CS325 Artificial Intelligence
Ch. 17 – Planning Under Uncertainty

Cengiz Günay, Emory Univ.

Spring 2013

Is This AI Course a Bit Schizo?

Classical AI vs. Machine Learning

Is This AI Course a Bit Schizo?

Classical AI vs. Machine Learning

- Classical AI
- Symbolic logic (propositional, first-order)
- Algorithms
- **Thinking and programming**

- Classical AI
- Symbolic logic (propositional, first-order)
- Algorithms
- **Thinking and programming**

- Probabilities
- Math
- Machine Learning
- **Automated methods, power of math**

Is This AI Course a Bit Schizo?

Classical AI vs. Machine Learning

- Classical AI
- Symbolic logic (propositional, first-order)
- Algorithms
- **Thinking and programming**

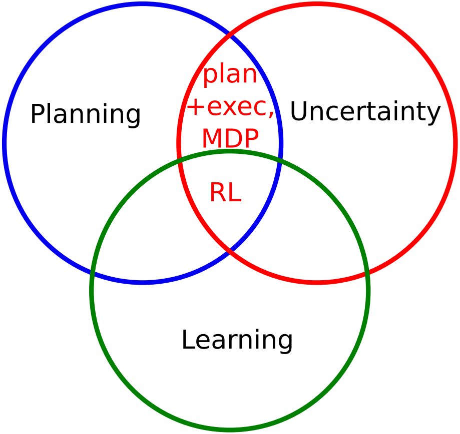
- Probabilities
- Math
- Machine Learning
- **Automated methods, power of math**

Planning Under Uncertainty

- Into Thrun territory
- Aim is to use more math, probabilities
- achieve **learnability** for hard-to-program scenarios (that is, real-life)

Planning Under Uncertainty

- Into Thrun territory
- Aim is to use more math, probabilities
- achieve **learnability** for hard-to-program scenarios (that is, real-life)



Entry/Exit Surveys

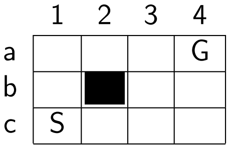
Exit survey: Planning

- Why do we need to alternate between plan and execution?
- Why do we need a belief state?

Entry survey: Planning Under Uncertainty (0.25 points of final grade)

- What algorithm would you use to plan under uncertain conditions?
- How do you think machine learning can be used in planning?

So What's Wrong with Classical Planning?

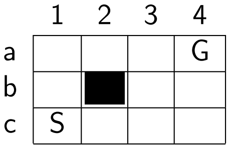


Grid World:

S: Start

G: Goal

So What's Wrong with Classical Planning?



Grid World:

S: Start

G: Goal

It's too slow

- Branching factor can get large
- Search tree gets too deep (may have loops)
- Same states can be repeated multiple times (although can be avoided with dynamic programming)

Start with Certainty: Deterministic Grid World

	1	2	3	4
a				$+1$
b				
c	S			

Reward function:
$R(s) = +1$ @ a4

- Remember utility values?
- State, s
- Action, a
- Optimal policy $\pi(s) \rightarrow a$?

Start with Certainty: Deterministic Grid World

	1	2	3	4
a				$+1$
b				-1
c	S			

Reward function:
$R(s) = +1$ @ a4

- Remember utility values?
- State, s
- Action, a
- Optimal policy $\pi(s) \rightarrow a$?

Start with Certainty: Deterministic Grid World

	1	2	3	4
a				$+1$
b				-1
c	S			

Reward function:
$R(s) = +1$ @ a4

- Remember utility values?
- State, s
- Action, a
- Optimal policy $\pi(s) \rightarrow a$?
  - @ a3?
  - @ b3?
  - @ c4?

Start with Certainty: Deterministic Grid World

	1	2	3	4
a			$\rightarrow$	$+1$
b		■	$\uparrow$	-1
c	S		$\uparrow$	$\leftarrow$

Reward function:
$R(s) = +1$ @ a4

- Remember utility values?
- State, s
- Action, a
- Optimal policy $\pi(s) \rightarrow a$?
    - @ a3?
    - @ b3?
    - @ c4?

Value Iteration: Movement Cost

	1	2	3	4
a				+1
b		■		−1
c	S			

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ -.1 & \text{everywhere else} \end{cases}$$

Value Iteration: Movement Cost

	1	2	3	4
a				+1
b				−1
c	S			

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ -.1 & \text{everywhere else} \end{cases}$$

- Optimal policy $\pi(s) \rightarrow a$?
  - @ a3?
  - @ b3?
  - @ c4?

