Probability and Time: Hidden Markov Models (HMMs)

Computer Science cpsc322, Lecture 32

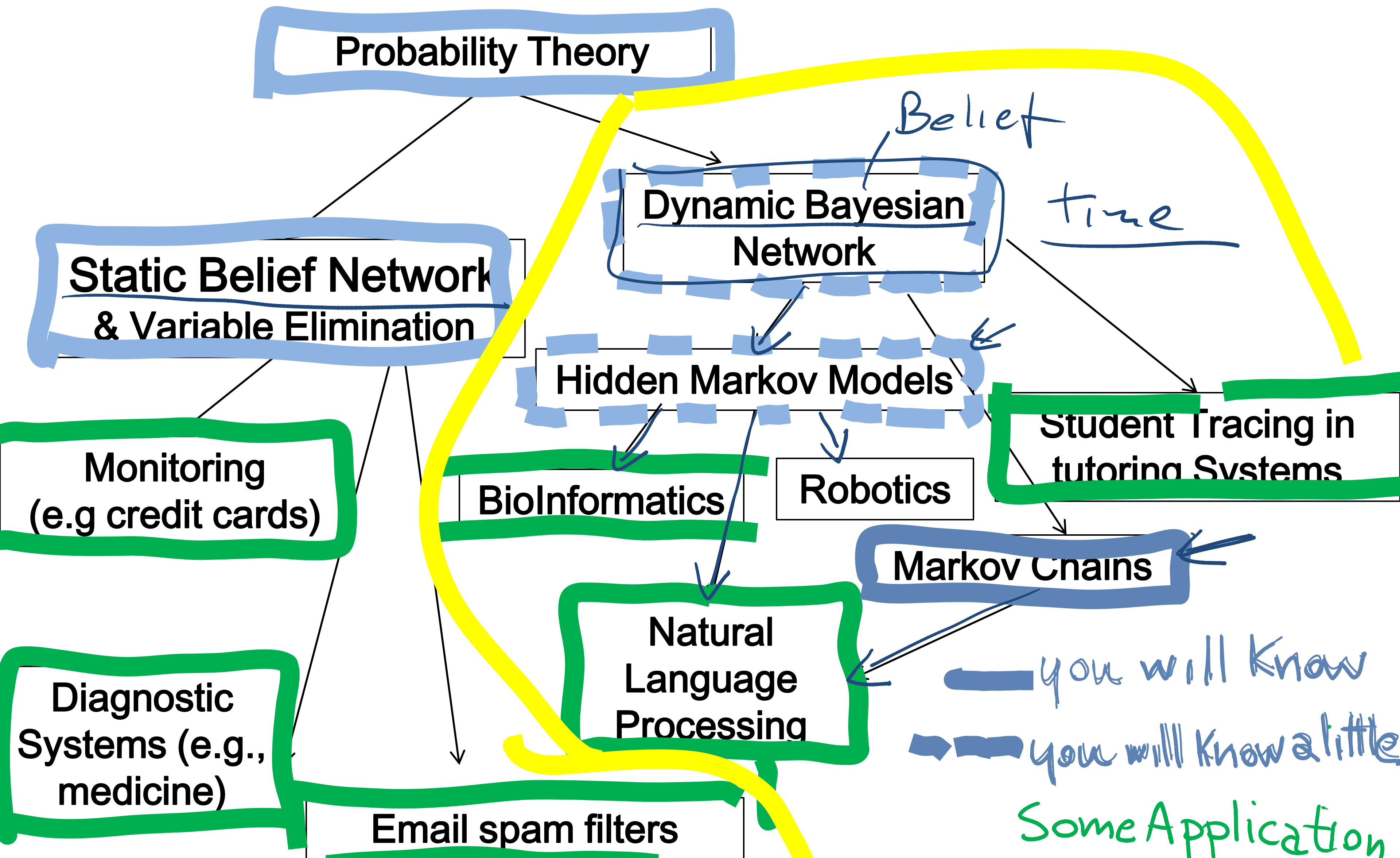
(Textbook Chpt 6.5.2)

