CS 188: Artificial Intelligence
Fall 2007

Lecture 9: Utilities

9/25/2007

Dan Klein – UC Berkeley

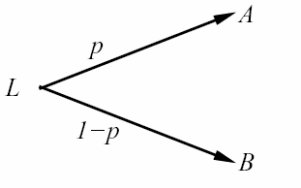
Announcements

- Project 2 (due 10/1)
- SVN groups available, email us to request
- Midterm
  - 10/16 in class
  - One side of a page cheat sheet allowed (provided you write it yourself)
  - Tell us NOW about conflicts!

Preferences

- An agent chooses among:
  - Prizes: A, B, etc.
  - Lotteries: situations with uncertain prizes

$$L = [p, A;\ (1-p), B]$$

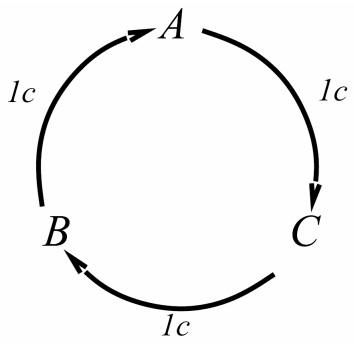


- Notation:

$A \succ B$	A preferred over B
$A \sim B$	indifference between A and B
$A \succeq B$	B not preferred over A

Rational Preferences

- We want some constraints on preferences before we call them rational
- For example: an agent with intransitive preferences can be induced to give away all its money
  - If B > C, then an agent with C would pay (say) 1 cent to get B
  - If A > B, then an agent with B would pay (say) 1 cent to get A
  - If C > A, then an agent with A would pay (say) 1 cent to get C



Rational Preferences

- Preferences of a rational agent must obey constraints.
  - These constraints are the axioms of rationality

Orderability

$$(A \succ B) \vee (B \succ A) \vee (A \sim B)$$

Transitivity

$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$

Continuity

$$A \succ B \succ C \Rightarrow \exists p \; [p, A; \; 1-p, C] \sim B$$

Substitutability

$$A \sim B \Rightarrow [p, A; \; 1-p, C] \sim [p, B; 1-p, C]$$

Monotonicity

$$A \succ B \Rightarrow$$
$$(p \geq q \Leftrightarrow [p, A; \; 1-p, B] \succeq [q, A; \; 1-q, B])$$

- Theorem: Rational preferences imply behavior describable as maximization of expected utility

MEU Principle

- Theorem:
  - [Ramsey, 1931; von Neumann & Morgenstern, 1944]
  - Given any preferences satisfying these constraints, there exists a real-valued function U such that:

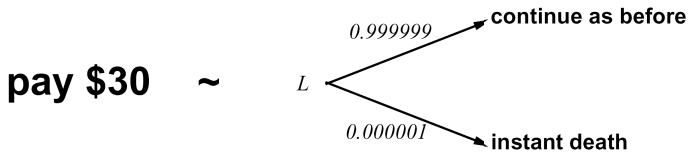
$$U(A) \geq U(B) \;\; \Leftrightarrow \;\; A \succeq B$$

$$U([p_1, S_1; \; \ldots \; ; \; p_n, S_n]) = \sum_i p_i U(S_i)$$

- Maximum expected likelihood (MEU) principle:
  - Choose the action that maximizes expected utility
  - Note: an agent can be entirely rational (consistent with MEU) without ever representing or manipulating utilities and probabilities
  - E.g., a lookup table for perfect tictactoe, reflex vacuum cleaner

Human Utilities

- Utilities map states to real numbers. Which numbers?
- Standard approach to assessment of human utilities:
  - Compare a state A to a standard lottery L_p between
    - ``best possible prize" u_+ with probability p
    - ``worst possible catastrophe" u_- with probability 1-p
  - Adjust lottery probability p until A ~ L_p
  - Resulting p is a utility in [0,1]



