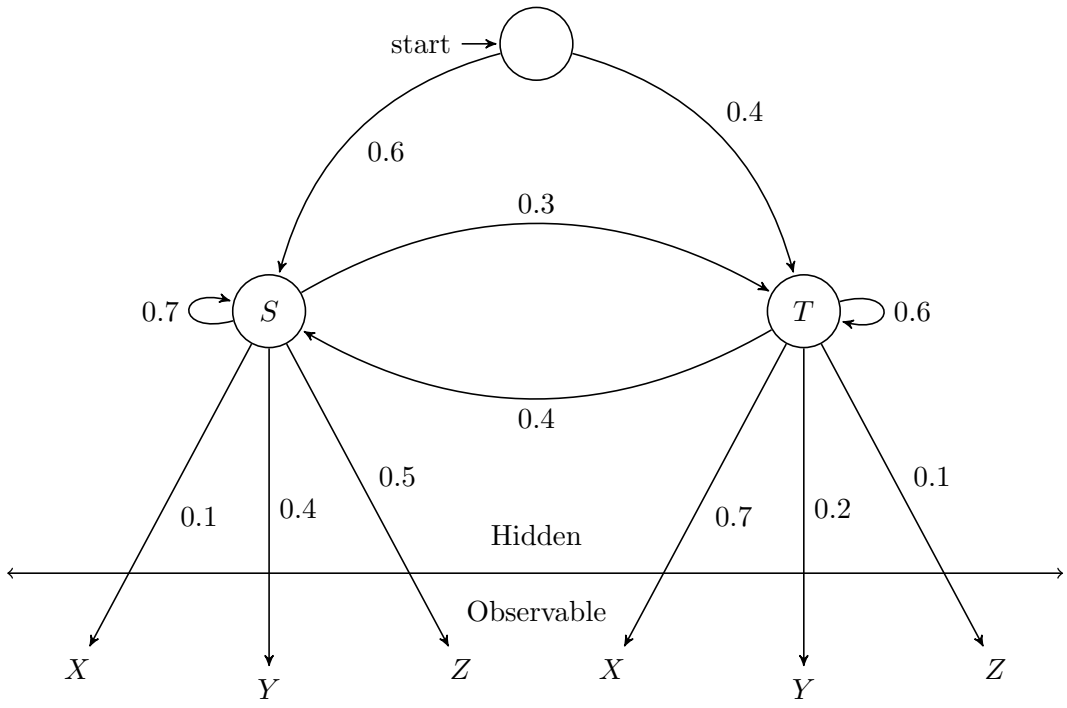
HIDDEN MARKOV MODELS

Hidden Markov Models(HMMs) are statistical tools to model sequential observations with the assumption that states of the system generating them follow a Markov process but these states are unobservable/hidden. However, an observation at any point is related to the underlying hidden state the system is in at that point.

1 Example



In this diagram, nodes $\{S, T\}$ represent the hidden states and $\{X, Y, Z\}$ represent observable states. The unlabelled node is the start node. These states together with the associated probabilities fully characterize the HMM. The state transition probabilites denoted by A can be represented as,

$$A = \begin{array}{c} \\ S \\ T \end{array} \begin{array}{c} \begin{array}{cc} S & T \end{array} \\ \begin{bmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{bmatrix} \end{array}$$

The emmission probabilities are denoted by B.

$$B = \begin{array}{c} \\ S \\ T \end{array} \begin{array}{c} \begin{array}{ccc} X & Y & Z \end{array} \\ \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \end{array}$$

Lastly, initial state distribution is denoted by π.

$$\pi = \begin{array}{c} \begin{array}{cc} S & T \end{array} \\ \begin{bmatrix} 0.6 & 0.4 \end{bmatrix} \end{array}$$

2 The three problems of HMM

2.1 The Evaluation Problem

Given a HMM, $\lambda = (A, B, \pi)$ and a set of observations O, find $P(O|\lambda)$ (probability that the observations were generated by the model).

Consider the HMM in the example above and let $O = (ZXY)$.

$$\begin{aligned} P(ZXY|\lambda) = & P(SSS, ZXY|\lambda) \\ & + P(SST, ZXY|\lambda) \\ & + P(STS, ZXY|\lambda) \\ & + P(STT, ZXY|\lambda) \\ & + P(TSS, ZXY|\lambda) \\ & + P(TST, ZXY|\lambda) \\ & + P(TTS, ZXY|\lambda) \\ & + P(TTT, ZXY|\lambda) \end{aligned}$$

Each of the terms on R.H.S. can be calculated using the following law of probability.

