1

DECISION PROCESSES

EECS 492 Lecture 21
March 29, 2011

Administrative

2

Count							1	1	5	4	6	6	2	7	7	8	12	9	4	1
Score	0.0 - 3.8	3.8 - 7.5	7.5 - 11.2	11.2 - 15.0	15.0 - 18.8	18.8 - 22.5	22.5 - 26.2	26.2 - 30.0	30.0 - 33.8	33.8 - 37.5	37.5 - 41.2	41.2 - 45.0	45.0 - 48.8	48.8 - 52.5	52.5 - 56.2	56.2 - 60.0	60.0 - 63.8	63.8 - 67.5	67.5 - 71.2	71.2 - 75.0

Class median: 55.0
Class average: 52.5
Class stddev: 12.0
Num scores: 73

Maximum Expected Utility

3

- Rational agents maximize expected utility
- Which would you prefer?
  - A) Roll a die, I pay you $1 for every pip on the die.
  - B) Flip a (fair) coin: if heads, I pay you $6. If tails, I pay you nothing.
  - What is our utility?
    - Utility = money in your pocket?

Why Utility and MEU?

- Maybe preferences could be more expressive than real-valued functions.
  - Preference order, ≥ (“at least as preferred”)
  - ranking outcomes of actions
- Outcomes are prospects:

$$\mu = [p, \omega_1; \omega_2]$$

  - means ω_1 with probability p, ω_2 otherwise
  - ω_1 and ω_2 may be prospects

Preference under Uncertainty: Axioms

orderability: $(\omega_1 \geq \omega_2) \vee (\omega_2 \geq \omega_1)$

transitivity: $(\omega_1 \geq \omega_2) \wedge (\omega_2 \geq \omega_3) \rightarrow (\omega_1 \geq \omega_3)$

continuity: $\omega_1 \geq \omega_2 \geq \omega_3 \rightarrow \exists\, p.\ \omega_2 \sim [p, \omega_1; \omega_3]$

substitution: $\omega_1 \sim \omega_2 \rightarrow [p, \omega_1; \omega_3] \sim [p, \omega_2; \omega_3]$

monotonicity:

$$\omega_1 \geq \omega_2 \wedge p > q \rightarrow [p, \omega_1; \omega_2] \geq [q, \omega_1; \omega_2]$$

decomposability:

$$[p, \omega_1; [q, \omega_2; \omega_3]] \sim [q, [p, \omega_1; \omega_2]; [p, \omega_1; \omega_3]]$$

indifference: $\omega_1 \sim \omega_2 \equiv (\omega_1 \geq \omega_2) \wedge (\omega_2 \geq \omega_1)$

Preference under Uncertainty

- If an ordering of preferences exists, then we can assign real-valued numbers to each outcome such that more desirable outcomes always have larger values.
  - $u([p, \omega_1; \omega_2]) = p\, u(\omega_1) + (1-p)u(\omega_2)$
- Given the following axioms of $\geq$:
  - orderability, transitivity, continuity, substitution, monotonicity, decomposability
  - ➔ An ordering exists.

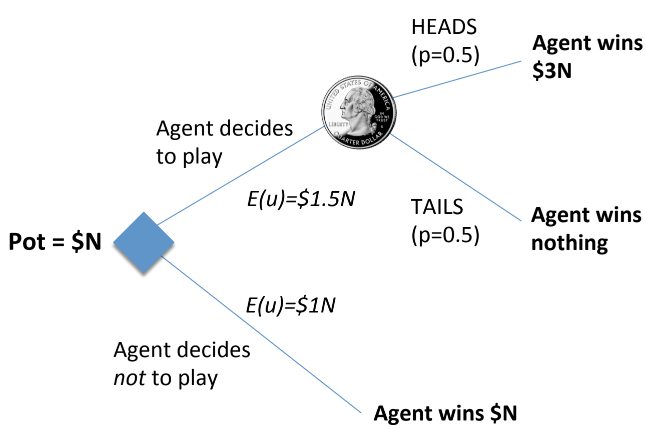
St. Petersburg Paradox

7

- Are you rational?
  - What's your utility function?

