Probability and Time: Hidden Markov Models (HMMs)

Computer Science cpsc322, Lecture 32

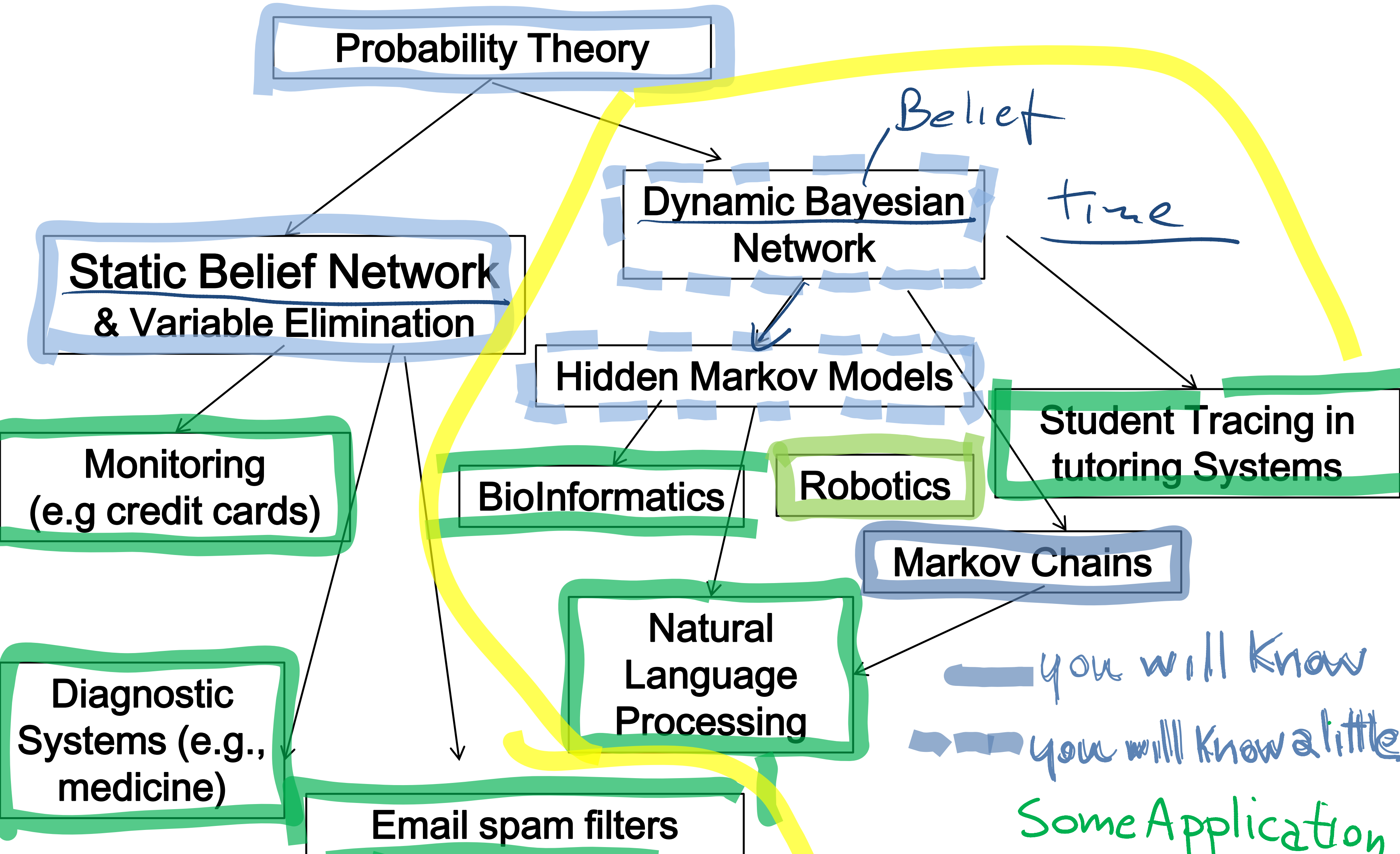
(Textbook Chpt 6.5)

