CS 181 LECTURE 4/4/24

Scribe Notes

Contents

1 Hidden Markov Model (HMM)

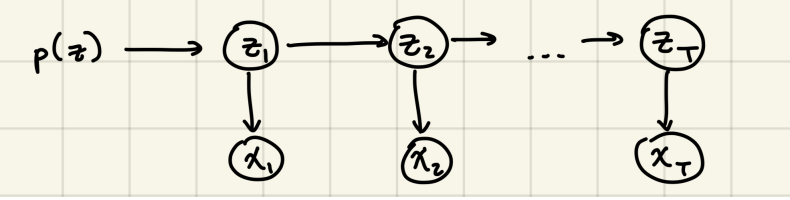


Figure 1: HMM.

In the Hidden Markov Model, we have there are **states,** which are represented by z.

Observations are visible outcomes that we can directly measure. Observations are represented by x.

State transition probabilities are probabilities of moving from one state to another.

Emission probabilities are the likelihoods of observing a particular observation at each point in time given a hidden state.

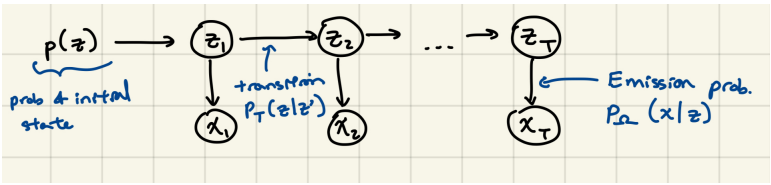


Figure 2: HMM with labels.

Markovian Assumption: the future state of a system depends only on its current state and not on the sequence of events that preceded it.

2 Use Cases

A field where HMMs had a huge impact is natural language processing. The part of speech of the next word depends on the the part of speech of the prior word. In this case, the Markovian assumption is reasonable.

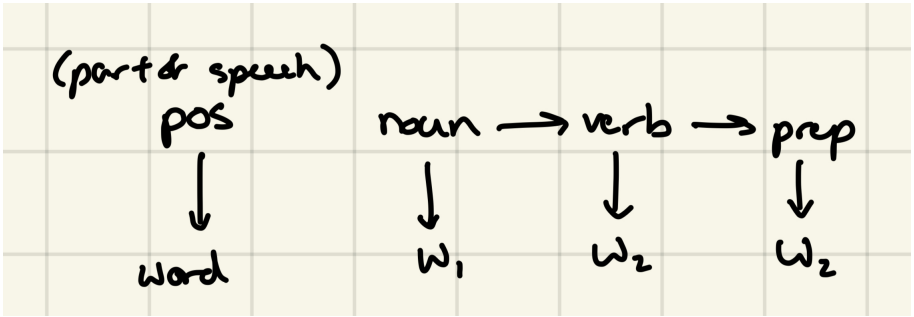


Figure 3: HMM in natural language processing.

Another case is healthcare. The progression of a disease leads to the vitals/symptoms/electronic health record that we observe.

An imfamous case of prediction that teaches us a lesson of not overfitting is the Google Flu Trends. In the early 2010s, Google claimed to be able to predict the flu before the CDC predictions.

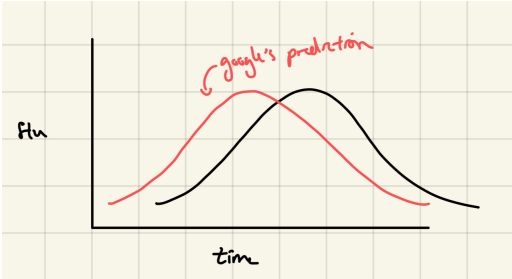


Figure 4: Google Flu Trend.

Using search trends, Google claimed to be able to make a prediciton.

But in 2013, Google missed a huge wave of the flu. Lazer et al. hypothesized that the method was overfitting to certain terms. A second reason is that the search algorithm was changing. After they introduced Google Flu Trends, Google released predictive search. Predictive search finishes a person's sentence as they are typing. Predictive search therefore changes what people may click on while searching.

3 Tasks of Interest

$z \in \{A, B, C\}$

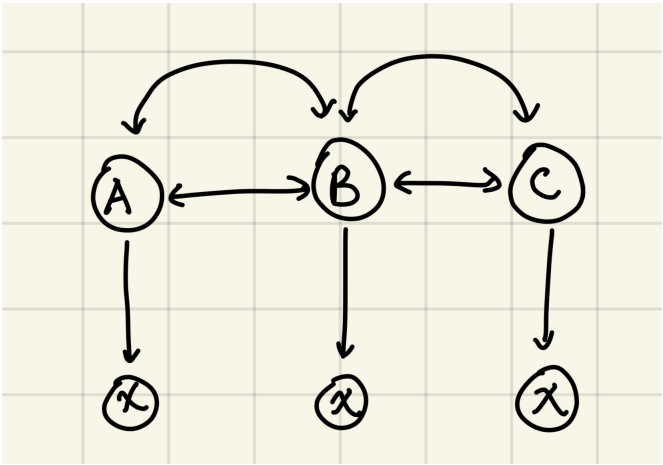


Figure 5: Example HMM.

