16 MAKING SIMPLE DECISIONS

- Let us associate each state S with a numeric *utility* $U(S)$, which expresses the desirability of the state
- A nondeterministic action A will have possible outcome states $Result_i(A)$, where the index i ranges over the different outcomes
- Prior to the execution of A the agent assigns probability $P(Result_i(A) \mid Do(A), E)$ to each outcome, where E summarizes the agent's available evidence of the world
- The expected utility of A can now be calculated:

$$EU(A \mid E) = \sum_i P(Result_i(A) \mid Do(A), E) \cdot U(Result_i(A))$$

- The principle of *Maximum expected utility* (MEU) says that a rational agent should choose an action that maximizes the agent's expected utility
- If we wanted to choose the best sequence of actions using this equation, we would have to enumerate all action sequences, which is clearly infeasible for long sequences
- If the utility function correctly reflects the performance measure by which the behavior is being judged, then using MEU the agent will achieve the highest possible performance score averaged over the environments in which it could be placed
- Let us model a nondeterministic action with a *lottery* L, where possible outcomes $C_1, \ldots, C_n$ can occur with probabilities $p_1, \ldots, p_n$

$$L = [p_1, C_1; p_2, C_2; \ldots; p_n, C_n]$$

16.2 The Basis of Utility Theory

$A \succ B$ Agent prefers lottery A over B
$A \sim B$ The agent is indifferent between A and B
$A \succeq B$ The agent prefers A to B or is indifferent between them

- Deterministic lottery $[1, A] \equiv A$
- Reasonable constraints on the preference relation (in the name of rationality)
  - **Orderability**: given any two states, a rational agent must either prefer one to the other or else rate the two as equally preferable.

    $$(A \succ B) \vee (B \succ A) \vee (A \sim B)$$
  - **Transitivity:**

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

- **Continuity**:

  $$A \succ B \succ C \Rightarrow \exists\, p: [p, A; 1\text{-}p, C] \sim B$$
- **Substitutability**:

  $$A \sim B \Rightarrow [p, A; 1\text{-}p, C] \sim [p, B; 1\text{-}p, C]$$
- **Monotonicity**:

  $$A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1\text{-}p, B] \succeq [q, A; 1\text{-}q, B])$$
- **Decomposability**: Compound lotteries can be reduced to simpler ones by the laws of probability

  $[p, A; 1\text{-}p, [q, B; 1\text{-}q, C]] \sim [p, A; (1\text{-}p)q, B; (1\text{-}p)(1\text{-}q), C]$

- Notice that the axioms of utility theory do not say anything about utility
- The existence of a utility function follows from them

And then there was Utility

1. **Utility principle**:
   If an agent's preferences follow the axioms of utility, then there exists a real-valued function U s.t.

$$U(A) > U(B) \Leftrightarrow A \succ B$$
$$U(A) = U(B) \Leftrightarrow A \sim B$$

2. **Maximum Expected Utility principle**:
   The utility of a lottery is

$$U([p_1, S_1; \ldots; p_n, S_n]) = \Sigma_{i=1,\ldots,n} p_i \cdot U(S_i)$$

   Because the outcome of a nondeterministic action is a lottery, this gives us the MEU decision rule from slide 247

16.3 Utility Functions

- Money (or an agent's total net assets) would appear to be a straightforward utility measure
- The agent exhibits a monotonic preference for definite amounts of money
- We need to determine a model for lotteries involving money
  - We have won a million euros in a TV game show
  - The host offers to flip a coin, if the coin comes up heads, we end up with nothing, but if it comes up tails, we win three millions
  - Is the only rational choice to accept the offer which has the expected monetary value of 1,5 million euros?
- The true question is maximizing total wealth (not winnings)

- The axioms of utility do not specify a unique utility function for an agent
- For example, we can transform a utility function $U(S)$ into

$$U'(S) = k_1 + k_2 \cdot U(S)$$

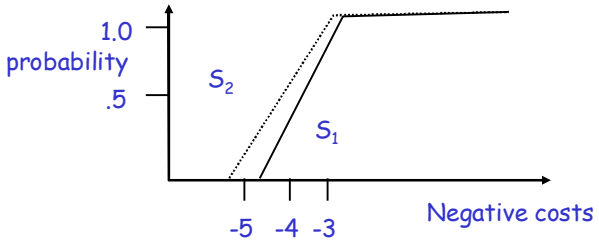
where k_1 is a constant and k_2 is any positive constant
- Clearly, this linear transformation leaves the agent's behavior unchanged
- In deterministic contexts, where there are states but no lotteries, behavior is unchanged by any monotonic transformation
- E.g., the cube root of the utility $\sqrt[3]{(U(S))}$
- Utility function is ordinal — it really provides just rankings of states rather than meaningful numerical values