March, 27, 2009

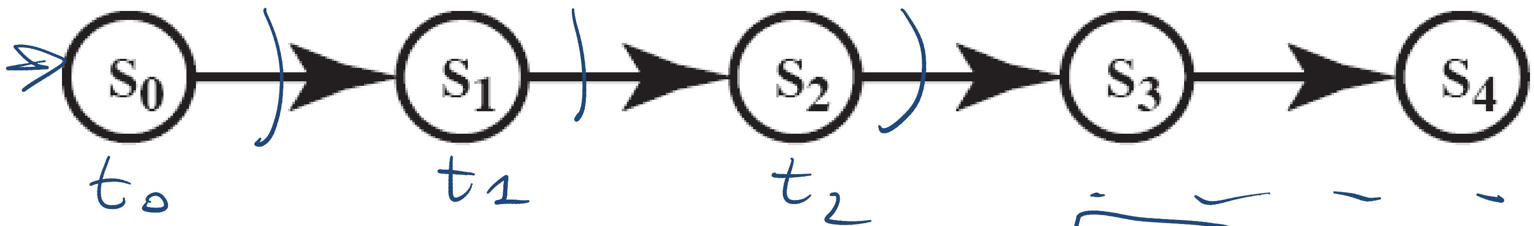
Lecture Overview

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

Answering Queries under Uncertainty



Stationary Markov Chain (SMC)



A stationary Markov Chain : for all $\boxed{t > 0}$ $|dom(S_i)| = K$

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

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

K

$K \times K$

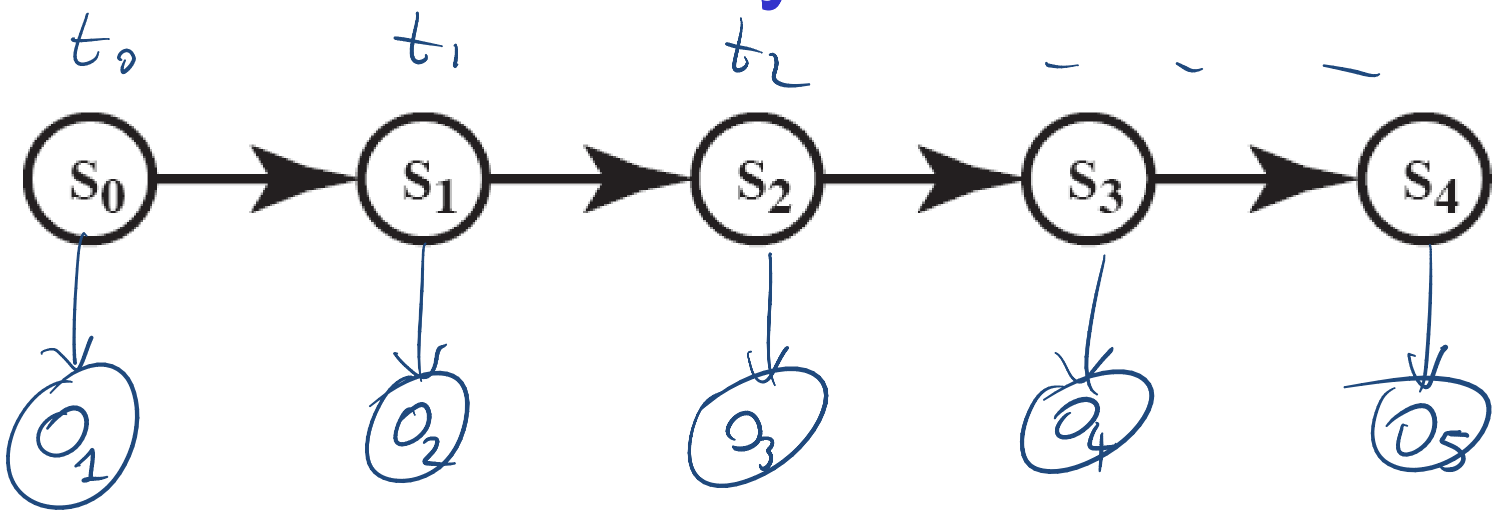
- Simple Model, easy to specify
- Often the natural model
- The network can extend indefinitely
- **Variations of SMC are at the core of most Natural Language Processing (NLP) applications!**

