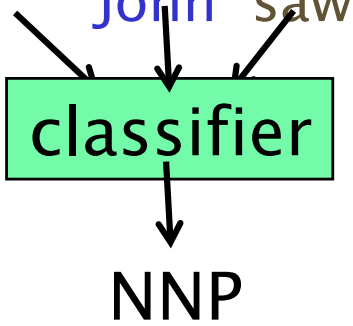
Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.



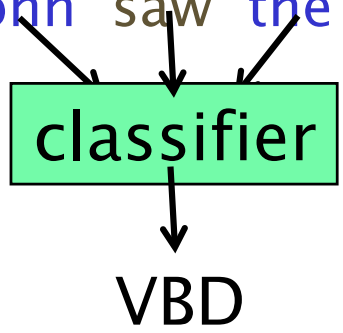
Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

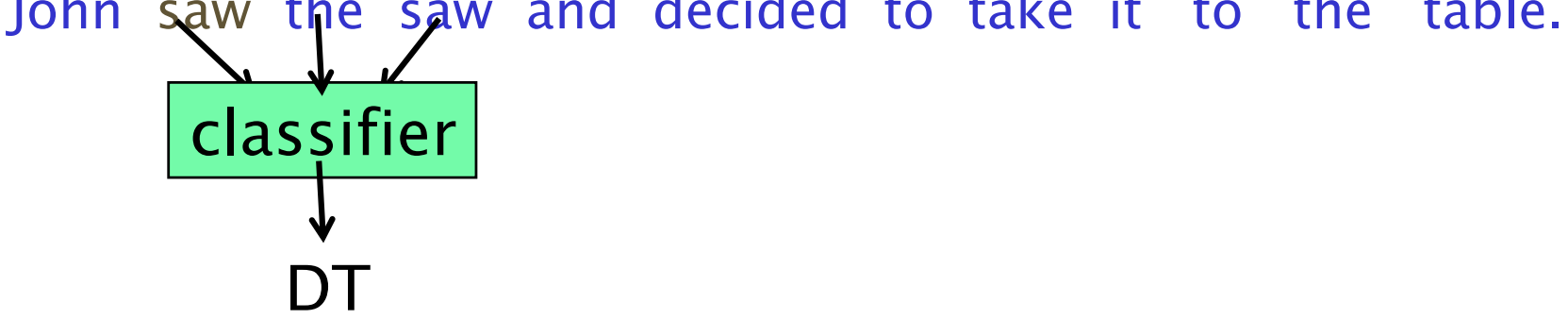


Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



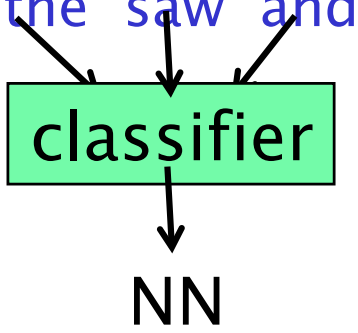
Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.



Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

5. Word-Window classification

- **Idea**: classify a word in its context window of neighboring words.

- For example, **Named Entity Classification** of a word in context:
  - Person, Location, Organization, None

- A simple way to classify a word in context might be to average the word vectors in a window and to classify the average vector
  - Problem: that would lose position information

26

Christopher Manning, Natural Language Processing with Deep Learning, Standford U. CS224n

Window classification: Softmax

- Train softmax classifier to classify a center word by taking concatenation of word vectors surrounding it in a window

- Example: Classify "Paris" in the context of this sentence with window length 2:

$$X_{window} = [\, x_{museums} \quad x_{in} \quad x_{Paris} \quad x_{are} \quad x_{amazing} \,]^T$$

- Resulting vector $x_{window} = x \in R^{5d}$, a column vector!

Christopher Manning, Natural Language Processing with Deep Learning, Standford U. CS224n

Simplest window classifier: Softmax

- With $x = x_{window}$ we can use the same softmax classifier as before

predicted model output probability

$$\hat{y}_y = p(y|x) = \frac{\exp(W_y.x)}{\sum_{c=1}^{C} \exp(W_c.x)}$$

- With cross entropy error as before:

same

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} -\log \left(\frac{e^{f_{y_i}}}{\sum_{c=1}^{C} e^{f_c}} \right)$$

- How do you update the word vectors?

  Short answer: Just take derivatives like last week and optimize

28

Christopher Manning, Natural Language Processing with Deep Learning, Standford U. CS224n

Slightly more complex: Multilayer Perceptron

- Introduce an additional layer in our softmax classifier with a non-linearity.
- MLPs are fundamental building blocks of more complex neural systems!
- Assume we want to classify whether the center word is a Location
- Similar to word2vec, we will go over all positions in a corpus. But this time, it will be supervised s.t. positions that are true NER Locations should assign high probability to that class, and others should assign low probability.

29

Christopher Manning, Natural Language Processing with Deep Learning, Standford U. CS224n

Neural Network Feed-forward Computation

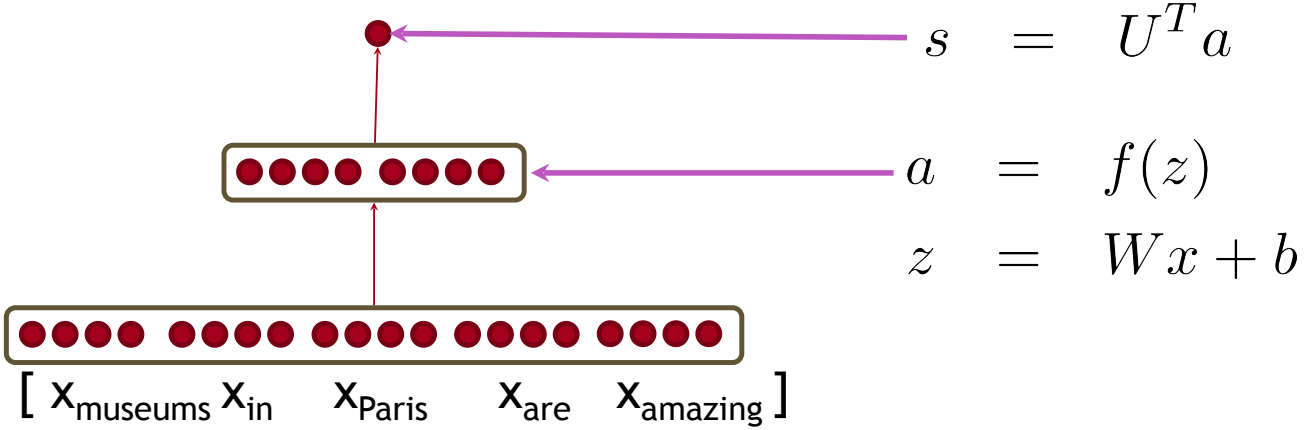
$$score(x) = U^T a \in \mathbb{R}$$

We compute a window's score with a 3-layer neural net:

- *s* = *score*("museums in Paris are amazing")

$$s = U^T f(Wx + b)$$

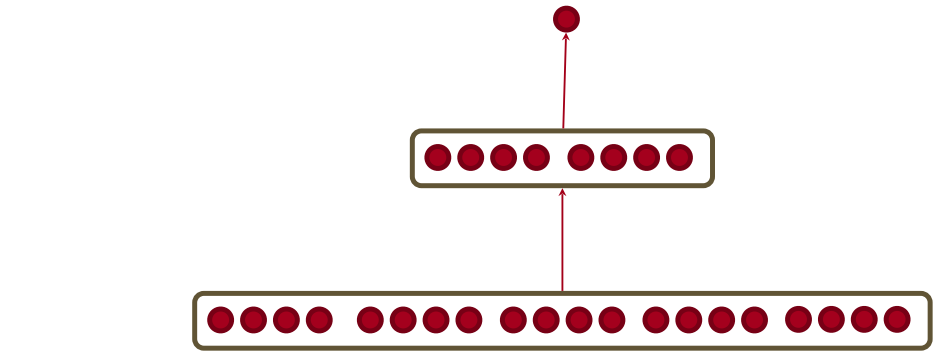
$$x \in \mathbb{R}^{20\times 1}, W \in \mathbb{R}^{8x20}, U \in \mathbb{R}^{8x2}$$



Christopher Manning, Natural Language Processing with Deep Learning, Standford U. CS224n

Main intuition for extra layer

The middle layer learns non-linear interactions between the input word vectors.



Example: only if "*museums*" is first vector should it matter that "*in*" is in the second position

31

Christopher Manning, Natural Language Processing with Deep Learning, Standford U. CS224n

Content

Weakness of word classification

- A weakness of the word classification method for sequence labeling is that it is not capable to make use of the dependencies between POS tags in prediction.
- An extreme example:

  It is true for all that that that that that that that refers to is not the same that that that that refers to.

It	is	true	for	all	that	that	that	that	that	that	that	refers	to	is	not	the	same	that	that	that	that	refers	to	.
					pron.	conj.	det.	noun	rel.pron.	det.	noun							noun	rel.pron.	det.	noun			
							(adj.)	"that"	which	(adj.)	"that"							"that"	which	(adj.)	"that"			

  - The word windows for some of the occurrences of “that” in this sentence are the same, if the window size is not greater than 2.
  - The word window classification method will not be able to distinguish those “that”.
  - The POS tags of the previous words may be helpful!

Sequence Labeling as Classification Using Outputs as Inputs

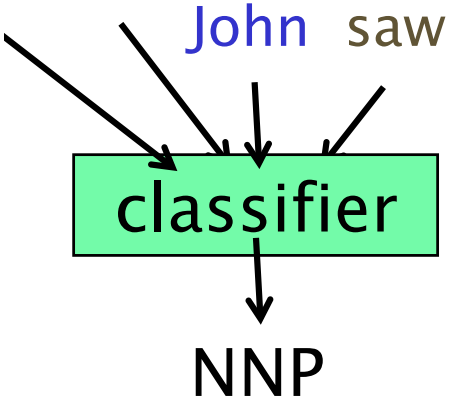
- Better input features are usually the categories of the surrounding tokens, but these are not available yet.
- Can use category of either the preceding or succeeding tokens by going forward or back and using previous output.

Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

Forward Classification

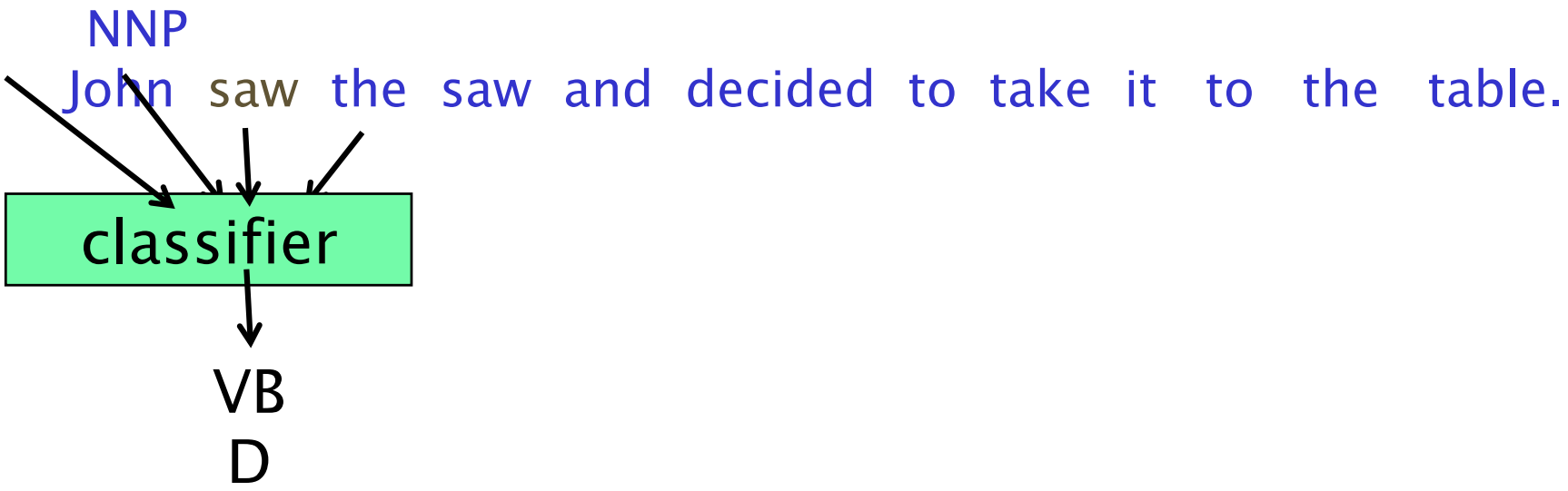
John saw the saw and decided to take it to the table.



Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

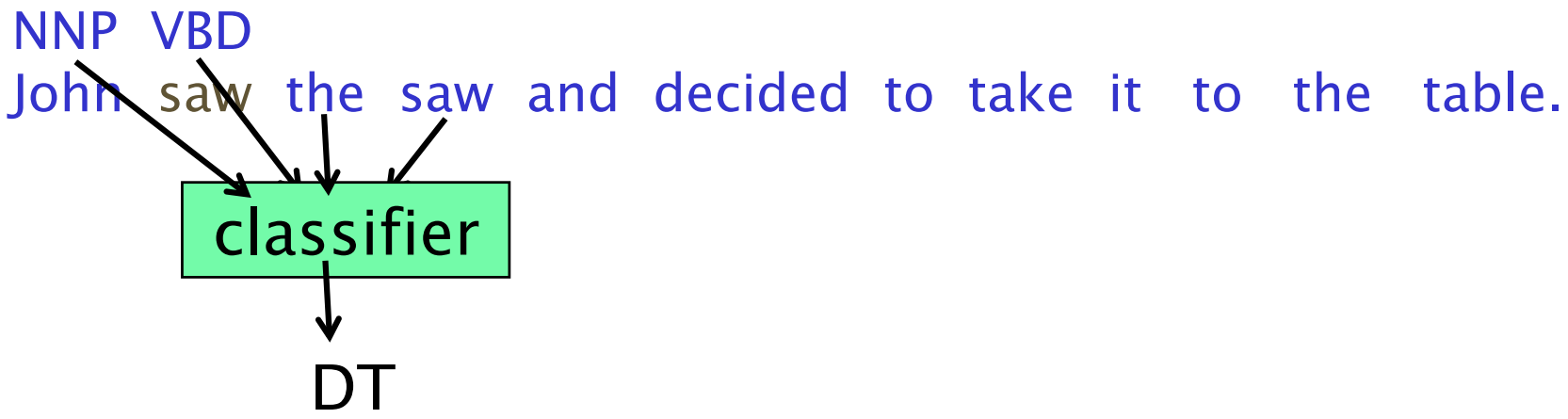
Forward Classification



Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

Forward Classification

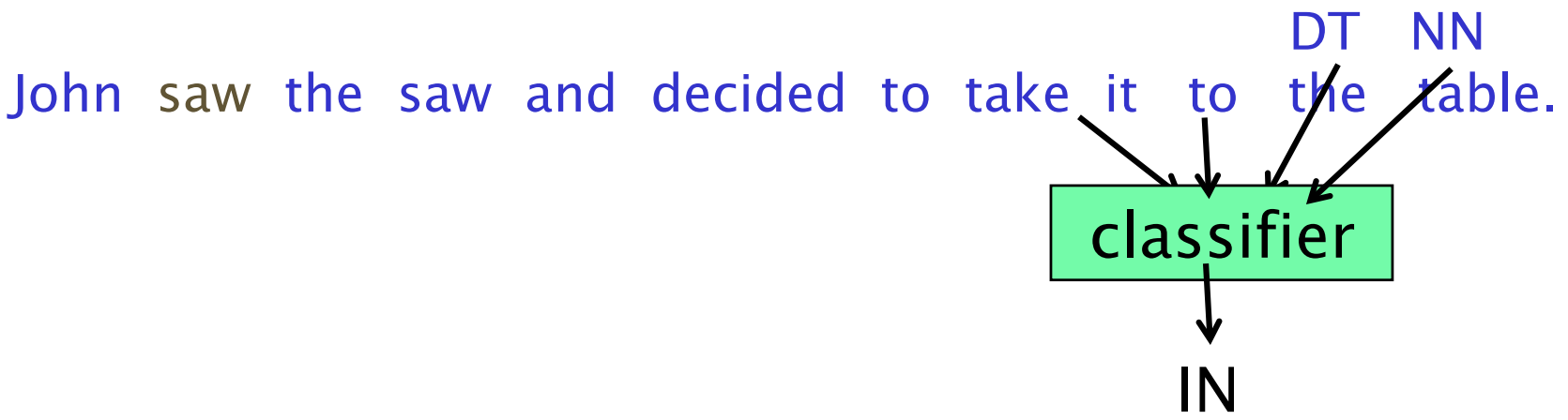


Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

Backward Classification

- Disambiguating “to” in this case would be even easier backward.

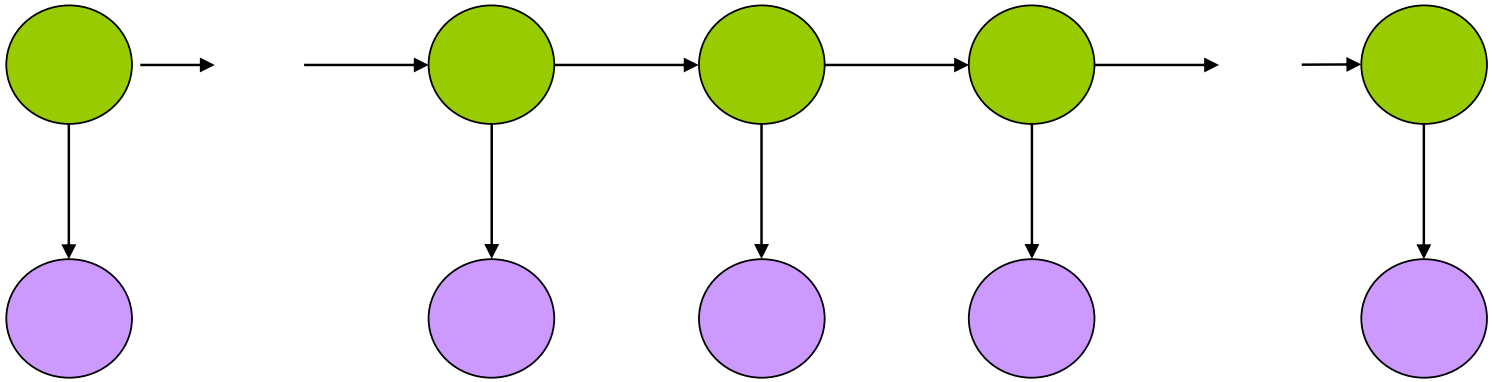


Slide from Ray Mooney

Daniel Jurafsky and James H. Martin, Part-of-speech tagging (slides)

Content

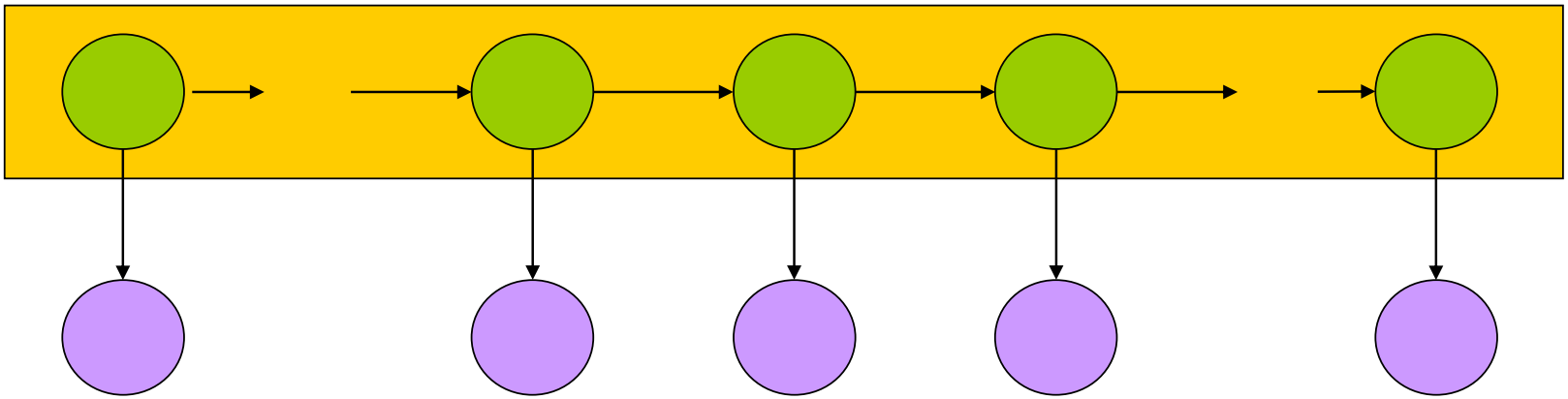
What is an HMM?



