An Exploration of Hidden Markov Models

by Anna Sandell
Course in Speech Recognition, KTH, Autumn 2003

Abstract

The concept of hidden Markov models (HMM) is explored by comparison of different introductory texts. Further, an experiment with Markov chains is performed, resulting in new texts built from old texts using the Markov assumption.

1. Introduction

The concept of hidden Markov models (HMM's) has fascinated me since I first heard of it, but I find it difficult to grasp. Therefore my aim with this paper is to visualise, for me and others, how HMM's work, for example in speech recognition.

2. Hidden Markov model

Let us first dissect the designation "hidden Markov model", word by word, starting from behind:

2.1. Model

The hidden Markov model is apparently some kind of **model**. The observable output which real-world processes (for example speech) generally produce can be characterized as signals [Rab89]. To characterize such real-world signals we need signal models. Rabiner gives three reasons for applying signal models:

- A signal model can provide the basis for a theoretical description of a signal processing system, which can be used to process the signal so as to provide a desired output.
- Signal models are potentially capable of letting us learn about the signal source without having the source available. Thus simulations of the source can be made.
- Signal models often work well in practice and enable us to realize important practical systems, for example recognition systems, in a very efficient manner.

Signal models can be deterministic (for example a sine wave, whose exact output we know) or statistical. Markov models are statistical, and the underlying assumption is that the signal can be well characterized as a parametric random process.

2.2. Markov

A **Markov** chain rests on the Markov assumption: the probability of the random variable at a given time depends only on the value at the preceding time [Hua01]. Putting it formally we get

$$P(X_i|X_1^{i-1}) = P(X_i|X_{i-1})$$

where P is a probability, X is a random variable, X_1^{i-1} is the sequence of random variables X_1, X_2,…, X_{i-1} and $P(X_i|X_1^{i-1})$ is the probability of X_i given all the earlier variables X_1^{i-1}.

The Markov assumption is simple yet powerful. In Appendix 1, the Markov assumption is further explored through an experiment where different kinds of texts are used as input, and the output is new texts where the word sequences are based on the input texts and the Markov assumption.

2.3. Hidden

The **hidden** Markov model has an extra layer constituted of another stochastic process, which makes the bottom layer hidden.

One way of visualising the HMM is the urn-and-ball model (used by Rabiner in both [Rab89] and [Rab93]). Think of being in a room where behind a curtain or

drapery there are an unknown number of glass urns containing balls of different colours. A genie picks a ball from an urn and tells you which colour the ball has but not from which urn it was taken. The ball is put back in the urn, and another ball is picked, from the same urn or from another, and the genie tells you the colour of the ball, and so on. Now you have to choose a model for the experiment, based only on the series of ball colours the genie tells you.

More formally, the HMM is characterised by five parameters: N, M, A, B and π. N and M are model parameters and A, B and π are probability parameters.

- N is the number of states in the HMM (the number of urns in the urn-and-ball example).
- M is the number of distinct observation symbols per state (the number of ball colours in each urn).
- A is an N×N matrix of parameters a_{ij}, 1≤i,j≤N, standing for the state transition probabilities (the genie's tendency to jump from one specific urn to another).
- B is an M×N matrix of parameters $b_j(k)$, 1≤j≤N, 1≤k≤M, standing for the observation symbol probabilities (the probabilities for the different ball to be picked by the genie in each state).
- π is the initial state distribution (the probability for each urn to be the genie's starting place).

The three probability measures A, B and π are sometimes written with the compact notation

$$\lambda = (A, B, \pi) .$$

Figure 1 shows an HMM with three states.