April, 7, 2010

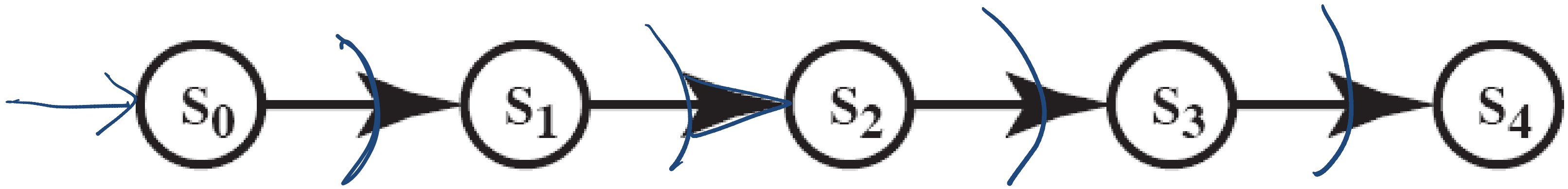
Lecture Overview

- Recap
- Markov Models
  - Markov Chain
  - Hidden Markov Models

Answering Queries under Uncertainty



Stationary Markov Chain (SMC)



$|dom(s_i)| = K$

A stationary Markov Chain : for all t >0

- $P(S_{t+1} | S_0, \ldots, S_t) = P(S_{t+1} | S_t)$ and
- $P(S_{t+1} | S_t)$ the same $\forall t$

We only need to specify $P(S_0)$ K and $P(S_{t+1} | S_t)$

$K \times K$

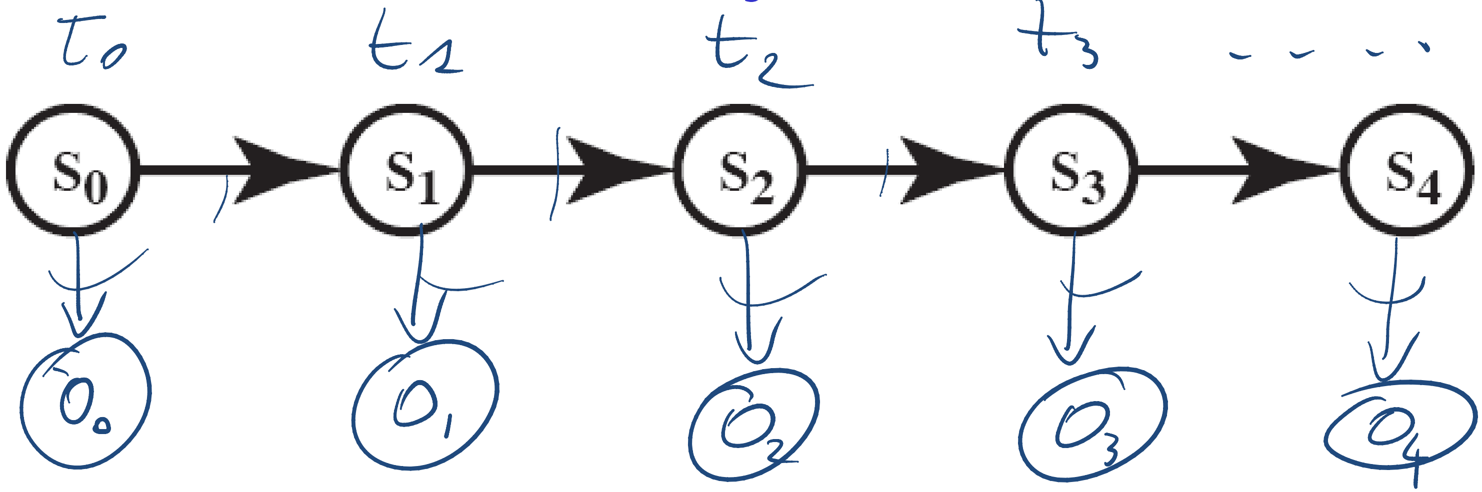
K prob distrib.

- Simple Model, easy to specify
- Often the natural model
- The network can extend indefinitely
- **Variations of SMC are at the core of most Natural Language Processing (NLP) applications!**

Lecture Overview

- Recap
- Markov Models
  - Markov Chain
  - **Hidden Markov Models**

How can we minimally extend Markov Chains?



