INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

Markov Models and Hidden Markov Models: A Brief Tutorial

Eric Fosler-Lussier

TR-98-041

December 1998

Abstract

This tutorial gives a gentle introduction to Markov models and Hidden Markov models as mathematical abstractions, and relates them to their use in automatic speech recognition. This material was developed for the Fall 1995 semester of *CS188: Introduction to Artificial Intelligence* at the University of California, Berkeley. It is targeted for introductory AI courses; basic knowledge of probability theory (*e.g.* Bayes' Rule) is assumed. This version is slightly updated from the original, including a few minor error corrections, a short "Further Reading" section, and exercises that were given as a homework in the Fall 1995 class.

1 Markov Models

Let's talk about the weather. Here in Berkeley, we have three types of weather: *sunny*, *rainy*, and *foggy*. Let's assume for the moment that the weather lasts all day, i.e. it doesn't change from rainy to sunny in the middle of the day.

Weather prediction is all about trying to guess what the weather will be like tomorrow based on a history of observations of weather. Let's assume a simplified model of weather prediction: we'll collect statistics on what the weather was like today based on what the weather was like yesterday, the day before, and so forth. We want to collect the following probabilities:

$$P(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_1) \tag{1}$$

Using expression 1, we can give probabilities of types of weather for tomorrow and the next day using n days of history. For example, if we knew that the weather for the past three days was {sunny, sunny, foggy} in chronological order, the probability that tomorrow would be rainy is given by:

$$P(w_4 = Rainy \mid w_3 = Foggy, w_2 = Sunny, w_1 = Sunny) \tag{2}$$

Here's the problem: the larger n is, the more statistics we must collect. Suppose that $n = 5$, then we must collect statistics for $3^5 = 243$ past histories. Therefore, we will make a simplifying assumption, called the *Markov Assumption*:

In a sequence $\{w_1, w_2, \ldots, w_n\}$:

$$P(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_1) \approx P(w_n \mid w_{n-1}) \tag{3}$$

This is called a *first-order* Markov assumption, since we say that the probability of an observation at time n only depends on the observation at time $n-1$. A *second-order* Markov assumption would have the observation at time n depend on $n-1$ and $n-2$. In general, when people talk about Markov assumptions, they usually mean first-order Markov assumptions; I will use the two terms interchangeably.

We can the express the joint probability using the Markov assumption.[1]

$$P(w_1, \ldots, w_n) = \Pi_{i=1}^{n} P(w_i \mid w_{i-1}) \tag{4}$$

So this now has a profound affect on the number of histories that we have to find statistics for— we now only need $3^2 = 9$ numbers to characterize the probabilities of all of the sequences. This assumption may or may not be a valid assumption depending on the situation (in the case of weather, it's probably not valid), but we use these to simplify the situation.

So let's arbitrarily pick some numbers for $P(w_{\text{tomorrow}} \mid w_{\text{today}})$, expressed in Table 1.

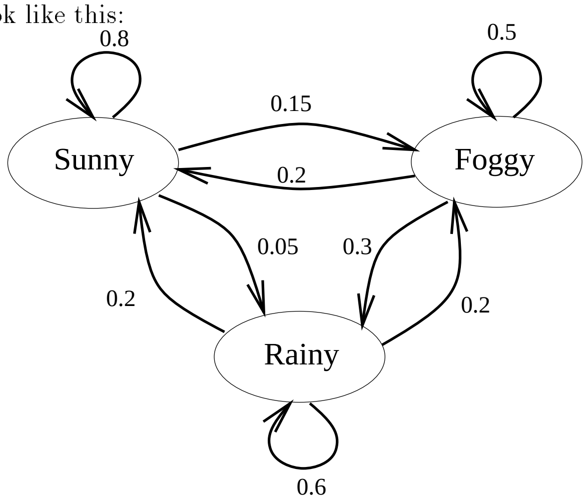
		Tomorrow's Weather		
		Sunny	Rainy	Foggy
Today's Weather	Sunny	0.8	0.05	0.15
	Rainy	0.2	0.6	0.2
	Foggy	0.2	0.3	0.5

Table 1: Probabilities of Tomorrow's weather based on Today's Weather

For first-order Markov models, we can use these probabilities to draw a probabilistic finite state automaton. For the weather domain, you would have three states (Sunny, Rainy, and Foggy), and

