CS 6840: Natural Language Processing

Sequence Tagging with HMMs:
Part of Speech Tagging

Razvan C. Bunescu

School of Electrical Engineering and Computer Science

bunescu@ohio.edu

Part of Speech (POS) Tagging

- Annotate each word in a sentence with its POS:
  - noun, verb, adjective, adverb, pronoun, preposition, interjection, …

			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

Parts of Speech

- Lexical categories that are defined based on:
  - **Syntactic function**:
    - nouns can occur with determiners: a goat.
    - nouns can take possessives: IBM's annual revenue.
    - most nouns can occur in the plural: goats.
  - **Morphological function**:
    - many verbs can be composed with the prefix "un".

- There are tendencies toward **semantic coherence**:
  - nouns often refer to "people, places, or things".
  - adjectives often refer to properties.

POS: Closed Class vs. Open Class

- **Closed Class**:
  - relatively fixed membership.
  - usually function words:
    - short common words which have a structuring role in grammar.
  - **Prepositions**: of, in, by, on, under, over, …
  - **Auxiliaries**: may, can, will had, been, should, …
  - **Pronouns**: I, you, she, mine, his, them, …
  - **Determiners**: a, an, the, which, that, …
  - **Conjunctions**: and, but, or (coord.), as, if, when, (subord.), …
  - **Particles**: up, down, on, off, …
  - **Numerals**: one, two, three, third, …

POS: Open Class vs. Closed Class

- **Open Class**:
  - new members are continually added.
    - *to fax, to google, futon*, …
  - English has 4: Nouns, Verbs, Adjectives, Adverbs.
    - Many languages have these 4, but not all (e.g. Korean).
  - **Nouns**: people, places, or things
  - **Verbs**: actions and processes
  - **Adjectives**: properties or qualities
  - **Adverbs**: a hodge-podge
    - *Unfortunately*, John walked *home extremely slowly yesterday*.
    - directional, locative, temporal, degree, manner, …

POS: Open vs. Closed Classes

- **Open Class**: new members are continually added.

1. Annie: Do you love me?

   Alvy: Love is too weak a word for what I feel... I **lurve** you. Y'know, I **loove** you, I, I **luff** you. There are two f's. I have to invent... Of course I love you. (*Annie Hall*)

2. 'Twas brillig, and the slithy toves
   Did gyre and gimble in the wabe;
   All mimsy were the borogoves,
   And the mome raths outgrabe.

   "Beware the Jabberwock, my son!
   The jaws that bite, the claws that catch!
   Beware the Jubjub bird, and shun
   The frumious Bandersnatch!"

   (*Jabberwocky, Lewis Caroll*)

Parts of Speech: Granularity

- Grammatical sketch of Greek [Dionysius Thrax, c. 100 B.C.]:
  - 8 tags: *noun, verb, pronoun, preposition, adjective, conjunction, participle*, and *article*.
- Brown corpus [Francis, 1979]:
  - 87 tags.
- Penn Treebank [Marcus et al., 1993]:
  - 45 tags.
- British National Corpus (BNC) [Garside et al., 1997]:
  - C5 tagset: 61 tags.
  - C7 tagset: 146 tags.

We will focus on the Penn Treebank POS tags.

Penn Treebank POS Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Penn Treebank POS tags

- Selected from the original 87 tags of the Brown corpus:
  - ⇒ lost finer distinctions between lexical categories.

1) Prepositions and subordinating conjunctions:
   - **after/CS** spending/VBG a/AT day/NN at/IN the/AT palace/NN
   - **after/IN** a/AT wedding/NN trip/NN to/IN Hawaii/NNP ./.

2) Infinitive to and prepositional to:
   - **to/TO** give/VB priority/NN **to/IN** teachers/NNS

3) Adverbial nouns:
   - Brown: Monday/NR, home/NR, west/NR, tomorrow/NR
   - PTB: Monday/NNP, (home, tomorrow, west)/(NN, RB)

POS Tagging ≡ POS Disambiguation

- Words often have more than one POS tag, e.g. **back**:
  - the **back/JJ** door
  - on my **back/NN**
  - win the voters **back/RB**
  - promised to **back/VB** the bill

- Brown corpus statistics [DeRose, 1998]:
  - 11.5% ambiguous English word types.
  - 40% of all word occurrences are ambiguous.
    - most are easy to disambiguate
      - the tags are not equaly likely, i.e. low tag entropy: *table*

POS Tag Ambiguity

		87-tag Original Brown		45-tag Treebank Brown	
Unambiguous (1 tag)		**44,019**		**38,857**	
Ambiguous (2–7 tags)		**5,490**		**8844**	
Details:	2 tags	4,967		6,731	
	3 tags	411		1621	
	4 tags	91		357	
	5 tags	17		90	
	6 tags	2	(*well*, *beat*)	32	
	7 tags	2	(*still*, *down*)	6	(*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags			4	(*'s*, *half*, *back*, *a*)
	9 tags			3	(*that*, *more*, *in*)

POS Tagging ≡ POS Disambiguation

- Some distinctions are difficult even for humans:
  - Mrs. Shaefer never got **around** to joining
    NNP NNP RB VBD **RP** TO VBG
  - All we gotta do is go **around** the corner
    DT PRP VBN VB VBZ VB **IN** DT NN
  - Chateau Petrus costs **around** 250
    NNP NNP VBZ **RB** CD
- Use heuristics [Santorini, 1990]:
  - She told **off/RP** her friends
  - She told her friends **off/RP**

She stepped **off/IN** the train

*She stepped the train **off/IN**

How Difficult is POS Tagging?

- Most current tagging algorithms: ~ 96% – 97% accuracy for Penn Treebank tagsets.
  - Current SofA 97.55% tagging accuracy. How good is this?
    - Bidirectional LSTM-CRF Models for Sequence Tagging [Huang, Xu, Yu, 2015].
  - **Human Ceiling**: how well humans do?
    - human annotators: about 96% – 97% [Marcus et al., 1993].
    - when allowed to discuss tags, consensus is 100% [Voutilainen, 95]
  - **Most Frequent Class Baseline**:
    - 90% – 91% on the 87-tag Brown tagset [Charniak et al., 1993].
    - 93.69% on the 45-tag Penn Treebank, with unknown word model [Toutanova et al., 2003].

POS Tagging Methods

- **Rule Based**:
  - Rules are designed by human experts based on linguistic knowledge.
- **Machine Learning**:
  - Trained on data that has been manually labeled by humans.
  - Rule learning:
    - **Transformation Based Learning (TBL)**.
  - **Sequence tagging**:
    - **Hidden Markov Models (HMM).**
    - **Maximum Entropy (Logistic Regression).**
    - **Sequential Conditional Random Fields (CRF)**.
    - Recurrent Neural Networks (RNN):
      - bidirectional, with a CRF layer (BI-LSTM-CRF).

POS Tagging: Rule Based

1) Start with a dictionary.

2) Assign all possible tags to words from the dictionary.

3) Write rules by hand to selectively remove tags, leaving the correct tag for each word.

POS Tagging: Rule Based

1) Start with a dictionary:

she:	PRP
promised:	VBN,VBD
to	TO
back:	VB, JJ, RB, NN
the:	DT
bill:	NN, VB

… for the ~100,000 words of English.

POS Tagging: Rule Based

2) Assign every possible tag:

			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

POS Tagging: Rule Based

3) Write rules to eliminate incorrect tags.
   – Eliminate VBN if VBD is an option when VBN|VBD follows "<S> PRP"

		NN			
		RB			
	VBN	JJ		VB	
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

POS Tagging as Sequence Labeling

- **Sequence Labeling**:
  - Tokenization and Sentence Segmentation.
  - Part of Speech Tagging.
  - Information Extraction
    - Named Entity Recognition
  - Shallow Parsing.
  - Semantic Role Labeling.
  - DNA Analysis.
  - Music Segmentation.

- Solved using **ML models** for classification:
  - Token-level vs. Sequence-level.

Sequence Labeling

- **Sentence Segmentation**:

Mr. Burns is Homer Simpson's boss. He is very rich.

O O O O O O … … … … O O **I** O O … O **I**

- **Tokenization**:

Mr. Burns is Homer Simpson's boss. He is very rich.

Sequence Labeling

- **Information Extraction**:
  - **Named Entity Recognition**

O	O	I	I	O	O	O	O	O	O	O
Drug	giant	**Pfizer**	**Inc.**	has	reached	an	agreement	to	buy	the

O	O	O	I	I	I
private	biotechnology	firm	**Rinat**	**Neuroscience**	**Corp.**

Sequence Labeling

- **Information Extraction**:
  - **Text Segmentation** into topical sections.

Vine covered cottage , near Contra Costa Hills . 2 bedroom house ,

modern kitchen and dishwasher . No pets allowed . $ 1050 / month

[Haghighi & Klein, NAACL '06]

Sequence Labeling

- **Information Extraction**:
  - segmenting classifieds into topical sections.

Vine covered cottage , near Contra Costa Hills . 2 bedroom house ,

modern kitchen and dishwasher . No pets allowed . $ 1050 / month

[Haghighi & Klein, NAACL '06]

  - Features
  - Neighborhood
  - Size
  - Restrictions
  - Rent

Sequence Labeling

- **Semantic Role Labeling**:
  - For each clause, determine the semantic role played by each noun phrase that is an argument to the verb:

    John drove Mary from Athens to Columbus in his Toyota Prius.

    The hammer broke the window.

    - agent
    - patient
    - source
    - destination
    - instrument

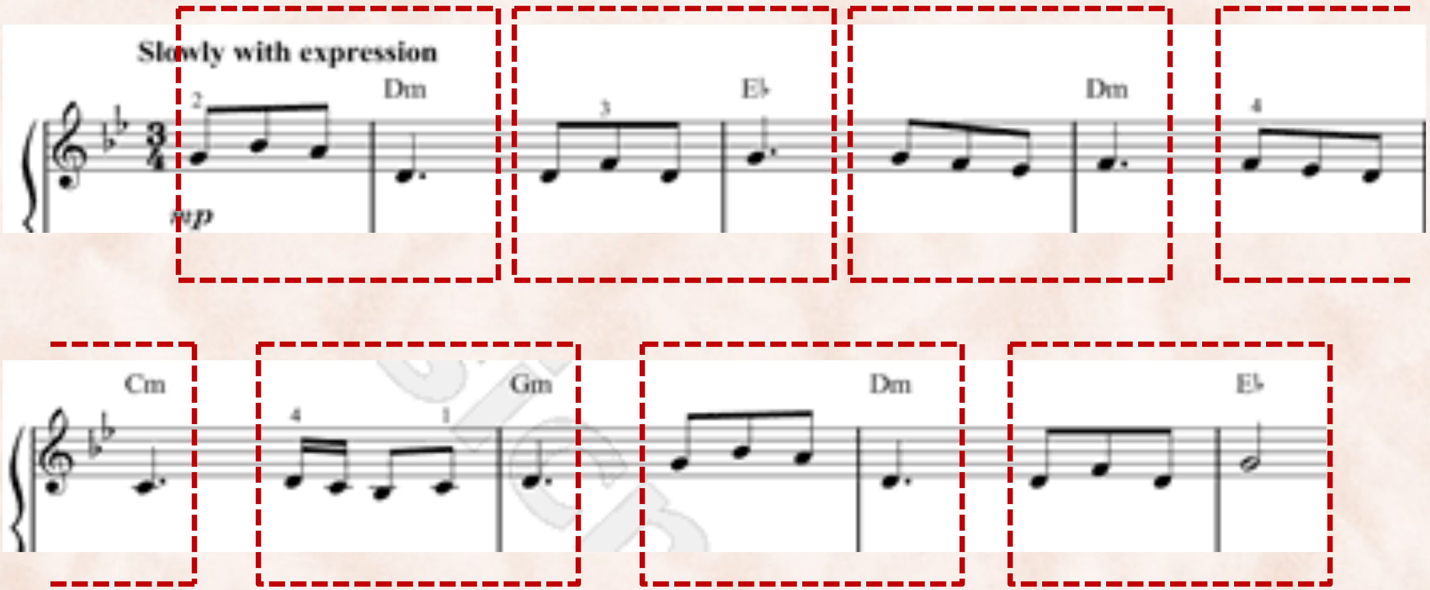
Sequence Labeling

- **DNA Analysis**:
  - transcription factor binding sites.
  - promoters.
  - introns, exons, …

AATGCGCTAACGTTCGATACGAGATAGCCTAAGAGTCA

Sequence Labeling

- **Music Analysis**:
  - segmentation into "musical phrases"



[*Romeo & Juliet*, Nino Rota]

