Markov Model and Part-of-Speech Tagging

Arjun Mukherjee†

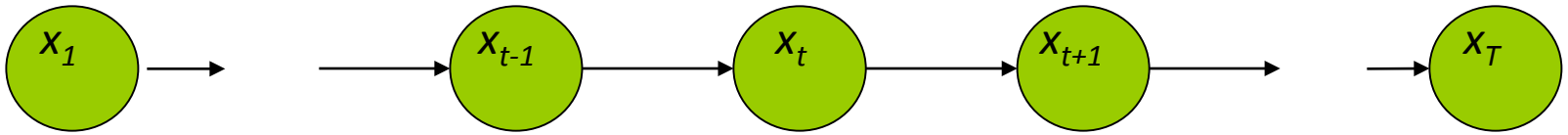
Course webpage:
http://www.cs.uh.edu/~arjun/courses/nlp

† Contains contents from [Manning et al., 2008] , and various other sources. Referenced in place.

Markov Models

- Sequence of random variables (e.g., through time) that are not independent
- Useful for modeling:
  - Speech/transcripts
  - Next word in the sequence
  - Sequence of words/events (over time)

- Consider $X = (X_1, X_2, \ldots X_T)$ as a sequence of random variables taking values in some finite state space $S = \{s_1, \ldots s_N\}$.



Markov Properties

- Sequence of random variables $X = (X_1, X_2, \ldots X_T)$ is said to be a Markov chain if it follows the following Markov properties

Limited Horizon:

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

First order Markov chain

Time invariant (stationary):

(9.2) $$= P(X_2 = s_k | X_1)$$

Independent of *t*

X is then said to be a Markov chain, or to have the Markov property.. One can describe a Markov chain by a stochastic transition matrix A:

(9.3) $$a_{ij} = P(X_{t+1} = s_j | X_t = s_i)$$

Stochastic property

Here, $a_{ij} \geq 0$, $\forall i, j$ and $\sum_{j=1}^{N} a_{ij} = 1, \forall i$.

Additionally one needs to specify Π, the probabilities of different initial states for the Markov chain:

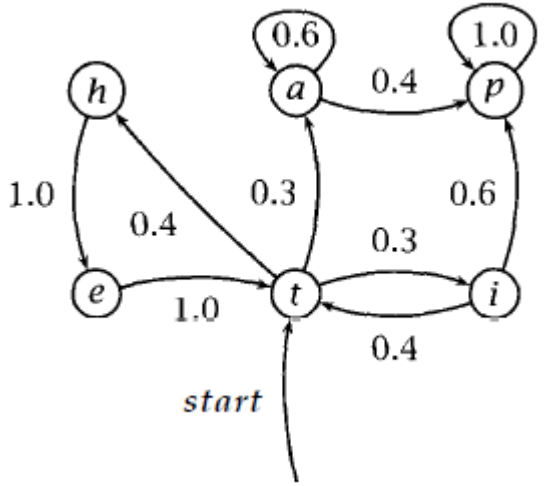
(9.4) $$\pi_i = P(X_1 = s_i)$$

Here, $\sum_{i=1}^{N} \pi_i = 1$. The need for this vector can be avoided by specifying that the Markov model always starts off in a certain extra initial state, s_0, and then using transitions from that state contained within the matrix $\mathbf{A}$ to specify the probabilities that used to be recorded in Π.

Markov Models as FSA

- One can think Markov models as non-deterministic (or stochastic) finite state automations (FSA) with fixed state transition probabilities.

- Consider the following state space

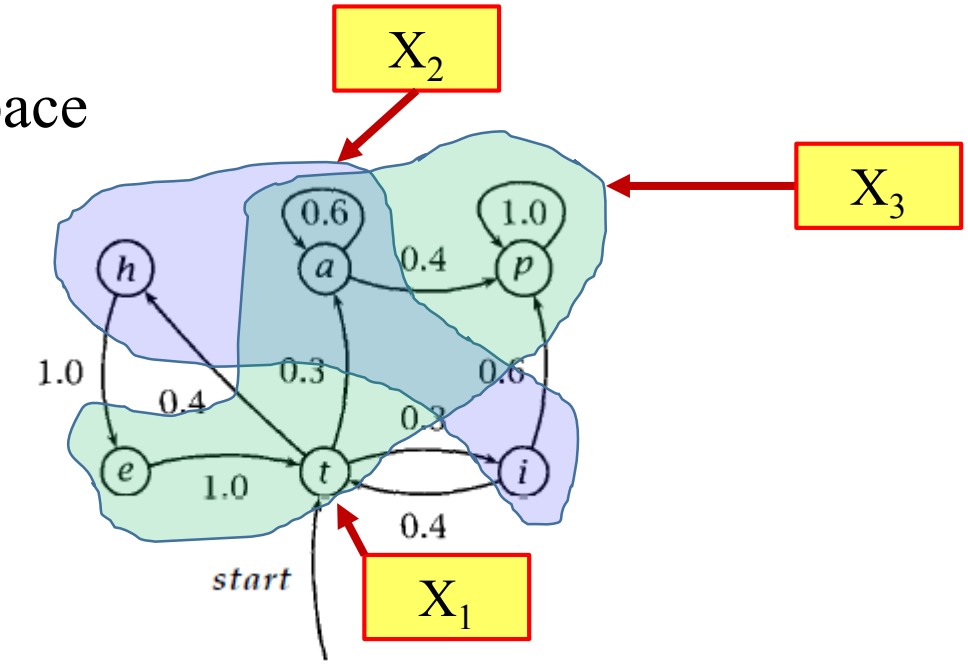


- **Q: What is the cardinality of the state space S, |S|?**
- **Q: Where are the random variables $X = (X_1, X_2, \ldots X_T)$?**
- **Q: What is the value of T?**

Markov Models as FSA

- One can think Markov models as non-deterministic (or stochastic) finite state automations (FSA) with fixed state transition probabilities.

