Hidden Markov Models

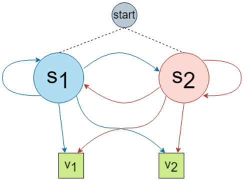
Course: Introduction to AI
Instructor: Saumya Jetley
Teaching Assistant(s): Raghav Awasty & Subhrajit Roy

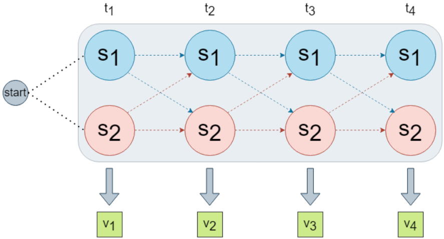
November 2, 2022

Outline

Introduction

Hidden Markov Models are applied to sequential problems of the following kind:





Where,

- observation sequence
- state sequence (underlying and invisible)
- a given state is independent of all historical states given the immediately preceding state

Defining the HMM

- N-hidden States: $S = \{s_1, s_2, \dots, s_N\}$
- M-observable states: $V = \{v_1, v_2, \dots, v_M\}$
- Initial probabilities – probability of state i being the starting state in the sequence
  - $\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$
- Transition probabilities – probability of moving from one hidden state s_i to another s_j
  - $a_{ij} \in A = \{a_{11}, a_{12}, \dots, a_{1N}, \dots, a_{NN}\}$

$$A = \begin{array}{c c} & \begin{array}{cccc} s_1 & s_2 & \dots & s_N \end{array} \\ \begin{array}{c} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_N \end{array} & \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_N \\ a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix} \end{array}$$

Defining the HMM

- Emission probabilities – probability of the generating an output V from a hidden state s_j
  - $b_j(v_k) \in B = \{s_1(v_1), s_1(v_2), \dots, s_1(v_M), \dots, s_N(v_M)\}$

$$B = \begin{array}{c c} & \begin{array}{cccc} v_1 & v_2 & \dots & v_M \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ \vdots \\ s_N \end{array} & \begin{pmatrix} s_1(v_1) & s_1(v_2) & \dots & s_1(v_M) \\ s_2(v_1) & s_2(v_2) & \dots & s_2(v_M) \\ \vdots & \vdots & \ddots & \vdots \\ s_N(v_1) & s_N(v_2) & \dots & s_N(v_M) \end{pmatrix} \end{array}$$

- HMM: $\lambda = (\pi, A, B)$

Markov Assumption

- A Markov chain is a stochastic model describing a sequence of events wherein, the probability of an event occurring only depends on the state in its previous event.
- Formally writing this assumption
  1. The state at time *t* is only dependent on the previous state at *t-1*
     $P(S_t|S_1 \dots S_{t-1}) = P(S_t|S_{t-1})$
     1.1 This leads into the transition probabilities:
     $a_{ij} = P(S_{t+1} = s_j|S_t = s_i)$ for simpler notation, we denote s_i, s_j as i, j
     $a_{ij} = P(S_{t+1} = j|S_t = i)$
  2. The emission at time *t* is only dependent on the hidden state at time *t*
     $P(V_t|S_1 \dots S_t, V_1 \dots V_{t-1}) = P(V_t|S_t)$
     2.1 This leads into the emission probabilities:
     $b_j(v_k) = P(V_t = v_k|S_t = j)$ or $b_j(k) = P(V_t = k|S_t = j)$

Problems to solve

- Observation likelihood
- Most probable hidden state sequence
- Learning HMM parameters

A. Observation Likelihood Computation

P1. Given an HMM λ and an observation sequence V, find the likelihood of observation i.e., $P(V \mid \lambda)$.