Sequence Labeling as Classification

1) **Classifiy** each token **individually** into one of a number of classes:
   - Token represented as a vector of features extracted from context.
   - To build classification model, use general ML algorithms:
     - **Maximum Entropy** (i.e. **Logistic Regression**)
     - **Support Vector Machines (SVMs)**
     - **Perceptrons**.
     - Winnow.
     - Naïve Bayes, Bayesian Networks.
     - Decision Trees.
     - k-Nearest Neighbor, …

A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Represent each position i in text as $\varphi(t, h_i) = \{\varphi_k(t, h_i)\}$:
  - t is the potential POS tag at position i.
  - h_i is the history/context of position i.

$$h_i = \{w_i, w_{i+1}, w_{i+2}, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}\}$$

  - $\varphi(t, h_i)$ is a vector of features $\varphi_k(t, h_i)$, for $k = 1..K$.

$$\phi_k(t, h_i) = \begin{cases} 1 & \text{if suffix}(w_i) = \text{"ing"} \ \& \ t = \text{VBG} \\ 0 & \text{otherwise} \end{cases}$$

- Represent the "unnormalized" score of a tag t as:

$$score(t, h_i) = \mathrm{w}^{\mathrm{T}} \phi(t, h_i) = \sum_{k=1}^{K} w_k \phi_k(t, h_i)$$

want w_k to be large here

A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

Condition	Features	
w_i is not rare	$w_i = X$	& $t_i = T$
w_i is rare	X is prefix of w_i, $\|X\| \leq 4$	& $t_i = T$
	X is suffix of w_i, $\|X\| \leq 4$	& $t_i = T$
	w_i contains number	& $t_i = T$
	w_i contains uppercase character	& $t_i = T$
	w_i contains hyphen	& $t_i = T$
$\forall\, w_i$	$t_{i-1} = X$	& $t_i = T$
	$t_{i-2}t_{i-1} = XY$	& $t_i = T$
	$w_{i-1} = X$	& $t_i = T$
	$w_{i-2} = X$	& $t_i = T$
	$w_{i+1} = X$	& $t_i = T$
	$w_{i+2} = X$	& $t_i = T$

Table 1: Features on the current history h_i

Word:	the	stories	about	well-heeled	communities	and	developers
Tag:	DT	NNS	IN	JJ	NNS	CC	NNS
Position:	1	2	3	4	5	6	7

Table 2: Sample Data

A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

Word:	the	stories	about	well-heeled	communities	and	developers
Tag:	DT	NNS	IN	JJ	NNS	CC	NNS
Position:	1	2	3	4	5	6	7

Table 2: Sample Data

Condition	Features	
w_i is not rare	$w_i = X$	$\& \ t_i = T$
w_i is rare	X is prefix of w_i, $\lvert X \rvert \leq 4$	$\& \ t_i = T$
	X is suffix of w_i, $\lvert X \rvert \leq 4$	$\& \ t_i = T$
	w_i contains number	$\& \ t_i = T$
	w_i contains uppercase character	$\& \ t_i = T$
	w_i contains hyphen	$\& \ t_i = T$
$\forall \ w_i$	$t_{i-1} = X$	$\& \ t_i = T$
	$t_{i-2}t_{i-1} = XY$	$\& \ t_i = T$
	$w_{i-1} = X$	$\& \ t_i = T$
	$w_{i-2} = X$	$\& \ t_i = T$
	$w_{i+1} = X$	$\& \ t_i = T$
	$w_{i+2} = X$	$\& \ t_i = T$

Table 1: Features on the current history h_i

feature templates

$w_i =$ about	$\& \ t_i =$ IN
$w_{i-1} =$ stories	$\& \ t_i =$ IN
$w_{i-2} =$ the	$\& \ t_i =$ IN
$w_{i+1} =$ well-heeled	$\& \ t_i =$ IN
$w_{i+2} =$ communities	$\& \ t_i =$ IN
$t_{i-1} =$ NNS	$\& \ t_i =$ IN
$t_{i-2}t_{i-1} =$ DT NNS	$\& \ t_i =$ IN

the non-zero features for position 3

A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

Word:	the	stories	about	well-heeled	communities	and	developers
Tag:	DT	NNS	IN	JJ	NNS	CC	NNS
Position:	1	2	3	4	5	6	7

Table 2: Sample Data

Condition	Features	
w_i is not rare	$w_i = X$	$\& \; t_i = T$
w_i is rare	X is prefix of w_i, $\lvert X\rvert \leq 4$	$\& \; t_i = T$
	X is suffix of w_i, $\lvert X\rvert \leq 4$	$\& \; t_i = T$
	w_i contains number	$\& \; t_i = T$
	w_i contains uppercase character	$\& \; t_i = T$
	w_i contains hyphen	$\& \; t_i = T$
$\forall \; w_i$	$t_{i-1} = X$	$\& \; t_i = T$
	$t_{i-2}t_{i-1} = XY$	$\& \; t_i = T$
	$w_{i-1} = X$	$\& \; t_i = T$
	$w_{i-2} = X$	$\& \; t_i = T$
	$w_{i+1} = X$	$\& \; t_i = T$
	$w_{i+2} = X$	$\& \; t_i = T$

Table 1: Features on the current history h_i

$w_{i-1} =$ about	$\& \; t_i =$ JJ
$w_{i-2} =$ stories	$\& \; t_i =$ JJ
$w_{i+1} =$ communities	$\& \; t_i =$ JJ
$w_{i+2} =$ and	$\& \; t_i =$ JJ
$t_{i-1} =$ IN	$\& \; t_i =$ JJ
$t_{i-2}t_{i-1} =$ NNS IN	$\& \; t_i =$ JJ
prefix(w_i)=w	$\& \; t_i =$ JJ
prefix(w_i)=we	$\& \; t_i =$ JJ
prefix(w_i)=wel	$\& \; t_i =$ JJ
prefix(w_i)=well	$\& \; t_i =$ JJ
suffix(w_i)=d	$\& \; t_i =$ JJ
suffix(w_i)=ed	$\& \; t_i =$ JJ
suffix(w_i)=led	$\& \; t_i =$ JJ
suffix(w_i)=eled	$\& \; t_i =$ JJ
w_i contains hyphen	$\& \; t_i =$ JJ

the non-zero features for position 4

A Maximum Entropy Model for POS Tagging

- How do we learn the weights **w**?
  - Train on manually annotated data (supervised learning).

- What does it mean "train **w** on annotated corpus"?
  - Probabilistic Discriminative Models:
    - Maximum Entropy (Logistic Regression).[Ratnaparkhi, EMNLP'96]
  - Distribution Free Methods:
    - (Average) Perceptrons. [Collins, ACL 2002]
    - Support Vector Machines (SVMs).

A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Probabilistic Discriminative Model:
  - ⇒ need to transform *score*(t,h_i) into probability $p(t\,|\,h_i)$.

$$p(t\,|\,h_i) = \frac{\exp(\mathrm{w}^{\mathrm{T}}\phi(t,h_i))}{\sum_{t'}\exp(\mathrm{w}^{\mathrm{T}}\phi(t',h_i))}$$

- Training using:
  - Maximum Likelihood (ML).
  - Maximum A Posteriori (MAP) with a Gaussian prior on **w**.
- Inference (i.e. Testing):

$$\hat{t}_i = \arg\max_{t_i \in T} p(t_i\,|\,h_i) = \arg\max_{t_i \in T} \exp(w^T\varphi(t_i,h_i)) = \arg\max_{t_i \in T} w^T\varphi(t_i,h_i)$$

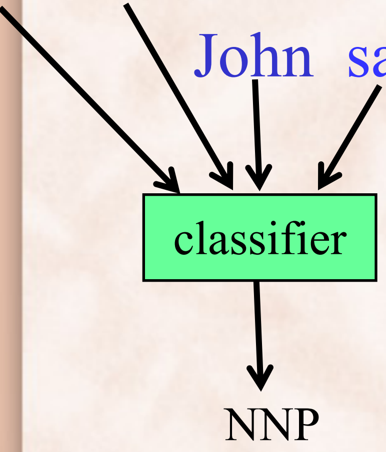
A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

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



A Maximum Entropy Model for POS Tagging

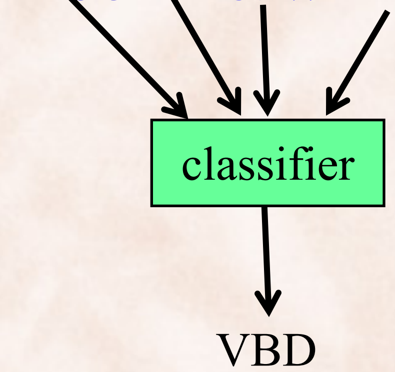
[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

NNP

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



A Maximum Entropy Model for POS Tagging

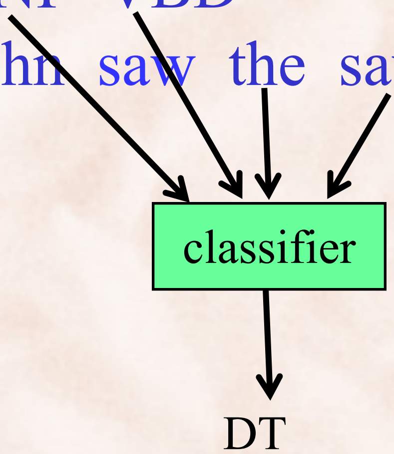
[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