- Consider the following state space

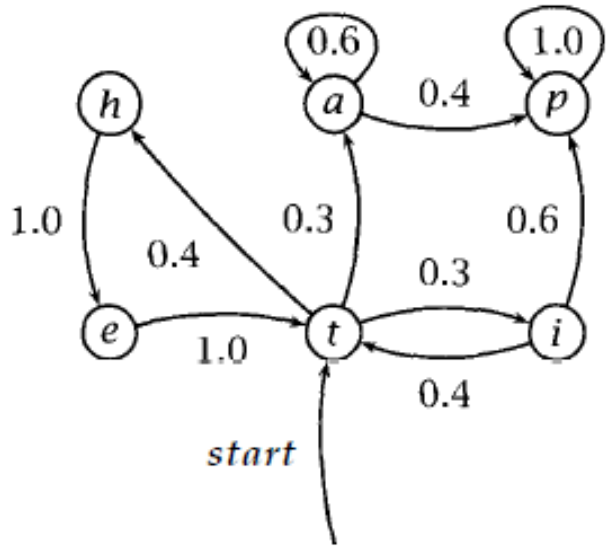


- **Q: What is the cardinality of the state space S, |S|?**
- **Q: Where are the random variables $X = (X_1, X_2, \ldots X_T)$?**
- **Q: What is the value of T?**

Probability of a State Sequence

- How to compute $P(X_1 = t, X_2 = i, X_3 = p)$?

- Recall that for any generic state sequence, we have



$$
\begin{aligned}
P(X_1, \ldots, X_T) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \cdots P(X_T|X_1, \ldots, X_{T-1}) \\
&= P(X_1)P(X_2|X_1)P(X_3|X_2) \cdots P(X_T|X_{T-1}) \\
&= \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}}
\end{aligned}
$$

(using Markov properties):

So, using the Markov model in figure 9.1, we have:

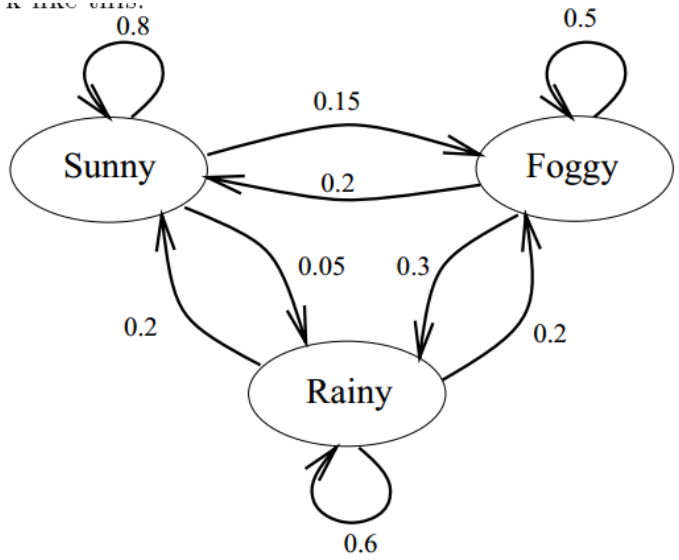
$$
\begin{aligned}
P(t, i, p) &= P(X_1 = t)P(X_2 = i|X_1 = t)P(X_3 = p|X_2 = i) \\
&= 1.0 \text{ x } 0.3 \text{ x } 0.6 \\
&= 0.18
\end{aligned}
$$

Weather Forecasting as a Markov Model

- Problem due to [Lussier, 1998]. Refer [Lussier, 1998] to for details
- Consider the following state transition probabilities for a weather sequence:

		Tomorrow's Weather		
		Sunny	Rainy	Foggy
Today's Weather	Sunny	0.8	0.05	0.15
	Rainy	0.2	0.6	0.2
	Foggy	0.2	0.3	0.5

Table 1: Probabilities of Tomorrow's weather based on Today's Weather

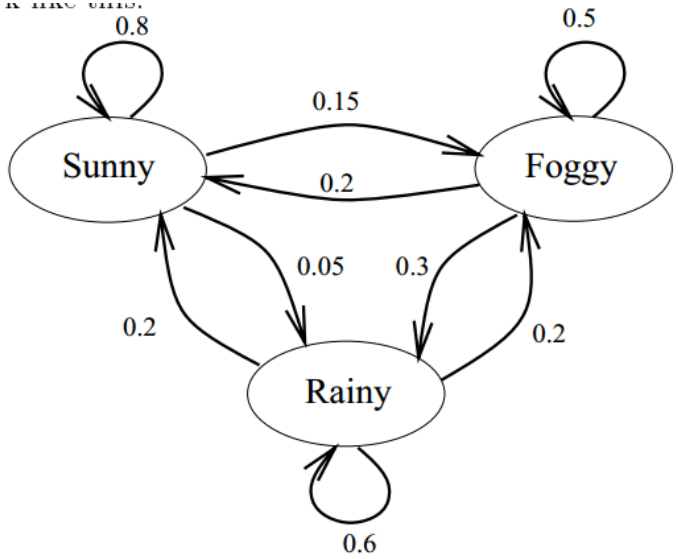


- And the corresponding state space

Weather Forecasting as a Markov Model

- Problem due to [Lussier, 1998]. Refer [Lussier, 1998] to for details

- Problem 1:
  Given that today is sunny what is the probability that tomorrow is sunny and the day after is rainy?