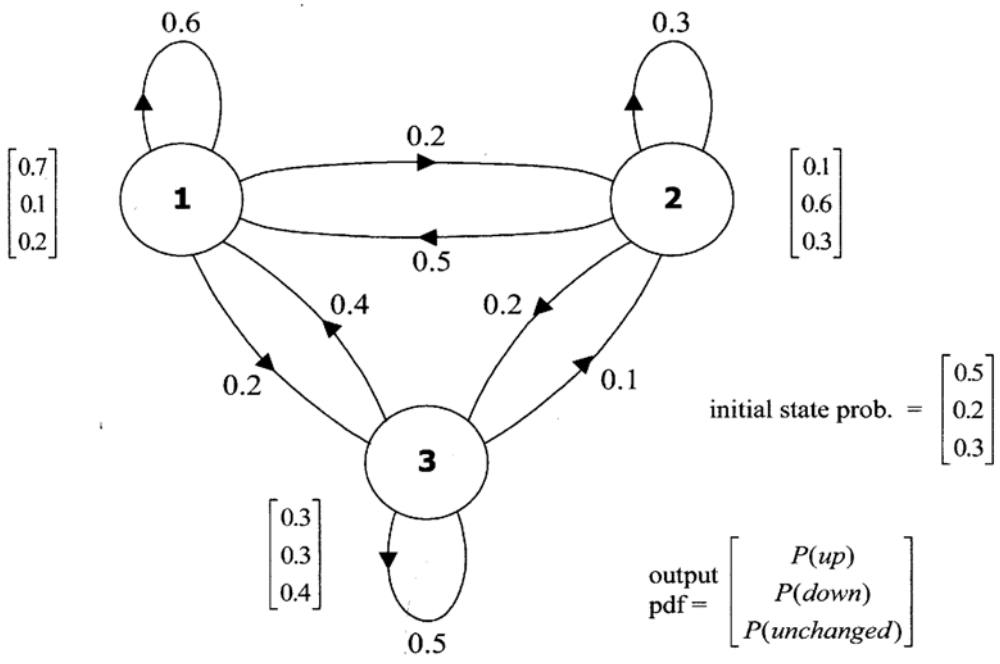

Figure 1. An HMM with three states (1, 2, 3). The probabilities for transitions between states are shown on the arcs, and the probabilities for different outputs are shown next to the states. (After [Hua01].)

3. Three fundamental problems

In most introductory texts on HMM's, the following three problems play leading parts as fundamental for HMM design: the evaluation problem, the decoding problem and the learning problem [Rab89], [Hua01].

3.1. The evaluation problem

This problem handles evaluation of the probability of a sequence of observations given a specific HMM:

Given the observation sequence $O = O_1 O_2 \cdots O_T$, *and a model* $\lambda = (A, B, \pi)$, *how do we efficiently compute* $P(O|\lambda)$, *the probability of the observation sequence, given the model?*

3.2. The decoding problem

This problem handles determination of a best sequence of model states:

Given the observation sequence $O = O_1 O_2 \cdots O_T$, *and a model* λ, *how do we choose a corresponding state sequence* $Q = q_1 q_2 \cdots q_T$ *which is optimal in some meaningful sense (i.e., best "explains" the observations)?*

3.3. The learning problem

This problem handles adjustment of model parameters so as to best account for the observed signal:

How do we adjust the model parameters $\lambda = (A, B, \pi)$ *to maximize* $P(O|\lambda)$ *?*

3.4. The roles of the fundamental problems in speech recognition

In a simple isolated word recognizer, the solution to the learning problem is used in building individual word models, by optimally estimating model parameters for each word model. With the solution to the decoding problem we can segment each of the word training sequences into states and use the result to make refinements on the model (for example introducing more states). Finally, recognition is performed using the solution of the evaluation problem [Rab89].

4. Another way of viewing HMM's

With complementing descriptions of the algorithms used in solving the three fundamental problems (for example the Viterbi algorithm, the Baum-Welch algorithm (also known as the forward-backward algorithm), maximum likelihood optimisation, and maximum mutual information estimation), the HMM theory is ready for use in building speech recognition systems. But still, I find HMM's a bit mystical. Are there other ways of viewing HMM's?

In [Cha93] the HMM's are introduced as finite state automata. The example in figure 1 (from [Hua01], chapter 8.2 Definition of the Hidden Markov Model) can be modified to reflect Charniak's way of explaining HMM's.

First, I choose to reduce the number of output pdf's (probability density functions) from three to two. (This is not necessary for this way of viewing the problem, but it will make the graphical presentation foreseeable.) In the [Hua01] example, the output pdf's represent the Dow Jones index (used in economy) going up, down or staying unchanged. Let us instead use an output pdf with two values, here representing "the Dow" going up or not going up. In other words: we add up the probabilities of "down" and "unchanged" to a new category "not up". The example modified in this manner is shown in figure 2.

Then we adapt the Charniak style of HMM's to the example. The HMM in form of a finite state automaton has one arc for each combination of output symbol and state transition. Therefore the probabilities for the output and for the state transition are multiplied, giving an M×N×N matrix C,

$$C_{ijk} = A_{ij} B_j(k)$$

Figure 3 shows the result for the new version of our modified Dow Jones example.

