Lecture 36
Decoding of HMMs
ECEN 5283 Computer Vision

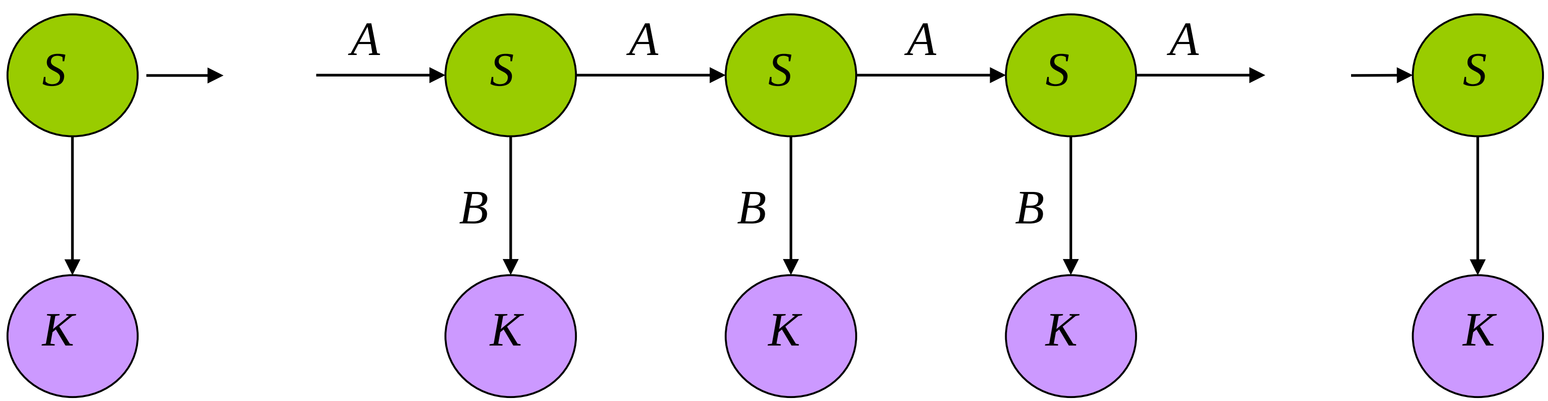
Dr. Guoliang Fan
School of Electrical and Computer Engineering
Oklahoma State University

Goals

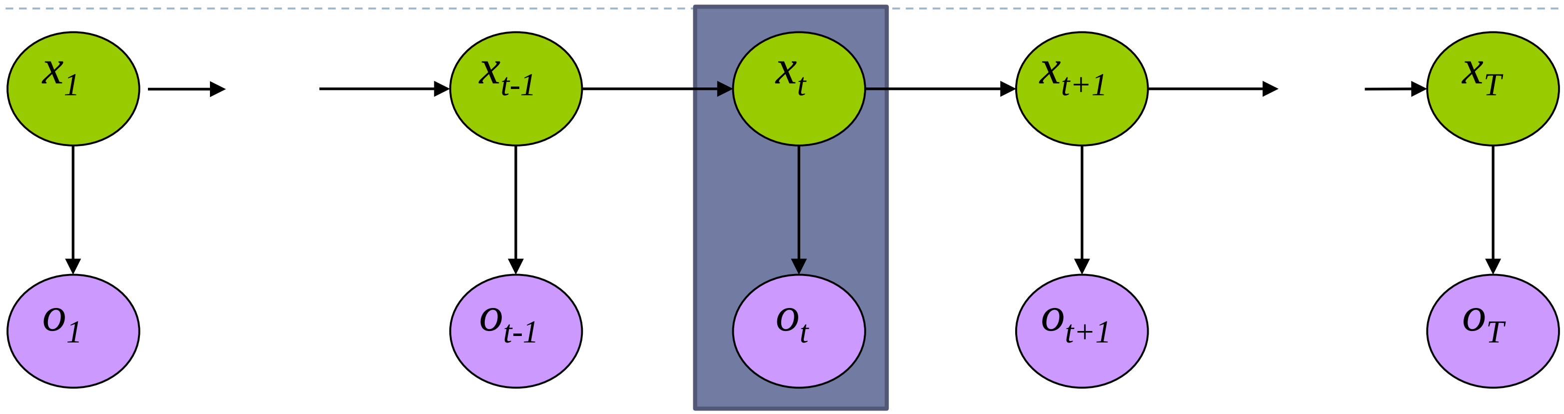
- To review the HMMs.
- To review the evaluation problem of HMMs.
- To discuss the decoding problem of HMMs.

