Decision Theory

Philipp Koehn

presented by Gaurav Kumar

13 April 2017

Outline

- Rational preferences
- Utilities
- Multiattribute utilities
- Decision networks
- Value of information
- Sequential decision problems
- Value iteration
- Policy iteration

preferences

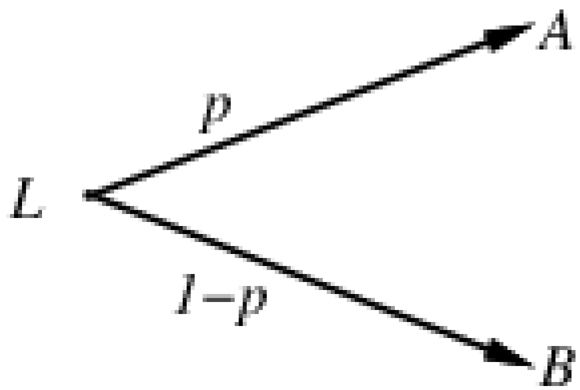
Preferences

- An agent chooses among prizes (A, B, etc.)

- Notation:

$A \succ B$	A preferred to B
$A \sim B$	indifference between A and B
$A \succsim B$	B not preferred to A

- Lottery $L = [p, A;\ (1-p), B]$, i.e., situations with uncertain prizes



Rational Preferences

- Idea: preferences of a rational agent must obey constraints

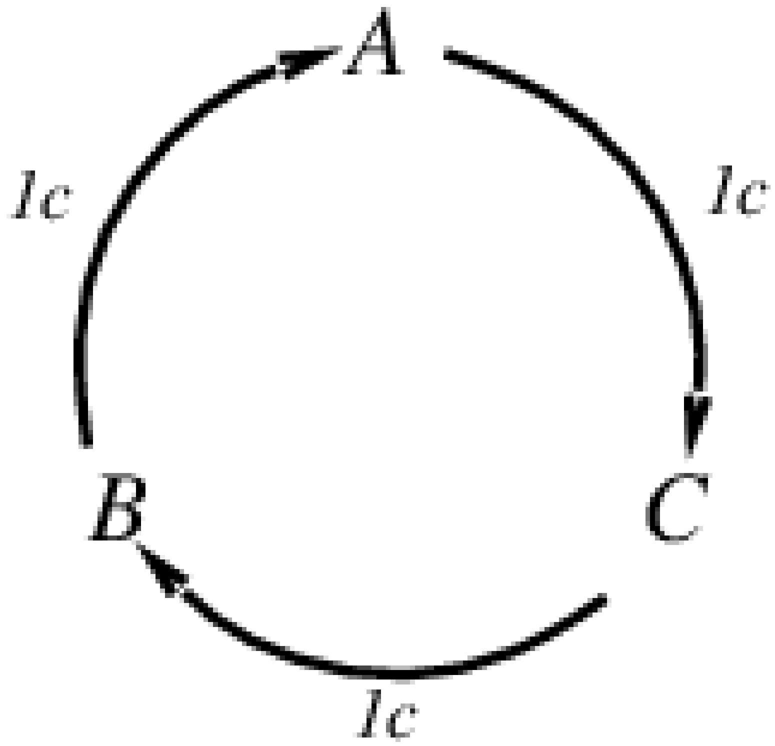
- Rational preferences $\implies$
  behavior describable as maximization of expected utility

- Constraints:
  Orderability
  $(A \succ B) \vee (B \succ A) \vee (A \sim B)$
  Transitivity
  $(A \succ B) \wedge (B \succ C) \implies (A \succ C)$
  Continuity
  $A \succ B \succ C \implies \exists p \; [p, A;\; 1-p, C] \sim B$
  Substitutability
  $A \sim B \implies [p, A;\; 1-p, C] \sim [p, B; 1-p, C]$
  Monotonicity
  $A \succ B \implies (p \geq q \Leftrightarrow [p, A;\; 1-p, B] \succsim [q, A;\; 1-q, B])$

Rational Preferences

- Violating the constraints leads to self-evident irrationality

- For example: an agent with intransitive preferences can be induced to give away all its money



- If $B \succ C$, then an agent who has C would pay (say) 1 cent to get B

- If $A \succ B$, then an agent who has B would pay (say) 1 cent to get A

- If $C \succ A$, then an agent who has A would pay (say) 1 cent to get C

Maximizing Expected Utility

- **Theorem** (Ramsey, 1931; von Neumann and Morgenstern, 1944):

  Given preferences satisfying the constraints
  there exists a real-valued function U such that

  $$U(A) \geq U(B) \quad \Leftrightarrow \quad A \succsim B$$
  $$U([p_1, S_1;\ \ldots\ ;\ p_n, S_n]) = \sum_i\ p_i U(S_i)$$

- 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

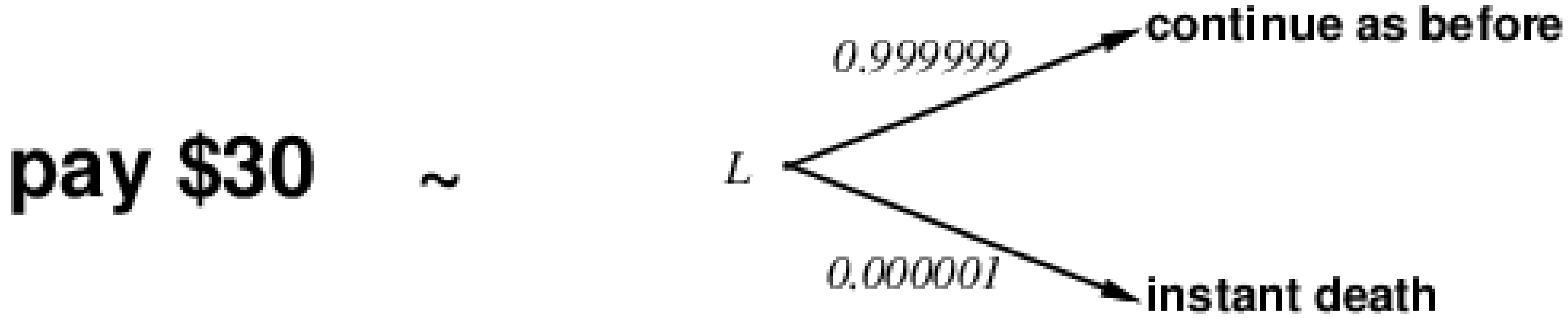
- E.g., a lookup table for perfect tictactoe

utilities

Utilities

- Utilities map states to real numbers. Which numbers?

- Standard approach to assessment of human utilities
  - compare a given state A to a standard lottery L_p that has
    - "best possible prize" $u_\top$ with probability p
    - "worst possible catastrophe" $u_\bot$ with probability $(1-p)$
  - adjust lottery probability p until $A \sim L_p$



Utility Scales

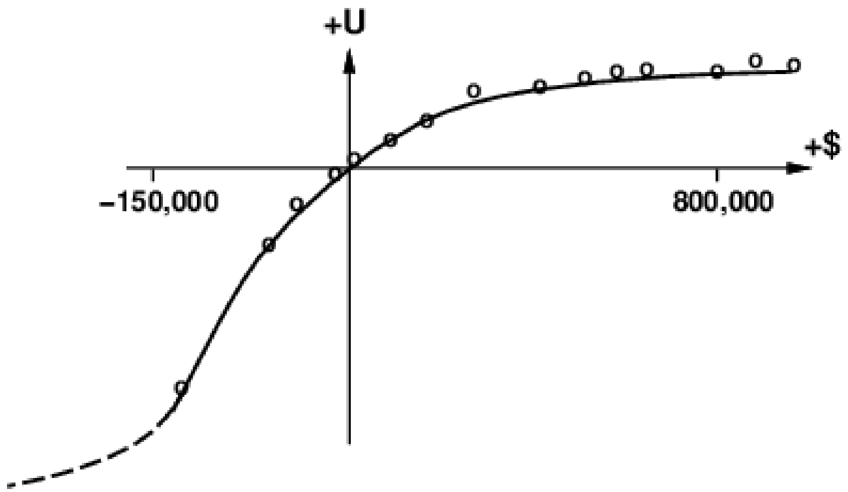
- Normalized utilities: $u_\top = 1.0$, $u_\bot = 0.0$
- Micromorts: one-millionth chance of death
  useful for Russian roulette, paying to reduce product risks, etc.
- QALYs: quality-adjusted life years
  useful for medical decisions involving substantial risk
- Note: behavior is **invariant** w.r.t. +ve 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

