Advanced Topics Data Science and AI
Automated Planning and Acting

Standard Decision Making

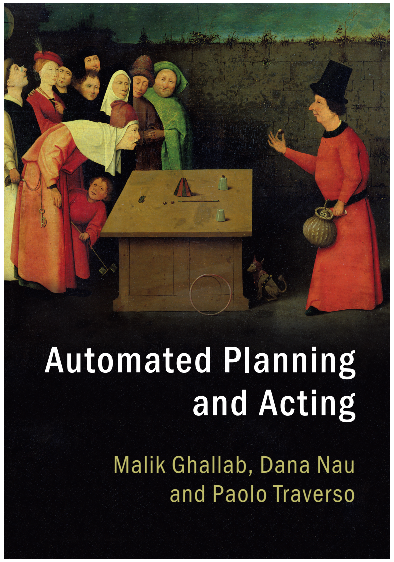
Tanya Braun

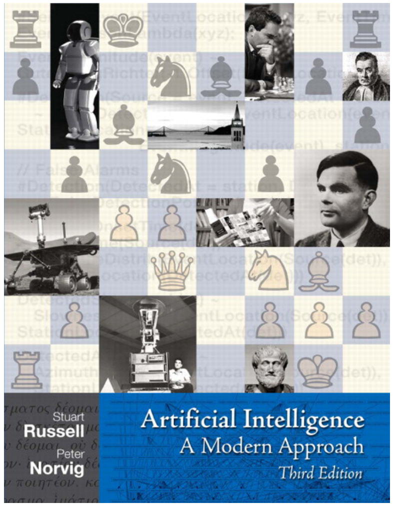
UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

Content

Literature

- We now switch from
  - Automated Planning and Acting
    - Malik Ghallab, Dana Nau, Paolo Traverso
    - Main source
- to
  - Artificial Intelligence:
    A Modern Approach (3rd ed.)
    - Stuart Russell, Peter Norvig
    - Decision theory
      - Ch. 16 + 17





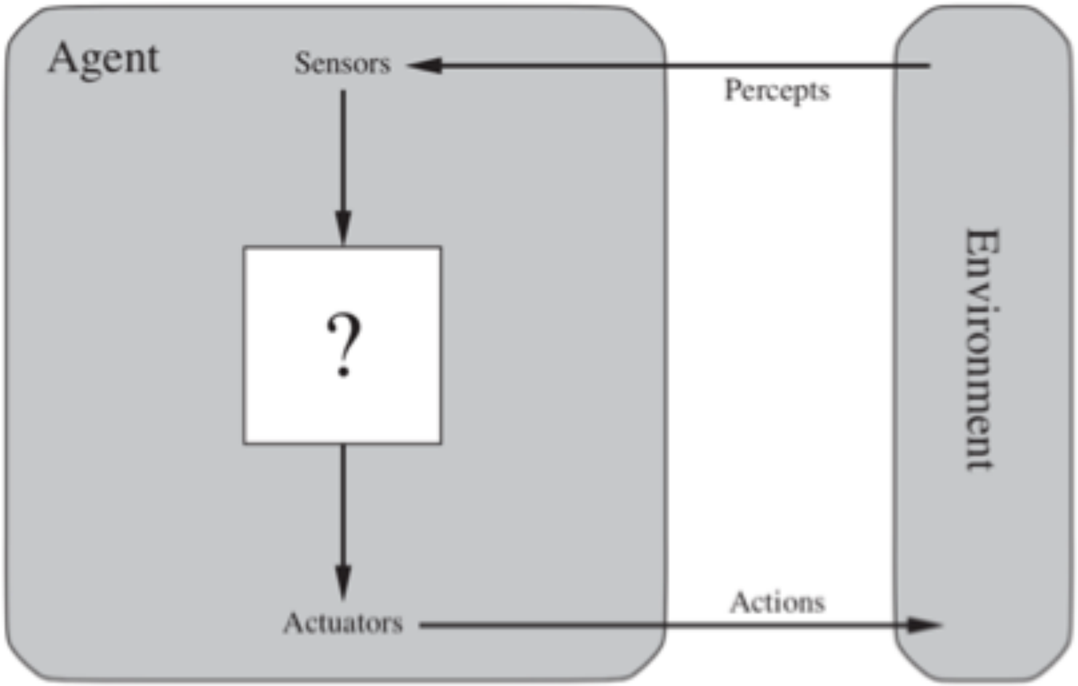
The first half of this lecture covers utility theory, which is also part of the module Intelligent Agents.

Acknowledgements

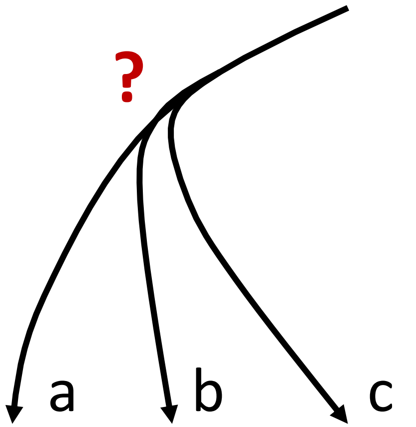
- Material from Lise Getoor, Jean-Claude Latombe, Daphne Koller, and Stuart Russell
- Compiled by Ralf Möller

Decision Making under Uncertainty

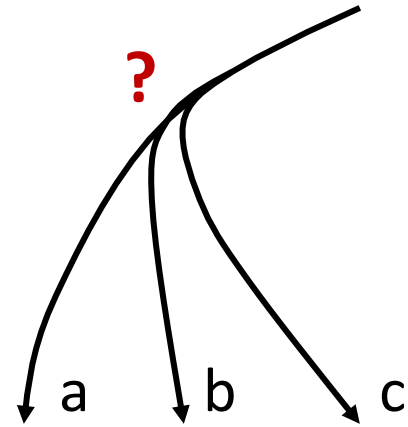
- Many environments have multiple possible outcomes
- Some of these outcomes may be good;
  others may be bad
- Some may be very likely;
  others unlikely



Nondeterministic vs. Probabilistic Uncertainty



Nondeterministic model



Probabilistic model

- $\{a, b, c\}$
- Decision that is
  best for worst case

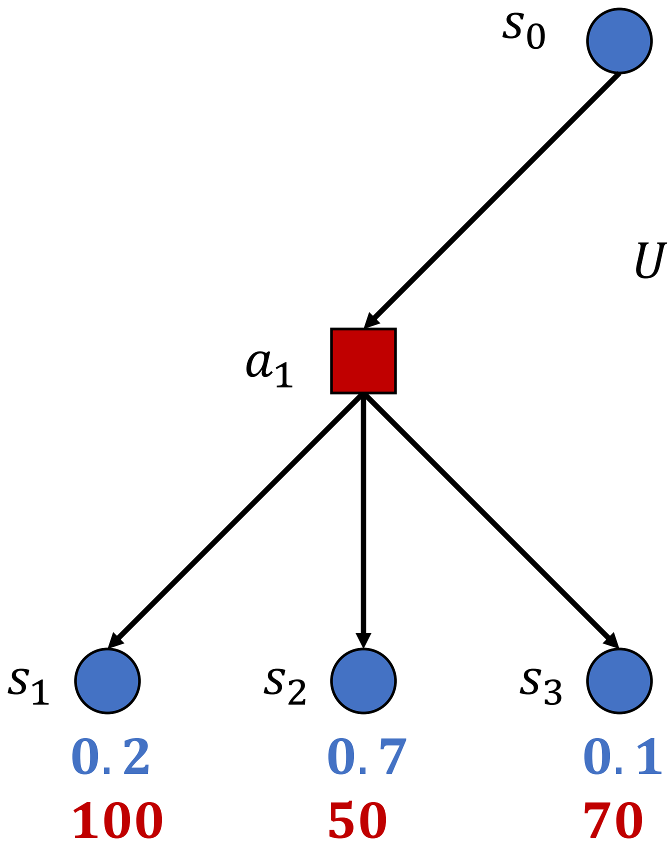
- $\{a(p_a), b(p_b), c(p_c)\}$
- Decision that
  maximises expected
  utility value

Expected Utility

- Random variable X with n range values $x_1, \dots, x_n$ and distribution $(p_1, \dots, p_n)$
  - E.g.: X is the state reached after doing an action $A = a$ under uncertainty
- Function U of X
  - E.g., U is the utility of a state
- The expected utility of $A = a$ is

$$EU[A = a] = \sum_{i=1}^{n} P(X = x_i | A = a) \cdot U(X = x_i)$$

One State/One Action Example



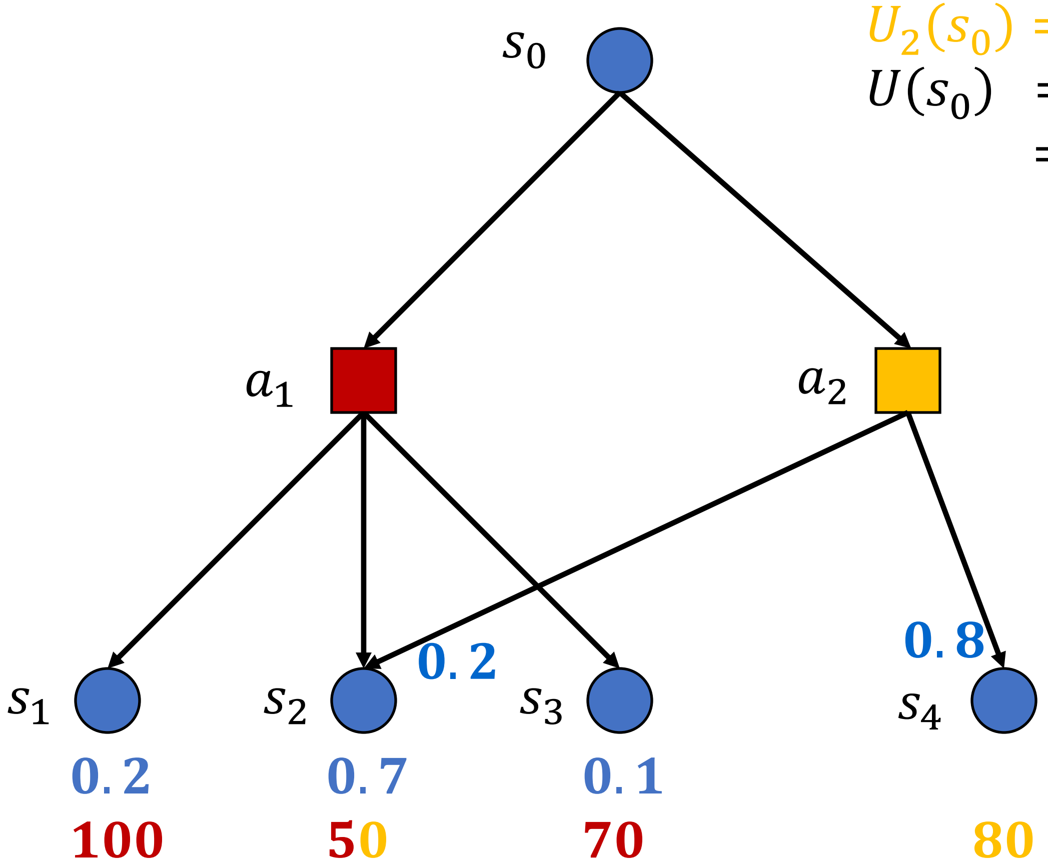
$$
\begin{aligned}
U(s_0) &= 100 \cdot 0.2 + 50 \cdot 0.7 + 70 \cdot 0.1 \\
&= 20 + 35 + 7 \\
&= 62
\end{aligned}
$$