HMM Parameterization

- $\mu=\{S, K, \Pi, A, B\}$
- $S : \{s_1 \ldots s_N\}$ are the values for the hidden states
- $K : \{k_1 \ldots k_M\}$ are the values for the observations
- $\Pi = \{\pi_i\}$ are the initial state probabilities
- $A = \{a_{ij}\}$ are the state transition probabilities
- $B = \{b_{ik}\}$ are the observation state probabilities.
- **Two conditional independent assumptions**



Evaluation of HMMs



$$P(O \mid \mu) = \sum_X P(O, X \mid \mu) = \sum_X \pi_{x_1} b_{x_1}(o_1) \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1}}(o_{t+1})$$

$O = \{o_1, o_2, o_3, \ldots o_T\}$

All possible state sequences

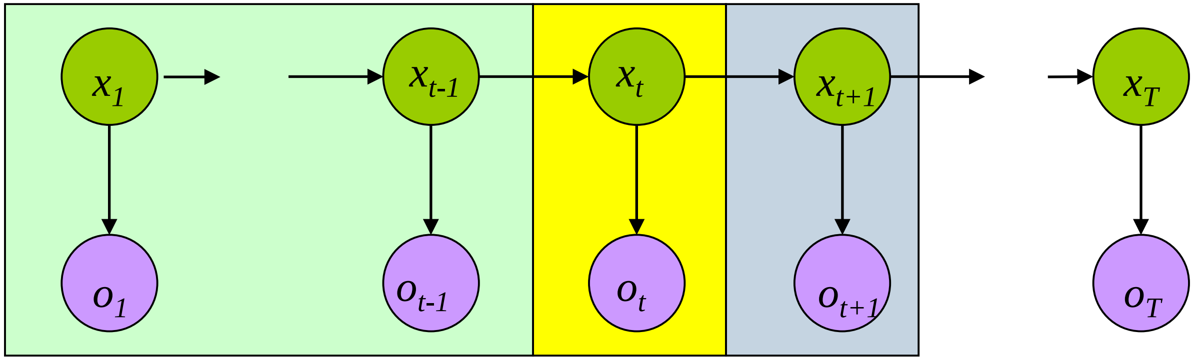
How many possible state sequences? N^T

Therefore, we introduce two terms that can be computed recursively either forward or background to evaluate HMMs at a linear complexity.

$$\alpha_t(i) = P(o_1 \ldots o_t, x_t = i) \qquad \beta_t(i) = P(o_{t+1} \ldots o_T \mid x_t = i)$$

$$\alpha_t(i)\beta_t(i) = P(o_1 \ldots o_t, o_{t+1} \ldots o_T, x_t = i) \rightarrow P(O \mid \mu) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$$

Evaluation of HMMs: Forward Process



$$\alpha_t(i) = P(o_1 \ldots o_t, x_t = i)$$

$$\alpha_t(i) \rightarrow \alpha_{t+1}(j)$$

$$(i, j = 1, \ldots, N)$$

$$\alpha_1(i) = P(o_1, x_1 = i) = P(o_1 \mid x_1 = i) P(x_1 = i) = \pi_i b_i(o_1)$$

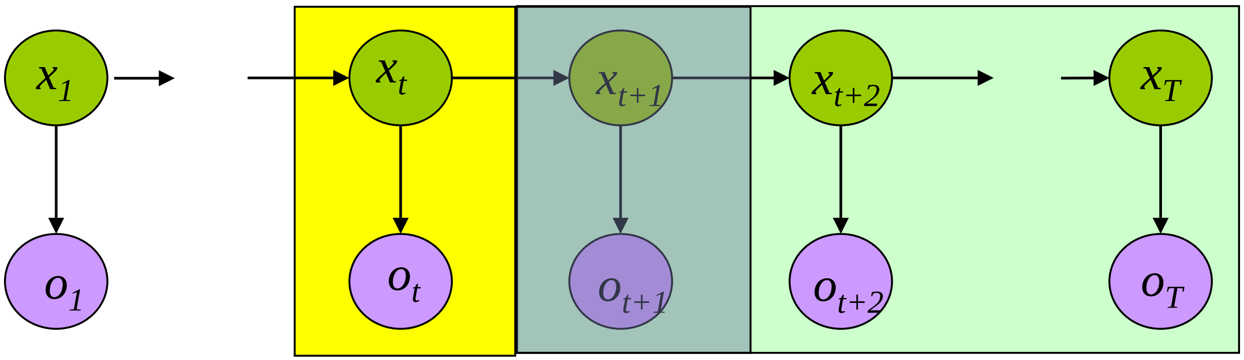
$$\alpha_{t+1}(j) = P(o_1 \ldots o_{t+1}, x_{t+1} = j) = \sum_{i=1 \ldots N} P(o_1 \ldots o_{t+1}, x_t = i, x_{t+1} = j)$$