[1] One question that comes to mind is "What is w_0?" In general, one can think of w_0 as the $START$ word, so $P(w_1|w_0)$ is the probability that w_1 can start a sentence. We can also just multiply the prior probability of w_1 with the product of $\Pi_{i=2}^{n} P(w_i \mid w_{i-1})$; it's just a matter of definitions.

every day you would transition to a (possibly) new state based on the probabilities in Table 1. Such an automaton would look like this:



1.1 Questions:

1. Given that today is sunny, what's the probability that tomorrow is sunny and the day after is rainy?

   Well, this translates into:

$$\begin{aligned}
P(w_2 = \text{Sunny}, w_3 = \text{Rainy}|w_1 = \text{Sunny}) &= P(w_3 = \text{Rainy} \mid w_2 = \text{Sunny}, w_1 = \text{Sunny}) * \\
&\quad P(w_2 = \text{Sunny} \mid w_1 = \text{Sunny}) \\
&= P(w_3 = \text{Rainy} \mid w_2 = \text{Sunny}) * \\
&\quad P(w_2 = \text{Sunny} \mid w_1 = \text{Sunny}) \\
&= (0.05)(0.8) \\
&= 0.04
\end{aligned}$$

   You can also think about this as moving through the automaton, multiplying the probabilities as you go.

2. Given that today is foggy, what's the probability that it will be rainy two days from now?

   There are three ways to get from foggy today to rainy two days from now: {foggy, foggy, rainy}, {foggy, rainy, rainy}, and {foggy, sunny, rainy}. Therefore we have to sum over these paths:

$$\begin{aligned}
P(w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) &= P(w_2 = \text{Foggy}, w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_2 = \text{Rainy}, w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_2 = \text{Sunny}, w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&= P(w_3 = \text{Rainy} \mid w_2 = \text{Foggy})P(w_2 = \text{Foggy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_3 = \text{Rainy} \mid w_2 = \text{Rainy})P(w_2 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_3 = \text{Rainy} \mid w_2 = \text{Sunny})P(w_2 = \text{Sunny} \mid w_1 = \text{Foggy})
\end{aligned}$$

$$\begin{aligned} &= (0.3)(0.5) + (0.6)(0.3) + (0.05)(0.2) \\ &= 0.34 \end{aligned}$$

Note that you have to know where you start from. Usually Markov models start with a null start state, and have transitions to other states with certain probabilities. In the above problems, you can just add a start state with a single arc with probability 1 to the initial state (sunny in problem 1, foggy in problem 2).

2 Hidden Markov Models

So what makes a Hidden Markov Model? Well, suppose you were locked in a room for several days, and you were asked about the weather outside. The only piece of evidence you have is whether the person who comes into the room carrying your daily meal is carrying an umbrella or not.

Let's suppose the following probabilities:

	Probability of Umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

Table 2: Probabilities of Seeing an Umbrella Based on the Weather

Remember, the equation for the weather Markov process before you were locked in the room was:

$$P(w_1, \ldots, w_n) = \Pi_{i=1}^{n} P(w_i \mid w_{i-1}) \tag{5}$$

Now we have to factor in the fact that the actual weather is *hidden* from you. We do that by using Bayes' Rule:

$$P(w_1, \ldots, w_n \mid u_1, \ldots, u_n) = \frac{P(u_1, \ldots, u_n | w_1, \ldots, w_n) P(w_1, \ldots, w_n)}{P(u_1, \ldots, u_n)} \tag{6}$$

where u_i is true if your caretaker brought an umbrella on day i, and false if the caretaker didn't. The probability $P(w_1, \ldots, w_n)$ is the same as the Markov model from the last section, and the probability $P(u_1, \ldots, u_n)$ is the prior probability of seeing a particular sequence of umbrella events (e.g. {True, False, True}). The probability $P(u_1, \ldots, u_n | w_1, \ldots, w_n)$ can be estimated as $\Pi_{i=1}^{n} P(u_i | w_i)$, if you assume that, for all i, given w_i, u_i is independent of all u_j and w_j, for all $j \neq i$.

