Graphical Models:

Hidden Markov Models: Learning & Inference

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Implement HMM learning & inference algorithms

Basic Problems in HMM

For a given HMM $\Lambda=\{\Theta,\Omega,A,B,\pi\}$

- Problem 1: Given an observation (sequence) $\boldsymbol{O}=\{o^1,o^2, \dots ,o^k\}$, what is the most likely state sequence $\boldsymbol{S}=\{s^1,s^2, \dots ,s^k\}$ that has produced $\boldsymbol{O}$?

- Problem 2: How likely is an observation $\boldsymbol{O}$ (i.e., what is $P(\boldsymbol{O})$) ?

- Problem 3: How to estimate the model parameters (A,B,π)?

Problem 1: State Estimation

- Given an observation (sequence) $\boldsymbol{O}=\{o^1,o^2, \dots ,o^k\}$, what is the most likely state sequence $\boldsymbol{S}=\{s^1,s^2, \dots ,s^k\}$ that has produced $\boldsymbol{O}$?

Formally, we need to solve

$$\underset{\boldsymbol{S}}{\operatorname{argmax}} P(\boldsymbol{S}|\boldsymbol{O})$$

Or, equivalently,

$$\underset{\boldsymbol{S}}{\operatorname{argmax}} \frac{P(\boldsymbol{S},\boldsymbol{O})}{P(\boldsymbol{O})} = \underset{\boldsymbol{S}}{\operatorname{argmax}} P(\boldsymbol{S},\boldsymbol{O})$$

Problem 1: State Estimation (cont'd)

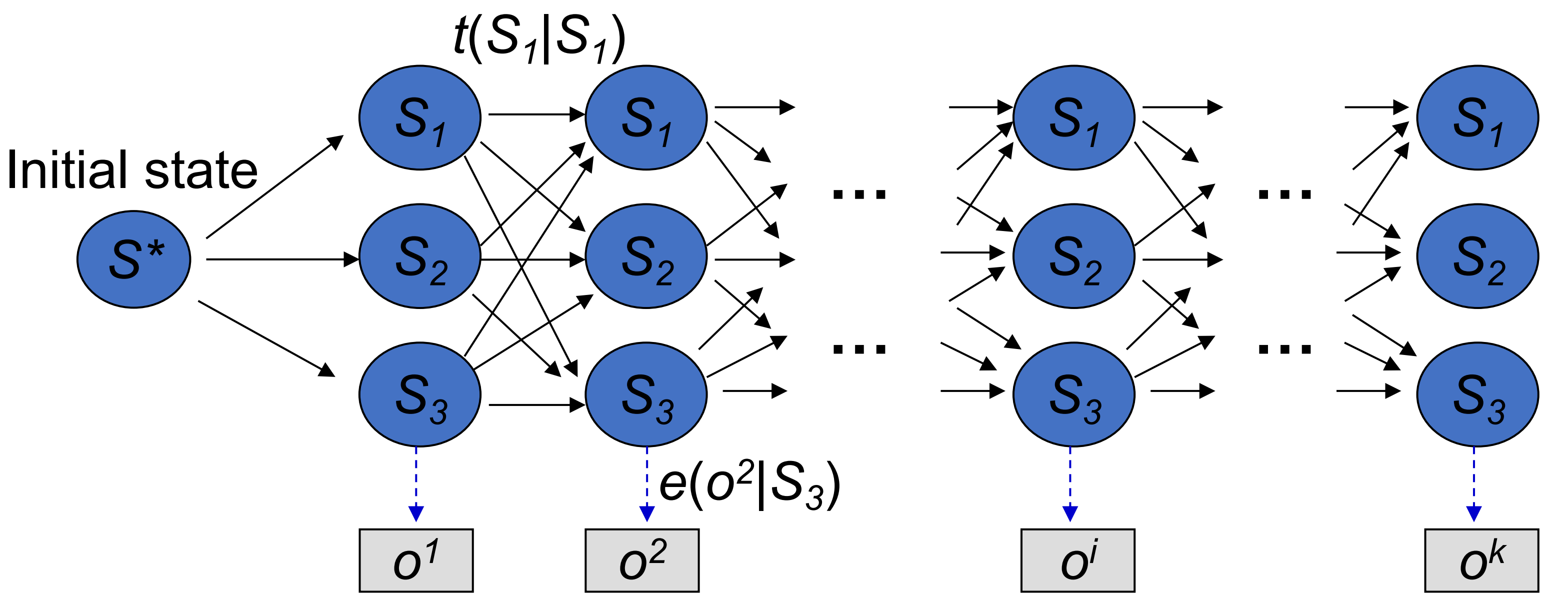
For a given HMM, we may simplify $P(\boldsymbol{S},\boldsymbol{O})$ as