NNP VBD

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

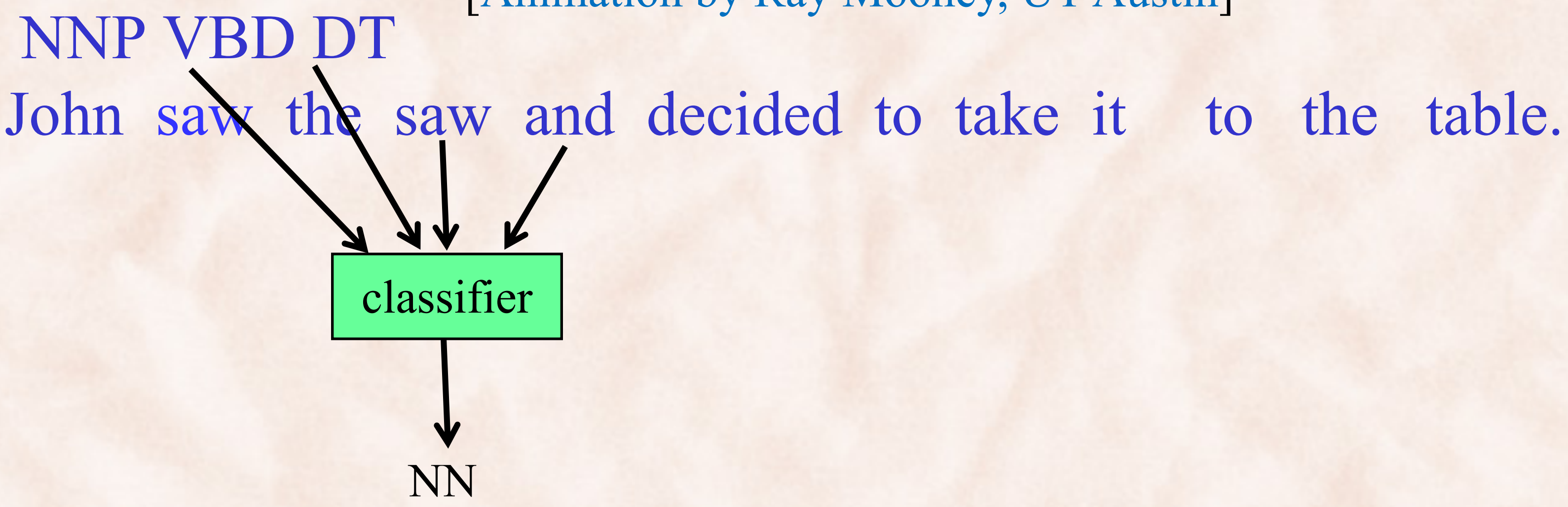


A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

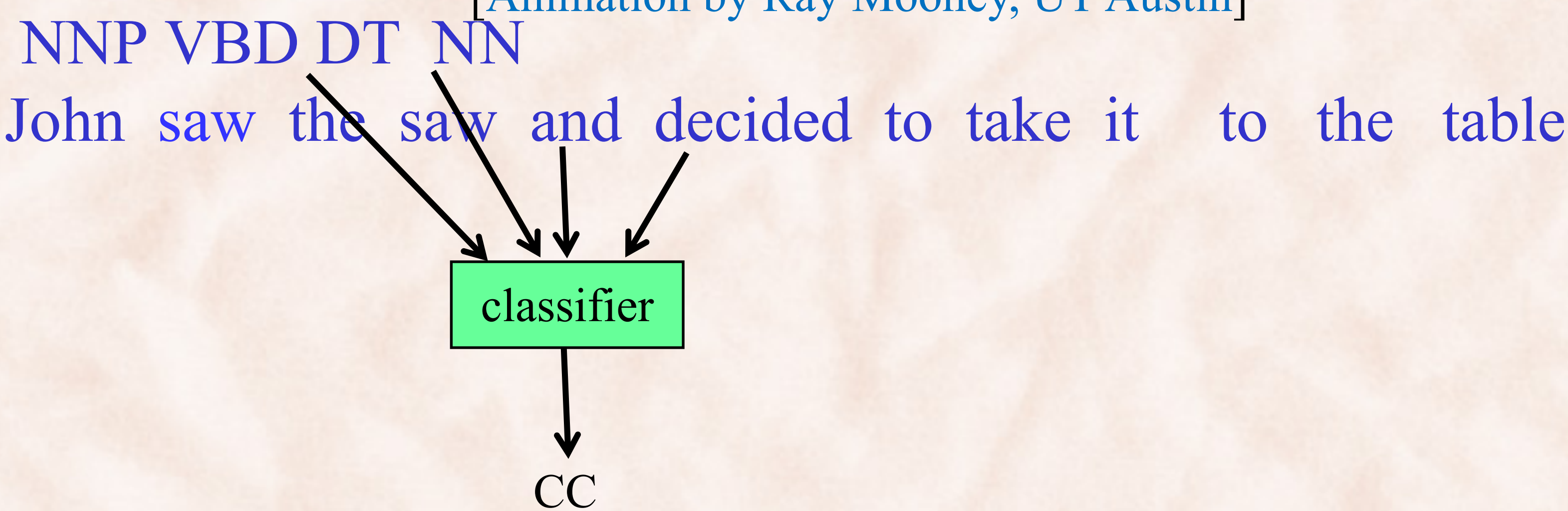


A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

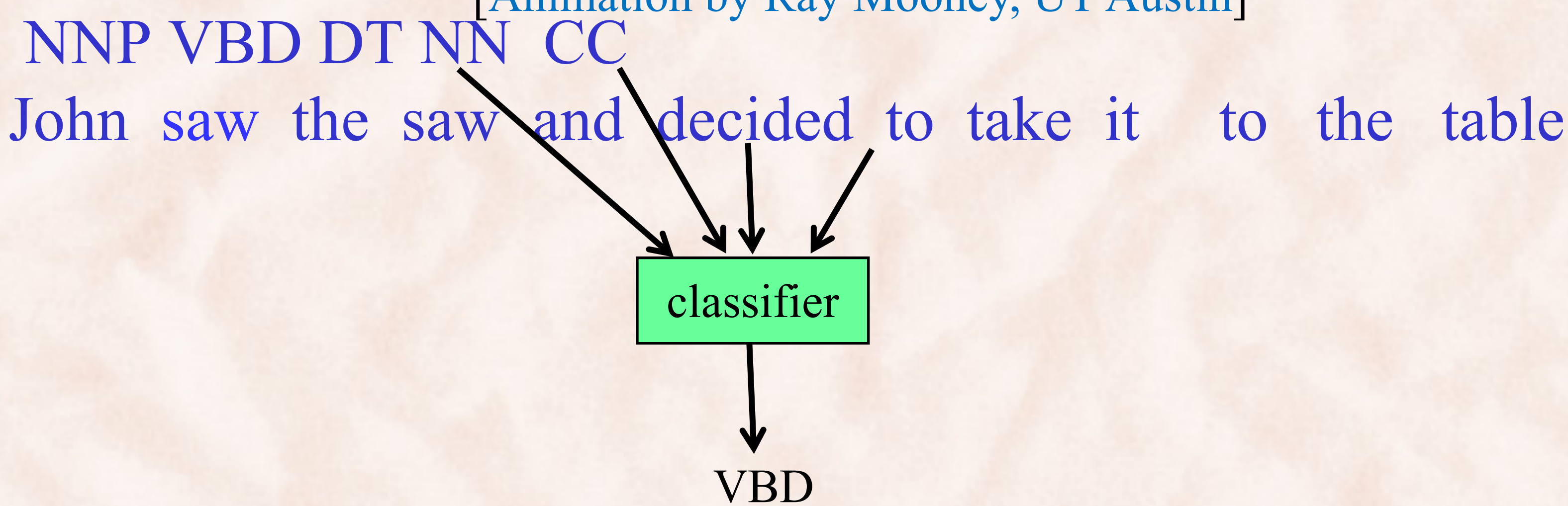


A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP’96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

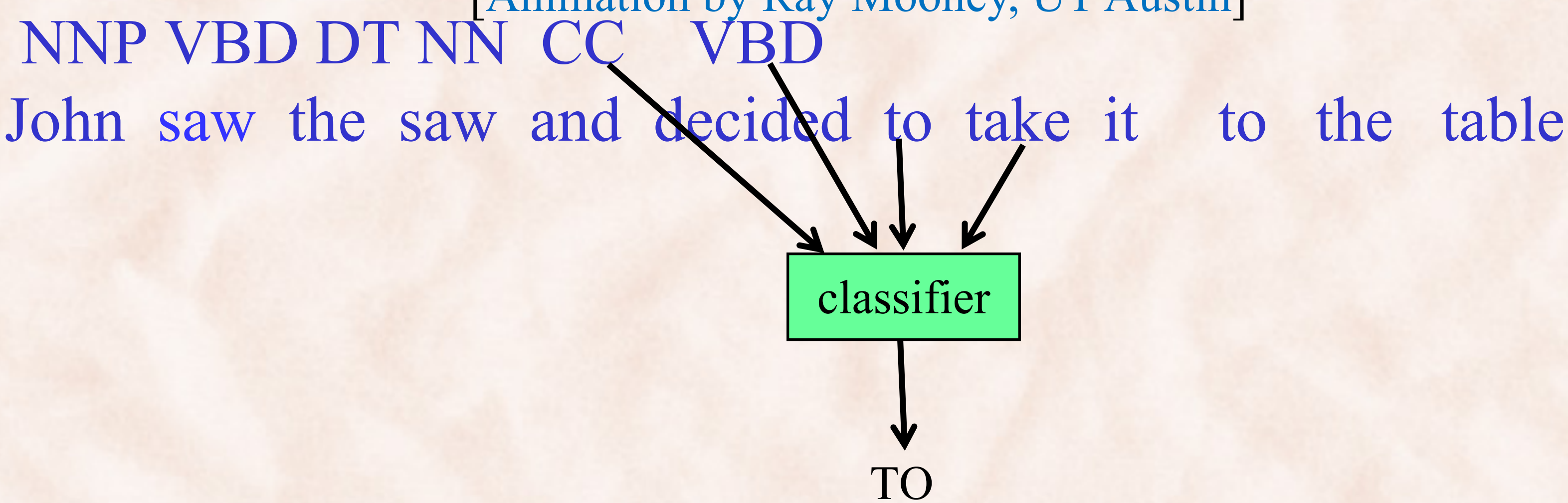


A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]



A Maximum Entropy Model for POS Tagging

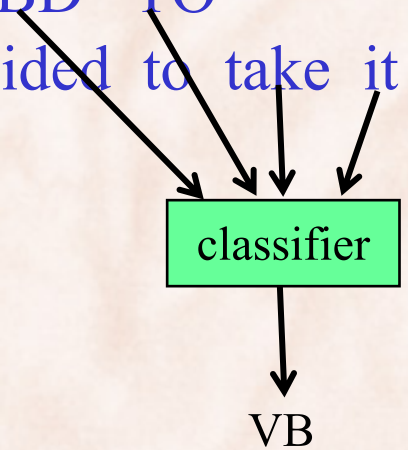
[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

NNP VBD DT NN CC VBD TO

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



A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

NNP VBD DT NN CC VBD TO VB

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

classifier

PRP

A Maximum Entropy Model for POS Tagging

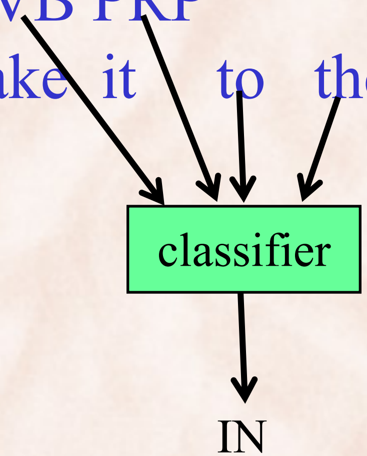
[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

NNP VBD DT NN CC VBD TO VB PRP

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



A Maximum Entropy Model for POS Tagging

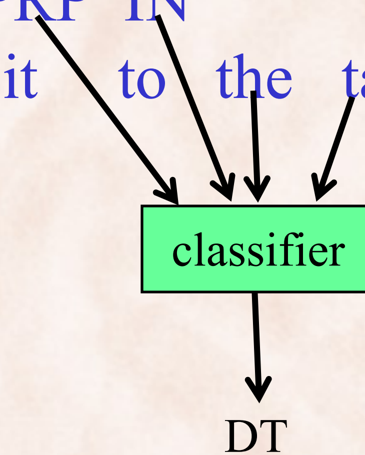
[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

NNP VBD DT NN CC VBD TO VB PRP IN

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



A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

NNP VBD DT NN CC VBD TO VB PRP IN DT

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

classifier

NN

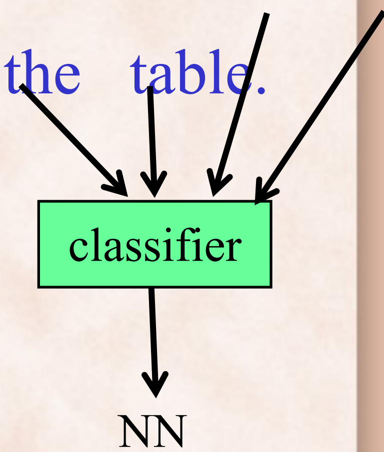
A Maximum Entropy Model for POS Tagging

[Ratnaparkhi, EMNLP'96]

- Inference, need to do Forward traversal of input sequence:

[Animation by Ray Mooney, UT Austin]

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



- Some POS tags would be easier to disambiguate backward, what can we do?
  - Use backward traversal, with backward features … but lose forward info.

Sequence Labeling as Classification

1) **Classifiy** each token **individually** into one of a number of classes.

2) **Classify** all tokens **jointly** into one of a number of classes:

$$\hat{t}_1 \ldots \hat{t}_n = \arg\max_{t_1,\ldots,t_n} \lambda^T \varphi(t_1,\ldots,t_n, w_1,\ldots,w_n)$$

- **Hidden Markov Models**.
- **Conditional Random Fields**.
- Structural SVMs.
- Discriminatively Trained HMMs [Collins, EMNLP'02].
- Bi-directional RNNs / LSTM-CRFs.

Hidden Markov Models

- **Probabilistic Generative Models**:

$$\hat{t}_1 \ldots \hat{t}_n = \arg\max_{t_1,\ldots,t_n} p(t_1,\ldots,t_n \mid w_1,\ldots,w_n)$$

$$= \arg\max_{t_1,\ldots,t_n} \underbrace{p(w_1,\ldots,w_n \mid t_1,\ldots,t_n)}_{\textit{Use state emission probs}} \underbrace{p(t_1,\ldots,t_n)}_{\textit{Use state transition probs}}$$

Hidden Markov Models: Assumptions

1) A word event depends only on its POS tag:

$$p(w_1,\ldots,w_n \mid t_1,\ldots,t_n) = \prod_{i=1}^{n} p(w_i \mid t_i)$$

2) A tag event depends only on the previous tag:

$$p(t_1,\ldots,t_n) = \prod_{i=1}^{n} p(t_i \mid t_{i-1})$$

$\Rightarrow$ POS tagging is $\hat{t}_1 \ldots \hat{t}_n = \arg\max_{t_1,\ldots,t_n} \prod_{i=1}^{n} p(w_i \mid t_i)\, p(t_i \mid t_{i-1})$

Interlude

Tales of HMMs

Structured Data

- For many applications, the i.i.d. assumption 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 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

Probabilistic Graphical Models

- PGMs use a graph for **compactly**:
  1. Encoding a complex distribution over a multi-dimensional space.
  2. Representing a set of independencies that hold in the distribution.
  - Properties 1 and 2 are, in a "deep sense", equivalent.

- Probabilistic Graphical Models:
  - **Directed**:
    - i.e. Bayesian Networks i.e. Belief Networks.
  - **Undirected**:
    - i.e. Markov Random Fields

Probabilistic Graphical Models

- **Directed PGMs**:
  - Bayesian Networks:
    - Dynamic Bayesian Networks:
      - State Observation Models:
        - » Hidden Markov Models.
        - » Linear Dynamical Systems (Kalman filters).

