


Hidden Markov Models

Chen Yu
Indiana University

Markov Property



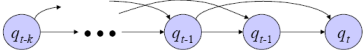
1st-order Markov model

q_t represents the state at time t

$$P(q_t | q_{t-1}, q_{t-2}, \dots) = P(q_t | q_{t-1})$$

$$P(q_t, q_{t-1}, q_{t-2}, \dots) = P(q_t | q_{t-1})P(q_{t-1} | q_{t-2}) \dots$$

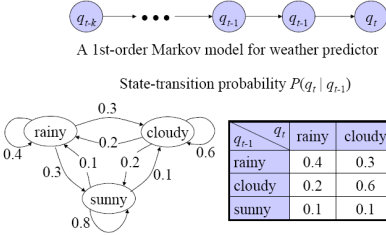
Markov Model



2nd-order Markov model

$$P(q_t | q_{t-1}, q_{t-2}, \dots) = P(q_t | q_{t-1}, q_{t-2})$$

An Example

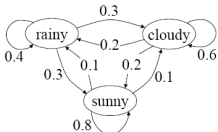


A 1st-order Markov model for weather predictor

State-transition probability $P(q_t | q_{t-1})$

$q_{t-1} \backslash q_t$	rainy	cloudy	sunny
rainy	0.4	0.3	0.3
cloudy	0.2	0.6	0.2
sunny	0.1	0.1	0.8

P(Observation | Model)



Question: given the day 1 is sunny, what is the probability that the weather for the next 7 days will be "sun-sun-rain-rain-sun-cloudy-sun"?

$$P(O|Model) = P[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|Model]$$

$$= P[S_3] \cdot P[S_3|S_3] \cdot P[S_3|S_3] \cdot P[S_1|S_3]$$

$$\cdot P[S_1|S_1] \cdot P[S_3|S_1] \cdot P[S_2|S_3] \cdot P[S_3|S_2]$$

Question: given the model is in a known state, what is the probability it stays in that state for exactly d days?

$$O = \{S_1, S_1, S_1, \dots, S_d, S_d \neq S_{d+1}\}$$

$$P(O|Model, q_1 = S_i) = (a_{ii})^{d-1}(1 - a_{ii})$$

Hidden Markov Models

Hidden states
Observed states

- Graphical Model
- Circles indicate states
- Arrows indicate probabilistic dependencies between states

Coin Toss Models

$O = O_1 O_2 O_3 \dots O_T$
 $= \mathcal{H} \mathcal{H} \mathcal{H} \mathcal{T} \mathcal{T} \mathcal{H} \mathcal{H} \mathcal{H} \mathcal{H} \mathcal{H} \mathcal{H} \mathcal{H} \dots \mathcal{H}$

where \mathcal{H} stands for heads and \mathcal{T} stands for tails.

$O = \text{H H T T H T H H T T H} \dots$ $O = \text{H H T T H T H H T T H} \dots$ $O = \text{H H T T H T H H T T H} \dots$
 $S = 1 1 2 2 1 2 1 1 2 2 1 \dots$ $S = 2 1 1 2 2 2 1 2 2 1 2 \dots$ $S = 3 1 2 3 3 1 1 2 3 1 3 \dots$

Urn and Ball Model

$P(\text{RED}) = b_1(1)$ $P(\text{RED}) = b_2(1)$ $P(\text{RED}) = b_N(1)$
 $P(\text{BLUE}) = b_1(2)$ $P(\text{BLUE}) = b_2(2)$ $P(\text{BLUE}) = b_N(2)$
 $P(\text{GREEN}) = b_1(3)$ $P(\text{GREEN}) = b_2(3)$ $P(\text{GREEN}) = b_N(3)$
 $P(\text{YELLOW}) = b_1(4)$ $P(\text{YELLOW}) = b_2(4)$ $P(\text{YELLOW}) = b_N(4)$
 \vdots \vdots \vdots
 $P(\text{ORANGE}) = b_1(M)$ $P(\text{ORANGE}) = b_2(M)$ $P(\text{ORANGE}) = b_N(M)$

$O = \{\text{GREEN, GREEN, BLUE, RED, YELLOW, RED, \dots, BLUE}\}$

Generative model: the first step is to select a state and the second step is to select a color ball.

