Advanced Artificial Intelligence

Part II. Statistical NLP

Hidden Markov Models

Wolfram Burgard, Luc De Raedt, Bernhard Nebel, Lars Schmidt-Thieme

Most slides taken (or adapted) from David Meir Blei, Figures from Manning and Schuetze and from Rabiner

Contents

- Markov Models
- Hidden Markov Models
  - Three problems - three algorithms
    - Decoding
    - Viterbi
    - Baum-Welsch
- Next chapter
  - Application to part-of-speech-tagging (POS-tagging)

Largely chapter 9 of Statistical NLP, Manning and Schuetze, or Rabiner, A tutorial on HMMs and selected applications in Speech Recognition, Proc. IEEE

Motivations and Applications

- Part-of-speech tagging / Sequence tagging
  - The representative put chairs on the table
  - AT NN VBD NNS IN AT NN
  - AT JJ NN VBZ IN AT NN
- Some tags :
  - AT: article, NN: singular or mass noun, VBD: verb, past tense, NNS: plural noun, IN: preposition, JJ: adjective

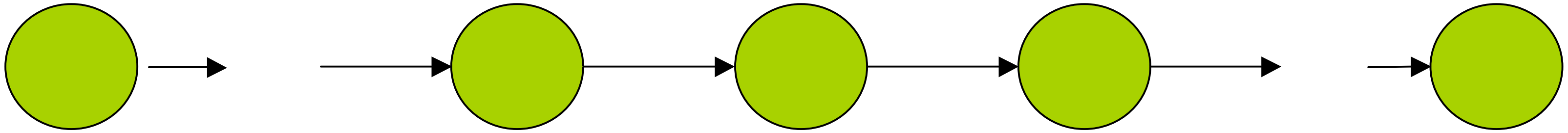
Bioinformatics

- Durbin et al. Biological Sequence Analysis, Cambridge University Press.
- Several applications, e.g. proteins
- From primary structure ATCPLELLLD
- Infer secondary structure HHHBBBBBC..

Other Applications

- Speech Recognition: from
  - From: Acoustic signals infer
  - Infer: Sentence
- Robotics:
  - From Sensory readings
  - Infer Trajectory / location …

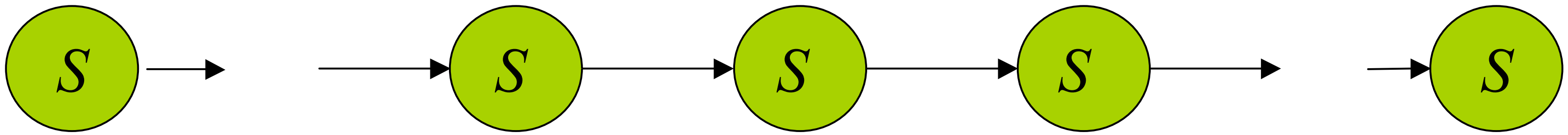
What is a (Visible) Markov Model ?



- Graphical Model (Can be interpreted as Bayesian Net)
- Circles indicate states
- Arrows indicate probabilistic dependencies between states
- State depends only on the previous state
- “The past is independent of the future given the present.”

- Recall from introduction to N-gramms !!!

Markov Model Formalization



- $\{S, \Pi, A\}$
- $S : \{s_1 \ldots s_N\}$ are the values for the hidden states

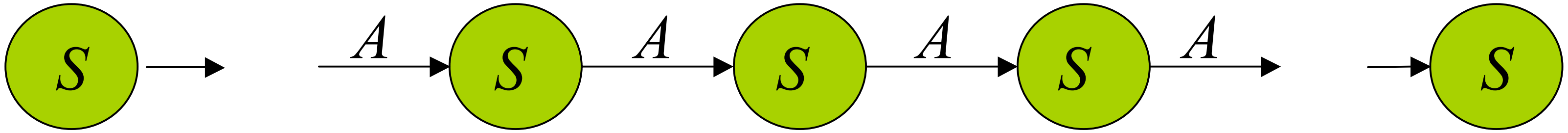
Limited Horizon (Markov Assumption)

$$P(X_{t+1} = s_k \mid X_1, \ldots, X_t) = P(X_{t+1} = s_k \mid X_t)$$

Time Invariant (Stationary) $= P(X_2 = s_k \mid X_1)$

Transition Matrix A $a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i)$

Markov Model Formalization



- $\{S, \Pi, A\}$
- S : $\{s_1 \ldots s_N\}$ are the values for the hidden states

- $\Pi = \{\pi_\iota\}$ are the initial state probabilities

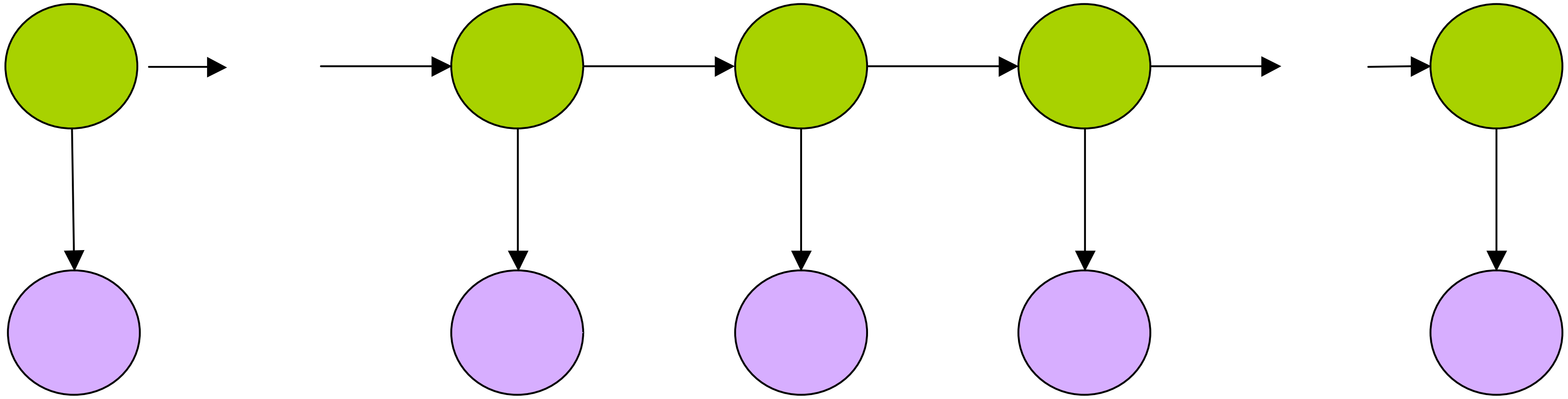
$$\pi_i = P(X_1 = s_i)$$

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

What is the probability of a sequence of states ?

$$P(X_1,\ldots,X_T)$$
$$= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1, X_2)\ldots P(X_T \mid X_1 \ldots, X_{T-1})$$
$$= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2)\ldots P(X_T \mid X_{T-1})$$
$$= \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}}$$

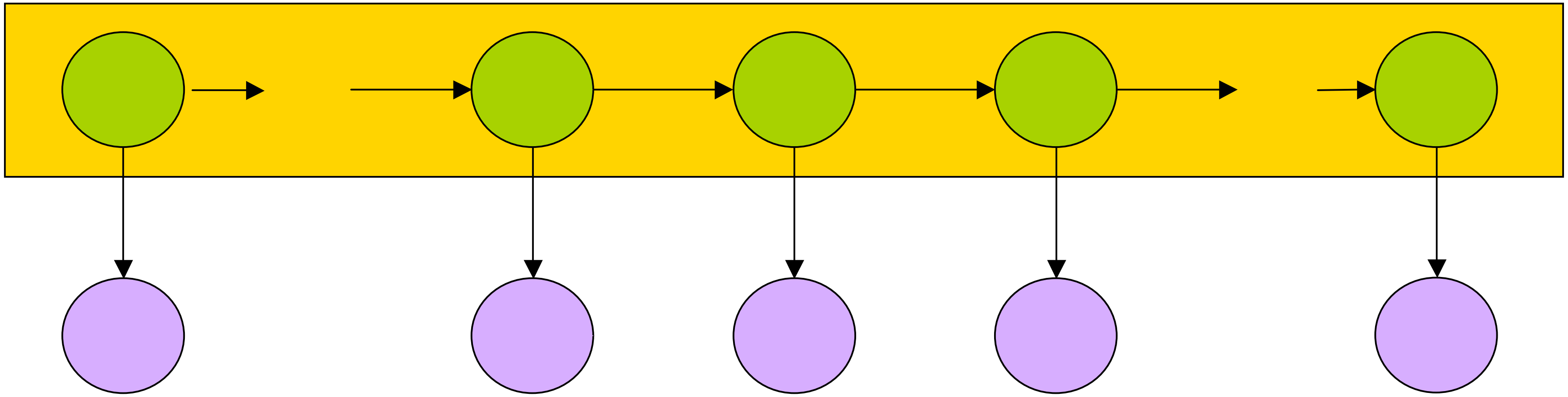
What is an HMM?



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

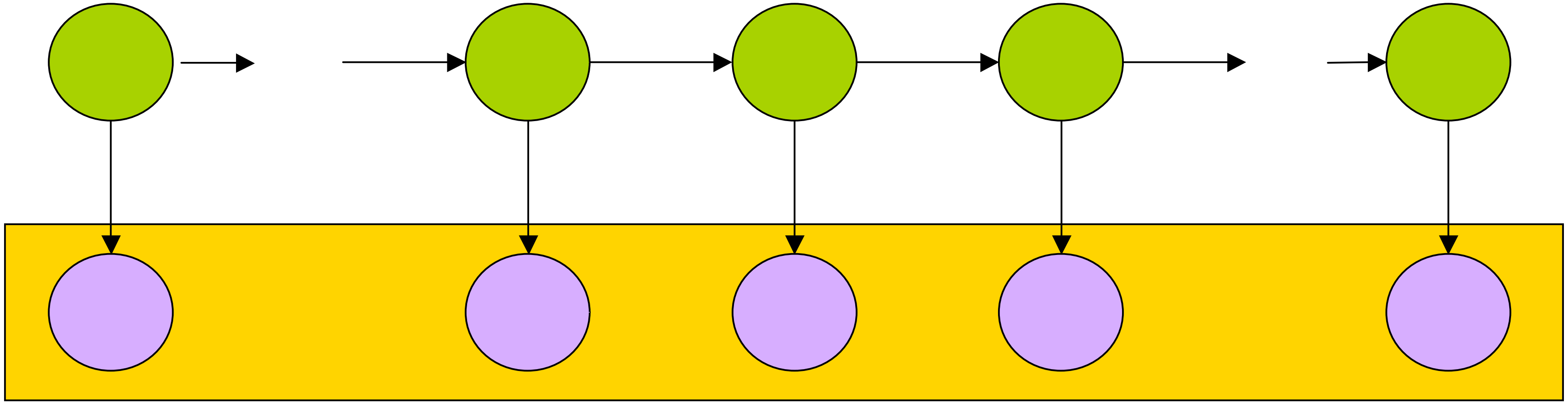
HMM = Hidden Markov Model

What is an HMM?



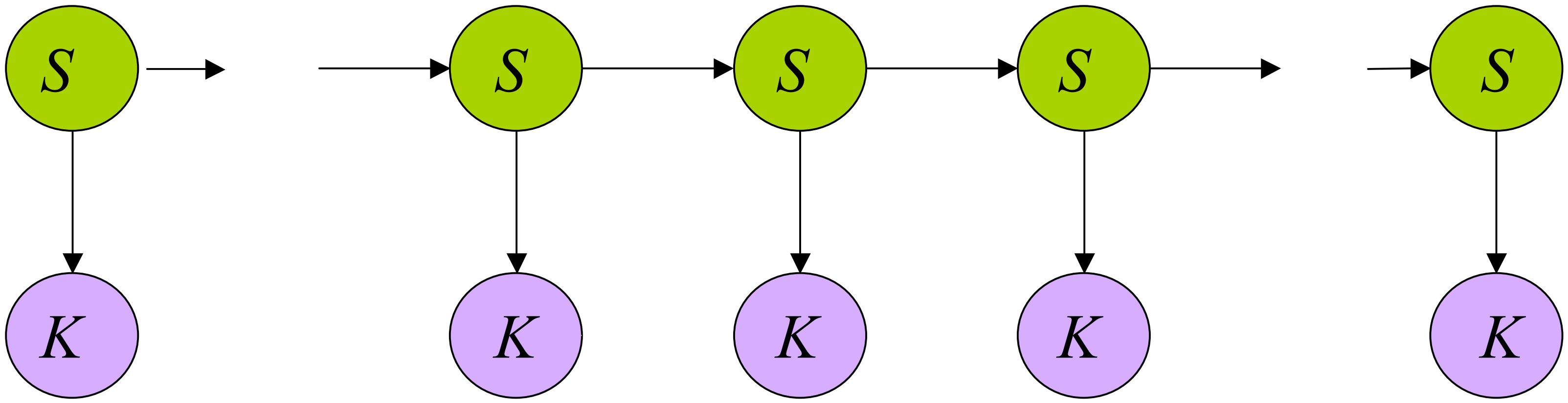
- Green circles are ***hidden states***
- Dependent only on the previous state

What is an HMM?



