Machine Learning
CS690

Lecture 08

Razvan C. Bunescu

School of Electrical Engineering and Computer Science

bunescu@ohio.edu

Structured Data

- For many applications, the i.i.d. assumption is does not hold:
  - pixels in images of real objects.
  - hyperlinked web pages.
  - cross-citations in scientific papers.
  - entities in social networks.
  - sequences of words/letters in text.
  - successive time frames in speech.
  - sequences of base pair in DNA.
  - musical notes in a tonal melody.
  - daily values of a particular stock.

Structured Data

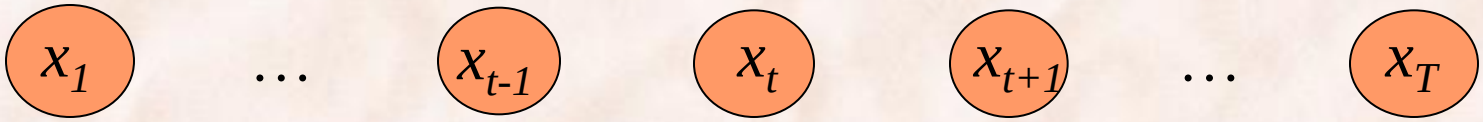
- For many applications, the i.i.d. assumption is does not hold:
  - pixels in images of real objects.
  - hyperlinked web pages.
  - cross-citations in scientific papers.
  - entities in social networks.
  - *sequences of words/letters in text.*
  - *successive time frames in speech.*
  - *sequences of base pair in DNA.*
  - *musical notes in a tonal melody.*
  - *daily values of a particular stock.*

Sequential Data

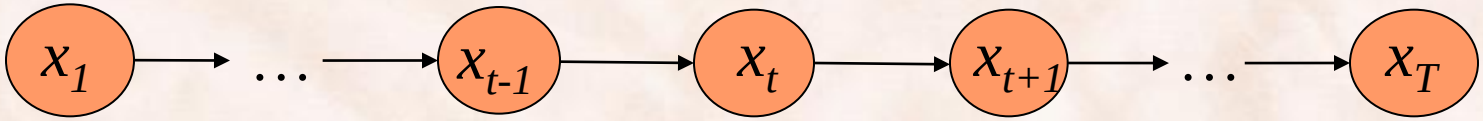
Sequential Data

Q: How can we model sequential data?

1) Ignore sequential aspects and treat the observations as i.i.d.



2) Relax the i.i.d. assumption by using a Markov model.



Markov Models

- $X = x_1, \ldots, x_T$ is a sequence of random variables.
- $S = \{s_1, \ldots, s_N\}$ is a state space, i.e. x_t takes values from S.

1) **Limited Horizon**:

$$P(x_{t+1} = s_k \mid x_1, \ldots, x_t) = P(x_{t+1} = s_k \mid x_t)$$

2) **Stationarity**:

$$P(x_{t+1} = s_k \mid x_t) = P(x_2 = s_k \mid x_1)$$

$\Rightarrow X$ is said to be a *Markov chain*.

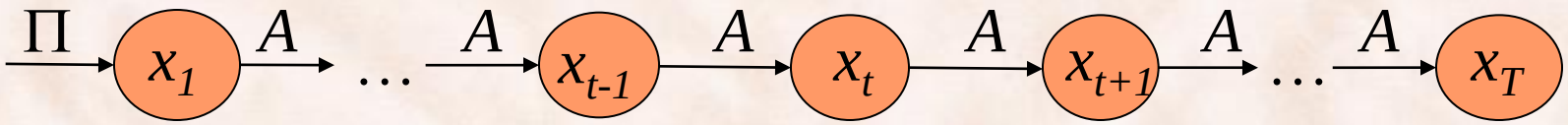
Markov Models: Parameters

- $S = \{s_1, \ldots, s_N\}$ are the *visible* states.
- $\Pi = \{\pi_i\}$ are the initial state probabilities.

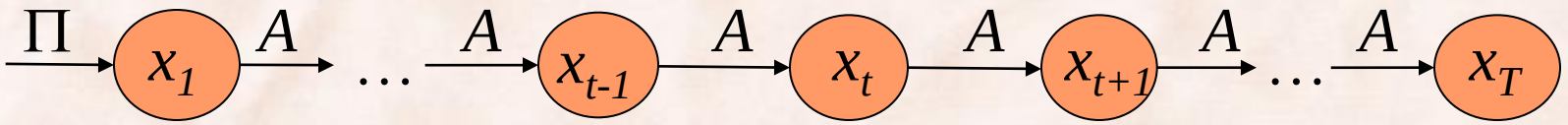
$$\pi_i = P(x_1 = s_i)$$

- $A = \{a_{ij}\}$ are the state transition probabilities.

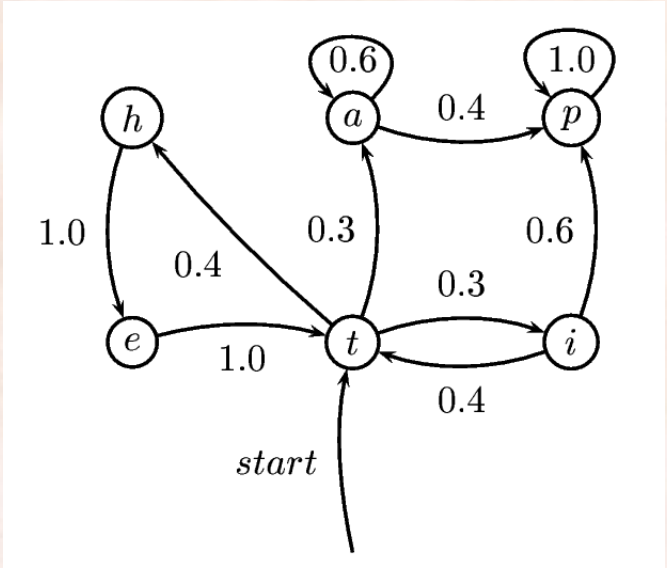
$$a_{ij} = P(x_{t+1} = s_j \mid x_t = s_i)$$



Markov Models: Inference



$$p(X) = p(x_1,\ldots,x_T)$$
$$= p(x_1)\prod_{t=1}^{T-1} P(x_{t+1} \mid x_t)$$
$$= \pi_{x_1}\prod_{t=1}^{T-1} a_{x_t x_{t+1}}$$



- Exercise: compute $p(t,a,p)$

m^{th} Order Markov Models

- First order Markov model:

$$p(X) = p(x_1)\prod_{t=1}^{T-1} P(x_{t+1} \mid x_t)$$

- Second order Markov model:

$$p(X) = p(x_1)p(x_2 \mid x_1)\prod_{t=2}^{T-1} P(x_{t+1} \mid x_t, x_{t-1})$$

- m^{th} order Markov model:

$$p(X) = p(x_1)p(x_2 \mid x_1)\ldots p(x_m \mid x_{m-1},\ldots, x_1)\prod_{t=m}^{T-1} P(x_{t+1} \mid x_t,\ldots, x_{t-m+1})$$

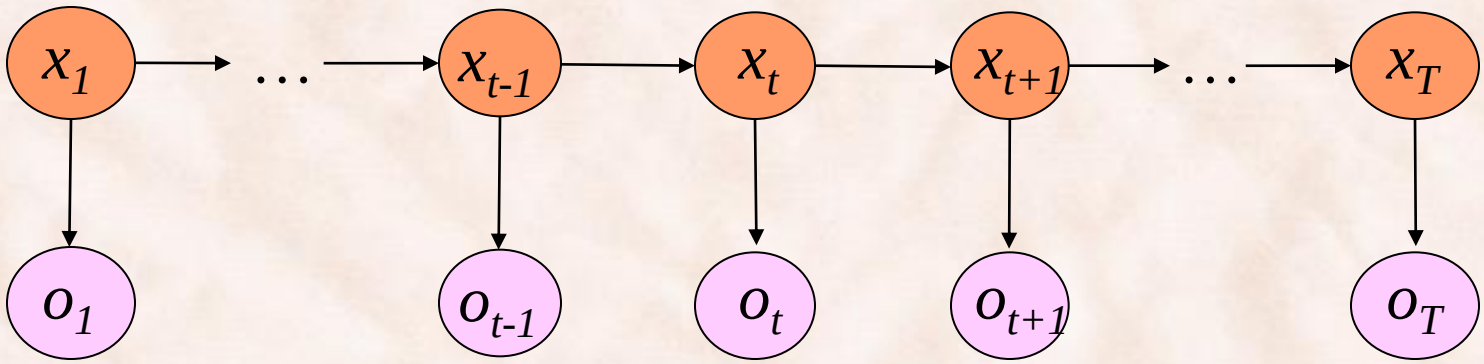
Markov Models

- **(Visible) Markov Models**:
  - Developed by Andrei A. Markov [Markov, 1913]
    - modeling the letter sequences in Pushkin's "Eugene Onyegin".