- **Undirected PGMs**:
  - Markov Random Fields (MRF).
    - Conditional Random Fields (CRF).
      - Sequential CRFs.

Bayesian Networks

- A **Bayesian Network** structure G is a directed acyclic graph whose nodes $X_1, X_2, \ldots, X_n$ represent random variables and edges correspond to "direct influences" between nodes:
  - Let $Pa(X_i)$ denote the parents of X_i in G;
  - Let $NonDescend(X_i)$ denote the variables in the graph that are not descendants of X_i.
  - Then G encodes the following set of conditional independence assumptions, called the local independencies:

For each X_i in G: $X_i \perp NonDescend(X_i) \mid Pa(X_i)$

Bayesian Networks

1. Because $X_i \perp NonDescend(X_i) \mid Pa(X_i)$, it follows that:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid Pa(X_i)\right)$$

2. More generally, **d-separation**:
   1. Two sets of nodes X and Y are conditionally independent given a set of nodes E ($X \perp Y \mid E$) if X and Y are d-separated by E.

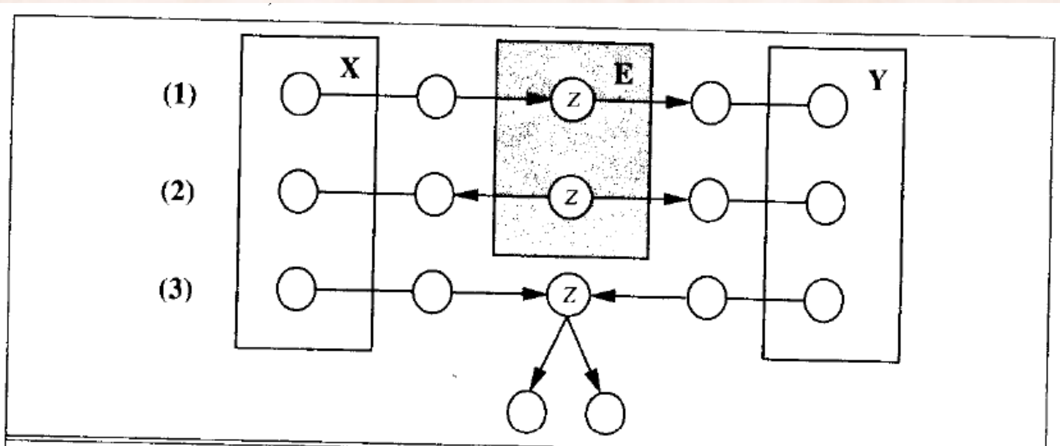
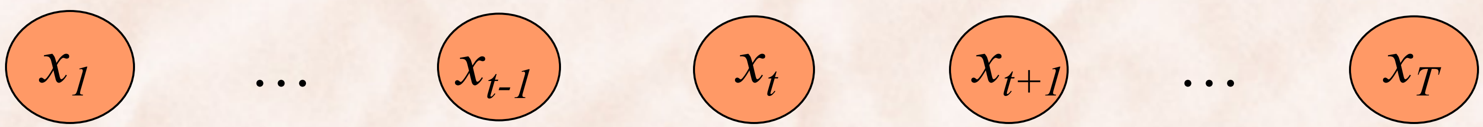


Figure 15.4 Three ways in which a path from X to Y can be blocked, given the evidence E. If every path from X to Y is blocked, then we say that E d-separates X and Y.

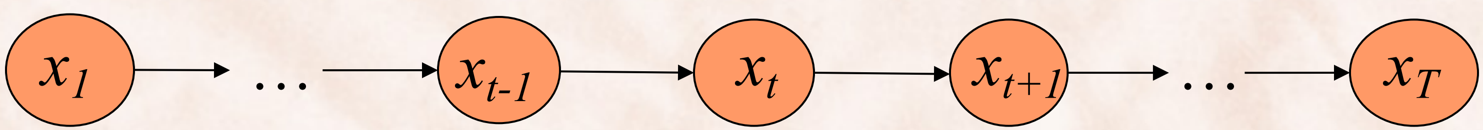
Sequential Data

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*.

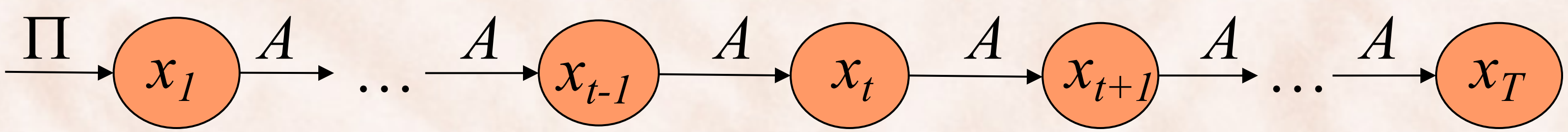
Markov Models: Parameters

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

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

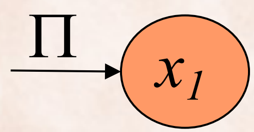
- $A = \{a_{ij}\}$ are the state transition probabilities.

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

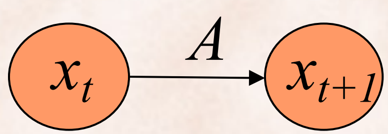


Markov Models as DBNs

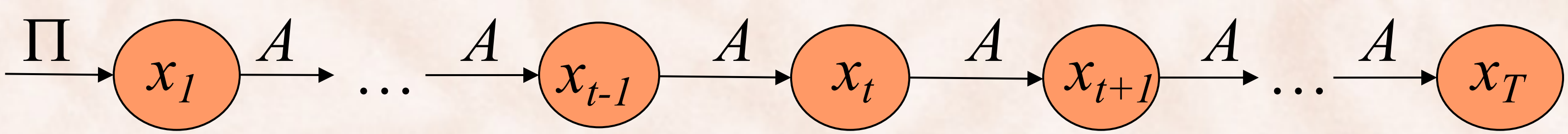
- A Markov Model is a **Dynamic Bayesian Network**:
  1. $B_0 = \Pi$ is the initial distribution over states.



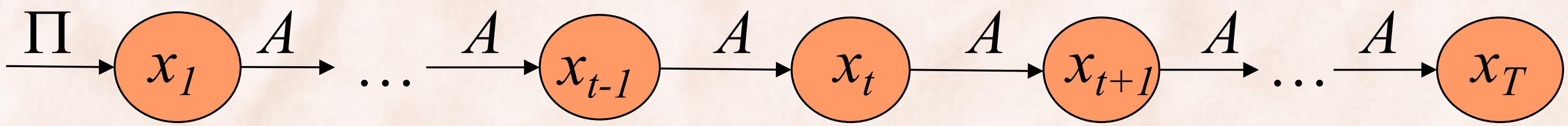
  1. $B_{\rightarrow} = A$ is the 2-time-slice Bayesian Network (2-TBN).



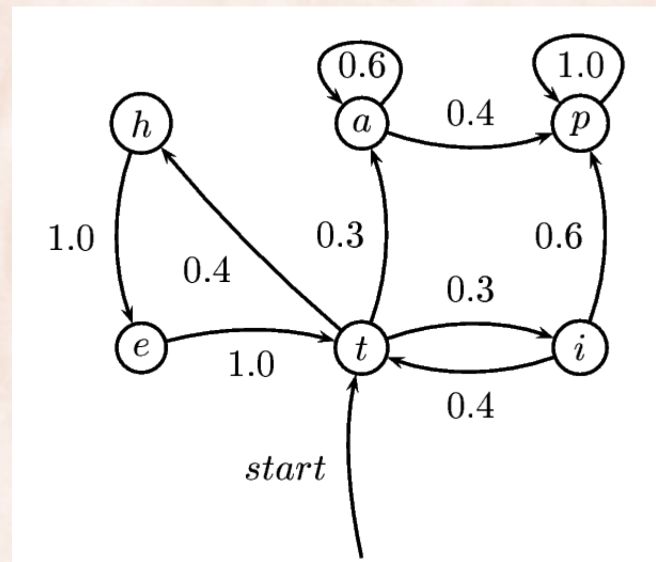
  – The **unrolled** DBN (Markov model) over T time steps:



Markov Models: Inference



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



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

mth 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:

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

- mth order Markov model:

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

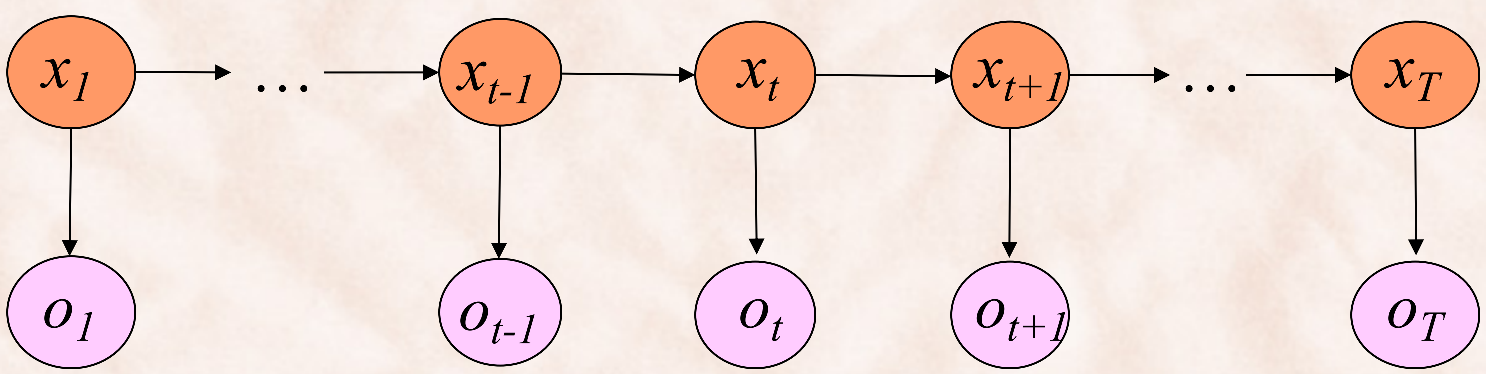
Markov Models

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

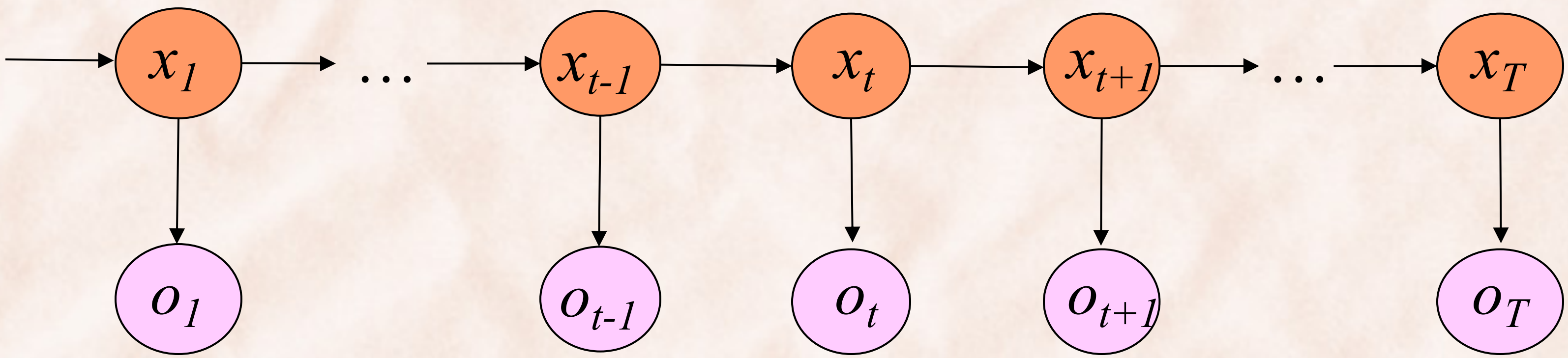
- **Hidden Markov Models**:
  - The *states* are hidden (latent) variables.
  - The states probabilistically generate surface events, or *observations*.
  - Efficient training using Expectation Maximization (EM)
    - Maximum Likelihood (ML) when tagged data is available.
  - Efficient inference using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

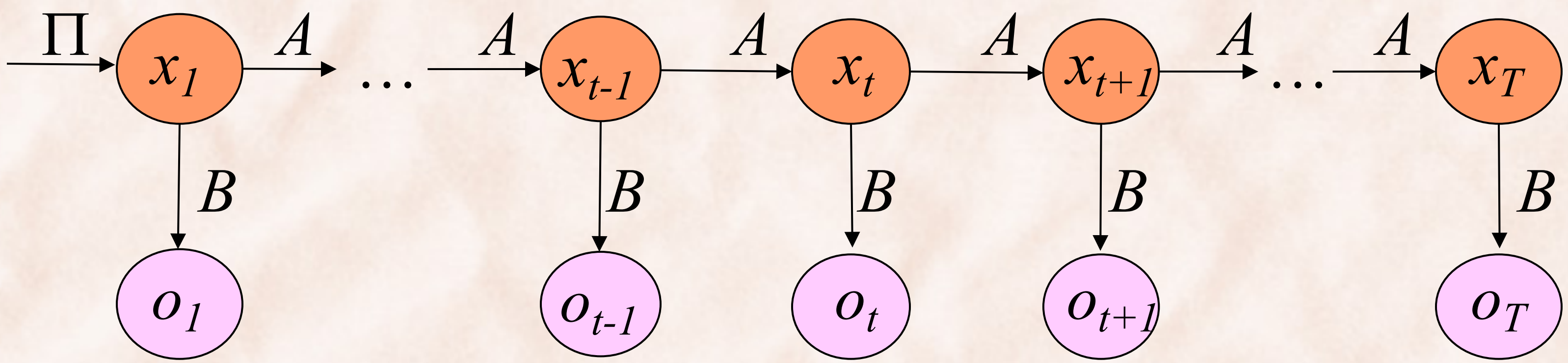


HMMs: Parameters



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

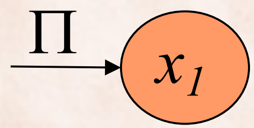


- $\Pi = \{\pi_i\}$, $i \in S$, are the initial state probabilities.
- $A = \{a_{ij}\}$ }, $i,j \in S$, are the state transition probabilities.
- $B = \{b_{ik}\}$, $i \in S$, $k \in K$, are the symbol emision probabilities.