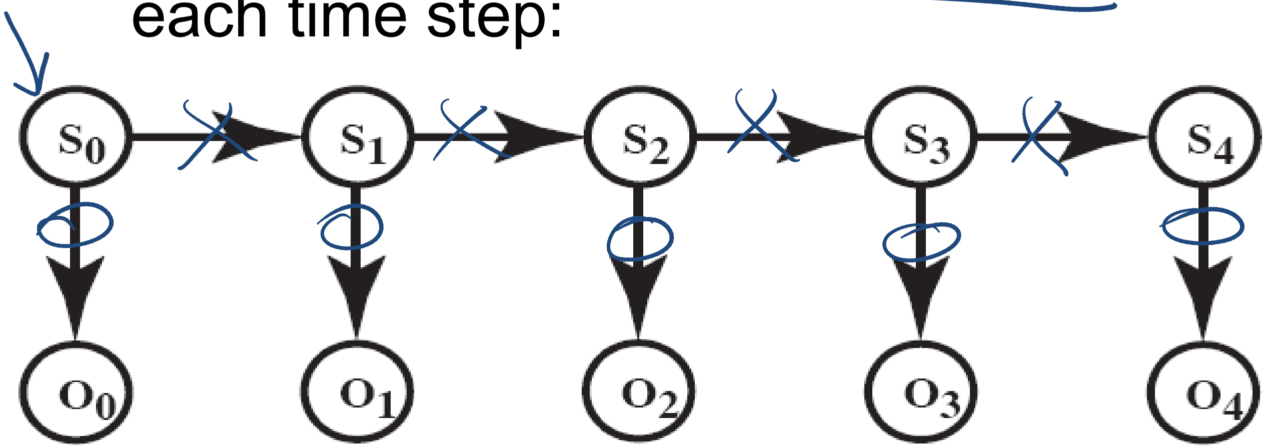
- Maintaining the Markov and stationary assumption?

A useful situation to model is the one in which:

- the reasoning system **does not have access** to the states
- but can **make observations** that give some information about the current state

Hidden Markov Model

- A Hidden Markov Model (HMM) starts with a Markov chain, and adds a noisy observation about the state at each time step:

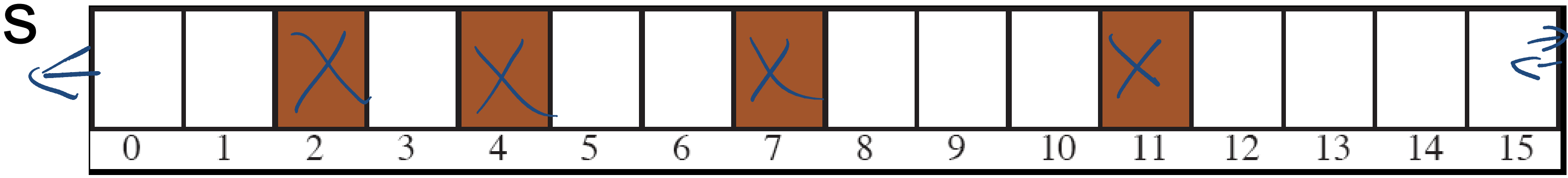


- |domain(S)| = k
- |domain(O)| = h

- $P(S_0)$ specifies initial conditions K
- X $P(S_{t+1}|S_t)$ specifies the dynamics K x K
- O $P(O_t|S_t)$ specifies the sensor model K x h { K prob. dist. over O }

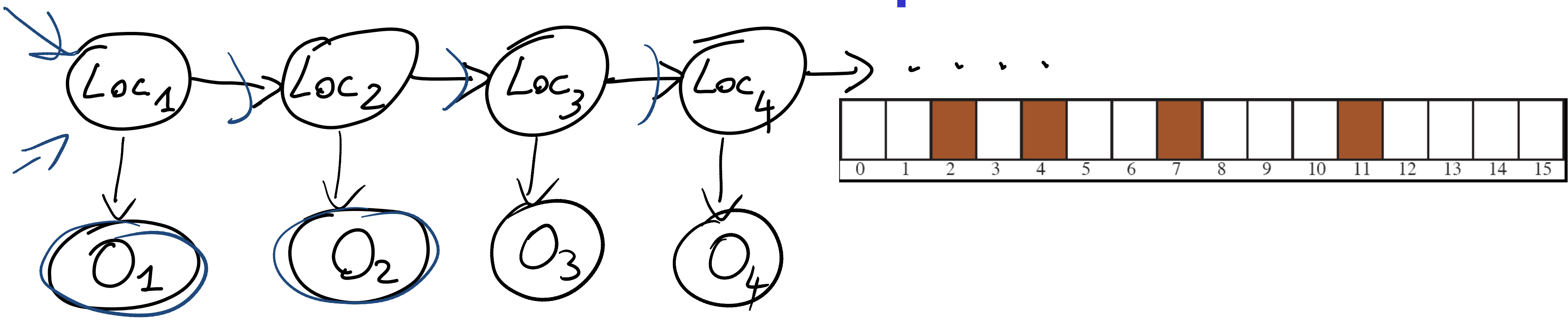
Example: Localization for “Pushed around” Robot