Brute Force method

- HMM parameters (λ) operate to yield observation sequences through hidden state sequences
- All hidden state sequences can generate a given observation sequence, but with different probabilities
- Thus, the total probability of observation has to be computed by summing over all possible hidden state sequences.

$$P(V|\lambda) = \sum_{S} P(V, S|\lambda)$$

A. Observation Likelihood Computation

- Let's take a sample observation sequence $V = V_1, V_2, V_3$
- For the hidden state sequence $S = S_1, S_2, S_3$

$$P(V_1 V_2 V_3, S_1 S_2 S_3) = P(S_1 \mid start) \times P(S_2 \mid S_1) \times P(S_3 \mid S_2) \times P(V_1 \mid S_1) \times P(V_2 \mid S_2) \times P(V_3 \mid S_3)$$
$$= \prod_{t=1}^{T} P(V_i \mid S_i) \times \prod_{t=1}^{T} P(S_i \mid S_{i-1})$$

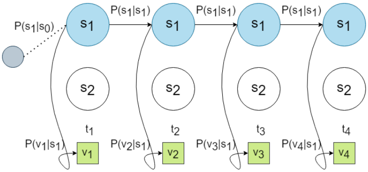
- Thus, the total probability of generating observation V, factorises as follows:

$$\sum_S P(V, S) = \sum_S P(V \mid S) \times P(S)$$
$$= \sum_S \prod_{t=1}^{T} P(V_i \mid S_i) \times \prod_{t=1}^{T} P(S_i \mid S_{i-1})$$

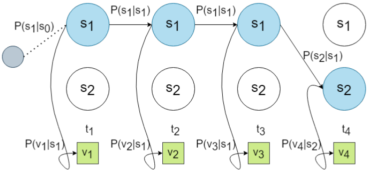
A. Observation Likelihood Computation

Complexity analysis: For an HMM with N hidden states and T observations, we need to sum over N^T**!!** possible hidden sequences to compute the total probability. For large models, this becomes computationally expensive very quickly.

A. Observation Likelihood Computation

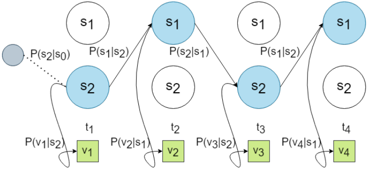


(a) Path 1

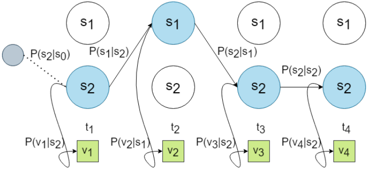


(b) Path 2

⋮



(c) Path 15



(d) Path 16

Figure: Some of the possible paths using the brute force method; N^T computations. Refer to Appendix A for all possible paths.

A. Observation Likelihood Computation

Forward Algorithm (Dynamic Programming)

This method uses a more efficient algorithm that goes on tabulating intermediate values as it build up the probability of the observation sequence, constructed called **forward trellis**

A cell $\alpha_t(j)$ in the trellis at time t represents the probability of being in a state j at time t, having seen previous observations up till time t, given the model λ. This can be represented as the sum over all paths that could lead to this cell.

$$\begin{aligned}
\alpha_t(j) &= P(V_1, V_2 \ldots V_t, S_t = j \mid \lambda) \\
&= \sum_{i=1}^{N} P(V_1, V_2 \ldots V_{t-1}, S_{t-1} = i \mid \lambda) \times P(S_t = j \mid S_{t-1} = i) \times P(V_t \mid S_t = j) \\
&= \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(V_t)
\end{aligned}$$

A. Observation Likelihood Computation

The forward algorithm can be formalised in the following steps:

1. Initialization:

$$\alpha_1(j) = \pi_j b_j(V_1) \quad 1 \leq j \leq N$$

2. Recursion:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(V_t) \quad 1 \leq j \leq N,\ 1 \leq t \leq T$$

3. Termination:

$$P(V \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

A. Observation Likelihood Computation

Complexity analysis

- For an HMM with N hidden states and T observations, each cell in the forward trellis is the sum over all paths leading into it from the previous time step.
- For each cell, there will be N computations thus, for N cells there will be N^2 computations in one time step.
- Over all time steps, there are $N^2 T$ computations.