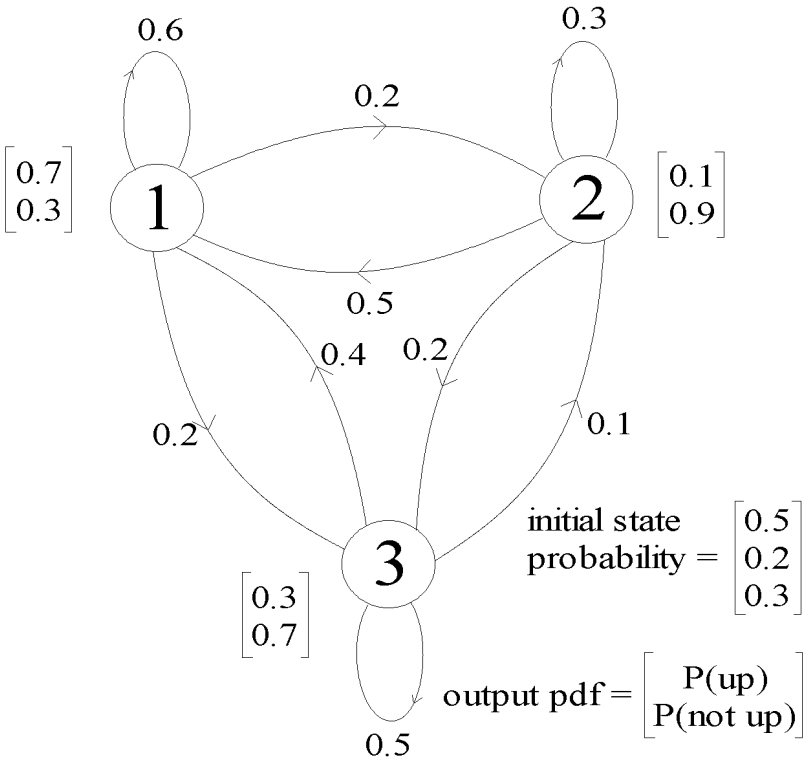

Figure 2. The example from figure 1, modified to two output symbols instead of three.

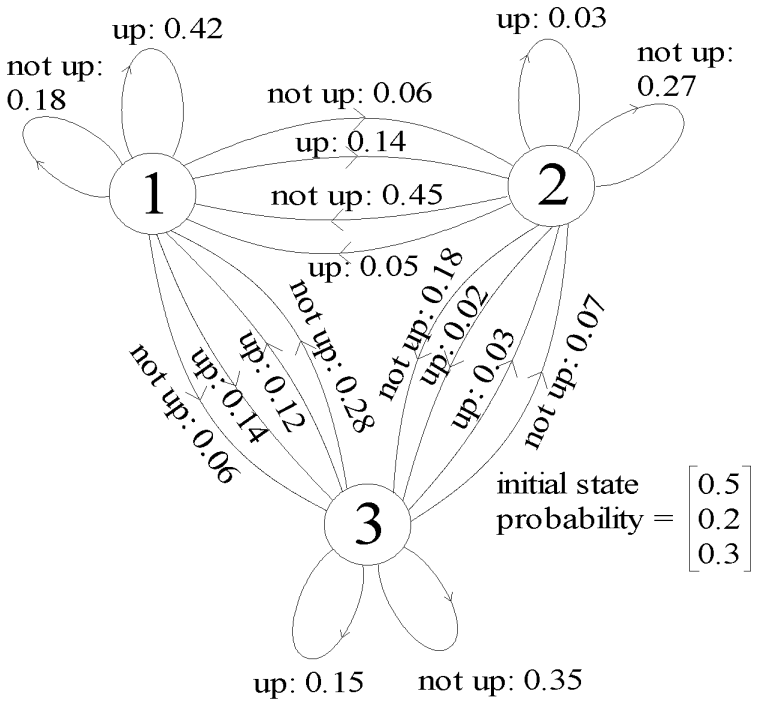

Figure 3. The example from figure 2, with HMM as a finite state automaton.

5. Graphical models

The above example was introduced in graphical form, and there is also a branch of speech recognition research called “graphical models” (GM for short). An HMM is a special case of a GM [Bil01].

6. Acknowledgements

I wish to thank my husband Håkan Sandell, who wrote the Markov chain scripts, my brother Johan Sundström, who collected data intended for use with the Markov chain scripts, and Chris Brew and Marc Moens whose yet unpublished book “Data-Intensive Linguistics” gave me the idea of comparing Rabiner’s and Charniak’s descriptions of HMM’s.

References

[Bil01] J.A. Bilmes. Graphical Models and Automatic Speech Recognition. UWEE Technical Report Number UWEETR-2001-0005, 2001. <https://www.ee.washington.edu/techsite/papers/documents/UWEETR-2001-0005.pdf>

[Cha93] E. Charniak. Statistical language learning. MIT Press, 1993.

[Enc03] Sonnet. Encyclopædia Britannica. Retrieved December 22, 2003, from Encyclopædia Britannica Premium Service. <http://www.britannica.com/eb/article?eu=70486>

[Hua01] X. Huang, A. Acero and H.-W. Hon. Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, 2001.

[Rab89] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, Vol. 77, NO. 2, Feb. 1989.

