

# **CS4770**

## **Pattern Recognition**

### **Hidden Markov Models**

**P. J. Narayanan**  
Monsoon 2006

# Markov Processes/Chains

- What is the likelihood of a sequence of events occurring?
- Stochasting modelling: Use Markov chains or processes.
- Consists of  $N$  states and arcs connecting them indicating state transitions.
- **State transition probability matrix** gives the probabilities of going from one state to another.
- Let  $s_t$  indicate the state at time  $t$ .

- Markov process when only the previous state matters:  
$$P(s_t = i | s_{t-1} = j, s_{t-2} = k, \dots) = P(s_t = i | s_{t-1} = j).$$
- It also doesn't depend on time. The transition probability between states  $i$  and  $J$  given by:  $P(s_t = j | s_{t-1} = i) = a_{ij}$ .
- Also,  $\sum_j a_{ij} = 1$  since it has to be in some state!
- Let us look at weather in Hyderabad. There are 3 states: sunny, cloudy, rainy. Observe state every 2 hours during the day.
- Known transition probabilities: sunny to cloudy, cloudy to rainy, rainy to sunny, rainy to cloudy, etc.

- The model consists of: States, transition matrix, starting state.
- Probability of observing a sequence SSSCRRRCS given the model:  $\pi(S) a_{SS} a_{SS} a_{SC} a_{CR} a_{RR} a_{RR} a_{RC} a_{CS}$
- Probability of getting exactly  $d$  consecutive measurements being rainy:  $p_r(d) = a_{rr}^{d-1}(1 - a_{rr})$ .
- Expected length of sunny state:  $\sum_{d=1}^{\infty} d p_s(d)$

# Hidden States

- What if the states remain hidden, away from observation?
- We can observe another random process with its own probability distribution.
- Observation is a stochastic process of the hidden state:  
**Hidden Markov Model.**
- A coin toss experiment is performed in the next room and the result (H or T) is read out aloud.
- Observed sequence: HTTTHT. How do we build an HMM to explain the observations?

# Single Coin Model

- Use one biased coin.
- Two states labelled H and T.
- Only 1 tunable parameter. Probability of H:  $p$
- $a_{hh} = a_{th} = p$  and  $a_{ht} = a_{tt} = 1 - p$ .
- $P(\text{HTTTHT}) = p(1 - p)(1 - p)(1 - p)p(1 - p)$ .
- No real transition probabilities. One state will do.

# Another Model

- Toss 3 coins. If the first coin is H, read out 2nd coin. Else read out the 3rd coin.
- Only 3 parameters:  $p_1, p_2, p_3$  for heads for the coins. Let  $p'_1 = 1 - p_1, p'_2 = 1 - p_2, p'_3 = 1 - p_3$ .
- $P(HT) = (p_1p_2 + p'_1p_3)(p_1p'_2 + p'_1p'_3)$ .
- No real transition probabilities. One state will do.

# Two Coin Model

- Use 2 coins with probabilities  $p_1$  and  $p_2$  for heads.
- 2 states: Use coin 1 in state 1 and coin 2 in state 2.
- Independent transition probabilities between states:  $a_{12}$  and  $a_{21}$ .
- Four tunable parameters:  $p_1, p_2, a_{12}, a_{21}$ .  
Let  $p'_1 = 1 - p_1, a'_{12} = 1 - a_{12}$ , etc.
- $P(HT) = \pi(1)p_1(a'_{12}p'_1 + a_{12}p'_2) + \pi(2)p_2(a'_{21}p'_2 + a_{21}p'_1)$
- Similarly a 3-coin model with 3 states and 9 parameters.

# Hidden Markov Model: Formulation

- A set of  $c$  states:  $\{\omega_1, \omega_2, \dots, \omega_c\}$ .
- Transition probabilities: (Given by a matrix  $A$ )  
 $a_{ij} = P(\omega_j(t+1)|\omega_i(t)), 1 \leq i, j \leq c$ . Also,  $\sum_j a_{ij} = 1$ .
- A set of  $k$  observable symbols:  $\{v_1, v_2, \dots, v_k\}$ .
- Emission probabilities: (Given by a matrix  $B$ )  
 $b_{ij} = P(v_j|\omega_i), 1 \leq i \leq c, 1 \leq j \leq k$ . Also,  $\sum_j b_{ij} = 1$ .
- Initial probabilities (for starting state):  $\pi(i), 1 \leq i \leq c$ .
- An optional final or absorbing state  $\omega_f$  such that  $a_{ff} = 1$ .
- HMM model:  $\Theta = (A, B, \pi)$ .

