Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 6

Sep, 20, 2017

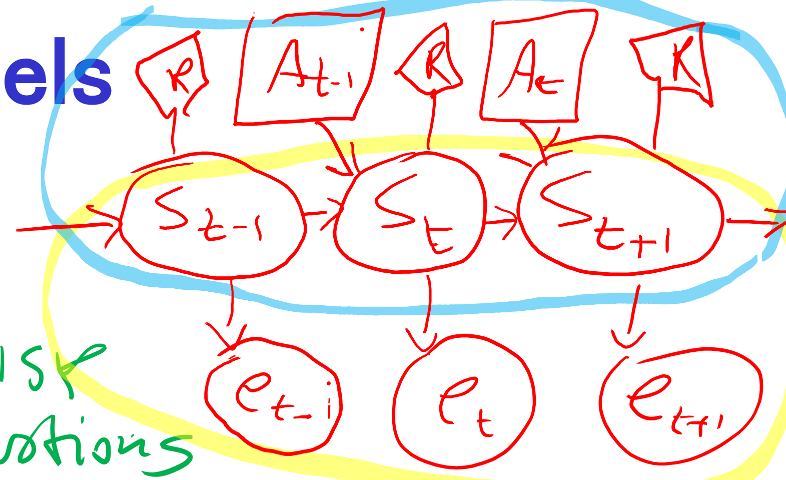
Slide credit POMDP: C. Conati and P. Viswanathan

Lecture Overview

Partially Observable Markov Decision Processes

- Summary
  - Belief State
  - Belief State Update
- Policies and Optimal Policy

Markov Models



Markov Chains

Noisy observations

Hidden Markov Model

Noisy Actions

Rewards

Partially Observable Markov Decision Processes (POMDPs)

Markov Decision Processes (MDPs)

Belief State and its Update

$$b(s)$$

$$b'(s') = \alpha P(e \mid s') \sum_{s} P(s' \mid s, a) b(s)$$

as

$$b' = \mathit{Forward}(b,a,e)$$

- To summarize: when the agent performs action ***a*** in belief state ***b***, and then receives observation ***e***, filtering gives a unique new probability distribution over state
  - ***deterministic transition from one belief state to another***

Belief Update: Example1

- Let's introduce a sensor that perceives the number of adjacent walls in a location with a 0.1 probability of error
  - P(2w|s) = 0.9 ; P(1w|s) = 0.1 if *s* is non-terminal and not in third column
  - P(1w|s) = 0.9 ; P(2w|s) = 0.1 if *s* is non-terminal and in third column
- Try to compute the new belief state if agent **moves *left*** and then **perceives 1 adjacent wall**

0.111	0.111	0.111	$\boxed{0.000}$
0.111		0.111	$\boxed{0.000}$
0.111	0.111	0.111	0.111

$$b'(s') = \alpha \quad P(e \mid s') \quad \sum_{s} P(s' \mid a, s) b(s)$$

$$b'(1,1) = \alpha \;\; X \;\; \left[P((1,1) \mid (1,1), left) b(1,1) + P((1,1) \mid (1,2), left) b(1,2) + P((1,1) \mid (2,1), left) b(2,1) \right]$$

X should be equal to ?

A. 0.1   B. 0.2   C. 0.9

Belief Update: Example 2

- Let's introduce a sensor that perceives the number of adjacent walls in a location with a 0.1 probability of error
  - P(2w|s) = 0.9 ; P(1w|s) = 0.1 if *s* is non-terminal and not in third column
  - P(1w|s) = 0.9 ; P(2w|s) = 0.1 if *s* is non-terminal and in third column
- Try to compute the new belief state if agent **moves *right*** and then **perceives 2 adjacent wall**

0.111	0.111	0.111	$\boxed{0.000}$
0.111		0.111	$\boxed{0.000}$
0.111	0.111	0.111	0.111

$$b'(s') = \alpha \quad P(e \mid s') \quad \sum_{s} P(s' \mid a, s) b(s)$$

$$b'(1,2) = \alpha \;\; P(2w \mid (1,2)) \times$$

$$\begin{bmatrix} P((1,2) \mid (1,1), right) b(1,1) + \\ P((1,2) \mid (1,2), right) b(1,2) + \\ P((1,2) \mid (1,3), right) b(1,3) \end{bmatrix}$$

Optimal Policies in POMDs ?