Value Iteration: Movement Cost

	1	2	3	4
a			0.9	+1
b				−1
c	S			

Reward function:

$$R(s) = \begin{cases} +1 & \texttt{@}\,\mathrm{a4} \\ -1 & \texttt{@}\,\mathrm{b4} \\ -.1 & \text{everywhere else} \end{cases}$$

- Optimal policy $\pi(s) \to a$?
  - @ a3?
  - @ b3?
  - @ c4?

Value Iteration: Movement Cost

	1	2	3	4
a			0.9	+1
b		[redacted]	0.8	−1
c	S			

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ -.1 & \text{everywhere else} \end{cases}$$

- Optimal policy $\pi(s) \rightarrow a$?
  - @ a3?
  - @ b3?
  - @ c4?

Value Iteration: Movement Cost

	1	2	3	4
a			0.9	+1
b		■	0.8	−1
c	S		0.7	0.6

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ -.1 & \text{everywhere else} \end{cases}$$

- Optimal policy $\pi(s) \rightarrow a$?
  - @ a3?
  - @ b3?
  - @ c4?

	1	2	3	4
a			0.9	+1
b		[redacted]	0.8	−1
c	S		0.7	0.6

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ -.1 & \text{everywhere else} \end{cases}$$

- Optimal policy $\pi(s) \rightarrow a$?
    - @ a3?
    - @ b3?
    - @ c4?

Value function:

$$V(s) \leftarrow \left[\arg\max_a V(s')\right] + R(s)$$

where s' is neighboring states.

Value Iteration Video

Value iteration video

Value Iteration: Discount Factor

	1	2	3	4
a			0.9	$+1$
b		■	0.8	-1
c	S		0.7	0.6

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ \mathbf{0} & \text{everywhere else} \end{cases}$$

Recursive definition

$$V(s) \leftarrow \left[\arg\max_a V(s')\right] + R(s)\,,$$

can be also written as expected reward

$$V(s) \leftarrow \arg\max_\pi E\left[\sum_{t=0}^{\infty} \gamma^t R_t \;|s_o = s\right]\,.$$

Instead of movement cost, it uses **discount factor**, γ, to decay future reward.

Value Iteration: Discount Factor

	1	2	3	4
a			0.9	+1
b		■	0.8	−1
c	S		0.7	0.6

Reward function:

$$R(s) = \begin{cases} +1 & @\,\mathrm{a4} \\ -1 & @\,\mathrm{b4} \\ \mathbf{0} & \text{everywhere else} \end{cases}$$

Recursive definition

$$V(s) \leftarrow \left[\arg\max_a V(s')\right] + R(s)\,,$$

can be also written as expected reward

$$V(s) \leftarrow \arg\max_\pi E\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid s_o = s\right]\,.$$

Instead of movement cost, it uses **discount factor**, γ, to decay future reward.

- Helps to keep it bounded $\leq \frac{1}{1-\gamma}|R_{\max}|$

Value Iteration: Bellman Equation

General case (Bellman, 1957) is stochastic

$$V(s) \leftarrow \left[\arg\max_{a} \gamma \sum_{s'} P(s'|a)V(s')\right] + R(s) \, .$$

- Recursive
- Used iteratively
- Converges to solution

Value Iteration: Bellman Equation

General case (Bellman, 1957) is stochastic

$$V(s) \leftarrow \left[\arg\max_a \gamma \sum_{s'} P(s'|a) V(s') \right] + R(s) \,.$$

- Recursive
- Used iteratively
- Converges to solution

Why stochastic? Remember we want to plan under **uncertainty**

Markov Decision Processes

Andrey Andreyevich Markov (1856–1922)

- Russian mathematician
- Stochastic processes

Andrey Andreyevich Markov
(1856–1922)

- Russian mathematician
- Stochastic processes

Markov Decision Processes (MDPs)

- Value iteration with stochasticity (Bellman, 1957)

Markov Decision Processes

Andrey Andreyevich Markov (1856–1922)

- Russian mathematician
- Stochastic processes

Markov Decision Processes (MDPs)

- Value iteration with stochasticity (Bellman, 1957)

Later

- Q-learning (1989) $\rightarrow$ (next class)

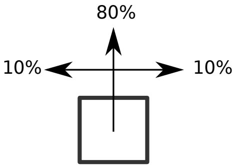
Robots in Real Life

Video: Robots gone wild

	1	2	3	4
a				$+1$
b		■		-1
c				

Reward function:

$$R(s) = \begin{cases} +1 & @\,\mathrm{a4} \\ -1 & @\,\mathrm{b4} \\ \mathbf{0} & \text{else} \end{cases}$$



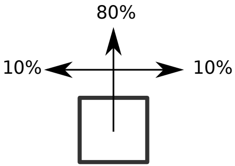
- Optimal policy $\pi(s) \to a$?