2.1 Questions:

1. Suppose the day you were locked in it was sunny. The next day, the caretaker carried an umbrella into the room. Assuming that the prior probability of the caretaker carrying an umbrella on any day is 0.5, what's the probability that the second day was rainy?

$$\begin{aligned} &\begin{array}{l} P(w_2 = \text{Rainy}| \\ w_1 = \text{Sunny}, u_2 = \text{True}) \end{array} &= \frac{P(w_2 = \text{Rainy}, w_1 = \text{Sunny} | u_2 = \text{T})}{P(w_1 = \text{Sunny} | u_2 = \text{T})} \\ &(u_2 \textit{ and } w_1 \textit{ independent}) &= \frac{P(w_2 = \text{Rainy}, w_1 = \text{Sunny} | u_2 = \text{T})}{P(w_1 = \text{Sunny})} \end{aligned}$$

$$
\begin{aligned}
(Bayes'\ Rule) &= \frac{P(u_2 = \mathrm{T} | w_1 = \mathrm{Sunny}, w_2 = \mathrm{Rainy}) P(w_2 = \mathrm{Rainy}, w_1 = \mathrm{Sunny})}{P(w_1 = \mathrm{Sunny}) P(u_2 = \mathrm{T})} \\
(Markov\ \ assumption) &= \frac{P(u_2 = \mathrm{T} | w_2 = \mathrm{Rainy}) P(w_2 = \mathrm{Rainy}, w_1 = \mathrm{Sunny})}{P(w_1 = \mathrm{Sunny}) P(u_2 = \mathrm{T})} \\
(P(A,B) = P(A|B)P(B)) &= \frac{P(u_2 = \mathrm{T} | w_2 = \mathrm{Rainy}) P(w_2 = \mathrm{Rainy} | w_1 = \mathrm{Sunny}) P(w_1 = \mathrm{Sunny})}{P(w_1 = \mathrm{Sunny}) P(u_2 = \mathrm{T})} \\
(Cancel : P(Sunny)) &= \frac{P(u_2 = \mathrm{T} | w_2 = \mathrm{Rainy}) P(w_2 = \mathrm{Rainy} | w_1 = \mathrm{Sunny})}{P(u_2 = \mathrm{T})} \\
&= \frac{(0.8)(0.05)}{0.5} \\
&= .08
\end{aligned}
$$

2. Suppose the day you were locked in the room it was sunny; the caretaker brought in an umbrella on day 2, but not on day 3. Again assuming that the prior probability of the caretaker bringing an umbrella is 0.5, what's the probability that it's foggy on day 3?

$$
\begin{aligned}
\begin{array}{r} P(w_3 = \mathrm{F} \mid \\ w_1 = \mathrm{S}, u_2 = \mathrm{T}, u_3 = \mathrm{F}) \end{array} &= P(w_2 = \mathrm{Foggy}, w_3 = \mathrm{Foggy} \mid \\
&\qquad\quad w_1 = \mathrm{Sunny}, u_2 = \mathrm{True}, u_3 = \mathrm{False}) + \\
&\quad P(w_2 = \mathrm{Rainy}, w_3 = \mathrm{Foggy} \mid \ldots) + \\
&\quad P(w_2 = \mathrm{Sunny}, w_3 = \mathrm{Foggy} \mid \ldots) \\
&= \frac{P(u_3 = F | w_3 = F) P(u_2 = T | w_2 = F) P(w_3 = F | w_2 = F) P(w_2 = F | w_1 = S) P(w_1 = S)}{P(u_3 = F) P(u_2 = T) P(w_1 = S)} + \\
&\quad \frac{P(u_3 = F | w_3 = F) P(u_2 = T | w_2 = R) P(w_3 = F | w_2 = R) P(w_2 = R | w_1 = S) P(w_1 = S)}{P(u_3 = F) P(u_2 = T) P(w_1 = S)} + \\
&\quad \frac{P(u_3 = F | w_3 = F) P(u_2 = T | w_2 = S) P(w_3 = F | w_2 = S) P(w_2 = S | w_1 = S) P(w_1 = S)}{P(u_3 = F) P(u_2 = T) P(w_1 = S)} \\
&= \frac{P(u_3 = F | w_3 = F) P(u_2 = T | w_2 = F) P(w_3 = F | w_2 = F) P(w_2 = F | w_1 = S)}{P(u_3 = F) P(u_2 = T)} + \\
&\quad \frac{P(u_3 = F | w_3 = F) P(u_2 = T | w_2 = R) P(w_3 = F | w_2 = R) P(w_2 = R | w_1 = S)}{P(u_3 = F) P(u_2 = T)} + \\
&\quad \frac{P(u_3 = F | w_3 = F) P(u_2 = T | w_2 = S) P(w_3 = F | w_2 = S) P(w_2 = S | w_1 = S)}{P(u_3 = F) P(u_2 = T)} \\
&= \frac{(0.7)(0.3)(0.5)(0.15)}{(0.5)(0.5)} + \\
&\quad \frac{(0.7)(0.8)(0.2)(0.05)}{(0.5)(0.5)} + \\
&\quad \frac{(0.7)(0.1)(0.15)(0.8)}{(0.5)(0.5)} \\
&= 0.119
\end{aligned}
$$