- Theorem (Astrom, 1965):
  - The optimal policy in a POMDP is a function $\pi^*(b)$ where b is the belief state (probability distribution over states)
- That is, $\pi^*(b)$ is a function from belief states (probability distributions) to actions
  - It does *not* depend on the actual state the agent is in
  - Good, because the agent does not know that, all it knows are its beliefs!
- Decision Cycle for a POMDP agent
  - Given current belief state *b*, execute $a = \pi^*(b)$
  - Receive observation *e*
  - compute : $b'(s') = \alpha P(e \mid s') \sum_{s} P(s' \mid s, a) b(s)$
  - Repeat

How to Find an Optimal Policy?

?

- Turn a POMDP into a corresponding MDP and then solve that MDP
- Generalize VI to work on POMDPs
- Develop Approx. Methods
  - Point-Based VI
  - Look Ahead

Finding the Optimal Policy: State of the Art

- Turn a POMDP into a corresponding MDP and then apply VI: only small models
- Generalize VI to work on POMDPs
  - 10 states in1998
  - 200,000 states in 2008-09
- Develop Approx. Methods 2009 - now
  - Point-Based VI and Look Ahead
  - Even 50,000,000 states
    http://www.cs.uwaterloo.ca/~ppoupart/software.html

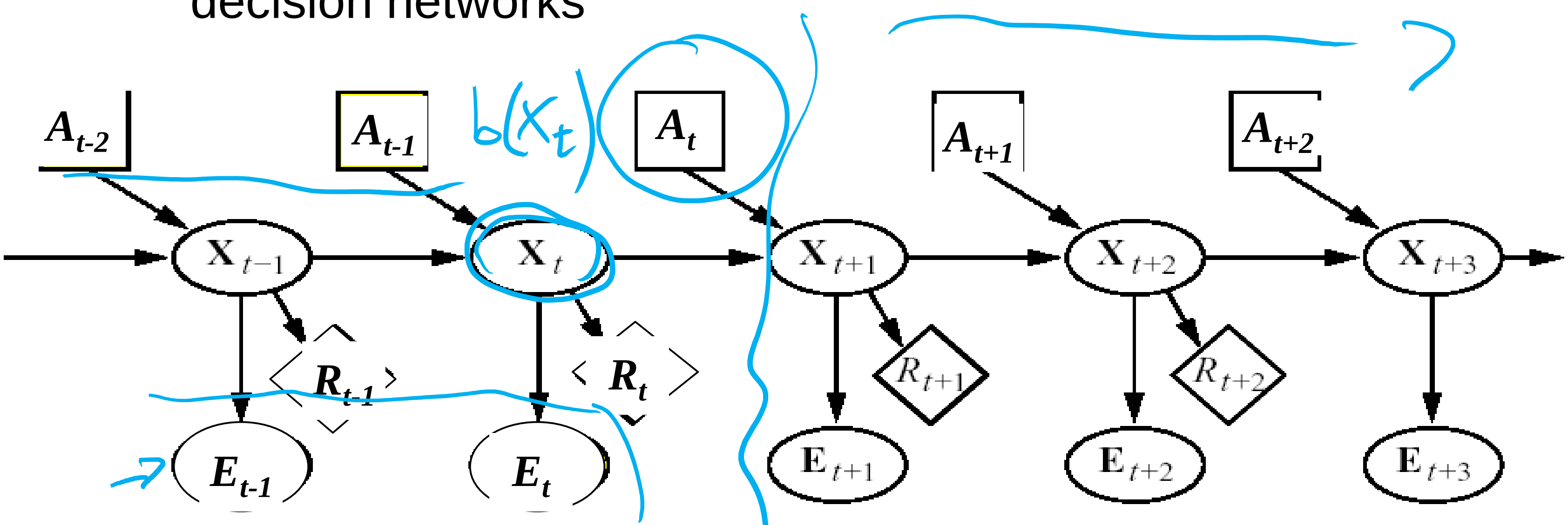
Recent Method: Point-based Value Iteration (not required)

- Find a solution **for a sub-set of all states**
- Not all states are necessarily reachable
- Generalize the solution to all states
- Methods include: PERSEUS, PBVI, and HSVI and other similar approaches (FSVI, PEGASUS)

Dynamic Decision Networks (DDN)