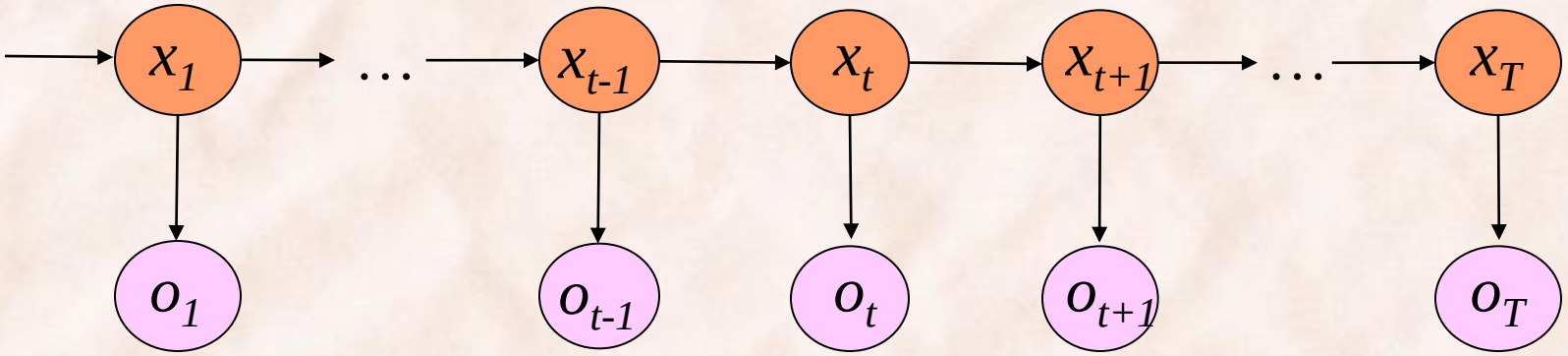
- **Hidden Markov Models**:
  - The *states* are hidden (latent) variables.
  - The states probabilistically generate surface events, or *observations*.
  - Efficient training using Expectation Maximization (EM)
    - Maximum Likelihood (ML) when tagged data is available.
  - Efficient inference using the Viterbi algorithm.

Hidden Markov Models (HMMs)

- Probabilistic *directed graphical models*:
  - Hidden *states* (shown in **brown**).
  - Visible *observations* (shown in **lavender**).
  - Arrows model probabilistic (in)dependencies.



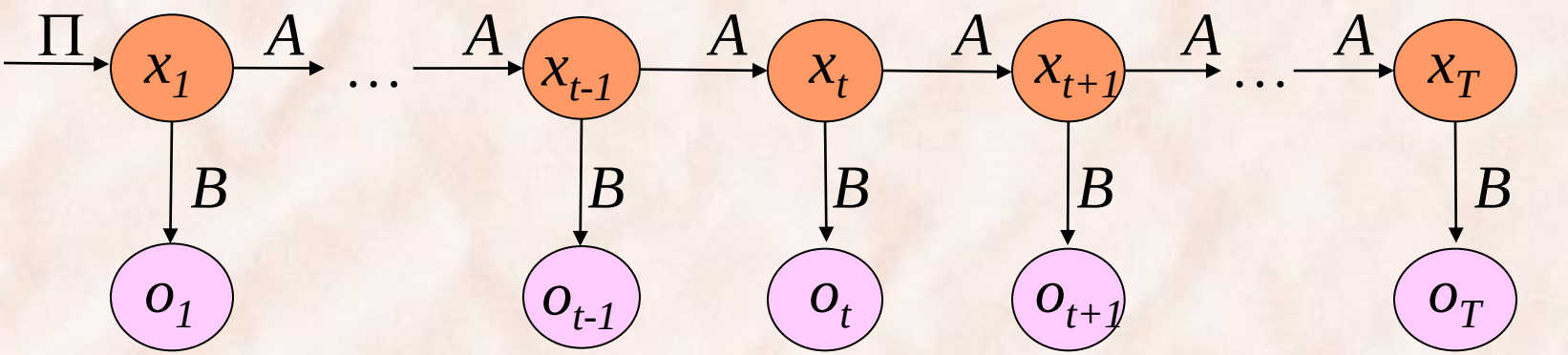
HMMs: Parameters



- $S = \{s_1, \ldots, s_N\}$ is the set of states.
- $\mathrm{K} = \{k_1, \ldots, k_M\} = \{1, \ldots, \mathrm{M}\}$ is the observations alphabet.

- $X = x_1, \ldots, x_T$ is a sequence of states.
- $O = o_1, \ldots, o_T$ is a sequence of observations.

HMMs: Parameters



- $\Pi = \{\pi_i\}$, $i \in S$, are the initial state probabilities.
- $A = \{a_{ij}\}$ }, $i,j \in S$, are the state transition probabilities.
- $B = \{b_{ik}\}$, $i \in S$, $k \in K$, are the symbol emision probabilities.

$$b_{ik} = P(o_t = k \mid x_t = s_i)$$

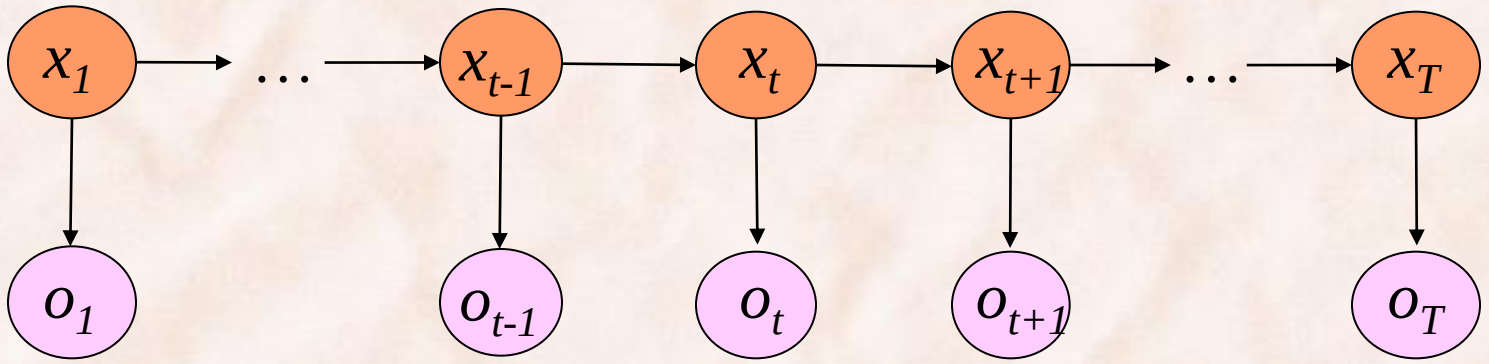
HMMs: Inference and Training

- Three fundamental questions:
  1) Given a model $\mu = (A, B, \Pi)$, compute the probability of a given observation sequence i.e. $p(O|\mu)$ (*Forward-Backward*).
  2) Given a model μ and an observation sequence O, compute the most likely hidden state sequence (*Viterbi*).

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

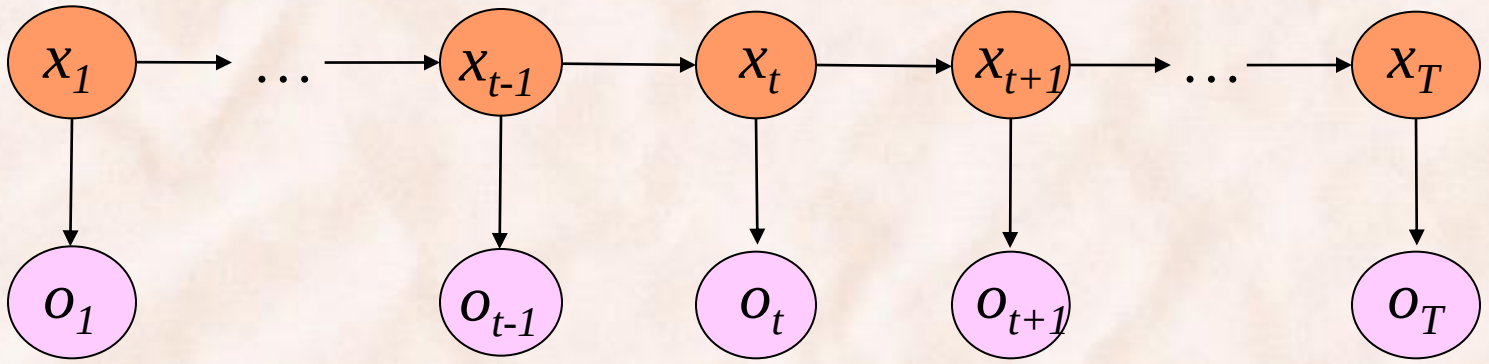
  3) Given an observation sequence O, find the model $\mu = (A, B, \Pi)$ that best explains the observed data (*EM*).
     - Given observation and state sequence O, X find μ (*ML*).