3 Relationship to Speech

Let's get away from umbrellas and such for a moment and talk about *real* things, like speech. In speech recognition, the basic idea is to find the most likely string of words given some acoustic input, or:

$$\text{argmax}_{\mathbf{w} \in \mathcal{L}} P(\mathbf{w}|\mathbf{y}) \tag{7}$$

Where $\mathbf{w}$ is a string of words, $\mathcal{L}$ is the language you're interested in, and $\mathbf{y}$ is the set of acoustic vectors that you've gotten from your front end processor[2]. To compare this to the weather example, the acoustics are our observations, similar to the umbrella observations, and the words are similar to the weather on successive days.

Remember that the basic equation of speech recognition is Bayes' Rule:

$$\text{argmax}_{\mathbf{w} \in \mathcal{L}} P(\mathbf{w}|\mathbf{y}) = \text{argmax}_{\mathbf{w} \in \mathcal{L}} \frac{P(\mathbf{y}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{y})} \tag{8}$$

For a single speech input (e.g. one sentence), the acoustics $(\mathbf{y})$ will be constant, and therefore, so will $P(\mathbf{y})$. So we only need to find:

$$\text{argmax}_{\mathbf{w} \in \mathcal{L}} P(\mathbf{y}|\mathbf{w})P(\mathbf{w}) \tag{9}$$

The first part of this expression is called the model *likelihood*, and the second part is a prior probability of the word string.

3.1 Word pronunciations

First, we want to be able to calculate $P(\mathbf{y}|\mathbf{w})$, the probability of the acoustics given the word. Note that many acoustic vectors can make up a word, so we choose an intermediate representation: the phone. Here is a word model for the word "of":

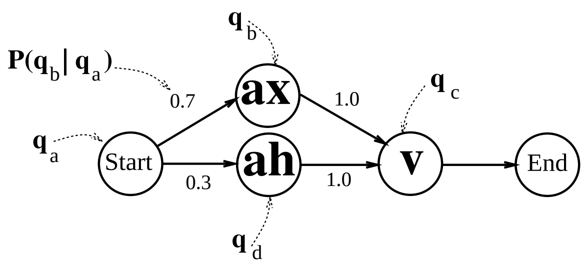


Figure 1: Word Model for "of"

In this model, we represent phone states as q_i's. We can now break up $P(\mathbf{y}|\mathbf{w})$ into $P(\mathbf{y}|\mathbf{q}, \mathbf{w})$ and $P(\mathbf{q}|\mathbf{w})$. Since the picture above only talks about one word w_{of}, we leave the $\mathbf{w}$ out of the conditioning here.

$$P(y \mid w_i) = P(y|\text{Start\_ax\_v\_End}) + P(y|\text{Start\_ah\_v\_End}) \tag{10}$$

$$P(y|w_{of}) = P(q_b|q_a)P(y_0|q_b)P(q_c|q_b)P(y_1|q_c) + P(q_d|q_a)P(y_0|q_d)P(q_c|q_d)P(y_1|q_c) \tag{11}$$

The individual $P(y_i|q_i)$ are given by a Gaussian or multi-layer-perceptron likelihood estimator. The transitions $P(q_j|q_i)$ depend on the pronunciations of words[3].

[2] This is usually an n-dimensional vector which represents 10-20 milliseconds of speech. A typical value for n is 8 to 40. Your mileage may vary based on the type of system.

[3] I've made a simplifying assumption here that for the word "of" we only have two acoustic vectors, y_0 and y_1.

3.2 Bigram grammars

In the second part of expression 9, we were concerned with $P(\mathbf{w})$. This is the prior probability of the string of words we are considering. Notice that we can also replace this with a Markov model, using the following equation:

$$P(\mathbf{w}) = \Pi_{i=1}^{n} P(w_i|w_{i-1}) \tag{12}$$

What is this saying? Well, suppose that we had the string "I like cabbages". The probability of this string would be:

$$P(\text{I like cabbages}) = P(\text{I}|\text{START})P(\text{like}|\text{I})P(\text{cabbages}|\text{like}) \tag{13}$$

This type of grammar is called a bigram grammar (since it relates two (bi) words (grams)). Using a higher order Markov assumption allows more context into the grammars; for instance, a second-order Markov language model is a trigram grammar, where the probabilities of word n are given by $P(w_n|w_{n-1}, w_{n-2})$.