A. Observation Likelihood Computation

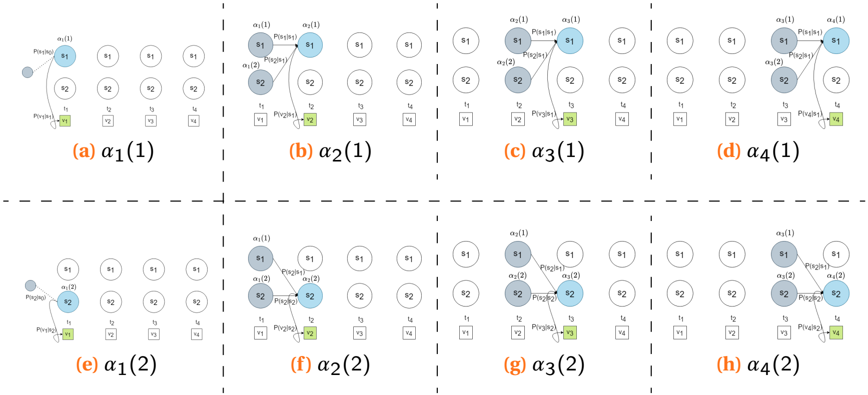


Figure: The trellis filled using the forward algorithm; N^2T computations.

B. Decoding the Hidden State Sequence

P2. Given an HMM λ and an observation sequence $V = V_1, V_2 \ldots V_T$, find the most probable sequence of states i.e., $S = S_1, S_2 \ldots S_T$.

Brute Force method

- Consider the sequence of observations and a potential hidden state sequence and find their joint probability $P(V, S \mid \lambda)$.
- Repeat for every possible hidden state sequence and pick the one that maximizes the probability.
- For N hidden states and T observations → $O(N^T)$ (💣 again)

B. Decoding the Hidden State Sequence

Viterbi Algorithm

Similar to the forward algorithm, this moves through each time step filling up a trellis.

A cell $F_t(j)$ in the trellis at time t represents the probability of being in a state j at time t, having seen previous observations up till time t and passing through the most probable state sequence $S_1 \ldots S_{t-1}$ in the past, given the model λ. This can be represented recursively as the most probable path over all possible paths that could lead to this cell.

$$
\begin{aligned}
F_t(j) &= \max_{S_1 \ldots S_{t-1}} P(V_1 \ldots V_t, S_1 \ldots S_{t-1}, S_t = j \mid \lambda) \\
&= \max_{i=1}^{N} P(V_1 \ldots V_{t-1}, S_1 \ldots S_{t-2}, S_{t-1} = i \mid \lambda) \times P(S_t = j \mid S_{t-1} = i) \times P(V_t \mid S_t = j) \\
&= \max_{i=1}^{N} F_{t-1}(i) a_{ij} b_j(V_t)
\end{aligned}
$$

B. Decoding the Hidden State Sequence

An additional component Viterbi algorithm has over the forward algorithm is **backtracing**.

Backtracing is used to find the most likely hidden state sequence by selecting the state at the previous time step that maximized the probability of reaching the state at the current time step, and doing this recursively.

B. Decoding the Hidden State Sequence

The Viterbi algorithm can be formalised in the following steps:

1. Initialization:

$$F_1(j) = \pi_j b_j(V_1) \quad 1 \leq j \leq N$$

$$bk\_track_1(j) = 0 \quad 1 \leq j \leq N$$

2. Recursion:

$$F_t(j) = \max_{i=1}^{N} F_{t-1}(i) a_{ij} b_j(V_t) \quad 1 \leq j \leq N, 1 \leq t \leq T$$

$$bk\_track_t(j) = \operatorname{argmax}_{i=1}^{N} F_{t-1}(i) a_{ij} b_j(V_t) \quad 1 \leq j \leq N, 1 \leq t \leq T$$