HMMs: Decoding



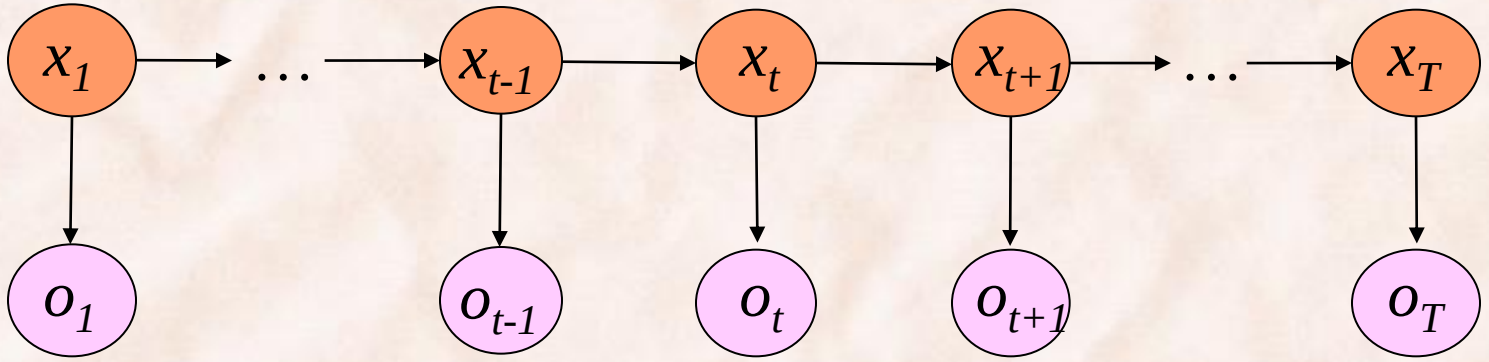
1) Given a model $\mu = (A, B, \Pi)$, compute the probability of a given observation sequence $O = o_1, \ldots, o_T$ i.e. $p(O|\mu)$

HMMs: Decoding



$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

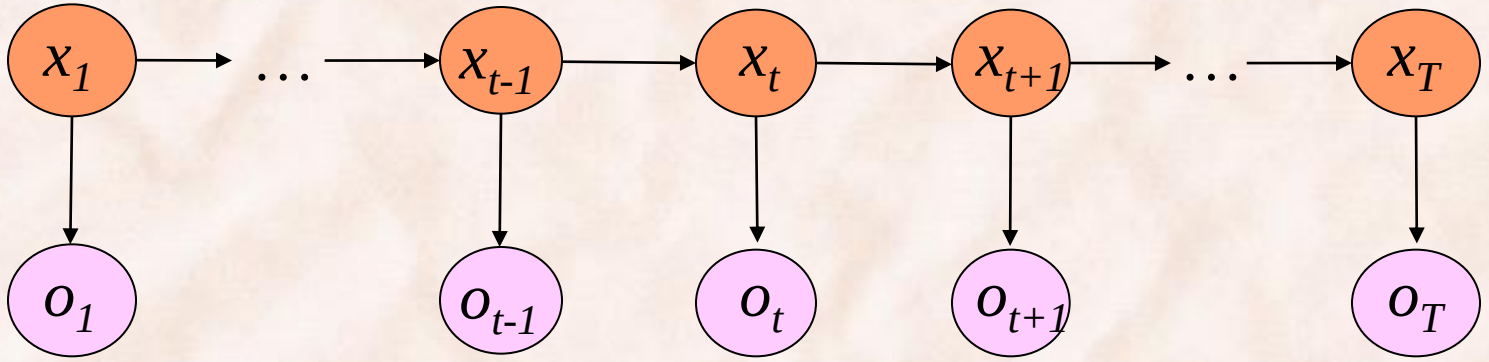
HMMs: Decoding



$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

$$P(X \mid \mu) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} \ldots a_{x_{T-1} x_T}$$

HMMs: Decoding

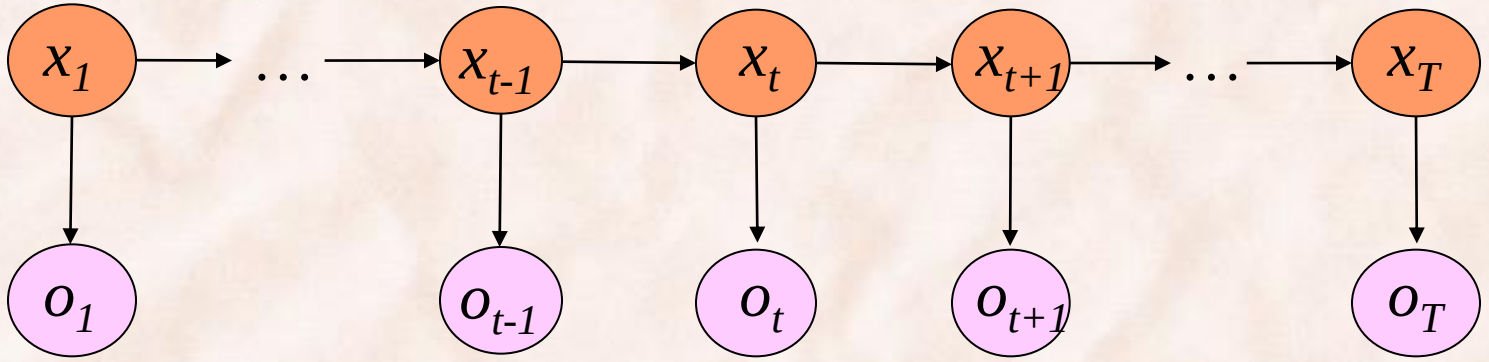


$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

$$P(X \mid \mu) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} \ldots a_{x_{T-1} x_T}$$

$$P(O, X \mid \mu) = P(O \mid X, \mu) P(X \mid \mu)$$

HMMs: Decoding



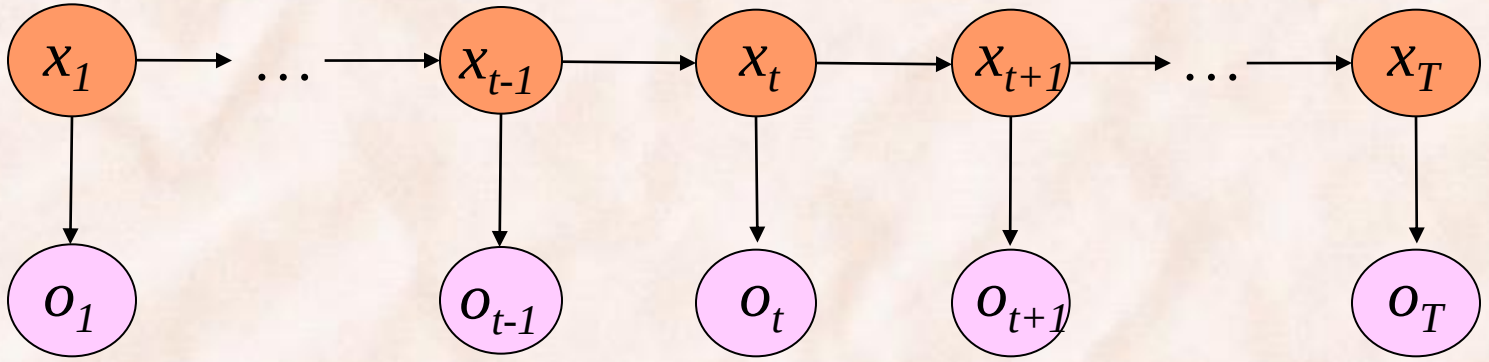
$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

$$P(X \mid \mu) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} \ldots a_{x_{T-1} x_T}$$

$$P(O, X \mid \mu) = P(O \mid X, \mu) P(X \mid \mu)$$

$$P(O \mid \mu) = \sum_X P(O \mid X, \mu) P(X \mid \mu)$$

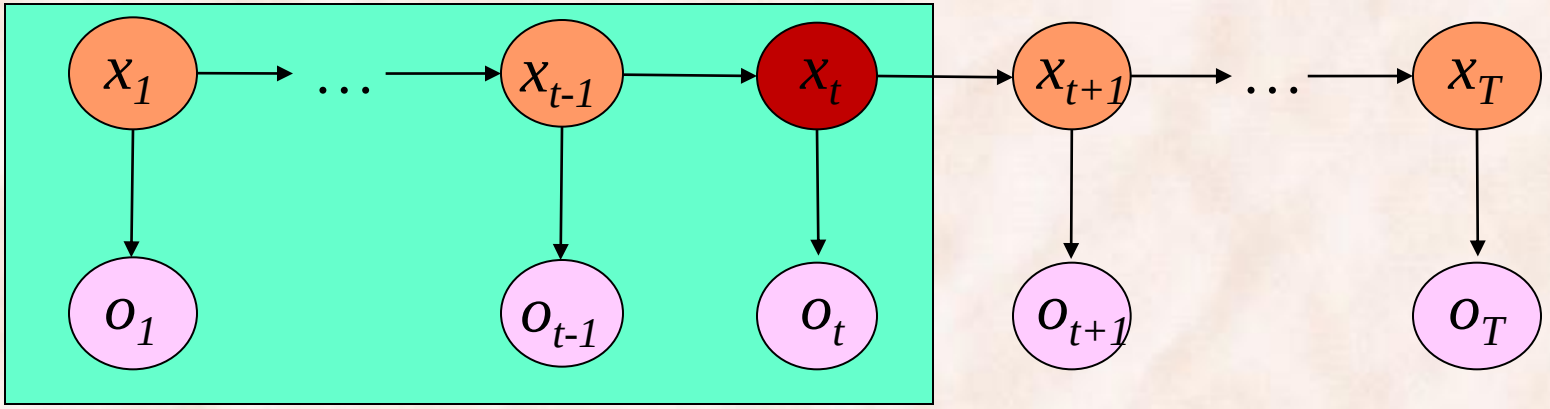
HMMs: Decoding



$$p(O \mid \mu) = \sum_{\{x_1 \ldots x_T\}} \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$

Time complexity?