One State/Two Actions Example

$U_1(s_0) = 62$

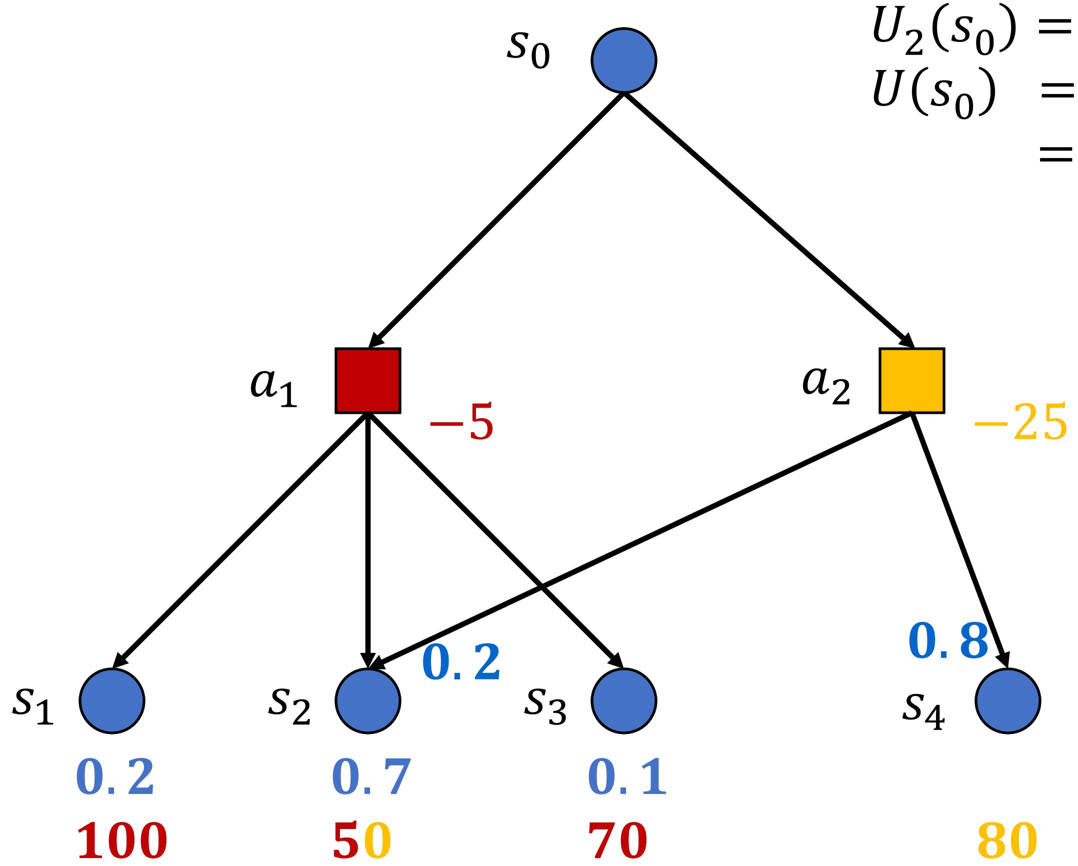
$U_2(s_0) = 74$

$$\begin{aligned} U(s_0) &= \max\{U_1(s_0), U_2(s_0)\} \\ &= 74 \end{aligned}$$



Introducing Action Costs

$U_1(s_0) = 62 - 5$
$U_2(s_0) = 74 - 25$
$U(s_0) \quad = \max\{U_1(s_0), U_2(s_0)\}$
$\quad = 57$



MEU Principle

- A rational agent should choose the action that maximizes agent's expected utility
- This is the basis of the field of decision theory
- The MEU principle provides a normative criterion for rational choice of action

AI is solved!!!

Not quite…

- Must have complete model of:
  - Actions
  - Utilities
  - States
- Even if you have a complete model, it might be computationally intractable
- In fact, a truly rational agent takes into account the utility of reasoning as well – bounded rationality
- Nevertheless, great progress has been made in this area, and we are able to solve much more complex decision-theoretic problems than ever before

Setting

- Agent can perform actions in an environment
  - Environment
    - Time: episodic or sequential
      - Episodic: Next episode does not depend on the previous episode
      - Sequential: Next episode depends on previous episodes
    - Non-deterministic
      - Outcomes of actions not unique
      - Associated with probabilities (→ probabilistic model)
    - Partially observable
      - Latent, i.e., not observable, random variables
  - Agent has preferences over states/action outcomes
    - Encoded in utility or utility function → Utility theory
- "Decision theory = Utility theory + Probability theory"
  - Model the world with a probabilistic model
  - Model preferences with a utility (function)
  - Find action that leads to the maximum expected utility, also called decision making

Outline

Utility Theory – mainly Ch. 16.1-16.4

- Preferences
- Utilities
- Dominance
- Preference structure

Markov Decision Process (MDP)

- Markov property
- Sequence of actions, history, policy
- Value iteration, policy iteration

Preferences

- An agent chooses among prizes (A, B, etc.) and lotteries, i.e., situations with uncertain prizes
  - Outcome of a nondeterministic action is a lottery
- Lottery $L = [p, A; (1-p), B]$
  - A and B can be lotteries again
  - Prizes are special lotteries: $[1, R; 0, \text{not } R]$
  - More than two outcomes:
    - $L = [p_1, S_1; p_2, S_2; \cdots; p_n, S_n], \sum_{i=1}^{n} p_i = 1$
- Notation
  - $A \succ B$ A preferred to B
  - $A \sim B$ indifference between A and B
  - $A \succsim B$ B not preferred to A

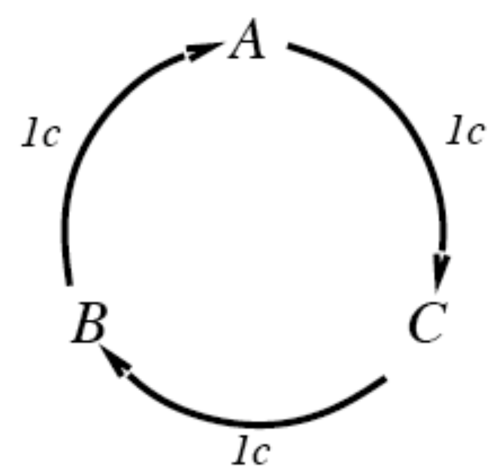
Rational preferences

- Idea: preferences of a rational agent must obey constraints
- Rational preferences $\Rightarrow$ behaviour describable as maximisation of expected utility

Rational preferences contd.

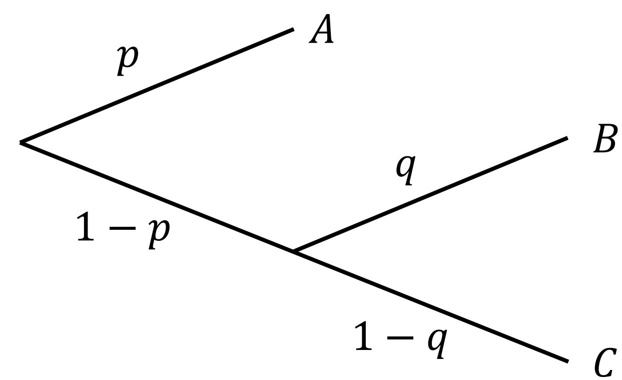
- Violating constraints leads to self-evident irrationality
- Example
  - An agent with intransitive preferences can be induced to give away all its money

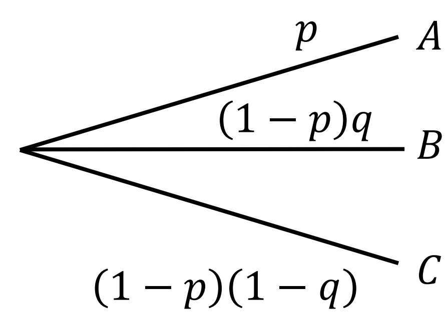
  - If $B \succ C$, then an agent who has C would pay (say) 1 cent to get B
  - If $A \succ B$, then an agent who has B would pay (say) 1 cent to get A
  - If $C \succ A$, then an agent who has A would pay (say) 1 cent to get C



Axioms of Utility Theory

1. Orderability
   - $(A \succ B) \vee (A \prec B) \vee (A \sim B)$
   - $\{\prec, \succ, \sim\}$ jointly exhaustive, pairwise disjoint
2. Transitivity
   - $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$
3. Continuity
   - $A \succ B \succ C \Rightarrow$ $\exists p\ [p, A;\ 1-p, C] \sim B$
4. Substitutability
   - $A \sim B \Rightarrow$ $[p, A; 1-p, C] \sim [p, B; 1-p, C]$
   - Also holds if replacing $\sim$ with $\succ$
5. Monotonicity
   - $A \succ B \Rightarrow$ $(p \geq q \Leftrightarrow$ $[p, A; 1-p, B]$ $\succsim\ [q, A; 1-q, B])$
6. Decomposability
   - $[p, A;\ 1-p, [q, B;\ 1-q, C]] \sim$ $[p, A;\ (1-p)q, B; (1-p)(1-q), C]$





Decomposability: There is no fun in gambling.