Money

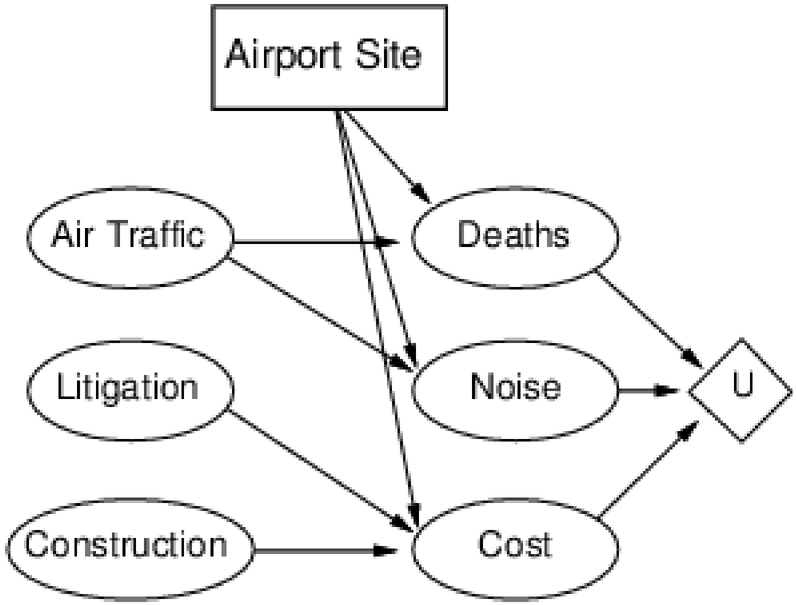
- Money does **not** behave as a utility function
- Given a lottery L with expected monetary value $EMV(L)$, usually $U(L) < U(EMV(L))$, i.e., people are risk-averse
- Utility curve: for what probability p am I indifferent between a prize x and a lottery $[p, \$M;\ (1-p), \$0]$ for large M?
- Typical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



decision networks

Decision Networks

- Add action nodes and utility nodes to belief networks to enable rational decision making



- Algorithm:

  For each value of action node

  compute expected value of utility node given action, evidence

  Return MEU action

Multiattribute Utility

- How can we handle utility functions of many variables $X_1 \ldots X_n$?
  E.g., what is $U(Deaths, Noise, Cost)$?

- How can complex utility functions be assessed from
  preference behaviour?

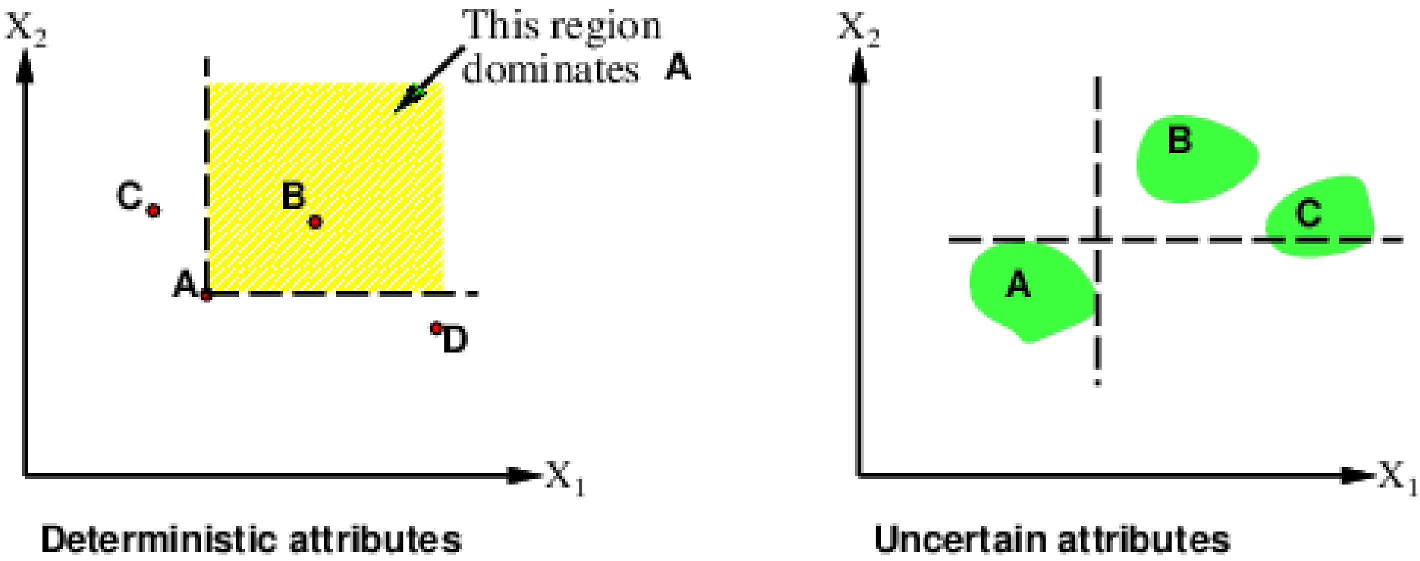
- Idea 1: identify conditions under which decisions can be made without complete identification of $U(x_1, \ldots, x_n)$

- Idea 2: identify various types of **independence** in preferences
  and derive consequent canonical forms for $U(x_1, \ldots, x_n)$

Strict Dominance

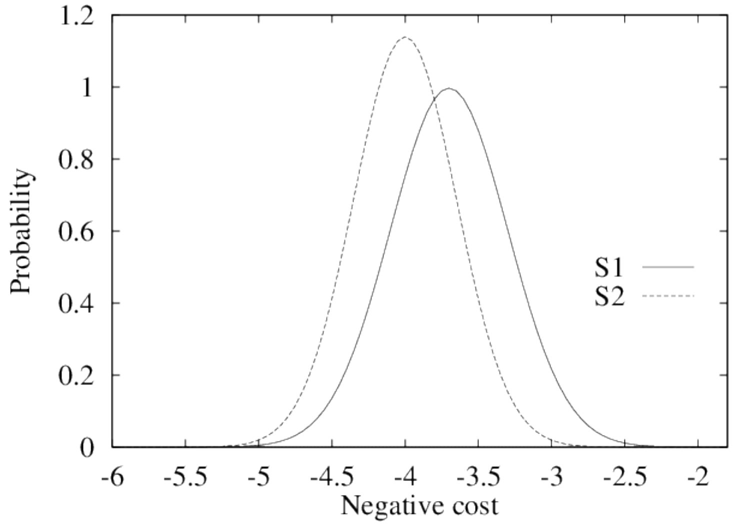
- Typically define attributes such that U is monotonic in each

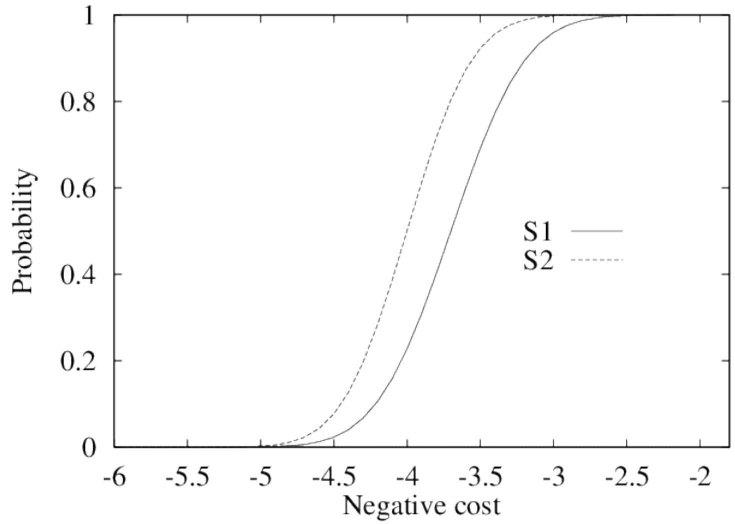
- Strict dominance: choice B strictly dominates choice A iff
  $\forall\, i \;\; X_i(B) \geq X_i(A)$ (and hence $U(B) \geq U(A)$)