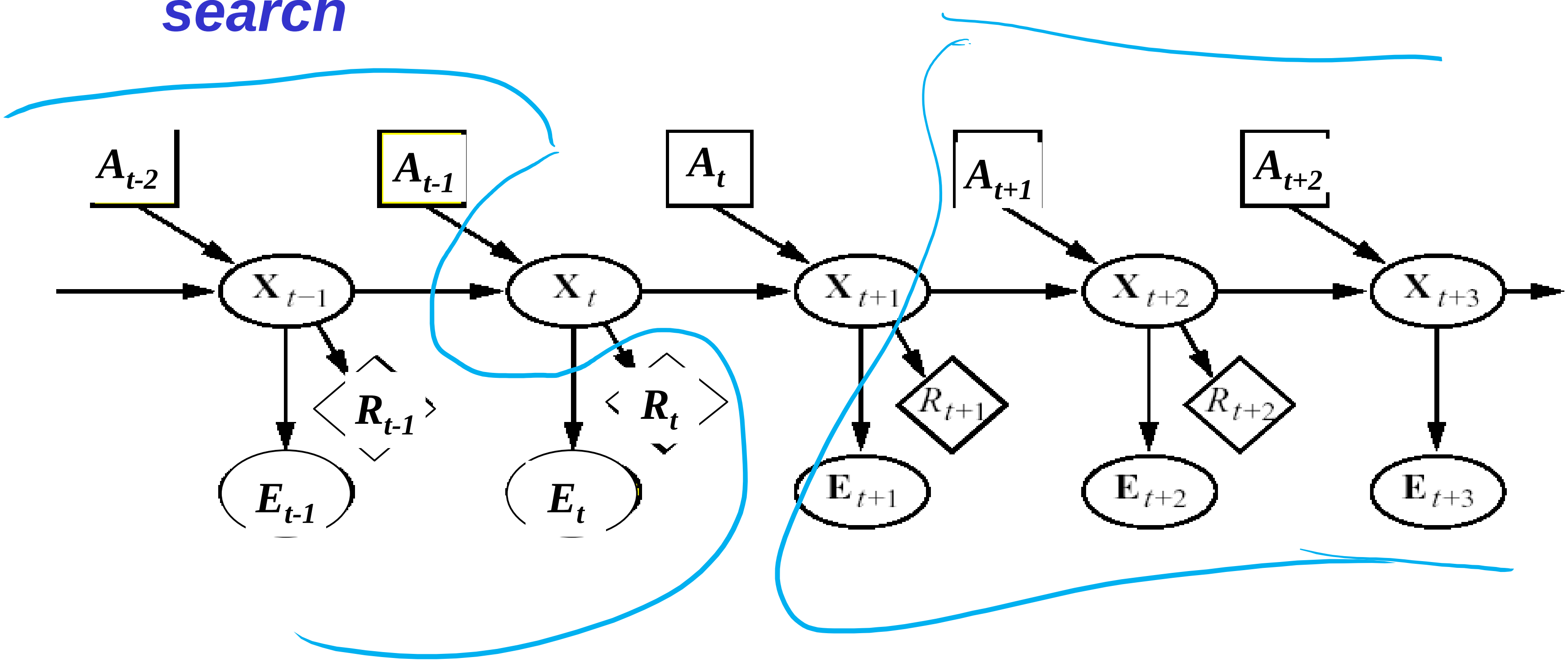
- Comprehensive approach to agent design in partially observable, stochastic environments

- Basic elements of the approach
  - Transition and observation models are represented via a Dynamic Bayesian Network (DBN).
  - The network is extended with decision and utility nodes, as done in decision networks