And Then There Was Utility

- Theorem (Ramsey, 1931; von Neumann and Morgenstern, 1944):
  - Given preferences satisfying the constraints, there exists a real-valued function U such that

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

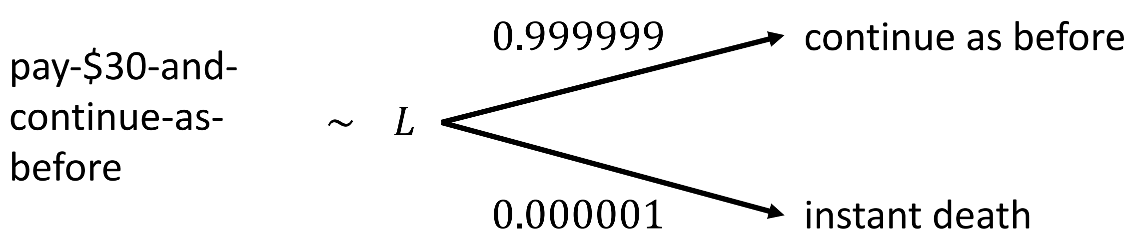
- MEU principle
  - Choose the action that maximises expected utility

- 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 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 outcome" $\top$ with probability p
    - "worst possible catastrophe" $\perp$ with probability $(1-p)$
  - Adjust lottery probability p until $A \sim L_p$



Utility Scales

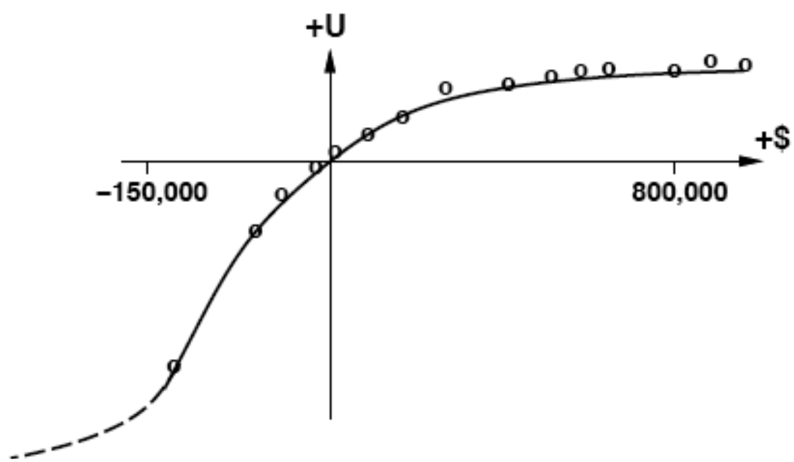
- Normalised utilities: $u_\top = 1.0, u_\bot = 0.0$
  - Utility of lottery $L \sim$ (pay-\$30-and-continue-as-before): $U(L) = u_\top \cdot 0.999999 + u_\bot \cdot 0.000001 = 0.999999$
- Micromorts: one-millionth chance of death
  - Useful for Russian roulette, paying to reduce product risks, etc.
- QALYs: quality-adjusted life years
  - Useful for medical decisions involving substantial risk
- Behaviour is invariant w.r.t. positive linear transformation
$$U'(r) = k_1 U(r) + k_2$$
  - No unique utility function; $U'(r)$ and $U(r)$ yield same behaviour

Ordinal Utility Functions

- With deterministic prizes only (no lottery choices), only ordinal utility can be determined, i.e., total order on prizes
  - Ordinal utility function also called value function
  - Provides a ranking of alternatives (states), but not a meaningful metric scale (numbers do not matter)

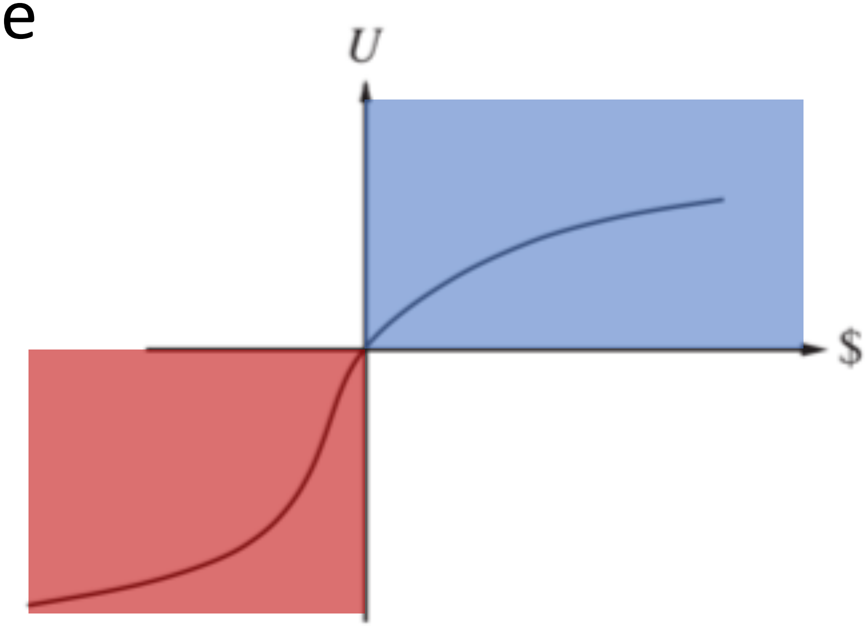
Money

- Money does not behave as a utility function
- Given a lottery L with expected monetary value $EMV(L)$, usually $U(L) < U\left(S_{EMV(L)}\right)$, i.e., people are risk-averse
  - S_n: state of possessing total wealth \$n
  - Utility curve
    - For what probability p am I indifferent between a prize x and a lottery $[p, \$M; (1-p), \$0]$ for large M?
    - Right: Typical empirical data, extrapolated with risk-prone behaviour for negative wealth



Money Versus Utility

- Money $\neq$ Utility
  - More money is better, but not always in a linear relationship to the amount of money
- Expected Monetary Value
  - Risk-averse
    - $U(L) < U\left(S_{EMV(L)}\right)$
  - Risk-seeking
    - $U(L) > U\left(S_{EMV(L)}\right)$
  - Risk-neutral
    - $U(L) = U\left(S_{EMV(L)}\right)$
    - Linear curve
    - For small changes in wealth relative to current wealth



Multi-attribute Utility Theory

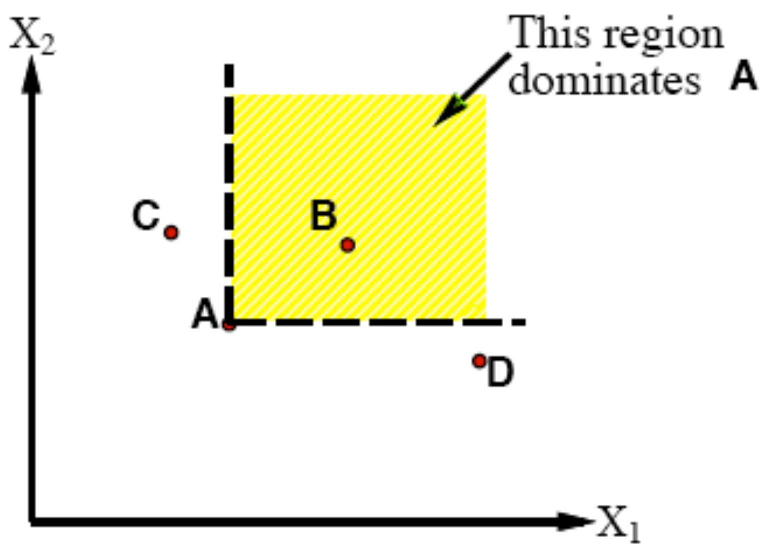
- A given state may have multiple utilities
  - ...because of multiple evaluation criteria
  - ...because of multiple agents (interested parties) with different utility functions

- We will look at
  - Cases in which decisions can be made *without* combining the attribute values into a single utility value
    - Strict dominance
    - Stochastic dominance
  - Cases in which the utilities of attribute combinations can be specified very concisely

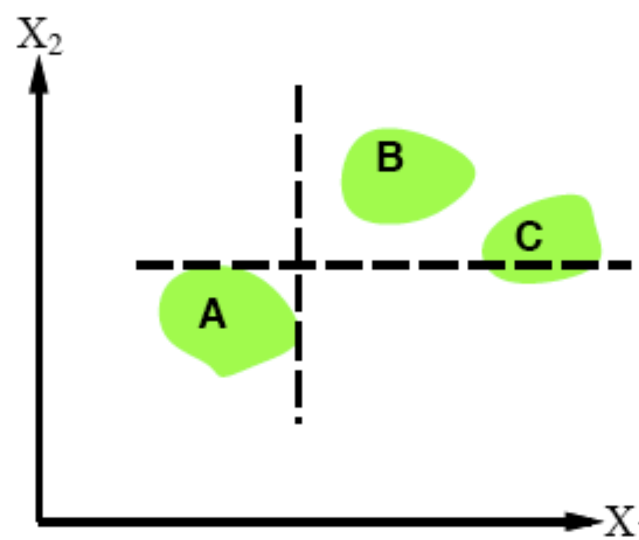
Strict Dominance

- Typically define attributes such that U is monotonic in each dimension
- Strict dominance
  - Choice B strictly dominates choice A iff

$$\forall\, i : X_i(B) \geq X_i(A) \text{ (and hence } U(B) \geq U(A))$$



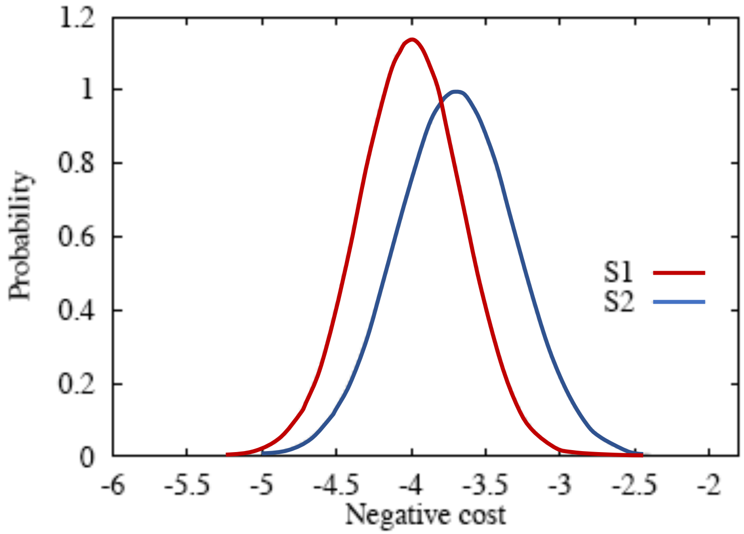
Deterministic attributes

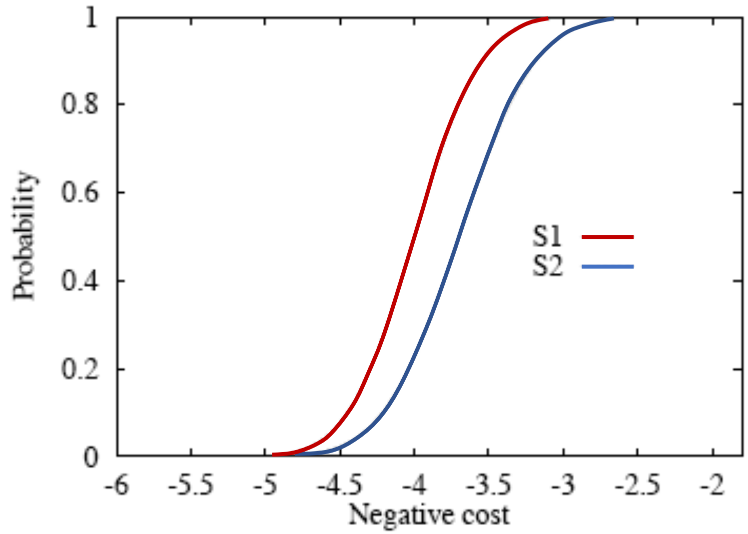


Uncertain attributes

Stochastic Dominance

- Cumulative distribution p_1 first-order stochastically dominates distribution p_2 iff
$$\forall x : p_2(x) \leq p_1(x)$$
  - With a strict inequality for some interval
  - Then, $E_{p_1} > E_{p_2}$ (E referring to expected value)
    - The reverse is not necessarily true
  - Does not imply that every possible return of the superior distribution is larger than every possible return of the inferior distribution
- Example:
  - As we have *negative costs*, S2 dominates S1 with $\forall x : p_{S_2}(x) \leq p_{S_1}(x)$