$$= \sum_{i=1,\ldots,N} \underline{P(o_1, .. o_t, x_t = i)} \underline{P(x_{t+1} = j \mid x_t = i)} \underline{P(o_{t+1} \mid x_{t+1} = j)}$$

$$= \sum_{i=1,\ldots,N} \alpha_t(i) a_{i,j} b_j(o_{t+1})$$

$$\Pr(A) = \sum_{i=1}^{n} \Pr(A \mid B_i) \Pr(B_i)$$

Evaluation of HMMs: Backward Process



$$\beta_t(i) = P(o_{t+1} \ldots o_T \mid x_t = i)$$

$$\beta_t(i) \leftarrow \beta_{t+1}(j)$$

$$(i, j = 1, \ldots, N)$$

$$\beta_T(i) = 1$$

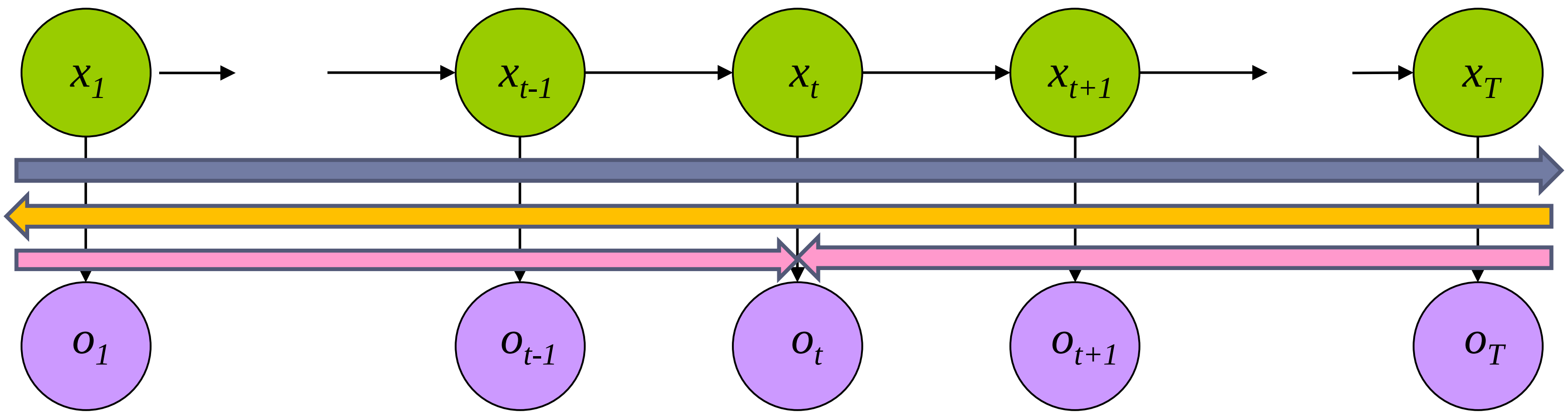
$$\beta_t(i) = P(o_{t+1} \ldots o_T \mid x_t = i) = \sum_{j=1,\ldots,N} P(x_{t+1} = j, o_{t+1} \ldots o_T \mid x_t = i)$$

$$= \sum_{j=1 \ldots N} \underline{P(o_{t+2} \ldots o_T \mid x_{t+1} = j)}\,\underline{P(x_{t+1} = j \mid x_t = i)}\,\underline{P(o_{t+1} \mid x_{t+1} = j)}$$

$$= \sum_{j=1 \ldots N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

$$\Pr(A) = \sum_{i=1}^{n} \Pr(A \mid B_i) \Pr(B_i)$$

Evaluation of HMMs: Forward and Backward Process



$\alpha_1(i) = \pi_i b_i(o_1)$

$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)$$ **Forward Procedure**

$\alpha_t(i) \to \alpha_{t+1}(j)$
$(i, j = 1, \ldots, N)$

$\beta_t(i) \leftarrow \beta_{t+1}(j)$
$(i, j = 1, \ldots, N)$

$$P(O \mid \mu) = \sum_{i=1}^{N} \pi_i b_{i,o_1} \beta_1(i)$$ **Backward Procedure**

$\beta_T(i) = 1$

$\alpha_t(i) \to \alpha_{t+1}(j)$
$(i, j = 1, \ldots, N)$

$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i)$$ **Combination (at *t*)**

$\beta_t(i) \leftarrow \beta_{t+1}(j)$
$(i, j = 1, \ldots, N)$

Why do we need both forward and backward processes for HMM evaluation?