# HMM: 3 Basic Problems

- **Evaluation:** Given  $\Theta$ , compute the probability of a sequence of observables  $V^T$ .
- **Decoding:** Given an HMM and a set  $V^T$  of observations, determine the most likely sequence of hidden states  $\omega^T$  that generated the observations.
- **Learning:** Given a number of training observations of visible symbols and the structure of the HMM ( $c$  and  $k$ ), determine the parameters  $a_{ij}$  and  $b_{ij}$ .

# Evaluation

- Given  $\Theta$ , compute the probability of a sequence of observables  $\mathbf{V}^T = \mathbf{v}(1), \mathbf{v}(2), \dots, \mathbf{v}(T)$
- Consider every sequence of states and sum their individual probabilities of generating the observations.
- $P(\mathbf{V}^T) = \sum_{i=1}^{r_M} P(\mathbf{V}^T | \omega_r^T) \mathbf{P}(\omega_r^T)$ , where  $r_M = c^T$ , the number of possible sequences of length  $T$ .
- $P(\omega_r^T) = \prod_{i=1}^T P(\omega(t) | \omega(t-1)) = \prod a_{ij}$ , for consecutive states in the sequence.
- $P(\mathbf{V}^T | \omega_r^T) = \prod_{i=1}^T P(v(t) | \omega(t)) = \prod b_{ij}$ , for the state-observation pairs.

- Simple algorithm: Evaluate by summing all sequences. Complexity:  $O(Tc^T)$ .
- $\alpha_j(t)$ : probability of state  $j$  at time  $t$ , having generated the first  $t$  visible symbols of the sequence.  
Initialization:  $\alpha_i(0) = 1$  for initial state; 0 for others.  
(Alternately,  $\alpha_i(1) = \pi(i)b_{iv(1)}$ )
- $\alpha_j(t) = [\sum_i \alpha_i(t-1)a_{ij}] b_{jv(t)}$  where  $v(t) = v_k$ , the symbol generated at time  $t$ .
- Similarly,  $\beta_j(t) = \sum_i \beta_i(t+1)a_{ji}b_{iv(t+1)}$  gives the probability that the rest of the sequence (after  $t$ ) will be generated by the HMM. Initialization:  $\beta_f(T) = 1$  for the final state; 0 for others. (Alternately,  $\beta_i(T) = 1, \forall i$ )

# Forward Algorithm

Initialize  $a_{ij}, b_{jk}, \mathbf{V}^T, \alpha_j(0)$

for  $t = 1$  to  $T$  do

$$\alpha_j(t) = b_{jv(t)} \sum_{i=1}^c \alpha_i(t-1) a_{ij}, \quad 1 \leq j \leq c$$

$P(\mathbf{V}^T) = \alpha_f(T)$  for the final state  $f$

- Complexity:  $O(c^2T)$ .
- An unfolding of the states over time in  $T$  steps.
- Compute forward the probabilities of being at state  $j$  and generating the inputs upto that time.

# Backward Algorithm

Initialize  $a_{ij}, b_{jk}, \mathbf{V}^T, \alpha_j(T)$

for  $t = T - 1$  to  $1$  do

$$\beta_j(t) = \sum_{i=1}^c \beta_i(t+1) a_{ij} b_{jv(t+1)}, \quad 1 \leq j \leq c$$

$P(\mathbf{V}^T) = \beta_i(T)$  for the initial state  $i$

- Complexity:  $O(c^2T)$ .
- An unfolding of the states backwards over time in  $T$  steps.
- Compute backward the probabilities of being at state  $j$  and generating the inputs from that time till end.