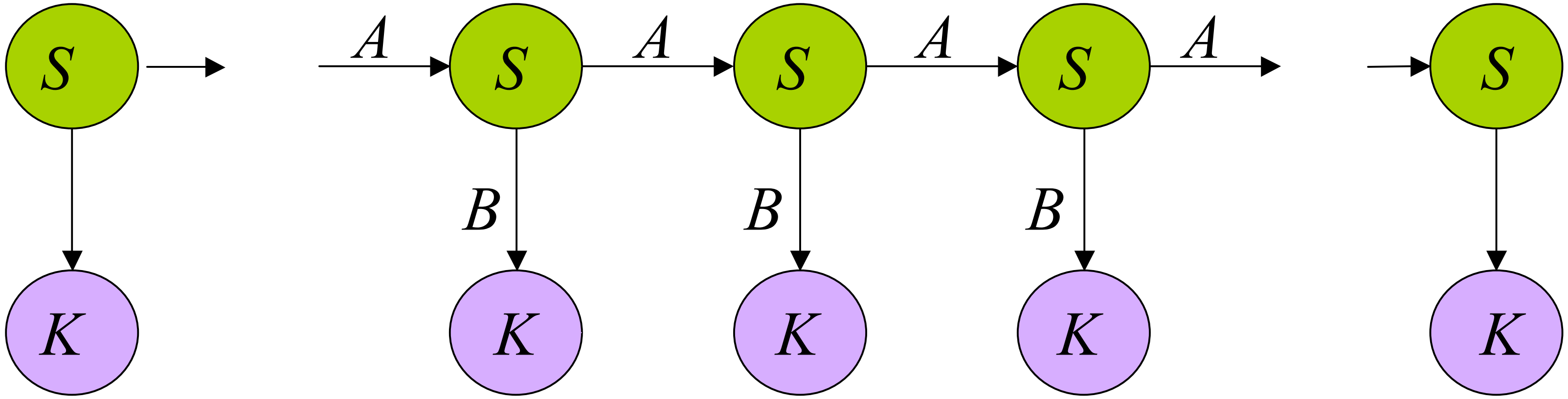
- Purple nodes are ***observed states***
- Dependent only on their corresponding hidden state
- The past is independent of the future given the present

HMM Formalism



- $\{S, K, \Pi, A, B\}$
- 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

HMM Formalism



- $\{S, K, \Pi, A, B\}$
- $\Pi = \{\pi_i\}$ are the initial state probabilities
- $A = \{a_{ij}\}$ are the state transition probabilities
- $B = \{b_{ik}\}$ are the observation state probabilities

Note : sometimes one uses $B = \{b_{ijk}\}$

output then depends on previous state / transition as well

The crazy soft drink machine

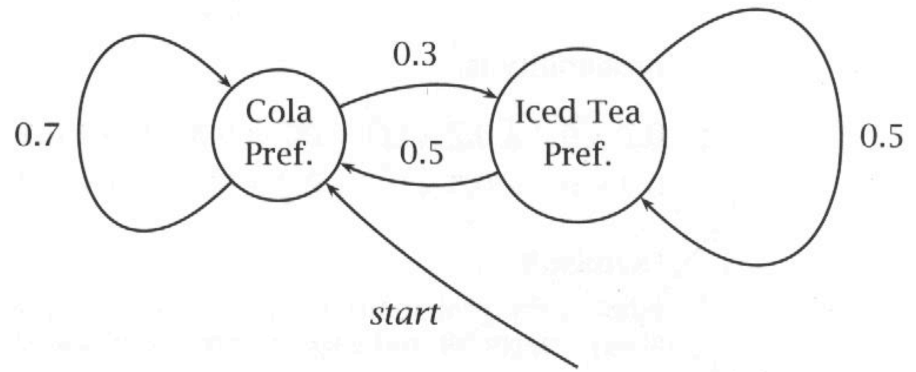


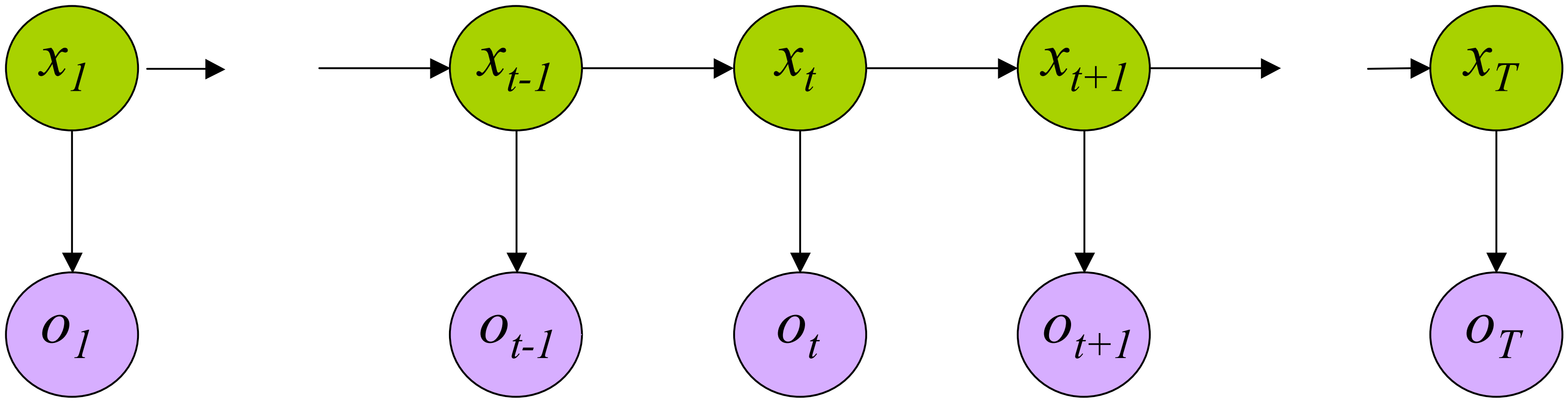
Figure 9.2 The crazy soft drink machine, showing the states of the machine and the state transition probabilities.

B	*cola*	*iced tea*	*lemonade*
CP	0.6	0.1	0.3
IP	0.1	0.7	0.2

Probability of {lem,ice} ?

- Sum over all paths taken through HMM
- Start in CP
  - 1 x 0.3 x 0.7 x 0.1 +
  - 1 x 0.3 x 0.3 x 0.7

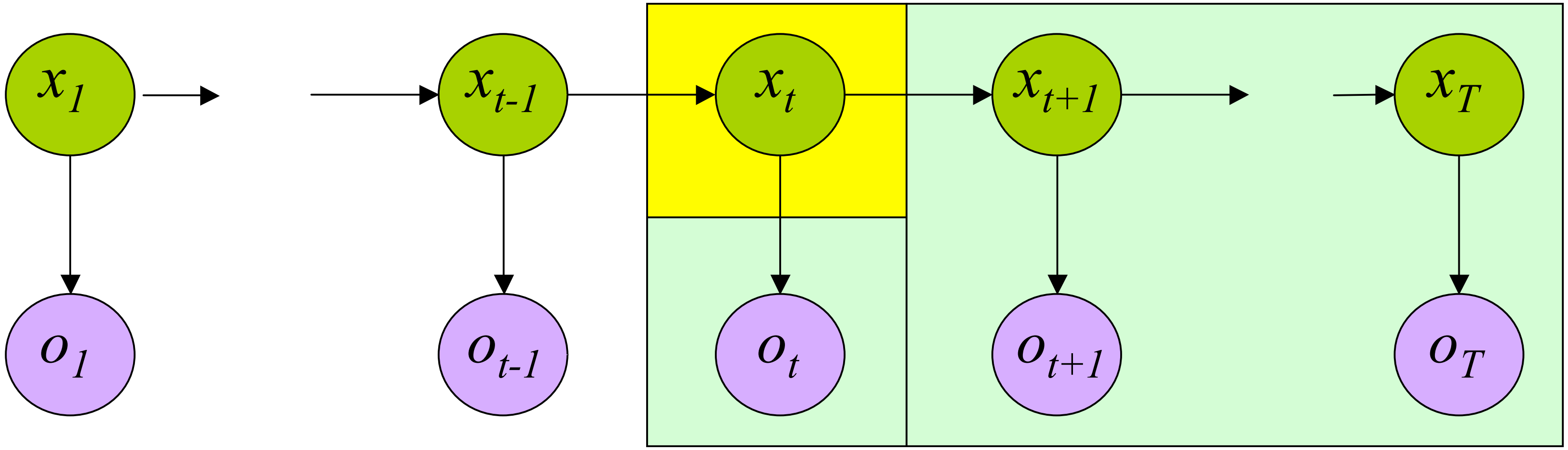
HMMs and Bayesian Nets (1)



$$P(x_1 \ldots x_T, o_1 \ldots o_T) = P(x_1)P(o_1 \mid x_1)\prod_{i=1}^{T-1} P(x_{i+1} \mid x_i).P(o_{i+1} \mid x_{i+1})$$

$$= \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$

HMM and Bayesian Nets (2)

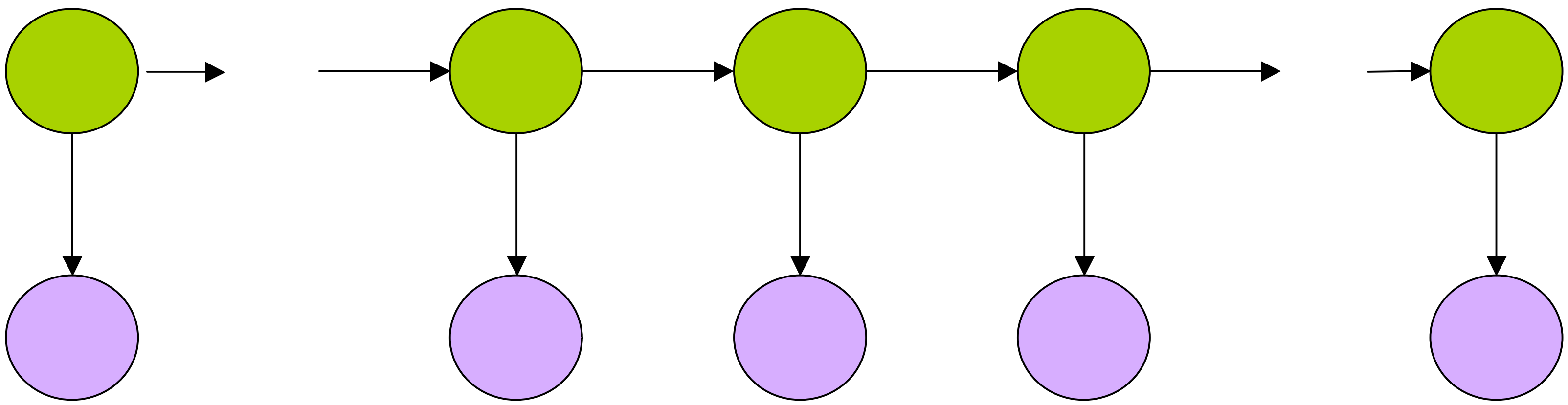


Conditionally independent of Given

Because of d-separation

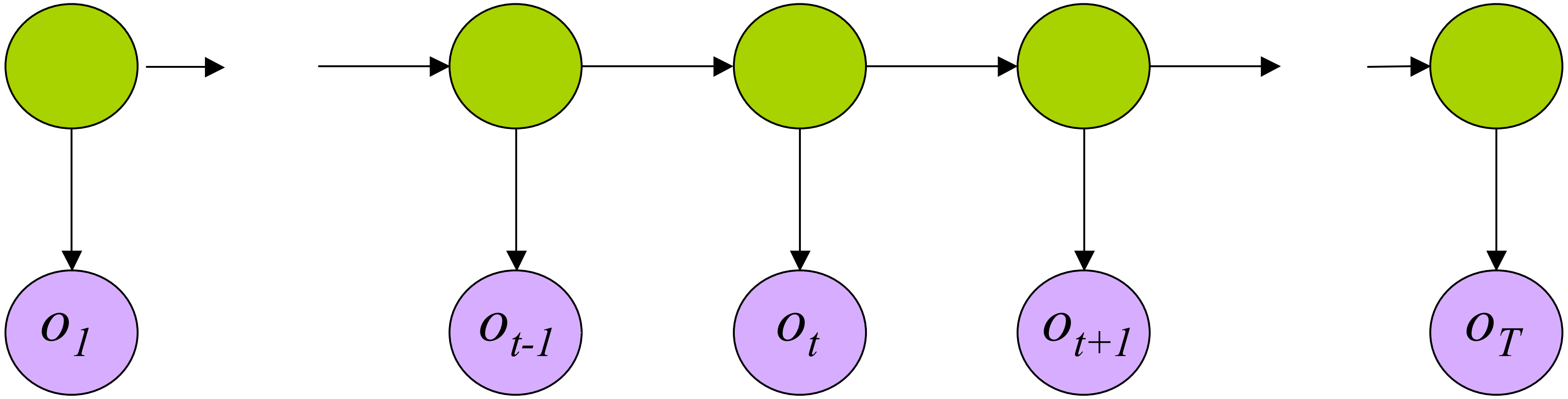
“The past is independent of the future given the present.”

Inference in an HMM



- Compute the probability of a given observation sequence
- Given an observation sequence, compute the most likely hidden state sequence
- Given an observation sequence and set of possible models, which model most closely fits the data?