Some possible tasks are:

- Filtering $p(z_t|x_1, ..., x_c)$, which is asking where am I at this point in time?
- Once we have observed the entire process, we can also ask the question in hindsight: $p(z_t|x_1, ..., x_T)$, this is called smoothing
- We can forecast the next time step: $p(x_{t+1}|x_1, ..., x_t)$

- We might generally ask what is the probability of a trajectory: $p(x_1, ..., x_T)$
- What is the "best path"? What is the most likely hidden state trajectory? $\max_{z_1,...,z_t} p(z_1, ..., z_T|x_1, ..., x_T)$, we call this "decoding"

4 How do we compute these tasks?

The first four tasks have the format of query given evidence, which we have talked about in previous lecture. We can first examine the complete data likelihood because the taks require conditioning and marginalizing out variables in the complete data likelihood:

$$p(z_1, ..., z_T, x_1, ..., x_T)$$

4.1 Forward-Backward Algorithm

We use Bayes rule.

$$p(z_t, x_1, ..., x_T) = p(z_t, x_1, ..., x_t)p(x_{t+1}, ..., x_T|z_t)$$

Forward-pass is $p(z_t, x_1, ..., x_t)$.

Then, the backward-pass is $p(x_{t+1}, ..., x_T|z_t)$.

Let $\alpha_t(z_t = k) = p(x_1, ..., x_t, z_t = k)$. So we can evaluate $\alpha_t(z_t = k)$ for different values of k. Then, we can create a vector of size K for $k \in [K]$.

To think about more general cases, where the k's are continuous instead of discrete, we can consider a random variable z.

$$\begin{aligned}
\alpha_t(z_t = z) &= \sum_{z'} p(x_1, ..., x_t, z_{t-1} = z', z_t = z) \\
&= \sum_{z'} p_\Omega(x_t|z)p_T(z|z')p(x_1, ..., x_{t-1}, z_{t-1} = z') \\
&= p_\Omega(x_t|z) \sum_{z'} p_T(z|z')\alpha_{t-1}(z')
\end{aligned}$$

We see we can use a recursive method. By induction, we can then compute all the values from 1 to t.

We will introduce another set of functions, β. These have a similar form but are forward-facing.

Let $\beta_t(z_t = z) = p(x_{t+1}, ..., x_T|z_t = z)$.

$$\beta_t(z_t = z) = \sum_{z'} p_\Omega(x_{t+1}|z_{t+1})p_T(z|z')p(x_{t+2}, ..., x_T|z_t = z)$$

Again, we can decompose the expression into expressions that we already have and a recursive part.

Observation: We can compute α_t and β_t recursively. α is computed from left to right. β is computed from right to left.

Base case:

$$\alpha_1(z) = p(x_1|z_1)p(z_1)$$

$$\beta_1(z) = 1$$

And this is known as the forward backward algorithm.

4.2 Using the Algorithm for Our Tasks

- Filtering: $p(z_t|x_1, ..., x_t) \propto p(z_t, x_1..., x_t) = \alpha_t(z)$
- Smoothing: $p(z_t|x_1, ..., x_T) \propto p(z_t, x_1, ..., x_T) = \alpha_t(z)\beta_t(z)$
- Probability of a sequence: $p(x_1, ..., x_T) = \sum_z p(x_1, ..., x_T, z_t = z) = \sum_z \alpha_t(z)\beta_t(z)$
- Forecasting: $p(x_{t+1}|x_1, ..., x_t) = \frac{p(x_1,...,x_{t+1})}{p(x_1,...,x_t)} = \frac{\alpha_{t+1}(z)}{\alpha_t(z)}$
- Best Path: this is equivalent to finding the argmax of the joint distribution

$$\max p(z_1, ..., z_T, x_1, ..., x_T)$$

We will introduce another set of variables to help us.