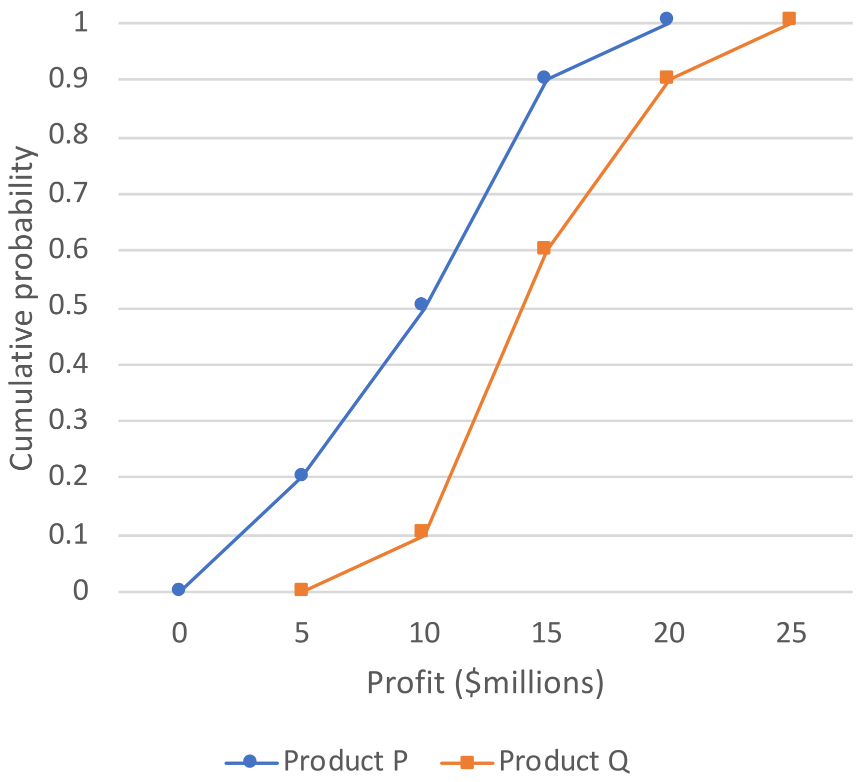
Example

- Product P

Profit ($m)	Probability
0 to under 5	0.2
5 to under 10	0.3
10 to under 15	0.4
15 to under 20	0.1

- Product Q

Profit ($m)	Probability
0 to under 5	0.0
5 to under 10	0.1
10 to under 15	0.5
15 to under 20	0.3
20 to under 25	0.1



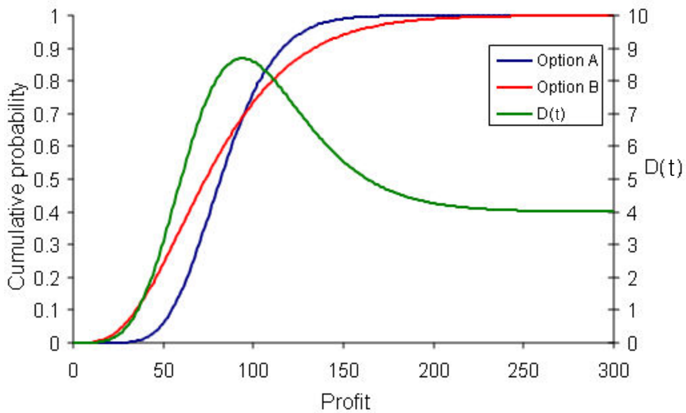
P first-order stochastically dominates Q

Stochastic Dominance

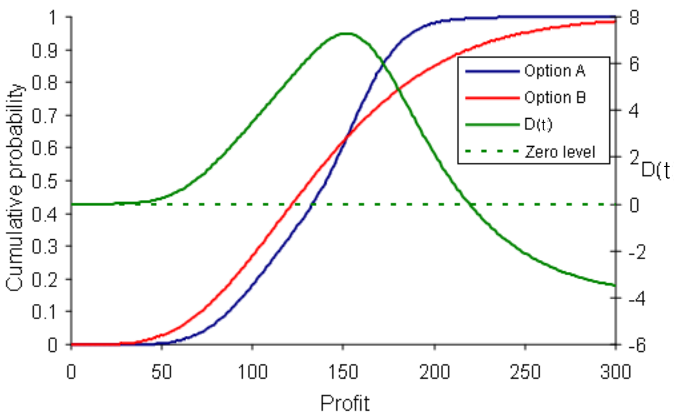
- Cumulative distribution p_1 second-order stochastically dominates distribution p_2 iff

$$\forall\, t : \int_{-\infty}^{t} p_2(x)\, dx \leq \int_{-\infty}^{t} p_1(x)\, dx$$

  - Or: $D(t) = \int_{-\infty}^{t} p_1(x) - p_2(x)\, dx \geq 0$
  - With a strict inequality for some interval
  - Then, $E_{p_1} \geq E_{p_2}$ (E referring to expected value)
- Example:
  - Second-order stochastic dominance



  - No dominance



https://people.duke.edu/~dgraham/ECO_463/Handouts/StochasticDominance.pdf
Figures: https://www.vosesoftware.com/riskwiki/Stochasticdominancetests.php (t=z)

Preference Structure

- To specify the complete utility function $U(r_1, \dots, r_n)$, we need d^n values in the worst case
  - n attributes
  - each attribute with d distinct possible values
  - Worst case meaning: Agent's preferences have no regularity at all
- Supposition in multi-attribute utility theory
  - Preferences of typical agents have much more structure
- Approach
  - Identify regularities in the preference behaviour
  - Use so-called representation theorems to show that an agent with a certain kind of preference structure has a utility function

$$U(r_1, \dots, r_n) = F[f_1(r_1), \dots, f_n(r_n)]$$

    - where F is hopefully a simple function such as addition

Preference structure: Deterministic

- R_1 and R_2 preferentially independent (PI) of R_3 iff
  - Preference between $\langle r_1, r_2, r_3 \rangle$ and $\langle r_1', r_2', r_3 \rangle$ does not depend on r_3
  - E.g., $\langle Noise, Cost, Safety \rangle$
    - $\langle 20{,}000\ suffer, \$4.6\ billion, 0.06\ deaths/month \rangle$
    - $\langle 70{,}000\ suffer, \$4.2\ billion, 0.06\ deaths/month \rangle$
- Theorem (Leontief, 1947)
  - If every pair of attributes is PI of its complement, then every subset of attributes is PI of its complement
    - Called mutual PI (MPI)
- Theorem (Debreu, 1960):
  - MPI ⇒ ∃ *additive* value function

$$V(r_1, \ldots, r_n) = \sum_i V_i(r_i)$$

  - Hence assess n single-attribute functions
  - Often a good approximation

Preference structure: Stochastic

- Need to consider preferences over lotteries
- R is utility-independent (UI) of S iff
  - Preferences over lotteries in R do not depend on s
- Mutual UI (Keeney, 1974): each subset is UI of its complement $\Rightarrow \exists$ *multiplicative* utility function
  - For $n = 3$:

$$U = k_1U_1 + k_2U_2 + k_3U_3 \\ +k_1k_2U_1U_2 + k_2k_3U_2U_3 + k_3k_1U_3U_1 \\ +k_1k_2k_3U_1U_2U_3$$

  - I.e., requires only n single-attribute utility functions and n constants

Intermediate Summary

- Preferences
  - Preferences of a rational agent must obey constraints
- Utilities
  - Rational preferences = describable as maximisation of expected utility
  - Utility axioms
  - MEU principle
- Dominance
  - Strict dominance
  - First-order + second-order stochastic dominance
- Preference structure
  - (Mutual) preferential independence
  - (Mutual) utility independence

Outline

Utility Theory

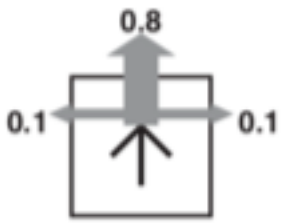
- Preferences
- Utilities
- Dominance
- Preference structure

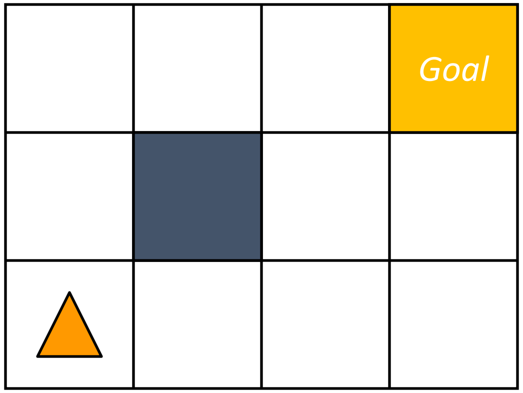
Markov Decision Process (MDP) – Ch. 17.1-17.3

- Markov property
- Sequence of actions, history, policy
- Value iteration, policy iteration

Simple Robot Navigation Problem

- In each state, the possible actions are U, D, R, and L
- The effect of U is as follows (transition model):
  - With probability 0.8, move up one square
    - If already in top row or blocked, no move
  - With probability 0.1, move right one square
    - If already in rightmost row or blocked, no move
  - With probability 0.1, move left one square
    - If already in leftmost row or blocked, no move
- Same transition model holds for D, R, and L and their respective directions





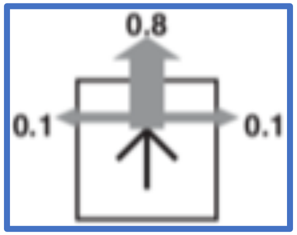
Markov Property

The transition properties depend only on the current state, not on previous history (how that state was reached).