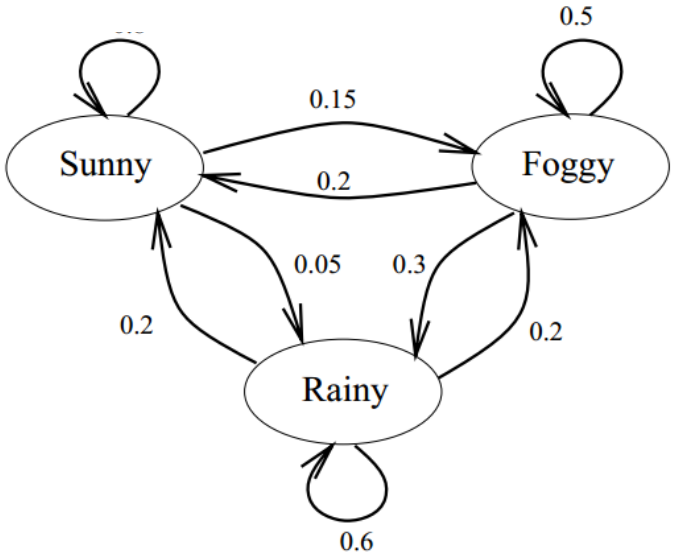
Weather Forecasting as a Markov Model

- Problems due to [Lussier, 1998]. Refer [Lussier, 1998] to for details

- Problem 2:

Given that today is foggy, what's the probability that it will be rainy two days from now?

There are three ways to get from foggy today to rainy two days from now: {foggy, foggy, rainy}, {foggy, rainy, rainy}, and {foggy, sunny, rainy}. Therefore we have to sum over these paths:



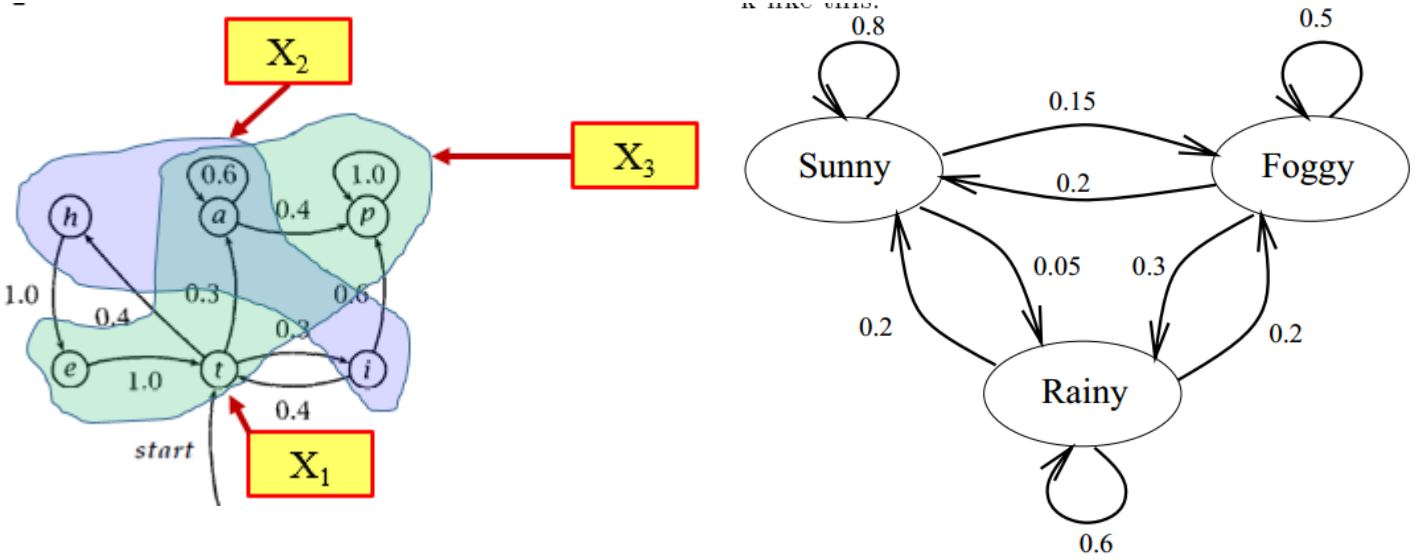
N-gram Models as Markov Models

- N-gram language models (i.e., $P(x_n|x_{n-1}, x_{n-2}, \ldots)$) are an instance of ($n$-1)$^{\text{th}}$ Markov models

> Note that what is important is whether we *can* encode a process as a Markov process, not whether we most naturally do. For example, recall the n-gram word models that we saw in chapter 6. One might think that, for $n \geq 3$, such a model is not a Markov model because it violates the Limited Horizon condition – we are looking a little into earlier history. But we can reformulate any n-gram model as a visible Markov model by simply encoding the appropriate amount of history into the state space (states are then $(n-1)$-grams, for example (was, *walking, down)* would be a state in a **fourgram** model). In general, any fixed finite amount of history can always be encoded in this way by simply elaborating the state space as a crossproduct of multiple previous states. In such cases, we sometimes talk of an m^{th} *order* Markov model, where m is the number of previous states that we are using to predict the next state. Note, thus, that an n-gram model is equivalent to an $(n-1)^{\text{th}}$ order Markov model.

Hidden Markov Model (HMM)

- In Markov models we see the states the machine/FSA is going through



- In an HMM, the state space is hidden/latent.
- But we do get to see a function of the state space, e.g., for every state the machine goes through, the machine outputs/signals some values which are observable.

Hidden Markov Model (HMM)

- Example problem due to [Lussier, 1998]

- Suppose you were locked in a room for several days and you were asked about the weather outside The only piece of evidence you have is whether the person who comes into the room carrying your daily meal is carrying an umbrella or not.

- How is the function (latent/hidden state to observation) coded? i.e., What is $P(u_t|w_t)$?

Let's suppose the following probabilities:

	Probability of Umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

Table 2: Probabilities of Seeing an Umbrella Based on the Weather

- Also what is the prior probability of carrying an umbrella on any day? i.e., $P(u_t)$?