# Decoding

- Choose the optimum state at each time step.
- Let  $\gamma_i(t) = P(s(t) = \omega_i | \mathbf{V}, \Theta) = \frac{\alpha_i(t)\beta_i(t)}{\sum_i \alpha_i(t)\beta_i(t)}$
- At each time  $t$ , select  $\omega_k(t)$  where  $k(t) = \arg \max_i \gamma_i(t)$ .
- Individually optimum. But, the sequence generated can have states  $i$  and  $j$  consecutively where  $a_{ij} = 0!!$
- Optimum sequence may not be a valid one.

# Simple Algorithm

Initialize  $a_{ij}, b_{jk}, \mathbf{V}^T, \alpha_j(0)$   
for  $t = 1$  to  $T$  do  
  for  $j = 1$  to  $c$  do  
     $\alpha_j(t) \leftarrow b_{jv(t)} \sum_{i=1}^c \alpha_i(t-1) a_{ij}$   
     $k(t) \leftarrow \arg \max_j \alpha_j(t)$   
    append  $\omega_{k(t)}$  to the path  
Return path

- Finds the maximum probability state at each time step. Path found may not be a valid one.
- Solution: A dynamic programming algorithm to maximize the sequence probability.

# Viterbi Algorithm

- Let  $\delta_i(t) = \max P(s(1) \cdots s(t) = \omega_i, v(1) \cdots v(t) | \Theta)$ , most likely path that accounts for first  $t$  symbols and ends in  $i$ .
- We can see:  $\delta_j(t + 1) = \max_i [\delta_i(t) a_{ij}] b_{jv(t+1)}$
- Initialize:  $\delta_i(1) = \pi(i) b_{iv(1)}$ ,  $\psi_i(1) = 0, \forall i$
- $\forall j > 1$ :  $\psi_j(t) = \arg \max_i (\delta_i(t - 1) a_{ij})$   
 $\delta_j(t + 1) = \max_i (\delta_i(t) a_{ij}) b_{jv(t+1)}$

- At the end,  $p^* = \max_i \delta_i(T)$  gives the optimum probability.  
 $q^*(T) = \arg \max_i \delta_i(T)$  is the last state.
- Backtrack:  $q^*(t) = \psi_{q^*(t+1)}(t+1)$ ,  $t = (T-1), \dots, 1$  to recover the states.
- A forward dynamic programming algorithm. Similar to Dijkstra's shortest path algorithm.
- Very similar to the forward algorithm with **max** replacing summation.

# Learning Problem

- $\gamma_{ij}(t) = \frac{\alpha_i(t-1) a_{ij} b_{jv(t)} \beta_j(t)}{P(\mathbf{V}^T | \Theta)}$  gives probability that the sequence used  $i$  to  $j$  transition at time  $t$ .  
(Where  $P(\mathbf{V}^T | \Theta) = \sum_i \sum_j \alpha_i(t-1) a_{ij} b_{jv(t)} \beta_j(t)$ ).
- Probability of using  $\omega_i$  at time  $t$ :  $\gamma_i(t) = \sum_{j=1}^c \gamma_{ij}(t)$ .
- Total expected number of  $i$  to  $j$  transitions:  $\sum_{t=1}^{T-1} \gamma_{ij}(t)$ .
- Total expected number of transitions from  $i$ :  $\sum_{t=1}^{T-1} \gamma_i(t)$ .
- Estimate for  $a_{ij}$ :  $\hat{a}_{ij} = \frac{\# \text{ transitions from } i \text{ to } j}{\# \text{ transitions from } i} = \frac{\sum_t \gamma_{ij}(t)}{\sum_t \sum_j \gamma_{ij}(t)}$

- Similarly:  $\hat{b}_{jk} = \frac{\sum_{t=1}^T \gamma_i(t) \text{ when } v(t)=v_k}{\sum_{t=1}^T \gamma_i(t)}$
- And,  $\pi(i) = \gamma_i(1)$
- Called the **forward-backward** algorithm.
- Input: Initialization of  $A, B, \pi$ , sequence  $\mathbf{V}^T$  accepted by the HMM.
- Iteratively compute  $\hat{A}, \hat{B}$ , and  $\hat{\pi}$  till values converge.
- Probability of observing  $\mathbf{V}^T$  improves. Final result is a maximum-likelihood estimate of the HMM.
- In practice: Random or uniform estimates for  $A, \pi$  will suffice. Good initial estimates are more critical for  $B$ .