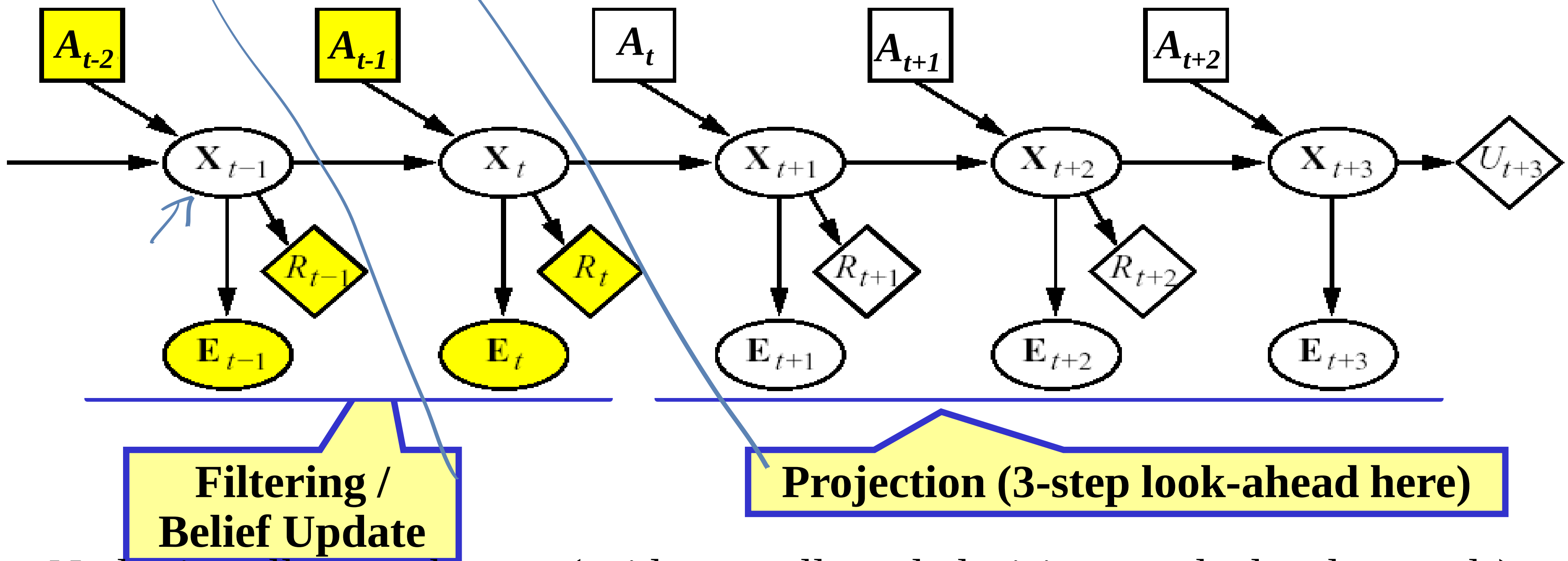


Dynamic Decision Networks (DDN)

- A filtering algorithm is used to incorporate each new percept and the action to update the belief state $\mathbf{X}_t$
- Decisions are made by projecting forward possible action sequences and choosing the best one: ***look ahead search***



Dynamic Decision Networks (DDN)



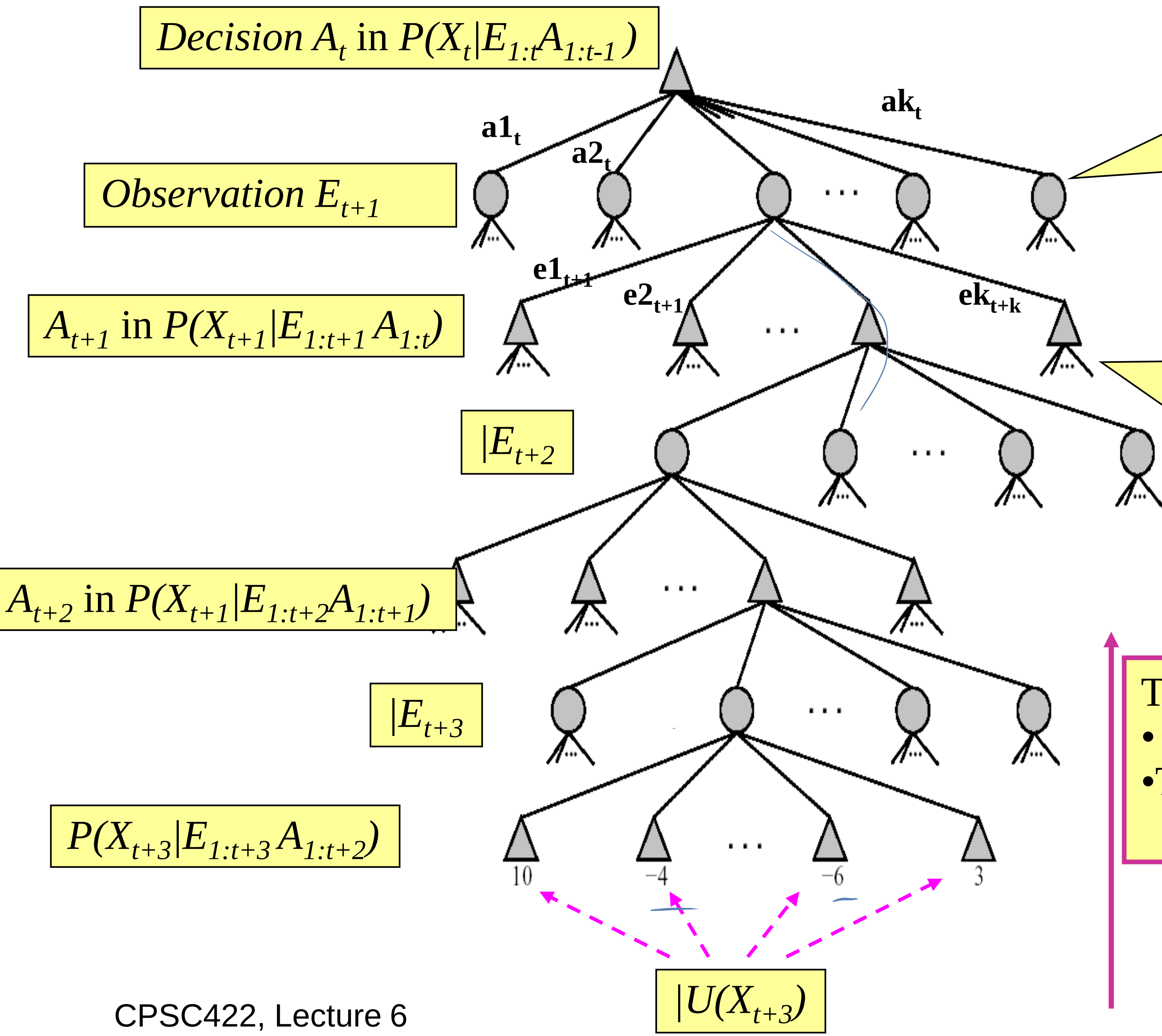
- Nodes in yellow are known (evidence collected, decisions made, local rewards)
- Agent needs to make a decision at time t (A_t node)
- Network unrolled into the future for 3 steps
- Node U_{t+3} represents the utility (or expected optimal reward V^*) in state X_{t+3}
  - i.e., the reward in that state and all subsequent rewards
  - Available only in approximate form (from another approx. method)