Hidden Markov Model (HMM)

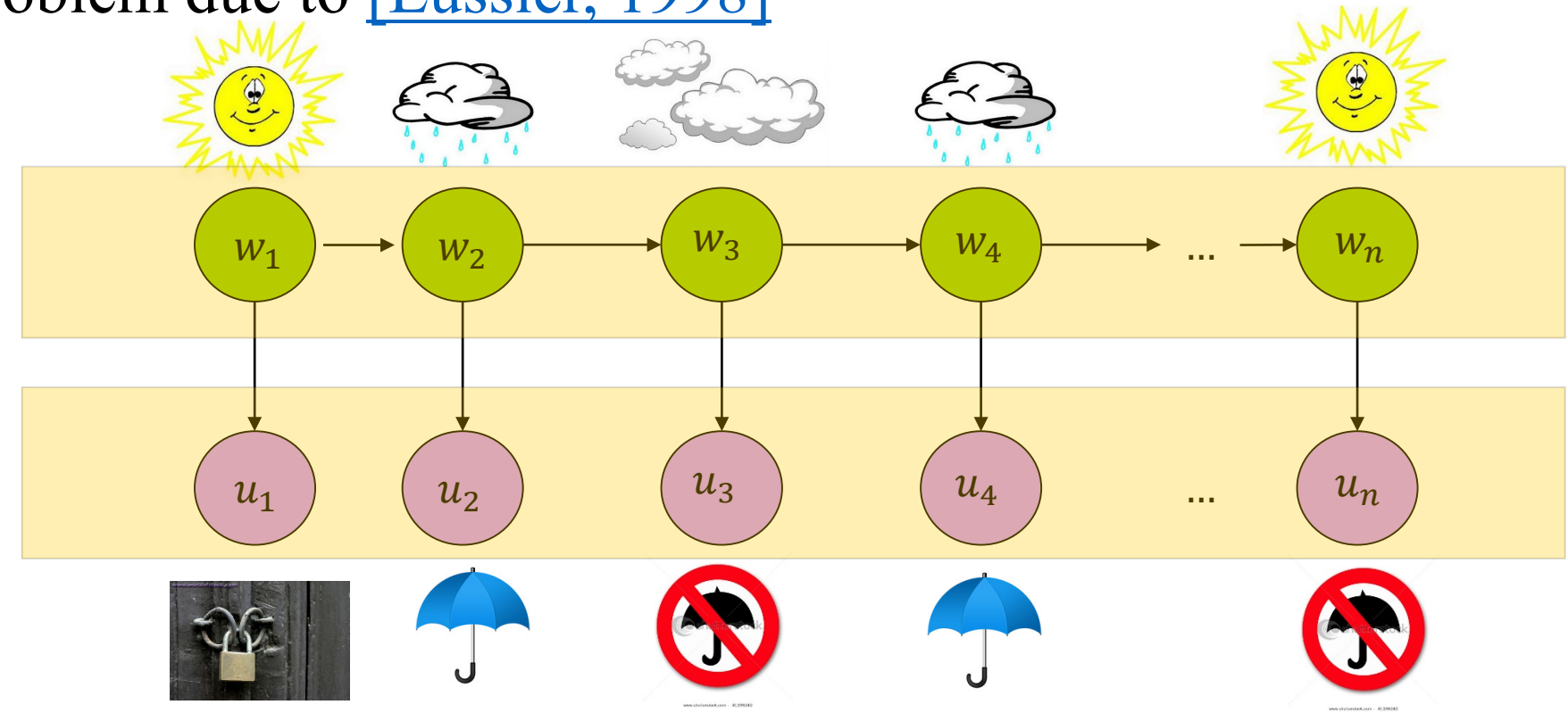
- Example problem due to [Lussier, 1998]

- Suppose the day you were locked in it was sunny. The next day the caretaker carried an umbrella into the room Assuming that the prior probability of the caretaker carrying an umbrella on any day ($P(u_t)$) is 0.5, what is the probability that the second day was rainy?

- i.e., Find $P(w_2 = R\ w_1 = S|u_2 = T)$

Hidden Markov Model (HMM)

- Example problem due to [Lussier, 1998]



- We do not see the states/values of the green nodes as they are hidden.
- We do see the states/values of the purple nodes as they are observable.

Hidden Markov Model (HMM)

- Recall that under a Markov model, we would have computed

$$P(w_1, \ldots, w_n) = \Pi_{i=1}^{n} P(w_i \mid w_{i-1})$$

- But now we need to compute the state sequence given certain observation. We need to use both state transitions $P(w_t|w_{t-1})$ and observation function, $P(u_t|w_t)$. Hence need to use Bayes rules as follows:

$$P(w_1, \ldots, w_n \mid u_1, \ldots, u_n) = \frac{P(u_1, \ldots, u_n|w_1, \ldots, w_n)P(w_1, \ldots, w_n)}{P(u_1, \ldots, u_n)}$$

- Upon using Markov properties and simplifying, we get:

$$P(w_1, \ldots w_n|u_1, \ldots u_n) = \frac{(\prod_{i=1}^{n} P(u_i|w_i)) \times (\prod_{i=1}^{n} P(w_i|w_{i-1}))}{\prod_{i=1}^{n} P(u_i)}$$

Hidden Markov Model (HMM)

- Example problem due to [Lussier, 1998]

- Suppose the (first) day you were locked in it was sunny. The next day the caretaker carried an umbrella into the room Assuming that the prior probability of the caretaker carrying an umbrella on any day ($P(u_t)$) is 0.5, what is the probability that the second day was rainy?