- I put a dollar in the pot.
- I flip a coin.
  - Heads: You can keep the pot, or **triple-or-nothing**.
  - Tails: I keep the pot, game over.

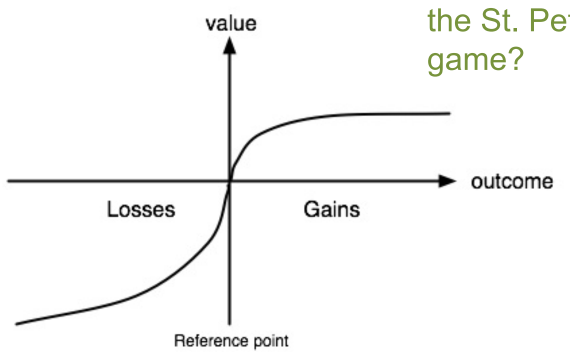
St. Petersburg Paradox: Decision Tree



- What are the expected utilities of each of the agent's choices?

Human Utility Functions

- How do humans gamble?



Kahneman & Tversky

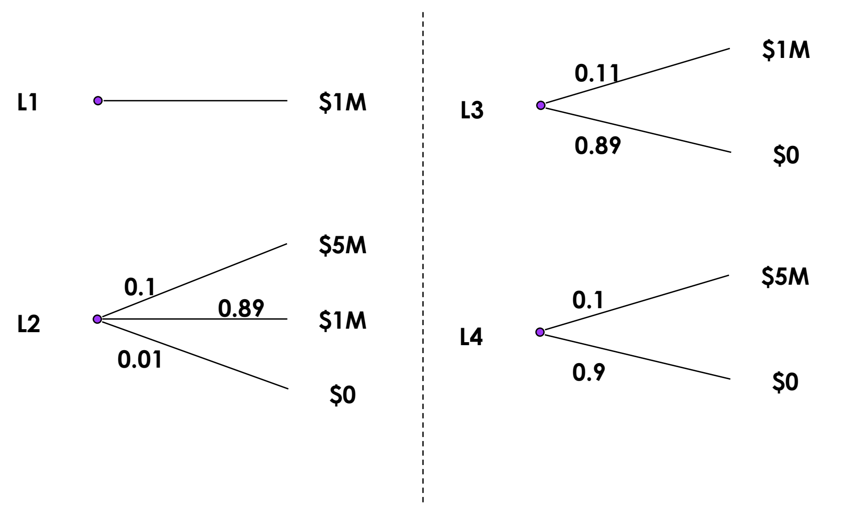
Does this provide a better strategy for the St. Petersburg game?

Human Utility Functions

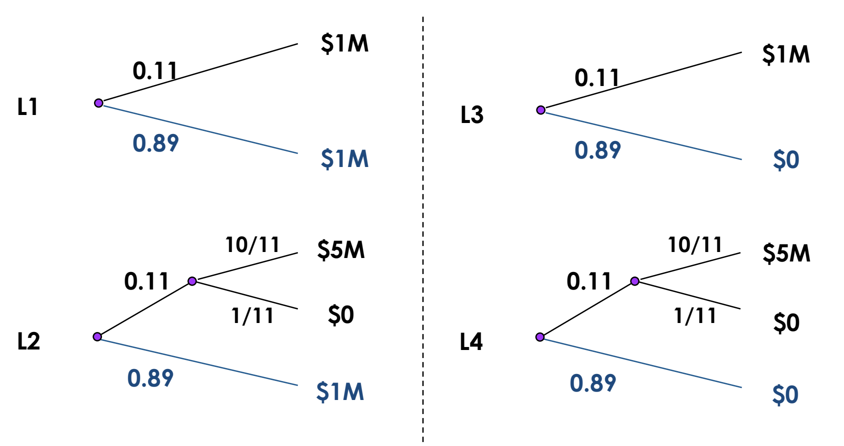
10

- Non-linear utility of money functions explain much of human behavior.
- For humans to be rational, there just needs to be *some* utility function that obeys the axioms.
  - So, are humans rational given “the right” utility function?