4 Further reading

I have only really scratched the surface of automatic speech recognition (ASR) here; for the sake of brevity I have omitted many of the details that one would need to consider in building a speech recognizer. For a more mathematically rigorous introduction to Hidden Markov Models and ASR systems, I would suggest (Rabiner 1989). A recent book by Jelinek (1997) gives a good overview of how to build large-vocabulary ASR systems, at about the same level of rigor as Rabiner. Hidden Markov models can also be viewed as one instance of statistical graphical models (Smyth *et al.* 1997). Other recommended introductions to HMMs, ASR systems, and speech technology are Rabiner & Juang (1993), Deller *et al.* (1993), and Morgan & Gold (forthcoming).

5 Exercises

Imagine that you work at ACME Chocolate Factory, confectioners extraordinaire. Your job is to keep an eye on the conveyor belt, watching the chocolates as they come out of the press one at a time.

Suppose that ACME makes two types of chocolates: ones with almonds and ones without. For the first few problems, assume you can tell with 100% accuracy what the chocolate contains. In the control room, there is a lever that switches the almond control on and off. When the conveyor is turned on at the beginning of the day, there is a 50% chance that the almond lever is on, and a 50% chance that it is off. As soon as the conveyor belt is turned on, it starts making a piece of candy.

Unfortunately, someone has let a monkey loose in the control room, and it has locked the door and started the conveyor belt. The lever cannot be moved while a piece of candy is being made. Between pieces, however, there is a 30% chance that the monkey switches the lever to the other position (i.e. turns almonds on if it was off, or off if it was on).

1. Draw a Markov Model that represents the situation. *Hint: you need three states.*

Now assume that there is a coconut lever as well, so that there are four types of candy: Plain, Almond, Coconut, and Almond+Coconut. Again, there is a 50% chance of the lever being on at the beginning of the day, and the chance of the monkey switching the state of the second lever between candies is also 30%. Assume that the switching of the levers is independent of each other.

However, we don't know *a priori* how long a word is. In order to handle multiple lengths of words, self-loops are often added to the pronunciation graph. Thus, one can stay in state q_b with probability $P(q_b|q_b)$; we can then handle these probabilities like any other transition probabilities on the graph.

2. Now draw a Markov Model for both production of all four types of chocolates.
3. What is the probability that the machine will produce in order: {Plain, Almond, Almond, Almond+Coconut}

Now assume that you can't tell what's inside the chocolate candy, only that the chocolate is light or dark. Use the following table for $P(\text{color}|\text{stuff inside})$:

Inside	P(color=light)	P(color=dark)
Plain	0.1	0.9
Almond	0.3	0.7
Coconut	0.8	0.2
Almond+Coconut	0.9	0.1

Assume that the prior probability of P(color=light)=P(color=dark)=0.5, and that the color of one chocolate is independent of the next, given the filling.

4. If the first two chocolates were dark and light respectively, what is the probability that they are Almond and Coconut respectively?

Acknowledgments

The weather story for Markov models has been floating around the speech community for a while; it is presented (probably originally) by Rabiner (1989). Thanks to Su-Lin Wu, Nelson Morgan, Nikunj Oza, Brian Kingsbury, and Jitendra Malik for proofreading and comments. Any mistakes in this tutorial, however, are the responsibility of the author, and not of the reviewers.

References

DELLER, J., J. PROAKIS, & J. HANSEN. 1993. *Discrete-Time Processing of Speech Signals*. New York: Macmillan Publishing.

JELINEK, F. 1997. *Statistical Methods for Speech Processing*. Language, Speech and Communcation Series. Cambridge, MA: MIT Press.

MORGAN, N., & B. GOLD. forthcoming. *Speech and Audio Signal Processing*. Wiley Press.

RABINER, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77.

——, & B.-H. JUANG. 1993. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series. Englewood Cliffs, NJ: Prentice Hall.

SMYTH, P., D. HECKERMAN, & M.I. JORDAN. 1997. Probabilistic independence networks for hidden Markov probability models. *Neural Computation* 9.227–270.