Look Ahead Search for Optimal Policy

General Idea:

- **Expand the decision process for n steps into the future, that is**
  - "Try" all actions at every decision point
  - Assume receiving all possible observations at observation points
- **Result: tree of depth *2n+1*** where
  - every branch represents one of the possible sequences of *n* actions and n observations available to the agent, and the corresponding belief states
  - The leaf at the end of each branch corresponds to the *belief state* reachable via that sequence of actions and observations – use filtering/belief-update to compute it
- **"Back Up" the utility values of the leaf nodes** along their corresponding branches, **combining it with the rewards** along that path
- **Pick the branch with the highest expected value**

Look Ahead Search for Optimal Policy



These are chance nodes, describing the probability of each observation

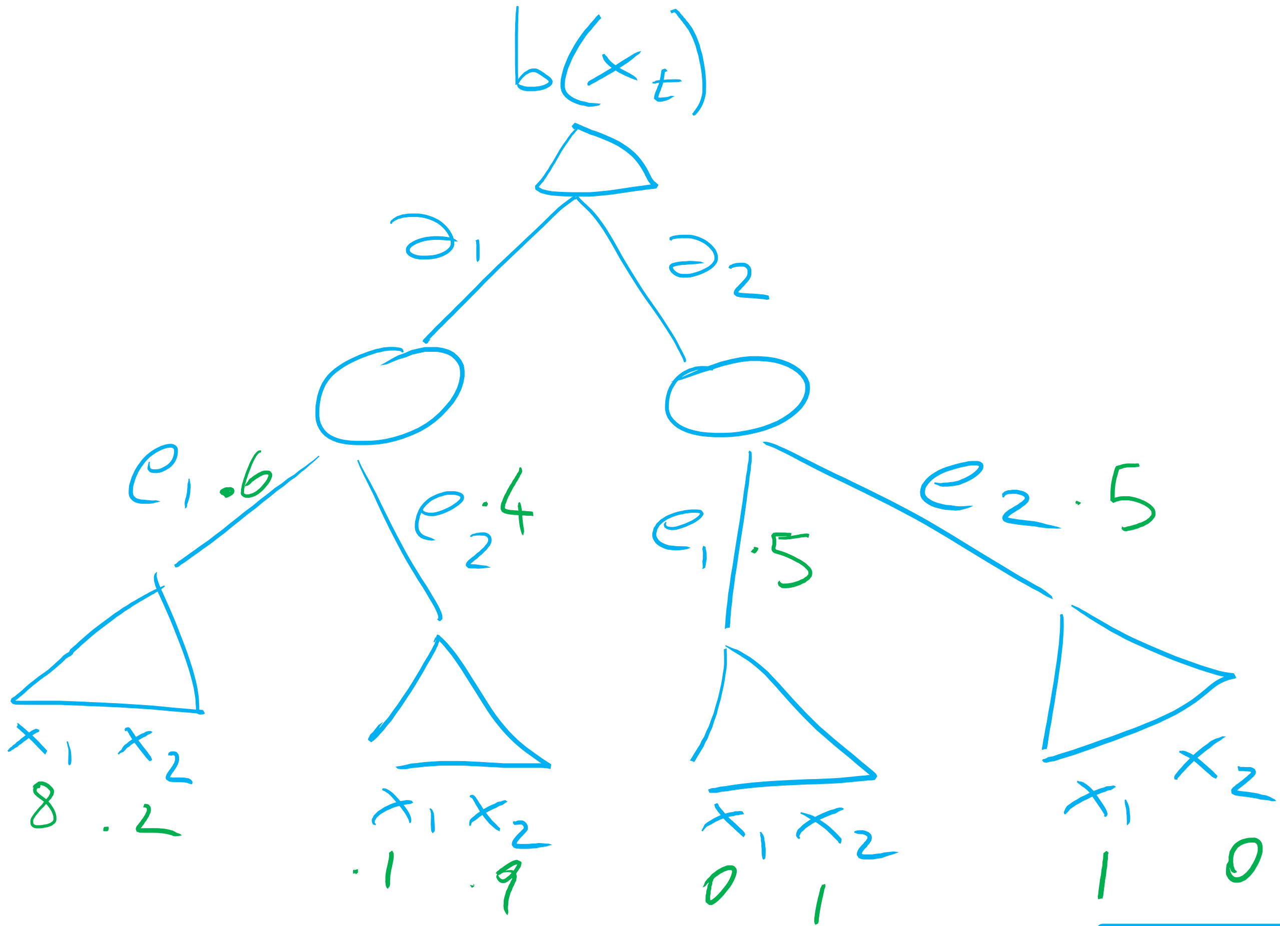
Belief states are computed via any filtering algorithm, given the sequence of actions and observations up to that point