- i.e., Find $P(w_2 = R\ w_1 = S|u_2 = T)$

$$= \frac{P(u_2 = \mathrm{T}|w_2 = \mathrm{Rainy})P(w_2 = \mathrm{Rainy}|w_1 = \mathrm{Sunny})}{P(u_2 = \mathrm{T})}$$

$$= \frac{(0.8)(0.05)}{0.5}$$

$$= .08$$

Hidden Markov Model (HMM)

- Example problem due to [Lussier, 1998]

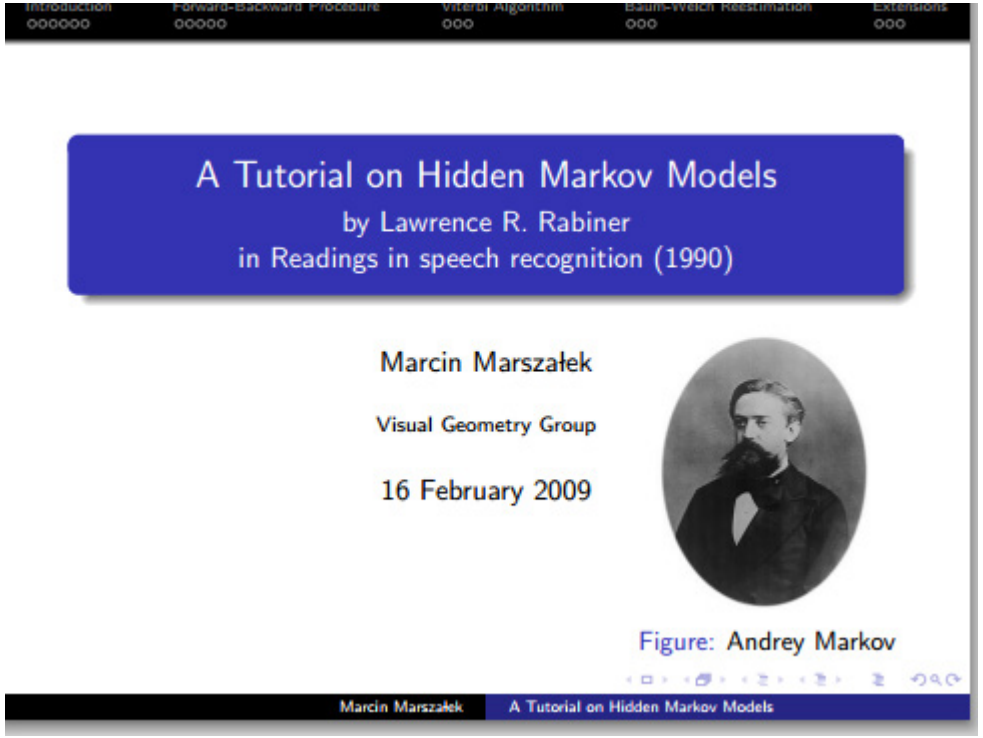
Suppose the day you were locked in the room it was sunny; the caretaker brought in an umbrella on day 2, but not on day 3. Again assuming that the prior probability of the caretaker bringing an umbrella is 0.5, what's the probability that it's foggy on day 3?

- Hint:

$$P(w_3 = \text{F} \mid w_1 = \text{S}, u_2 = \text{T}, u_3 = \text{F}) = P(w_2 = \text{Foggy}, w_3 = \text{Foggy} \mid w_1 = \text{Sunny}, u_2 = \text{True}, u_3 = \text{False}) + P(w_2 = \text{Rainy}, w_3 = \text{Foggy} \mid \ldots) + P(w_2 = \text{Sunny}, w_3 = \text{Foggy} \mid \ldots)$$

HMMs: Decoding, Recognition, Learning

- We refer to slides (4-19) by M. Marszalek



Part-of-Speech (POS) Tagging

- Tagging refers to labeling each word with its corresponding Part-of-Speech
- A big step towards understanding natural language
- Consider the following "tagged" sentence

  The-AT representative-NN put-VBD chairs-NNS on-IN the-AT table-NN.

- The tags are
  AT (Article/Determiner)
  NN/NNS (Singular Noun/Plural Noun)
  VBD (Verb. Past tense)
  IN (Preposition)

Penn Treebank Tag Set

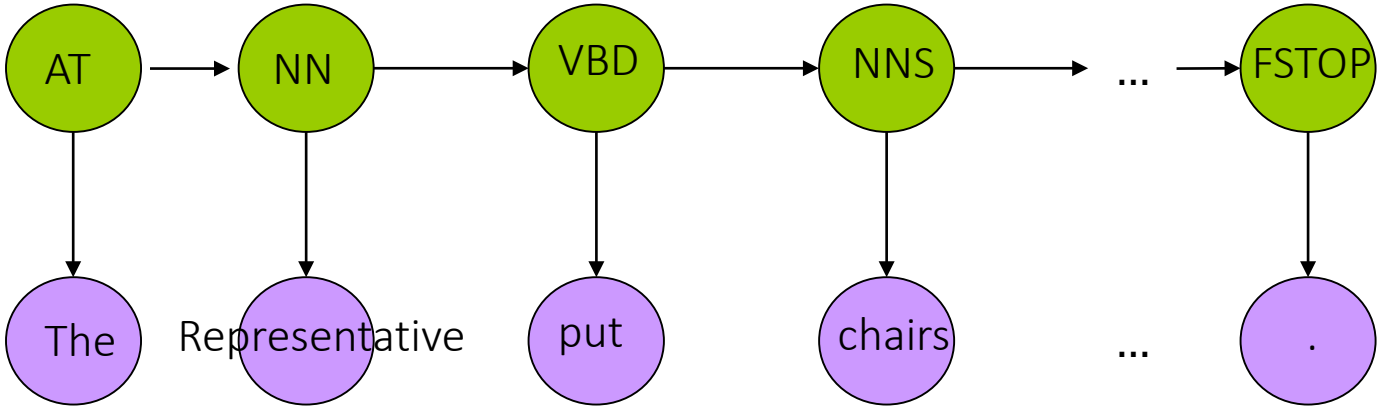
Table 2
The Penn Treebank POS tagset.

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	”	Right close double quote

Part-of-Speech (POS) Tagging

- Tagging can be viewed as an instance of HMM

- Access to large text collections (or corpora), how to estimate the POS tags?