Utility Scales

- Normalized utilities: $u_+ = 1.0$, $u_- = 0.0$
- Micromorts: one-millionth chance of death, useful for paying to reduce product risks, etc.
- QALYs: quality-adjusted life years, useful for medical decisions involving substantial risk
- Note: behavior is invariant under positive linear transformation

$$U'(x) = k_1 U(x) + k_2 \quad \text{where } k_1 > 0$$

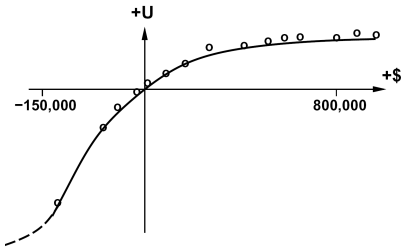
- With deterministic prizes only (no lottery choices), only ordinal utility can be determined, i.e., total order on prizes

Example: Insurance

- Consider the lottery [0.5,$1000; 0.5,$0]
  - What is its expected monetary value? ($500)
  - What is its certainty equivalent?
    - Monetary value acceptable in lieu of lottery
    - $400 for most people
  - Difference of $100 is the insurance premium
    - There's an insurance industry because people will pay to reduce their risk
    - If everyone were risk-prone, no insurance needed!

Money

- Money does not behave as a utility function
- Given a lottery L:
  - Define expected monetary value EMV(L)
  - Usually U(L) < U(EMV(L))
  - I.e., people are risk-averse
- Utility curve: for what probability p am I indifferent between:
  - A prize x
  - A lottery [p,$M; (1-p),$0] for large M?
- Typical empirical data, extrapolated with risk-prone behavior:



Example: Human Rationality?

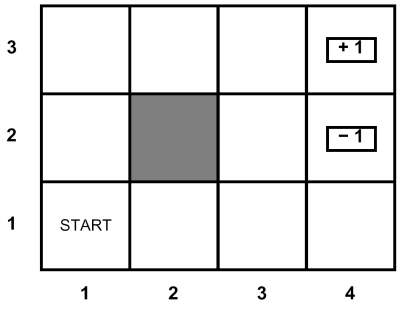
- Famous example of Allais (1953)
  - A: [0.8,$4k; 0.2,$0]
  - B: [1.0,$3k; 0.0,$0]
  - C: [0.2,$4k; 0.8,$0]
  - D: [0.25,$3k; 0.75,$0]
- Most people prefer B > A, C > D
- But if U($0) = 0, then
  - B > A $\Rightarrow$ U($3k) > 0.8 U($4k)
  - C > D $\Rightarrow$ 0.8 U($4k) > U($3k)

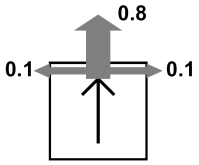
Reinforcement Learning

- [DEMOS]
- Basic idea:
  - Receive feedback in the form of rewards
  - Agent's utility is defined by the reward function
  - Must learn to act so as to maximize expected rewards
  - Change the rewards, change the learned behavior
- Examples:
  - Playing a game, reward at the end for winning / losing
  - Vacuuming a house, reward for each piece of dirt picked up
  - Automated taxi, reward for each passenger delivered

Markov Decision Processes

- An MDP is defined by:
  - A set of states $s \in S$
  - A set of actions $a \in A$
  - A transition function T(s,a,s')
    - Prob that a from s leads to s'
    - i.e., P(s' | s,a)
    - Also called the model
  - A reward function R(s, a, s')
    - Sometimes just R(s) or R(s')
  - A start state (or distribution)
  - Maybe a terminal state
- MDPs are a family of non-deterministic search problems
  - Reinforcement learning: MDPs where we don't know the transition or reward functions