Allais Paradox



Allais Paradox 2



substitution: $\omega_1 \sim \omega_2 \rightarrow [p, \omega_1; \omega_3] \sim [p, \omega_2; \omega_3]$

Making Sequences of Decisions

13

A Sequential Decision Process

14

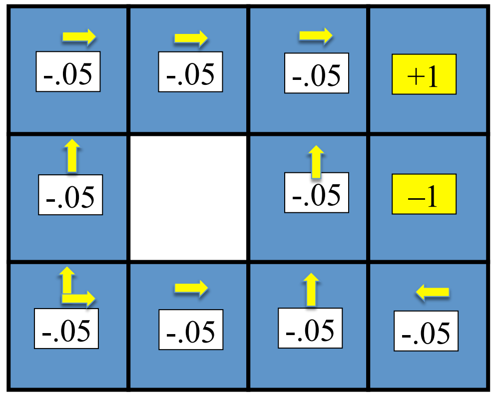
- Deterministic maze world:
  - Agent can move to any adjacent square
- What sequence of actions maximizes the utility?
  - Utility = sum of “rewards” in each grid

-.05	-.05	-.05	+1
-.05		-.05	–1
-.05	-.05	-.05	-.05

Optimal Policy

15

- Deterministic maze world:
  - We can pre-compute the action at each state that will maximize the utility of the agent.
  - Result: Simple reflexive agent



This is boring because the world is deterministic. How do we handle non-determinism?

Policy notation

16

- The policy π says to perform action a when in state s:

  $\pi(s) = a$

- The optimal policy is written π^*

Simplifying Assumptions

17

- We'll make several assumptions…
  - Markov Assumption
  - Stationary Preferences

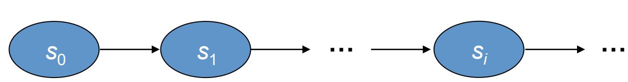
Markov Models

18

- Sequence of states, $s_0, s_1, \ldots, s_i \in S$
- *Markov property*:

$$\Pr(s_i \mid s_0, s_1, \ldots, s_{i-1}) = \Pr(s_i \mid s_{i-1})$$

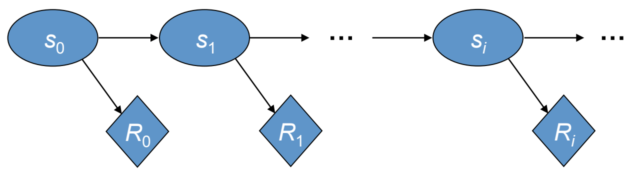
  - Next state conditionally independent of history, given current state
  - Graphical (Bayes net) rep'n:



Adding Rewards to Markov Model

19

- Can associate *reward* (immediate utility) with each state



Overall utility is a function of immediate rewards

Stationarity of Preferences

20

- Utility of state sequence = sum of rewards at each state
- Stationarity:
  - Suppose I'm in some state s.
  - Is the utility of a state sequence beginning with s unchanging?
- Suppose there's a time limit (game ends after move N)
  - Utility of reaching goal state changes, depending on how many moves have been performed so far.
  - Not stationary

Stationary Preferences

21

- The property of Stationary Preferences has an important consequence:

The utility of a state sequence can always be written:

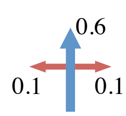
$U([s_0,s_1,s_2,\ldots]) = R(s_0) + \gamma U([s_1,s_2,\ldots])$