PageRank

Lecture Overview

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

How can we minimally extend Markov Chains?



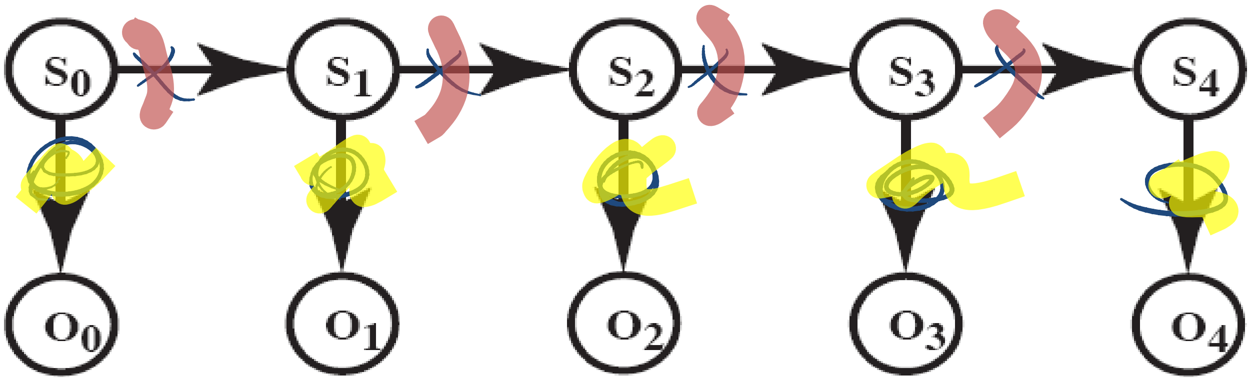
- Maintaining the Markov and stationary assumption?

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:



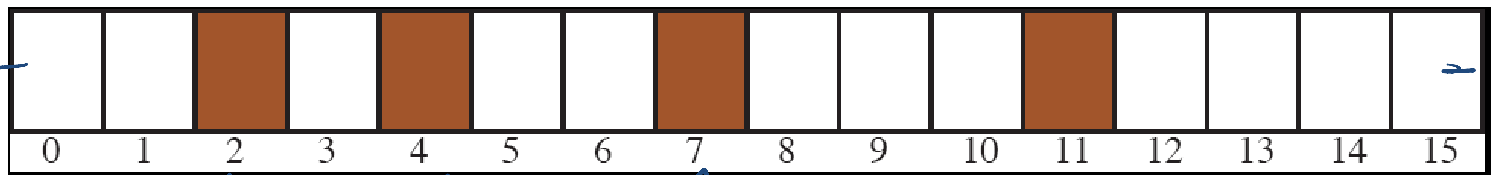
- |domain(S)| = k
- |domain(O)| = h

- $P(S_0)$ specifies initial conditions K
- $P(S_{t+1}|S_t)$ specifies the dynamics K × K
- $\boxed{P(O_t|S_t)}$ specifies the sensor model K × h