$\gamma(z) = \max_{z_1,...,z_{t-1}} p(x_1, ..., x_t, z_1, ..., z_{t-1}, z_t = z)$

For the first time step, we are looking at just the max of $p(x_1, z_1)$.

$$\gamma_1 = \max p(x_1, z_1) = p_\Omega(x_1, z_1)p(z_1)$$

Then, when we calculate γ_2 we use the best path up to z_2, which is $\max_{z'} \gamma_1(z')p_T(z_2|z')$ and the emission at $t = 2$, which is $p_\Omega(x_2|z_2)$.

$$\gamma_2 = [\max_{z'} \gamma_1(z')p_T(z_2|z')]p_\Omega(x_2|z_2)$$

Once we reach T, we choose the value of z_T that maximizes $\gamma_T(z)$. We call that z_T^*. Then, we go backwards and choose $z_{t-1}^* = argmax_z p(z_t^*|z_{t-1})\gamma_{t-1}(z_{t-1})$

This is the Viterbi algorithm.

5 Parameter Learning

So far, we have done inference, assuming that we already have p_Ω and p_T.

Given z, we can write

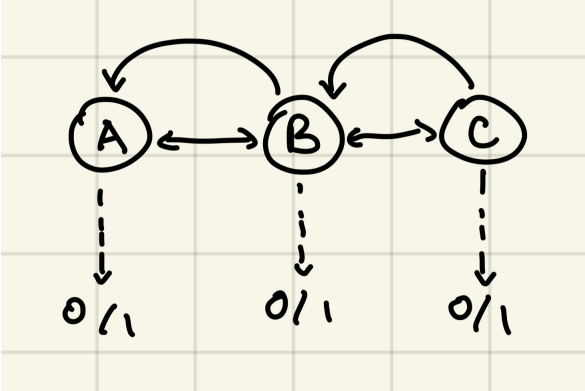
$$p(x_1, ..., x_T, z_1, ..., z_T) = p(z_0) \prod_t p_\Omega(x_t|z_t) \prod_t p_T(z_{t+1}|z_t)$$

$$\log p(x_1, ..., x_T, z_1, ..., z_T) = \log p(z_0) + \sum_t \log p_\Omega(x_t|z_t) + \sum_t \log p_T(z_{t+1}|z_t)$$

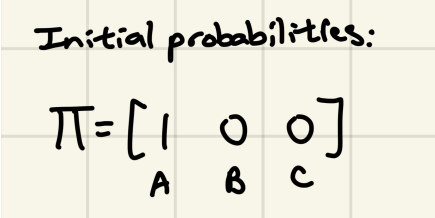
Given p_0, p_t, p_Ω, we can compute $\mathbf{E}[Z]$ with the forward-backward algorithm. We can also compute the maximizing z's through Viterbi. Viterbi is referred to as a "hard" EM while the forward-backward algorithm is regular EM.

Main Takeaway: HMMs have a particularly simple dependence structure. We can partition the future and the past and compute things in a recursive manner.

6 Concept Check

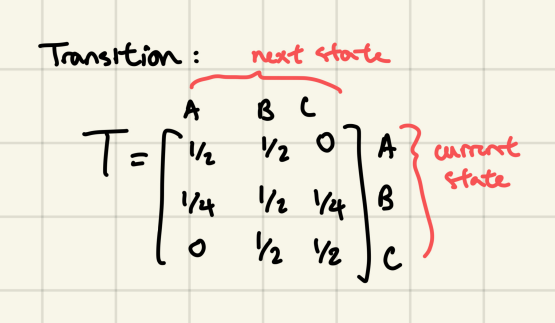


We have initial probability.



Let us say initially, the probability is concentrated on the A state.

Then, we have a transition matrix that tells us the probability from moving from one state to another.



Questions:

1) Assume at time $t = 1$, you observe 0. What is your $p(s_1)$?

2) Assume at time $t = 2$, you observe 0. What is your $p(s_2)$?

3) Write Viterbi path for observed "0", "0."

7 Concept Check Solution

1) Assume at time $t = 1$, you observe 0. What is your $p(s_1)$?

The starting matrix is:

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Then we incorporate the transition matrix:

$$\begin{bmatrix} 1/2 & 1/2 & 0 \end{bmatrix}$$

Then, we compute the joint probability after observing 0:

$$\begin{bmatrix} 0 & 1/4 & 0 \end{bmatrix}$$

Lastly, we normalize the probabilities:

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

2) Assume at time $t = 2$, you observe 0. What is your $p(s_2)$?

We start with this matrix from the last step of the previous question:

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

Then, we incorporate the transition matrix:

$$\begin{bmatrix} 1/4 & 1/2 & 1/4 \end{bmatrix}$$

Next, we observe 0:

$$\begin{bmatrix} 0 & 1/4 & 1/4 \end{bmatrix}$$

Lastly, we normalize so that the probabilities add up to 1:

$$\begin{bmatrix} 0 & 1/2 & 1/2 \end{bmatrix}$$

3) Write Viterbi path for observed "0", "0."

Using the matrices from part 1 and part 2, we see that the possible paths are A, B, C or A, B, B.