Uncertain Movement in Grid World

	1	2	3	4
a				$+1$
b		■		-1
c				

Reward function:

$$R(s) = \begin{cases} +1 & @\,\text{a4} \\ -1 & @\,\text{b4} \\ \mathbf{0} & \text{else} \end{cases}$$



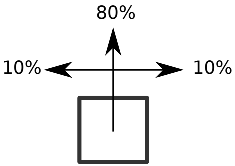
- Optimal policy $\pi(s) \rightarrow a$?
  - @ a3?
  - @ b3?
  - @ c4?

Uncertain Movement in Grid World

	1	2	3	4
a			$\rightarrow$	$+1$
b				-1
c				

Reward function:

$$R(s) = \begin{cases} +1 & @\,\text{a4} \\ -1 & @\,\text{b4} \\ \color{red}{\mathbf{0}} & \text{else} \end{cases}$$



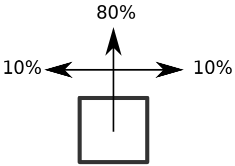
- Optimal policy $\pi(s) \rightarrow a$?
  - @ a3?
  - @ b3?
  - @ c4?

Uncertain Movement in Grid World

	1	2	3	4
a			$\rightarrow$	$+1$
b			$\leftarrow$	-1
c				

Reward function:

$$R(s) = \begin{cases} +1 & @\,\mathrm{a4} \\ -1 & @\,\mathrm{b4} \\ \color{red}{\mathbf{0}} & \text{else} \end{cases}$$

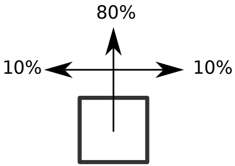


- Optimal policy $\pi(s) \rightarrow a$?
  - @ a3?
  - @ b3?
  - @ c4?

	1	2	3	4
a	$\rightarrow$	$\rightarrow$	$\rightarrow$	$+1$
b	$\uparrow$	■	$\leftarrow$	-1
c	$\uparrow$	$\leftarrow$	$\leftarrow$	$\downarrow$

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ \color{red}{\mathbf{0}} & \text{else} \end{cases}$$



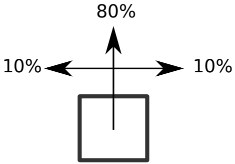
- Optimal policy $\pi(s) \rightarrow a$?
    - @ a3?
    - @ b3?
    - @ c4?

Stochastic Value Iteration

	1	2	3	4
a				$+1$
b				-1
c				

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ -.03 & \text{else} \end{cases}$$



$$V(s) \leftarrow \left[\arg\max_a \gamma \sum_{s'} P(s'|a) V(s') \right] + R(s), \ \gamma = 1$$

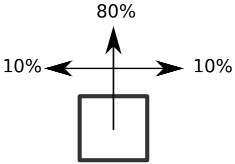
Optimal policy $\pi(s) \rightarrow a$?

Stochastic Value Iteration

	1	2	3	4
a				+1
b		■		−1
c				

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ \mathbf{-.03} & \text{else} \end{cases}$$



$$V(s) \leftarrow \left[\arg\max_a \gamma \sum_{s'} P(s'|a) V(s') \right] + R(s), \;\; \gamma = 1$$

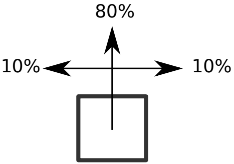
Optimal policy $\pi(s) \rightarrow a$?

- @ a3?
- @ b3?

	1	2	3	4
a			.77	+1
b				−1
c				

Reward function:

$$R(s) = \begin{cases} +1 & \text{@ a4} \\ -1 & \text{@ b4} \\ \color{red}{-.03} & \text{else} \end{cases}$$



$$V(s) \leftarrow \left[\arg\max_a \gamma \sum_{s'} P(s'|a)V(s')\right] + R(s),\ \gamma = 1$$

Optimal policy $\pi(s) \rightarrow a$?

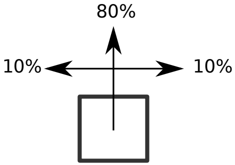
- @ a3?
- @ b3?

