Speech Recognition (DSAI 456)

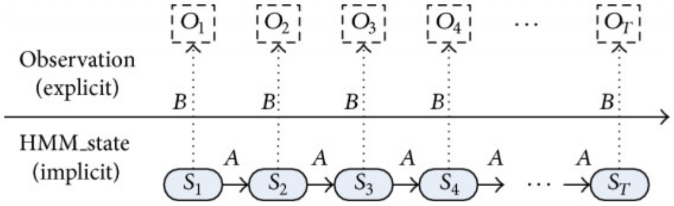
Lecture 7

Mohamed Ghalwash
mghalwash@zewailcity.edu.eg

Recording is NOT allowed

Lecture 5 Recap

- HMM : sequence of observed and hidden states
- HMM model $\lambda = (A, B, \pi)$
- Problems
  - **Evaluation**: $p(O \mid \lambda)$
    - Direct Computation $O(TN^T)$
    - Forward Algorithm $O(TN^2)$
  - **Decoder**: $p(Q|O, \lambda)$
  - **Learning**: $\arg\max_\lambda p(O|\lambda)$



The Decoding Problem $p(Q|O, \lambda)$

Find the most likely sequence of hidden states that produced a given sequence of observations

The HMM Decoding Problem

Given an observation sequence $O = (o_1, o_2, \ldots, o_T)$ and model λ, find the most likely state sequence $Q = (q_1, q_2, \ldots, q_T)$

The HMM Decoding Problem

Given an observation sequence $O = (o_1, o_2, \ldots, o_T)$ and model λ, find the most likely state sequence $Q = (q_1, q_2, \ldots, q_T)$

Define the highest probability of a state sequence ending in state i at time t generating observations $o_1, \ldots, o_t$ as:

$$\delta_t(i) = \max_{q_1, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_{t-1}, q_t = i, o_1, o_2, \ldots, o_t \mid \lambda)$$

The HMM Decoding Problem

Given an observation sequence $O = (o_1, o_2, \ldots, o_T)$ and model λ, find the most likely state sequence $Q = (q_1, q_2, \ldots, q_T)$

Define the highest probability of a state sequence ending in state i at time t generating observations $o_1, \ldots, o_t$ as:

$$\delta_t(i) = \max_{q_1, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_{t-1}, q_t = i, o_1, o_2, \ldots, o_t \mid \lambda)$$

In this case, our objective is to compute $\delta_T(i)$

The HMM Decoding Problem

Given an observation sequence $O = (o_1, o_2, \ldots, o_T)$ and model λ, find the most likely state sequence $Q = (q_1, q_2, \ldots, q_T)$

Define the highest probability of a state sequence ending in state i at time t generating observations $o_1, \ldots, o_t$ as:

$$\delta_t(i) = \max_{q_1, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_{t-1}, q_t = i, o_1, o_2, \ldots, o_t \mid \lambda)$$

In this case, our objective is to compute $\delta_T(i)$

You got it right! 👍 use Induction

Viterbi Algorithm

Viterbi Algorithm

- Initialization: the highest probability along a single path, at time 1, which accounts for the first observation o_1 and ends in state i

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

Viterbi Algorithm

- Initialization: the highest probability along a single path, at time 1, which accounts for the first observation o_1 and ends in state i

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

- Assume $\delta_t(i)$ is given, calculate $\delta_{t+1}(j)$

Viterbi Algorithm

- Initialization: the highest probability along a single path, at time 1, which accounts for the first observation o_1 and ends in state i

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

- Assume $\delta_t(i)$ is given, calculate $\delta_{t+1}(j)$
- Induction (for $t = 2, \ldots, T$):

$$\delta_{t+1}(j) = \max_{1 \leq i \leq N} \left[\delta_t(i) a_{ij}\right] b_j(o_t), \quad 1 \leq j \leq N$$

Viterbi Algorithm

- Initialization: the highest probability along a single path, at time 1, which accounts for the first observation o_1 and ends in state i

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

- Assume $\delta_t(i)$ is given, calculate $\delta_{t+1}(j)$

- Induction (for $t = 2, \ldots, T$):

$$\delta_{t+1}(j) = \max_{1 \leq i \leq N} \left[\delta_t(i) a_{ij}\right] b_j(o_t), \quad 1 \leq j \leq N$$

- Termination:

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

Viterbi Algorithm

- Initialization: the highest probability along a single path, at time 1, which accounts for the first observation o_1 and ends in state i

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N$$

- Assume $\delta_t(i)$ is given, calculate $\delta_{t+1}(j)$
- Induction (for $t = 2, \ldots, T$):

$$\delta_{t+1}(j) = \max_{1 \le i \le N} \left[\delta_t(i) a_{ij}\right] b_j(o_t), \quad 1 \le j \le N$$

- Termination:

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

Where is the state sequence? 😢

We need to keep track of selected states along the path

Viterbi Algorithm

Viterbi Algorithm

- Initialization

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$
$$\psi_1(i) = 0$$

Viterbi Algorithm

- Initialization

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$
$$\psi_1(i) = 0$$

- Induction

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t), \quad 1 \leq j \leq N$$
$$\psi_t(j) = \arg\max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right]$$

Viterbi Algorithm

- Initialization

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$
$$\psi_1(i) = 0$$

- Induction

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t), \quad 1 \leq j \leq N$$
$$\psi_t(j) = \arg\max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right]$$

- Termination

$$P^* = \max_{1 \leq i \leq N} \delta_T(i)$$
$$q_T^* = \arg\max_{1 \leq i \leq N} \delta_T(i)$$

Viterbi Algorithm

- Initialization

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N$$
$$\psi_1(i) = 0$$

- Induction

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t), \quad 1 \le j \le N$$
$$\psi_t(j) = \arg\max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right]$$

- Termination

$$P^* = \max_{1 \le i \le N} \delta_T(i)$$
$$q_T^* = \arg\max_{1 \le i \le N} \delta_T(i)$$

Backtrack from q_T^* to q_1^* using $\psi_t(i)$ to get the most probable state sequence

The Viterbi algorithm efficiently solves the decoding problem with complexity $O(N^2T)$

Learn More

Course Homepage