- Strict dominance seldom holds in practice

Stochastic Dominance





- Distribution p_1 stochastically dominates distribution p_2 iff
$$\forall\, t \quad \int_{-\infty}^{t} p_1(x)dx \leq \int_{-\infty}^{t} p_2(x)dx$$

- If U is monotonic in x, then A_1 with outcome distribution p_1 stochastically dominates A_2 with outcome distribution p_2:
$$\int_{-\infty}^{\infty} p_1(x)U(x)dx \geq \int_{-\infty}^{\infty} p_2(x)U(x)dx$$
Multiattribute case: stochastic dominance on all attributes $\Longrightarrow$ optimal

Stochastic Dominance

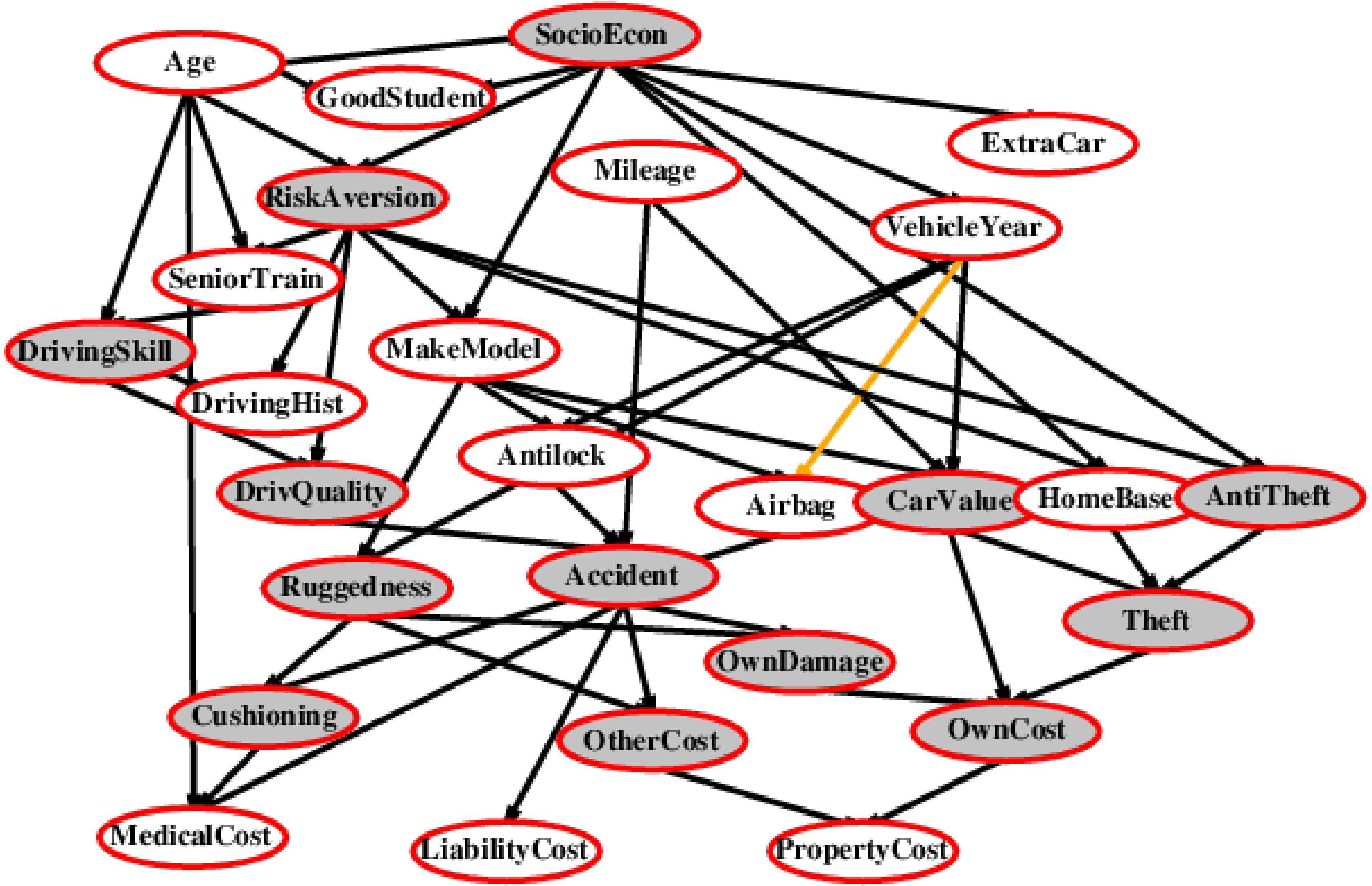
- Stochastic dominance can often be determined without exact distributions using **qualitative** reasoning

- E.g., construction cost increases with distance from city
  S_1 is closer to the city than S_2
  $\implies S_1$ stochastically dominates S_2 on cost

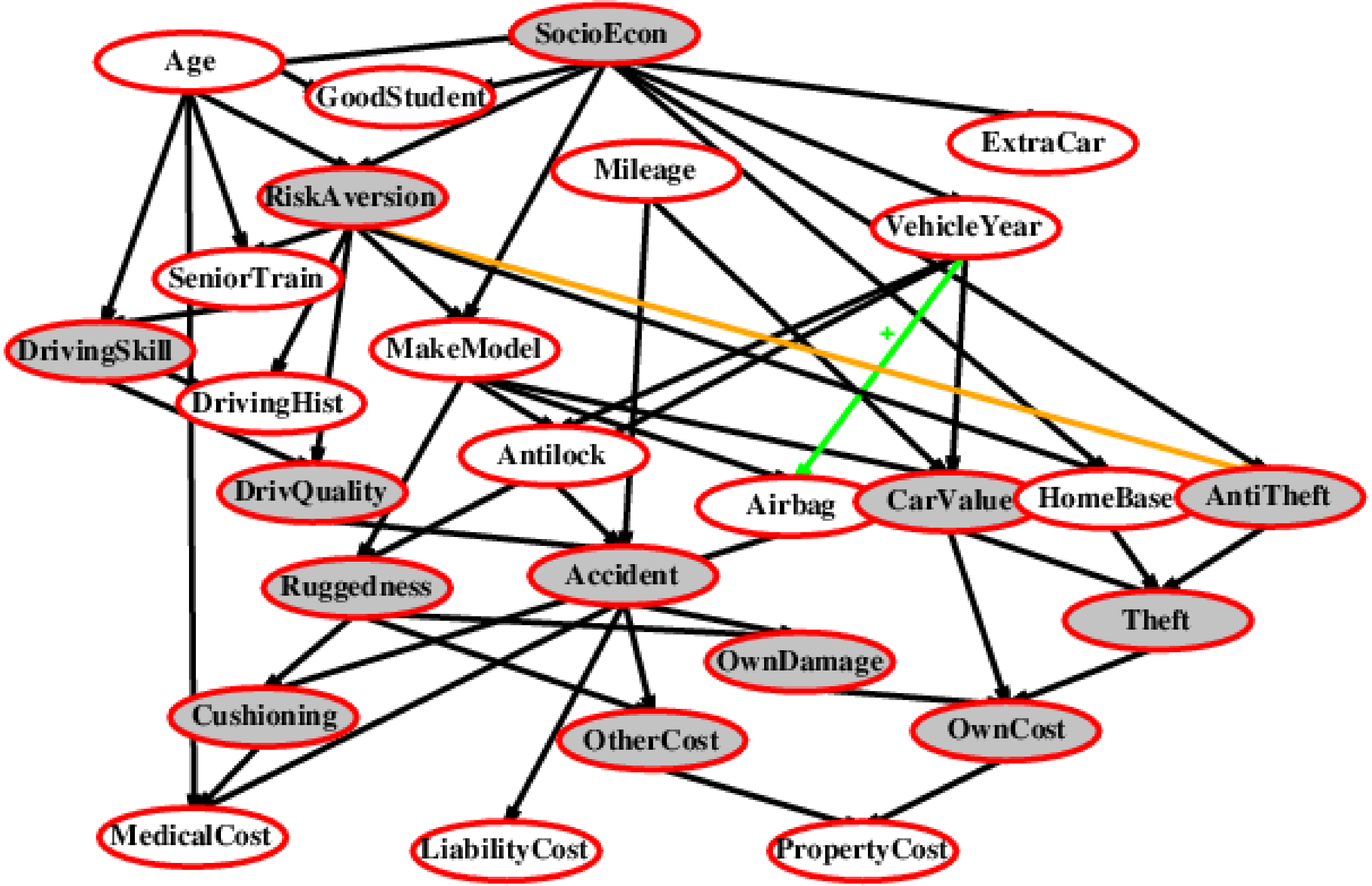
- E.g., injury increases with collision speed