HMMs: Forward Procedure



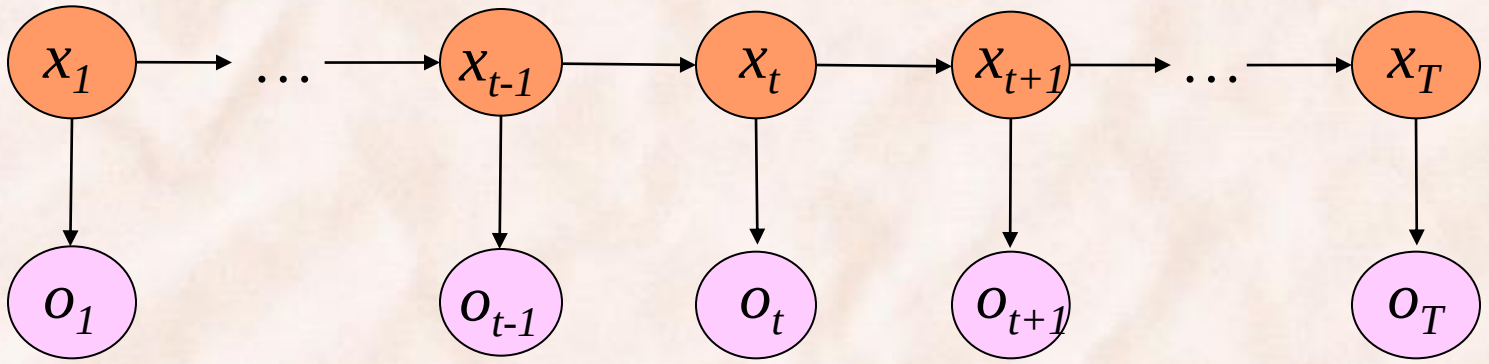
- Define:

$$\alpha_i(t) = P(o_1 \ldots o_t, x_t = i \mid \mu)$$

- Then solution is:

$$p(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(T)$$

HMMs: Decoding



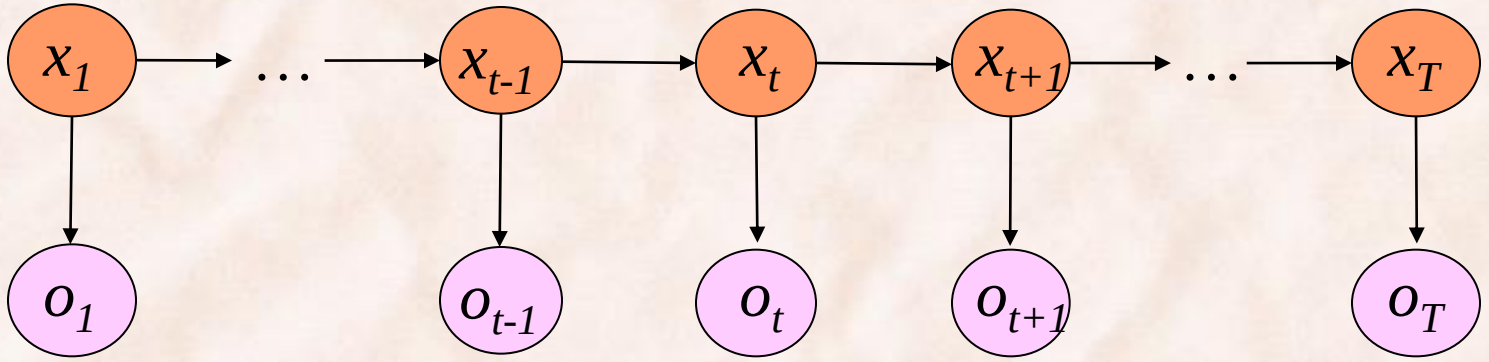
$$\alpha_j(t+1) = P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

$$= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

$$= P(o_1 \ldots o_t \mid x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

$$= P(o_1 \ldots o_t, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

HMMs: Decoding



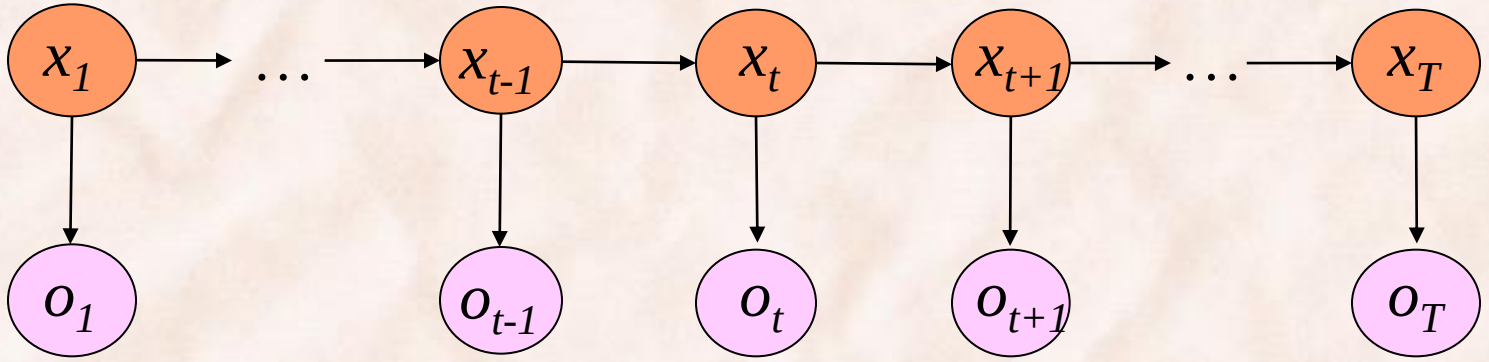
$$\alpha_j(t+1) = P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

$$= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

$$= P(o_1 \ldots o_t \mid x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

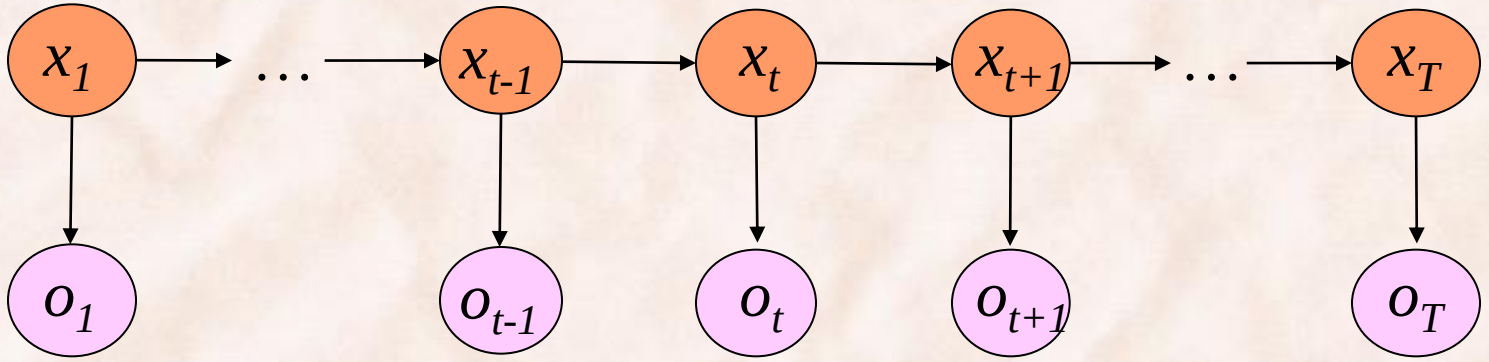
$$= P(o_1 \ldots o_t, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