Stochastic Value Iteration

	1	2	3	4
a			.77	+1
b			.48	−1
c				

Reward function:

$$R(s) = \begin{cases} +1 & @\ \mathrm{a4} \\ -1 & @\ \mathrm{b4} \\ \color{red}{-.03} & \text{else} \end{cases}$$

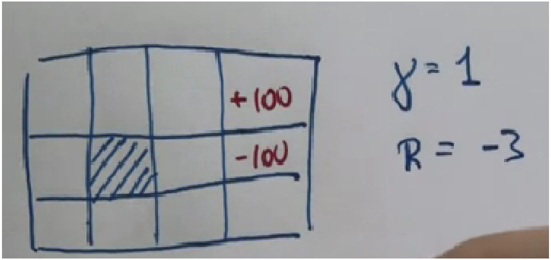


$$V(s) \leftarrow \left[\arg\max_{a} \gamma \sum_{s'} P(s'|a)V(s') \right] + R(s),\ \gamma = 1$$

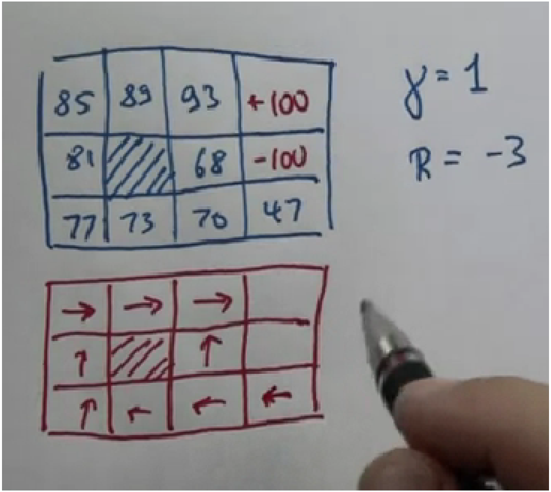
Optimal policy $\pi(s) \rightarrow a$?

- @ a3?
- @ b3?

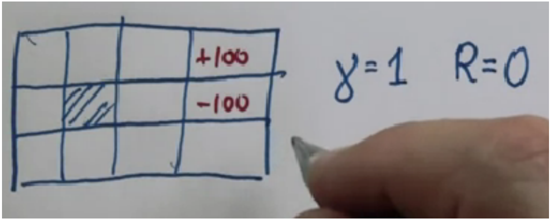
Values and Policy Examples



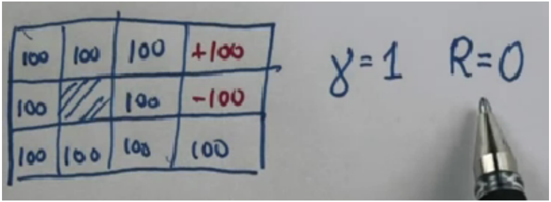
Values and Policy Examples



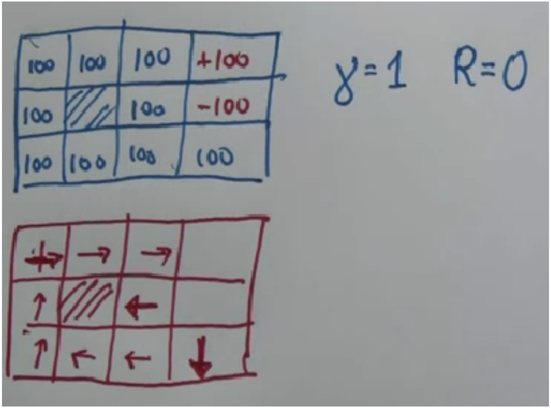
Values and Policy Examples



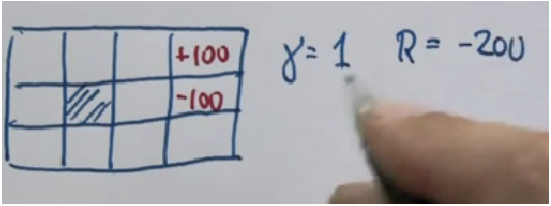
Values and Policy Examples



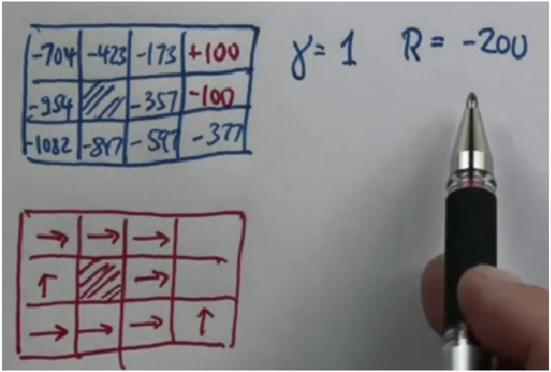
Values and Policy Examples



Values and Policy Examples



Values and Policy Examples



Markov Decision Processes Summary

- Fully observable: $s_1, \ldots, s_n \quad a_1, \ldots, a_m$
- Stochastic $P(s'|a, s)$
- Reward $R(s)$
- Objective $\max_\pi \ E\left[\sum_{t=0}^{\infty} \gamma^t R_t \ |s_o = s\right]$.
- Value iteration $V(s)$
- Converges to optimal policy, $\pi = \arg\max \ldots$

Partially Observable MDPs