$U([s_0,s_1,s_2,\ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots$

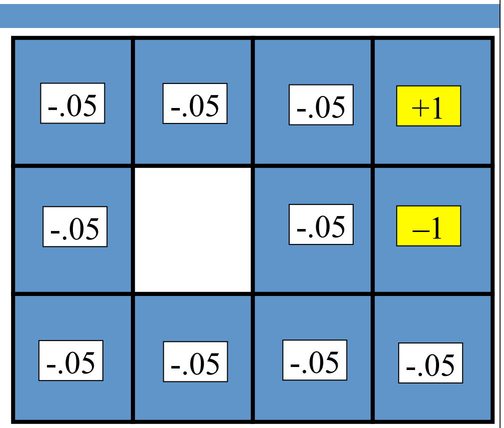
- What is γ in our simple example?
- What does it mean if γ is < 1?

Non-Deterministic Example

22



- Actions succeed with probability 0.6.
- Probability 0.1 of going in orthogonal direction.
- Probability 0.2 of nothing happening.
- Reward of –.05 for nonterminal state.



Question: Does an optimal policy exist?

Non-Determinism: Formulation

23

- Transition probability:

P(in state s_i | was in state s_{i-1}, performed action a)
= T(s_{i-1}, a, s_i)

- Immediate rewards function:
  = R(s)

Optimal Policy

24

- Our definition of rationality: maximize expected utility
- Optimal policy must maximize the expected utility for whatever state we might be in…
- Note: try to keep "reward" and "utility" straight
  - A reward is an immediate payouts
  - Utilities are a function of all future payouts.

Expected Utility of a State

25

- Suppose (for just a second) that we *know* the optimal policy π^*
- Suppose that $U^*(s)$ is the expected utility for an agent in state s that follows the optimal policy.

$$U([s_0,s_1,s_2,\ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots$$

$$U^*(s) = R(s) + \gamma(\text{Expected utility of next state})$$

$$U^*(s) = R(s) + \gamma \sum_{s'} T(s, \pi^*(s), s')U^*(s')$$

Optimal Action

26

- From previous slide:

$$U^*(s) = R(s) + \gamma \sum_{s'} T(s, \pi^*(s), s')U^*(s')$$

- The optimal action $\pi^*(s)$ is the action that maximizes that expression!

$$\pi^*(s) = \text{argmax}_a \sum_{s'} T(s, a{=}\pi^*(s), s')U^*(s')$$

- If finite number of actions, we can just try all actions and pick the one with the maximum expected utility.

Recursive definition of U*

27

- Combining the equations yields:

$$U^*(s) = R(s) + \max_a \gamma \sum_{s'} T(s, a, s')U^*(s')$$

- Of course, we don't know U*
  - But this suggests a way to compute it…

Value Iteration

28

- Initialize $U_0(s)$ to arbitrary values (zeros, maybe)
- Iterate:

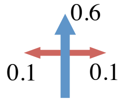
$$U_i(s) = R(s) + \max_a \gamma \sum_{s'} T(s, a, s')U_{i-1}(s')$$

- Intuition:
  - Immediate rewards are discounted and "percolated" to adjacent states, then states adjacent to adjacent states, and so on.
  - U_i approaches U*…. (maybe?)

Value Iteration

29

- Initialize all estimates to 0.
- Transition probabilities:



- Each non-terminal state has reward -0.05

.118	.28	.55	+1
.0208		.18	–1
-.03934	-.0182	.053	-.05

$U(i) \leftarrow R(i) + \max_a \Sigma_j T(i,a,j) U(j)$

Second Iteration Utility Values

30

- Values and policy after second pass.

→ .2886	→ .5018	→ .733	+1
↑ .1315		↑ .3438	–1
↑ .0181	→ .0364	↑ .1561	↓ -.0897

$U(i) \leftarrow R(i) + \max_a \Sigma_j T(i,a,j) U(j)$

Value Iteration: Convergence

31

- Value iteration converged in this case. Will it always?