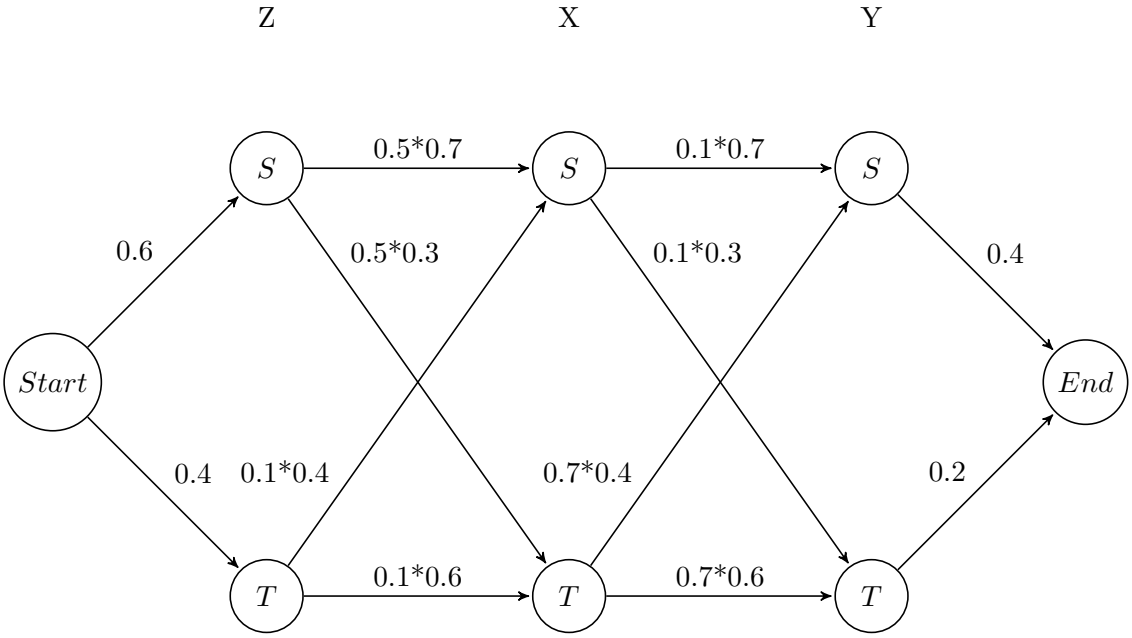
$$\begin{aligned} P(a, b|c) &= \frac{P(a, b, c)}{P(c)} \\ &= \frac{P(a, c)}{P(c)} * \frac{P(a, b, c)}{P(a, c)} \\ &= P(a|c) * P(b|a, c) \end{aligned}$$

Table 1: $P(ZXY|\lambda)$ calculation

HS	OS	$P(HS\|\lambda)$	$P(OS\|HS,\lambda)$	$P(HS,OS\|\lambda)$
SSS	ZXY	0.6*0.7*0.7=0.294	0.5*0.1*0.4=0.020	0.005880
SST	ZXY	0.6*0.7*0.3=0.126	0.5*0.1*0.2=0.010	0.001260
STS	ZXY	0.6*0.3*0.4=0.072	0.5*0.7*0.4=0.140	0.010080
STT	ZXY	0.6*0.3*0.6=0.108	0.5*0.7*0.2=0.070	0.007560
TSS	ZXY	0.4*0.4*0.7=0.112	0.1*0.1*0.4=0.004	0.000448
TST	ZXY	0.4*0.4*0.3=0.048	0.1*0.1*0.2=0.002	0.000096
TTS	ZXY	0.4*0.6*0.4=0.096	0.1*0.7*0.4=0.028	0.002688
TTT	ZXY	0.4*0.6*0.6=0.144	0.1*0.7*0.2=0.014	0.002016
			$P(ZXY\|\lambda)$	0.030028

Calculation in this manner is expensive because probabilities corresponding to all permutations of hidden states of the same length as the length of observed sequence have to be calculated.

To visualize the above calculation, consider the following diagrammatic represntation.