Decoding

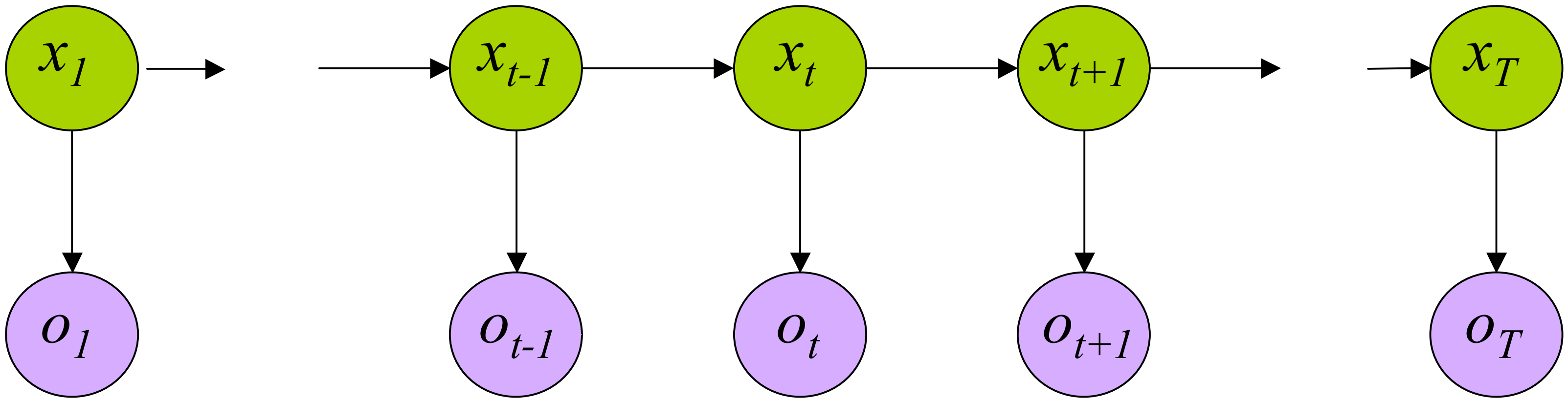


Given an observation sequence and a model, compute the probability of the observation sequence

$$O = (o_1 \ldots o_T),\ \mu = (A, B, \Pi)$$

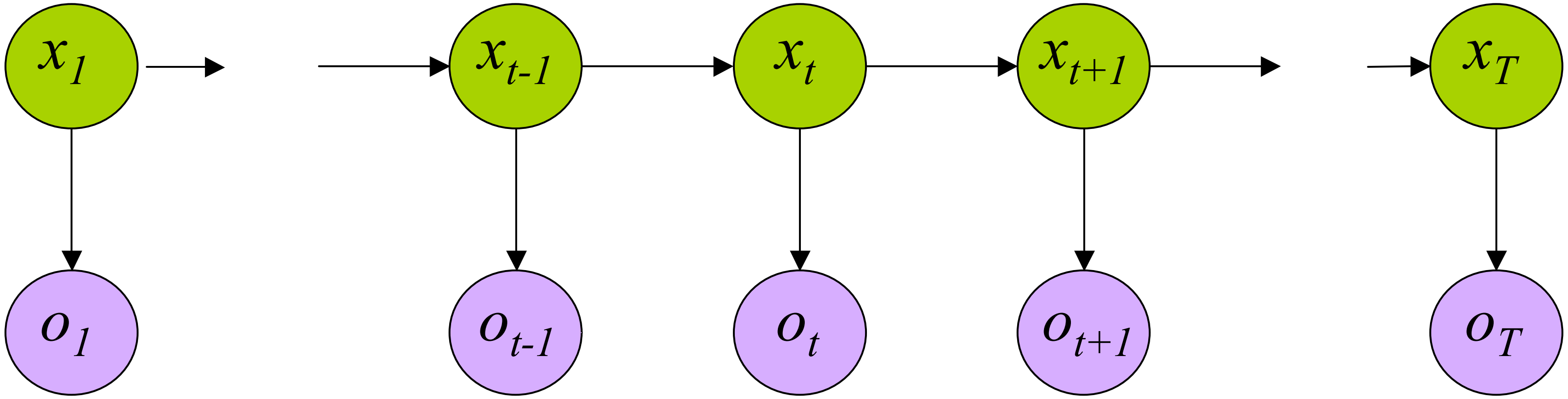
Compute $P(O \mid \mu)$

Decoding



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

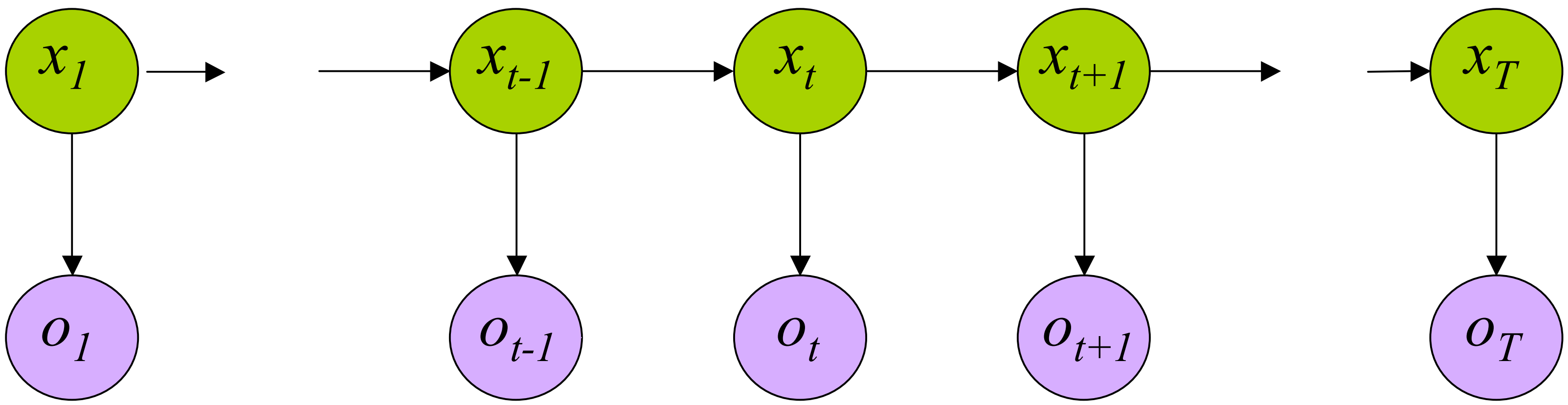
Decoding



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

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

Decoding

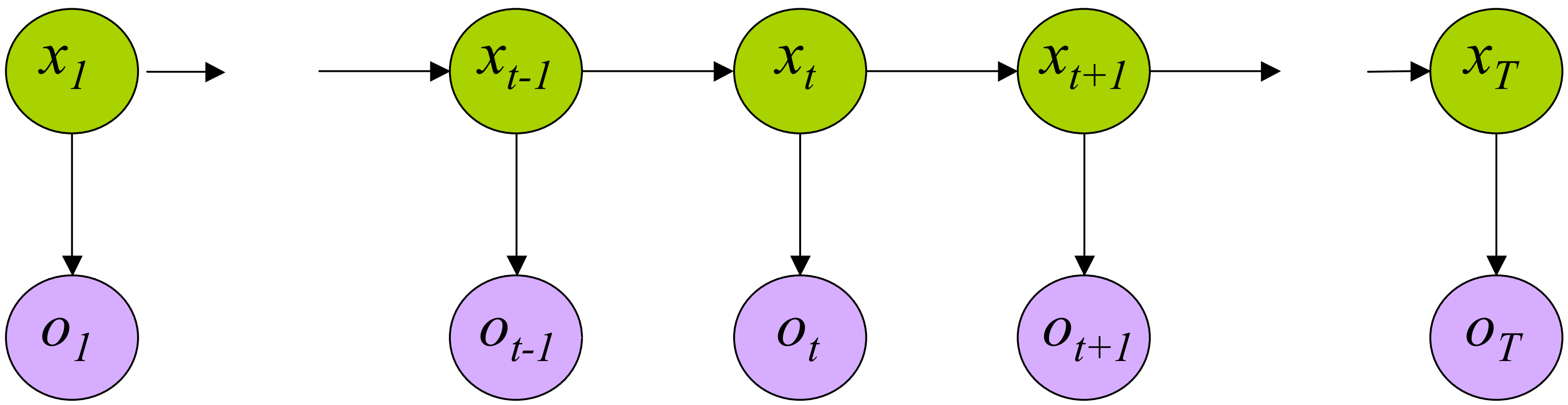


$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

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

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

Decoding



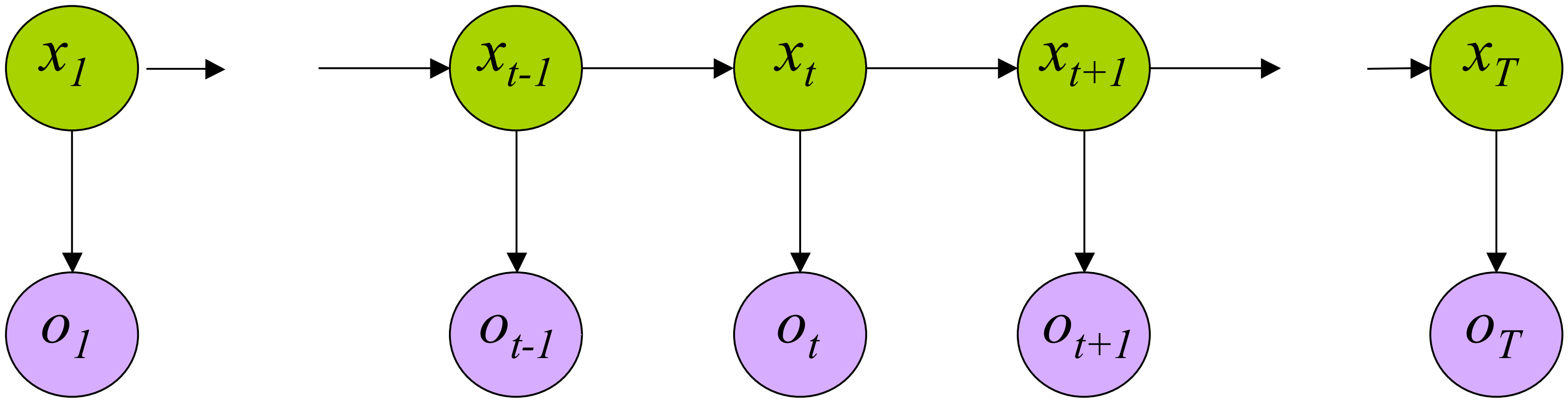
$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

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

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

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

Decoding



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

Complexity $O(N^T.2T)$

$E.g.\ N = 5, T = 100$ gives $2.100.5^{100} \approx 10^{72}$

Dynamic Programming

(a)

(b)

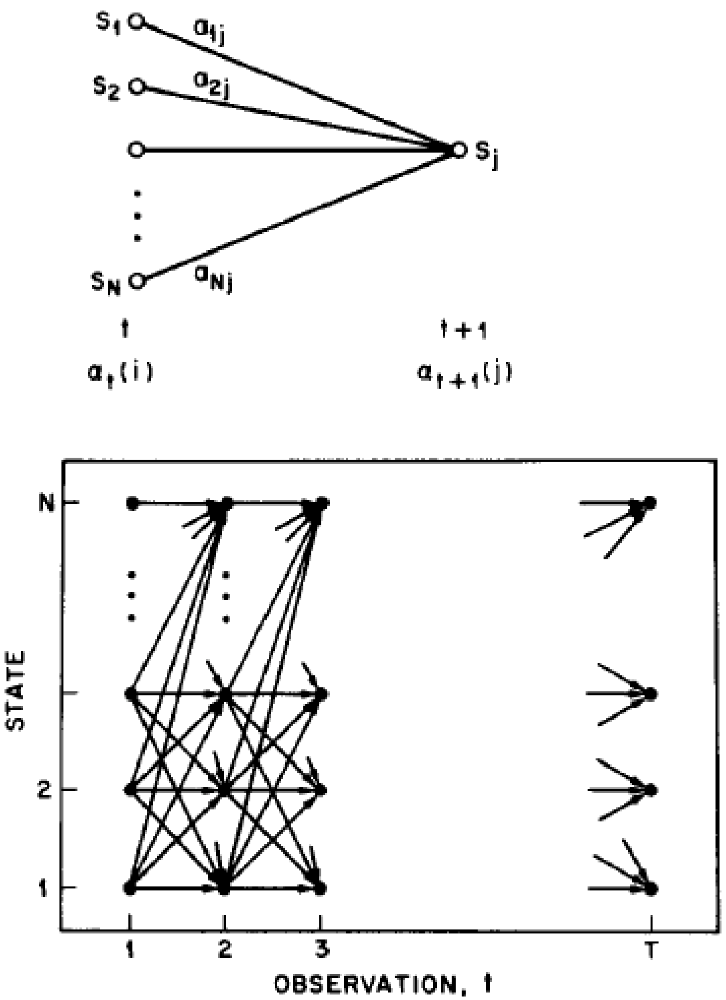
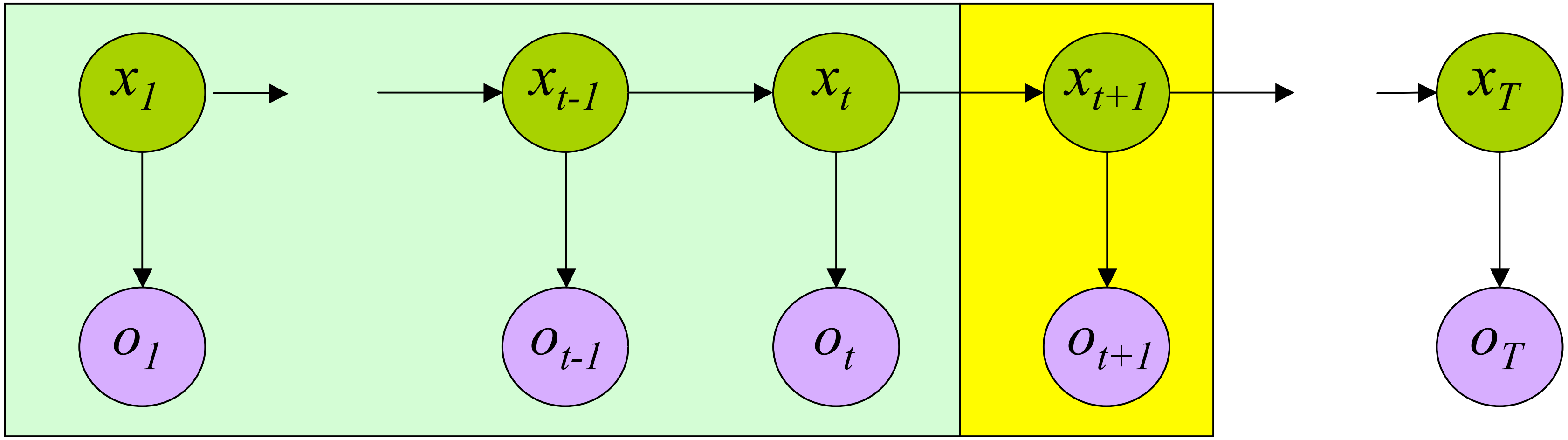