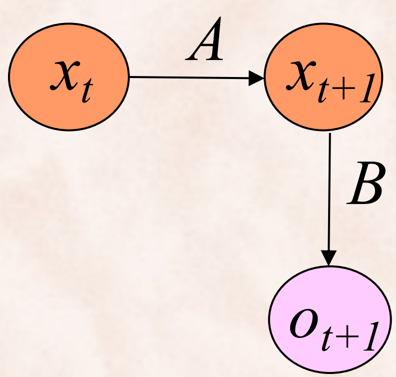
$$b_{ik} = P(o_t = k \mid x_t = s_i)$$

Hidden Markov Models as DBNs

- A Hidden Markov Model is a **Dynamic Bayesian Network**:
  1. $B_0 = \Pi$ is the initial distribution over states.



  1. $B_{\rightarrow} = A$ is the 2-time-slice Bayesian Network (2-TBN).



  – The **unrolled** DBN (Markov model) over T time steps (prev. slide).

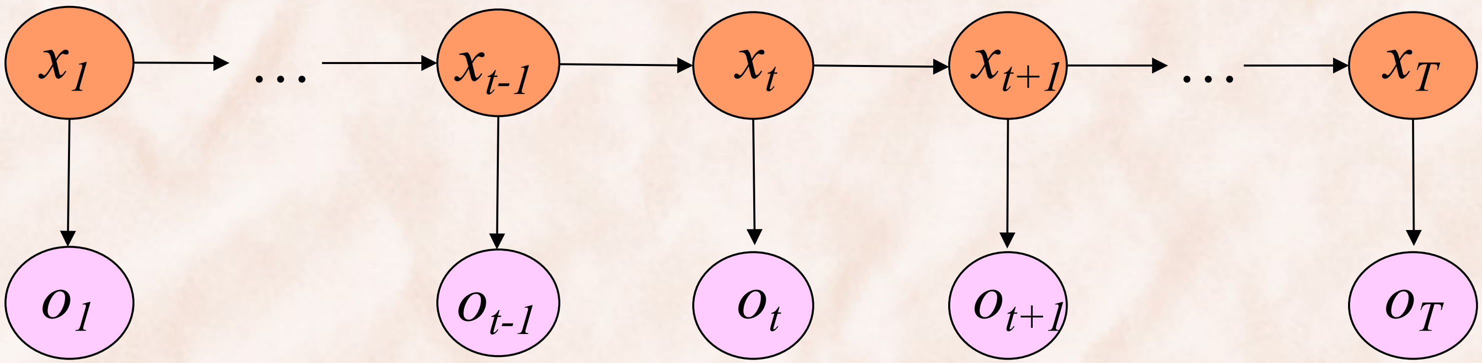
HMMs: Inference and Training

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

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

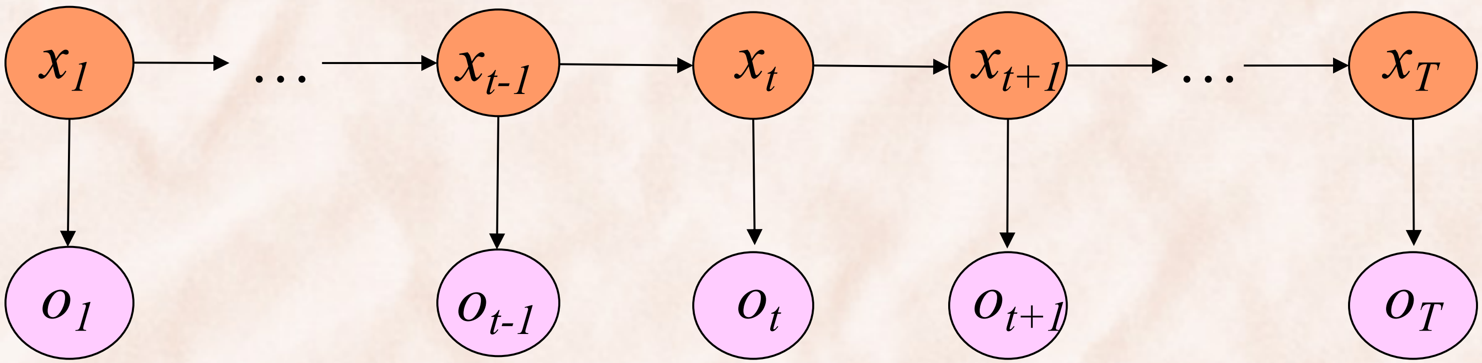
  3) Given an observation sequence O, find the model $\mu = (A, B, \Pi)$ that best explains the observed data (*EM*).
     - Given observation and state sequence O, X find μ (*ML*).

HMMs: Decoding



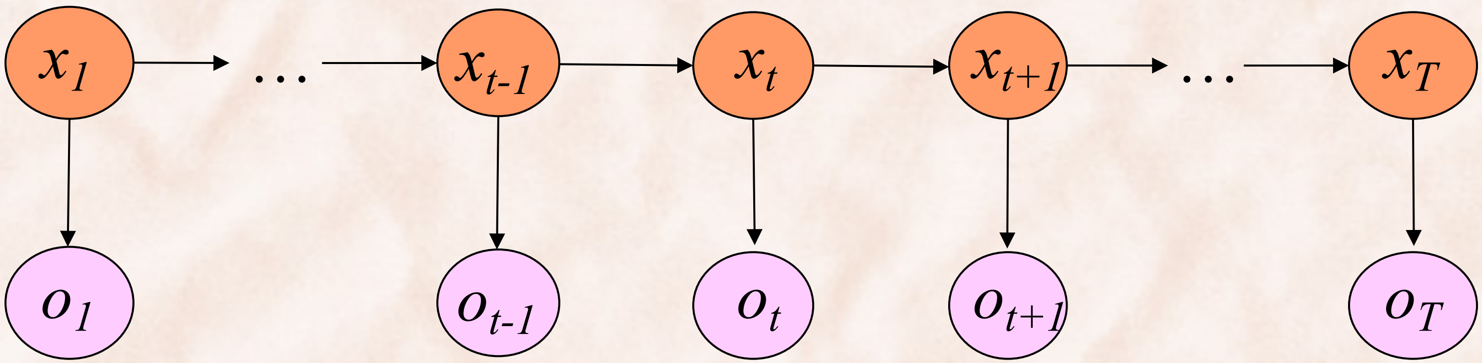
1) Given a model $\mu = (A, B, \Pi)$, compute the probability of a given observation sequence $O = o_1, \ldots, o_T$ i.e. $p(O|\mu)$

HMMs: Decoding



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

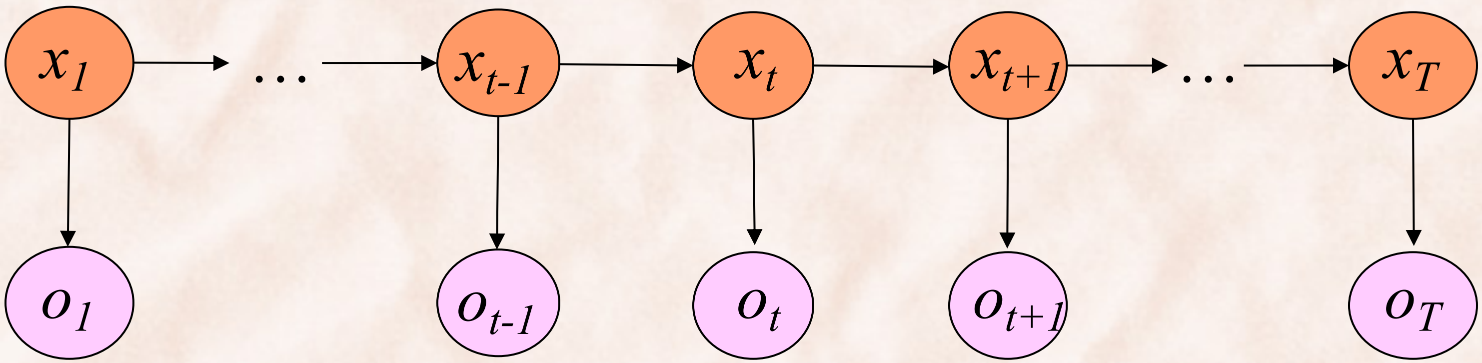
HMMs: Decoding



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

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

HMMs: Decoding

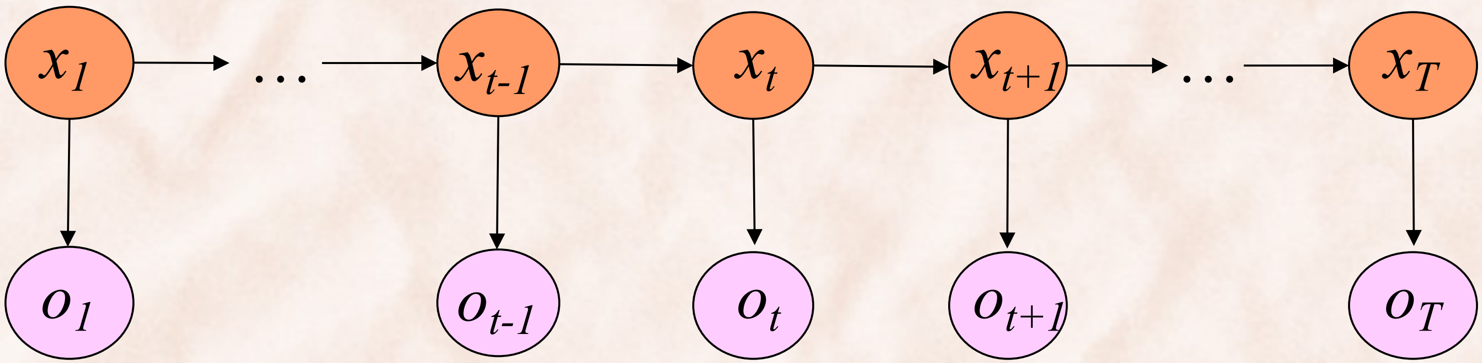


$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

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

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

HMMs: Decoding



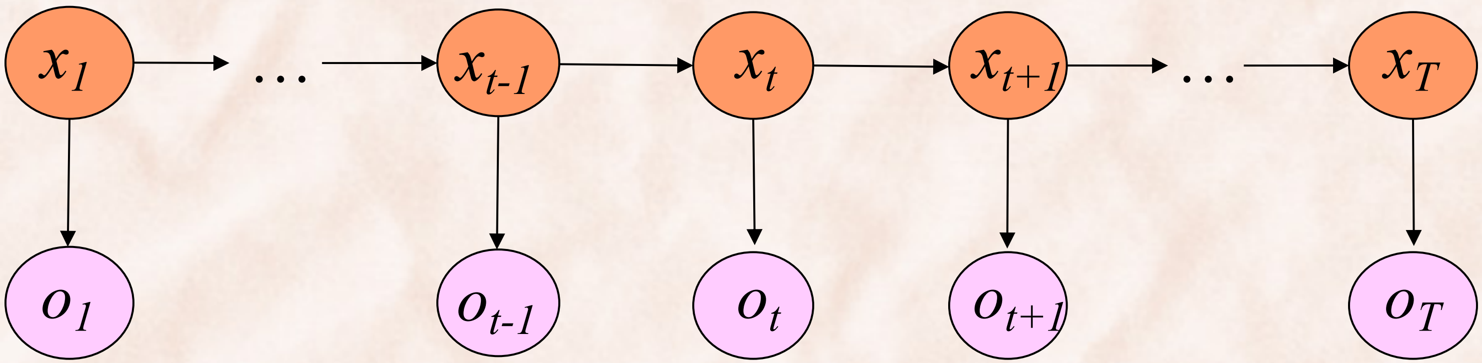
$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \dots b_{x_T o_T}$$