- Observed information: words in sequence
- Hidden/Latent information (to be estimated):POS tags per words
- How to learn?
- Making use of Information sources:
  (1) Tags of other (nearby words), (2) knowing the nearby words itself, (3) sequential patterns and their likelihoods, etc.

Information Sources in Tagging

- Knowing the tags of other nearby words in a context can be helpful

- Given the n-gram (bigram): (w_1, w_2) = "new play", $P(w_2 = NN|w_1 = JJ) > P(w_2 = VBP|w_1 = JJ)$

- However, without context (i.e., if not preceded by "new") play usually acts as a verb, $P(w_2 = VBP) > P(w_2 = NN)$

- Similarly, in English usage we find some patterns more frequent than others:
- P(AT-JJ-NN)>P(AT-JJ-VBP)

- Such a syntax rule based tagger can tag about 77% of tags correctly [Greene and Rubin, 1971] **Q: Does that mean POS tagging is easy? 23% err on what?**

Ambiguity in Tagging

- Tagging using Stanford Parser

- I/PRP **like/VBP** candy/NN ./.
  Time/NNP flies/VBZ **like/JJ** and/CC arrow/JJ ./

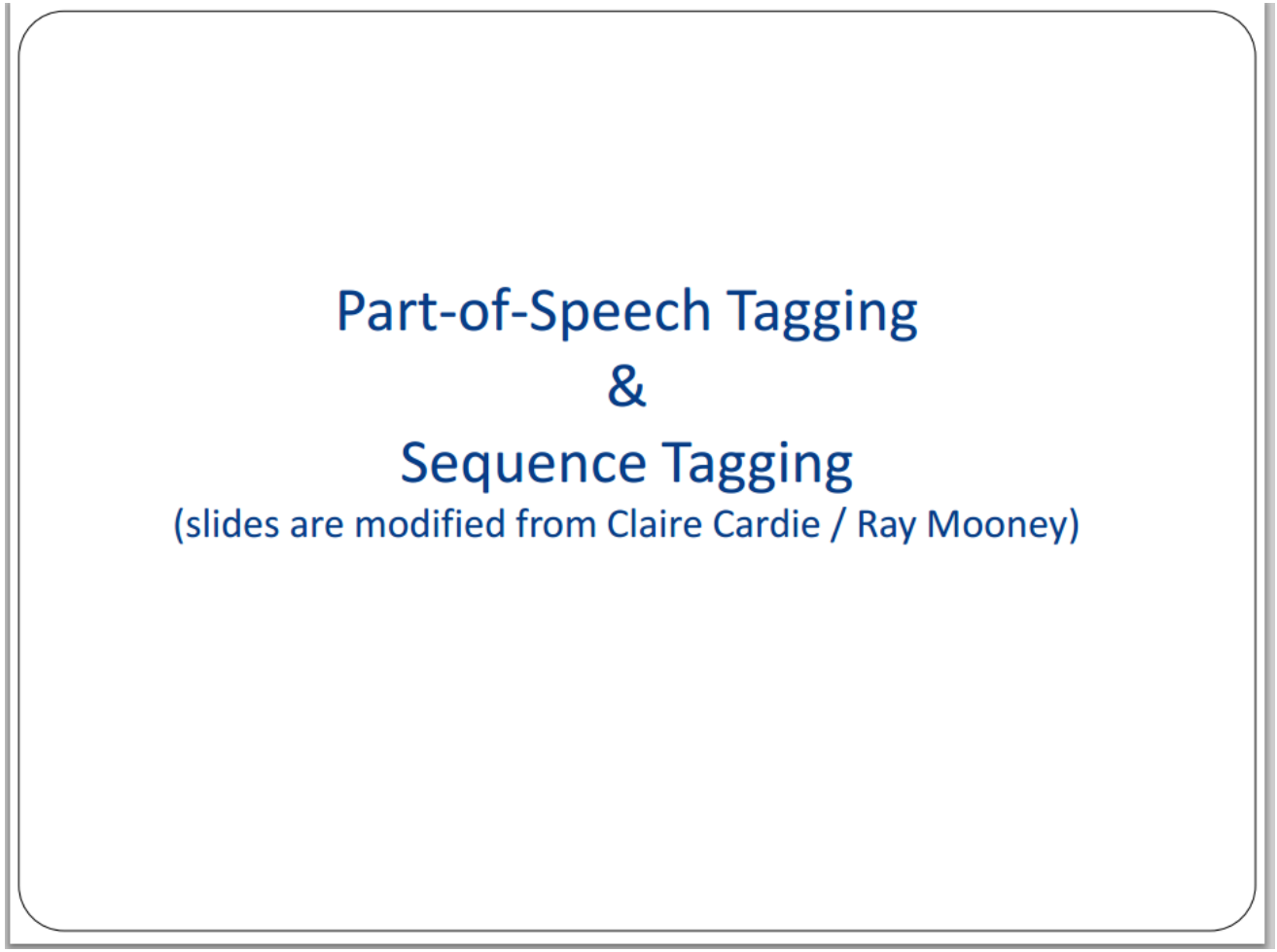
- Next/RB ,/, you/PRP **flour/VBP** the/DT pan/NN ./.
  Need/VBN to/TO buy/VB **flour/NN** for/IN cake/NN ./.

- The/DT world/NN wide/JJ **web/NN** is/VBZ an/DT integral/JJ part/NN of/IN human/JJ lives/NNS ./.
  I/PRP need/VBP you/PRP to/TO **web/VB** our/PRP$ annual/JJ report/NN ./.

- Knowing nearby words also gives us clues beyond just tags of previous words.

