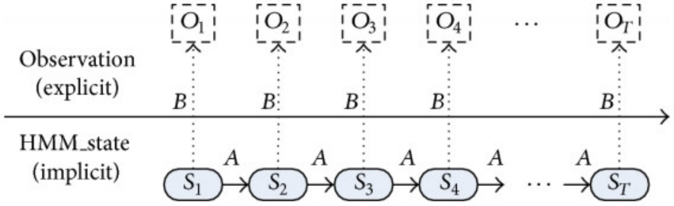
Speech Recognition (DSAI 456)

Lecture 6

Mohamed Ghalwash
mghalwash@zewailcity.edu.eg

Lecture 5 Recap

- HMM : sequence of observed and hidden states
- HMM model $\lambda = (A, B, \pi)$
- Problems
  - Evaluation: $p(O \mid \lambda)$
  - Decoder: $p(Q|O, \lambda)$
  - Learning: $\arg\max_{\lambda} p(O|\lambda)$



Agenda

- Evaluation
- Decoder
- Learning

Evaluation Problem of HMM - Direct Computation

$$P(O \mid \lambda) = \sum_{all\ Q} P(O \mid Q, \lambda) P(Q|\lambda)$$

Evaluation Problem of HMM - Direct Computation

$$P(O \mid \lambda) = \sum_{all\ Q} P(O \mid Q, \lambda) P(Q|\lambda)$$

$$P(O \mid Q, \lambda) = b_1(o_1)\ .\ b_2(o_2)\ .\ b_3(o_3) \ldots b_T(o_T)$$

$$P(Q \mid \lambda) = \pi_1\ .\ a_{12}\ .\ a_{23}\ .\ a_{34} \ldots a_{(T-1)T}$$

Then $P(O, Q \mid \lambda) = \pi_1 b_1(o_1) \prod_{t=2}^{T} a_{(t-1)t} b_t(o_t)$

Evaluation Problem of HMM - Direct Computation

$$P(O \mid \lambda) = \sum_{all\ Q} P(O \mid Q, \lambda) P(Q|\lambda)$$

$$P(O \mid Q, \lambda) = b_1(o_1)\ .\ b_2(o_2)\ .\ b_3(o_3) \ldots b_T(o_T)$$

$$P(Q \mid \lambda) = \pi_1\ .\ a_{12}\ .\ a_{23}\ .\ a_{34} \ldots a_{(T-1)T}$$

Then $P(O, Q \mid \lambda) = \pi_1 b_1(o_1) \prod_{t=2}^{T} a_{(t-1)t} b_t(o_t)$

- Initially we are at time 1, we are in state q_1 with probability π_1
- Generate the symbol o_1 with probability $b_1(o_1)$
- Make a transition from state q_1 to state q_2 with probability a_{12}
- Generate the symbol o_2 with probability $b_2(o_2)$
- Continue until the last state

====

Evaluation Problem of HMM - Direct Computation

$$P(O \mid \lambda) = \sum_{all\ Q} P(O \mid Q, \lambda) P(Q|\lambda)$$

$$P(O \mid Q, \lambda) = b_1(o_1)\ .\ b_2(o_2)\ .\ b_3(o_3) \ldots b_T(o_T)$$

$$P(Q \mid \lambda) = \pi_1\ .\ a_{12}\ .\ a_{23}\ .\ a_{34} \ldots a_{(T-1)T}$$

Then $P(O, Q \mid \lambda) = \pi_1 b_1(o_1) \prod_{t=2}^{T} a_{(t-1)t} b_t(o_t)$

- Initially we are at time 1, we are in state q_1 with probability π_1
- Generate the symbol o_1 with probability $b_1(o_1)$
- Make a transition from state q_1 to state q_2 with probability a_{12}
- Generate the symbol o_2 with probability $b_2(o_2)$
- Continue until the last state

====

There are N^T possible state sequences, each requires $2T$ calculations → exponential complexity

Can we find a faster algorithm to compute the likelihood?