K prob distributions

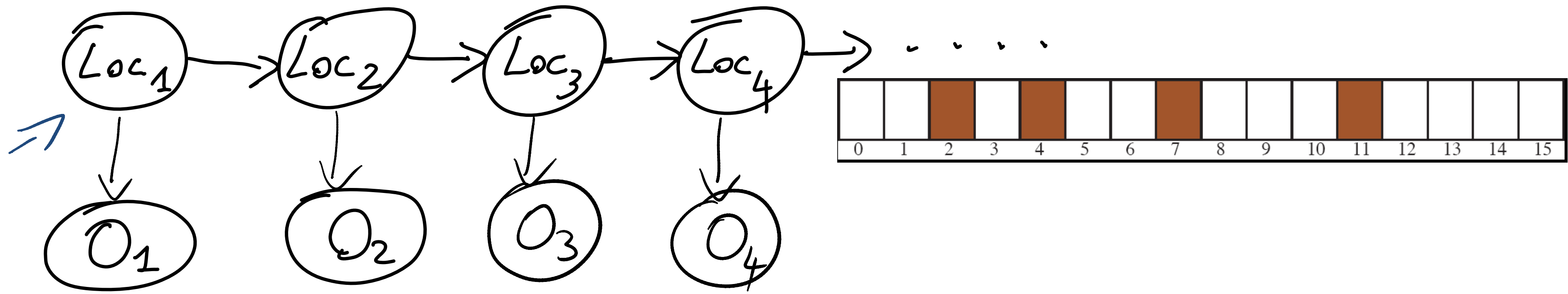
Example: Localization for "Pushed around" Robot

- **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 .4 same location p=0.2, moves left or right with equal probability

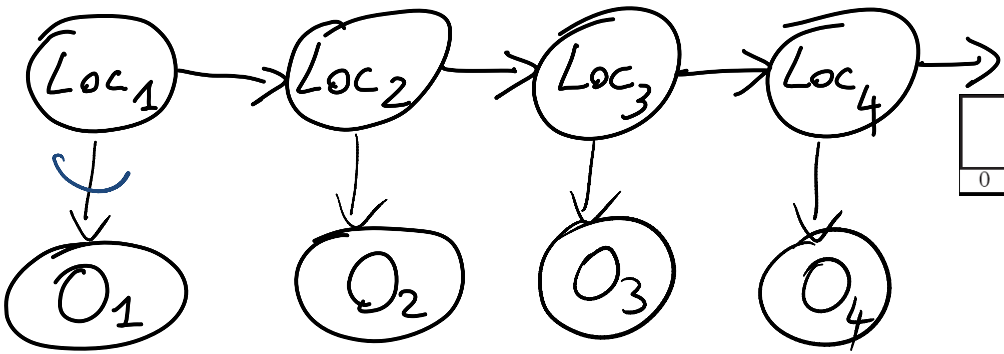
$P(Loc_{t+1} / Loc_t)$

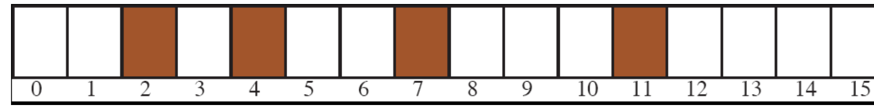
Loc_t \ Loc_{t+1}	0	1	2	3	…	15
0	.2	.4	0	…	…	.4
1	.4	.2	.4	0	0 …	
2						
3						
⋮						
15						

$P(Loc_1) = \frac{1}{16}, \frac{1}{16}, \ldots, \frac{1}{16}$

0 1 2 3 … 15

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.

- If it is in front of a door $P(O_t = T) = .8$
- If not in front of a door $P(O_t = T) = .1$

16 prob. distributions

$P(O_t \mid Loc_t)$

	$P(O_t = T)$	$P(O_t = F)$
0	$\boxed{.1}$	$\boxed{.9}$
1	"	"
2	$\boxed{.8}$	$\boxed{.2}$
3	.1	.9
⋮		
15		