Fig. 4. (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(j)$. (b) Implementation of the computation of $\alpha_t(i)$ in terms of a lattice of observations t, and states i.

Forward Procedure

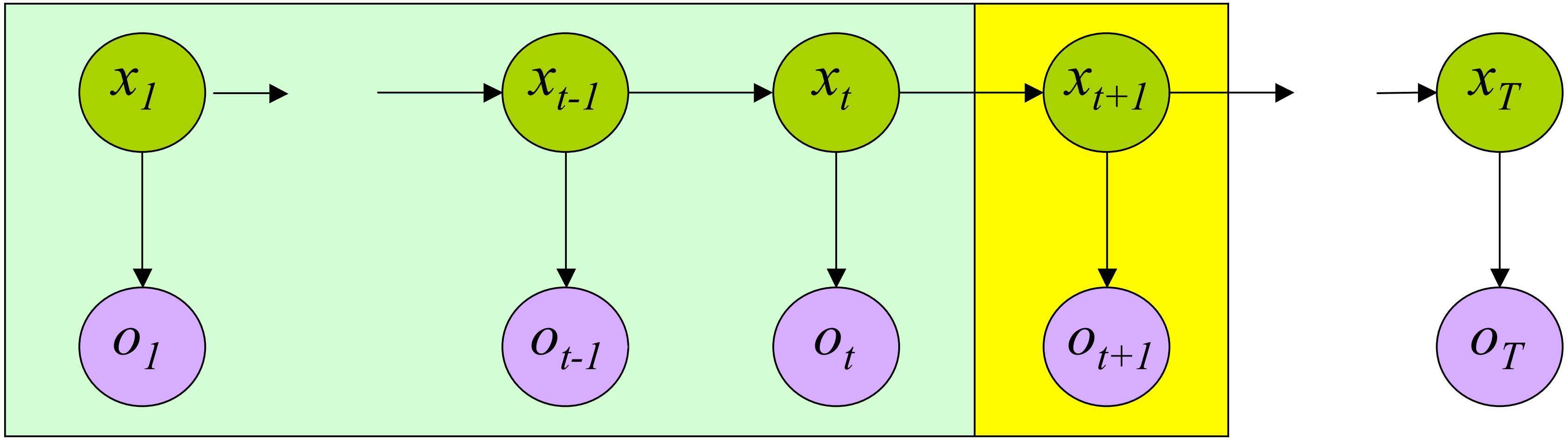


- Special structure gives us an efficient solution using *dynamic programming*.
- **Intuition**: Probability of the fi the same for all possible t+1 le sequences.

$$\alpha_i(1) = P(o_1, x_1 = i \mid \mu)$$
$$= \pi_i . b_{io_1}$$

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

Forward Procedure



$$\alpha_j(t+1)$$

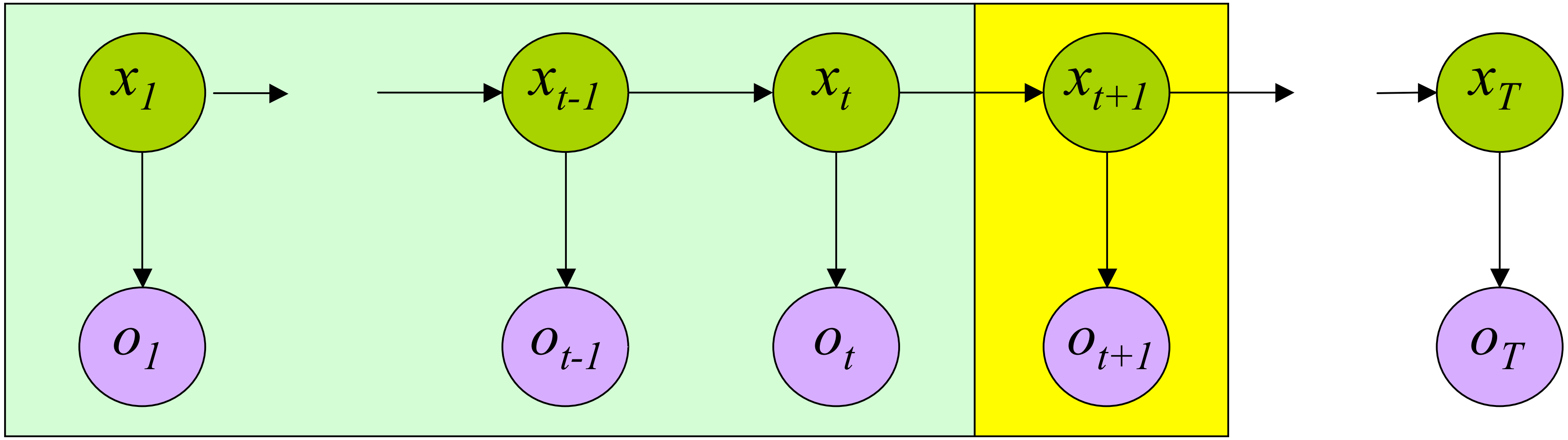
$$= P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

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

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

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

Forward Procedure



$$\alpha_j(t+1)$$

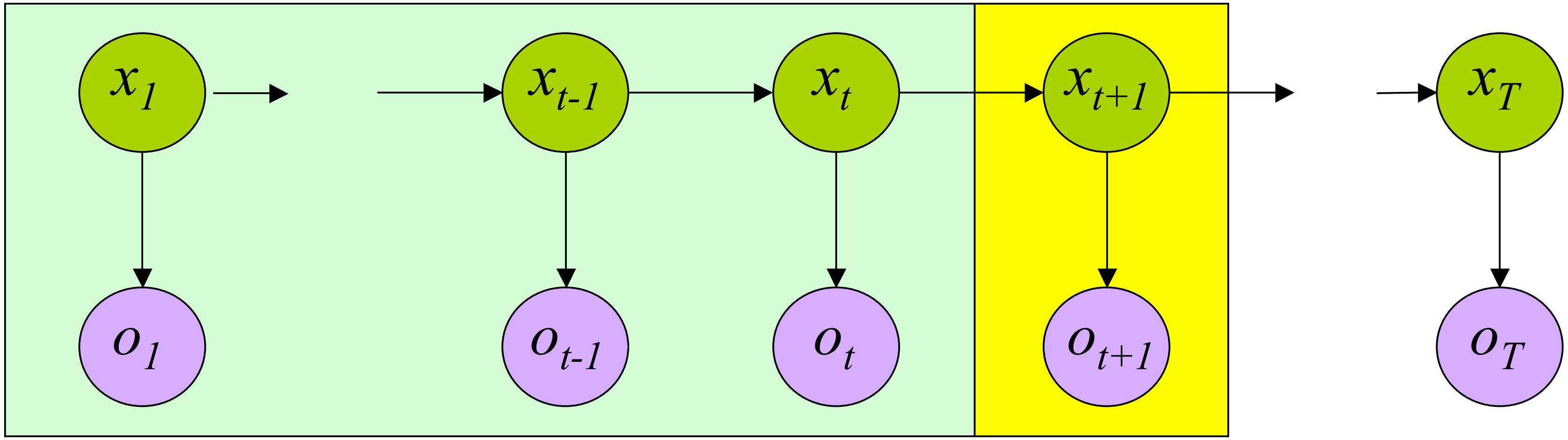
$$= P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

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

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

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

Forward Procedure



$$\alpha_j(t+1)$$

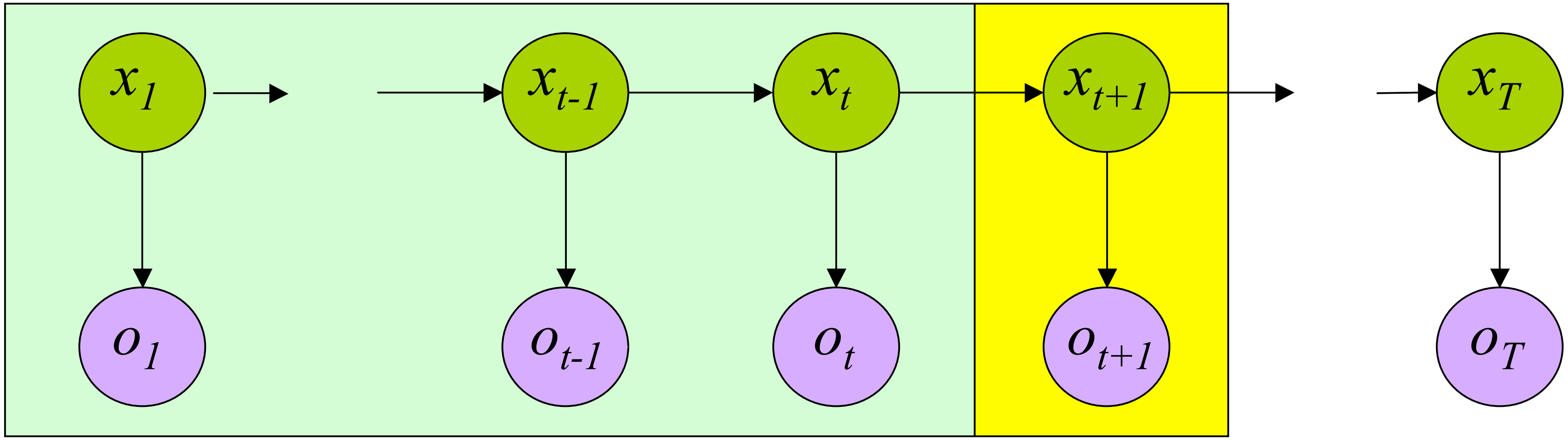
$$= P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

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

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

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

Forward Procedure



$$\alpha_j(t+1)$$

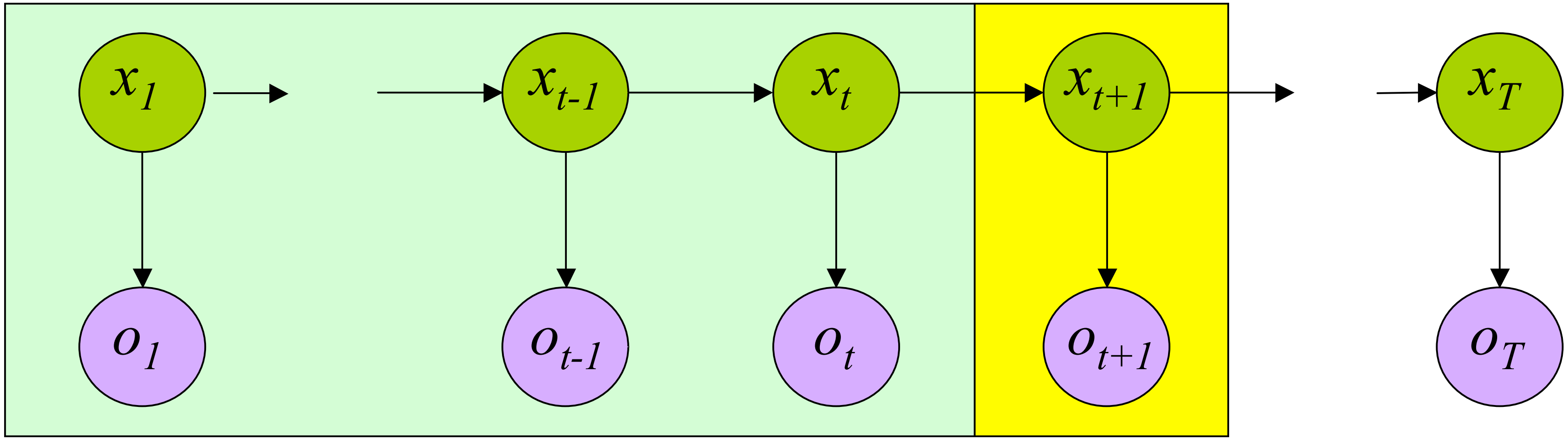
$$= P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