[Rab93] L.R. Rabiner and B.-H. Juang. Fundamentals of Speech Recognition. Prentice Hall Signal Processing Series, 1993.

[Sha] William Shakespeare’s sonnets <http://www.webterrace.com/shakespeare/sonnets.htm>

Let me not to the marriage of true minds
Admit impediments, love is not love
Which alters when it alteration finds,
Or bends with the remover to remove.
O no, it is an ever-fixed mark
That looks on tempests and is never shaken;
It is the star to every wand'ring bark,
Whose worth's unknown, although his height be taken.
Love's not Time's fool, though rosy lips and cheeks
Within his bending sickle's compass come,
Love alters not with his brief hours and weeks,
But bears it out even to the edge of doom:
If this be error and upon me proved,
I never writ, nor no man ever loved.

Figure 4. William Shakespeare's sonnet number 116.

Appendix 1: Playing with Markov chains

As stated earlier, a first order Markov chain consists of states related to each other through the Markov assumption: the probability of the random variable at a given time depends only on the value at the preceding time.

In this experiment we generate new texts from old ones with help of the Markov assumption for a Markov chain of words. Each word is a random variable, and to extend a sequence of words, the next word is chosen according to its probability to follow the last word of the sequence, based on statistics from the input texts (which are used as training material).

The program takes text files or html files as input (training data) and outputs a new text according to the statistics of the training data. Each word is a link in the Markov chain, and end-of-line and end-of-file are also considered as individual words.

The first texts we use are sonnets, to be precise the 154 sonnets written by William Shakespeare. The sonnet is a "fixed verse form of Italian origin consisting of 14 lines that are typically five-foot iambics rhyming according to a prescribed scheme" [Enc03]. Shakespeare is the most famous writer of sonnets in English, and his complete sonnets are available at several Internet sites, for example [Sha]. An example of a Shakespearean sonnet can be seen in figure 4. Now all 154 sonnets are used as training data, giving text output which is more or less grammatical. In figure 5, 14 lines of the output can be seen.

Perhaps the text in figure 5 can be called poetry, but it is definitely not a sonnet. The underlined words ("it alteration finds") in figures 4 and 5 remind us that the texts are in some way related. The word "alteration" only occurs once in all Shakespeare sonnets, and therefore (as our model, a first order Markov chain, only remembers the previous word and not the words

before), the words immediately before and after “alteration” of the synthetic text must be the same as in the original occurrence (the probability for that is 1). To get new texts, which look more like sonnets, we would need more training material and/or a better language model. As Shakespeare did not publish more sonnets than the 154 we have trained our system with, we must use something else, for example sonnets by other renaissance poets. (Or if we were after “the Shakespearian style” rather than the sonnet form, we could use plays by Shakespeare.) A better language model could be a Markov model of higher order than one (which we used now), but with such a model we must increase the training material further, if we do not want the output to be copies of the training material. (This is a common problem in speech recognition: that finer, better tuned models need more training data than the simple models to work well.) Probably another kind of model would work better, as we must also take care of the rhymes, which are in some cases very distant apart.

The program was also tested with other kinds of data, for example an entire novel in Swedish (“Kallocain” by Karin Boye), but that experiment did hardly yield one meaningful sentence.

Finally, I fed the program with my collection of recipes (in Swedish), and though not very meaningful, the output was partly quite fun:

> *Skala och pensla skarvarna med ingefäran i den kokande vatten som en stekpanna. Blanda i kylskåp, skär ingefäran, häll ned fläderblomssaft. Blanda alltsammans koka vattnet från ett par veckor eller djupfrysta blåbär. Halvtina renskaven i en stor röd chilifrukt.*

(Peel and paint the joints with the ginger in that boiling water like a frying pan. Mix in refrigerator, cut the ginger, pour down elder juice. Mix everything boil the water from a couple of weeks or deep frozen blueberries. Half defrost the sliced reindeer meat in a big red chili fruit.)

As an adjunct pleasure thou hast thou away,
Of year, that honour from thy scythe and 'Will' in such a bastard be a false plague this book, and death,
But what's in me,
Where I dare to the summer ere long since my love, in the course untrimmed:
From thy neglect I have sworn thee, in our maladies unseen to be seen,

By this were it alteration finds,
Or at grievances foregone,
Unthrifty loveliness why,
As fast as I felt, next self were iniquity.
But wherefore says beauty,
To brief minutes hasten to fears, before the cold.
In the motion sounds,
For joy behind a third nor dare to hear,
Thou art, art the soil is so be it out of all men when swift extremity can be, smell,

Figure 5. 14 lines of text output from the Markov program, trained with Shakespeare’s 154 sonnets.