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

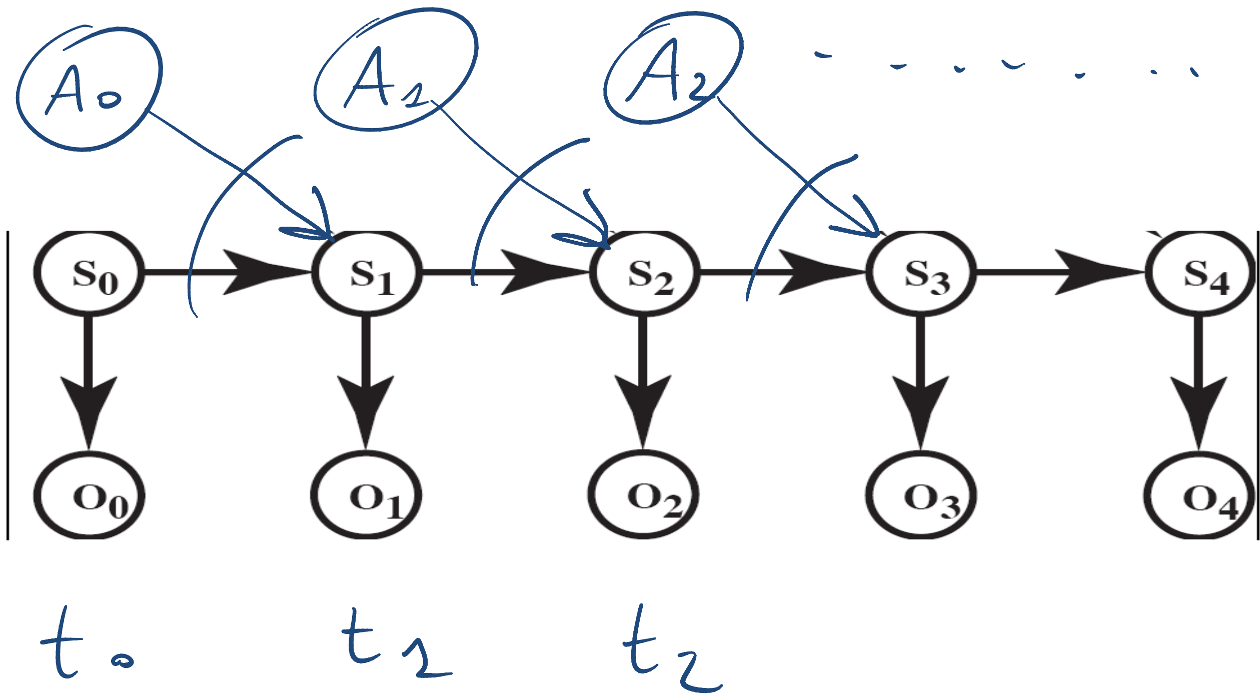
$$P(Loc_t \mid O_1, \ldots, O_t)$$

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

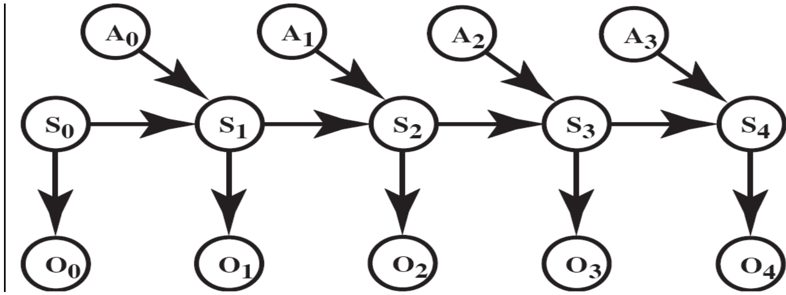
$$P(S_t \,/\, O_0 \ldots O_t)$$

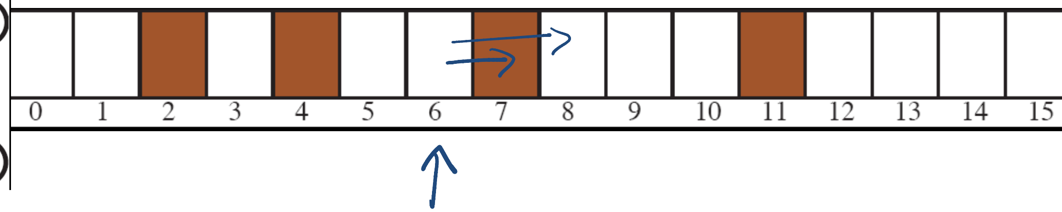
Example : Robot Localization

- 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





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

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

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

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

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

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