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

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

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

Forward Procedure



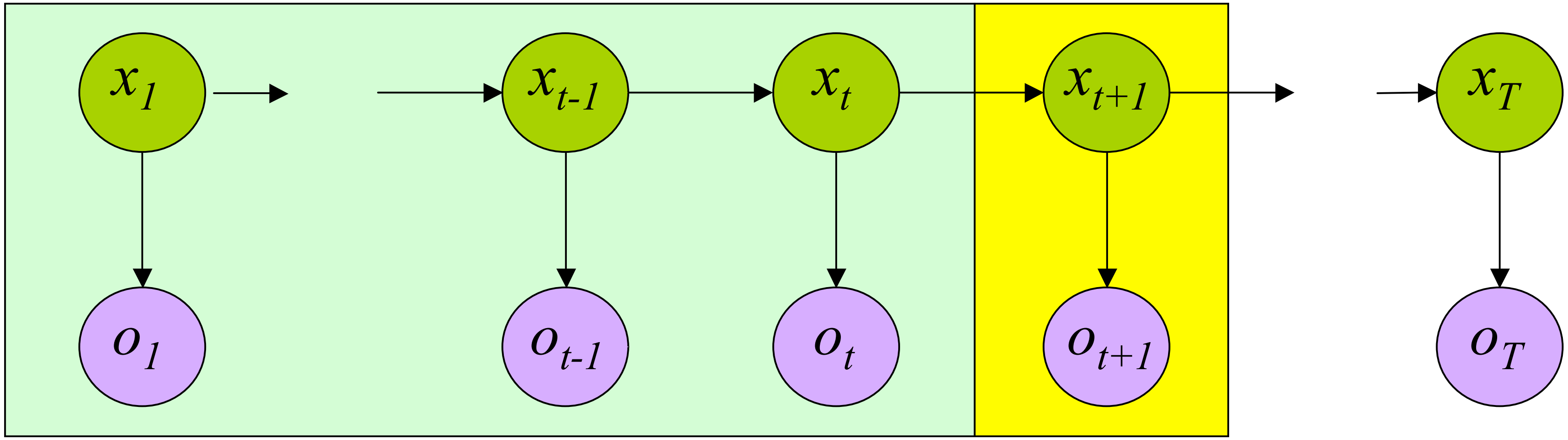
$$= \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

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

Forward Procedure



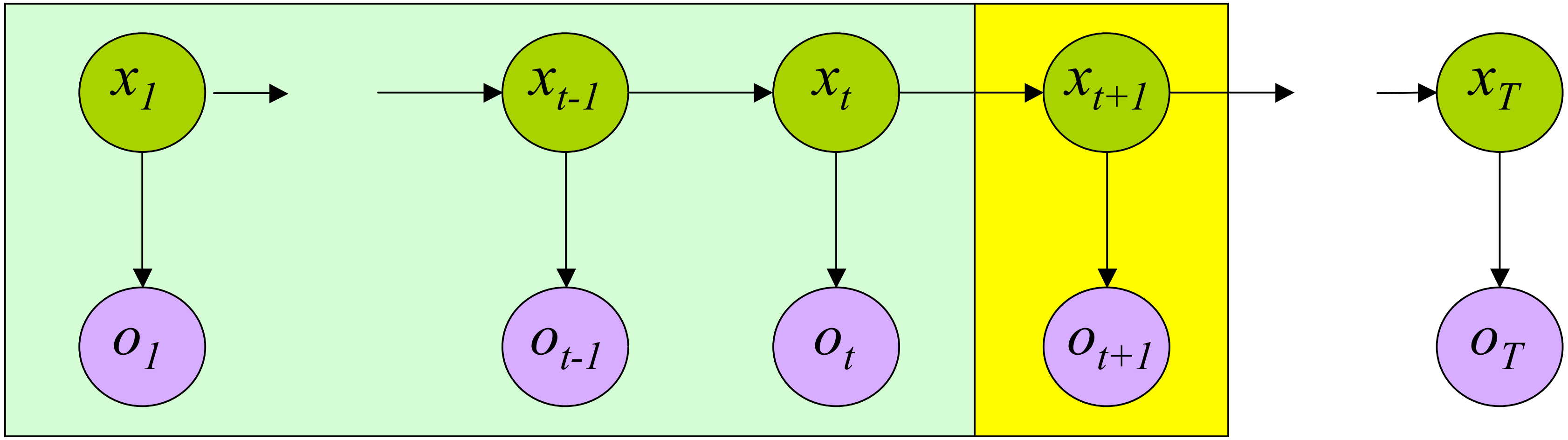
$$= \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

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

Forward Procedure



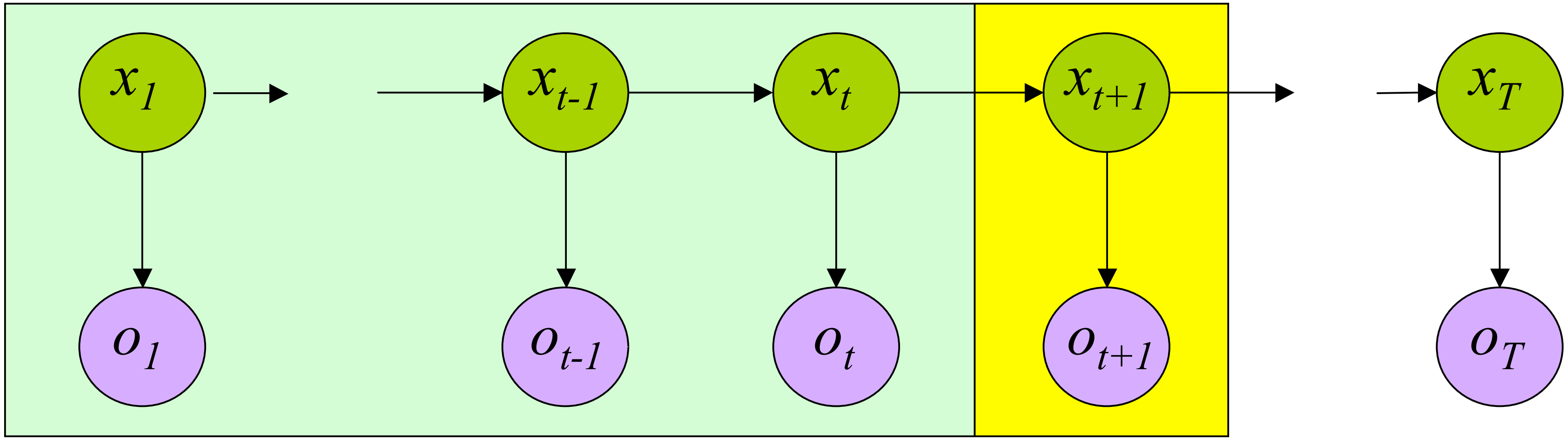
$$= \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

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

Forward Procedure



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

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

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

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

Dynamic Programming

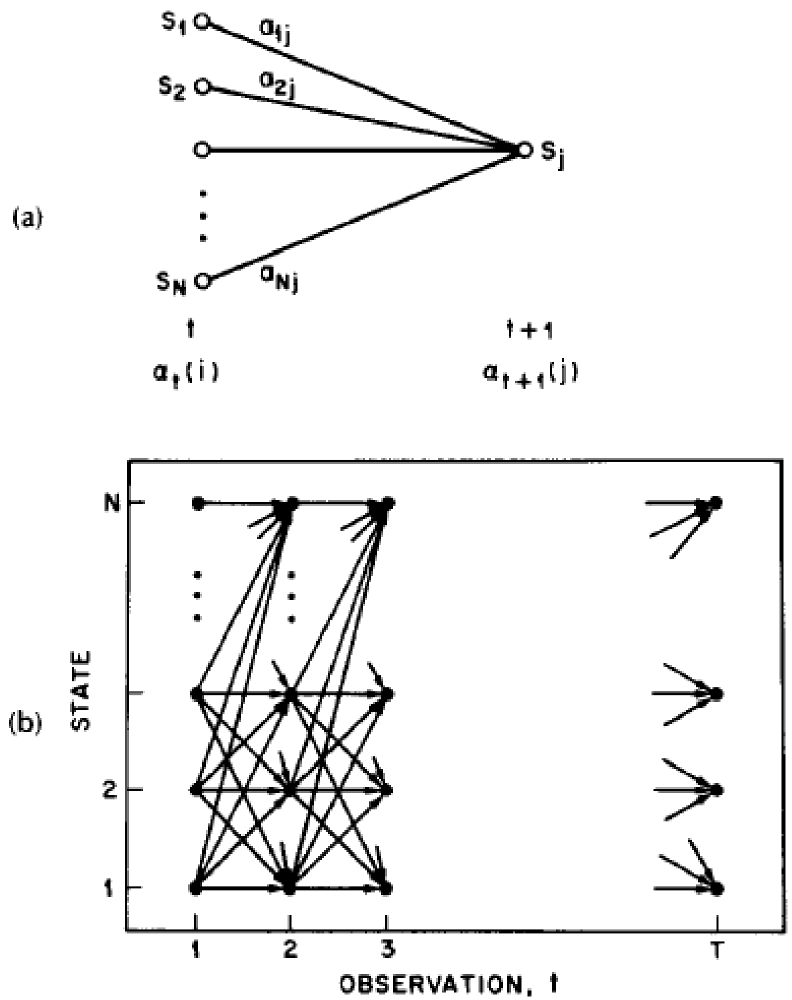


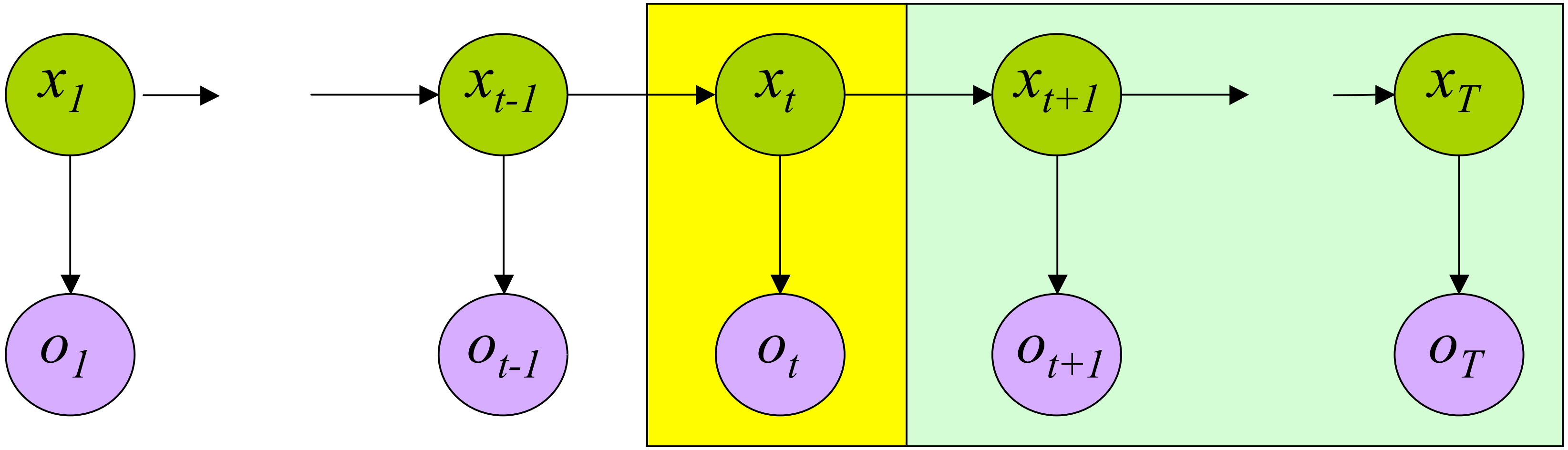
Fig. 4. (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(j)$. (b) Implementation of the computation of $\alpha_t(i)$ in terms of a lattice of observations t, and states i.