HMMs: Decoding



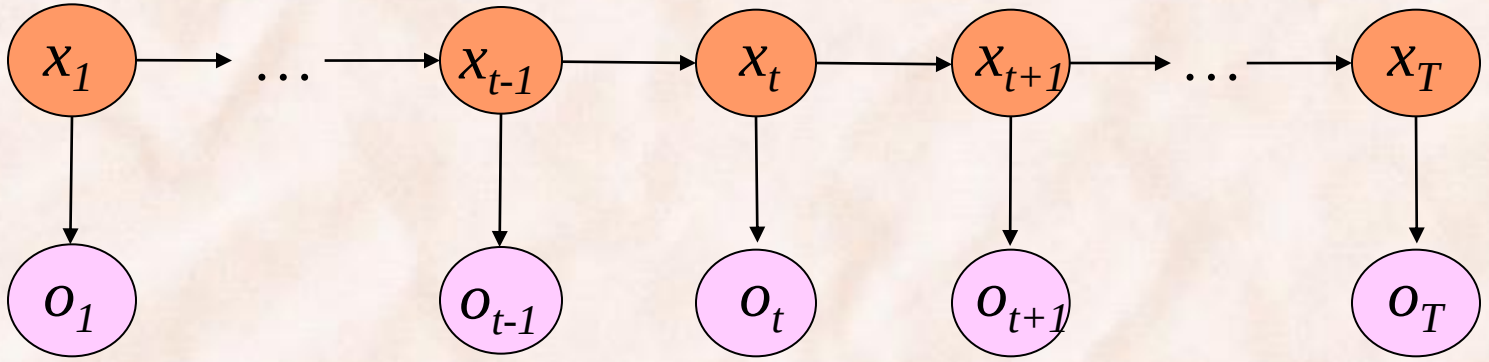
$$\begin{aligned}
\alpha_j(t+1) &= P(o_1 \ldots o_{t+1}, x_{t+1} = j) \\
&= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j) \\
&= P(o_1 \ldots o_t \mid x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j) \\
&= P(o_1 \ldots o_t, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)
\end{aligned}$$

HMMs: Decoding



$$\alpha_j(t+1) = P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$
$$= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$
$$= P(o_1 \ldots o_t \mid x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$
$$= P(o_1 \ldots o_t, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

HMMs: Decoding



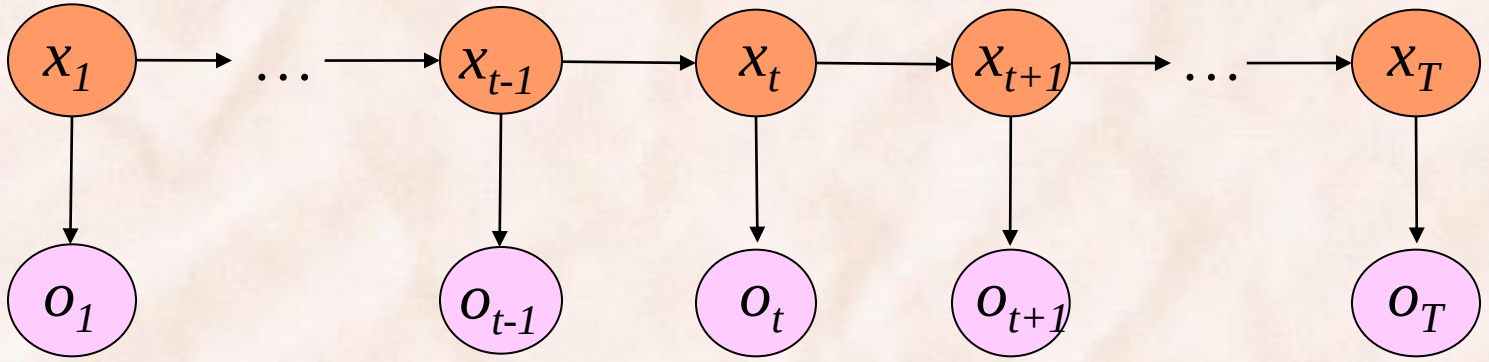
$$\alpha_j(t+1) = \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

$$= \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}$$

HMMs: Decoding



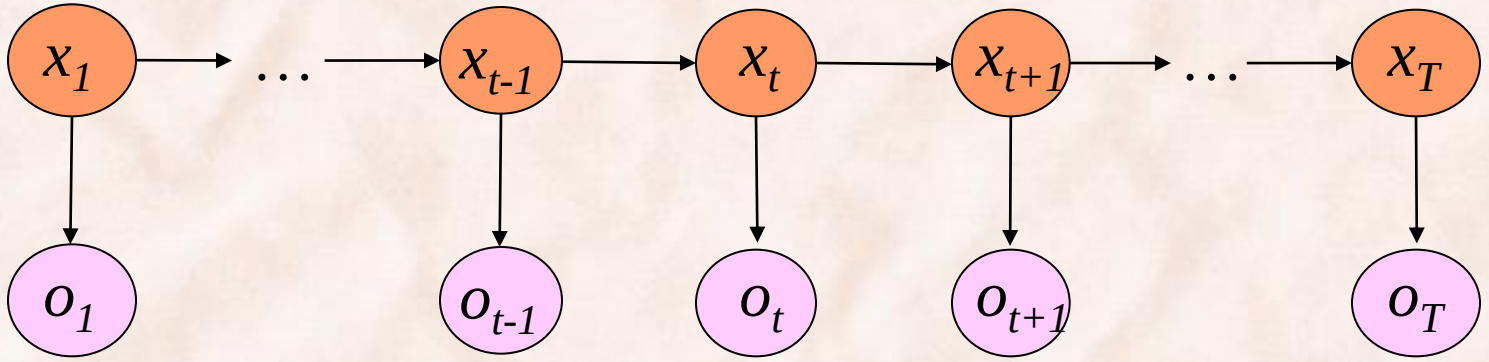
$$\alpha_j(t+1) = \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

$$= \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}$$

HMMs: Decoding



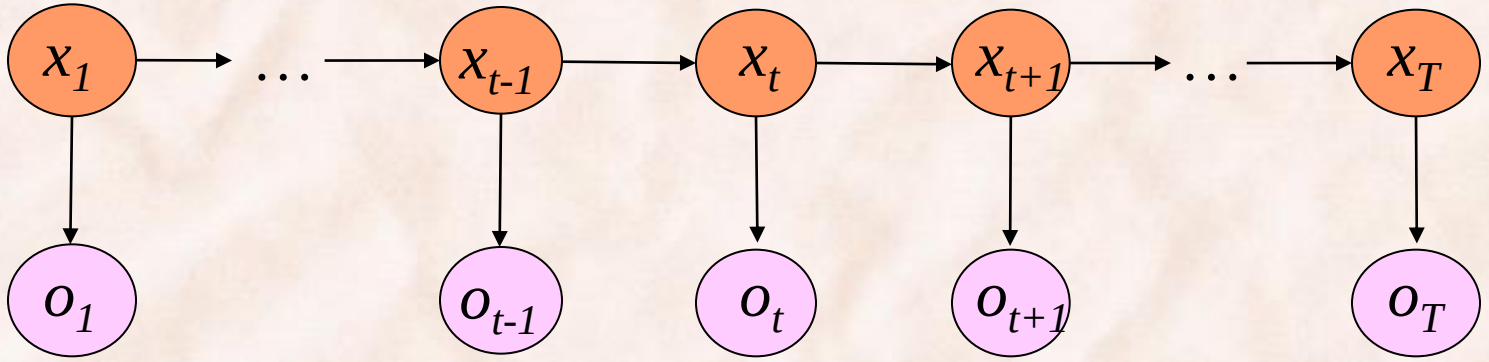
$$\alpha_j(t+1) = \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

