_t} \end{aligned} \tag{7.12}$$

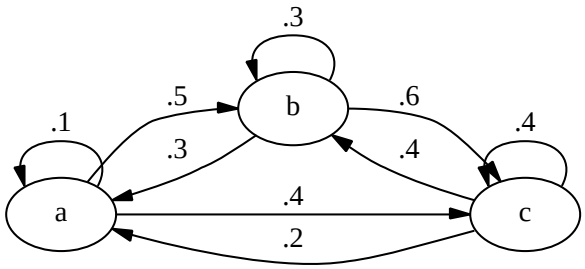

Figure 7.7: Simple HMM for a bigram model

7.5 Bigrams and HMMs

It is a straightforward matter to treat bigram models in terms of HMMs. Imagine we have a vocabulary of three words *a b c*. We simply create a HMM with a state for each item in the vocabulary and then arcs indicate the conditional probability of each bigram. An example is given in figure 7.7. Here, for example, the conditional probability $P(b|a) = .5$. A complete text given this model would get an overall probability in the usual fashion.

7.6 Higher-order n-grams

This is not, in fact, how such models are constructed, as we will see in the following chapter. However, it's important to first consider how such a model might be extended to higher-order n-grams. At first blush, we might think there's a problem. After all, the limited horizon property says that the history an HMM is sensitive to can be restricted to the immediately preceding state. A trigram model would appear to require more.

This, however, doesn't reckon with the assumption of a finite vocabulary (albeit a large finite vocabulary). In the previous example, we took each state as equivalent to a vocabulary item. To treat a trigram model, we must allow for states to be equivalent to both single words in the vocabulary and every possible combination of words in the vocabulary. For example, to construct a HMM for a trigram model for the same vocabulary as the previous examples,

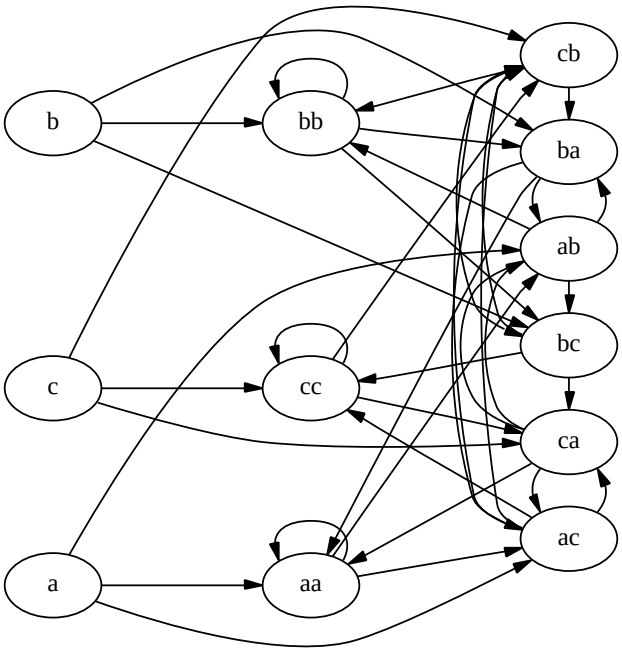

Figure 7.8: HMM for a trigram model

we would augment the model to include nine additional states representing each combination of words. This is shown in figure 7.8. (Probabilities have been left off to enhance legibility.)