Elements in HMM

$S = \{1, 2, \dots, N\}$ – all possible values of hidden states.
 o_t – The observation at time t
 $V = \{1, 2, \dots, M\}$ – all possible values of observed states.

$A = [a_{ij}]_{i,j}$ – state transition probabilities.
 $a_{ij} = P(q_{t+1} = j | q_t = i), \quad a_{ij} \geq 0, \quad 1 \leq i, j \leq N, \quad \sum_j a_{ij} = 1.$

$B = [b_{ik}]_{i,k}$ – emission probabilities.
 $b_{ik} = P(o_t = k | q_t = i), \quad b_{ik} \geq 0, \quad 1 \leq i \leq N \ \& \ 1 \leq k \leq M, \quad \sum_k b_{ik} = 1.$

$\pi = \{\pi_i\}$ – initial state probabilities. $\pi_i = P(q_1 = i), \sum_i \pi_i = 1.$

$\lambda = (A, B, \pi)$

Problem 1

Problem 1: Given the observation sequence $O = O_1 O_2 \dots O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

How well a given model matches a given observation sequence. If we consider the case in which we are trying to choose among several competing models, the solution to Problem 1 allows us to choose the model which best matches the observations.

Problem 2

Problem 2: Given the observation sequence $O = O_1 O_2 \dots O_T$, and the model λ , how do we choose a corresponding state sequence $Q = q_1 q_2 \dots q_T$ which is optimal in some meaningful sense (i.e., best “explains” the observations)?

Problem 3

Problem 3: How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

Solution to Problem 1

Given $O = o_1 o_2 \dots o_T$ and λ , compute $P(O|\lambda)$.

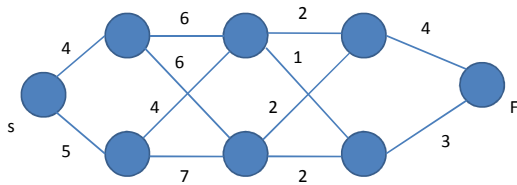
$$P(O|\lambda) = \sum_Q P(O, Q|\lambda) = \sum_Q P(O|Q, \lambda) P(Q|\lambda)$$

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t | q_t, \lambda) \quad P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

However, there are N^T possible hidden state sequences!

There is an efficient way – dynamic programming.

Dynamic Programming

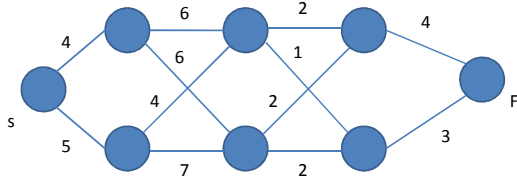


From S to F, 4 steps, two states in each step.

Principle of Optimality (Bellman 1965)

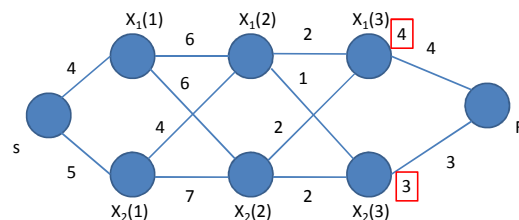
- From any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point.
 - Problem 1: from 0 to T
 - Problem 2: from t1 to T
 - Problem 3: from 0 to t1
- Any intermediate point in the optimal path must be the optimal point linking the optimal partial paths before and after that point.

Dynamic Programming



- At each step we need to make a decision, up or down.
- The basic idea of principle optimality is that we proceed backward and at each step, we find the best path at this time, from the current state to the destination.
- At each step, we go back one more step back and deal with the sub-problem based on previous solutions.

Step 1



$$\text{Step 1: } t = 3 \quad J(x_1(3), 3) = 4$$

$$J(x_2(3), 3) = 3$$

Step 2

$$J(x_1(2), 2) = \min \begin{pmatrix} J(x_1(3), 3) + D(x_1(2), x_1(3)) \\ J(x_2(3), 3) + D(x_1(2), x_2(3)) \end{pmatrix}$$

$$J(x_2(2), 2) = \min \begin{pmatrix} J(x_1(3), 3) + D(x_2(2), x_1(3)) \\ J(x_2(3), 3) + D(x_2(2), x_2(3)) \end{pmatrix}$$