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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

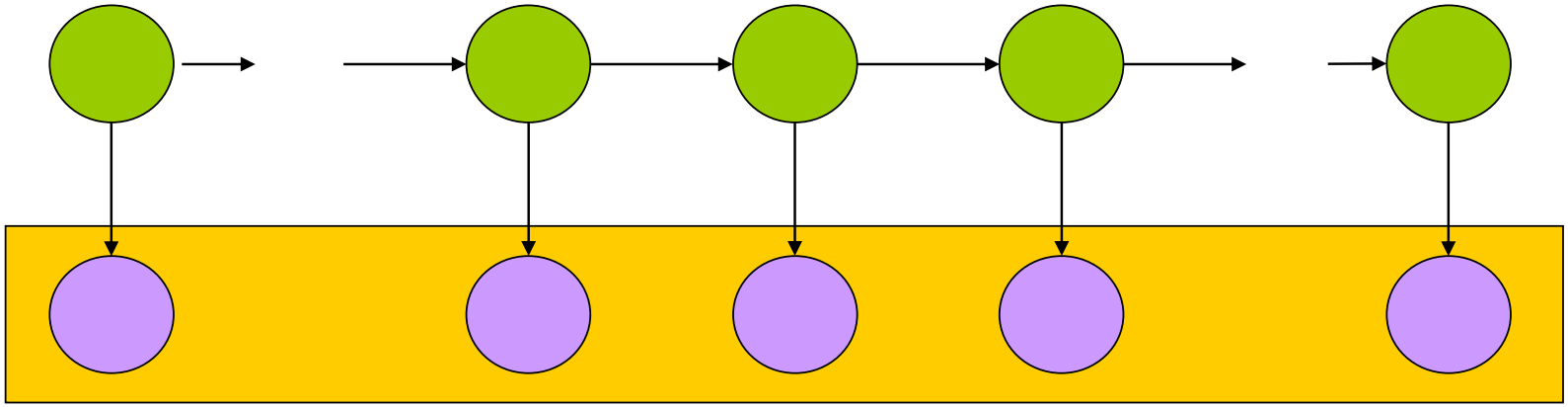
What is an HMM?



- Green circles are ***hidden states***
- Dependent only on the previous state
- "The past is independent of the future given the present."

David Meir Blei, Hidden Markov Models, 1999 (Slides)

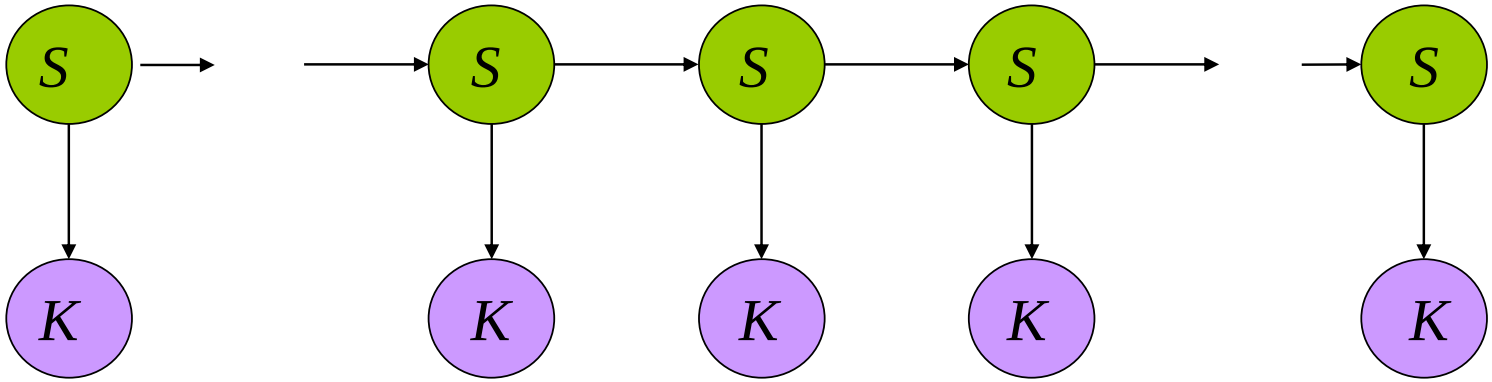
What is an HMM?



- Purple nodes are ***observed states***
- Dependent only on their corresponding hidden state

David Meir Blei, Hidden Markov Models, 1999 (Slides)

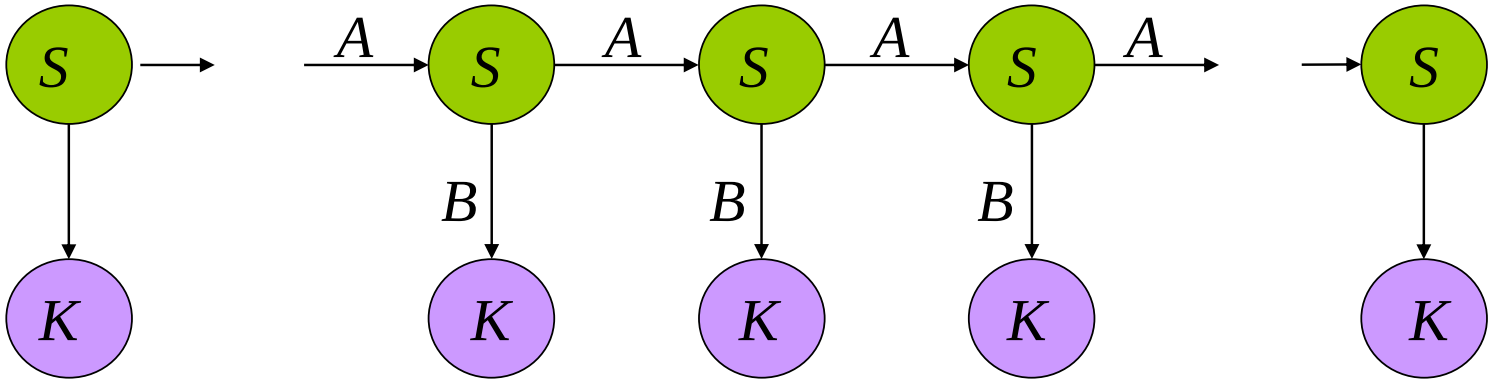
HMM Formalism



- $\{S, K, \Pi, A, B\}$
- $S : \{s_1 \ldots s_N\}$ are the values for the hidden states
- $K : \{k_1 \ldots k_M\}$ are the values for the observations

David Meir Blei, Hidden Markov Models, 1999 (Slides)

HMM Formalism

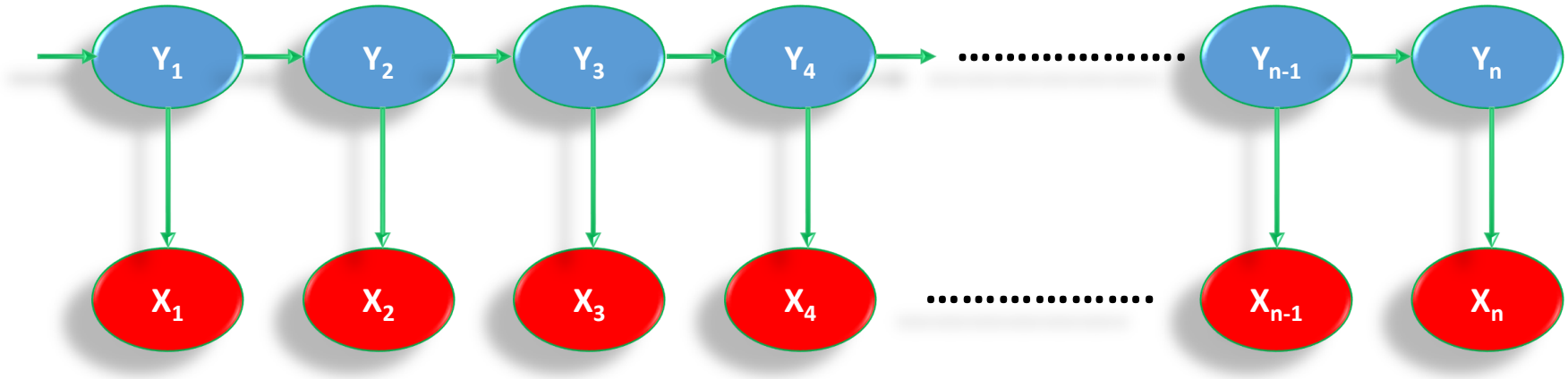


- $\{S, K, \Pi, A, B\}$
- $\Pi = \{\pi_\iota\}$ are the initial state probabilities
- $A = \{a_{ij}\}$ are the state transition probabilities
- $B = \{b_{ik}\}$ are the observation state probabilities

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Trellis Diagram

- An HMM can be graphically depicted by Trellis diagram



Anantharaman Narayana Iyer, Natural Language Processing, Unit 2 – Tagging Problems and HMM (Slides)