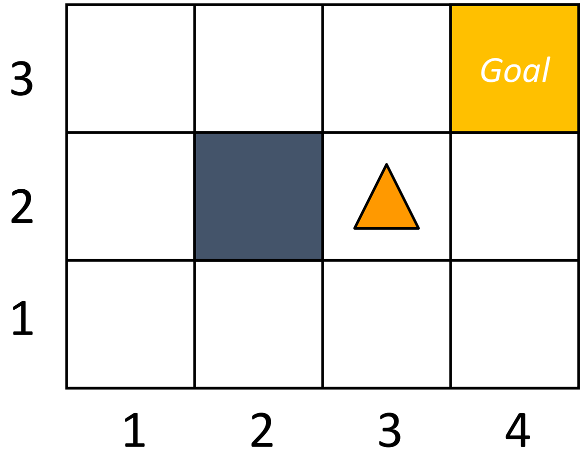
- Also known as Markov-k with $k = 1$
  - $k \leq t$
    $$P(x_{t+1} \mid x_t, \dots, x_0) = P(x_{t+1} \mid x_t, \dots, x_{t-k+1})$$
  - $k = 1$
    $$P(x_{t+1} \mid x_t, \dots, x_0) = P(x_{t+1} \mid x_t)$$

Sequence of Actions

- In each state, the possible actions are U, D, R, and L; transition model:



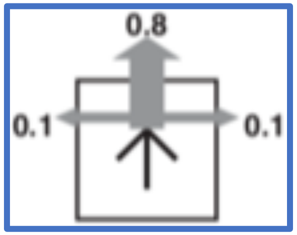
- Current position: [3,2]
- Planned sequence of actions: (U, R)

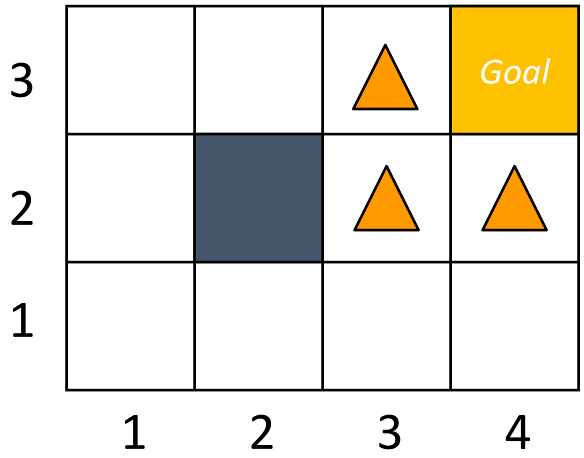


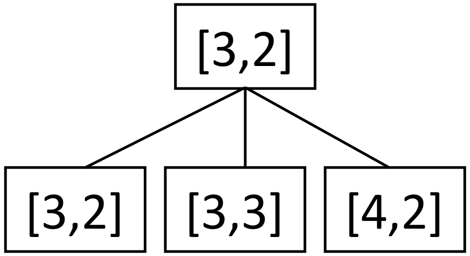
[3,2]

Sequence of Actions

- In each state, the possible actions are U, D, R, and L; transition model:
- Current position: [3,2]
- Planned sequence of actions: (U, R)
  - U is executed

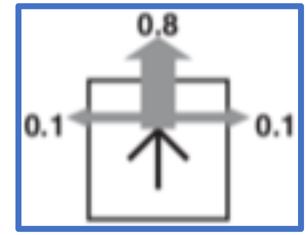




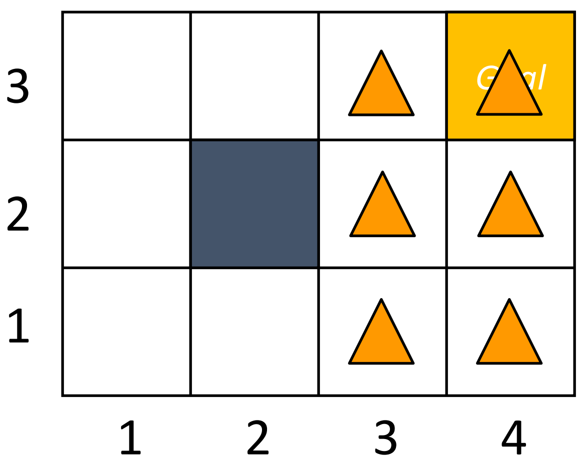


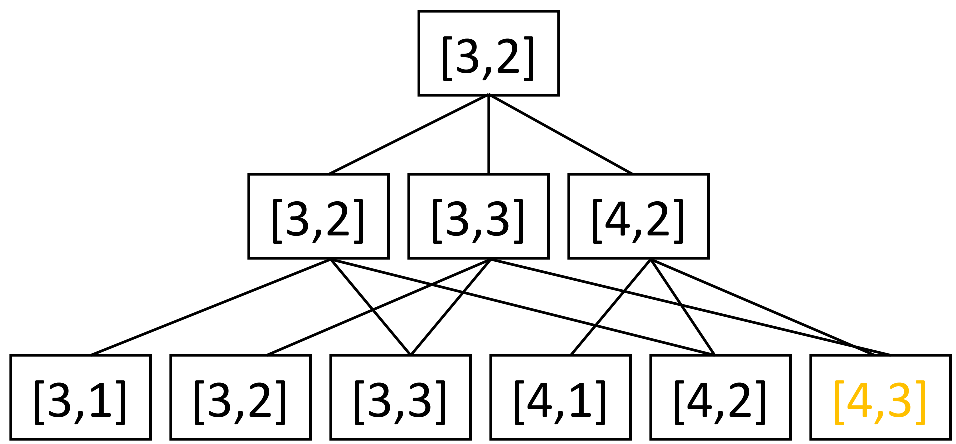
Sequence of Actions

- In each state, the possible actions are U, D, R, and L; transition model:



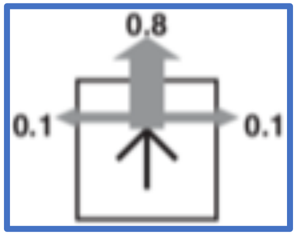
- Current position: [3,2]
- Planned sequence of actions: (U, R)
  - U has been executed
  - R is executed





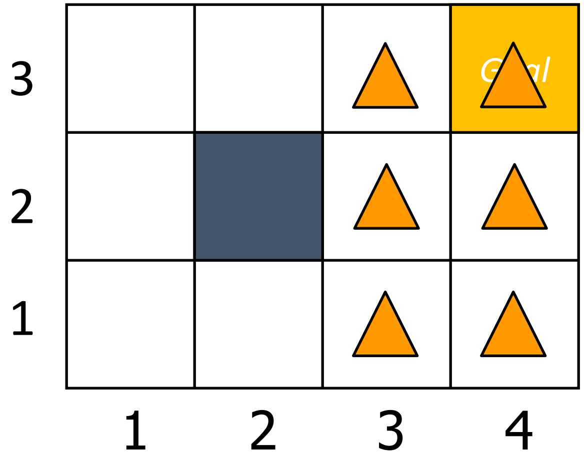
Histories

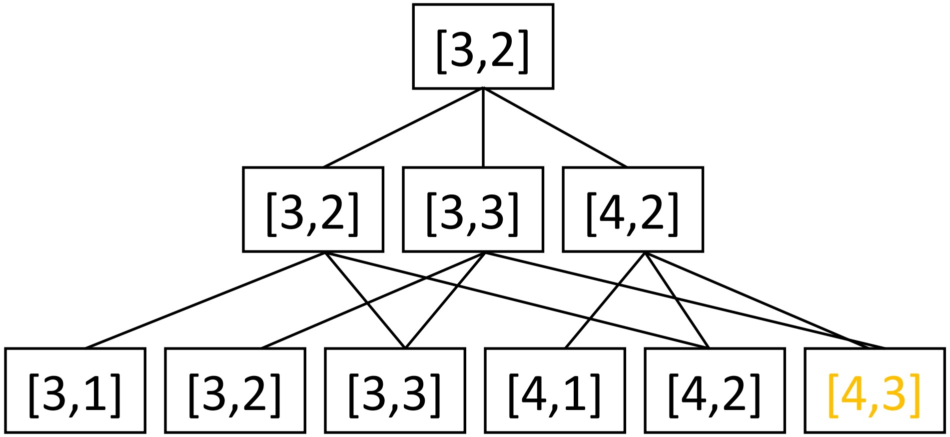
- In each state, the possible actions are U, D, R, and L; transition model:



- Current position: [3,2]
- Planned sequence of actions: (U, R)
  - U has been executed
  - R is executed

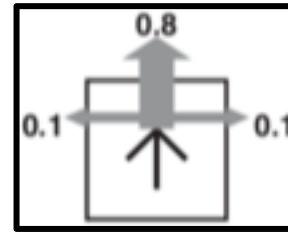
9 possible sequences of states, called histories, and 6 possible final states





Probability of Reaching the Goal

- In each state, the possible actions are U, D, R, and L; transition model:

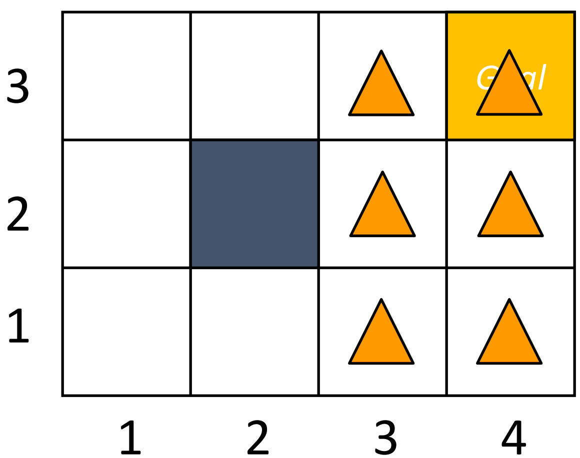


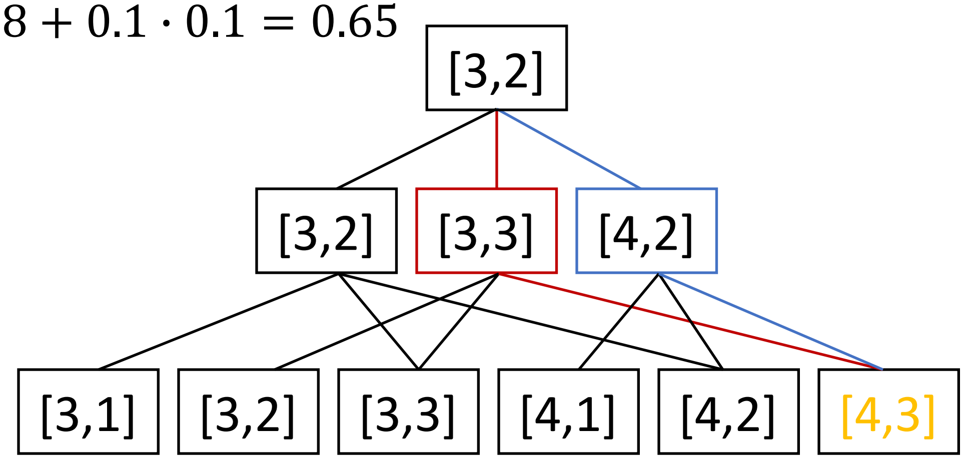
$$P([4,3] \mid (U,R).[3,2]) =$$
$$P([4,3] \mid R.[3,3]) \cdot P([3,3] \mid U.[3,2]) + P([4,3] \mid R.[4,2]) \cdot P([4,2] \mid U.[3,2])$$