Implementing POS Taggers

- We refer to slides (17-36) by Y.Choi

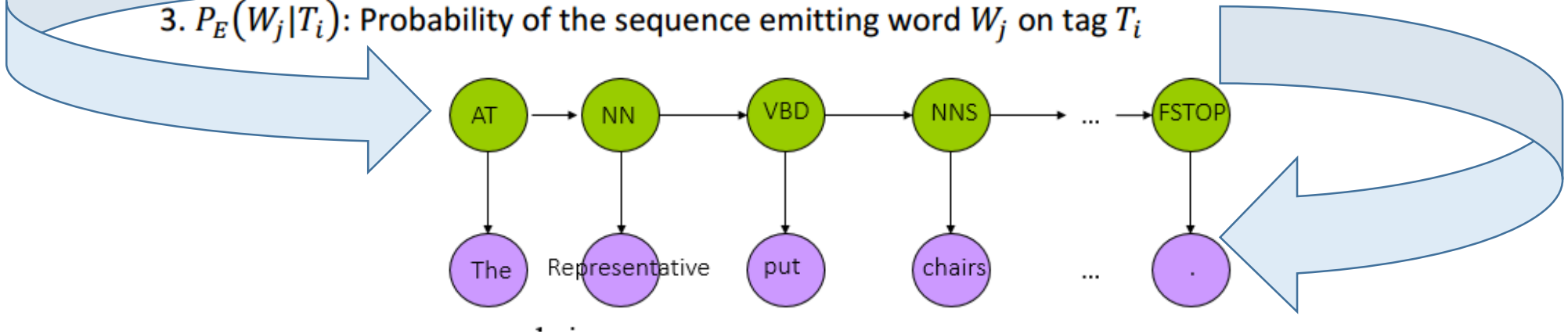


POS Tagging using HMM: Formalisms

- A direct modeling using HMM

In the specific case of our Part of Speech Tagger, the tags are assumed to be the states and the words are assumed to be the outputs. Hence, our Part of Speech Tagger consists of:

1. $P_S(T_i)$: Probability of the sequence starting in tag T_i
2. $P_T(T_j|T_i)$: Probability of the sequence transitioning from tag T_i to tag T_j
3. $P_E(W_j|T_i)$: Probability of the sequence emitting word W_j on tag T_i



- Tagging is obtained by estimating the most likely sequence of states given observed words

$$T_1 T_2 \dots T_n =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} P(T_1 T_2 \dots T_n | W_1 W_2 \dots W_n) =$$

POS Tagging using HMM: Formalisms

- A direct estimation

$$T_1 T_2 \dots T_n =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} P(T_1 T_2 \dots T_n | W_1 W_2 \dots W_n) =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} P(W_1 W_2 \dots W_n | T_1 T_2 \dots T_n) \frac{P(T_1 T_2 \dots T_n)}{P(W_1 W_2 \dots W_n)} =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} P(W_1 W_2 \dots W_n | T_1 T_2 \dots T_n) P(T_1 T_2 \dots T_n) =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} \prod_{i=1}^{n} P(W_i | T_i) \prod_{i=2}^{n} P(T_i | T_{i-1}) \, P(T_1)$$

- But this is impractical. Why?

POS Tagging using HMM: Formalisms

- A direct estimation

$$T_1 T_2 \dots T_n =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} P(T_1 T_2 \dots T_n | W_1 W_2 \dots W_n) =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} P(W_1 W_2 \dots W_n | T_1 T_2 \dots T_n) \frac{P(T_1 T_2 \dots T_n)}{P(W_1 W_2 \dots W_n)} =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} P(W_1 W_2 \dots W_n | T_1 T_2 \dots T_n) P(T_1 T_2 \dots T_n) =$$

$$\underset{T_1 T_2 \dots T_n}{argmax} \prod_{i=1}^{n} P(W_i | T_i) \prod_{i=2}^{n} P(T_i | T_{i-1}) \, P(T_1)$$

- But this is impractical. Why?

Suppose that our corpus is a k-tag Treebank with tags $t_1, t_2, \dots, t_k$ and m words $w_1, w_2, \dots, w_m$ in the dictionary. If we compute the most likely sequence of n tags by enumerating all possible sequence of tags, then the running time of our algorithm is $O(k^n)$. This is clearly very inefficient

POS Tagging using HMM: Details

- Using Viterbi Algorithm (Dynamic Programming):

Suppose that our corpus is a k-tag Treebank with tags $t_1, t_2, \dots, t_k$ and m words $w_1, w_2, \dots, w_m$ in the dictionary. Let $P[r,s]$ for $1 \le r \le n, 1 \le s \le k$ be the greatest probability among all probabilities of sequence of tags $T_1 T_2 \dots T_r$ with $T_r = t_s$. Let $L[r,s]$ for $1 \le r \le n, 1 \le s \le k$ be the sequence of tags $T_1 T_2 \dots T_r$ with $T_r = t_s$ corresponding to that probability. Then, the Viterbi Algorithm for our Part of Speech Tagger can be described as follows:

1. Set $P[1,s] = P(W_1 = w_i | T_1 = t_s) P(T_1 = t_s)$ for $1 \le s \le k$

2. Set $L[1,s] = \{t_s\}$ for $1 \le s \le k$

3. Set $P[r,s] = \max_{1 \le j \le k} P[r-1,j] P(W_r = w_i | T_r = t_s) P\left(T_r = t_s | T_{r-1} = t_j\right)$ for $2 \le r \le n$ and $1 \le s \le k$

4. Set $L[r,s] = \{L[r-1, \operatorname{argmax}_{1 \le j \le k} P[r-1,j] P(W_r = w_i | T_r = t_s) P\left(T_r = t_s | T_{r-1} = t_j\right)], t_l\}$ for $2 \le r \le n$ and $1 \le s \le k$