$$= \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}$$

HMMs: Decoding



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

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

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

$$= \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}$$

The Forward Procedure

1. Initialization

$$\alpha_i(1) = \pi_i b_{io_1}, \qquad 1 \le \mathrm{i} \le \mathrm{N}$$

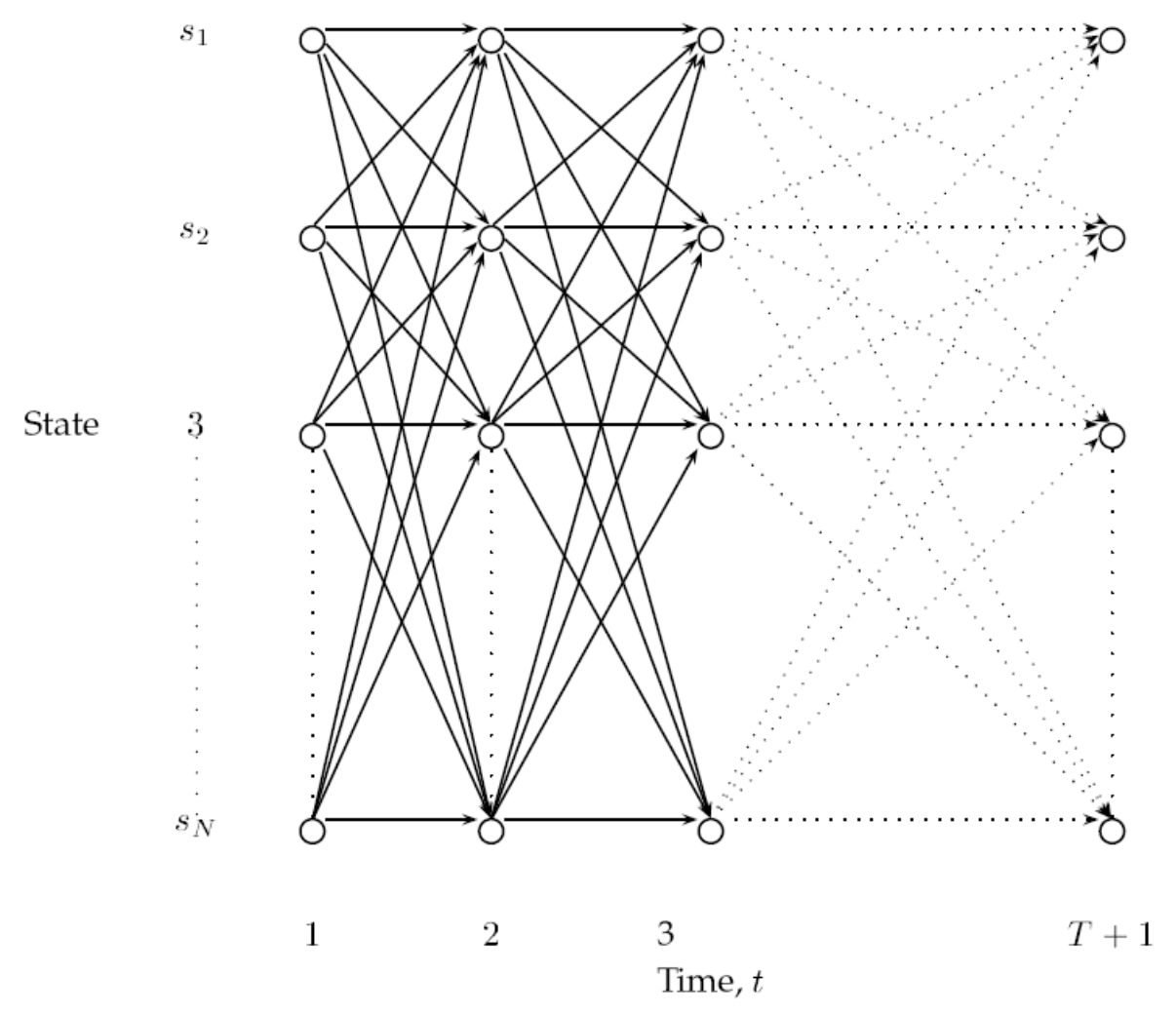
2. Recursion:

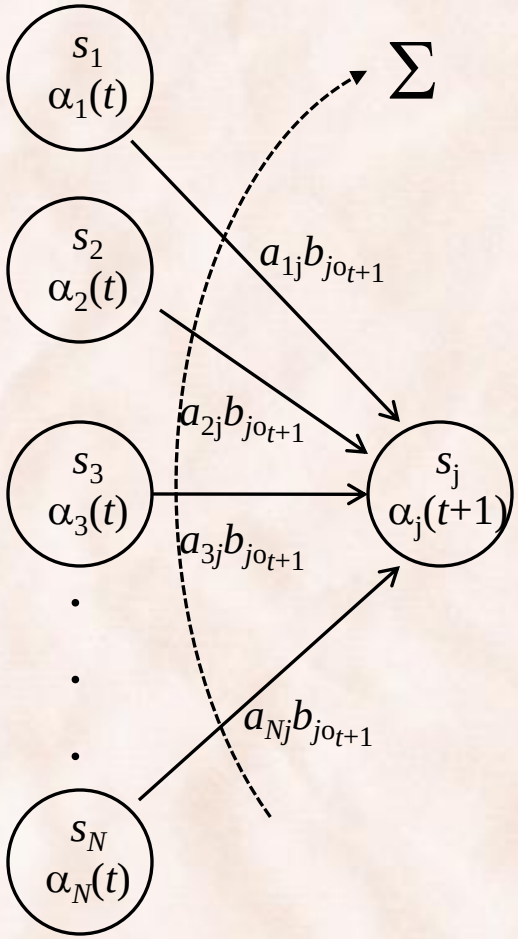
$$\alpha_j(t+1) = \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}, \qquad 1 \le \mathrm{j} \le \mathrm{N}, 1 \le \mathrm{t} < \mathrm{T}$$

3. Termination:

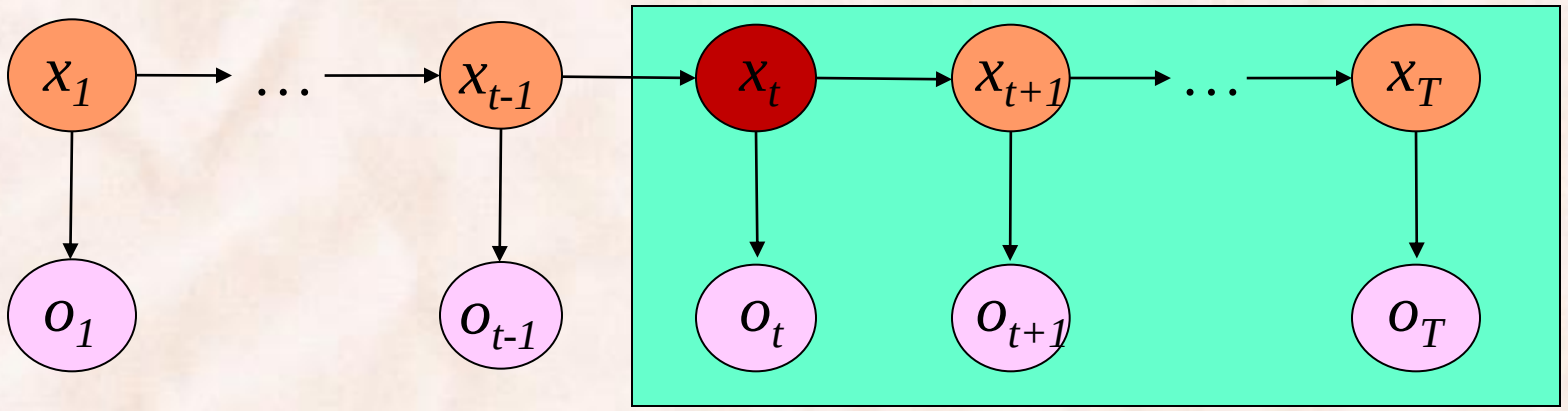
$$p(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(T)$$

The Forward Procedure: Trellis Computation





HMMs: Backward Procedure



- Define:

$$\beta_i(t) = P(o_{t+1}...o_T \mid x_t = i, \mu)$$

- Then solution is:

$$p(O \mid \mu) = \sum_{i=1}^{N} \pi_i b_{io_1} \beta_i(1)$$

The Backward Procedure

1. Initialization

$$\beta_i(T) = 1, \quad 1 \le \mathrm{i} \le \mathrm{N}$$

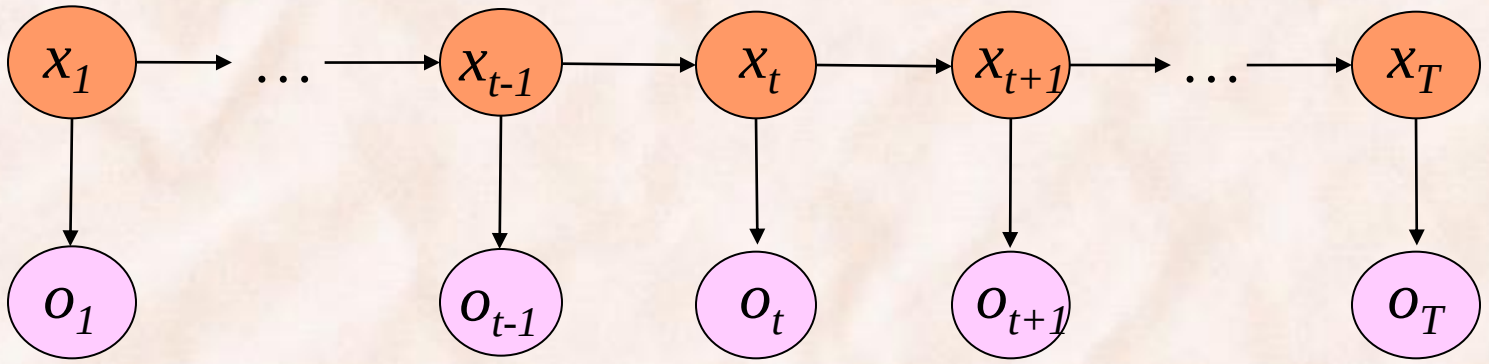
2. Recursion:

$$\beta_i(t) = \sum_{j=1\ldots N} a_{ij} b_{jo_{t+1}} \beta_j(t+1), \quad 1 \le \mathrm{i} \le \mathrm{N}, 1 \le \mathrm{t} < \mathrm{T}$$