Dynamics Model More Details

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

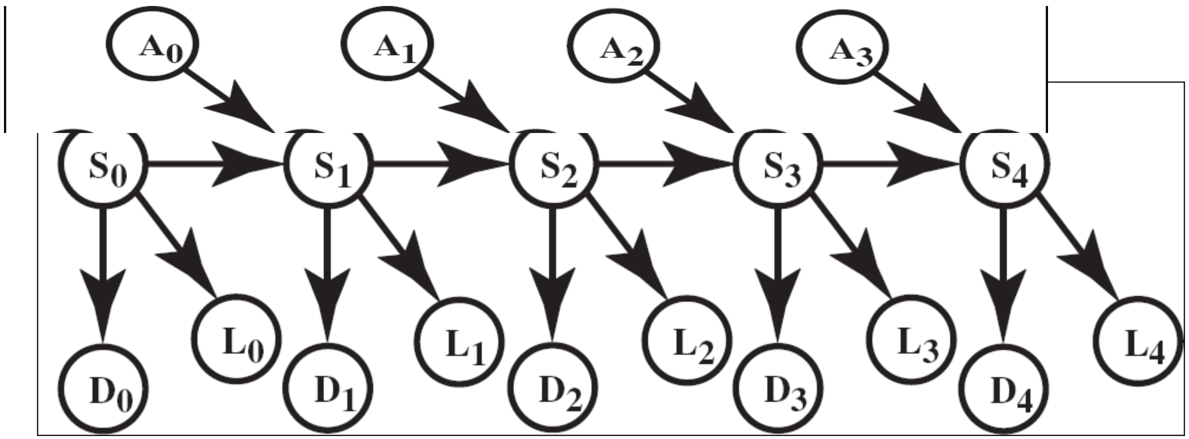
$P(Loc_{t+1} = L \mid Action_t = goRight, Loc_t = L) = 0.1$

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

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

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

Robot Localization additional sensor



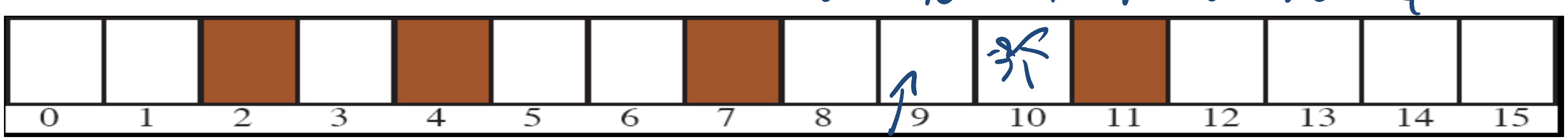
$L_t = T$

the Robot senses light

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

$P(L_t=F)$ .05 .01 .05

$P(L_t=T)$ .8 .95 .99 .95 .8 .6 .4 ....



- **Info from the two sensors is be 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

Do you think inference will work?