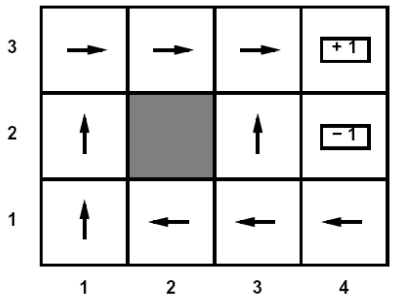




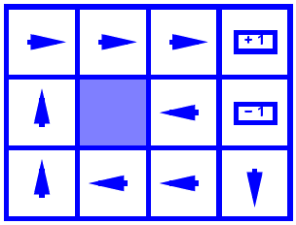
Solving MDPs

- In deterministic single-agent search problem, want an optimal plan, or sequence of actions, from start to a goal
- In an MDP, we want an optimal policy $\pi(s)$
  - A policy gives an action for each state
  - Optimal policy maximizes expected if followed
  - Defines a reflex agent

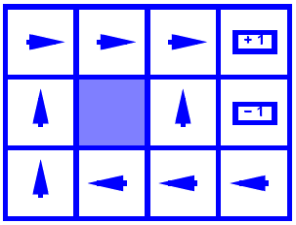
Optimal policy when R(s, a, s') = -0.04 for all non-terminals s



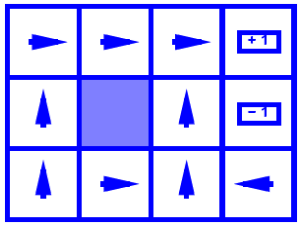
Example Optimal Policies



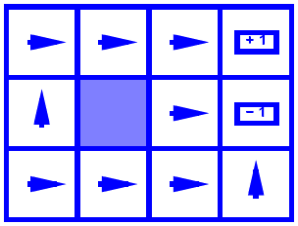
R(s) = -0.01



R(s) = -0.03



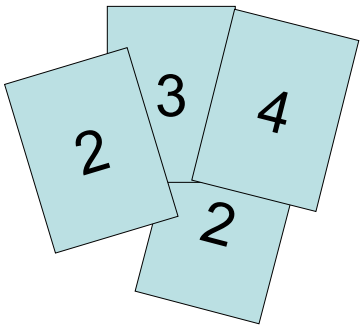
R(s) = -0.4



R(s) = -2.0

Example: High-Low

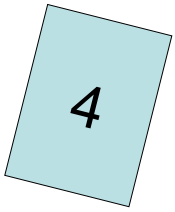
- Three card types: 2, 3, 4
- Infinite deck, twice as many 2's
- Start with 3 showing
- After each card, you say "high" or "low"
- New card is flipped
- If you're right, you win the points shown on the new card
- Ties are no-ops
- If you're wrong, game ends



- Differences from expectimax:
  - #1: get rewards as you go
  - #2: you might play forever!