- **Localization** (where am I?) is a fundamental problem in robotics
- Suppose a robot is in a circular corridor with 16 locations



  - There are four doors at positions: 2, 4, 7, 11
  - The Robot initially doesn’t know where it is
  - The Robot is pushed around. After a push it can stay in the same location, move left or right.
  - The Robot has Noisy sensor telling whether it is in front of a door

This scenario can be represented as…



- **Example Stochastic Dynamics**: when pushed, it stays in the same location p=0.2, moves left or right with equal probability .4

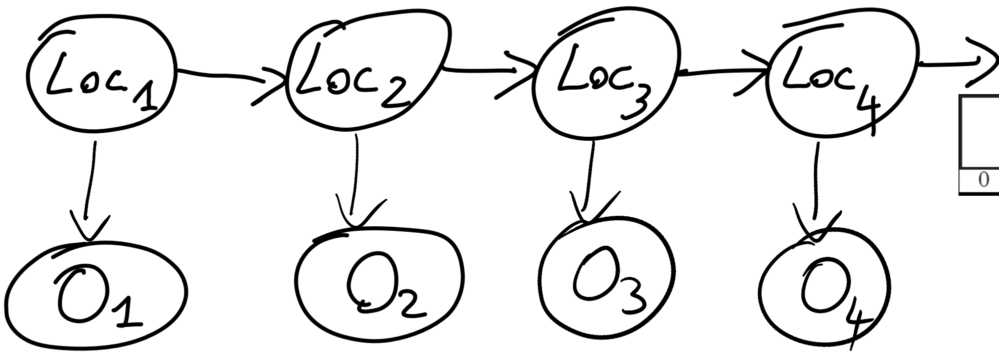
$P(Loc_{t+1} / Loc_t)$

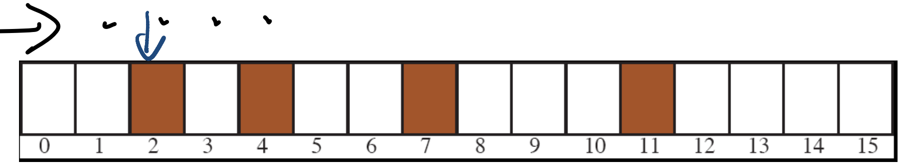
Loc_t \ Loc_{t+1}	0	1	2	....	15
0	.2	.4	0	...0	.4
1	.4	.2	.4	0....	0
2					
3					
⋮					
15					

$P(Loc_1) =$

0	1	2	...	15
1/16	1/16	1/16	...	

This scenario can be represented as…





$dom(Loc_i) = \{0, 1, \ldots, 15\}$

Example of Noisy sensor telling whether it is in front of a door.

WRONG!

- If it is in front of a door $P(O_t = T) = .8$ $\boxed{.2}$
- If not in front of a door $P(O_t = T) = \boxed{.1}$ .9

$\boxed{P(O_t \mid Loc_t)}$

Loc_t	$P(O_t=T)$	$P(O_t=F)$
0	.1	.9
1	.1	.9
2	.8	.2
3	.1	.9
4	.8	.2
⋮		

16 prob. distributions

Useful inference in HMMs

- **Localization**: Robot starts at an unknown location and it is pushed around *t* times. It wants to determine where it is