To back up the utilities
- take average at chance points
- Take max at decision points

$X \quad x_1 \; x_2$

$E \quad e_1 \; e_2$

$A \quad a_1 \quad a_2$



$U(x_1) = 1$

$V(x_2) = 0$

➢ Best action at time t?

A. a_1

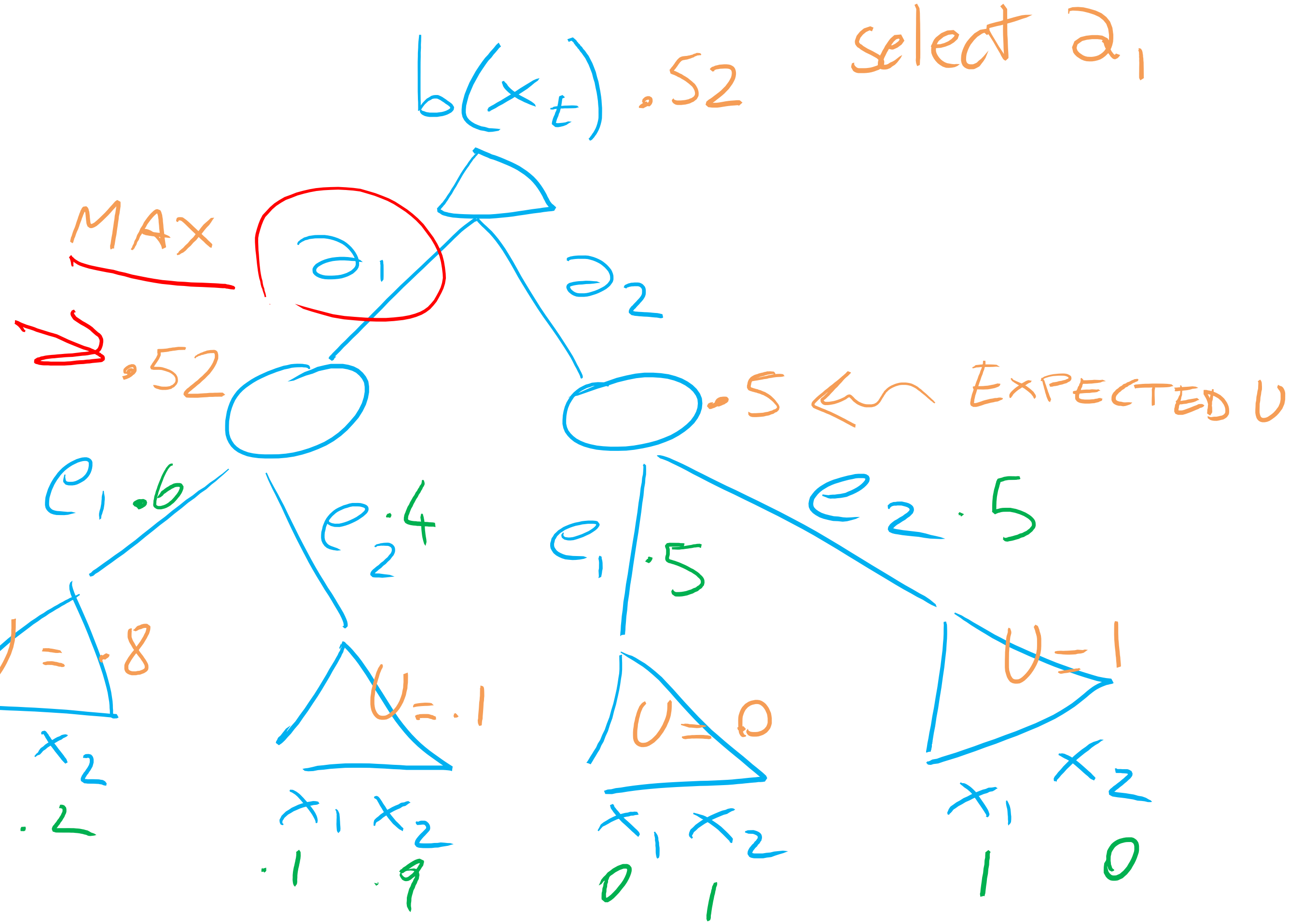
B. a_2

C. *indifferent*

$X \quad x_1 \; x_2$

$E \quad e_1 \; e_2$

$A \quad a_1 \quad a_2$



$U(x_1) = 1$

$U(x_2) = 0$

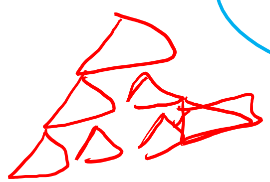
Look Ahead Search for Optimal Policy

➢ What is the time complexity for exhaustive search at depth *d*, with |A| available actions and |E| possible observations?

A. $O(d * |A| * |E|)$

B. $O(|A|^d * |E|^d)$

C. $O(|A|^d * |E|)$



$\overbrace{|A|*|E|*|A|*|E|*|A| \ldots\ldots}^{d \text{ times}}$

- Would Look ahead work better when the discount factor is?

A. *Close to 1*

B. *Not too close to 1*

Some Applications of POMDPs……

- Jesse Hoey, Tobias Schröder, Areej Alhothali (**2015**), Affect control processes: Intelligent affective interaction using a POMDP, *AI Journal*
- S Young, M Gasic, B Thomson, J Williams (**2013**) POMDP-based Statistical Spoken Dialogue Systems: a Review, *Proc IEEE*,
- J. D. Williams and S. Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, **2007**.
- S. Thrun, et al. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotic Research*, 19(11):972–999, **2000**.
- A. N.Rafferty,E. Brunskill,Ts L. Griffiths, and Patrick Shafto. Faster teaching by POMDP planning. In *Proc. of Ai in Education*, pages 280–287, **2011**
- P. Dai, Mausam, and D. S.Weld. Artificial intelligence for artificial artificial intelligence. In *Proc. of the 25th AAAI Conference on AI* , **2011**. [intelligent control of workflows]