Step 3

$$J(x_1(1), 1) = \min \begin{pmatrix} J(x_1(2), 2) + D(x_1(1), x_1(2)) \\ J(x_2(2), 2) + D(x_1(1), x_2(2)) \end{pmatrix}$$

$$J(x_2(1), 1) = \min \begin{pmatrix} J(x_1(2), 2) + D(x_2(1), x_1(2)) \\ J(x_2(2), 2) + D(x_2(1), x_2(2)) \end{pmatrix}$$

Step 4

$$J(x_1(1), 1) = \min \begin{pmatrix} J(x_1(2), 2) + D(x_1(1), x_1(2)) \\ J(x_2(2), 2) + D(x_1(1), x_2(2)) \end{pmatrix}$$

$$J(x_2(1), 1) = \min \begin{pmatrix} J(x_1(2), 2) + D(x_2(1), x_1(2)) \\ J(x_2(2), 2) + D(x_2(1), x_2(2)) \end{pmatrix}$$

The forward Procedure

$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda)$

The probability of the partial observation sequence and state S_i at time t given the model.

(1) Initialize
 $\alpha_1(i) = \pi_i b_{i, o_1}, \quad 1 \leq i \leq N$

(2) Induction
 $\alpha_t(i) = [\sum_{j=1}^N \alpha_{t-1}(j) a_{ji}] b_{i, o_t}, \quad 2 \leq t \leq T$
 $1 \leq i \leq N$

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

$\lambda = \{A, B, \pi\} \quad P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \quad \alpha_t(i) = [\sum_{j=1}^N \alpha_{t-1}(j) a_{ji}] b_{i, o_t}$

$\pi \Rightarrow$

$A = [a_{ij}]_{i,j}$

$B = [b_{i,k}]_{i,k}$

$\Sigma \rightarrow P(O | \lambda)$

The backward Procedure

Analogous to the forward variable, define a backward variable

$\beta_t(i) = P(o_{t+1} \dots o_T | q_t = S_i, \lambda)$

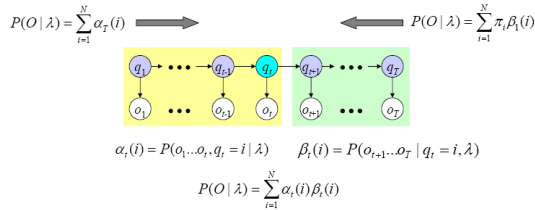
Probability of the partial observation, given state S_i at time t and the model.

(1) Initialize
 $\beta_T(i) = 1, \quad 1 \leq i \leq N$

(2) Induction
 $\beta_t(i) = \sum_{j=1}^N a_{ij} b_{j, o_{t+1}} \beta_{t+1}(j), \quad 1 \leq t \leq T-1$
 $1 \leq i \leq N$

(3) Termination
 $P(O | \lambda) = \sum_{i=1}^N \pi_i \beta_1(i)$

The forward-backward Procedure



Solution to find optimal states

$$\gamma_t(i) = P(q_t = S_i | O, \lambda)$$

The probability of being in state S_i at time t , given the observation sequence O , and the model.

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

$$q_t = \underset{1 \leq i \leq N}{\operatorname{argmax}} [\gamma_t(i)], \quad 1 \leq t \leq T.$$

A better Solution

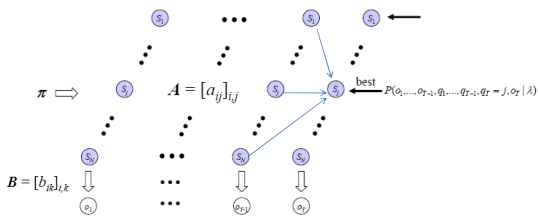
$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_t = i, O_1, O_2, \dots, O_t | \lambda]$$

The highest probability along a state sequence that accounts for the first t observations and ends at state S_i .