Teacher HMM: Forward Algorithm

Forward Algorithm / Teacher

- Initialization: $\alpha_i(1) = \pi_i b_i(A)$

	A	C	B	A	C
good	0.23				
neutral	0.1				
bad	0.0				

$$\pi_g = \pi_n = \pi_b = 1/3$$

$$A = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0 & 0.2 & 0.8 \end{bmatrix}$$

$$b_1(A) = 0.7$$
$$b_1(B) = 0.2$$
$$b_1(C) = 0.1$$

$$b_2(A) = 0.3$$
$$b_2(B) = 0.4$$
$$b_2(C) = 0.3$$

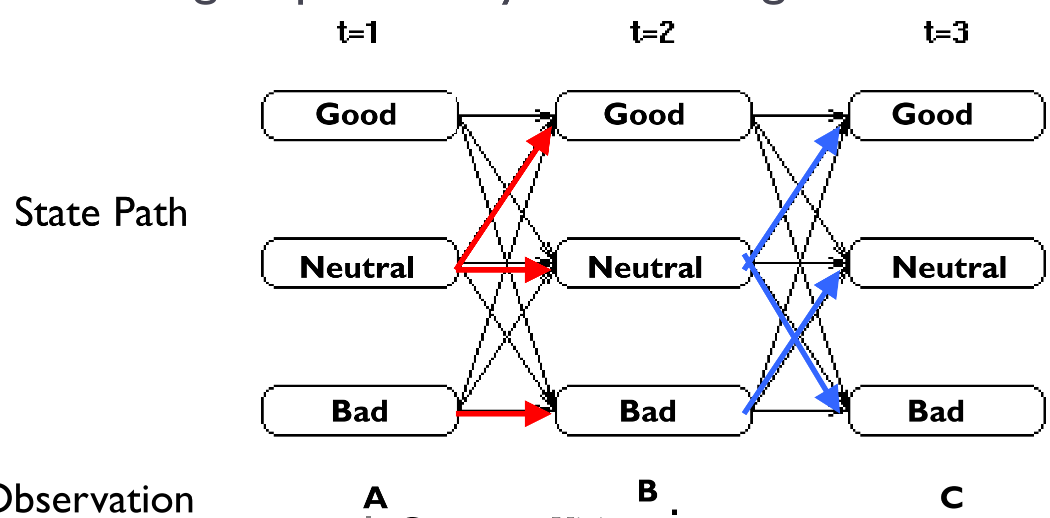
$$b_3(A) = 0.0$$
$$b_3(B) = 0.1$$
$$b_3(C) = 0.9$$

Decoding of HMMs: Viterbi Algorithm

- The goal is to find the state sequence that best explains the observations.

$$\arg\max_{X} P(X \mid O, \mu)$$

- How to find the optimal sequence efficiently?
  - The key idea is local competition/early elimination. In other words, for each state in time step *t*, find the best local path from step *t*-1 to step *t* that has the largest probability considering the observations at *t*-1 and *t*.



Viterbi Algorithm (1)

- Initialization:

$$\delta_1(j) = \pi_j b_j(o_1) \quad 1 \le j \le N$$

The probability of the first state is *j*

$$\psi_0(j) = 0$$

State Path

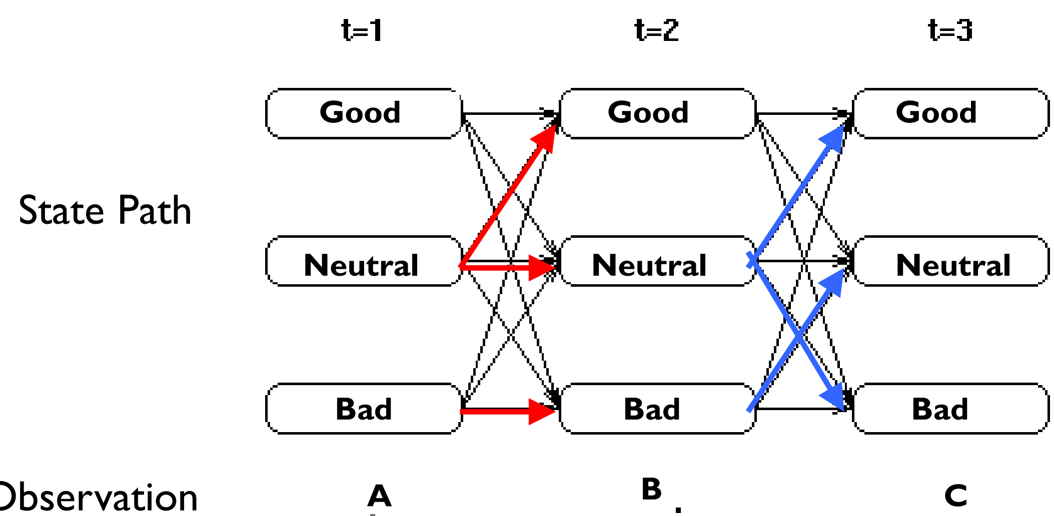
- Recursion:

$$\delta_t(j) = \max_i \left(\delta_{t-1}(i) a_{i,j} b_j(o_t)\right)$$

The probability of the best path from time t-1 to time t

$$\psi_t(j) = \arg\max_i \left(\delta_{t-1}(i) a_{i,j}\right)$$

The most possible state in time t-1 in order to get state j in time t



Viterbi Algorithm (2)

- Termination: $$p^* = \max_i(\delta_T(i))$$

The probability of the best path from time 1 to time T.

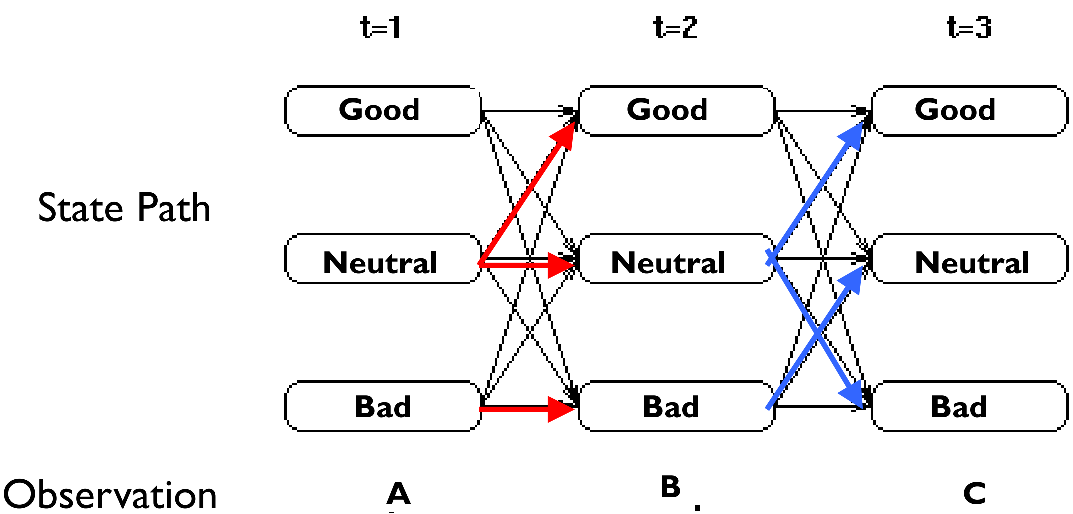
- Path backtracking

$$q_T^* = \arg\max_i(\delta_T(i))$$

The most possible state in time T (last time step).

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

The most possible state for previous time t=T-1,T-2,…,1.



Teacher HMM: Viterbi Algorithm

Viterbi Algorithm / Teacher

$$\delta_1(j) = \pi_j b_j(A) \qquad 1 \le j \le N$$

	A	C	B	A	C
good	0.23				
neutral	0.1				
bad	0.0				

$$\pi_g = \pi_n = \pi_b = 1/3$$

$$\mathbf{A} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0 & 0.2 & 0.8 \end{bmatrix}$$

$b_1(A) = 0.7$
$b_1(B) = 0.2$
$b_1(C) = 0.1$

$b_2(A) = 0.3$
$b_2(B) = 0.4$
$b_2(C) = 0.3$

$b_3(A) = 0.0$
$b_3(B) = 0.1$
$b_3(C) = 0.9$