Another "famous" Application

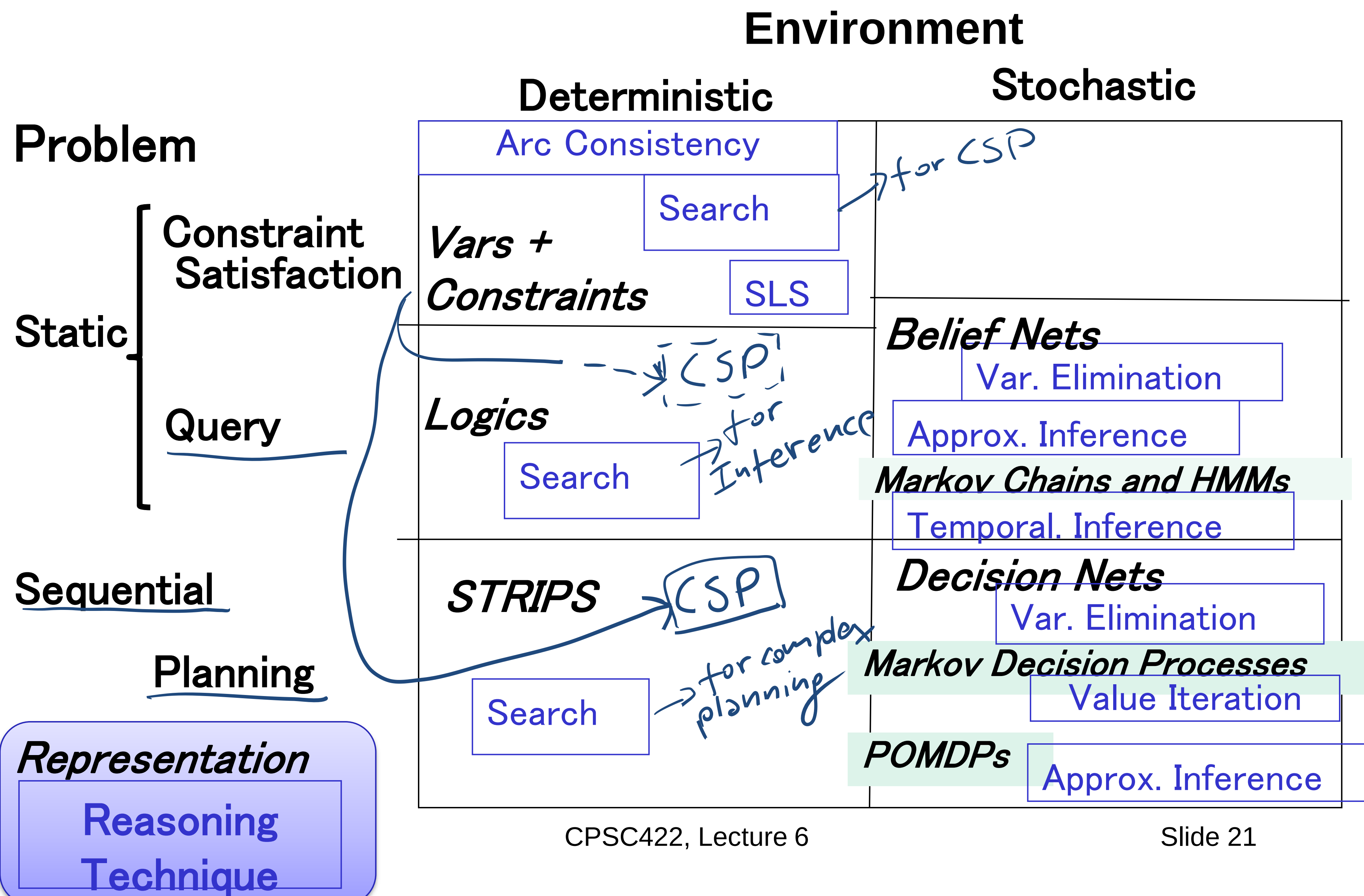
Learning and Using POMDP models of Patient–Caregiver Interactions During Activities of Daily Living

Goal: Help Older adults living with cognitive disabilities (such as Alzheimer's) when they:

- **forget the proper sequence of tasks** that need to be completed
- **they lose track of the steps** that they have already completed.

Source: *Jesse Hoey UofT 2007*

R&R systems BIG PICTURE



422 big picture

STAR AI (Statistical Relational)

Hybrid: Det +Sto

Prob CFG
Prob Relational Models
Markov Logics

MAP MaxWalkSAT

Templates

	Deterministic	**Stochastic**
Query	*Logics* *First Order Logics* *Ontologies* *Temporal rep.* • Full Resolution • SAT	*Belief Nets* Approx. : Gibbs *Markov Chains and HMMs* Forward, Viterbi…. Approx. : Particle Filtering *Undirected Graphical Models* *Conditional Random Fields*
Planning		*Markov Decision Processes and* *Partially Observable MDP* • Value Iteration • Approx. Inference *Reinforcement Learning*

Diff def. for Markov Blanket et

rewrite as ops on factors

Applications of AI

Representation

Reasoning Technique

Learning Goals for today's class

You can:

- Define a **Policy** for a POMDP
- Describe space of possible methods for computing optimal policy for a given POMDP
- Define and trace Look Ahead Search for finding an (approximate) Optimal Policy
- Compute Complexity of Look Ahead Search

TODO for next Fri

- **Read textbook 11.3 (Reinforcement Learning)**
  - 11.3.1 Evolutionary Algorithms
  - 11.3.2 Temporal Differences
  - 11.3.3 Q-learning

- Assignment 1 will be posted on Connect today
  - VInfo and VControl
  - MDPs (Value Iteration)
  - POMDPs