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

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

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

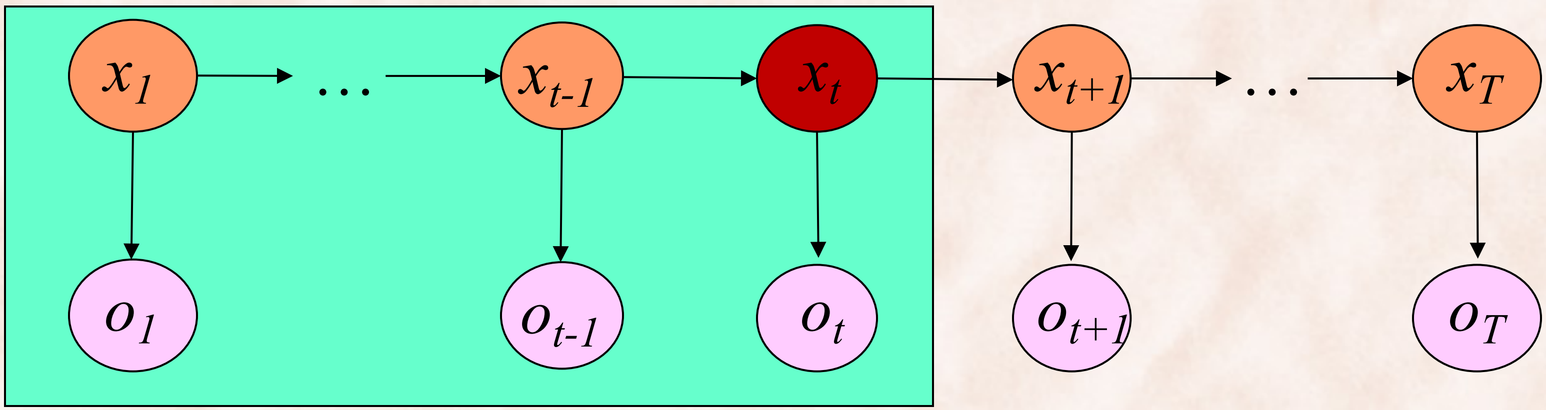
HMMs: Decoding



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

Time complexity?

HMMs: Forward Procedure



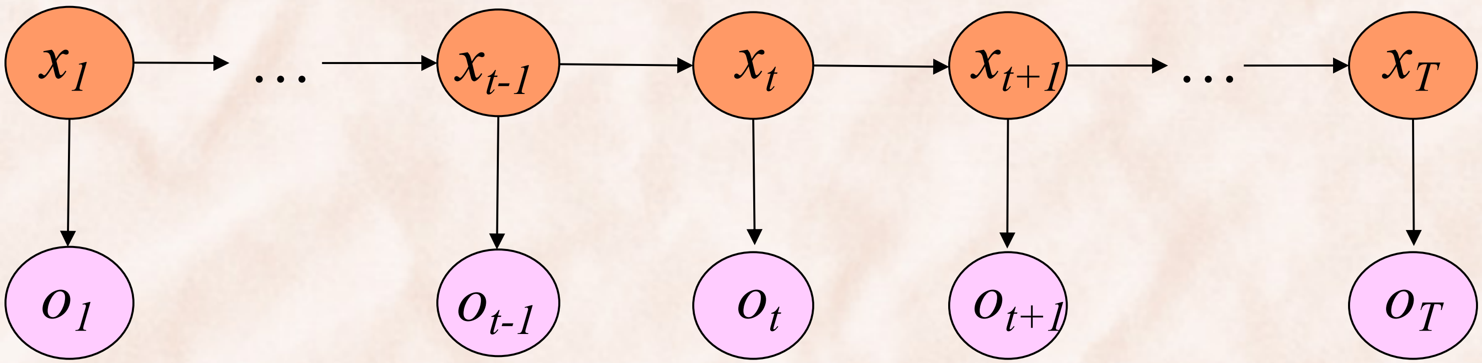
- Define:

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

- Then solution is:

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

HMMs: Decoding



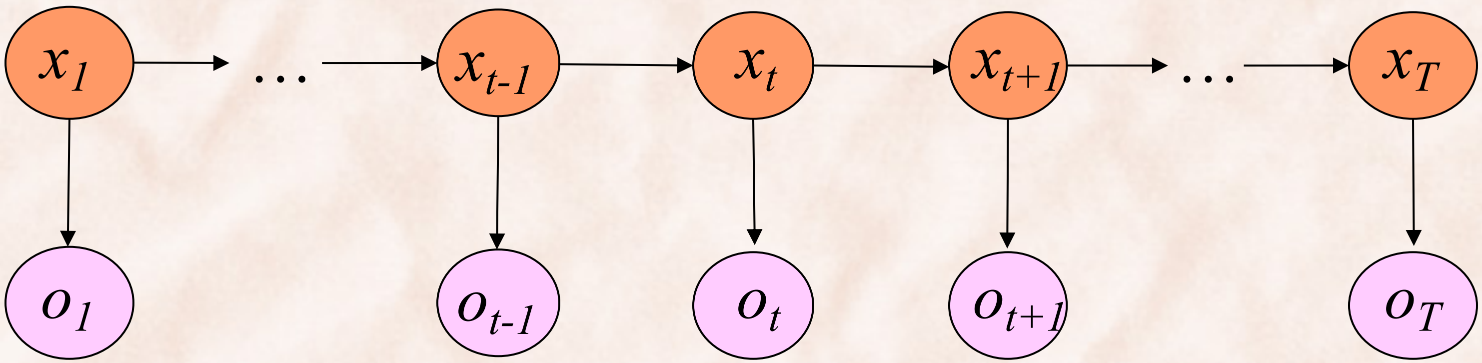
$$\alpha_j(t+1) \boxed{= P(o_1 \ldots o_{t+1}, x_{t+1} = j)}$$

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

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

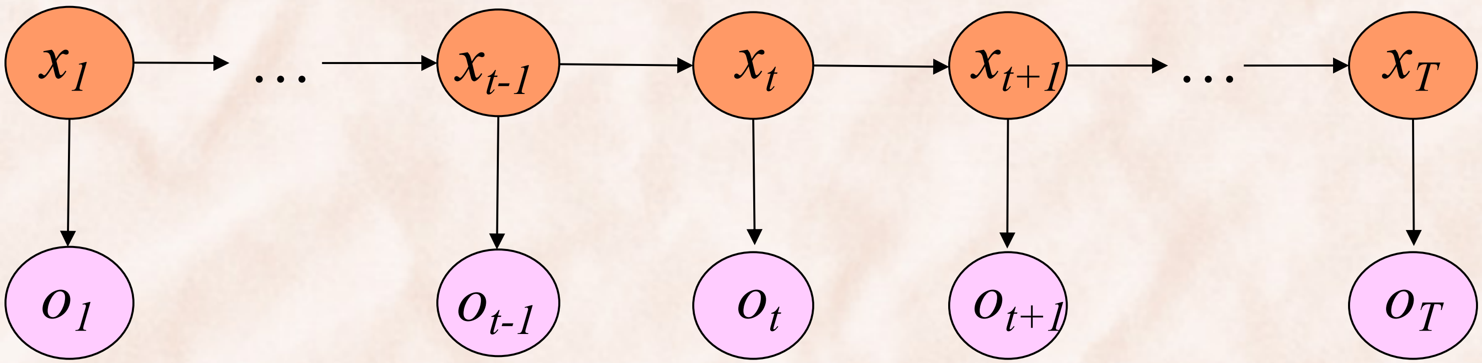
$$= P(o_1 \ldots o_t, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

HMMs: Decoding



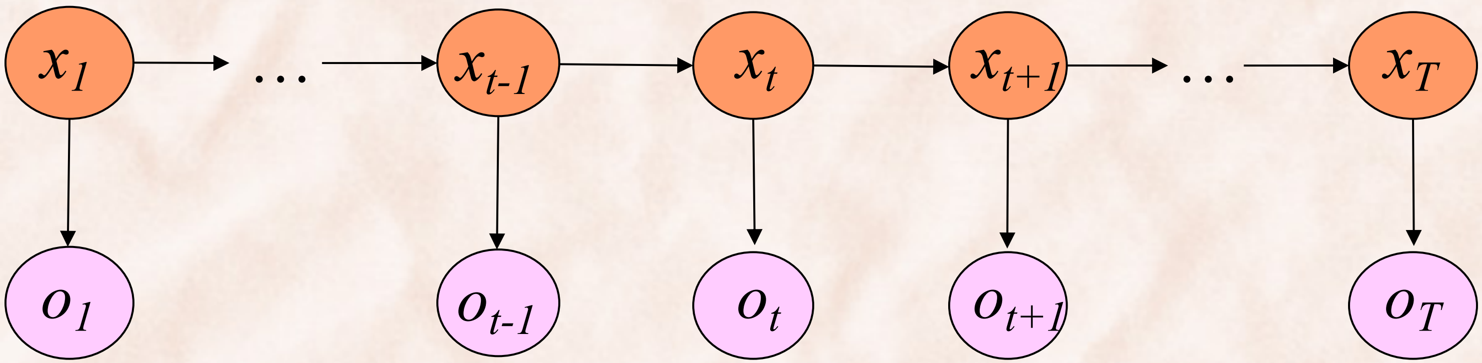
$$
\begin{aligned}
\alpha_j(t+1) &= P(o_1 \ldots o_{t+1}, x_{t+1} = j) \\
&= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j) \\
&= P(o_1 \ldots o_t \mid x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j) \\
&= P(o_1 \ldots o_t, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)
\end{aligned}
$$

HMMs: Decoding



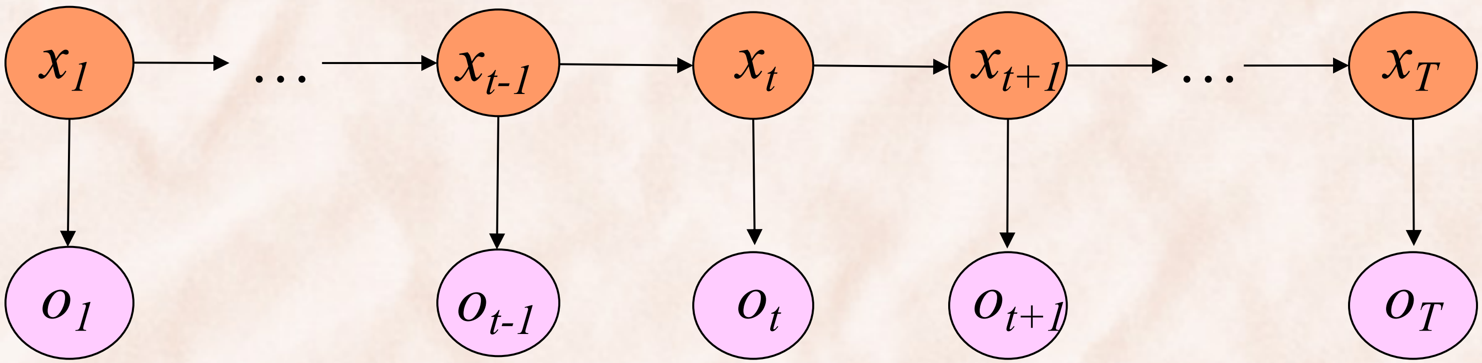
$$
\begin{aligned}
\alpha_j(t+1) &= P(o_1 \ldots o_{t+1}, x_{t+1} = j) \\
&= P(o_1 \ldots o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j) \\
&= \boxed{P(o_1 \ldots o_t \mid x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j) P(x_{t+1} = j)} \\
&= P(o_1 \ldots o_t, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)
\end{aligned}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