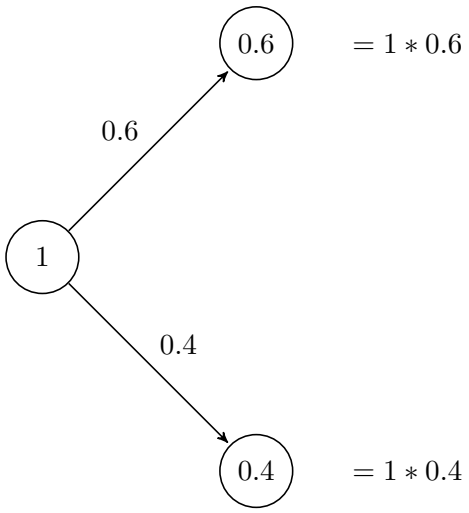


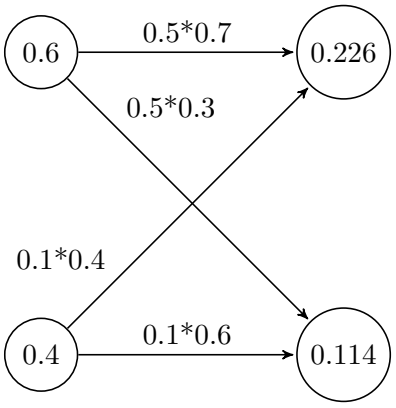
The arrow from S under Z and S under X is labelled 0.5*0.7 as probability of emitting Z given S is 0.5 and probability of transition from S to S is 0.7. The arrow from $Start$ to S is labelled 0.6 as initial probability of transitioning to S is 0.6. Similarly, the arrow from S to End is labelled 0.4 as probability of emitting Y given S is 0.4.

Every path in the diagram from *Start* to *End* has a special meaning in the sense that the multiplication of all labels on the path (which henceforth shall be called its value) is equal to the joint probability of the observation sequence and the hidden state transition sequence defined by the path given the model.

The diagram also leads to the fact that $P(ZXY|\lambda)$ can be thought of as arising out of the enumeration of all paths from *Start* to *End* and then summation over their value.

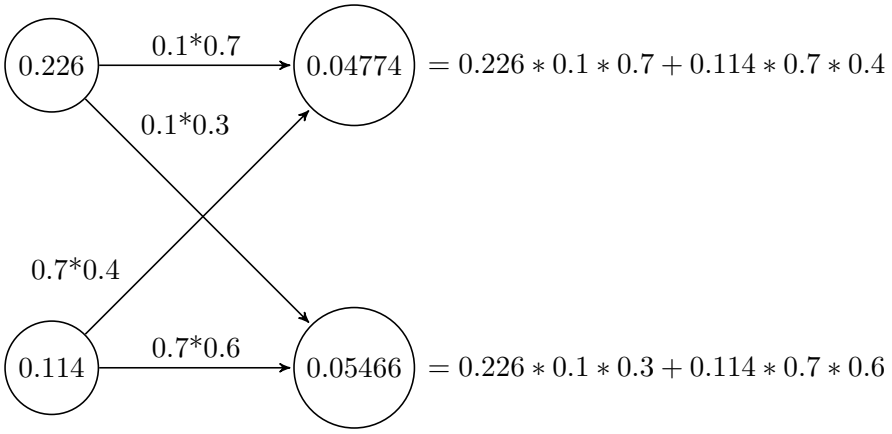
Dynamic Programming can be used to circumvent enumeration and speed up the computation as shown below,

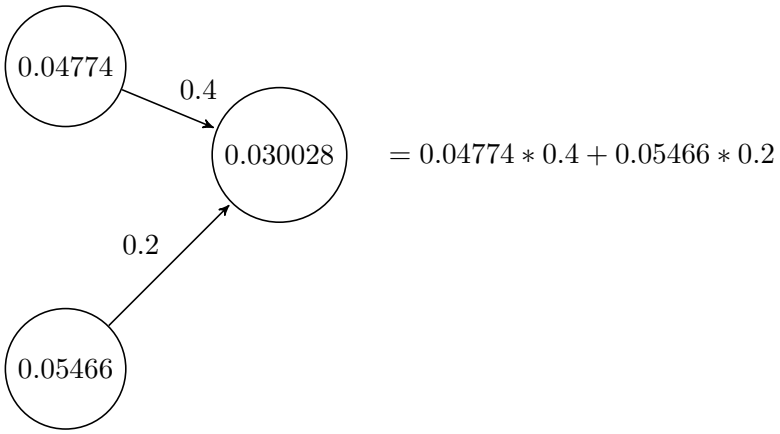




$= 0.6 * 0.5 * 0.7 + 0.4 * 0.1 * 0.4$

$= 0.6 * 0.5 * 0.3 + 0.4 * 0.1 * 0.6$





In this algorithm, each node on the right keeps track of sum of values of all paths from *Start* to that node denoted by $\alpha_t(state)$.