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

Complexity $O(N^2.T)$

$E.g.\ N = 5, T = 100$ gives ≈ 3000

Backward Procedure



$$\beta_i(T) = 1$$

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

$$\beta_i(t) = \sum_{j=1\ldots N} a_{ij} b_{io_{t+1}} \beta_j(t+1)$$

Probability of the rest of the states given the first state

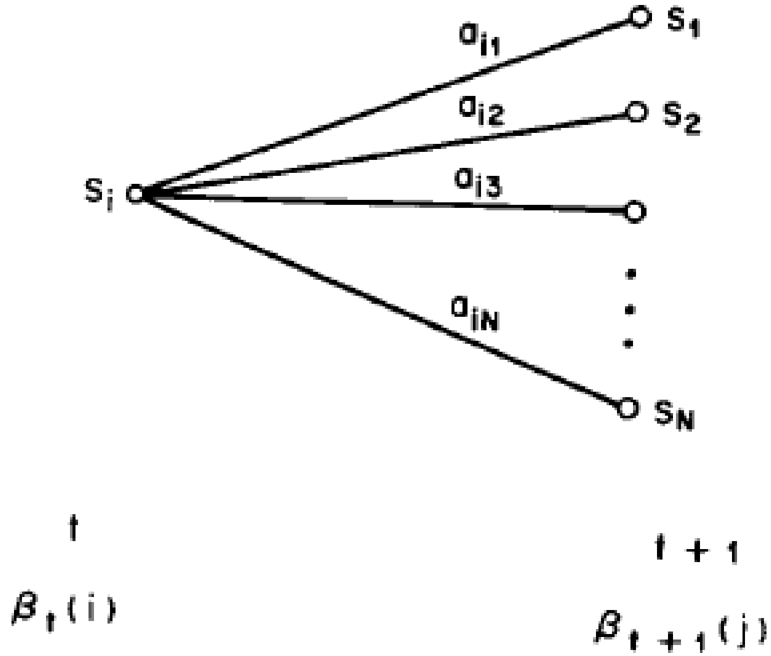
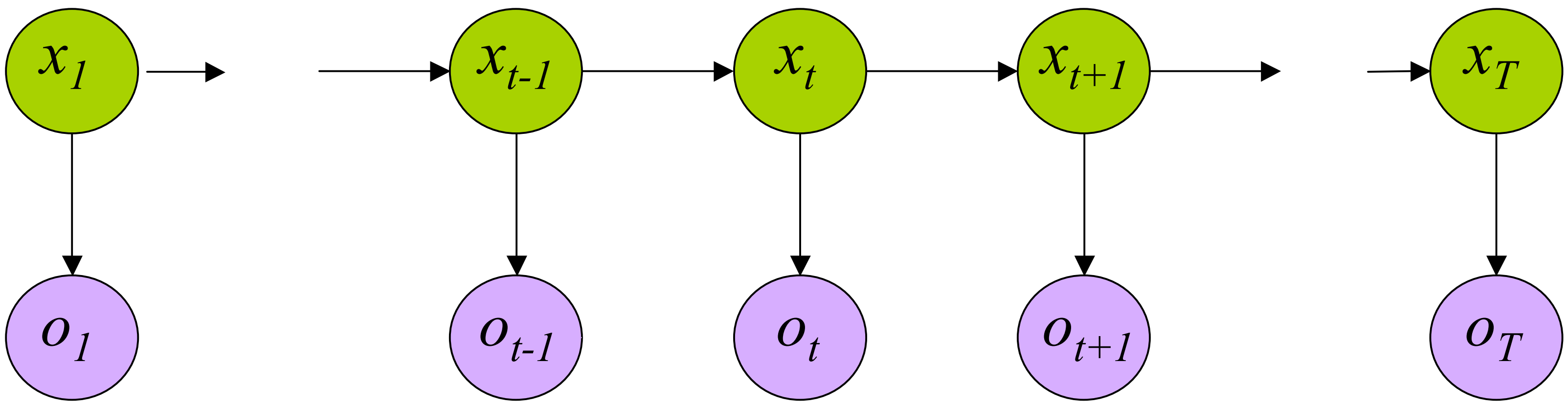


Fig. 5. Illustration of the sequence of operations required for the computation of the backward variable $\beta_t(i)$.

Decoding Solution



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

Forward Procedure

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

Backward Procedure

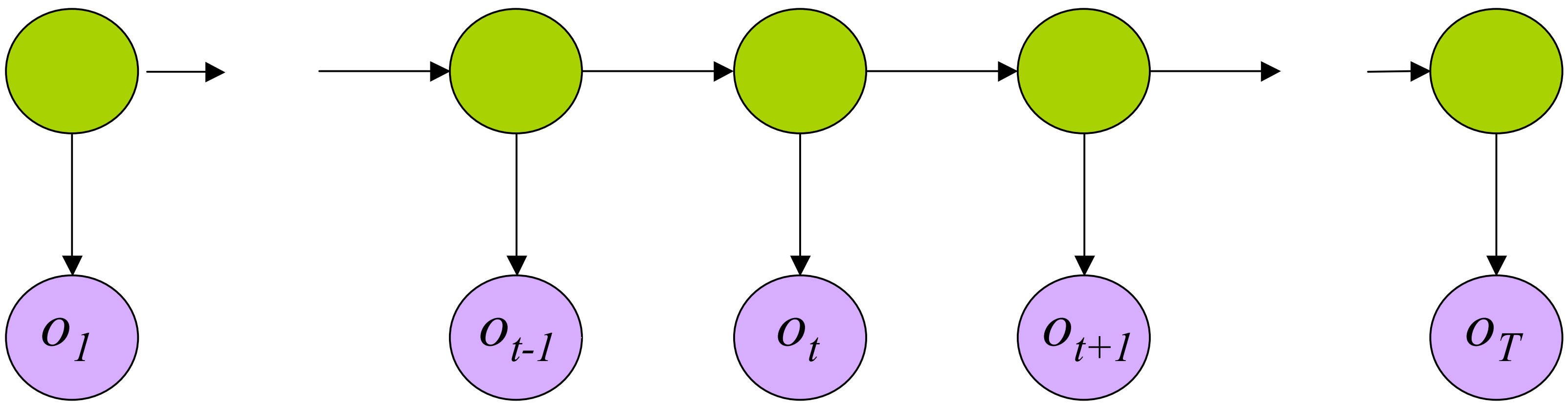
$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(t) \beta_i(t)$$

Combination

$$\begin{aligned} P(O, X_t = i \mid \mu) &= P(o_1 \ldots o_t, X_t = i, o_{t+1} \ldots o_T \mid \mu) \\ &= P(o_1 \ldots o_t, X_t = i \mid \mu).P(o_{t+1} \ldots o_T \mid o_1 \ldots o_t, X_t = i, \mu) \\ &= P(o_1 \ldots o_t, X_t = i \mid \mu).P(o_{t+1} \ldots o_T \mid X_t = i, \mu) \\ &= \alpha_i(t).\beta_i(t) \end{aligned}$$

$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t)$$

Best State Sequence



- Find the state sequence that best explains the observations

- **Two approaches**
  - **Individually most likely states**
  - **Most likely sequence (Viterbi)**

$$\arg\max_X P(X \mid O)$$

-

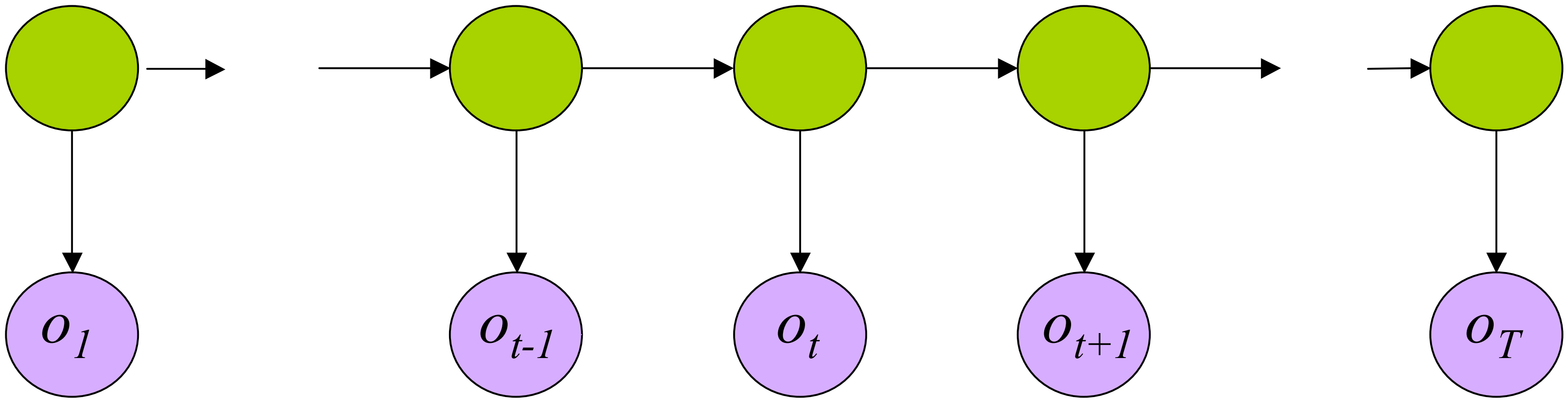
Best State Sequence (1)

$$\gamma_i(t) = P(X_t = i \mid O, \mu)$$
$$= \frac{P(X_t = i, O \mid \mu)}{P(O \mid \mu)}$$
$$= \frac{\alpha_i(t).\beta_i(t)}{\sum_{j=1}^{n} \alpha_j(t).\beta_j(t)}$$

Most likely state at each point in time

$$\hat{X}_t = \arg\max \gamma_i(t)$$

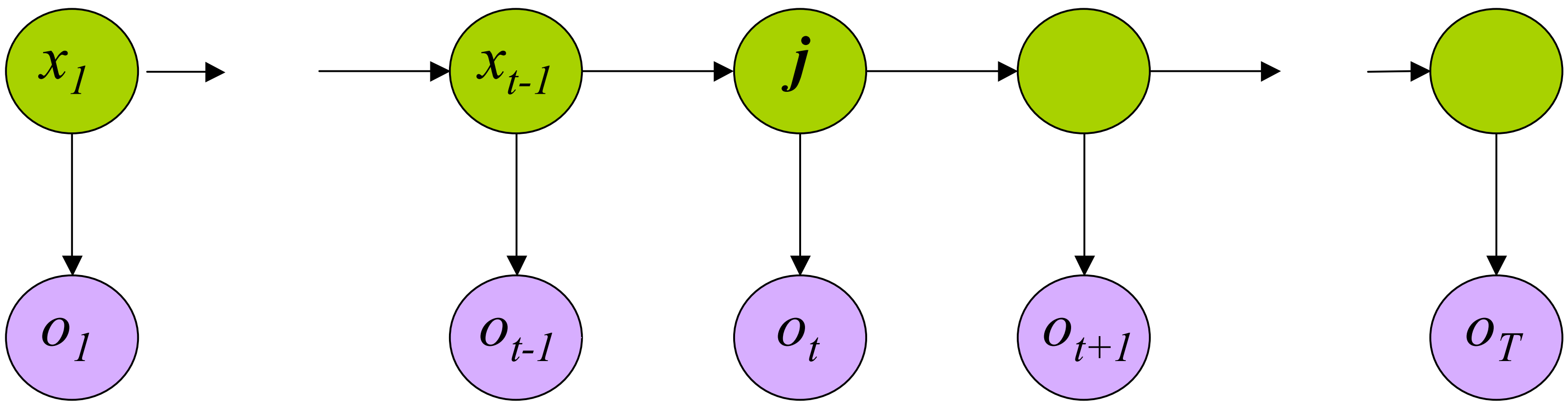
Best State Sequence (2)



- Find the state sequence that best explains the observations
- Viterbi algorithm

$$\arg\max_X P(X \mid O)$$

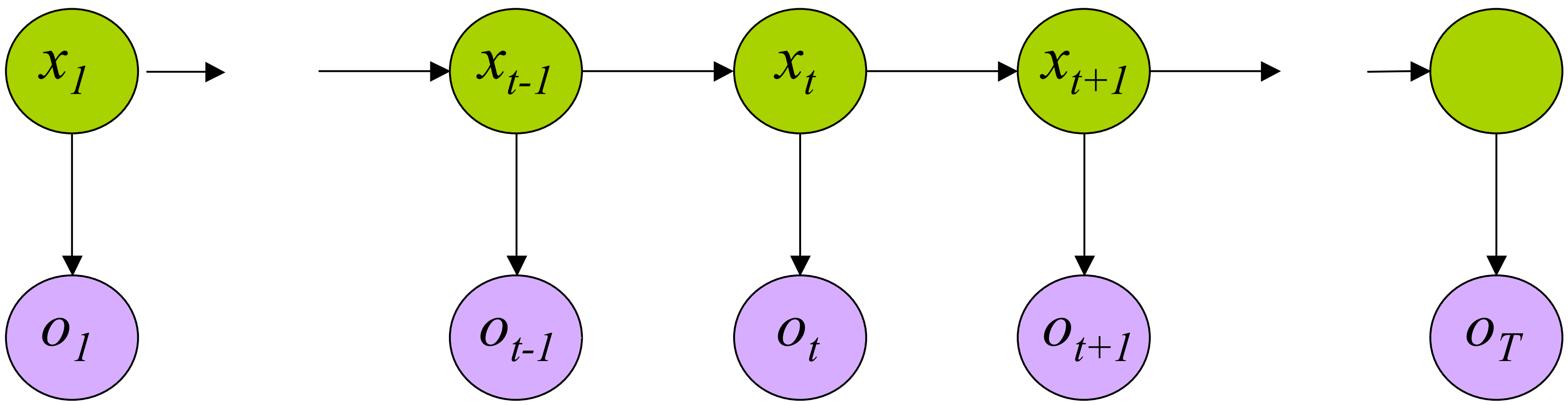
Viterbi Algorithm



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

The state sequence which maximizes the probability of seeing the observations to time t-1, landing in state j, and seeing the observation at time t