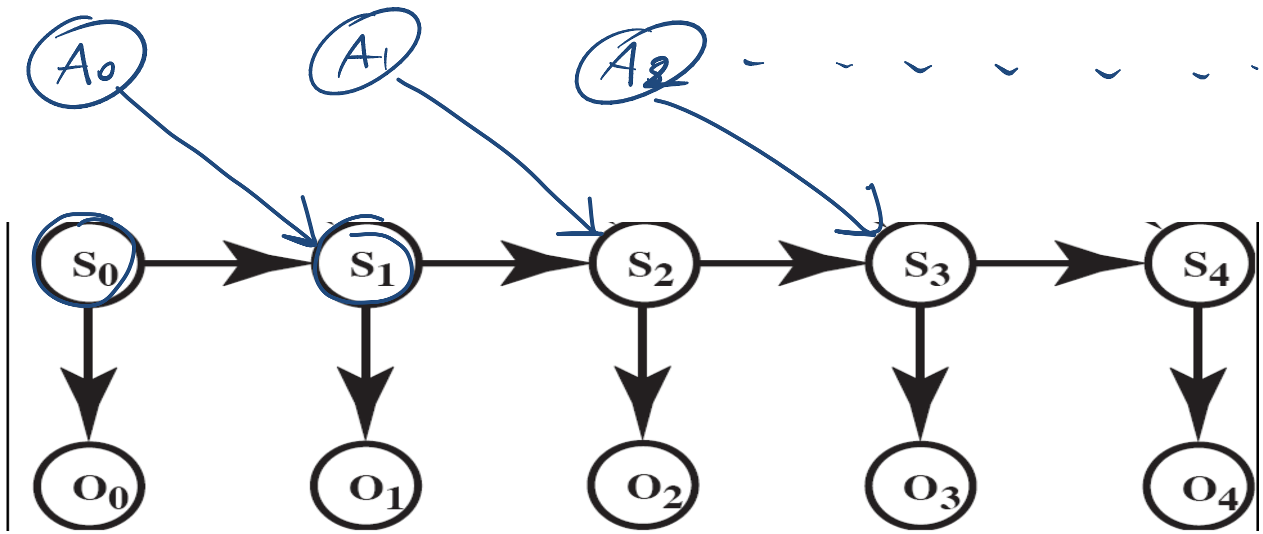
$$P(Loc_t \mid \underline{o_1 \ldots\ldots o_t})$$

- **In general:** compute the posterior distribution over the current state given all evidence to date

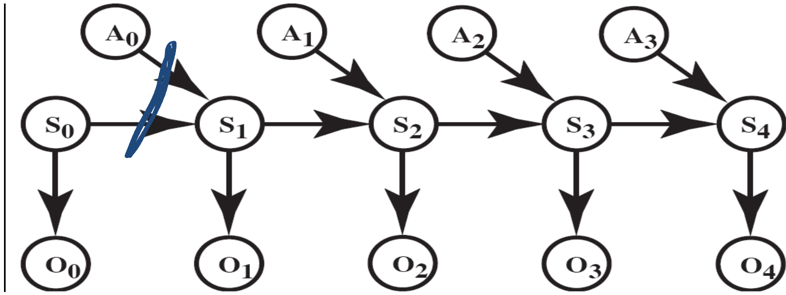
$$P(S_t / O_0 \ldots O_t)$$

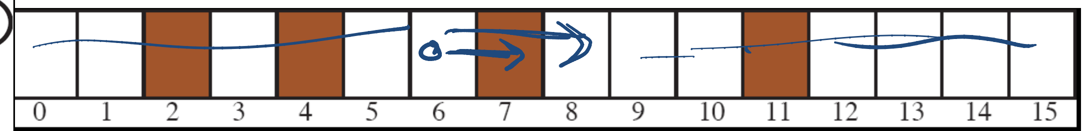
Example : Robot Localization

- Suppose a robot wants to determine its location based on its actions and its sensor readings
- Three actions: *goRight, goLeft, Stay*
- This can be represented by an augmented HMM



Robot Localization Sensor and Dynamics Model





$P(O_t \mid Loc_t)$

- **Sample Sensor Model** (assume same as for pushed around)
- **Sample Stochastic Dynamics:** $P(Loc_{t+1} \mid Action_t, Loc_t)$