Another example: Coin tossing

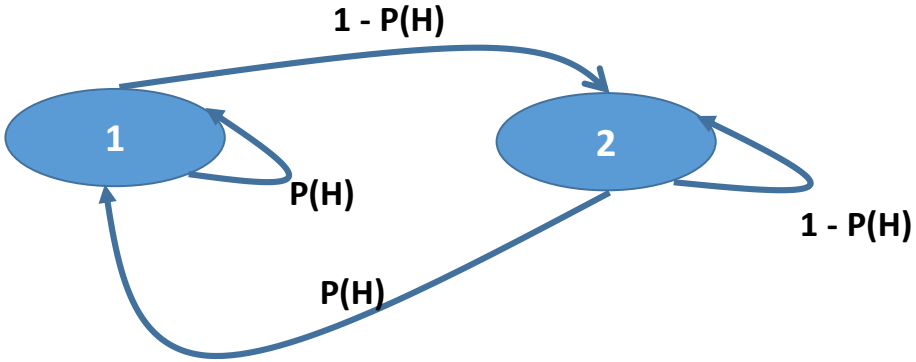


Fig (a)

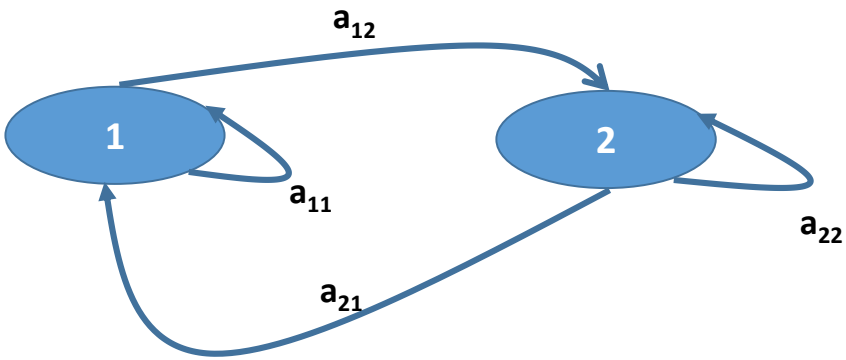


Fig (b)

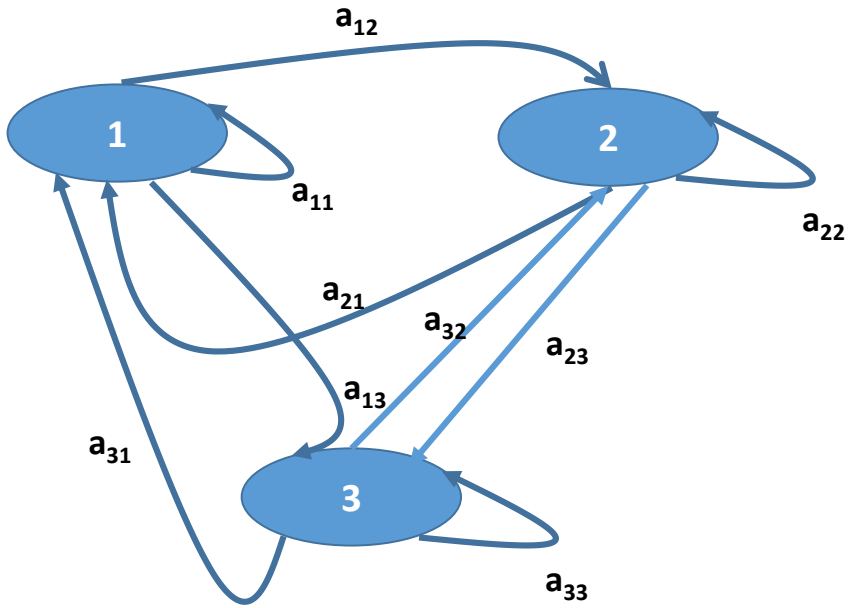


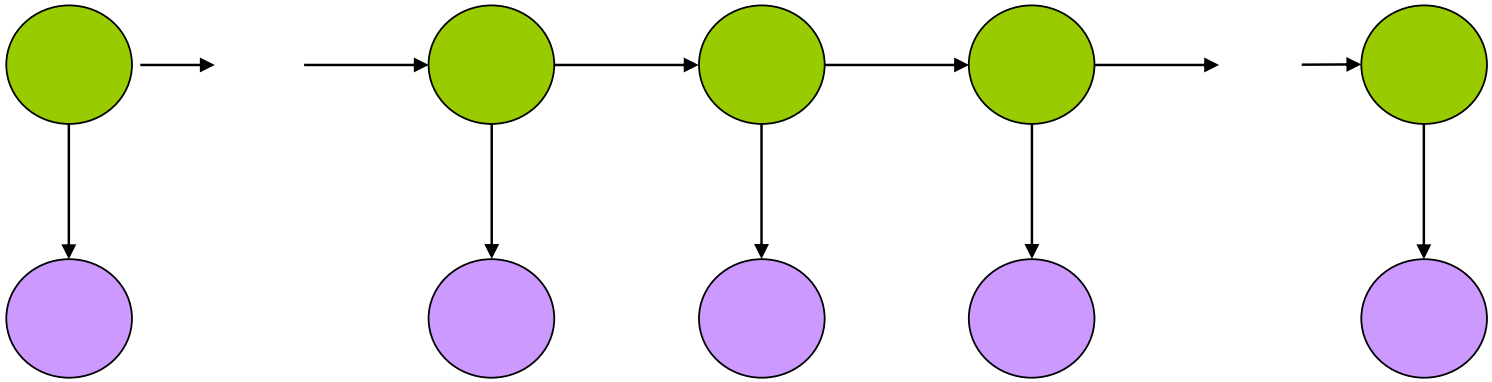
Fig (c)

	State 1	State 2	State 3
P(H)	0.5	0.75	0.25
P(T)	0.5	0.25	0.75

Anantharaman Narayana Iyer, Natural Language Processing, Unit 2 – Tagging Problems and HMM (Slides)

Content

Inference in an HMM

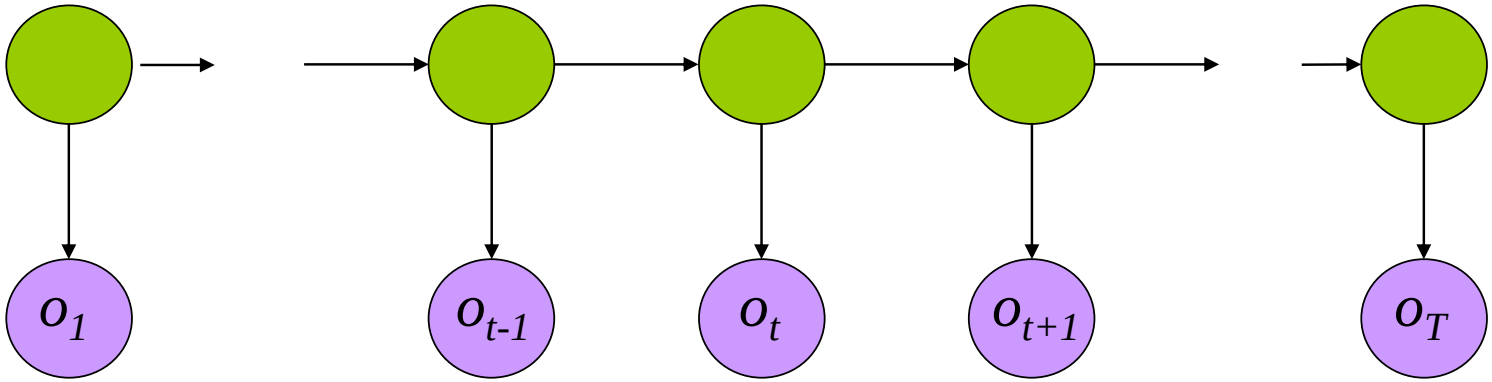


- Compute the probability of a given observation sequence
- Given an observation sequence, compute the most likely hidden state sequence
- Given an observation sequence and set of possible models, which model most closely fits the data?

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Content

Decoding



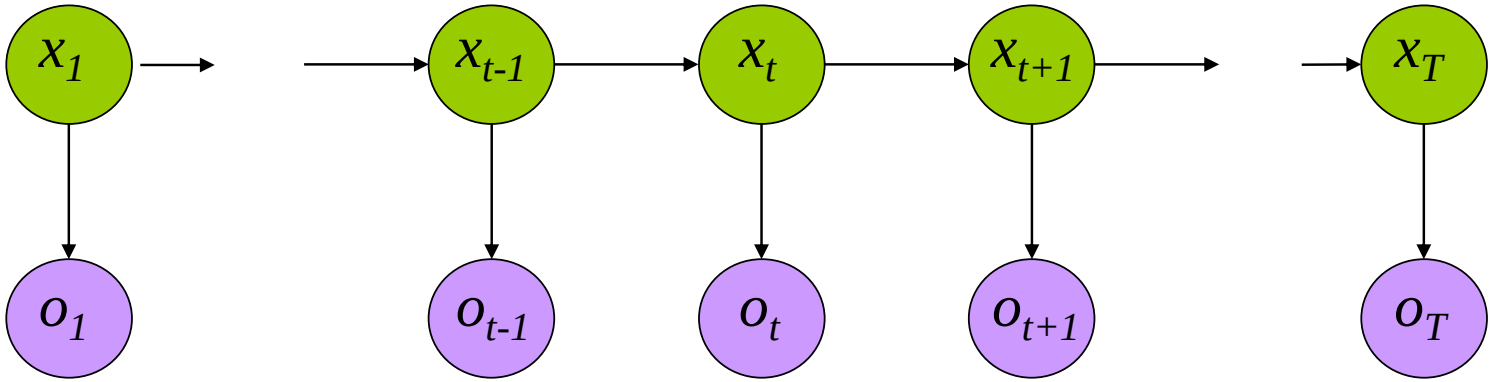
Given an observation sequence and a model, compute the probability of the observation sequence