	$t=1$	$t=2$	$t=3$
$\alpha_t(S)$	0.6	0.226	0.04774
$\alpha_t(T)$	0.4	0.114	0.05466

The same algorithm can be run backwards from the *End* node and work its way towards the *Start* node. Now, each node in the left will keep track of sum of values of all paths from *End* to that node denoted by $\beta_t(state)$.

	$t=1$	$t=2$	$t=3$
$\beta_t(S)$	0.0413	0.034	0.4
$\beta_t(T)$	0.01312	0.196	0.2

It is apparent that the following relation must hold,

$$\sum_{\text{all states}} \alpha_t(state) * \beta_t(state) = 0.030028 \ \forall \ t$$

2.2 The Decoding Problem

Given a HMM, $\lambda = (A, B, \pi)$ and a set of observations O, find the most likely sequence of hidden states that led the model to emit O. The solution to this is straightforward. At each time step, the state corrresponding to maximum value of $\alpha_t(state) * \beta_t(state)$ is chosen.

Table 2: Decoding solution

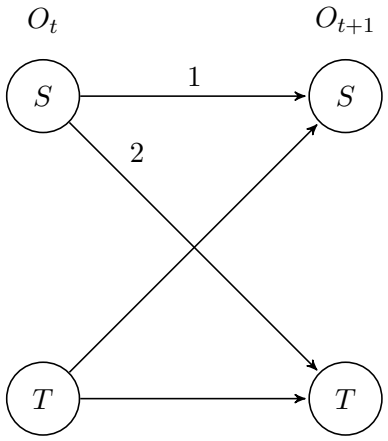
	$t = 1$	$t = 2$	$t = 3$
S	0.6*0.0413	0.226*0.034	0.04774*0.4
T	0.4*0.01312	0.114*0.196	0.05466*0.2
Chosen State	S	T	S

2.3 The Learning Problem

If only the observation sequence O were given, how can one estimate λ? Make a good guess of the number of hidden states and follow:

1. Non uniformly initialize a model $\lambda = (A, B, \pi)$.
2. Compute $\alpha_t(state)$ and $\beta_t(state)$ for all times and states using the solution to Problem 1.
3. Re-estimate the model λ.
4. Go to step 2 if $P(O|\lambda)$ increases reasonably or if out of iterations.

The only question that remains is: How to re-estimate λ?



Remember that $P(O|\lambda)$ is the sum of values if all paths from *Start* to *End*. A fraction of $P(O|\lambda)$ comes from the sum of values of all paths that contain transtion ($state_t = S \rightarrow state_{t+1} = S$) labelled by 1 in the above figure. Denote it by $\gamma_t(S \rightarrow S)$.

A yet another fraction comes from the sum of values of all paths that contain transtion ($state_t = S \rightarrow state_{t+1} = T$) labelled by 2 denoted by $\gamma_t(S \rightarrow T)$. Also let,

$$\gamma_t(S) = \gamma_t(S \rightarrow S) + \gamma_t(S \rightarrow T)$$

Is $\frac{\gamma_t(S \rightarrow S)}{\gamma_t(S)}$ a good estimate of $A(S \rightarrow S)$? Kind of. But remember that only one time t is considered. A better estimate would be,

$$A(S \rightarrow S) \approx \frac{\sum_t \gamma_t(S \rightarrow S)}{\sum_t \gamma_t(S)}$$

Similarly, $B(S \rightarrow OS)$ is estimated by,

$$B(S \rightarrow OS) \approx \frac{\sum_{t|O_t=OS} \gamma_t(S)}{\sum_t \gamma_t(S)}$$

This completes the discussion of the solution to the learning problem in HMMs.

3 Applications

HMMs are widely used in speech recognition, hand writing recognition, part of speech tagging, DNA sequence analysis etc. Wherever there is sequential aspect to data and the data can be thought of as arising out of an underlying simpler hidden probabilistic process, HMMs may be suited for application.