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

HMM1 HMM2

- *Observations*: acoustic signal \ phoneme

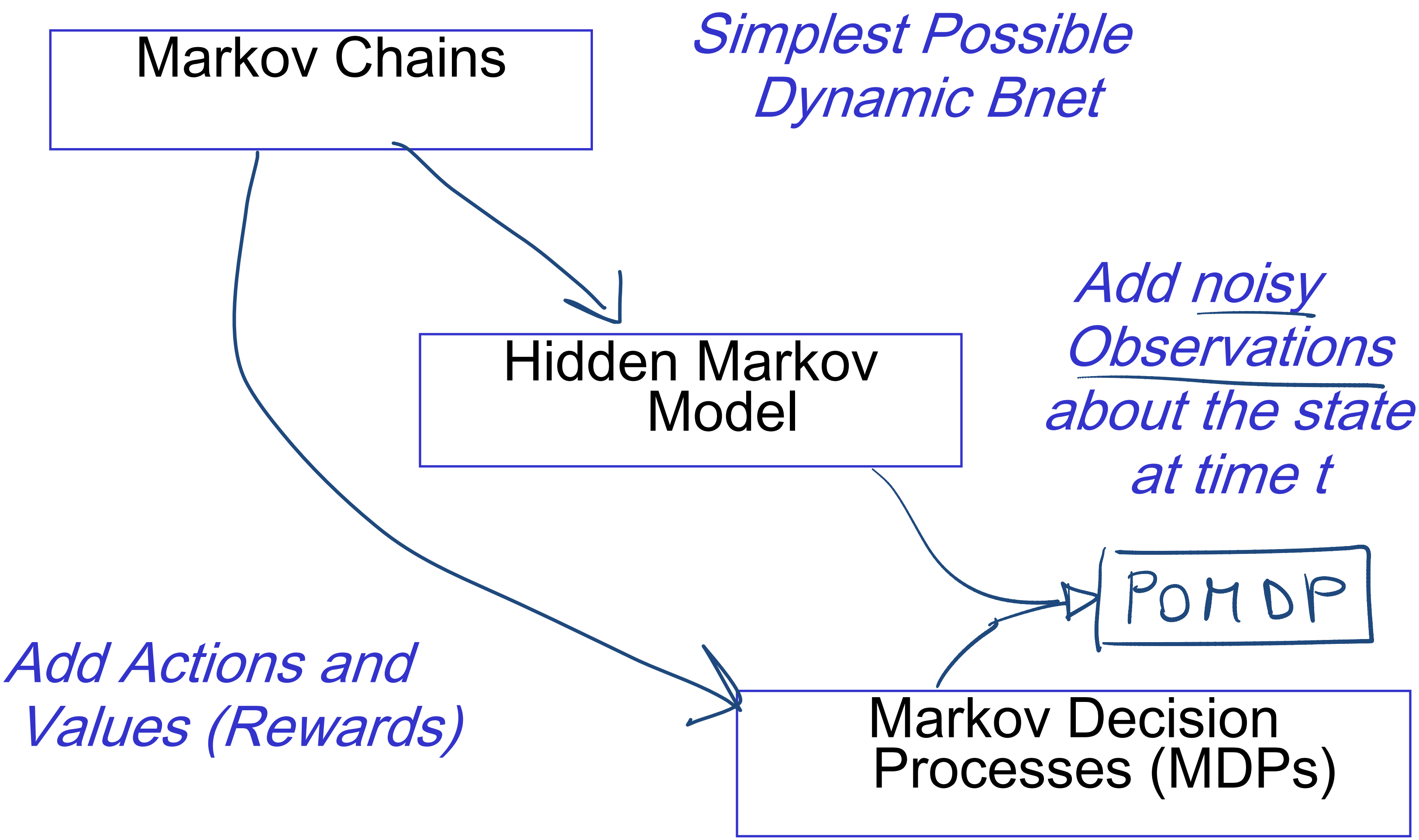
Bioinformatics: Gene Finding

- *States:* coding / non-coding region
- *Observations:* DNA Sequences

For these problems the critical inference is:

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

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

Problem		Environment: Deterministic	Environment: Stochastic
Static	Constraint Satisfaction	*Vars + Constraints* Arc Consistency Search → for CSP SLS	
Static	Query	*Logics* CSP Search → for Inference	*Belief Nets* Var. Elimination *Markov Chains and HMMs*
Sequential	Planning	*STRIPS* → CSP Search → for complex planning	*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)*