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

Compute $P(O \mid \mu)$

David Meir Blei, Hidden Markov Models, 1999 (Slides)

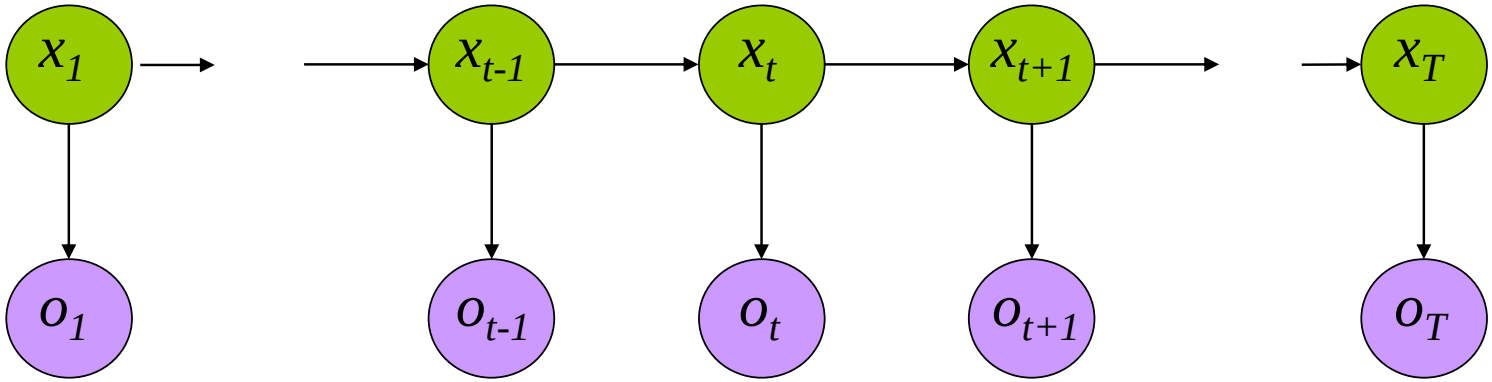
Decoding



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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Decoding

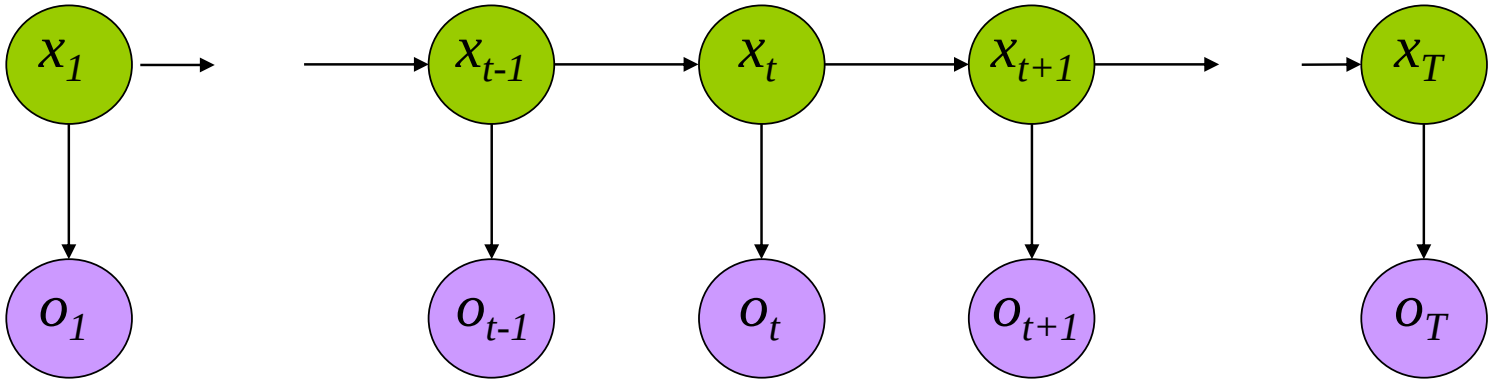


$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Decoding



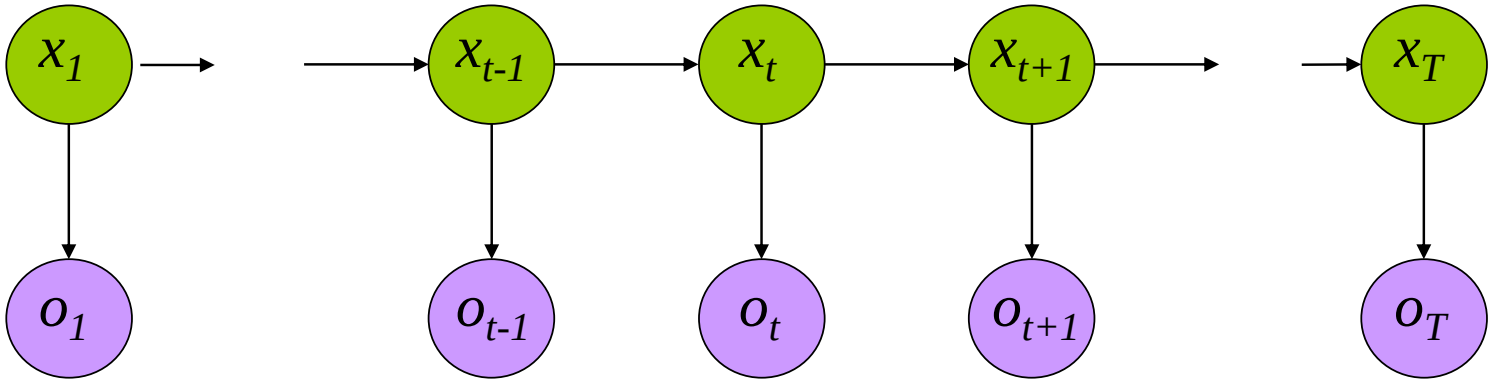
$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

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

$$\boxed{P(O, X \mid \mu) = P(O \mid X, \mu) P(X \mid \mu)}$$

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Decoding



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

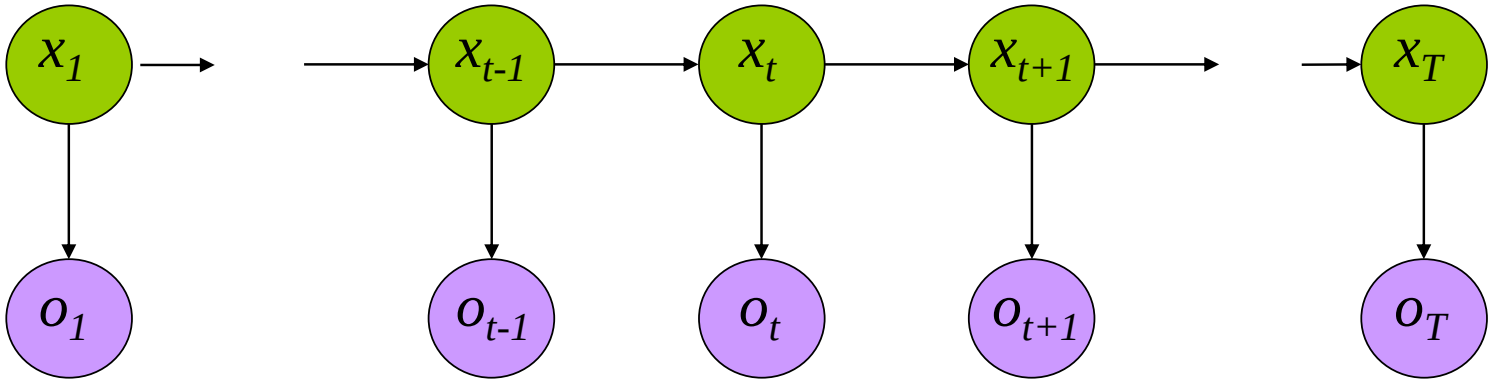
$$P(X \mid \mu) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} \ldots a_{x_{T-1} x_T}$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Decoding

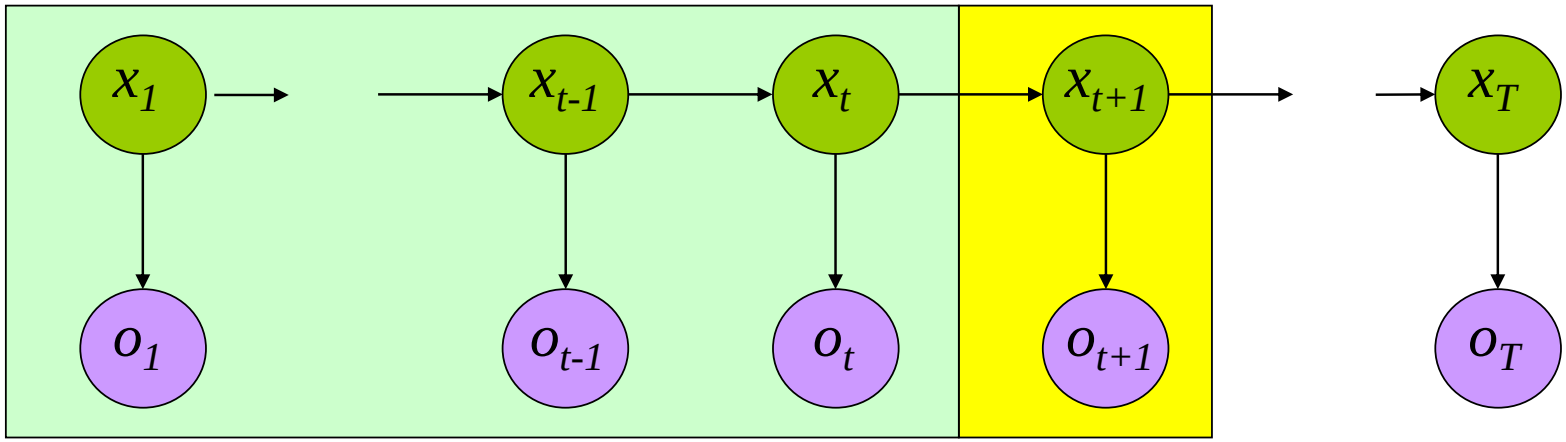


$$P(O \mid \mu) = \sum_{\{x_1 \ldots x_T\}} \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Content