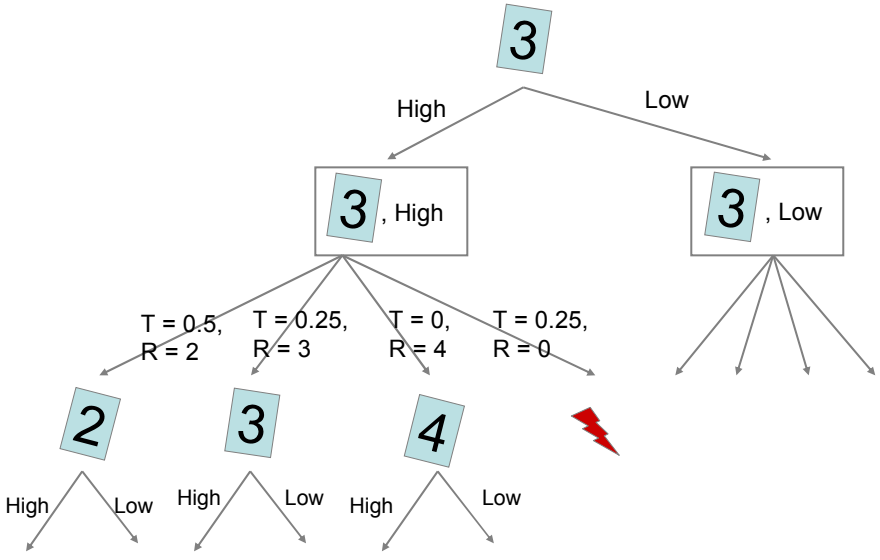
High-Low

- States: 2, 3, 4, done
- Actions: High, Low
- Model: T(s, a, s'):
  - P(s'=done | 4, High) = 3/4
  - P(s'=2 | 4, High) = 0
  - P(s'=3 | 4, High) = 0
  - P(s'=4 | 4, High) = 1/4
  - P(s'=done | 4, Low) = 0
  - P(s'=2 | 4, Low) = 1/2
  - P(s'=3 | 4, Low) = 1/4
  - P(s'=4 | 4, Low) = 1/4
  - …
- Rewards: R(s, a, s'):
  - Number shown on s' if $s \neq s'$
  - 0 otherwise
- Start: 3



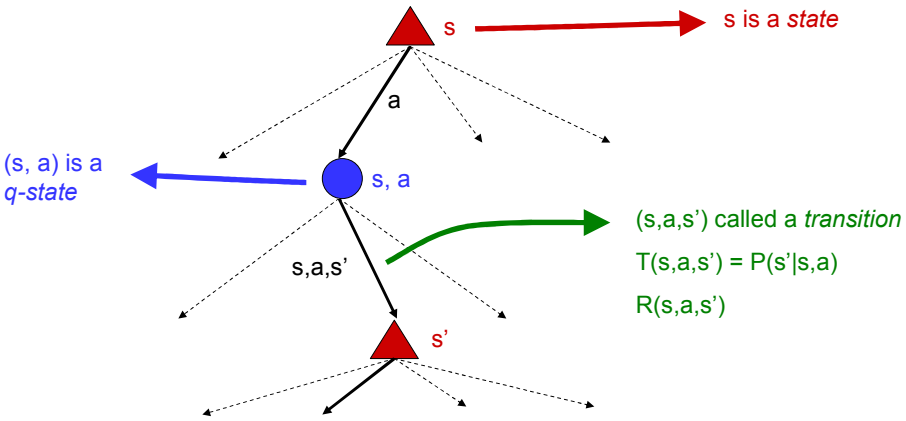
Note: could choose actions with search. How?

Example: High-Low



MDP Search Trees

- Each MDP state gives an expectimax-like search tree



Utilities of Sequences

- In order to formalize optimality of a policy, need to understand utilities of sequences of rewards
- Typically consider stationary preferences:

$$[r, r_0, r_1, r_2, \ldots] \succ [r, r'_0, r'_1, r'_2, \ldots]$$
$$\Leftrightarrow$$
$$[r_0, r_1, r_2, \ldots] \succ [r'_0, r'_1, r'_2, \ldots]$$

Assuming that reward depends only on state for these slides!

- Theorem: only two ways to define stationary utilities
  - Additive utility:

$$V([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \cdots$$

  - Discounted utility:

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

Infinite Utilities?!

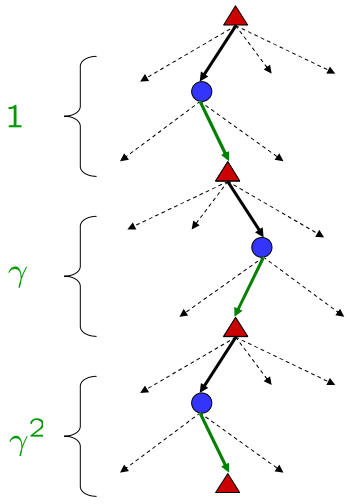
- Problem: infinite sequences with infinite rewards
- Solutions:
  - Finite horizon:
    - Terminate after a fixed T steps
    - Gives nonstationary policy (π depends on time left)
  - Absorbing state(s): guarantee that for every policy, agent will eventually "die" (like "done" for High-Low)
  - Discounting: for $0 < \gamma < 1$

$$V([s_0, \ldots s_\infty]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq R_{\max}/(1-\gamma)$$