Contraction

32

- Bellman equation:

$$U_i(s) = R(s) + \max_a \gamma \Sigma_{s'} T(s, a, s')U_{i-1}(s')$$

- Error = $||U_i - U^*|| = \max_s ||U_i(s) - U^*(s)||$
- Let B be the Bellman operator.
  - Error at step i: $||U_i - U^*||$
  - Error at step i+1: $||BU_i - U^*||$

Contraction

33

- Bellman equation:

$$U_i(s) = R(s) + \max_a \gamma \Sigma_{s'} T(s, a, s')U_{i-1}(s')$$

Error at step i+1: $||BU_i - U^*||$

$= ||BU_i - BU^*||$

$$Error_{i+1} = \max_s (R(s) + \max_a \gamma \Sigma_{s'} T(s, a, s')U_i(s') - R(s) - \max_a \gamma \Sigma_{s'} T(s, a, s')U^*(s'))$$

$$= \max_s \gamma(\max_a \Sigma_{s'} T(s, a, s')U_i(s') - \max_a \Sigma_{s'} T(s, a, s')U^*(s'))$$

$$Error_{i+1} \leq \gamma || U_i - U^* ||$$

Contraction

34

- Key result: Bellman iteration reduces error by factor γ

$$Error_{i+1} \leq \gamma || U_i - U^* ||$$

- Does this make sense?

$\gamma = 0$

$\gamma = .9999$

$\gamma = 1.0$

Your Turn: Acrophobe at the Canyon

35

- Wants to gaze upon a grand vista (be close to the edge)
- Afraid of slipping & falling into the canyon! $\gamma = 0.5$

Action	Result
Back up	Back up with Pr = 1
Stay	Stay with Pr = 0.9, Forward with Pr = 0.1 ("slip")
Forward	Forward with Pr = 1

	2 steps from edge	1 step from edge	Right at edge	Oops!
Reward	1	10	20	-50 or -100

Policy Loss Bound

36

- Suppose we iteratively update U_i using value iteration
  - We can compute the change in error $||U_{i+1} - U_i||$
  - We can also compute the policy π_i
- If we execute π_i instead of π^*, what will be the expected utility of the agent in comparison to U*?
- Important Result (see R&N for some more details)
  $||U_{i+1} - U_i|| < \varepsilon (1-\gamma)/ \gamma \Rightarrow ||U_{i+1} - U^*|| < \varepsilon$
  $||U^{\pi i} - U^*|| < 2 \varepsilon \gamma / (1- \gamma)$

Policy Loss

37

- Do we need optimal U* to compute π*?
  - Hint: We pick the action with the greatest expected utility
  - At what point did we know the Acrophobe's best policy?
    - Did we have to wait until U_i converged?

Policy Iteration

38

- A second way to compute optimal policies
- Begin with an initial policy π_0
- Iterate:
  - **Policy Evaluation:** given a policy π_i, compute $U_i = U^{\pi i}$
  - **Policy Improvement:** Calculate a new MEU policy π_i using one-step look-ahead based on U_i

$$\pi^*(s) = \text{argmax}_a \Sigma_{s'} T(s, a, s')U^*(s')$$

Policy Iteration

39

- **Policy Evaluation:** given a policy π_i, compute $U_i = U^{\pi i}$
- Similar to a value iteration step:

$$U_i(s) = R(s) + \max_a \gamma \sum_{s'} T(s, a, s') U_{i-1}(s')$$

...except that we don't have to consider all actions: we are assuming a policy! (no **max!**)

$$U_i(s) = R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s') U_i(s')$$

$$U(0) = R(0) + B_0 U_i(0) + B_1 U_i(1) + B_2 U_i(2) + \ldots$$
$$U(1) = R(1) + C_0 U_i(0) + C_1 U_i(1) + C_2 U_i(2) + \ldots$$
$$\ldots$$