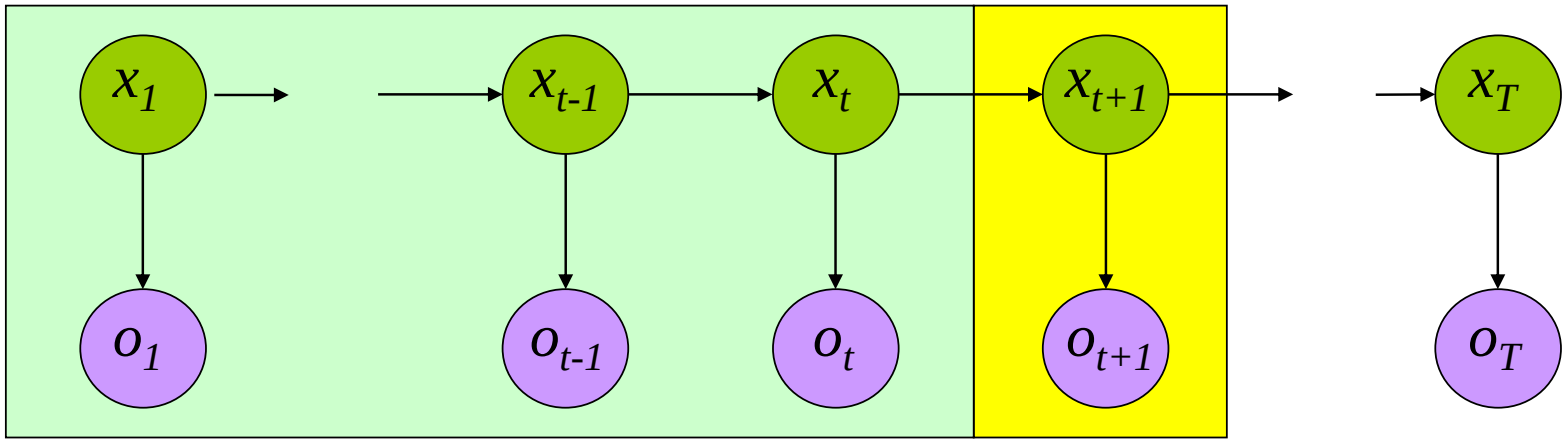
Forward Procedure



- Special structure gives us an efficient solution using *dynamic programming*.
- **Intuition**: Probability of the first t observations is the same for all possible t+1 length state sequences.
- **Define:** $\alpha_i(t) = P(o_1 \ldots o_t, x_t = i \mid \mu)$

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



$$\alpha_j(t+1)$$

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

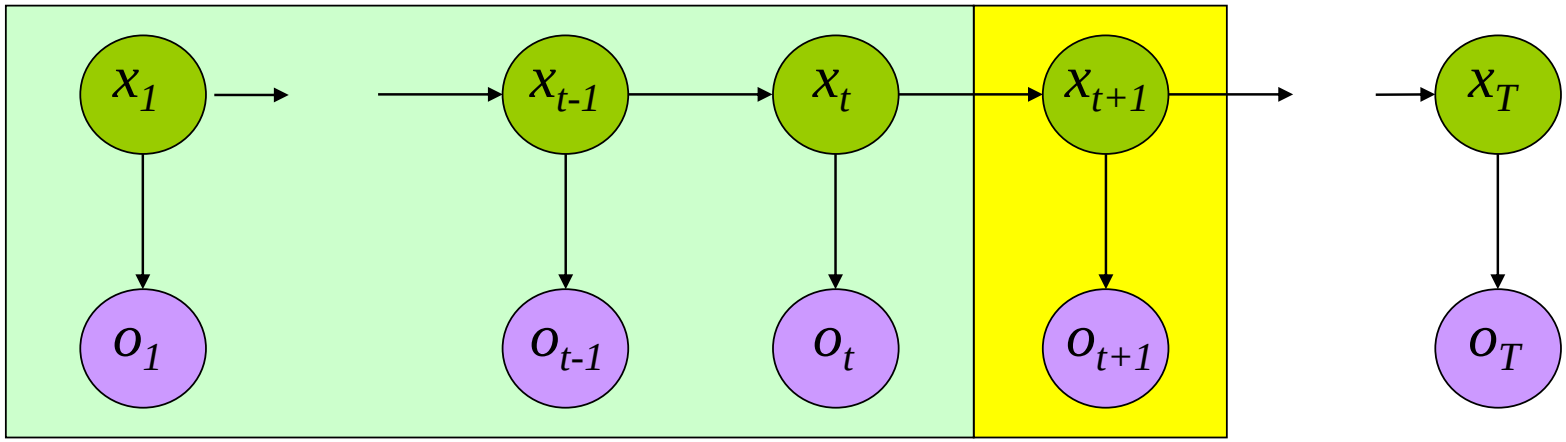
$$= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



$$\alpha_j(t+1)$$

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

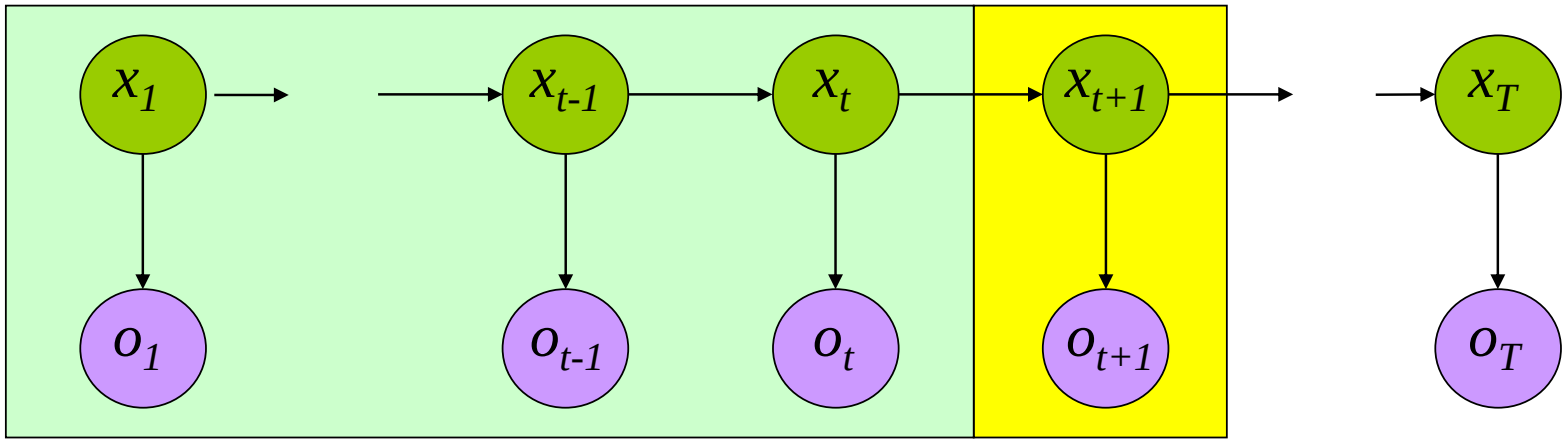
$$= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



$$\alpha_j(t+1)$$

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

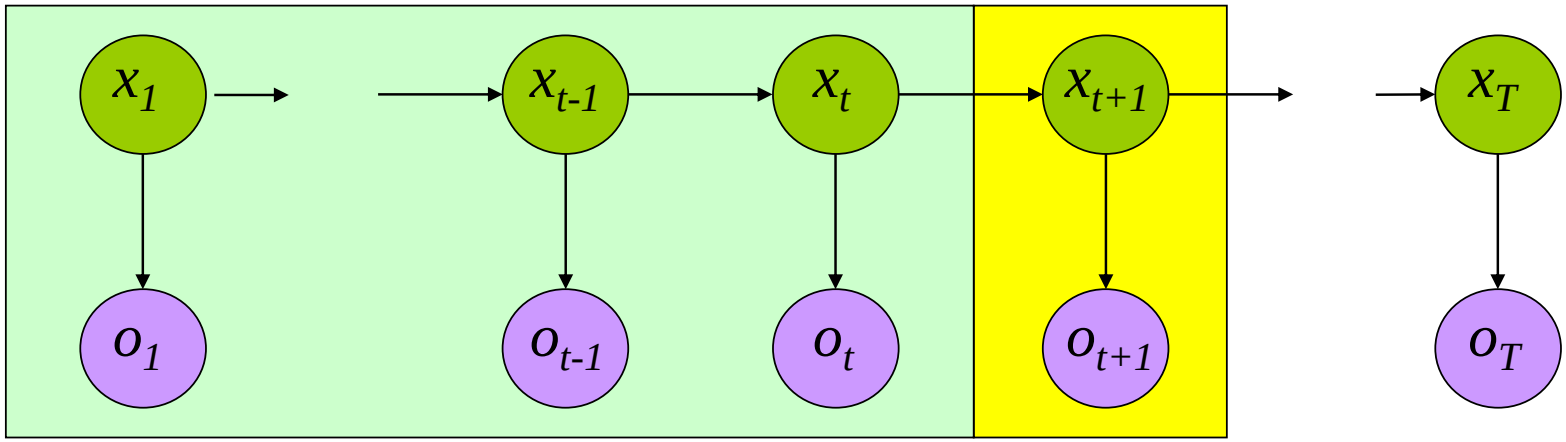
$$= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