- Can annotate belief networks with stochastic dominance information:
  $X \xrightarrow{+} Y$ (X positively influences Y) means that
  For every value $\mathbf{z}$ of Y's other parents $\mathbf{Z}$
  $\forall\, x_1, x_2\;\; x_1 \geq x_2 \implies \mathbf{P}(Y|x_1, \mathbf{z})$ stochastically dominates $\mathbf{P}(Y|x_2, \mathbf{z})$

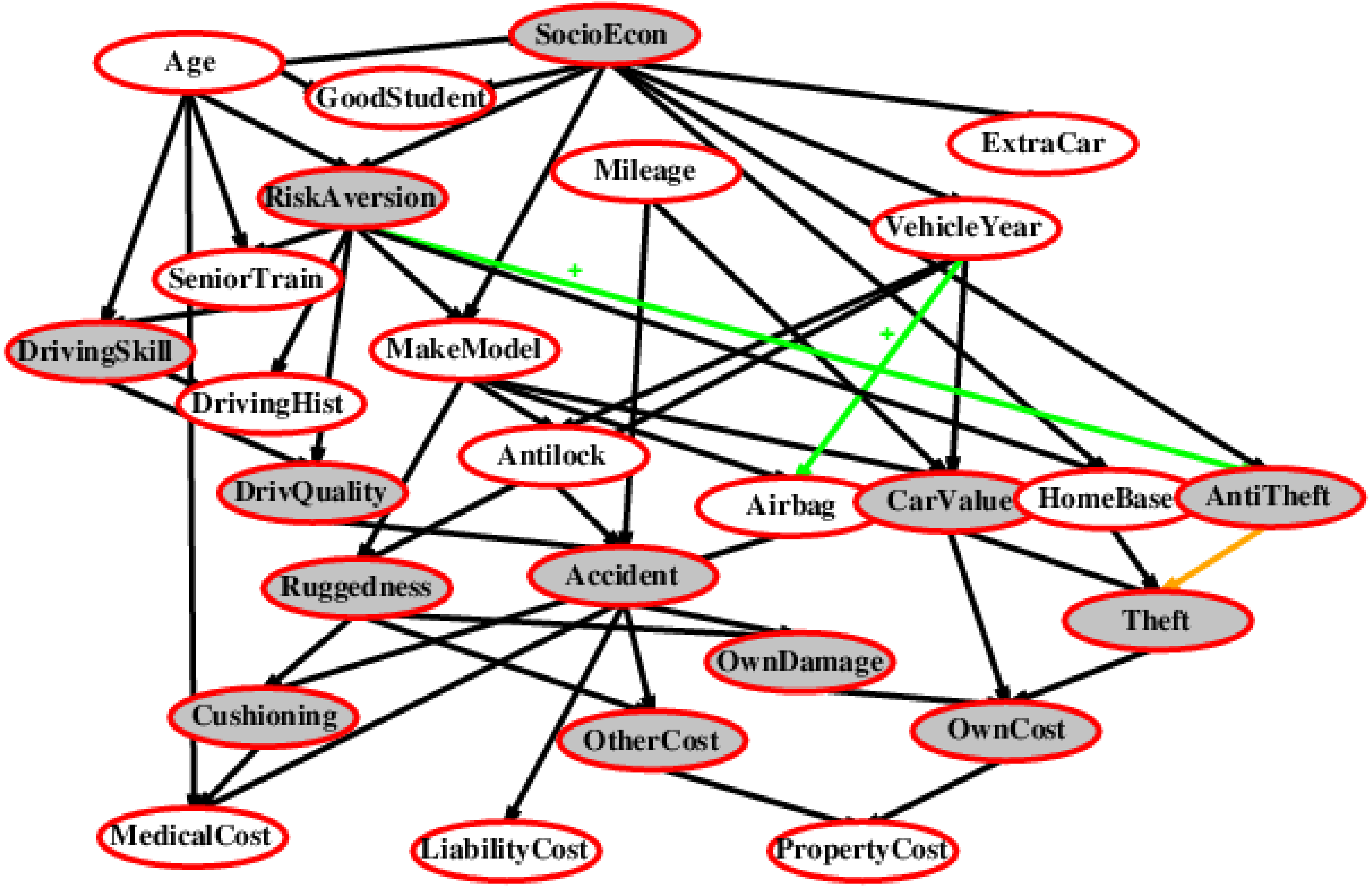
Label the Arcs + or –



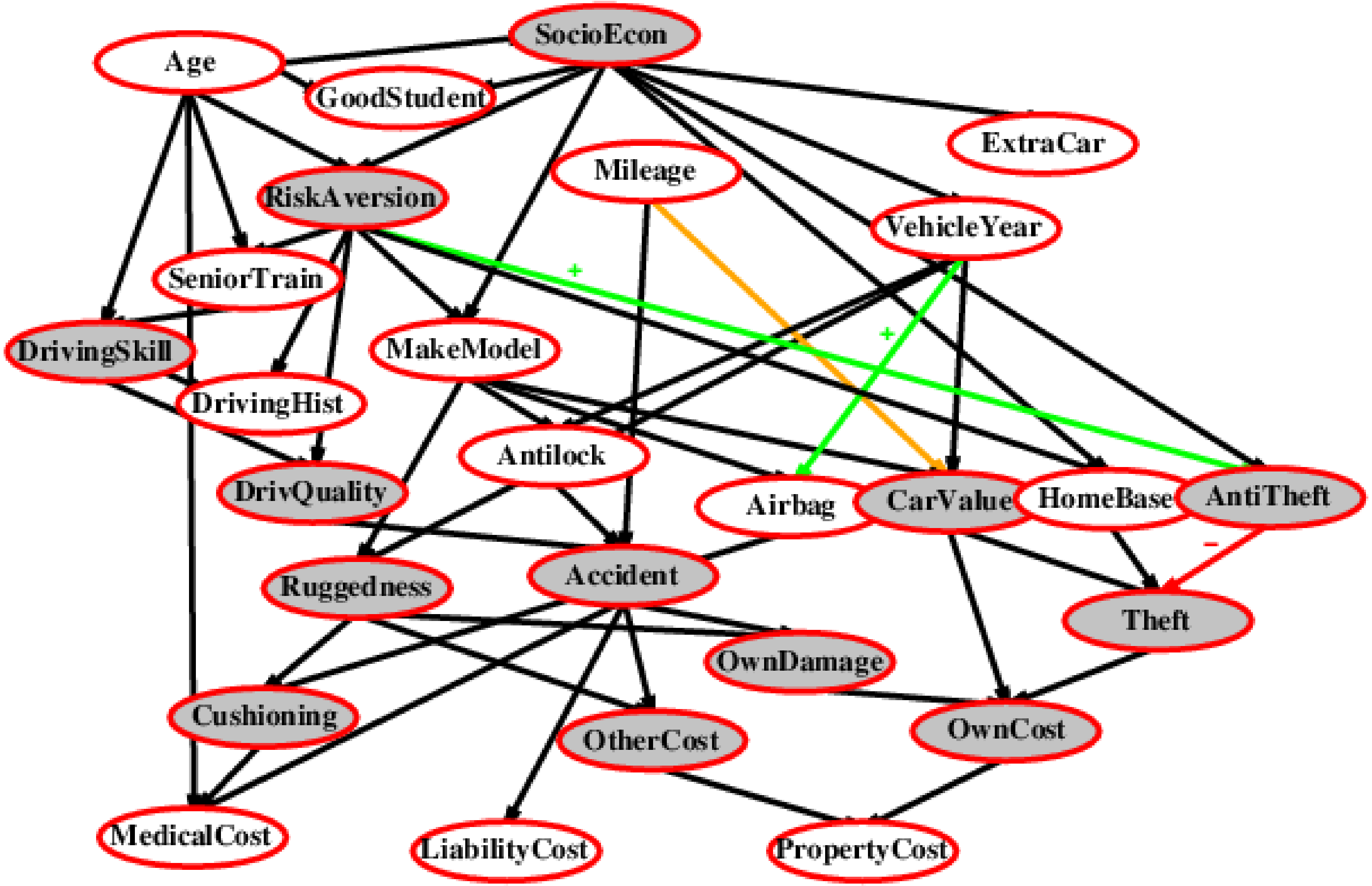
Label the Arcs + or –



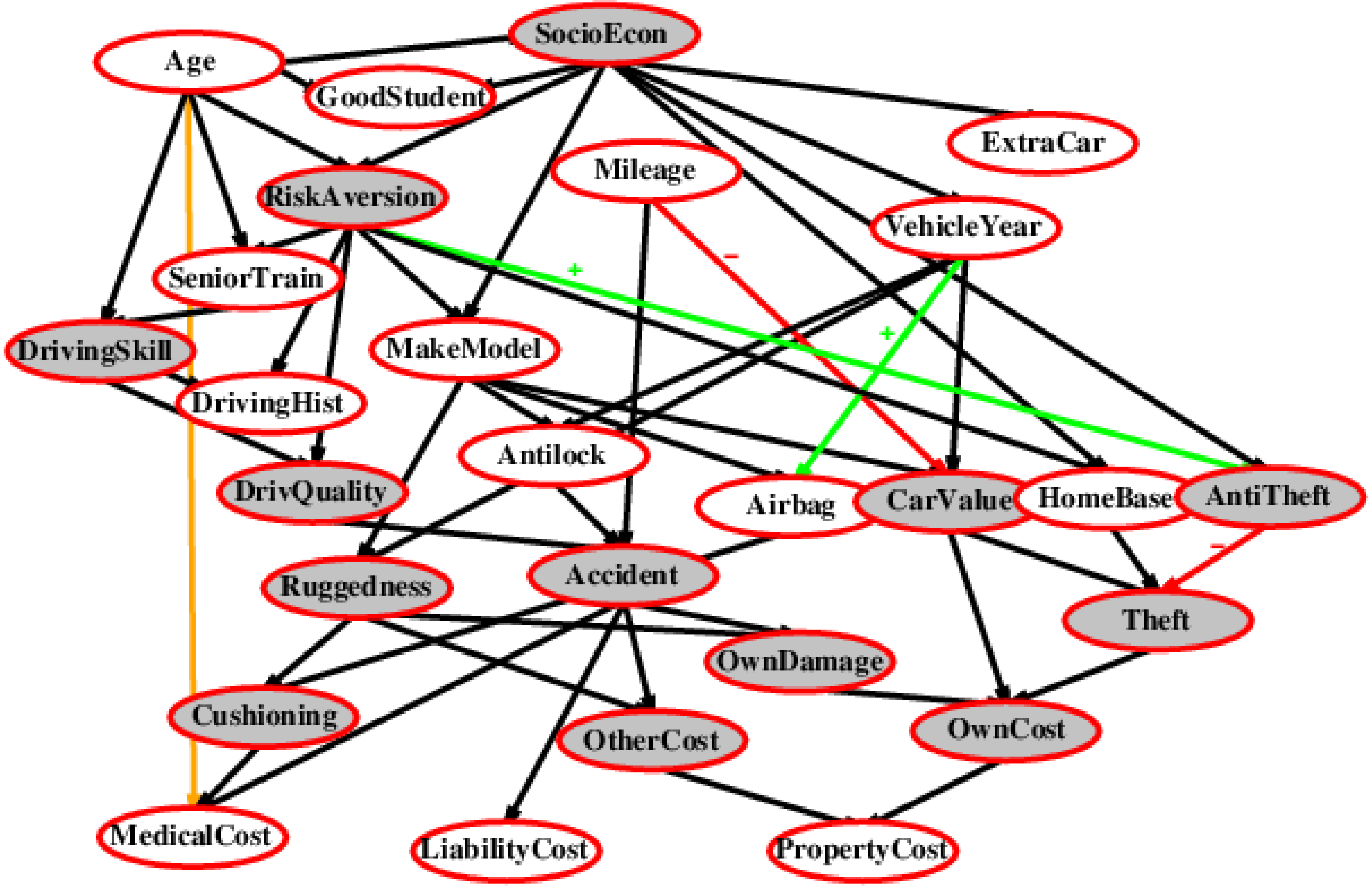
Label the Arcs + or –



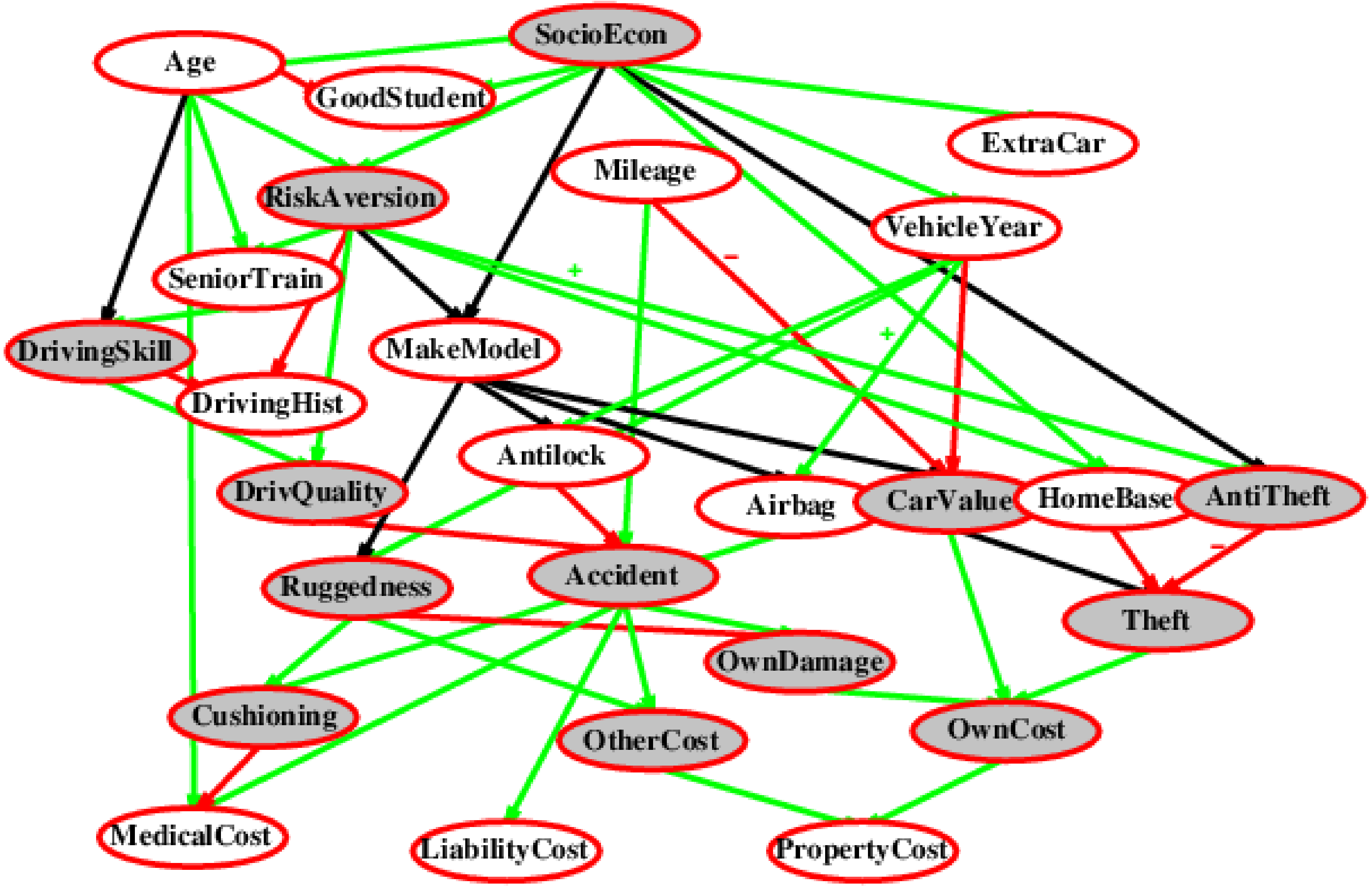
Label the Arcs + or –



Label the Arcs + or –



Label the Arcs + or –



Preference Structure: Deterministic

- X_1 and X_2 preferentially independent of X_3 iff
  preference between $\langle x_1, x_2, x_3 \rangle$ and $\langle x'_1, x'_2, x_3 \rangle$
  does not depend on x_3

- E.g., $\langle Noise, Cost, Safety \rangle$:
  ⟨20,000 suffer, $4.6 billion, 0.06 deaths/mpm⟩ vs.
  ⟨70,000 suffer, $4.2 billion, 0.06 deaths/mpm⟩

- **Theorem** (Leontief, 1947): if every pair of attributes is P.I. of its complement, then every subset of attributes is P.I of its complement: mutual P.I.

- **Theorem** (Debreu, 1960): mutual P.I. $\implies$ $\exists$ additive value function:

$$V(S) = \sum_i V_i(X_i(S))$$

Hence assess n single-attribute functions; often a good approximation

Preference Structure: Stochastic

- Need to consider preferences over lotteries:
  X is utility-independent of **Y** iff
  preferences over lotteries in **X** do not depend on **y**

- Mutual U.I.: each subset is U.I of its complement
  $\implies \exists$ multiplicative utility function:
  $$U = k_1U_1 + k_2U_2 + k_3U_3 + k_1k_2U_1U_2 + k_2k_3U_2U_3 + k_3k_1U_3U_1 + k_1k_2k_3U_1U_2U_3$$

- Routine procedures and software packages for generating preference tests to identify various canonical families of utility functions

value of information

Value of Information

- Idea: compute value of acquiring each possible piece of evidence
  Can be done **directly from decision network**

- Example: buying oil drilling rights
  Two blocks A and B, exactly one has oil, worth k
  Prior probabilities 0.5 each, mutually exclusive
  Current price of each block is $k/2$
  "Consultant" offers accurate survey of A. Fair price?

- Solution: compute expected value of information
  = expected value of best action given the information
  minus expected value of best action without information

- Survey may say "oil in A" or "no oil in A", **prob. 0.5 each** (given!)
  = [$0.5 \times$ value of "buy A" given "oil in A"
  + $0.5 \times$ value of "buy B" given "no oil in A"]
  – 0
  = $(0.5 \times k/2) + (0.5 \times k/2) - 0 = k/2$

General Formula

- Current evidence E, current best action α
- Possible action outcomes S_i, potential new evidence E_j

$$EU(\alpha|E) = \max_a \sum_i U(S_i)\ P(S_i|E,a)$$

- Suppose we knew $E_j = e_{jk}$, then we would choose $\alpha_{e_{jk}}$ s.t.

$$EU(\alpha_{e_{jk}}|E, E_j = e_{jk}) = \max_a \sum_i U(S_i)\ P(S_i|E,a,E_j = e_{jk})$$

- E_j is a random variable whose value is *currently* unknown
- $\implies$ must compute expected gain over all possible values:

$$VPI_E(E_j) = \left(\sum_k P(E_j = e_{jk}|E) EU(\alpha_{e_{jk}}|E, E_j = e_{jk})\right) - EU(\alpha|E)$$

(VPI = value of perfect information)

Properties of VPI

- **Nonnegative**—in **expectation**, not **post hoc**

$$\forall\, j, E \;\; VPI_E(E_j) \geq 0$$

- **Nonadditive**—consider, e.g., obtaining E_j twice

$$VPI_E(E_j, E_k) \neq VPI_E(E_j) + VPI_E(E_k)$$

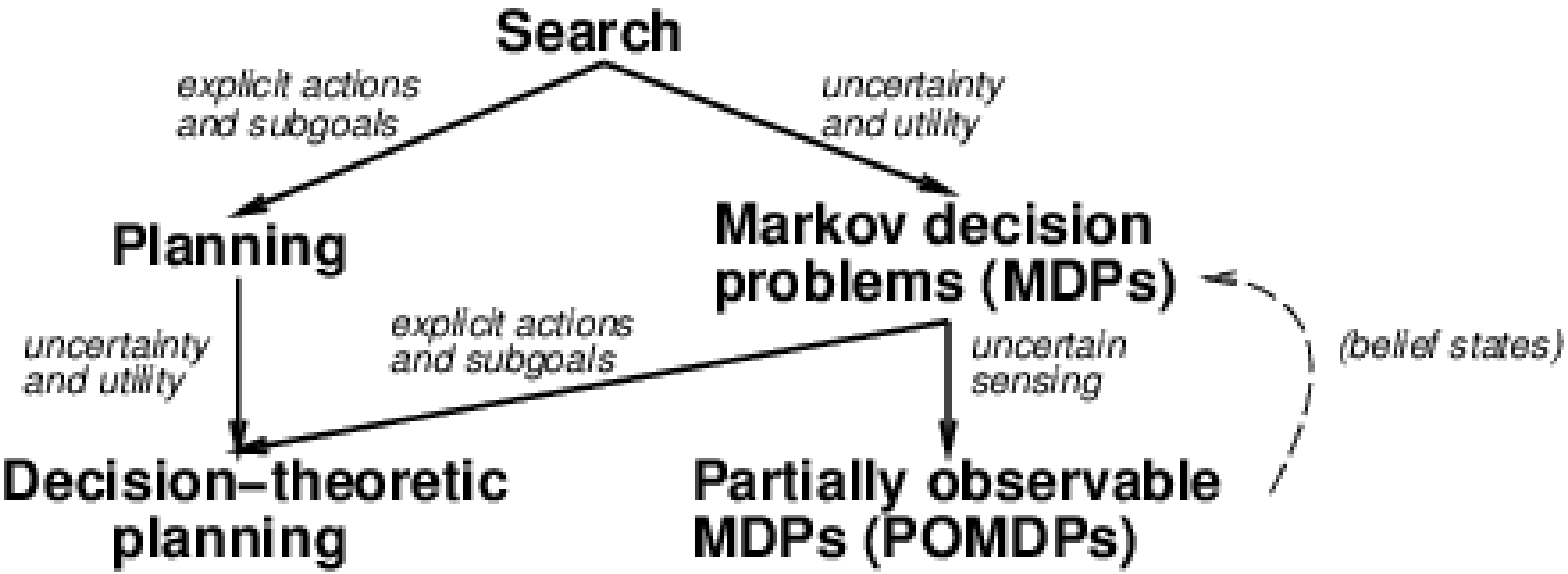
- **Order-independent**

$$VPI_E(E_j, E_k) = VPI_E(E_j) + VPI_{E,E_j}(E_k) = VPI_E(E_k) + VPI_{E,E_k}(E_j)$$