HMMs: Decoding



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

HMMs: Decoding

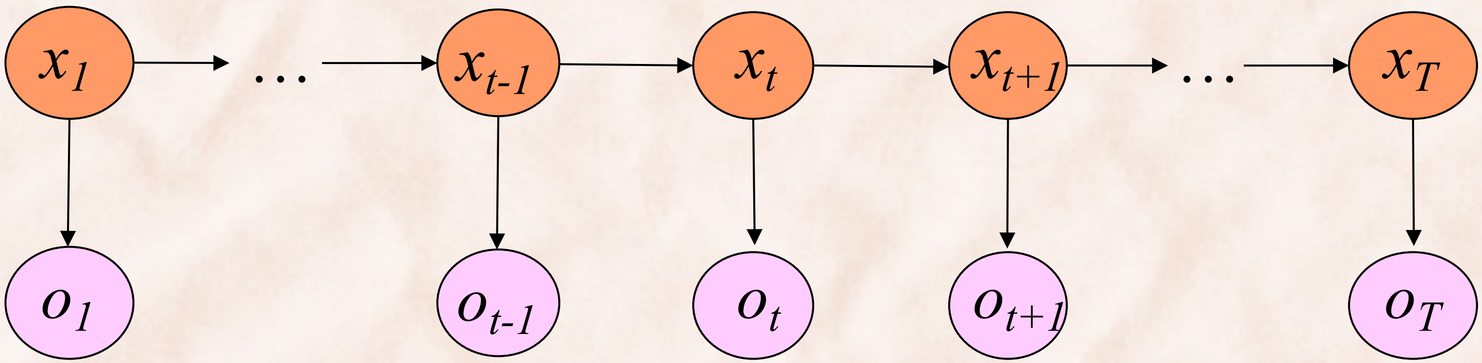


$$\alpha_j(t+1) = \boxed{\sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)}$$

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

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

HMMs: Decoding

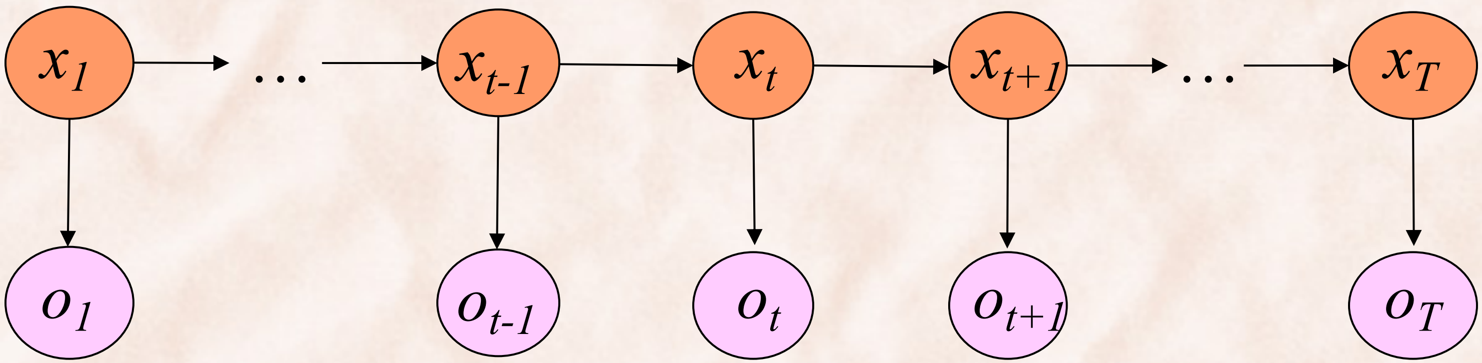


$$\alpha_j(t+1) = \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i, x_{t+1} = j) P(o_{t+1} \mid x_{t+1} = j)$$

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

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

HMMs: Decoding



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

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

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

The Forward Procedure

1. Initialization

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

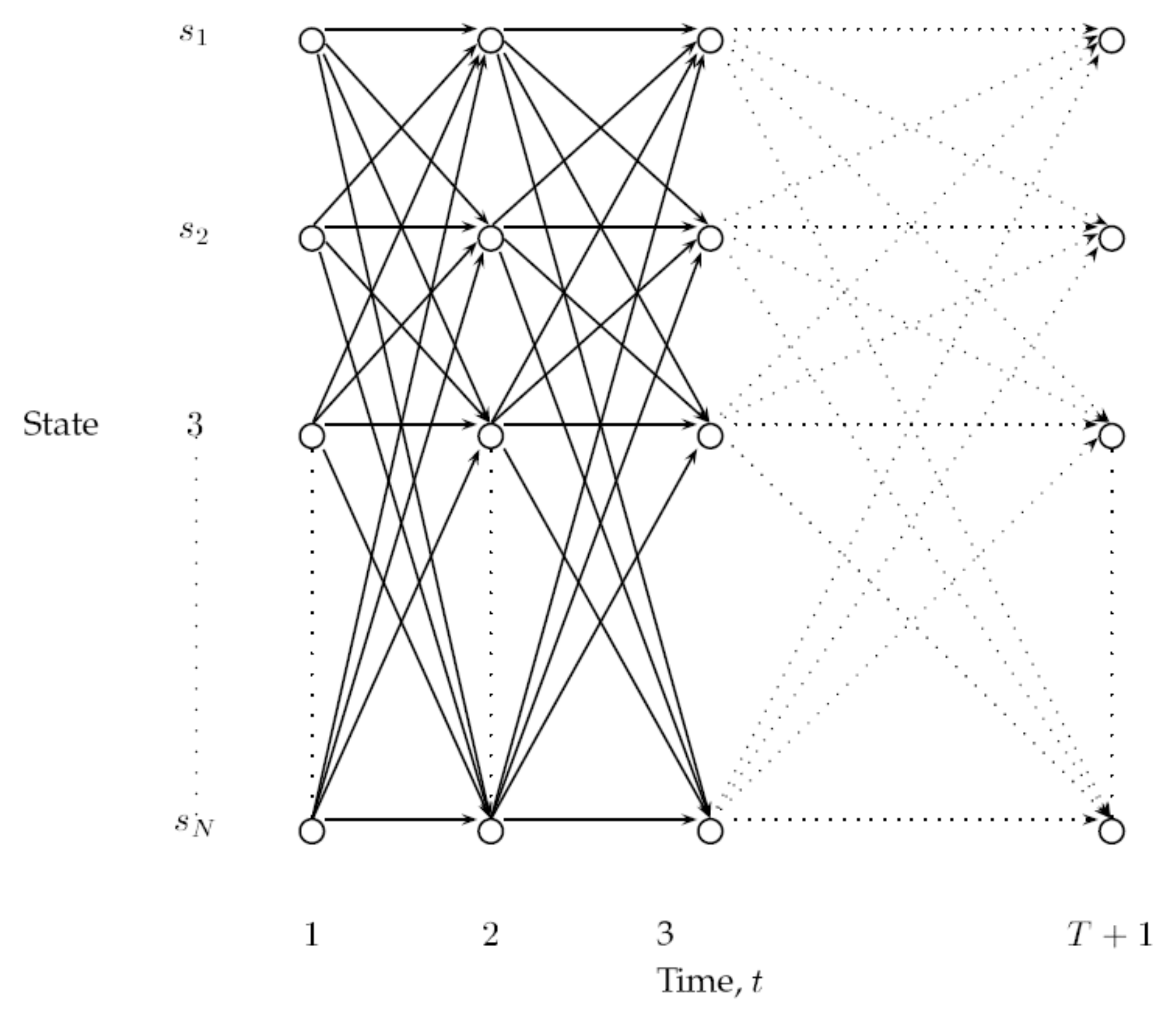
2. Recursion:

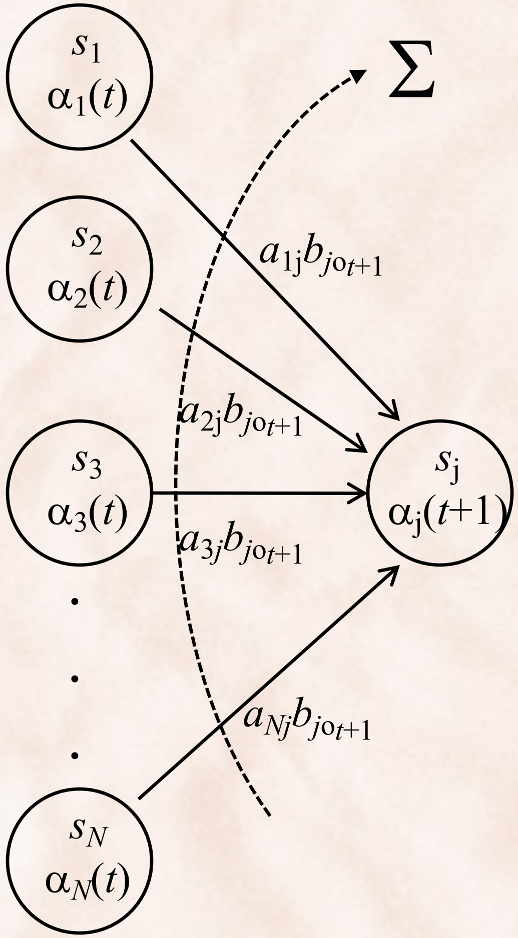
$$\alpha_j(t+1) = \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}, \qquad 1 \leq \mathrm{j} \leq \mathrm{N}, 1 \leq \mathrm{t} < \mathrm{T}$$

3. Termination:

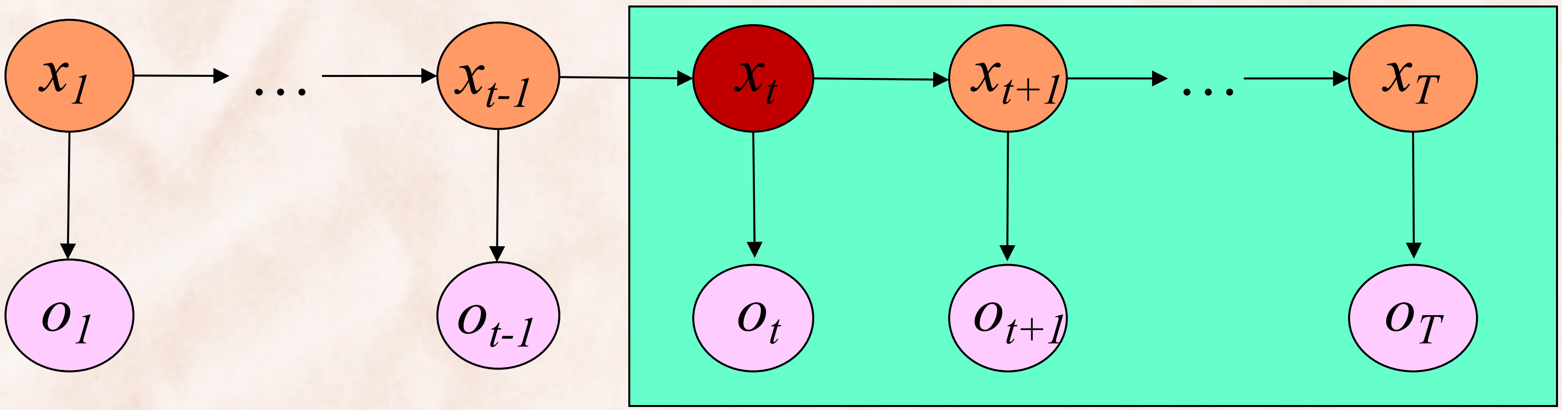
$$p(O \mid \mu) = \sum_{i=1}^{N} \alpha_i(T)$$

The Forward Procedure: Trellis Computation





HMMs: Backward Procedure



- Define:

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

- Then solution is:

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

The Backward Procedure

1. Initialization

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

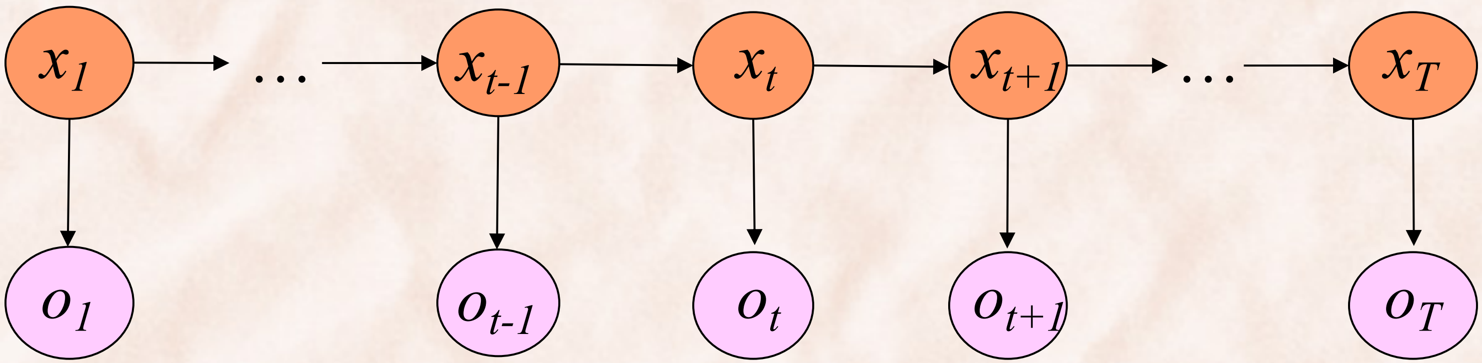
2. Recursion:

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

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

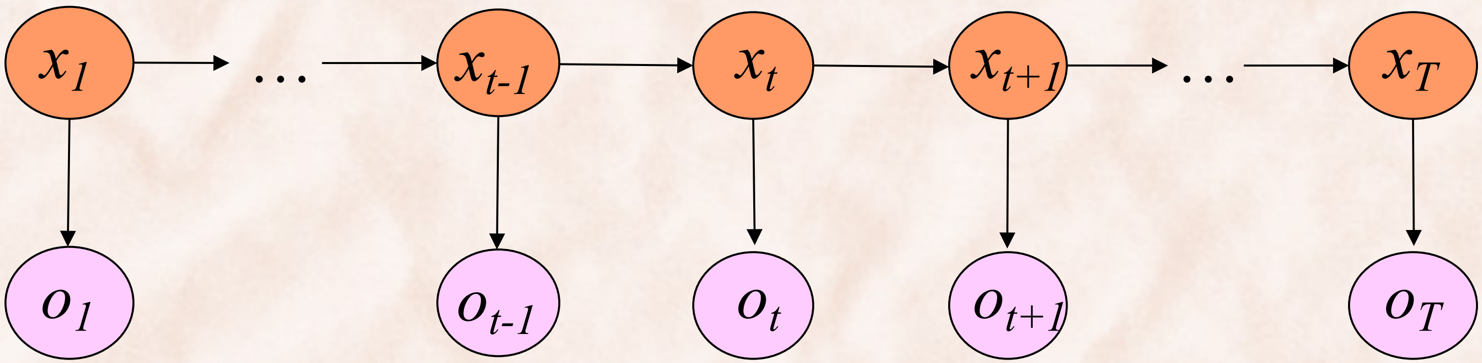
HMMs: Inference and Training

- 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



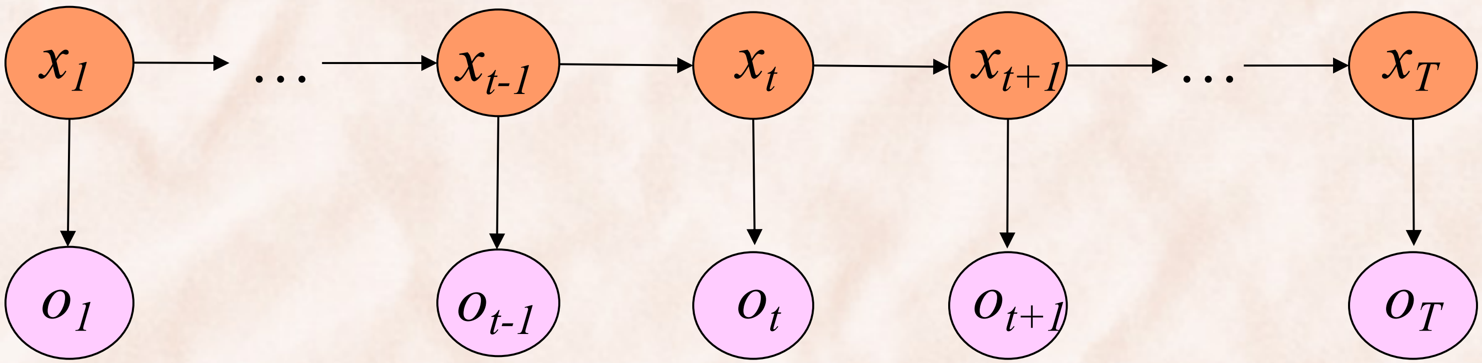
$$\hat{X} = \arg\max_{X} p(X \mid O, \mu)$$

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

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

Time complexity?

The Viterbi Algorithm



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

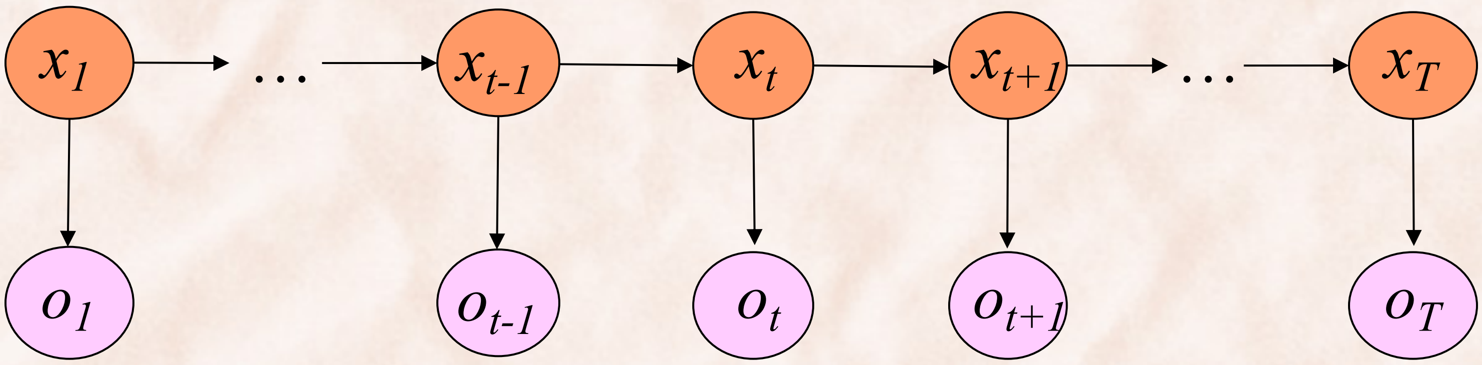
$$p(\hat{X}) = \max_{x_1,\ldots,x_T} p(x_1,\ldots,x_T, o_1,\ldots,o_T \mid \mu)$$

- 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 $x_t = j$:

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

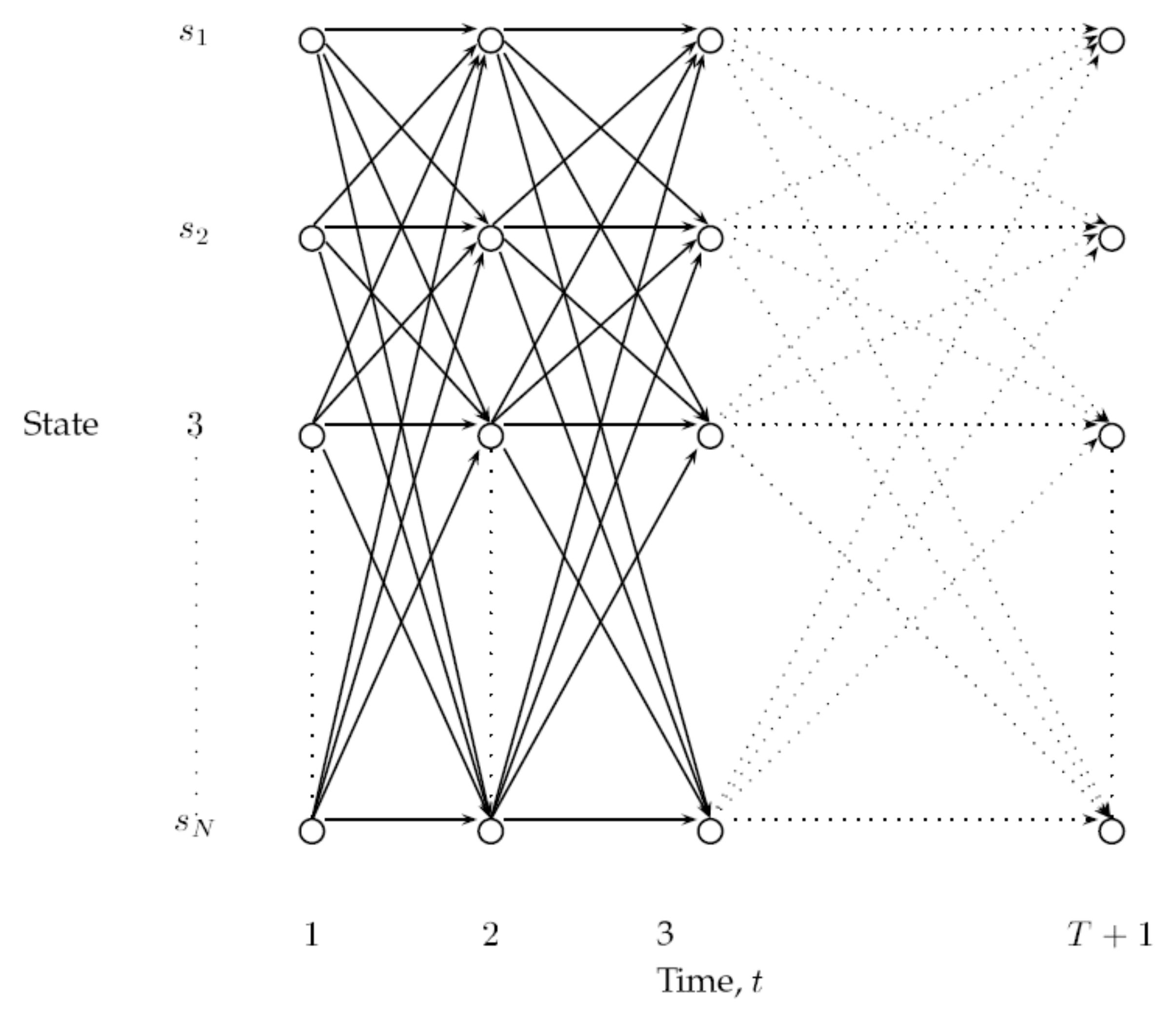
- It can be shown that:

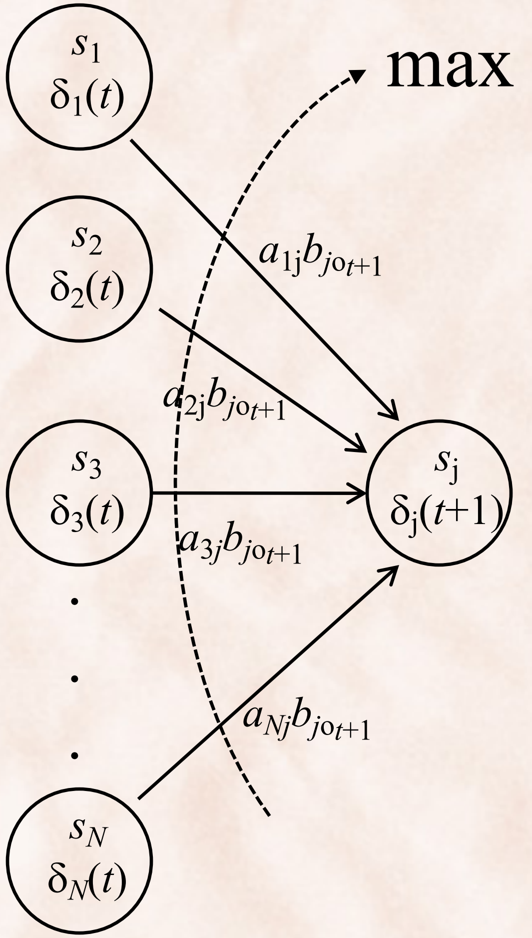
$$\delta_j(t+1) = \max_{1 \le i \le N} \delta_i(t) a_{ij} b_{jo_{t+1}}$$

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

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

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. $p(\mathrm{O}|\mu)$ (*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 μ =(A,B,Π).

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

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

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

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

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

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

Exercise:
Rewrite to use Laplace smoothing.

Parameter Estimation with Expectation Maximization

- Given observation sequences O find μ =(A,B,Π).

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

- There is no known analytic method to find solution.

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

  ⇒ 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.

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

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

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

$$\gamma_i(t) = \sum_{j=1 \ldots N} \xi_t(i, j) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{m=1..N} \alpha_m(t)\beta_m(t)}$$

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

The Baum-Welch Algorithm

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

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

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

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. $p(\mathrm{O}|\mu)$ (*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*).

Supplemental Reading

- Section 7.1, 7.2, 7.3, and 7.4 from Eisenstein.
- Chapter 8 in Jurafsky & Martin:
  – https://web.stanford.edu/~jurafsky/slp3/8.pdf
- Appendix A in Jurafsky & Martin:
  – https://web.stanford.edu/~jurafsky/slp3/A.pdf

POS Disambiguation: Context

"Here's a movie where you forgive the **preposterous** because it takes you to the **perplexing**."

[*Source Code*, by Roger Ebert, March 31, 2011]

"The **good**, the **bad**, and the **ugly**"

"The **young** and the **restless**"

"The **bold** and the **beautiful**"