- The scale of utilities reaches from the best possible prize $u_\top$ to the worst possible catastrophe $u_\bot$
- *Normalized utilities* use a scale with $u_\bot = 0$ and $u_\top = 1$
- Utilities of intermediate outcomes are assessed by asking the agent to indicate a preference between the given outcome state S and a standard lottery $[p, u_\top; 1\text{-}p, u_\bot]$
- The probability p is adjusted until the agent is indifferent between S and the standard lottery
- Assuming normalized utilities, the utility of S is given by p

16.4 Multiattribute Utility Functions

- Most often the utility is determined by the values $\mathbf{x} = [x_1, \ldots, x_n]$ of multiple variables (attributes) $\mathbf{X} = X_1, \ldots, X_n$
- For simplicity, we will assume that each attribute is defined in such a way that, all other things being equal, higher values of the attribute correspond to higher utilities
- If for a pair of attribute vectors $\mathbf{x}$ and $\mathbf{y}$ it holds that $x_i \geq y_i \ \forall\ i$, then $\mathbf{x}$ *strictly dominates* $\mathbf{y}$
- Suppose that airport site S_1 costs less, generates less noise pollution, and is safer than site S_2, one would not hesitate to reject the latter
- In the general case, where the action outcomes are uncertain, strict dominance occurs less often than in the deterministic case
- *Stochastic dominance* is more useful generalization

- Suppose we believe that the cost of siting an airport is uniformly distributed between
  - S_1: 2.8 and 4.8 billion euros
  - S_2: 3.0 and 5.2 billion euros
- Then by examining the cumulative distributions, we see that S_1 stochastically dominates S_2 (because costs are negative)



- Cumulative distribution integrates the original distribution
- If two actions A_1 and A_2 lead to probability distributions $p_1(x)$ and $p_2(x)$ on attribute X
- A_1 stochastically dominates A_2 on X if

$$\forall\, x: \int_{-\infty,\ldots,x} p_1(x')\, dx' \le \int_{-\infty,\ldots,x} p_2(x')\, dx'$$

- If
  - A_1 stochastically dominates A_2,
  - then for any monotonically nondecreasing utility function $U(x)$,

  the expected utility of A_1 is at least as high as that of A_2
- Hence, if an action is stochastically dominated by another action on all attributes, then it can be discarded

16.6 The Value of Information

- BP is hoping to buy one of n indistinguishable blocks of ocean drilling rights at the Gulf of Mexico
- Exactly one of the blocks contains oil worth C euros
- The price for each block is C/n euros
- A seismologist offers BP the results of a survey of block #3, which indicates definitively whether the block contains oil
- How much should BP be willing to pay for the information?
  - With probability $1/n$, the survey will indicate oil in block #3, in which case BP will buy the block for C/n euros and make a profit of $(n-1)C/n$ euros
  - With probability $(n-1)/n$, the survey will show that the block contains no oil, in which case BP will buy a different block

- Now the probability of finding oil in one of the other blocks changes to $1/(n-1)$, so BP makes an expected profit of $C/(n-1) - C/n = C/n(n-1)$ euros

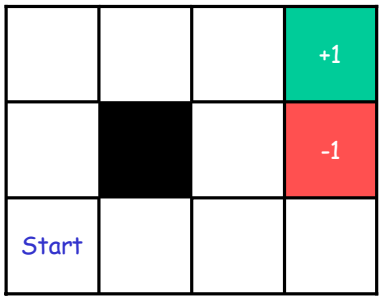
- Now we can calculate the expected profit, given the survey information:

$$(1/n)\cdot((n-1)C/n) + ((n-1)/n)\cdot(C/n(n-1)) = C/n$$

- Therefore, BP should be willing to pay the seismologist up to the price of the block itself
- With the information, one's course of action can be changed to suit the actual situation
- Without the information, one has to do what's best on average over the possible situations

17 MAKING COMPLEX DECISIONS

- The agent's utility now depends on a sequence of decisions
- In the following 4×3 grid environment the agent makes a decision to move (U, R, D, L) at each time step
- When the agent reaches one of the goal states, it terminates
- The environment is fully observable — the agent always knows where it is