3. Termination:

$$p(O \mid \mu) = \sum_{i=1}^{N} \pi_i b_{io_1} \beta_i(1)$$

HMMs: Decoding



- **Forward Procedure**: $p(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(T)$
- **Backward Procedure**: $p(O \mid \mu) = \sum_{i=1}^{N} \pi_i b_{io_1} \beta_i(1)$
- **Combination**: $p(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t)$

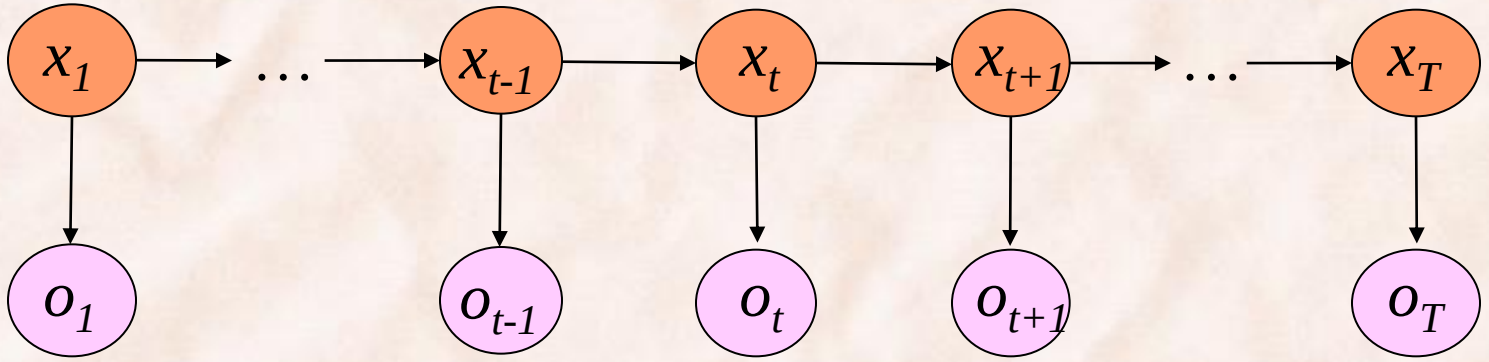
HMMs: Inference and Training

- Three fundamental questions:
  1) Given a model $\mu = (A, B, \Pi)$, compute the probability of a given observation sequence i.e. $p(O|\mu)$ (*Forward-Backward*).
  2) Given a model μ and an observation sequence O, compute the most likely hidden state sequence (*Viterbi*).

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

  3) Given an observation sequence O, find the model $\mu = (A, B, \Pi)$ that best explains the observed data (*EM*).
     - Given observation and state sequence O, X find μ (*ML*).

Best State Sequence with Viterbi Algorithm



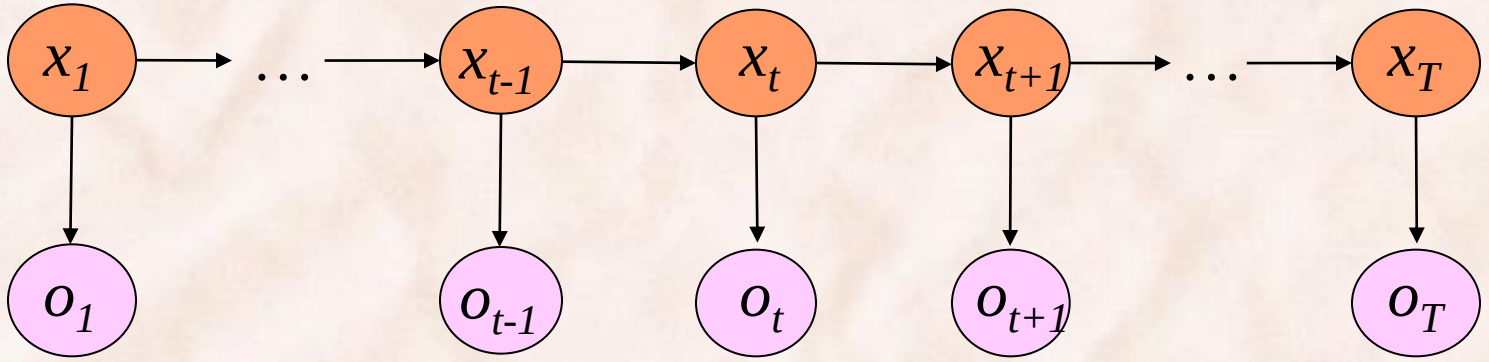
$$\hat{X} = \arg\max_X p(X \mid O, \mu)$$

$$= \arg\max_X p(X, O \mid \mu)$$

$$= \arg\max_{x_1,\ldots,x_T} p(x_1,\ldots,x_T, o_1,\ldots,o_T \mid \mu)$$

Time complexity?

The Viterbi Algorithm



$$\hat{X} = \arg\max_{x_1,\ldots,x_T} p(x_1,\ldots,x_T,o_1,\ldots,o_T \mid \mu)$$

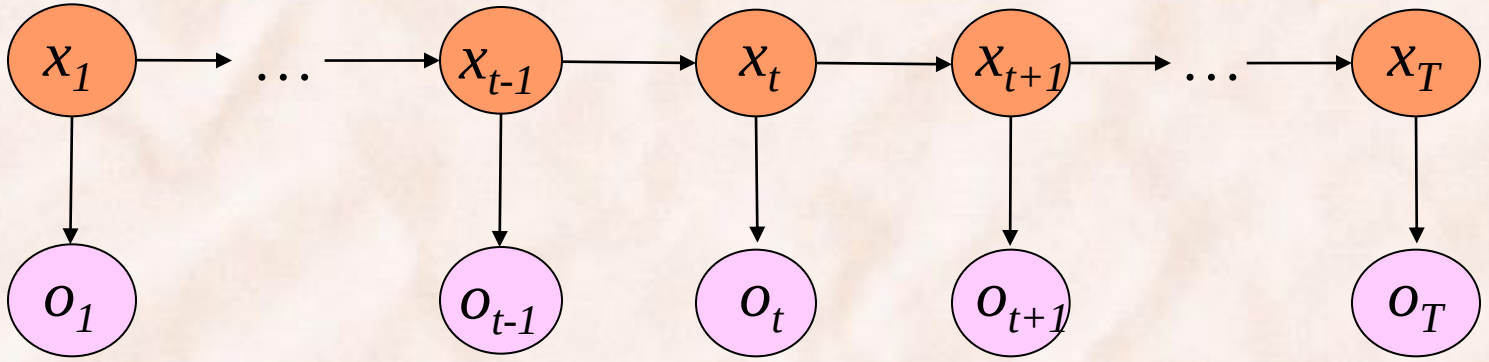
$$p(\hat{X}) = \max_{x_1,\ldots,x_T} p(x_1,\ldots,x_T,o_1,\ldots,o_T \mid \mu)$$

- The probability of the most probable path that leads to $x_t = j$:

$$\delta_j(t) = \max_{x_1\ldots x_{t-1}} p(x_1\ldots x_{t-1}, o_1\ldots o_{t-1}, x_t = j, o_t)$$

$$p(\hat{X}) = \max_{1\le j\le N} \delta_j(T)$$

The Viterbi Algorithm



- The probability of the most probable path that leads to $x_t = j$:

$$\delta_j(t) = \max_{x_1 \ldots x_{t-1}} p(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