Evaluation Problem of HMM - Forward Algorithm

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

- Define probability of partial observation $o_1, o_2, \ldots o_t$ and being in state i at time t given λ as

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

- Define probability of partial observation $o_1, o_2, ... o_t$ and being in state i at time t given λ as

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i \mid \lambda)$$

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

- Define probability of partial observation $o_1, o_2, \ldots o_t$ and being in state i at time t given λ as

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i \mid \lambda)$$

- At last time point T:

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

- Define probability of partial observation $o_1, o_2, \dots o_t$ and being in state i at time t given λ as

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = i \mid \lambda)$$

- At last time point T:

$$\alpha_T(i) = P(o_1, o_2, \dots, o_T, q_T = i \mid \lambda) = p(O, q_T = i \mid \lambda)$$

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

- Define probability of partial observation $o_1, o_2, \ldots o_t$ and being in state i at time t given λ as

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i \mid \lambda)$$

- At last time point T:

$$\alpha_T(i) = P(o_1, o_2, \ldots, o_T, q_T = i \mid \lambda) = p(O, q_T = i \mid \lambda)$$

- What we need is:

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

- Define probability of partial observation $o_1, o_2, \ldots o_t$ and being in state i at time t given λ as

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i \mid \lambda)$$

- At last time point T:

$$\alpha_T(i) = P(o_1, o_2, \ldots, o_T, q_T = i \mid \lambda) = p(O, q_T = i \mid \lambda)$$

- What we need is:

$$p(O \mid \lambda) = \sum_i p(O, q_T = i \mid \lambda) = \sum_i \alpha_T(i)$$

Evaluation Problem of HMM - Forward Algorithm

A very nice trick is using **induction** or recursion

- Define probability of partial observation $o_1, o_2, \ldots o_t$ and being in state i at time t given λ as

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i \mid \lambda)$$

- At last time point T:

$$\alpha_T(i) = P(o_1, o_2, \ldots, o_T, q_T = i \mid \lambda) = p(O, q_T = i \mid \lambda)$$

- What we need is:

$$p(O \mid \lambda) = \sum_i p(O, q_T = i \mid \lambda) = \sum_i \alpha_T(i)$$

How to compute $\alpha_t(i)$? inductively

Evaluation Problem of HMM - Forward Algorithm

Evaluation Problem of HMM - Forward Algorithm

- Initialization: initialize the forward probabilities as the joint probability of state i and initial observation o_1

Evaluation Problem of HMM - Forward Algorithm

- Initialization: initialize the forward probabilities as the joint probability of state i and initial observation o_1

$$\alpha_1(i) = P(o_1, q_1 = i \mid \lambda)$$

Evaluation Problem of HMM - Forward Algorithm

- Initialization: initialize the forward probabilities as the joint probability of state i and initial observation o_1

$$\alpha_1(i) = P(o_1, q_1 = i \mid \lambda)$$

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

Evaluation Problem of HMM - Forward Algorithm

- Initialization: initialize the forward probabilities as the joint probability of state i and initial observation o_1

$$\alpha_1(i) = P(o_1, q_1 = i \mid \lambda)$$

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

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

Evaluation Problem of HMM - Forward Algorithm

- Initialization: initialize the forward probabilities as the joint probability of state i and initial observation o_1

$$\alpha_1(i) = P(o_1, q_1 = i \mid \lambda)$$

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

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

- Induction: how state j be reached at time $t+1$ from N possible states at time t. Summation over all possible states i at time t, transition to state j at $t+1$, multiplied by emission probability at $t+1$

Evaluation Problem of HMM - Forward Algorithm

- Initialization: initialize the forward probabilities as the joint probability of state i and initial observation o_1

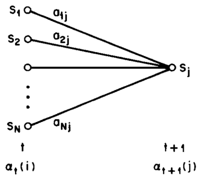
$$\alpha_1(i) = P(o_1, q_1 = i \mid \lambda)$$

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N$$

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

- Induction: how state j be reached at time $t+1$ from N possible states at time t. Summation over all possible states i at time t, transition to state j at $t+1$, multiplied by emission probability at $t+1$

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) b_j(o_{t+1}), \quad 1 \le j \le N,\ 1 \le t \le T-1$$



Evaluation Problem of HMM - Forward Algorithm

Evaluation Problem of HMM - Forward Algorithm

- Termination: Sum of probabilities over all states at final time

Evaluation Problem of HMM - Forward Algorithm

- Termination: Sum of probabilities over all states at final time

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

Evaluation Problem of HMM - Forward Algorithm

- Termination: Sum of probabilities over all states at final time

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

- Computing $\alpha_t(i)$ using above steps enables efficient likelihood evaluation with complexity $O(N^2T)$

Now we improved the time complexity of the likelihood evaluation $p(O \mid \lambda)$ from $N^T T$ to $N^2 T$ using Forward algorithm (induction trick)

Learn More

Course Homepage