Viterbi Algorithm



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

$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_{j o_{t+1}}$$

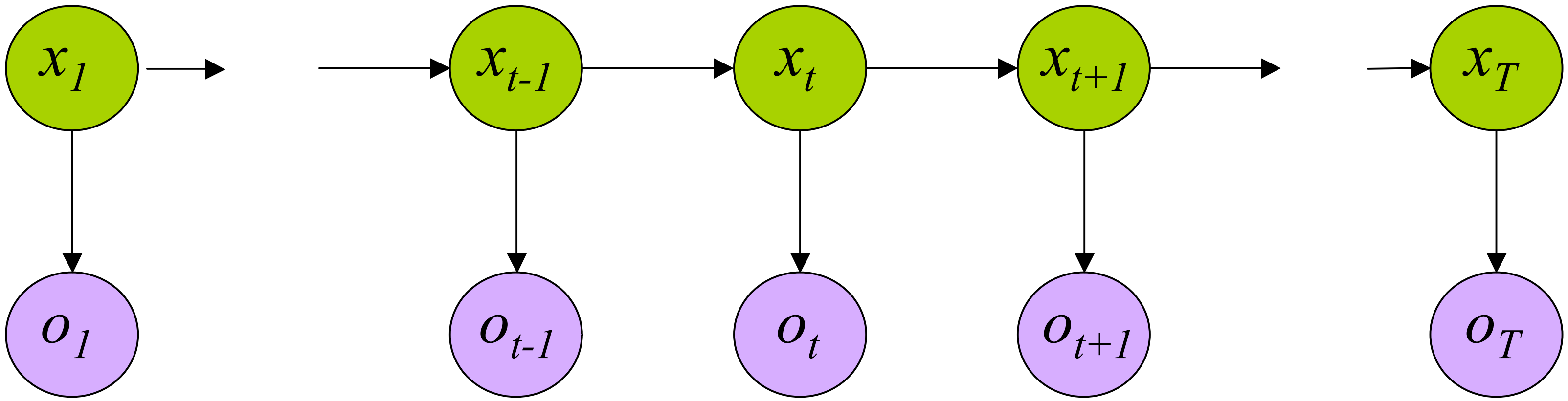
$$\psi_j(t+1) = \arg\max_i \delta_i(t) a_{ij} b_{j o_{t+1}}$$

Initialization

$$\delta_1(i) = \pi_i b_{i o_1}$$

$$\psi_1(i) = 0$$

Viterbi Algorithm



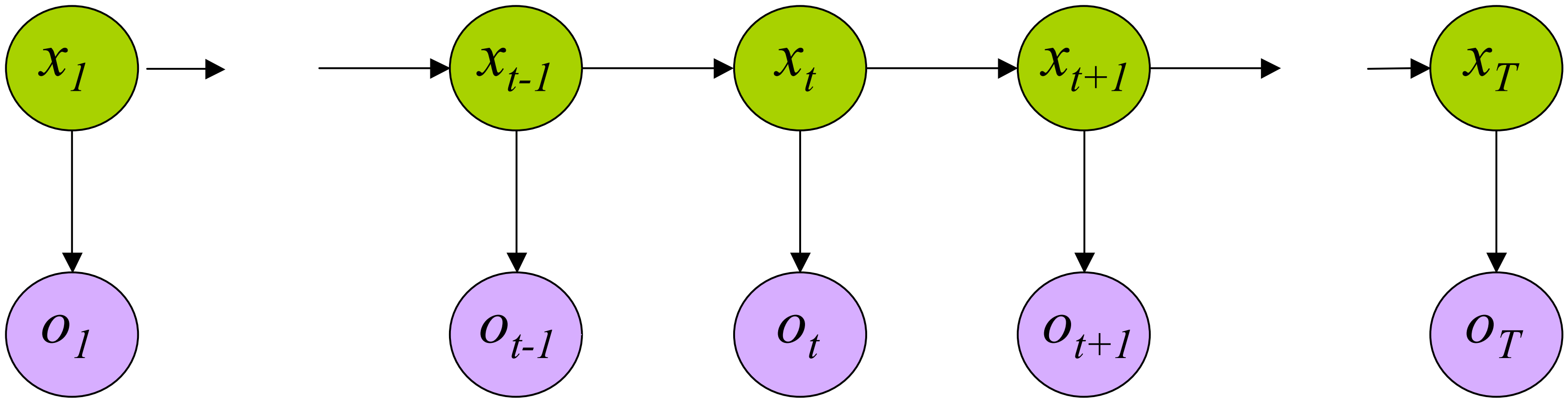
$$\hat{X}_T = \arg\max_i \delta_i(T)$$

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

$$P(\hat{X}) = \arg\max_i \delta_i(T)$$

Compute the most likely state sequence by working backwards

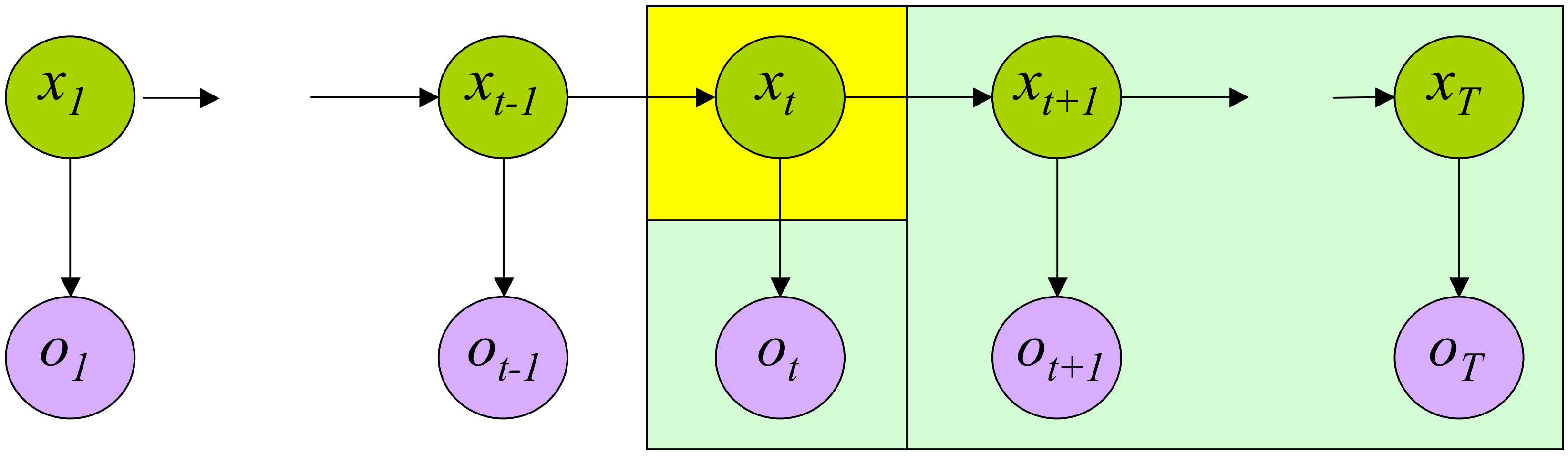
HMMs and Bayesian Nets (1)

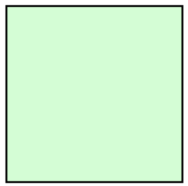


$$P(x_1 \ldots x_T, o_1 \ldots o_T) = P(x_1)P(o_1 \mid x_1)\prod_{i=1}^{T-1} P(x_{i+1} \mid x_i).P(o_{i+1} \mid x_{i+1})$$

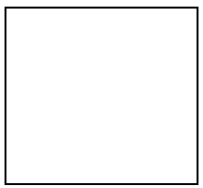
$$= \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$

HMM and Bayesian Nets (2)

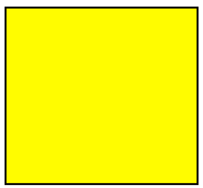




Conditionally independent of



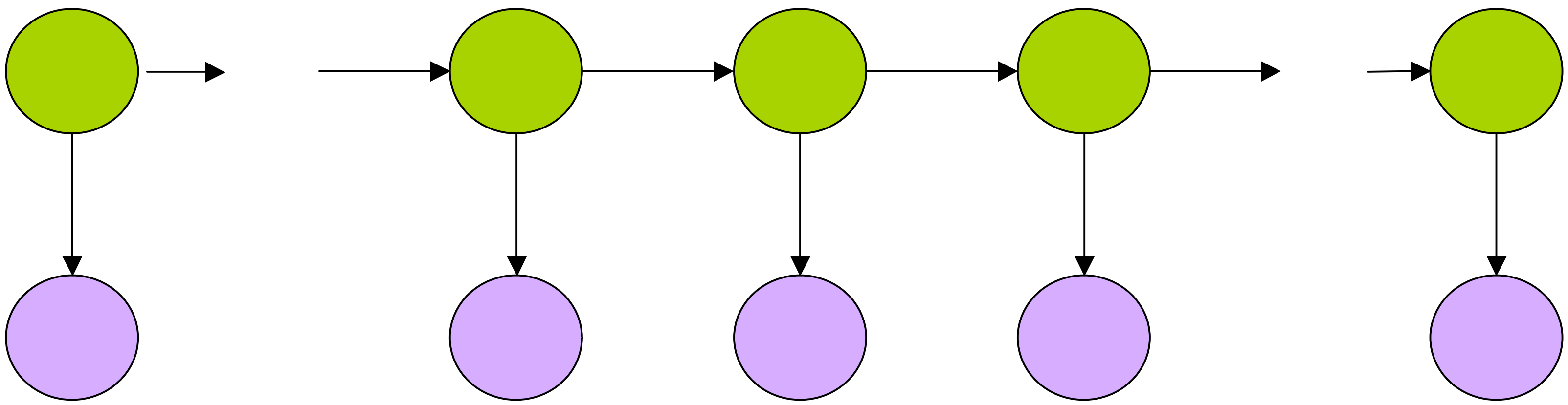
Given



Because of d-separation

"The past is independent of the future given the present."

Inference in an HMM



- Compute the probability of a given observation sequence
- Given an observation sequence, compute the most likely hidden state sequence
- Given an observation sequence and set of possible models, which model most closely fits the data?

Dynamic Programming

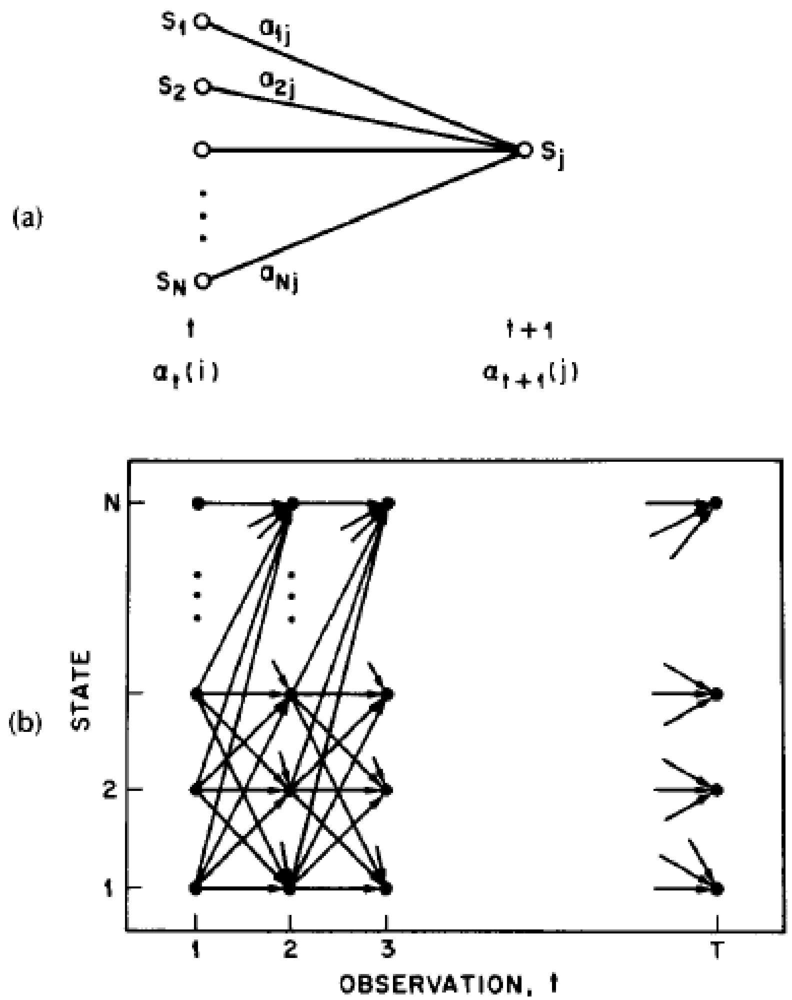
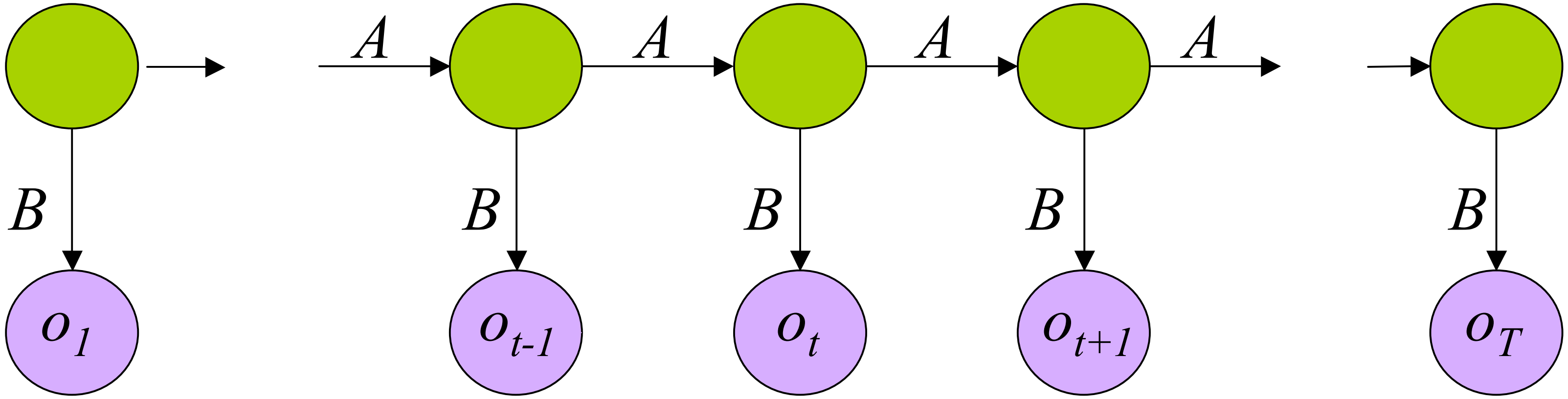


Fig. 4. (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(j)$. (b) Implementation of the computation of $\alpha_t(i)$ in terms of a lattice of observations t, and states i.