$P([4,3] \mid R.[3,3]) = 0.8$ $\quad$ $P([3,3] \mid U.[3,2]) = 0.8$
$P([4,3] \mid R.[4,2]) = 0.1$ $\quad$ $P([4,2] \mid U.[3,2]) = 0.1$

Note importance of Markov property in this derivation

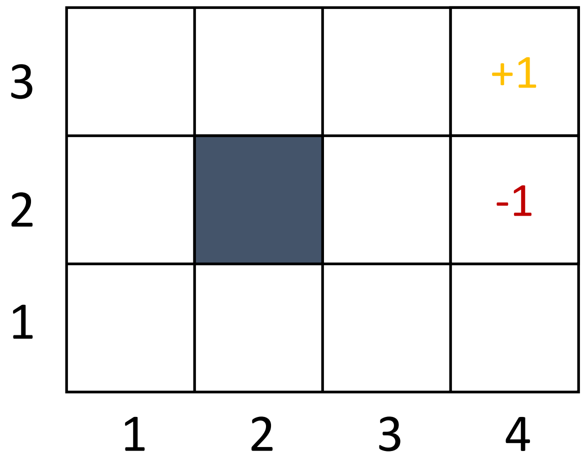
$$P([4,3] \mid (U,R).[3,2]) = 0.8 \cdot 0.8 + 0.1 \cdot 0.1 = 0.65$$





Utility Function

- [4,3] : power supply
- [4,2] : sand area the robot cannot escape
- Goal: robot needs to recharge its batteries
- [4,3] and [4,2] are terminal states
- In this example, we define the utility of a history by
  - the utility of the last state (+1 or –1) minus $0.04 \cdot n$
    - n is the number of moves
    - I.e., each move costs 0.04, which provides an incentive to reach the goal fast



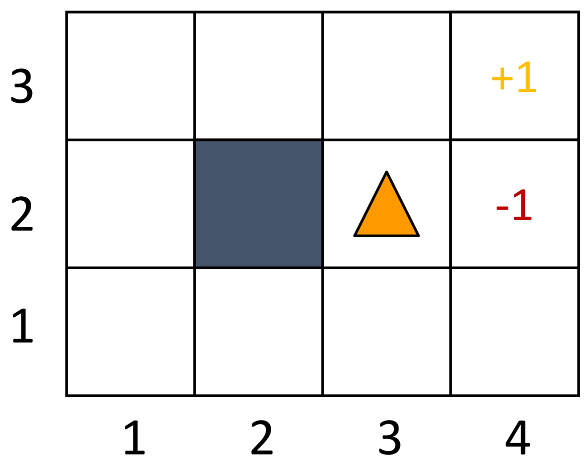
Utility of an Action Sequence

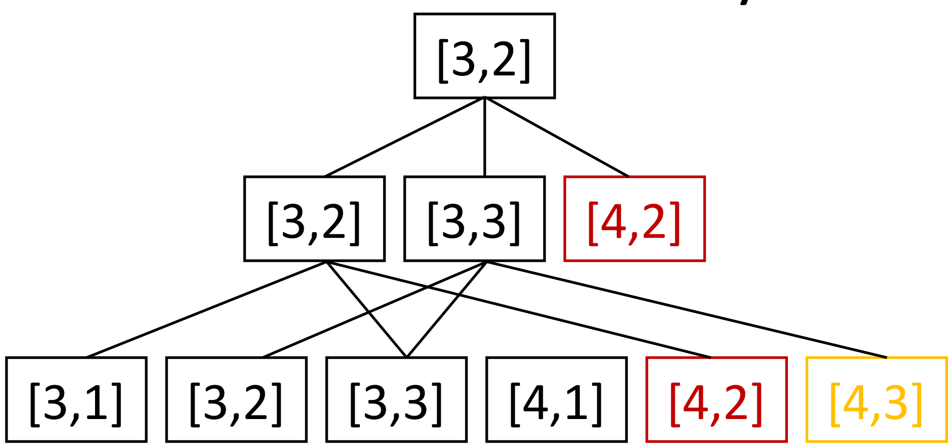
- Consider the action sequence (U,R) from [3,2]
- A run produces one among 7 possible histories, each with some probability
- Utility of the sequence is the expected utility of histories h:

$$U = \sum_h U_h P(h)$$

- Optimal sequence = the one with maximum utility

Is the optimal sequence what we want?

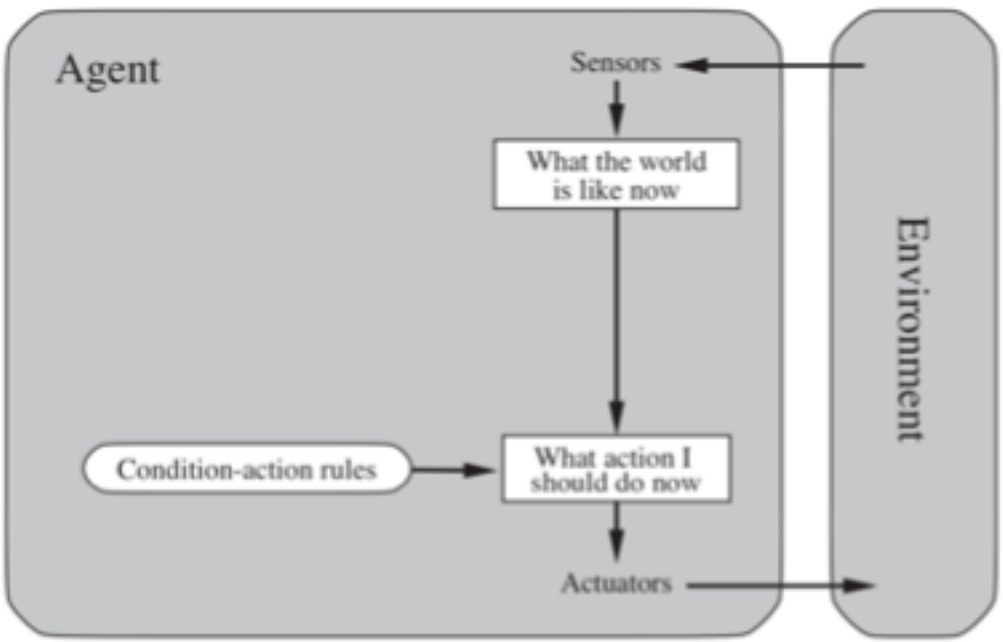




Reactive Agent Algorithm

Accessible or observable state

```
Act()
   repeat
      s ← sensed state
      if s is terminal then
         exit
      a ← choose action (given s)
      perform a
```



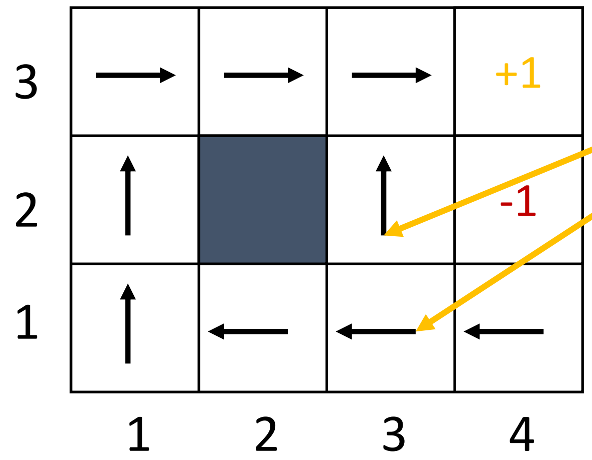
AIMA, Russell/Norvig

Policy (Reactive/Closed-loop Strategy)

- Policy π
  - Complete mapping from states to actions
- Optimal policy π^*
  - Always yields a history (ending at terminal state) with maximum expected utility
    - Due to Markov property

```
Act()
  repeat
    s ← sensed state
    if s is terminal then
      exit
    a ← π(s)
    perform a
```

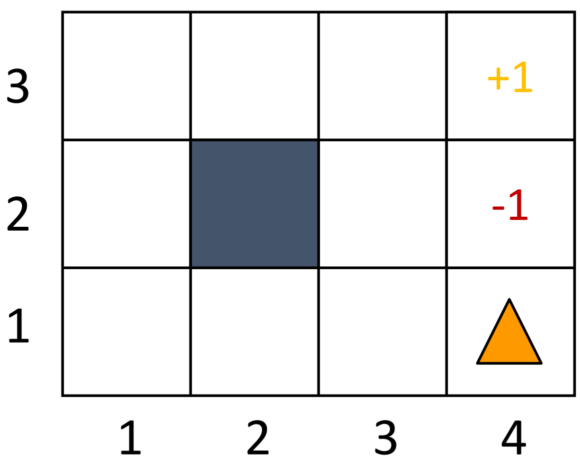
Note that [3,2] is a "dangerous" state that the optimal policy tries to avoid



How to compute π^*?
Solving a Markov Decision Process (MDP)

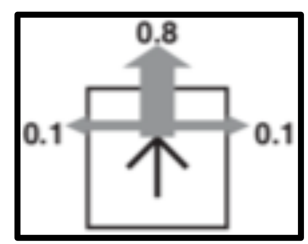
MDP

- *Sequential* decision problem for a fully observable, stochastic environment with a Markovian transition model and additive rewards (next slide)
- Components
  - a set of states S (with an initial state s_0)
  - a set $A(s)$ of actions in each state
  - a transition model $P(s'|s,a)$
  - a reward function $R(s)$



U, D, L, R

each move costs 0.04

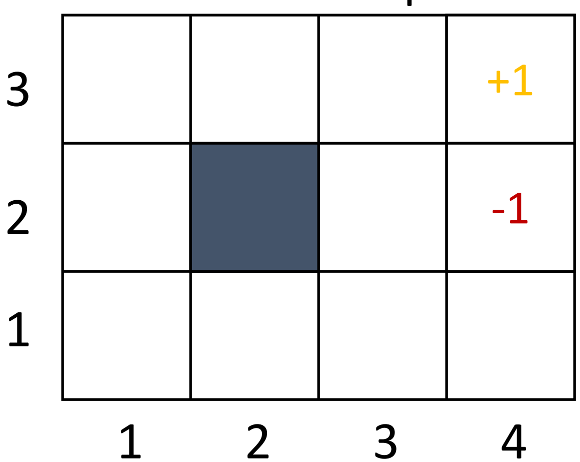


Additive Utility

- History $H = (s_0, s_1, \dots, s_n)$
- In each state s, agent receives reward $R(s)$
- Utility of H is additive iff

$$U(s_0, s_1, \dots, s_n) = R(s_0) + U(s_1, \dots, s_n) = \sum_{i=0}^{n} R(s_i)$$