3. Termination:

$$P* = \max_{i=1}^{N} F_T(i)$$

$$S* = \operatorname{argmax}_{i=1}^{N} F_T(i)$$

B. Decoding the Hidden State Sequence

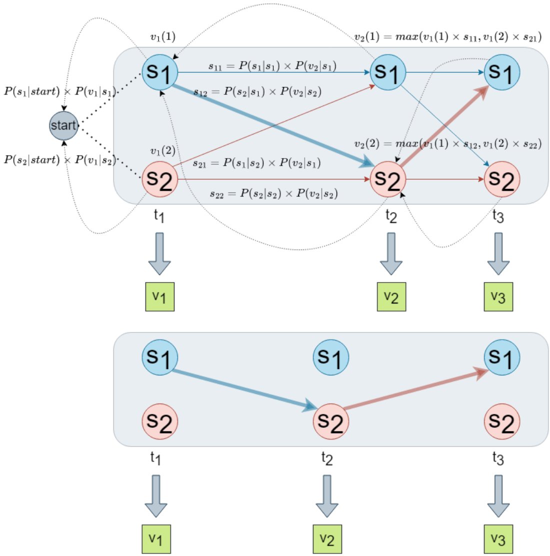


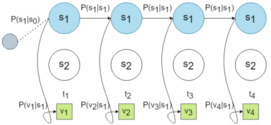
Figure: Top: Propagation of probabilities along the transition paths. Bottom: Backtracing algorithm picking the maximising state from termination to the start.

Overview

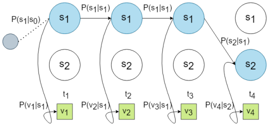
References

- Speech and Language Processing by Daniel Jurafsky and James H. Martin, 2021 (Appendix A)
- Artificial Intelligence - A Modern Approach by Stuart Russell and Peter Norvig, 2010 (Chapter 15.3)
- A Tutorial on Hidden Markov Models by Rakesh Dugad and U. B. Desai, 1996
- An Introduction to Hidden Markov Models by L. R. Rabiner and B. H. Juang, 1986
- Expectation-maximization: theory and intuition by Matthew N. Bernstein, 2020

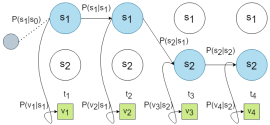
Appendix A



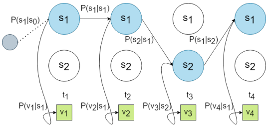
(a) Path 1



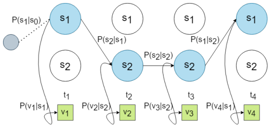
(b) Path 2



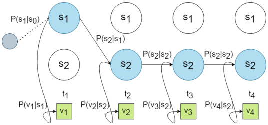
(c) Path 3



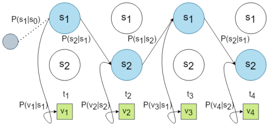
(d) Path 4



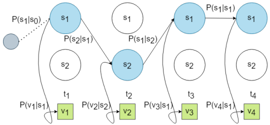
(e) Path 5



(f) Path 6



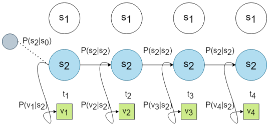
(g) Path 7



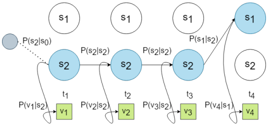
(h) Path 8

Figure: Paths 1-8 out of all possible paths using the brute force method; N^T computations.

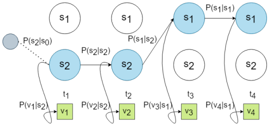
Appendix A



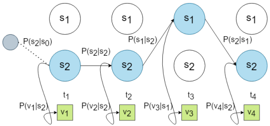
(a) Path 9



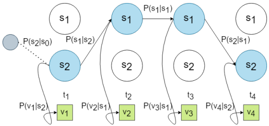
(b) Path 10



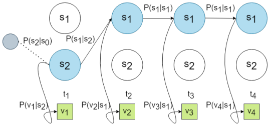
(c) Path 11



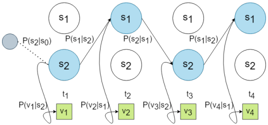
(d) Path 12



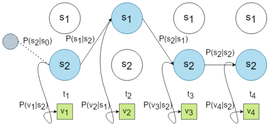
(e) Path 13



(f) Path 14



(g) Path 15



(h) Path 16

Figure: Paths 9-16 out of all possible paths using the brute force method; N^T computations.