$$\alpha_j(t+1)$$

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

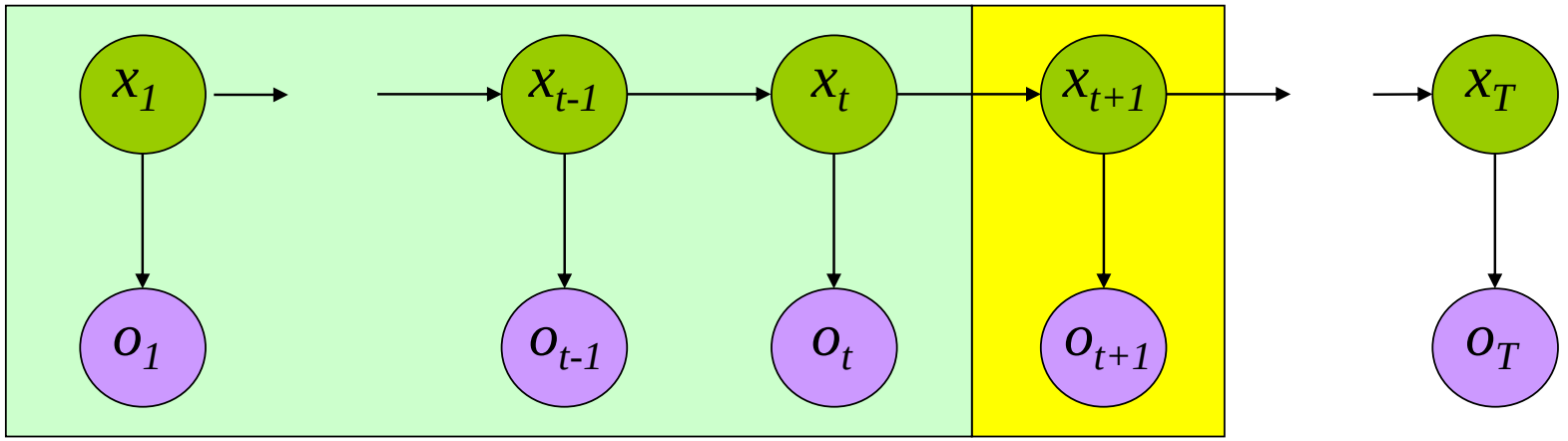
$$= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



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

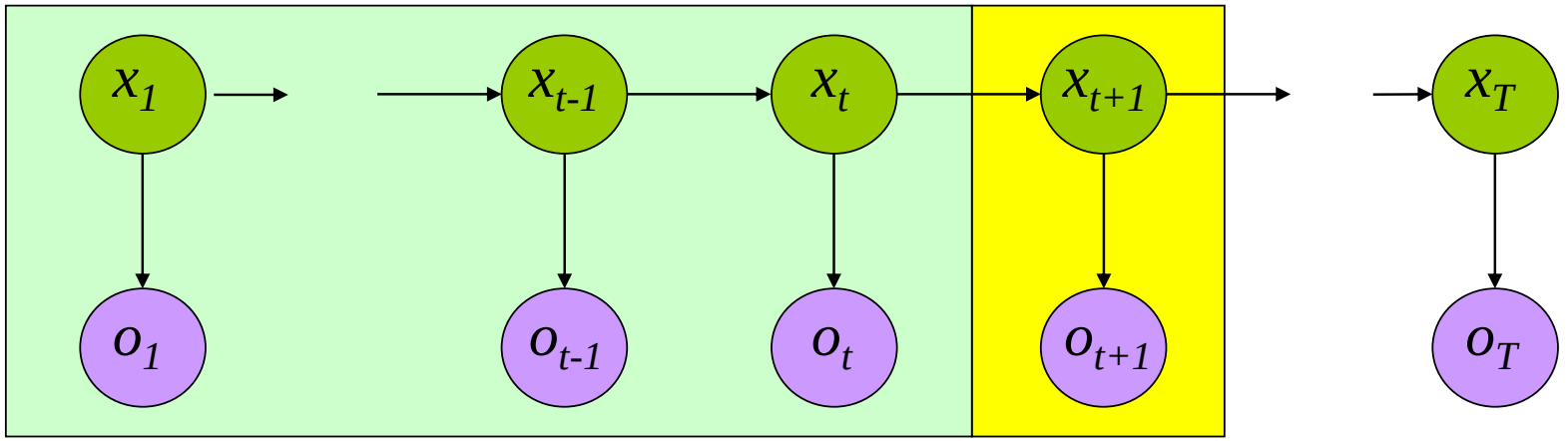
$$= \sum_{i=1...N} P(o_1 ... o_t, x_{t+1} = j \mid x_t = i) P(x_t = i) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



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

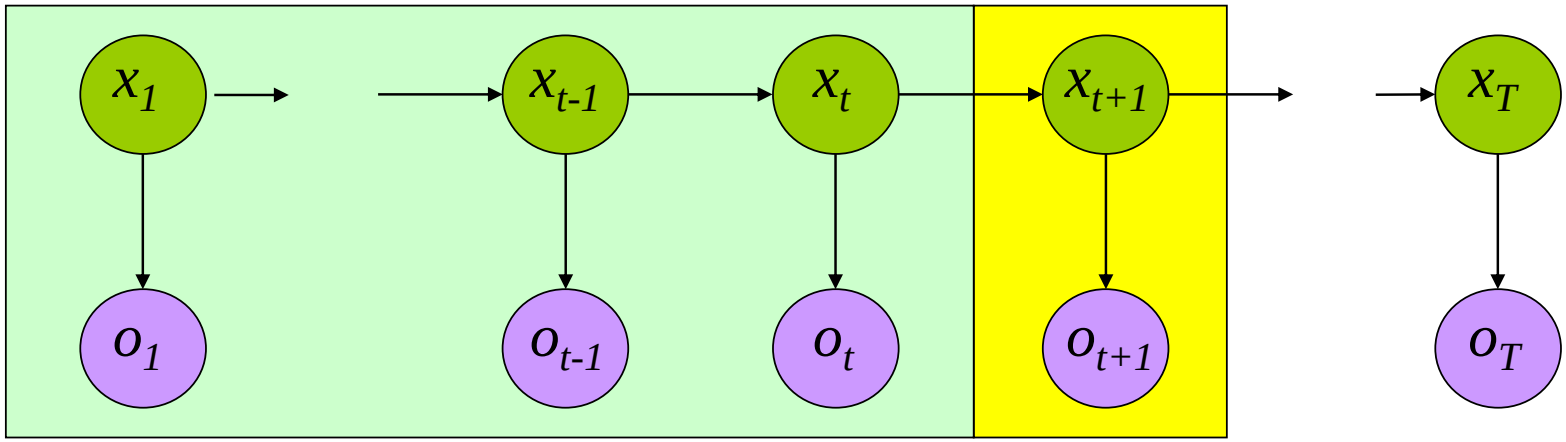
$$= \sum_{i=1...N} P(o_1 ... o_t, x_{t+1} = j \mid x_t = i) P(x_t = i) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



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

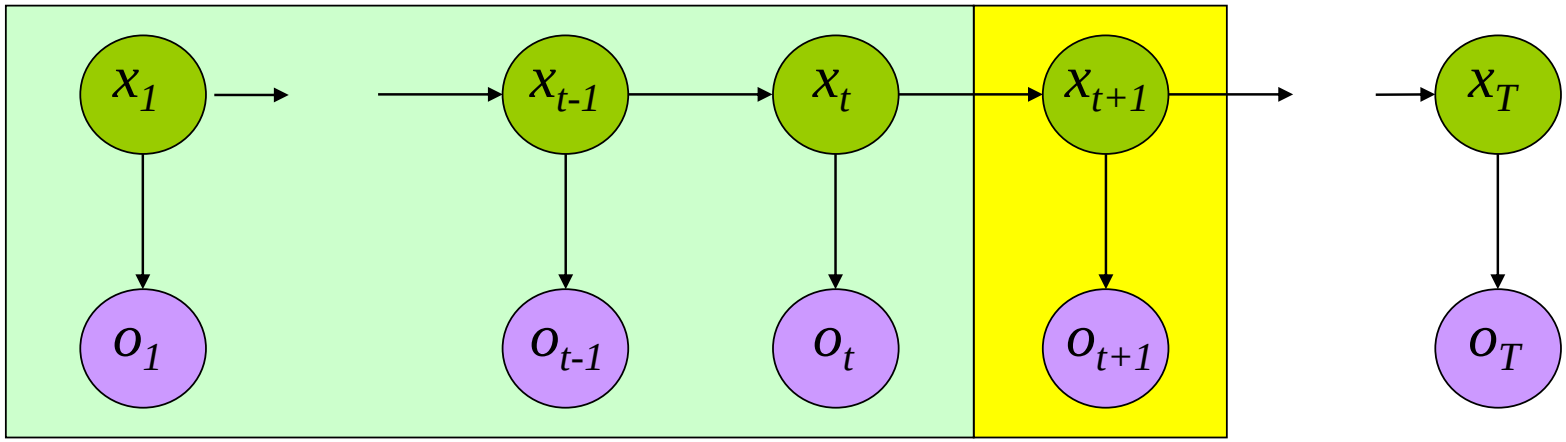
$$= \sum_{i=1...N} P(o_1 ... o_t, x_{t+1} = j \mid x_t = i) P(x_t = i) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)

Forward Procedure



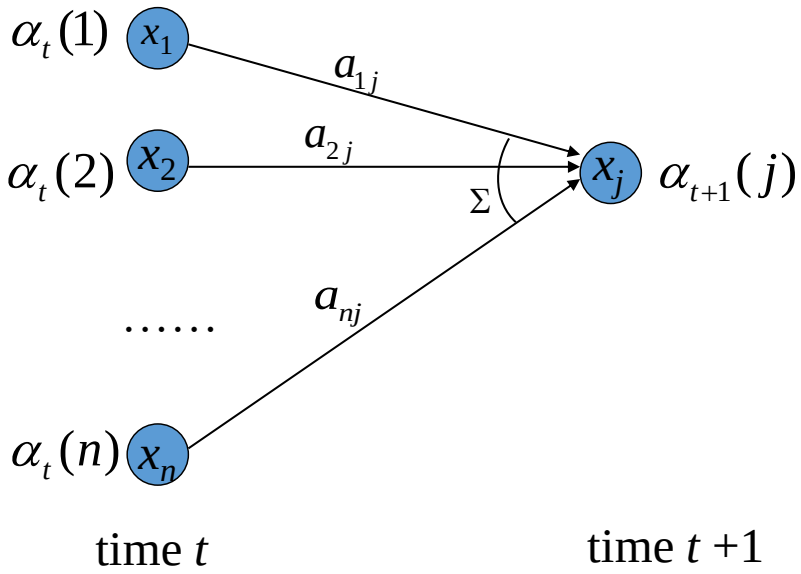
$$= \sum_{i=1...N} P(o_1 ... o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