    - Smaller γ means smaller "horizon" – shorter term focus

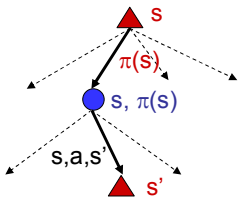
Discounting

- Typically discount rewards by $\gamma < 1$ each time step
  - Sooner rewards have higher utility than later rewards
  - Also helps the algorithms converge



Utilities of States

- Fundamental operation: compute the utility of a state s
- Define the utility of a state s, under a fixed policy π:
  $V^{\pi}(s)$ = expected total discounted rewards (return) starting in s and following π
- Recursive relation (one-step look-ahead):



$$V^{\pi}(s) = \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^{\pi}(s')]$$

Policy Evaluation

- How do we calculate the V's for a fixed policy?
- Idea one: turn recursive equations into updates

$$V_0^{\pi}(s) = 0$$

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_i^{\pi}(s')]$$

- Idea two: it's just a linear system, solve with Matlab (or whatever)

Example: High-Low

- Policy: always say “high”
- Iterative updates:

$$V_0 = \{2:0, \quad 3:0, \quad 4:0, \quad d:0\}$$

$$V_1(2) = \tfrac{1}{2}(R(2,H,2) + V_0(2)) + \tfrac{1}{4}(R(2,H,3) + V_0(3)) +$$
$$\tfrac{1}{4}(R(2,H,4) + V_0(4)) + \ 0(R(2,H,d) + V_0(d))$$

$$V_1(2) = \tfrac{1}{2}(0+0) + \tfrac{1}{4}(3+0) + \tfrac{1}{4}(4+0) + 0(0+0)$$

$$V_1(2) = \tfrac{7}{4}$$

$$V_1 = \{2:\frac{7}{4}, \quad 3:1, \quad 4:0, \quad d:0\}$$

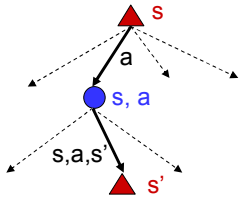
Equivalent to doing fixed depth search and plugging in zero at leaves

Example: GridWorld

- [DEMO]

Q-Functions

- To simplify things, introduce a q-value, for a state and action (q-state) under a policy
  - Utility of starting in state s, taking action a, then following π thereafter



$$Q^{\pi}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^{\pi}(s')]$$

$$V^{\pi}(s) = Q^{\pi}(s,\pi(s))$$

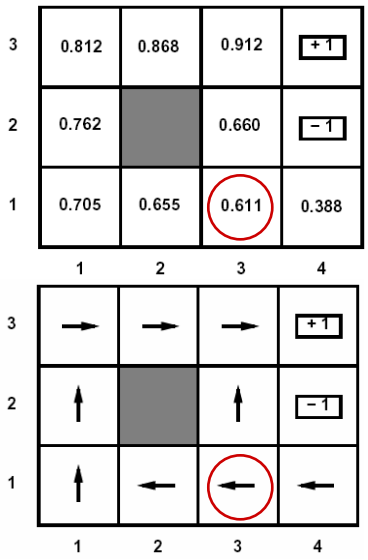
$$Q^{\pi}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma Q^{\pi}(s',\pi(s'))]$$

Optimal Utilities

- Goal: calculate the optimal utility of each state

  V*(s) = expected (discounted) rewards with optimal actions

- Why?
  - Given optimal utilities, MEU lets us compute the optimal policy



Practice: Computing Actions

- Which action should we chose from state s:
  - Given optimal q-values Q?

$$\arg\max_a Q^*(s,a)$$

  - Given optimal values V?

$$\arg\max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$