- Note: when more than one piece of evidence can be gathered, maximizing VPI for each to select one is not always optimal
  $\Longrightarrow$ evidence-gathering becomes a **sequential** decision problem

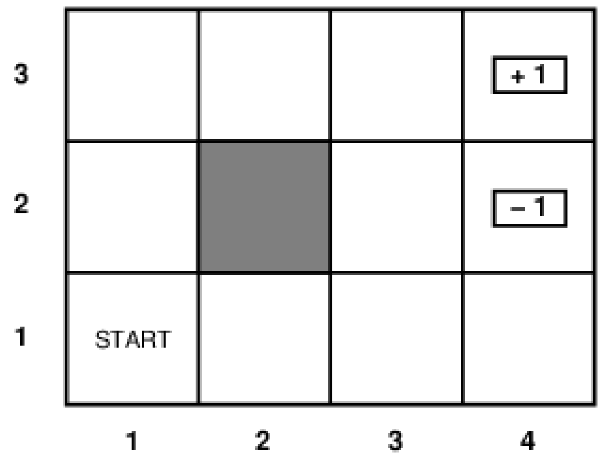
sequential decision problems

Sequential Decision Problems

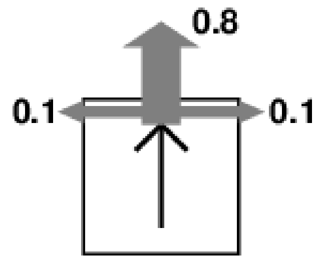


Example Markov Decision Process

State Map



Stochastic Movement



- States $s \in S$, actions $a \in A$

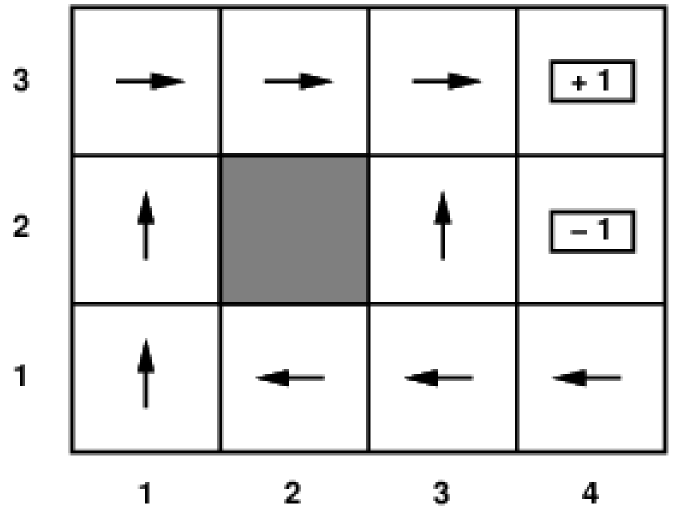
- Model $T(s, a, s') \equiv P(s'|s, a)$ = probability that a in s leads to s'

- Reward function $R(s)$ (or $R(s, a)$, $R(s, a, s')$)

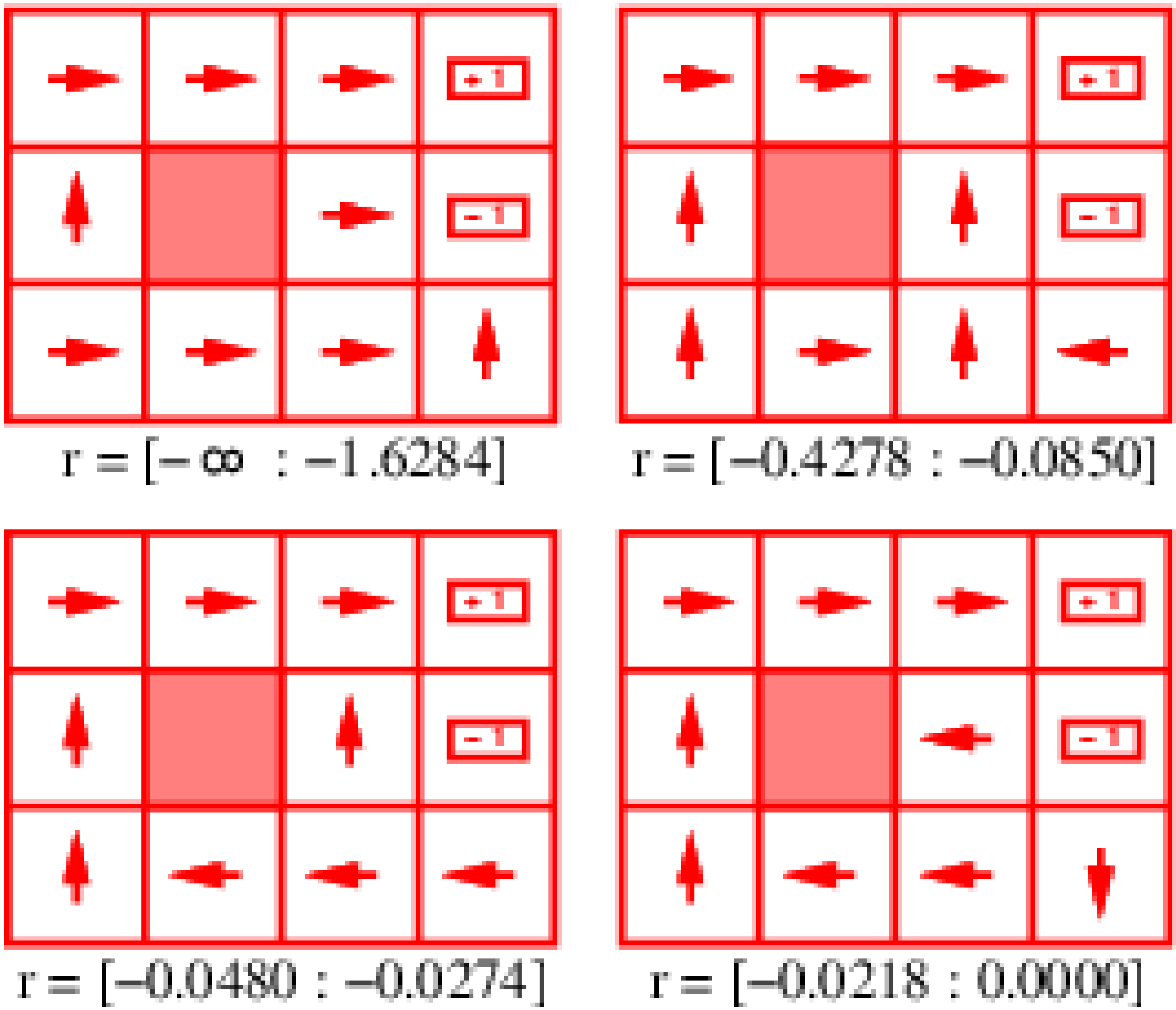
$$= \begin{cases} -0.04 & \text{(small penalty) for nonterminal states} \\ \pm 1 & \text{for terminal states} \end{cases}$$

Solving Markov Decision Processes

- In search problems, aim is to find an optimal *sequence*
- In MDPs, aim is to find an optimal *policy* $\pi(s)$
  i.e., best action for every possible state s
  (because can't predict where one will end up)
- The optimal policy maximizes (say) the *expected sum of rewards*
- Optimal policy when state penalty $R(s)$ is –0.04:



Risk and Reward



Utility of State Sequences

- Need to understand preferences between *sequences* of states

- Typically consider stationary preferences on reward sequences:

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

- There are two ways to combine rewards over time

1. *Additive* utility function:
   $U([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \cdots$