$$
\begin{aligned}
P(\boldsymbol{S},\boldsymbol{O}) &= P(\boldsymbol{O}|\boldsymbol{S})P(\boldsymbol{S}) \\
&= P(o^1 \ldots o^k | s^1 \ldots s^k) \prod_{j=1}^{k} P(s^j | s^1 \ldots s^{j-1}) \\
&\simeq P(o^1 \ldots o^k | s^1 \ldots s^k) \prod_{j=1}^{k} P(s^j | s^{j-1}) \\
&= \prod_{i=1}^{k} P(o^i | o^1 \ldots o^{i-1}, s^1 \ldots s^i) \prod_{j=1}^{k} P(s^j | s^{j-1}) \\
&\simeq \prod_{i=1}^{k} P(o^i | s^i) \prod_{j=1}^{k} P(s^j | s^{j-1}) = \prod_{i=1}^{k} P(o^i | s^i)\, P(s^i | s^{i-1})
\end{aligned}
$$

The "Weather" Example

Let's expand the state space as a trellis, for the earlier example:

S_1-rain, S_2-cloudy, S_3-sunny



-- $t(.|.)$ is the transition probability and $e(.|.)$ the emission probability.

➔ To identify a path for which the product of t's and the e's is maximized.

Viterbi Algorithm for Problem 1

A dynamic programming solution

- For each state in the trellis, we record:

1. $\delta_{s_i}(t)$ is the probability of taking the maximal path up to time *t-1* ending at state S_i at time *t* and while generating $o^1 \ldots o^t$

2. $\psi_{s_i}(t)$ is the state sequence that resulted in the maximal probability up to state S_i at time *t*.

Viterbi Algorithm (cont'd)

1. Initialization

$$\delta_{S_i}(1) = t(S_i|s^*)e(o^1|S_i), \quad \forall S_i \in \Theta$$

2. Induction:
   for 2≤*t*≤*k*, do

$$\delta_{S_i}(t) = \max_{S_j} t(S_i|S_j)e(o^t|S_i)\delta_{S_j}(t-1)$$

$$\psi_{S_i}(t) = \operatorname*{argmax}_{S_j} t(S_i|S_j)e(o^t|S_i)\delta_{S_j}(t-1)$$

3. Termination: The probability of the best state sequence $\max_{S_j} \delta_{S_j}(k)$

The best last state $\hat{s}^k = \operatorname*{argmax}_{S_j} \delta_{S_j}(k)$

Back trace to get other states:

$$\hat{s}^t = \psi_{\hat{s}^{t+1}}(t), \text{ for } t = k-1, \ldots, 1.$$

Problem 2: Evaluate $P(\boldsymbol{O})$

- To evaluate $P(O)$, we can do $P(\boldsymbol{O}) = \sum_{\boldsymbol{S}} P(\boldsymbol{S}, \boldsymbol{O})$

- From the trellis, a solution can be found by summing the probabilities of all paths generating the given observation sequence.

- A dynamic programming solution: <u>the forward algorithm</u> or the backward algorithm.

The Forward Algorithm

Define the forward probability $\alpha_{S_i}(t)$, which is the probability for all paths up to time *t-1* ending at state S_i at time *t* and generating $o^1 \ldots o^t$.

1. Initialization: $$\alpha_{S_i}(1) = t(S_i|s^*)e(o^1|S_i), \quad \forall S_i \in \Theta$$

2. Induction:
   for $2 \le t \le k$, do $$\alpha_{S_i}(t) = \sum_{S_j} t(S_i|S_j)e(o^t|S_i)\alpha_{S_j}(t-1)$$

3. Termination: $$P(\boldsymbol{O}) = \sum_{S_j} \alpha_{S_j}(k)$$

Problem 3: Parameter Learning

- Case 1: we have a set of labeled data – sequences in which we have the <state, observation> information
  - Use relative frequency for estimating the probabilities

    → the MLE solution

$$t(S_i|S_j) = \frac{\text{number of } (s^t = S_i, s^{t-1} = S_j)}{\text{number of } S_j} \qquad e(o_r|S_j) = \frac{\text{number of } (o^t = o_r, s^t = S_j)}{\text{number of } S_j}$$

- Case 2: we have only the observation sequence
  - The Forward-Backward Algorithm (a.k.a. Baum-Welch Algorithm): An EM approach.