$P(Loc_{t+1} = L \mid Action_t = goRight, Loc_t = L) = 0.1$

$P(Loc_{t+1} = L+1 \mid Action_t = goRight, Loc_t = L) = 0.8$

$P(Loc_{t+1} = L+2 \mid Action_t = goRight, Loc_t = L) = 0.074$

$P(Loc_{t+1} = L' \mid Action_t = goRight, Loc_t = L) = \boxed{0.002}$ *for all other locations L'*

x13

- All location arithmetic is modulo 16
- The action *goLeft* works the same but to the left

Dynamics Model More Details

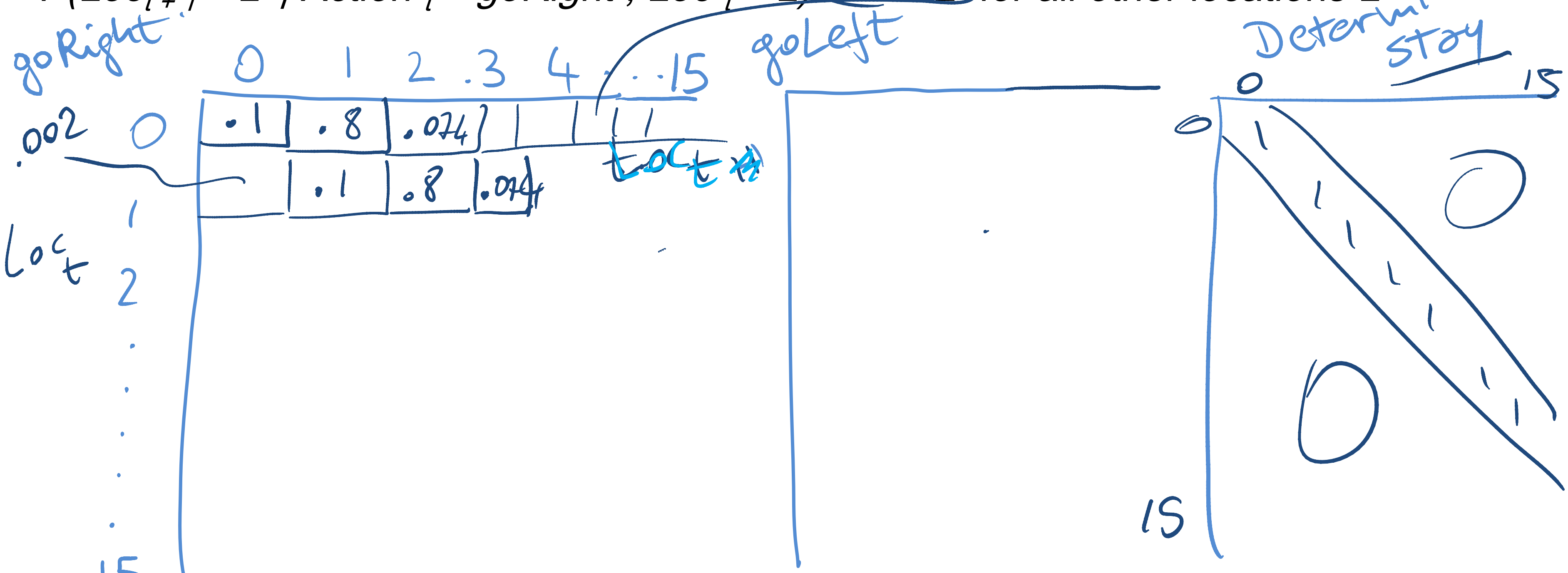
- **Sample Stochastic Dynamics:** $P(Loc_{t+1} \mid Action, Loc_t)$