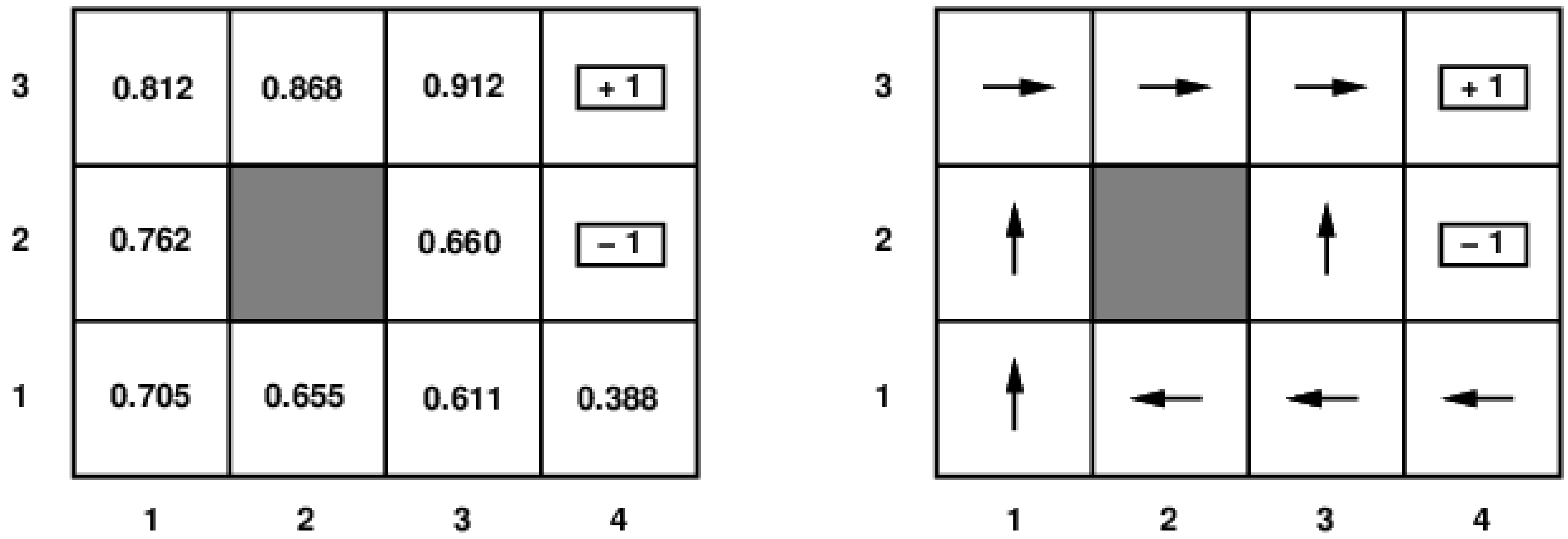
2. *Discounted* utility function:
   $U([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$
   where γ is the discount factor

Utility of States

- Utility of a *state* (a.k.a. its *value*) is defined to be

  $U(s)$ = <u>expected (discounted) sum of rewards (until termination) assuming optimal actions</u>

- Given the utilities of the states, choosing the best action is just MEU: maximize the expected utility of the immediate successors



Utilities

- Problem: infinite lifetimes $\implies$ additive utilities are infinite

- 1) **Finite horizon**: termination at a *fixed time* T
  $\implies$ **nonstationary** policy: $\pi(s)$ depends on time left

- 2) **Absorbing state(s)**: w/ prob. 1, agent eventually "dies" for any π
  $\implies$ expected utility of every state is finite

- 3) **Discounting**: assuming $\gamma < 1$, $R(s) \le R_{\max}$,

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

  Smaller $\gamma \Rightarrow$ shorter horizon

- 4) Maximize **system gain** = average reward per time step
  Theorem: optimal policy has constant gain after initial transient
  E.g., taxi driver's daily scheme cruising for passengers

Dynamic Programming: Bellman Equation

- Definition of utility of states leads to a simple relationship among utilities of neighboring states:

- **Expected sum of rewards**
  = current reward
  + $\gamma \times$ expected sum of rewards after taking best action

- Bellman equation (1957):

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

- $U(1,1) = -0.04$
  $+ \gamma \max\{0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1),$ *up*
  $0.9U(1,1) + 0.1U(1,2)$ *left*
  $0.9U(1,1) + 0.1U(2,1)$ *down*
  $0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1)\}$ *right*

- One equation per state = n **nonlinear** equations in n unknowns

inference algorithms

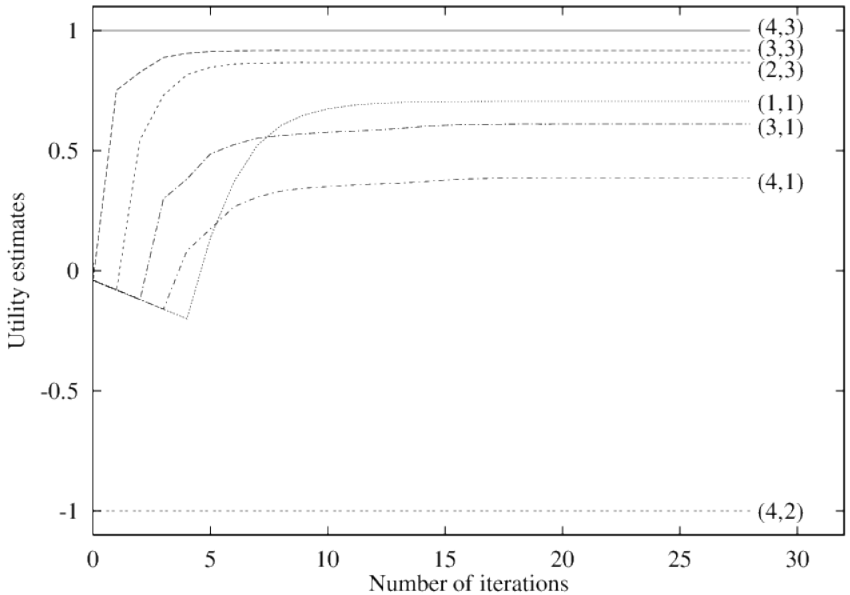
Value Iteration Algorithm

- Idea: Start with arbitrary utility values
  Update to make them locally consistent with Bellman eqn.
  Everywhere locally consistent ⇒ global optimality

- Repeat for every s simultaneously until "no change"

$$U(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} U(s')T(s,a,s') \qquad \text{for all } s$$

- Example:
  utility estimates
  for selected states



Policy Iteration

- Howard, 1960: search for optimal policy and utility values simultaneously

- Algorithm:

  $\pi \leftarrow$ an arbitrary initial policy
  repeat until no change in π
  compute utilities given π
  update π as if utilities were correct (i.e., local MEU)

- To compute utilities given a fixed π (value determination):

$$U(s) = R(s) + \gamma \sum_{s'} U(s')T(s,\pi(s),s') \qquad \text{for all } s$$

- i.e., n simultaneous linear equations in n unknowns, solve in $O(n^3)$

Modified Policy Iteration

- Policy iteration often converges in few iterations, but each is expensive
- Idea: use a few steps of value iteration (but with π fixed)
  starting from the value function produced the last time
  to produce an approximate value determination step.
- Often converges much faster than pure VI or PI
- Leads to much more general algorithms where Bellman value updates and Howard policy updates can be performed locally in any order
- Reinforcement learning algorithms operate by performing such updates based on the observed transitions made in an initially unknown environment

Partial Observability

- POMDP has an <u>observation model</u> $O(s, e)$ defining the probability that the agent obtains evidence e when in state s

- Agent does not know which state it is in
  $\implies$ makes no sense to talk about policy $\pi(s)$!!

- <u>Theorem</u> (Astrom, 1965): the optimal policy in a POMDP is a function
  $\pi(b)$ where b is the <u>belief state</u> (probability distribution over states)

- Can convert a POMDP into an MDP in belief-state space, where
  $T(b, a, b')$ is the probability that the new belief state is b'
  given that the current belief state is b and the agent does a.
  I.e., essentially a filtering update step

Partial Observability

- Solutions automatically include information-gathering behavior

- If there are n states, b is an n-dimensional real-valued vector
  $\Longrightarrow$ solving POMDPs is very (actually, PSPACE-) hard!

- The real world is a POMDP (with initially unknown T and O)