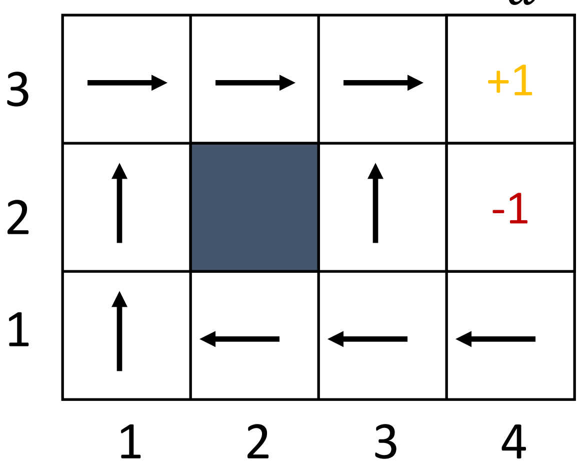
  - Discount factor $\gamma \in]0,1]$: $U(s_0, s_1, \dots, s_n) = \sum_{i=0}^{n} \gamma^i R(s_i)$
    - Close to 0: future rewards insignificant
    - Corresponds to an interest rate of ${}^{1-\gamma}/_{\gamma}$



- Robot navigation example:
  - $R(s_n) = +1$ if $s_n = [4,3]$
  - $R(s_n) = -1$ if $s_n = [4,2]$
  - $R(s_i) = -0.04$ if $i = 0, \dots, n-1$
  - $\gamma = 1$

Principle of MEU

- History $h = (s_0, s_1, \dots, s_n)$
  - Utility of h: $U(s_0, s_1, \dots, s_n) = \sum_{i=0}^{n} R(s_i)$
- Bellman equation:
  - $U(s_i) = R(s_i) + \gamma \max_{a} \sum_{s_j} P\big(s_j | a. s_i\big) U\big(s_j\big)$
- Optimal policy:
  - $\pi^*(s_i) = \operatorname{argmax}_{a} \sum_{s_j} P\big(s_j | a. s_i\big) U\big(s_j\big)$



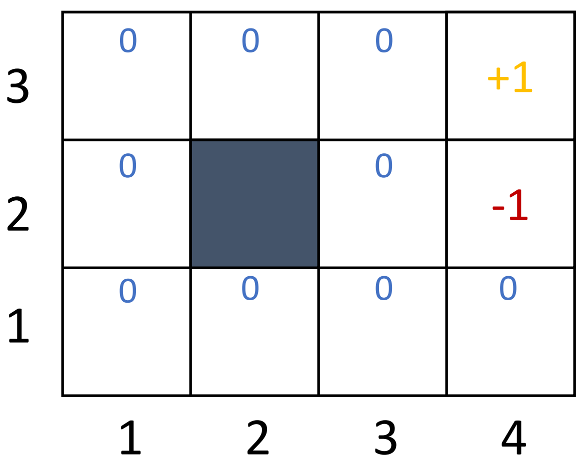
- Bellman equation for [1,1]
  - $U(1,1) = -0.04 + \gamma \max_{U,L,D,R}$

$$
\begin{array}{ll}
\{\, 0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1), & \text{(U)} \\
\;\; 0.8U(1,1) + 0.1U(1,1) + 0.1U(1,2), & \text{(L)} \\
\;\; 0.8U(1,1) + 0.1U(2,1) + 0.1U(1,1), & \text{(D)} \\
\;\; 0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1) \,\} & \text{(R)}
\end{array}
$$

  - with $\gamma = 1$ as discount factor

Value Iteration

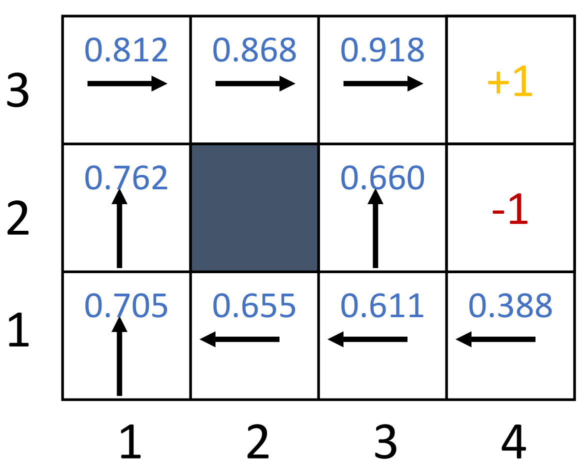
- Initialise the utility of each non-terminal state s_i to $U_0(s_i) = 0$
- For $t = 0, 1, 2, \ldots,$ do
  - $U_{t+1}(s_i) \leftarrow R(s_i) + \gamma \max_a \sum_{s_j} P(s_j | a.s_i) U_t(s_j)$
    - So called Bellman update

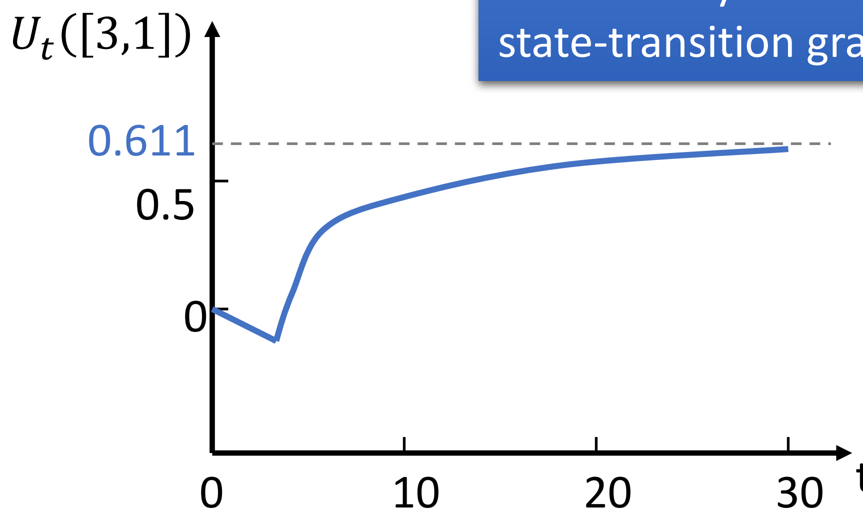


Value Iteration

- Initialise the utility of each non-terminal state s_i to $U_0(s_i) = 0$
- For $t = 0, 1, 2, \ldots,$ do
  - $U_{t+1}(s_i) = R(s_i) + \gamma \max_a \sum_{s_j} P(s_j | a. s_i) U_t(s_j)$
    - So called Bellman update

Note the importance of terminal states and connectivity of the state-transition graph





Value Iteration: Algorithm

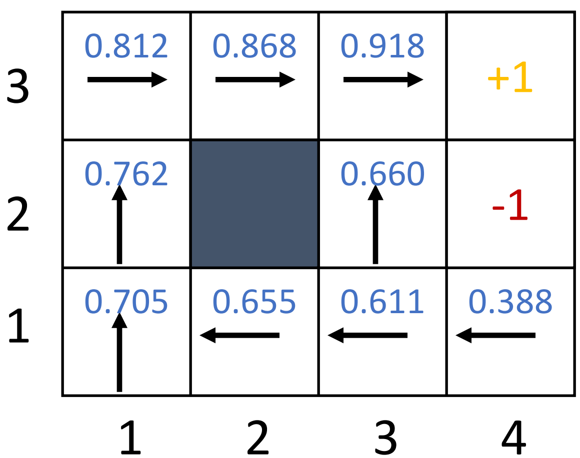
```
function value-iteration(mdp,ε)
    U′ ← 0
    repeat
        U ← U′
        δ ← 0
        for each state s ∈ S do
            U′[s] ← R(s) + γ max_{a∈A(s)} Σ_{s′} P(s′|a.s) U[s′]
            if |U′[s] - U[s]| > δ then
                δ ← |U′[s] - U[s]|
    until δ < ε(1-γ)/γ
```

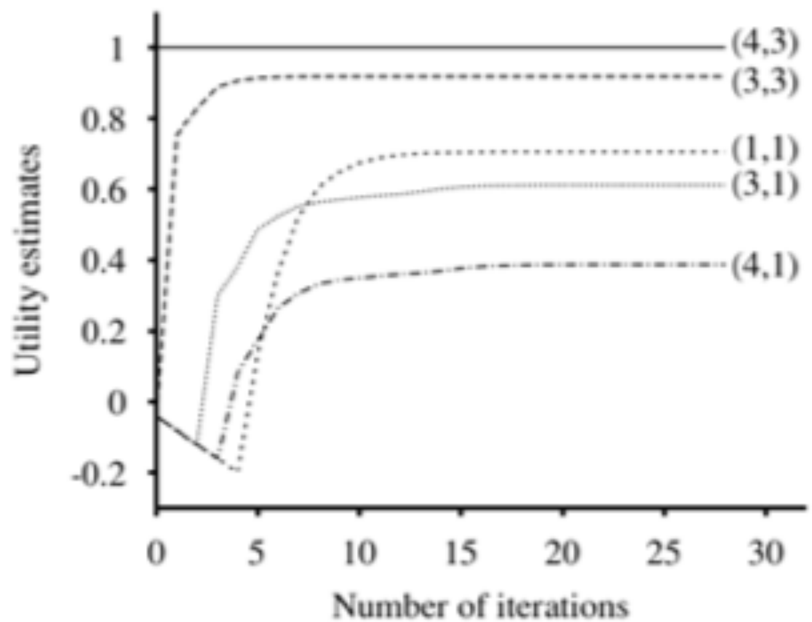
- Inputs
  - an MDP, which includes
    - States S
    - For all $s \in S$, actions $A(s)$, transition model $P(s' | a.s)$, rewards $R(s)$
    - Discount γ
  - Maximum error allowed ϵ
- Local variables
  - U, U' vectors of utilities for states in S, initially 0
  - δ maximum change in utility of any state in an iteration

Evolution of Utilities

- For $t = 0, 1, 2, \ldots$, do
  - $U_{t+1}(s_i) = R(s_i) + \gamma \max_a \sum_{s_j} P(s_j | \, a. s_i) U_t(s_j)$

- Value iteration ≈ information propagation

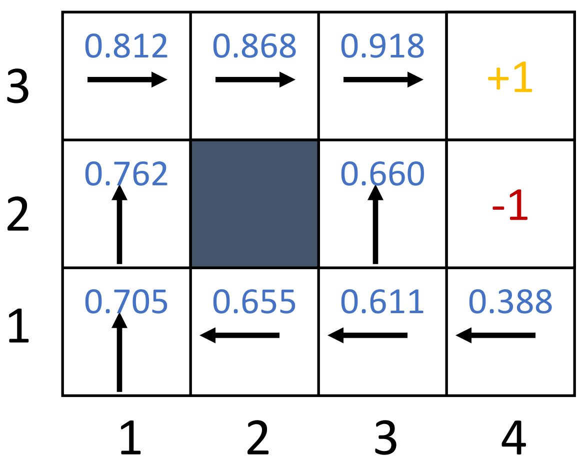




Argmax Action

- For $t = 0, 1, 2, \ldots$, do
  - $U_{t+1}(s_i) = R(s_i) + \gamma \max_a \sum_{s_j} P(s_j | \, a. s_i) U_t(s_j)$