Parameter Estimation



- Given an observation sequence, find the model that is most likely to produce that sequence.
- No analytic method $\arg\max_{\mu} P(O_{training} \mid \mu)$
- Given a model and observation sequence, update the model parameters to better fit the observations.

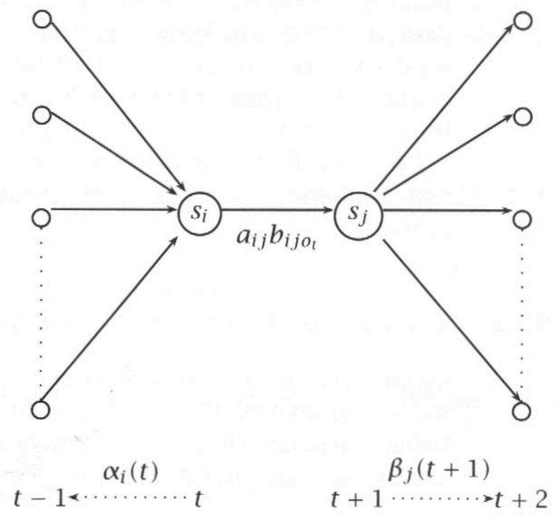


Figure 9.7 The probability of traversing an arc. Given an observation sequence and a model, we can work out the probability that the Markov process went from state s_i to s_j at time t.

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

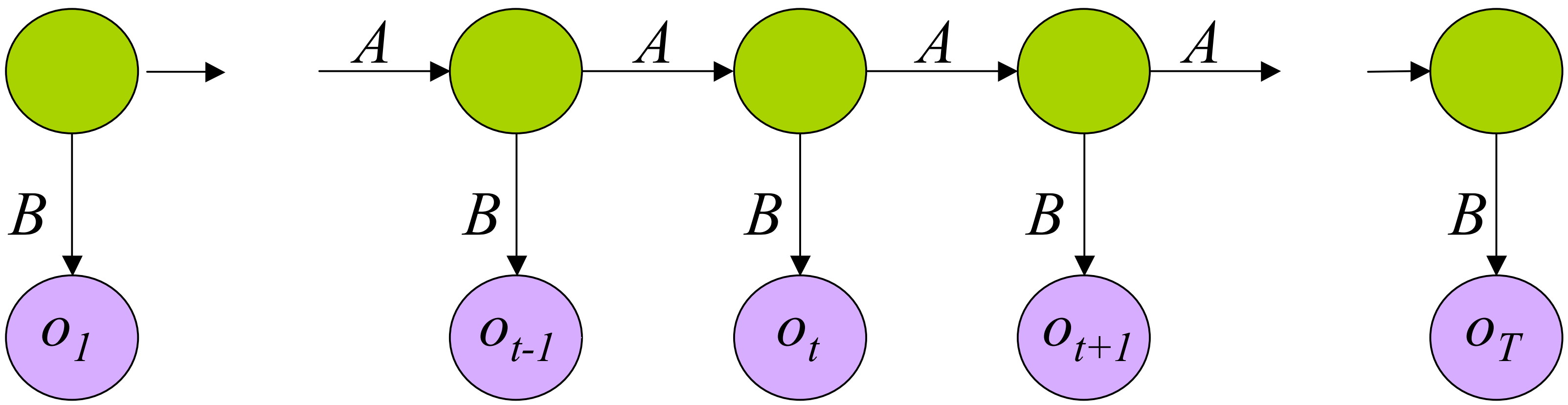
$$\beta_i(t) = P(o_{t+1} \dots o_T \mid x_t = i)$$

$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t)$$

$$p_t(i,j) = P(X_t = i, X_{t+1} = j \mid O, \mu)$$

$$= \frac{P(X_t = i, X_{t+1} = j, O \mid \mu)}{P(O \mid \mu)}$$

Parameter Estimation



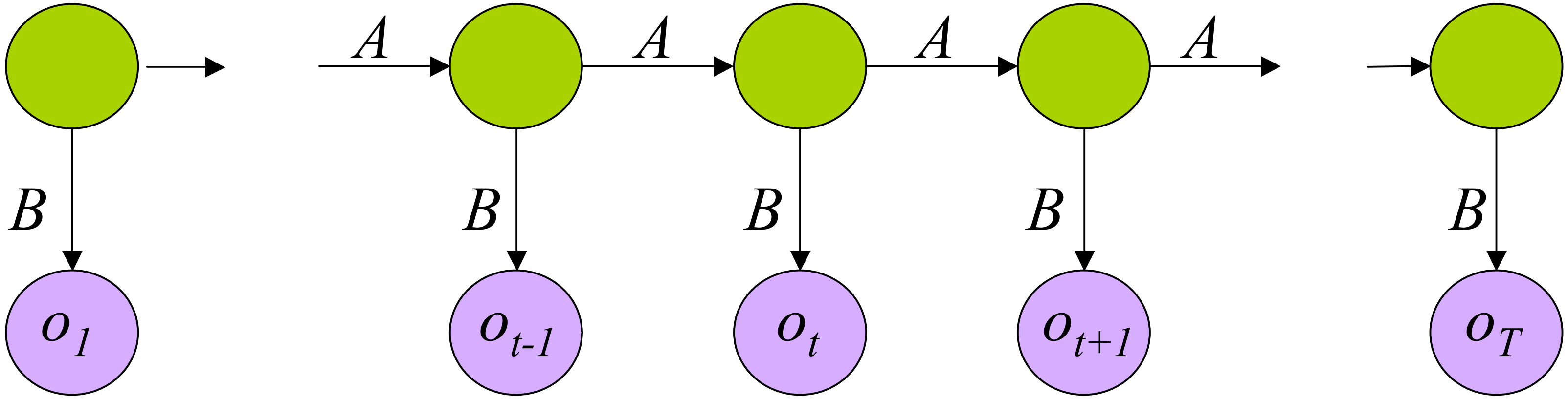
$$p_t(i,j) = \frac{\alpha_i(t)a_{ij}b_{jo_{t+1}}\beta_j(t+1)}{\sum_{m=1\ldots N}\alpha_m(t)\beta_m(t)}$$

Probability of traversing an arc

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

Probability of being in state *i*

Parameter Estimation



$$\hat{\pi}_i = \gamma_i(1)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} p_t(i,j)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$\hat{b}_{ik} = \frac{\sum_{\{t:o_t=k\}} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

Now we can compute the new estimates of the model parameters.

Instance of Expectation Maximization

- We have that

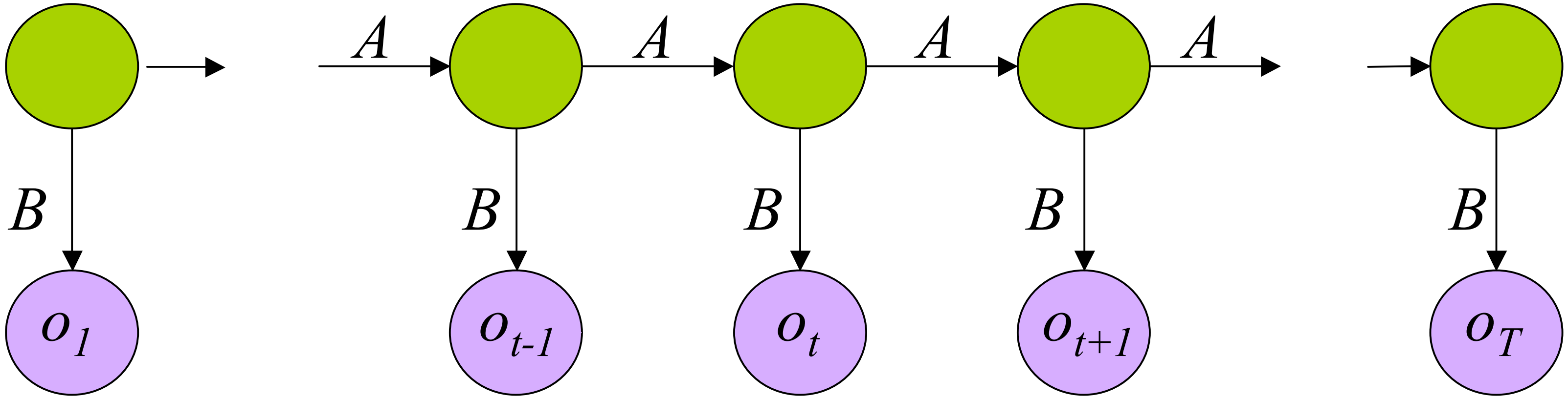
$$P(O \mid \hat{\mu}) \geq P(O \mid \mu)$$

- We may get stuck in local maximum (or even saddle point)
- Nevertheless, Baum-Welch usually effective

Some Variants

- So far, ergodic models
  - All states are connected
  - Not always wanted
- Epsilon or null-transitions
  - Not all states/transitions emit output symbols
- Parameter tying
  - Assuming that certain parameters are shared
  - Reduces the number of parameters that have to be estimated
- Logical HMMs (Kersting, De Raedt, Raiko)
  - Working with structured states and observation symbols
- Working with log probabilities and addition instead of multiplication of probabilities (typically done)

The Most Important Thing



We can use the special structure of this model to do a lot of neat math and solve problems that are otherwise not solvable.

HMM's from an Agent Perspective

- AI: a modern approach
  - AI is the study of rational agents
  - Third part by Wolfram Burgard on Reinforcement learning
- HMMs can also be used here
  - Typically one is interested in P(state)

$$P(X_t = i | o_1, ..., o_T)$$

Example

- Possible states
  - {snow, no snow}
- Observations
  - {skis , no skis }
- Questions
  - Was there snow the day before yesterday (given a sequence of observations) ?
  - Is there now snow (given a sequence of observations) ?
  - Will there be snow tomorrow, given a sequence of observations? Next week ?

HMM and Agents

- Question $P(X_t = i | o_1, ..., o_T)$
  - Case 1 : often called smoothing
    - t < T : see last time

$$\gamma_i(t) = P(X_t = i \mid O, \mu)$$
$$= \frac{\alpha_i(t).\beta_i(t)}{\sum_{j=1}^{n} \alpha_j(t).\beta_j(t)}$$

Most likely state at each point in time

♣ Only part of trellis between t and T needed

$$P(X_t = i | o_1, ..., o_T)$$

- Case 2 : often called filtering
  - t= T : last time

$$\gamma_i(t) = P(X_t = i \mid O, \mu) = \frac{\alpha_i(t).\beta_i(t)}{\sum_{j=1}^{n} \alpha_j(t).\beta_j(t)}$$

Most likely state at each point in time

♣ Can we make it recursive ? I.e go from T-1 to T ?

$$P(X_t = i | o_1, \ldots, o_T)$$

- Case 2 : often called filtering
  - t= T : last time

$$\lambda_i(T) = P(X_T = i \mid o_1 \ldots o_T, \mu)$$

$$= \gamma_i(T)$$

$$= \frac{\alpha_i(T).\beta_i(T)}{\sum_{j=1}^{n} \alpha_j(T).\beta_j(T)}$$

$$= \frac{\alpha_i(T)}{\sum_{j=1}^{n} \alpha_j(T)}$$

HMM and Agents

$$P(X_t = i | o_1, ..., o_T)$$

- Case 3 : often called prediction
  - t= T+1 (or T+K) not yet seen

$$\begin{aligned} & P(X_{T+1} = i | o_1, ..., o_T) \\ = & \sum_j P(X_{T+1} = i | X_T = j, o_1, ..., o_T).P(X_T = j | o_1, ..., o_T) \\ = & \sum_j P(X_{T+1} = i | X_T = j).P(X_T = j | o_1, ..., o_T) \end{aligned}$$

  - Interesting : recursive
  - Easily extended towards k > 1

Extensions

- Use Dynamic Bayesian networks instead of HMMs
  - One state corresponds to a Bayesian Net
  - Observations can become more complex
- Involve actions of the agent as well
  - Cf. Wolfram Burgard’s Part