Policy Iteration Example

40

Action	Result
Back up	Back up with Pr = 1
Stay	Stay with Pr = 0.9, Forward with Pr = 0.1 ("slip")
Forward	Forward with Pr = 1

	2 steps from edge State 0	1 step from edge State 1	Right at edge State 2	Oops! State 3
Reward	1	10	20	-100
Policy	F	F	S	---

U0 = 1 + 0.5 * U1	U0 = 12.818
U1 = 10 + 0.5 * U2	U1 = 23.636
U2 = 20 + 0.5 * (0.9*U2 + 0.1*U3)	U2 = 27.273
U3 = -100	U3 = -100

Value Iteration vs. Policy Iteration

41

- Value Iteration:
  - Iterations are cheap, but information flows slowly.
- Policy iteration
  - Iterations are expensive (matrix inversion), but information flows rapidly between states.
- Modified Policy iteration
  - Compromise between the two: periodically recompute policy, but update utilities approximately (instead of via matrix inversion)

POMDPs

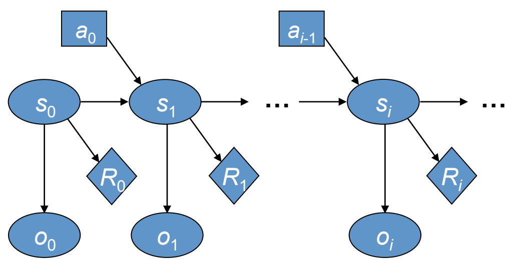
42

- We've studied Markov decision processes (MDPs)
  - World is observable (what does that mean?)
- What if our state is uncertain?

Partial Observability (POMDP)

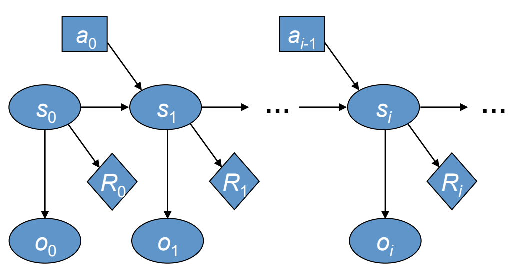
43

- Agent cannot necessarily determine current state
- Available evidence specified by observability model, $\Pr(o_i \mid s_i)$
  - We do *NOT* observe s_i



Policies

44



- *Observations* do not obey Markov property
- ∴ Policies:
  - function of entire history
  - nonstationary
  - Complexity of inference rapidly becomes expensive

Belief States

45

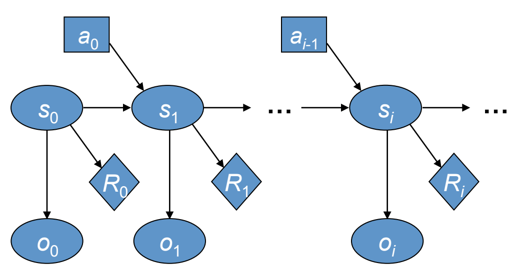
- Sequence of observations induces probability distribution over states

  $b_i(s) = \Pr(s_i = s \mid o_0, o_1, \ldots, o_{i-1})$
- Idea: Represent policies as function from beliefs to actions
  - MDP methods, results apply
  - Not generally practical, as belief state is continuous and highly dimensional
  - Approximation techniques available

Dynamic Decision Networks

46

- Use forward search techniques over limited horizon version of POMDP network



Summary

47

- Planning in probabilistic domains + Markov → Markov Decision Process (MDP)
  - Stationary Preferences lead to notion of discounted rewards.
- Two approaches for solving MDPs
  - Value Iteration
    - Compute good U estimates using non-linear Bellman updates
    - Compute policy from final U estimate.
  - Policy Iteration
    - Alternately update policy and U estimates
    - Having a policy estimate allows linear Bellman updates
- POMDPs

Next Time

48

- Learning
- Classification/Regression