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}).$$

The Viterbi Algorithm

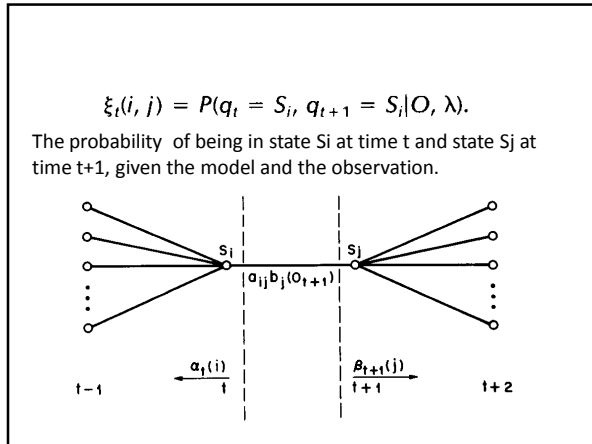
1. Initialization
 $\delta_1(i) = \pi_i b_{i o_1}$ and $\psi_1(i) = 0, \quad 1 \leq i \leq N$
2. for $t = 1$ to $T-1$
 $\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_{j o_{t+1}}$
 $\psi_{t+1}(j) = \underset{i}{\operatorname{argmax}} \delta_t(i) a_{ij} b_{j o_{t+1}}$
endfor
3. Termination
 $P^* = \max_{1 \leq i \leq N} \delta_T(i) \quad q_T^* = \underset{1 \leq i \leq N}{\operatorname{argmax}} \delta_T(i)$
4. State sequence backtracking
 $q_t^* = \psi_{t+1}(q_{t+1}^*)$



Solution to Problem 3

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}} P(O | \lambda)$$

Easy if the hidden states are known.



Baum-Welch Method

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}$$

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j). \quad \gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

$\sum_{t=1}^{T-1} \gamma_t(i)$ = expected number of transitions from S_i

$\sum_{t=1}^{T-1} \xi_t(i, j)$ = expected number of transitions from S_i to S_j .

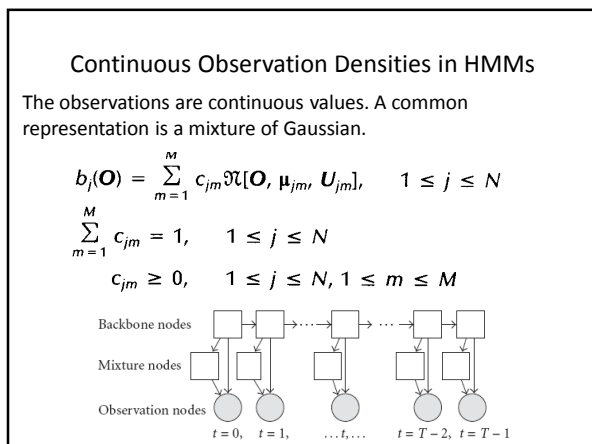
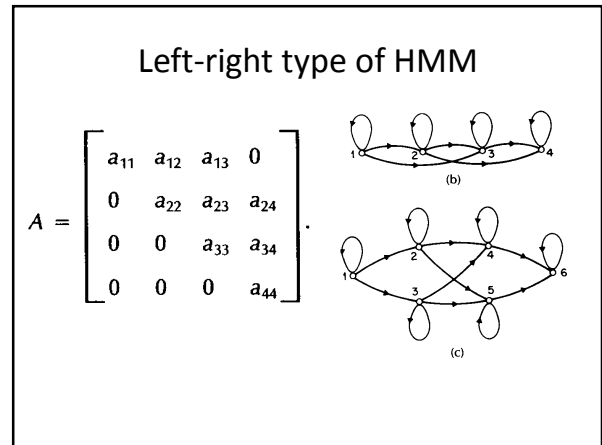
\bar{n}_i = expected frequency (number of times) in state S_i at time $(t = 1) = \gamma_t(i)$

$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i}$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$

$$= \frac{\sum_{t=1}^T \gamma_t(j) \cdot \mathbb{1}_{\{O_t = v_k\}}}{\sum_{t=1}^T \gamma_t(j)}$$



Training

$$\bar{c}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)}$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot O_t}{\sum_{t=1}^T \gamma_t(j, k)}$$

$$\bar{U}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot (O_t - \mu_{jk})(O_t - \mu_{jk})'}{\sum_{t=1}^T \gamma_t(j, k)}$$

$$\gamma_t(j, k) = \left[\frac{\alpha_t(j) \beta_t(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} \right] \left[\frac{c_{jk} \mathcal{N}(O_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^M c_{jm} \mathcal{N}(O_t, \mu_{jm}, U_{jm})} \right]$$

The probability of being in state j at time t with the k th mixture component account for O_t .