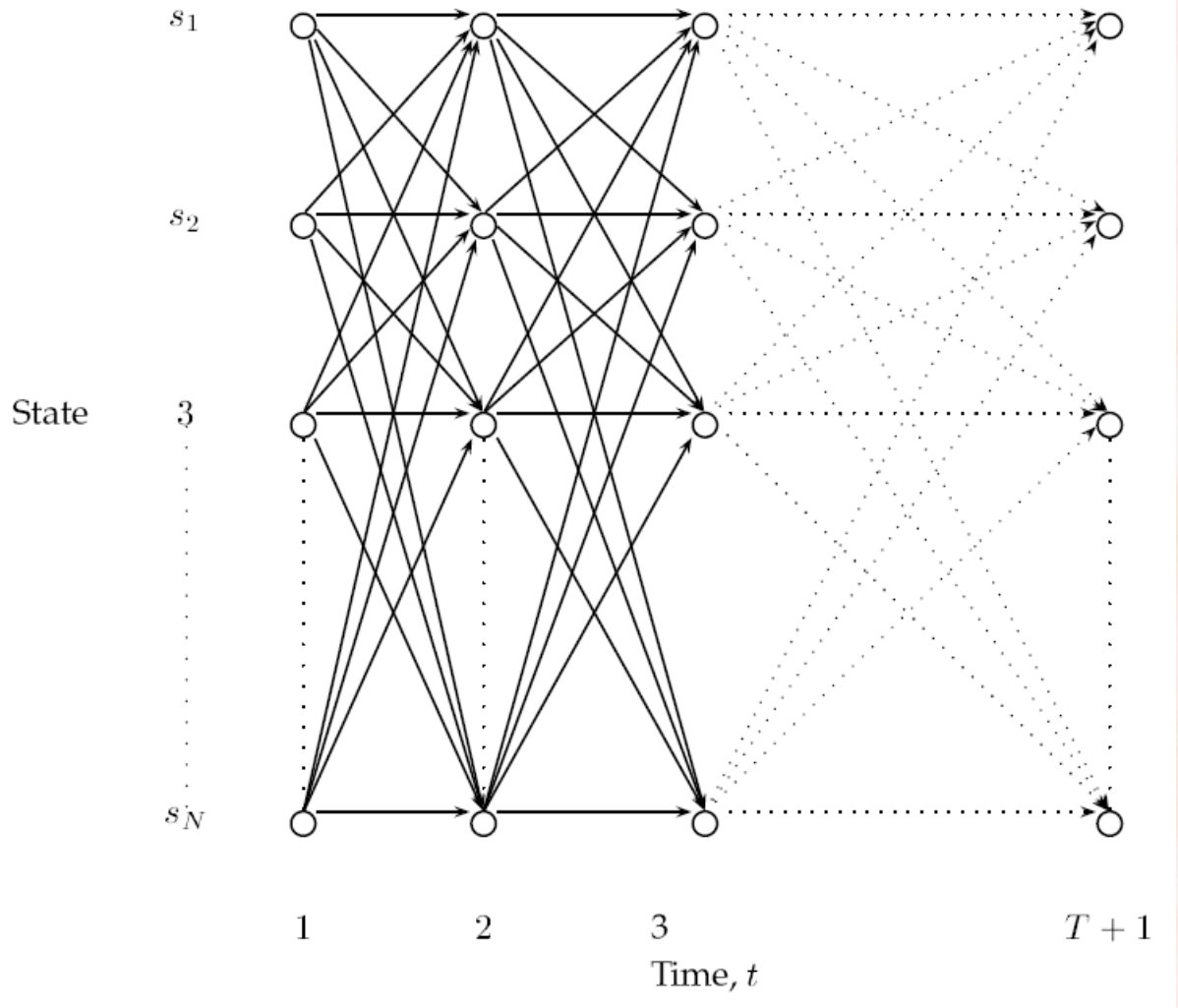
- It can be shown that:

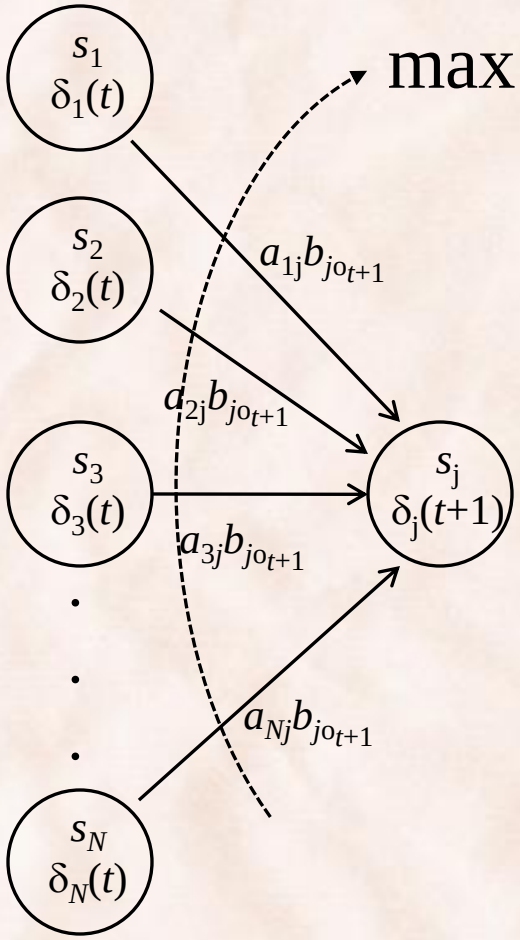
$$\delta_j(t+1) = \max_{1 \le i \le N} \delta_i(t) a_{ij} b_{jo_{t+1}}$$

Compare with:

$$\alpha_j(t+1) = \sum_{i=1 \ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}$$

The Viterbi Algorithm: Trellis Computation





The Viterbi Algorithm

1. Initialization

$$\delta_j(1) = \pi_j b_{jo_1}$$

$$\psi_j(1) = 0$$

2. Recursion

$$\delta_j(t+1) = \max_{1 \le i \le N} \delta_i(t) a_{ij} b_{jo_{t+1}}$$

$$\psi_j(t+1) = \arg\max_{1 \le i \le N} \delta_i(t) a_{ij} b_{jo_{t+1}}$$

3. Termination

$$p(\hat{X}) = \max_{1 \le j \le N} \delta_j(T)$$

$$\hat{x}_T = \arg\max_{1 \le j \le N} \delta_j(T)$$

4. State sequence backtracking

$$\hat{x}_t = \psi_{\hat{x}_{t+1}}(t+1)$$

Time complexity?

HMMs: Inference and Training

- Three fundamental questions:
  1) Given a model $\mu = (A, B, \Pi)$, compute the probability of a given observation sequence i.e. $p(O|\mu)$ (*Forward-Backward*).
  2) Given a model μ and an observation sequence O, compute the most likely hidden state sequence (*Viterbi*).
  3) Given an observation sequence O, find the model $\mu = (A, B, \Pi)$ that best explains the observed data (*EM*).
     - Given observation and state sequence O, X find μ (*ML*).

Parameter Estimation with Maximum Likelihood

- Given observation and state sequences O, X find $\mu = (A, B, \Pi)$.

$$\hat{\mu} = \arg\max_{\mu} p(O, X \mid \mu)$$

$$a_{ij} = p(x_{t+1} = s_j \mid x_t = s_i)$$

$$\hat{a}_{ij} = \frac{C(x_{t+1} = s_j, x_t = s_i)}{C(x_t = s_i)}$$

$$b_{ik} = p(o_t = k \mid x_t = s_i)$$

$$\hat{b}_{ik} = \frac{C(o_t = k, x_t = s_i)}{C(x_t = s_i)}$$

$$\pi_i = p(x_1 = s_i) \quad \hat{\pi}_i = \frac{C(x_1 = s_i)}{|X|}$$

Exercise:
Rewrite to use Laplace smoothing.

Parameter Estimations with Expectation Maximization

- Given observation sequences O find $\mu = (A, B, \Pi)$.

$$\hat{\mu} = \arg\max_{\mu} p(O \mid \mu)$$

- There is no known analytic method to find solution.

- Locally maximize $p(O|\mu)$ using iterative hill-climbing:

  $\Rightarrow$ the Baum-Welch or Forward-Backward algorithm:

  - Given a model μ and observation sequence, update the model parameters to $\hat{\mu}$ to better fit the observations.
  - A special case of the *Expectation Maximization* method.

The Baum-Welch Algorithm (EM)

[E] Assume μ is known, compute "hidden" parameters ξ, γ:

1) $\xi_t(i, j)$ = the probability of being in state s_i at time t and state s_j at time t+1.

$$\xi_t(i, j) = \frac{\alpha_i(t) a_{ij} b_{jo_{t+1}} \beta_j(t+1)}{\sum_{m=1\ldots N} \alpha_m(t) \beta_m(t)}$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from } s_i \text{ to } s_j$$

2) $\gamma_t(i)$ = the probability of being in state s_i at time t.

$$\gamma_i(t) = \sum_{j=1\ldots N} \xi_t(i, j)$$

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

The Baum-Welch Algorithm

[M] Re-estimate μ using expectations of ξ, γ:

$$\hat{\mu} \begin{cases} \hat{\pi}_i = \gamma_i(1) \\ \hat{a}_{ij} = \dfrac{\sum_{t=1}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_i(t)} \\ \hat{b}_{ik} = \dfrac{\sum_{\{t:o_t=k\}} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_i(t)} \end{cases}$$

- Baum has proven that $p(O \mid \hat{\mu}) \geq p(O \mid \mu)$

The Baum-Welch Algorithm

1. Start with some (random) model $\mu = (\mathrm{A},\mathrm{B},\Pi)$.
2. [E step] Compute $\xi_t(i, j)$, $\gamma_t(i)$ and their expectations.
3. [M step] Compute ML estimate $\hat{\mu}$.
4. Set $\mu = \hat{\mu}$ and repeat from 2. until convergence.

HMMs

- Three fundamental questions:
  1) Given a model $\mu = (A, B, \Pi)$, compute the probability of a given observation sequence i.e. $p(O|\mu)$ (*Forward/Backward*).
  2) Given a model μ and an observation sequence O, compute the most likely hidden state sequence (*Viterbi*).
  3) Given an observation sequence O, find the model $\mu = (A, B, \Pi)$ that best explains the observed data (*Baum-Welch*, or *EM*).
     - Given observation and state sequence O, X find μ (*ML*).