$P(Loc_{t+1} = L \mid Action_t = goRight, Loc_t = L) = \boxed{0.1}$

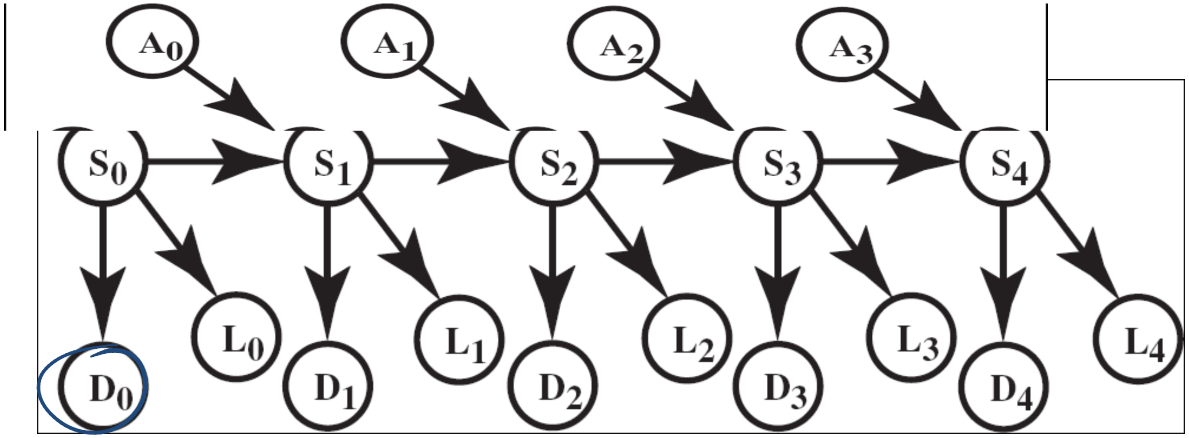
$P(Loc_{t+1} = L+1 \mid Action_t = goRight, Loc_t = L) = 0.8$

$P(Loc_{t+1} = L+2 \mid Action_t = goRight, Loc_t = L) = \underline{0.074}$

$P(Loc_{t+1} = L' \mid Action_t = goRight, Loc_t = L) = 0.002$ for all other locations L'



Robot Localization additional sensor



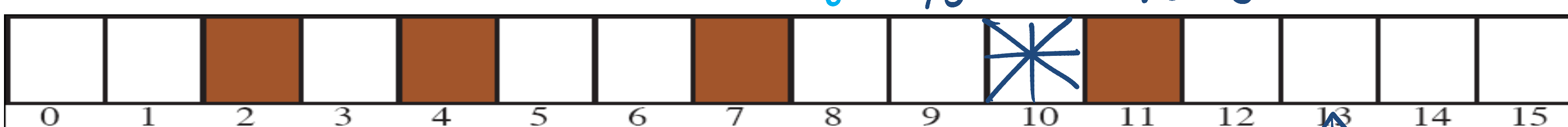
$L_t = T$

the Robot senses light

- **Additional Light Sensor:** there is light coming through an opening at location 10 $P(L_t / Loc_t)$

P(Lt=F) .2 .05 .01 .05 .2 .4

P(Lt=T) .8 .95 .99 .95 .8 .6



- **Info from the two sensors is combined :"Sensor Fusion"**

The Robot starts at an unknown location and must determine where it is

The model appears to be too ambiguous

- Sensors are too noisy
- Dynamics are too stochastic to infer anything

But inference actually works pretty well.

Let's check:

`http://www.cs.ubc.ca/spider/poole/demos/localization/localization.html`

You can use standard Bnet inference. However you typically take advantage of the fact that time moves forward (not in 322)

Sample scenario to explore in demo

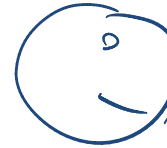
- Keep making observations without moving. What happens?
- Then keep moving without making observations. What happens?
- Assume you are at a certain position alternate moves and observations
- ….

HMMs have many other applications….