- Argmax action may change over iterations



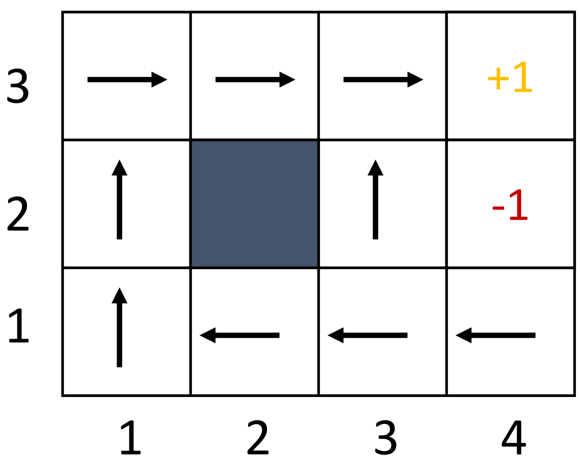
- Bellman equation for [1,1]
  - $U(1{,}1) = -0.04 + \gamma \max_{U,L,D,R}$

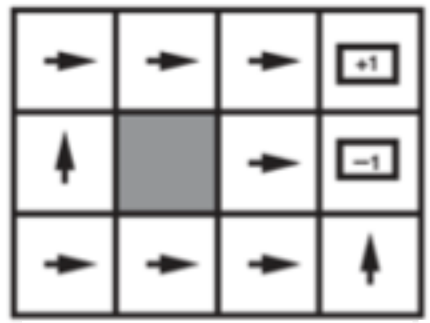
$$\begin{Bmatrix} \boxed{0.8U(1{,}2) + 0.1U(2{,}1) + 0.1U(1{,}1), & \text{(U)}} \\ 0.8U(1{,}1) + 0.1U(1{,}1) + 0.1U(1{,}2), & \text{(L)} \\ 0.8U(1{,}1) + 0.1U(2{,}1) + 0.1U(1{,}1), & \text{(D)} \\ 0.8U(2{,}1) + 0.1U(1{,}2) + 0.1U(1{,}1) & \text{(R)} \end{Bmatrix}$$

  - with $\gamma = 1$ as discount factor

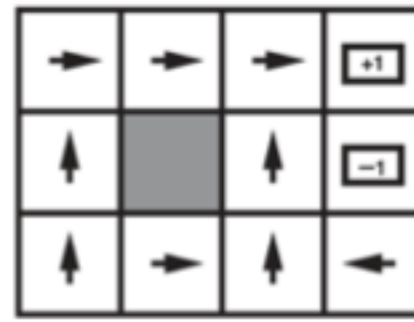
Effect of Rewards

- For $t = 0, 1, 2, \ldots,$ do
  - $U_{t+1}(s_i) = R(s_i) + \gamma \max_a \sum_{s_j} P(s_j | \, a. s_i) U_t(s_j)$
- Optimal policies for different rewards
  - For $R(s) = -0.04$, see below (⇩)

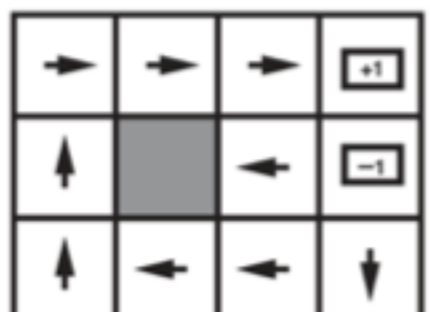




$R(s) < -1.6284$



$-0.4278 < R(s) < -0.0850$

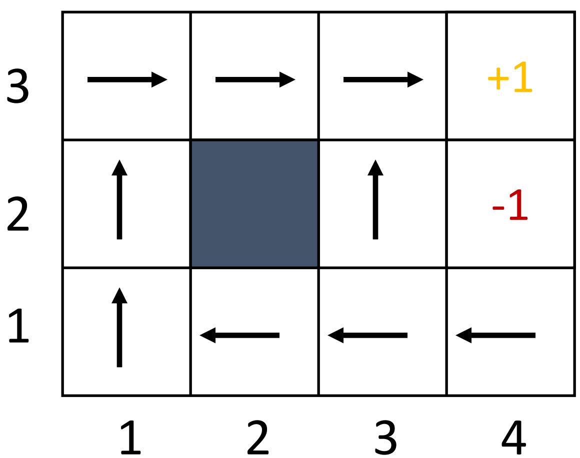


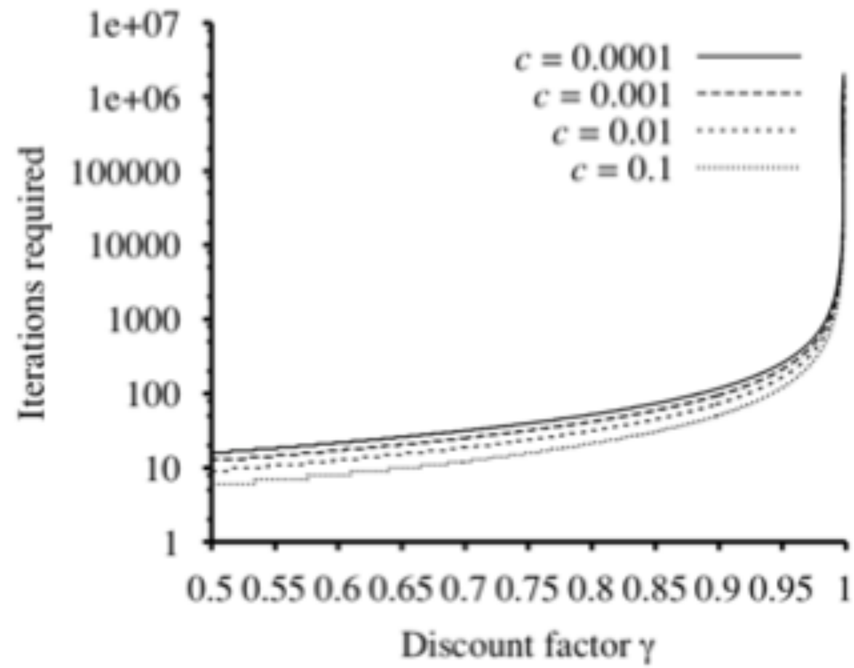
$-0.0221 < R(s) < 0$

$R(s) > 0$

Effect of Allowed Error & Discount

- For $t = 0, 1, 2, \ldots,$ do
  - $U_{t+1}(s_i) = R(s_i) + \gamma \max_a \sum_{s_j} P(s_j | \, a. s_i) U_t(s_j)$
- Right figure: Iterations required to ensure a maximum error of $\varepsilon = c \cdot R_{max}$
  - R_{max} maximum reward
    - +1 in the example





Policy Iteration

- Pick a policy π_0 at random
- Repeat:
  - Policy evaluation: Compute the utility of each state for π_t
    - $\boxed{U_t(s_i) = R(s_i) + \gamma \sum_{s_j} P\big(s_j | \pi_t(s_i).\, s_i\big) U_t\big(s_j\big)}$
      - No longer involves a max operation as action is determined by π_t
  - Policy improvement: Compute the policy π_{t+1} given U_t
    - $\pi_{t+1}\,(s_i) = \arg\max_a \sum_{s_j} P\big(s_j | \pi_t(s_i).\, s_i\big) U_t\big(s_j\big)$
  - If $\pi_{t+1} = \pi_t$, then return π_t

Solve the set of linear equations:

$$U(s_i) = R(s_i) + \gamma \sum_{s_j} P\big(s_j | \pi(s_i).\, s_i\big) U\big(s_j\big)$$

(often a sparse system)

Policy Iteration: Algorithm

```
function policy-iteration(mdp)
    repeat
        U ← policy-evaluation(π,U,mdp)
        unchanged ← true
        for each state s ∈ S do
            if max_{a∈A(s)} Σ_{s'} P(s'|a.s) U[s'] > Σ_{s'} P(s'|π[s].s) U[s'] then
                π[s] ← argmax_{a∈A(s)} Σ_{s'} P(s'|a.s) U[s']
                unchanged ← false
    until unchanged
    return π
```

- Inputs
  - an MDP, which includes
    - States S
    - For all $s \in S$, actions $A(s)$, transition model $P(s' \mid a.s)$, rewards $R(s)$
- Local variables
  - U vectors of utilities for states in S, initially 0
  - π a policy vector indexed by state, initially random

Policy Evaluation

- Compute the utility of each state for π
  - $U_t(s_i) = R(s_i) + \gamma \sum_{s_j} P\big(s_j | \pi_t(s_i). s_i\big) U_t\big(s_j\big)$
- Complexity of policy evaluation: $O(n^3)$
  - For n states, n linear equations with n unknowns
  - Prohibitive for large n
- Approximation of utilities
  - Perform k value iteration steps with fixed policy π_t, return utilities
    - Simplified Bellman update: $U_{t+1}(s_i) = R(s_i) + \gamma \sum_{s_j} P\big(s_j | \pi(s_i). s_i\big) U_t\big(s_j\big)$
  - Asynchronous policy iteration (next slide)
    - Pick any subset of states

Asynchronous Policy Iteration

- Further approximation of policy iteration
  - Pick any subset of states and do one of the following
    - Update utilities
      - Using simplified value iteration as described on previous slide
    - Update the policy
      - Policy improvement as before
- Is not guaranteed to converge to an optimal policy
  - Possible if each state is still visited infinitely often, knowledge about unimportant states, etc.
- Freedom to work on any states allows for design of domain-specific heuristics
  - Update states that are likely to be reached by a good policy

Intermediate Summary

- MDP
  - Markov property
    - Current state depends only on previous state
  - Sequence of actions, history, policy
    - Sequence of actions may yield multiple histories, i.e., sequences of states, with a utility
    - Policy: complete mapping of states to actions
    - Optimal policy: policy with maximum expected utility
  - Value iteration, policy iteration
    - Algorithms for calculating an optimal policy for an MDP

Online Decision Making

- Decision making based on probabilistic graphical models (PGMs)
  - Do not precompute a policy beforehand but decide on an action (sequence) online given current observations
- Static case (episodic, without effects on next state)
  - PGMs extended with action and utility nodes
  - MEU query: Calculate expected utility for each action, decide to execute action with highest expected utility
- Dynamic case (temporal, with effects on next state)
  - Dynamic PGMs extended with action and utility nodes
  - MEU query: Calculate expected utility for sequence of actions, decide to execute action sequence with highest expected utility
- More in module **Intelligent Agents** (IFIS, winter term)

Outline

Utility Theory

- Preferences
- Utilities
- Dominance
- Preference structure

Markov Decision Process (MDP)

- Markov property
- Sequence of actions, history, policy
- Value iteration, policy iteration

$\Longrightarrow$ Next: Probabilistic Models