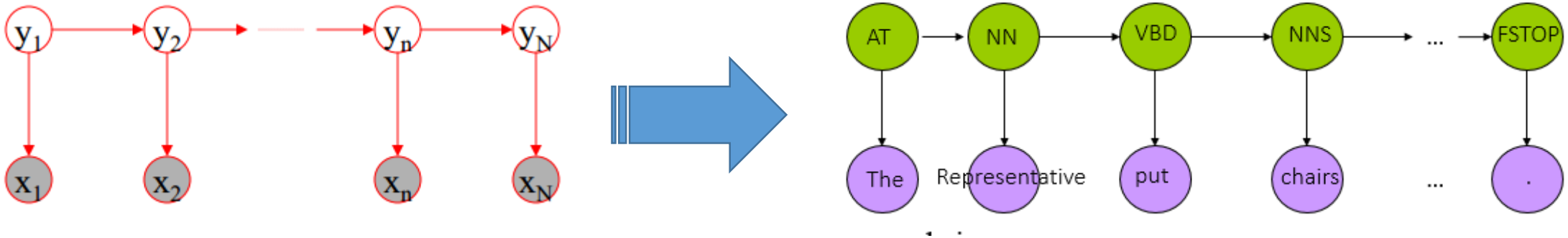
5. Then the most likely sequence of tags is given by $L[n, \operatorname{argmax}_{1 \le j \le k} P[n,j]]$

It is easy to see that the running time of the Viterbi Algorithm for our Part of Speech Tagger is $O(nk^2)$ which is much more efficient and consequently, feasible.

- More details on actual implementation, see this draft by S. Parawira.

POS Tagging using HMM: Drawbacks

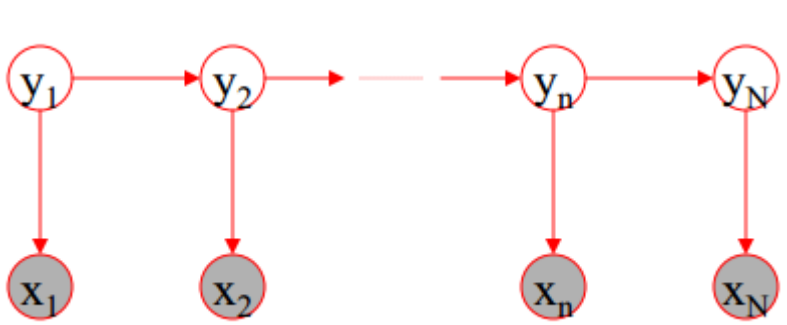
- HMM POS Taggers usually give around 90% tagging accuracy.
- 10% errors are mostly due to the following downsides of HMM modeling:
  - Assumes word depends upon current tag
  - Assumes current tag depends on previous tags only (limited time horizon)
  - Assumes the dependence is static (stationary)
  - Unable to directly use previous words/tags, forthcoming words as features in learning

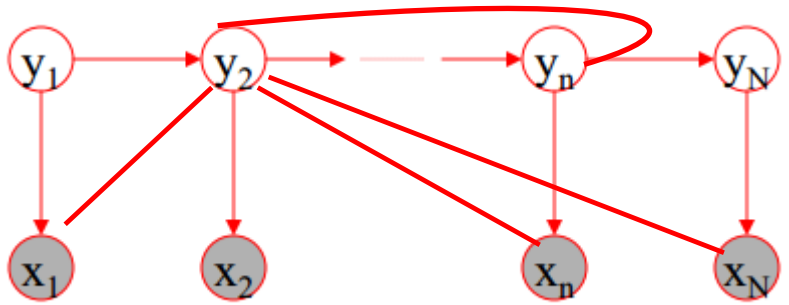


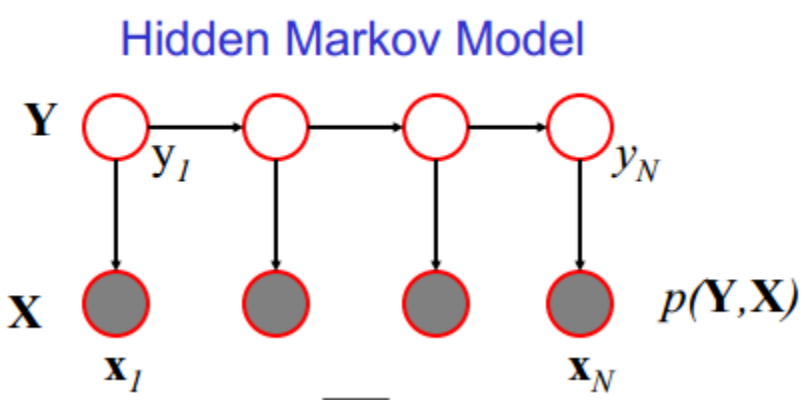
- Way out: Directly model P(**y**|**x**) using Conditional Random Fields (CRFs)

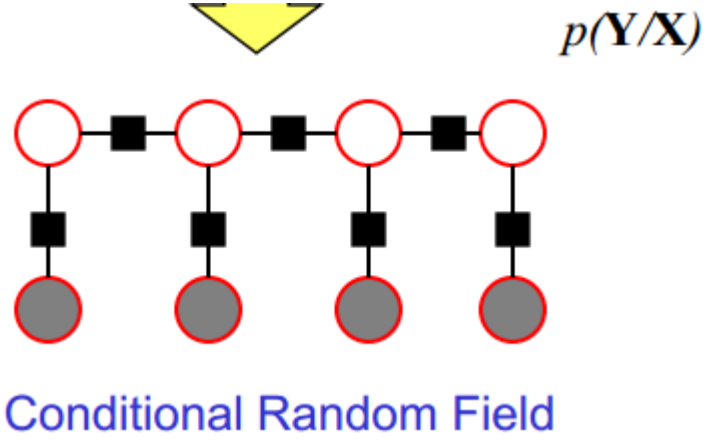
POS Tagging: HMMs to CRFs

- We want to model P(y|x) directly and encode all possible information we can.









- Conditional Random Fields (CRFs) are discriminative counterparts of HMMs

Conditional Random Fields

- Refer to slides (1-12, 16) by A. Quattoni
- This is optional (but useful for projects/research ideas)