Natural Language Processing: e.g., Speech Recognition

- *States:* phoneme \ word

HMM 1 HMM2



mm ⇒ My name is

- *Observations*: acoustic signal \ phoneme

Bioinformatics: Gene Finding

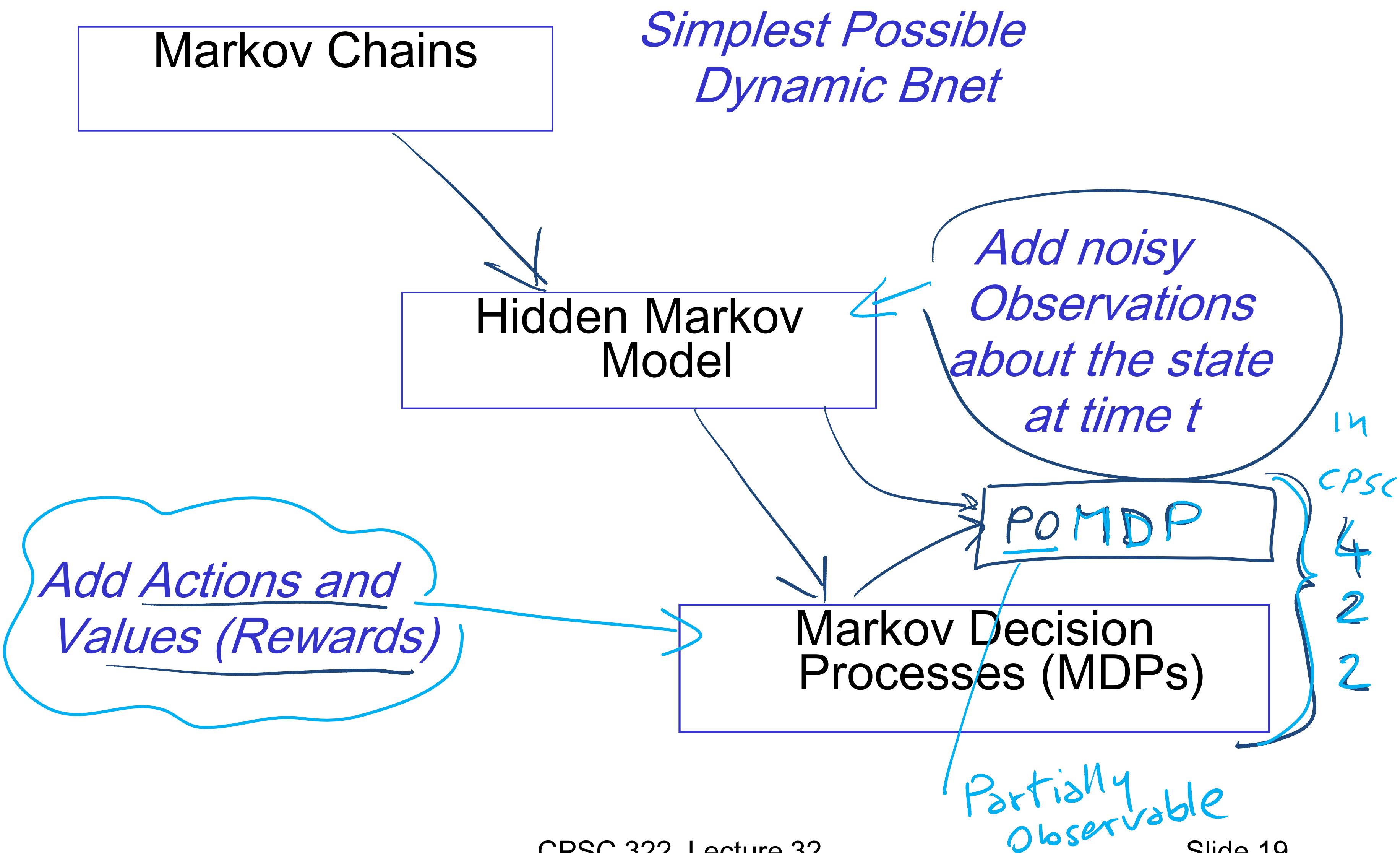
- *States:* coding / non-coding region × × ✓ ✓ ✓ × ×
- *Observations:* DNA Sequences → ATCGGAA

For these problems the critical inference is:

find the most likely sequence of states given a sequence of observations

Viterbi Algo

Markov Models



Learning Goals for today's class

You can:

- Specify the components of an Hidden Markov Model (HMM)
- Justify and apply HMMs to Robot Localization

Clarification on second LG for last class

You can:

- Justify and apply Markov Chains to compute the probability of a Natural Language sentence (NOT to estimate the conditional probs- slide 18)

Next week

Environment

Problem		Deterministic	Stochastic
Static	Constraint Satisfaction	*Vars + Constraints*: Arc Consistency, Search (for CSP), SLS	
Static	Query	*Logics*: Search (for Inference); CSP	*Belief Nets*: Var. Elimination; *Markov Chains and HMMs*
Sequential	Planning	*STRIPS*: Search (for complex planning); CSP	*Decision Nets*: Var. Elimination; ~~*Markov Decision Processes*~~: ~~Value Iteration~~

Representation

Reasoning Technique

Next Class

- **One-off decisions** *(TextBook 9.2)*
- **Single Stage Decision networks** *(9.2.1)*