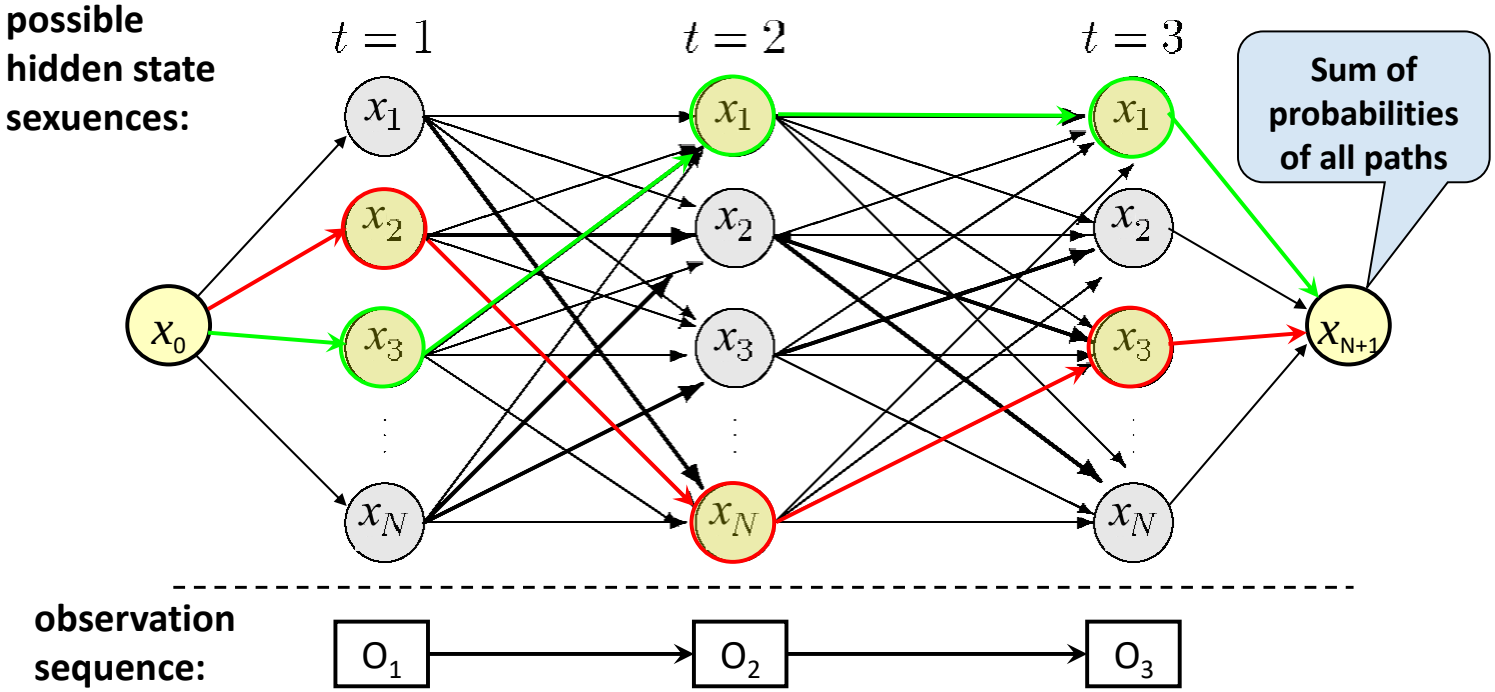
$$= \sum_{i=1...N} P(o_1 ... o_t, x_{t+1} = j \mid x_t = i) P(x_t = i) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

David Meir Blei, Hidden Markov Models, 1999 (Slides)


$\alpha_t(1)$
x_1
a_{1j}
$\alpha_t(2)$
x_2
a_{2j}
x_j
$\alpha_{t+1}(j)$
Σ
.....
a_{nj}
$\alpha_t(n)$
x_n
time t
time t +1



# of possible hidden state sequences: N^T

Time complexity of Forward algorithm: O(N2T)

Forward Algorithm Summary

1. $\alpha_1(i) = \pi_i b_i(O_1),\ 1 \le i \le N$

2. for $t = 1, 2, \ldots, T-1,\ 1 \le j \le N$
$\alpha_{t+1}(j) = [\Sigma_{i=1 \text{ to } N}\ \alpha_t(i) * a_{ij}] * b_j(O_{t+1})$

3. Finally we have:
$P(O|\ \lambda) = \Sigma_{i=1 \text{ to } N}\ \alpha_T(i)$

Multiplications

$= N$

$= (N+1)N(T-1)$

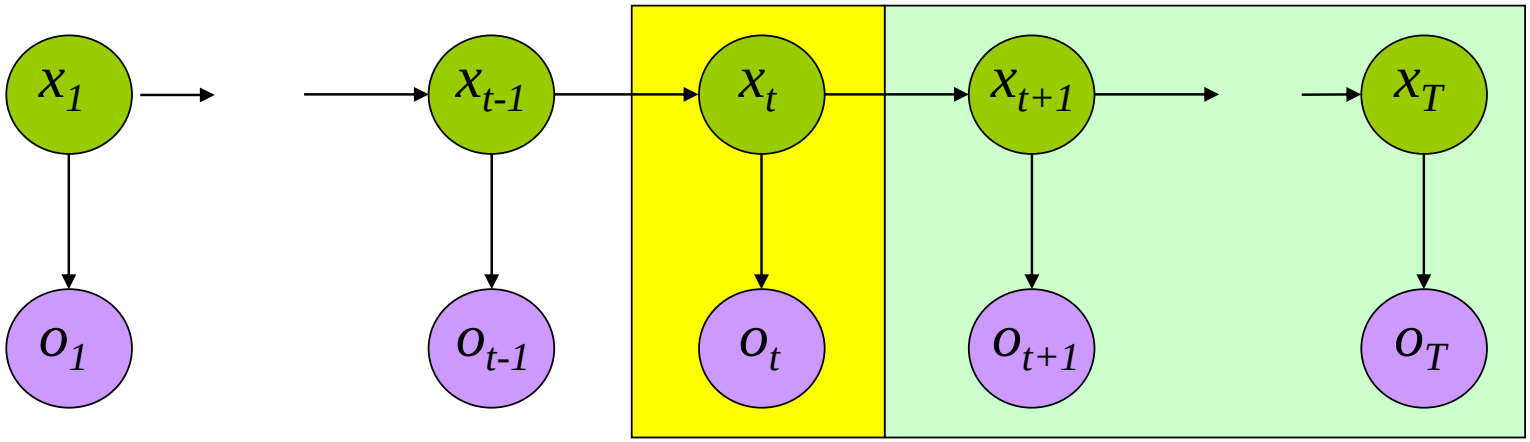
$= 0$

Total:
$N+(N+1)N(T-1)$

Anantharaman Narayana Iyer, Natural Language Processing, Unit 2 – Tagging Problems and HMM (Slides)

Content

Backward Procedure



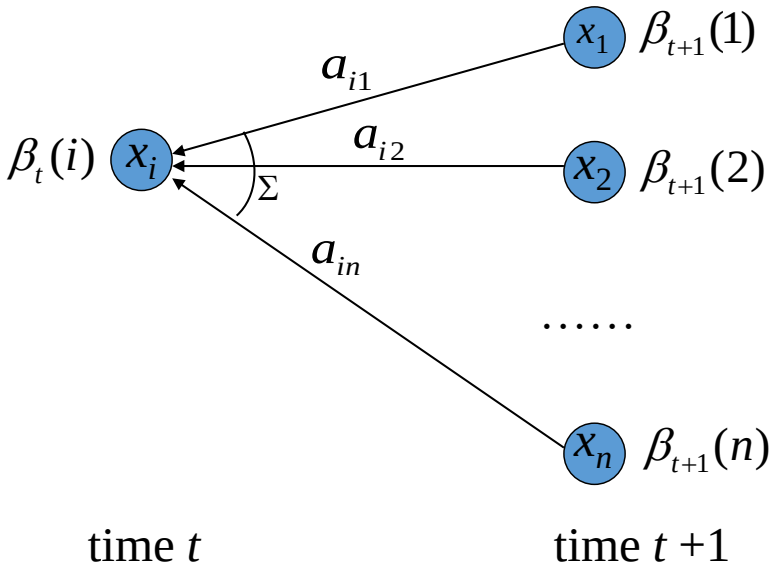
$$\beta_i(T+1) = 1$$

$$\beta_i(t) = P(o_t \ldots o_T \mid x_t = i)$$

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

Probability of the rest of the states given the first state

David Meir Blei, Hidden Markov Models, 1999 (Slides)


x_1
$\beta_{t+1}(1)$
a_{i1}
$\beta_t(i)$
x_i
a_{i2}
x_2
$\beta_{t+1}(2)$
Σ
a_{in}
.....
x_n
$\beta_{t+1}(n)$
time t
time t +1

Backward Algorithm: Summary

1. $\beta_T(i) = 1, 1 \leq i \leq N$

Multiplications

= 0

2. for t = T-1, T-2, ..., 1, $1 \leq i \leq N$

$\beta_t(i) = \Sigma_{j=1 \text{ to } N} a_{ij} * b_j(O_{t+1}) * \beta_{t+1}(j)$

= (2N)N(T-1)

3. Finally we have:

$P(O|\lambda) = \Sigma_{i=1 \text{ to } N} \pi_i * b_i(O_1) * \beta_1(i)$

= 2N

Total:
2N+(2N)N(T-1)

Anantharaman Narayana Iyer, Natural Language Processing, Unit 2 – Tagging Problems and HMM (Slides)