- If the environment were deterministic, a solution would be easy: the agent will always reach +1 with moves [U, U, R, R, R]
- Because actions are unreliable, a sequence of moves will not always lead to the desired outcome
- Let each action achieve the intended effect with probability 0.8 but with probability 0.1 the action moves the agent to either of the right angles to the intended direction
- If the agent bumps into a wall, it stays in the same square
- Now the sequence [U, U, R, R, R] leads to the goal state with probability $0.8^5 = 0.32768$
- In addition, the agent has a small chance of reaching the goal by accident going the other way around the obstacle with a probability $0.1^4 \times 0.8$, for a grand total of 0.32776

- A *transition model* specifies outcome probabilities for each action in each possible state
- Let $T(s, a, s')$ denote the probability of reaching state s' if action a is done in state s
- The transitions are *Markovian* in the sense that the probability of reaching s' depends only on s and not the earlier states
- We still need to specify the utility function for the agent
- The decision problem is sequential, so the utility function depends on a sequence of states — an environment history — rather than on a single state
- For now, we will simply stipulate that is each state s, the agent receives a *reward* $R(s)$, which may be positive or negative

- For our particular example, the reward is -0.04 in all states except in the terminal states
- The utility of an environment history is just (for now) the sum of rewards received
- If the agent reaches the state +1, e.g., after ten steps, its total utility will be 0.6
- The small negative reward gives the agent an incentive to reach [4, 3] quickly

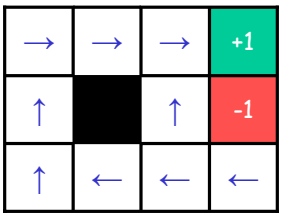
- A sequential decision problem for a fully observable environment with
  - A Markovian transition model and
  - Additive rewards

  is called a *Markov decision problem* (MDP)

- An MDP is defined by the following three components:
  - Initial state S_0,
  - Transition model $T(s, a, s')$, and
  - Reward function $R(s)$

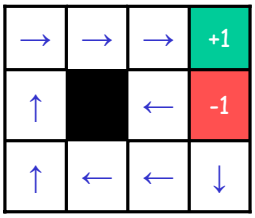
- As a solution to an MDP we cannot take a fixed action sequence, because the agent might end up in a state other than the goal
- A solution must be a policy, which specifies what the agent should do for any state that the agent might reach
- The action recommended by policy π for state s is $\pi(s)$
- If the agent has a complete policy, then no matter what the outcome of any action, the agent will always know what to do next

- Each time a given policy is executed starting from the initial state, the stochastic nature of the environment will lead to a different environment history
- The quality of a policy is therefore measured by the expected utility of the possible environment histories generated by the policy
- An optimal policy π^* yields the highest expected utility

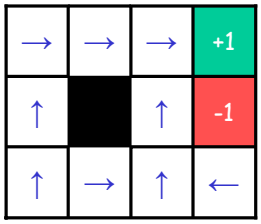


- A policy represents the agent function explicitly and is therefore a description of a simple reflex agent

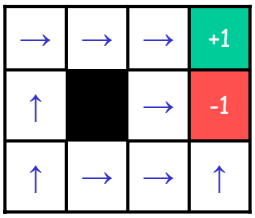
- $-0.0221 < R(s) < 0$:



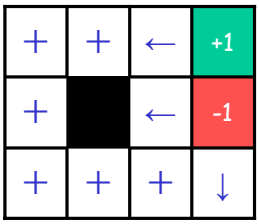
- $-0.4278 < R(s) < -0.0850$:



- $R(s) < -1.6284$:



- $R(s) > 0$:



Optimality in sequential decision problems

- In case of an *infinite horizon* the agent's action time has no upper bound
- With a finite time horizon, the optimal action in a given state could change over time — the optimal policy for a finite horizon is *nonstationary*
- With no fixed time limit, on the other hand, there is no reason to behave differently in the same state at different times, and the optimal policy is stationary

- The *discounted* utility of a state sequence $s_0, s_1, s_2, \ldots$ is
  $$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots,$$
  where $0 < \gamma \leq 1$ is the discount factor

- When $\gamma = 1$, discounted rewards are exactly equivalent to additive rewards
- The latter rewards are a special case of the former ones
- When γ is close to 0, rewards in the future are viewed as insignificant

- If an infinite horizon environment does not contain a terminal state or if the agent never reaches one, then all environment histories will be infinitely long
- Then, utilities with additive rewards will generally be infinite
- With discounted rewards ($\gamma < 1$), the utility of even an infinite sequence is finite

- Let R_{max} be an upper bound for rewards. Using the standard formula for the sum of an infinite geometric series yields:

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

- *Proper policy* guarantees that the agent reaches a terminal state when the environment contains such
- With proper policies infinite state sequences do not pose a problem, and we can use $\gamma = 1$ (i.e., additive rewards)

- An optimal policy using discounted rewards is

$$\pi^* = \arg\max_\pi E[\, \sum_{t=0,\ldots,\infty} \gamma^t R(s_t) \mid \pi \,],$$

where the expectation is taken over all possible state sequences that could occur, given that the policy is executed

17.2 Value Iteration

- For calculating an optimal policy we
  - calculate the utility of each state and
  - then use the state utilities to select an optimal action in each state
- The utility of a state is the expected utility of the state sequence that might follow it
- Obviously, the state sequences depend on the policy π that is executed
- Let s_t be the state the agent is in after executing π for t steps
- Note that s_t is a random variable
- Then we have

$$U^{\pi}(s) = E[\, \sum_{t=0,\ldots,\infty} \gamma^t R(s_t) \mid \pi, s_0 = s \,]$$

- The true utility of a state $U(s)$ is just $U^{\pi*}(s)$
- $R(s)$ is the short-term reward for being in s, whereas $U(s)$ is the long-term total reward from s onwards
- In our example grid the utilities are higher for states closer to the +1 exit, because fewer steps are required to reach the exit

0.812	0.868	0.912	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

- The agent may select actions using the MEU principle

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s, a, s') U(s') \qquad (*)$$

- The utility of state s is the expected sum of discounted rewards from this point onwards, hence, we can calculate it:
  - Immediate reward in state s, $R(s)$, +
  - The expected discounted utility of the next state, assuming that the agent chooses the optimal action

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

- This is called the Bellman equation
- If there are n possible states, then there are n Bellman equations, one for each state

$$\begin{aligned} U(1,1) = -0.04 + \gamma \max\{ & 0.8\,U(1,2) + 0.1\,U(2,1) + 0.1\,U(1,1), && (U)\\ & 0.9\,U(1,1) + 0.1\,U(1,2), && (L)\\ & 0.9\,U(1,1) + 0.1\,U(2,1), && (D)\\ & 0.8\,U(2,1) + 0.1\,U(1,2) + 0.1\,U(1,1)\ \} && (R) \end{aligned}$$

Using the values from the previous picture, this becomes:

$$\begin{aligned} U(1,1) = -0.04 + \\ \gamma \max\{ & 0.6096 + 0.0655 + 0.0705 = 0.7456, && (U)\\ & 0.6345 + 0.0762 = 0.7107, && (L)\\ & 0.6345 + 0.0655 = 0.7000, && (D)\\ & 0.5240 + 0.0762 + 0.0705 = 0.6707\ \} && (R) \end{aligned}$$

Therefore, Up is the best action to choose

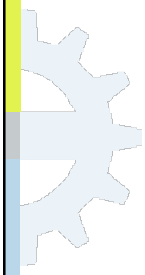
- Simultaneously solving the Bellman equations using does not work using the efficient techniques for systems of linear equations, because max is a nonlinear operation
- In the iterative approach we start with arbitrary initial values for the utilities, calculate the right-hand side of the equation and plug it into the left-hand side

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

  where index i refers to the utility value of iteration i
- If we apply the Bellman update infinitely often, we are guaranteed to reach an equilibrium, in which case the final utility values must be solutions to the Bellman equations
- They are also the unique solutions, and the corresponding policy is optimal

17.3 Policy Iteration

- Beginning from some initial policy π_0 alternate:
  - **Policy evaluation**: given a policy π_i, calculate $U_i = U^{\pi_i}$, the utility of each state if π_i were to be executed
  - **Policy improvement**: Calculate the new MEU policy π_{i+1}, using one-step look-ahead based on U_i (Equation (*))
- The algorithm terminates when the policy improvement step yields no change in utilities
- At this point, we know that the utility function U_i is a fixed point of the Bellman update and a solution to the Bellman equations, so π_i must be an optimal policy
- Because there are only finitely many policies for a finite state space, and each iteration can be shown to yield a better policy, policy iteration must terminate



- Because at the ith iteration the policy π_i specifies the action $\pi_i(s)$ in state s, there is no need to maximize over actions in policy iteration
- We have a simplified version of the Bellman equation:

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

- Now the nonlinear max has been removed, and we have linear equations
- A system of linear equations with n equations with n unknowns can be solved exactly in time $O(n^3)$ by standard linear algebra methods
- Instead of using a cubic amount of time to reach the exact solution, we can instead perform some number simplified value iteration steps to give a reasonably good approximation of the utilities