Markov Decision Process Notes (Math 562)

notes taken from Aritificial Intelligence, Chapter 16/ Chapter 17, Making complex decisions, Russell and Norvig

https://en.wikipedia.org/wiki/Artificial_Intelligence:_A_Modern_Approach

http://aima.cs.berkeley.edu/

Example Evironment

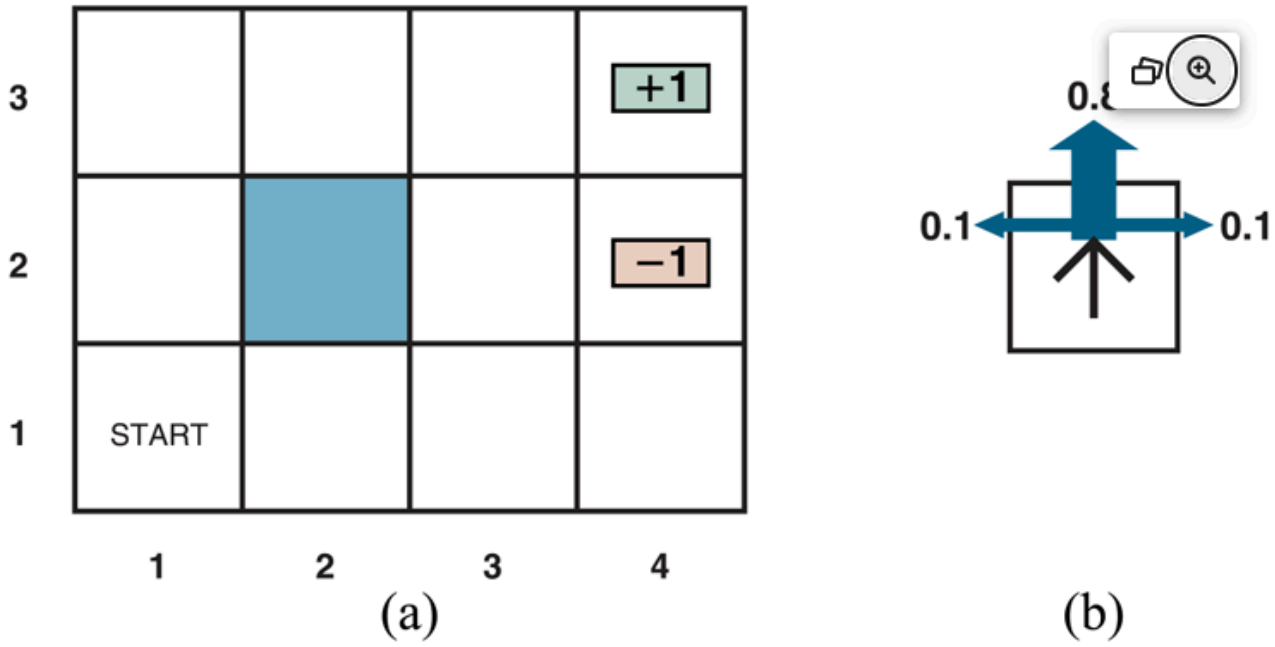
Evironment

- Suppose that an agent is situated in the 4×3 environment shown in Figure 17
- Beginning in the start state, it must choose an action at each time step.
- The interaction with the environment **terminates** when the agent reaches one of the goal states, marked +1 or -1
- the actions available to the agent in each state are given by Actions (s), sometimes abbreviated to $A(s)$
- We assume for now that the environment is fully observable, so that the agent always knows where it is.

Actions

- the actions in every state are $A(s) =$ Up, Down, Left, and Right.

Figure 17.1



(a) A simple, stochastic 4×3 environment that presents the agent with a sequential decision problem. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision with a wall results in no movement. Transitions into the two terminal states have reward $+1$ and -1, respectively, and all other transitions have a reward of -0.04.

If the environment were deterministic, a solution would be easy

$[Up, Up$, Right, Right, Right]$

Unfortunately, the environment won't always go along with this solution, because the actions are unreliable. The particular model of stochastic motion that we adopt is illustrated in Figure 17.1(b).

Each action achieves

- the intended effect with probability 0.8 ,
- but the rest of the time, the action moves the agent at right angles to the intended direction.

Furthermore, if the agent bumps into a wall, it stays in the same square.

Example

For example, from the start square $(1,1)$, the action Up moves the agent to $(1,2)$ with probability 0.8 , but with probability 0.1 , it moves right to $(2,1)$, and with probability 0.1 , it moves left, bumps into the wall, and stays in $(1,1)$. In such an environment, the sequence [Up, Up,Right,Right,Right] goes up around the barrier and reaches the goal state at $(4,3)$ with probability $0.8^5 = 0.32768$. There is also a small chance of accidentally reaching the goal by going the other way around with probability $0.1^4 \times 0.8$, for a grand total of

0.32776 .

Transition Model

The **transition model** (or just "model," when the meaning is clear) describes the outcome of each action in each state. Here, the outcome is stochastic,

- so we write $P\left(s' \mid s, a\right)$ for the probability of reaching state s' if action a is done in state s. (Some authors write $T\left(s, a, s'\right)$ for the transition model.) We will assume that transitions are Markovian: the probability of reaching s' from s depends only on s and not on the history of earlier states.

utility function for the agent.

To complete the definition of the task environment, we must specify the utility function for the agent. Because the decision problem is sequential, the utility function will depend on a sequence of states and actions-an environment history-rather than on a single state.

For now, we simply stipulate:

- that for every transition from s to s' via action a, the agent receives a reward $R\left(s, a, s'\right)$. The rewards may be positive or negative, but they are bounded by $\pm R_{\max}$.

For our particular example,

- the reward is -0.04 for all transitions except those entering terminal states (which have rewards +1 and -1).
- The utility of an environment history is just (for now) the sum of the rewards received.
- For example, if the agent reaches the +1 state after 10 steps, its total utility will be $(9 \times -0.04) + 1 = 0.64$. The negative reward of -0.04 gives the agent an incentive to reach $(4, 3)$ quickly Another way of saying this is that the agent does not enjoy living in this environment and so it wants to leave as soon as possible.

Definition of Markov decision process, or MDP,

To sum up: a sequential decision problem for a fully observable, stochastic environment with a Markovian transition model and additive rewards is called a Markov decision process, or MDP, and consists of

- a set of states (with an initial state s_0);
- a set ACTIONS(s) of actions in each state;
- a transition model $P\left(s' \mid s, a\right)$; and
- a reward function $R\left(s, a, s'\right)$.

Methods for solving MDPs usually involve **dynamic programming:** simplifying a problem by recursively breaking it into smaller pieces and remembering the optimal solutions to the pieces.

Solution of an MDP

- No fixed action sequence can solve the problem, because the agent might end up in a state other than the goal.
- Therefore, a solution must specify what the agent should do for any state that the agent might reach. A solution of this kind is called a policy.
- It is traditional to denote a policy by π, and $\pi(s)$ is the action recommended by the policy π for state s.

No matter what the outcome of the action, the resulting state will be in the policy, and the agent will know what to do next.

definition of a policy:

Function $\pi(s) \in A(s)$

definition of expected utility of a policy:

Each time a given policy is executed starting from the initial state, the stochastic nature of the environment may 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 that policy.

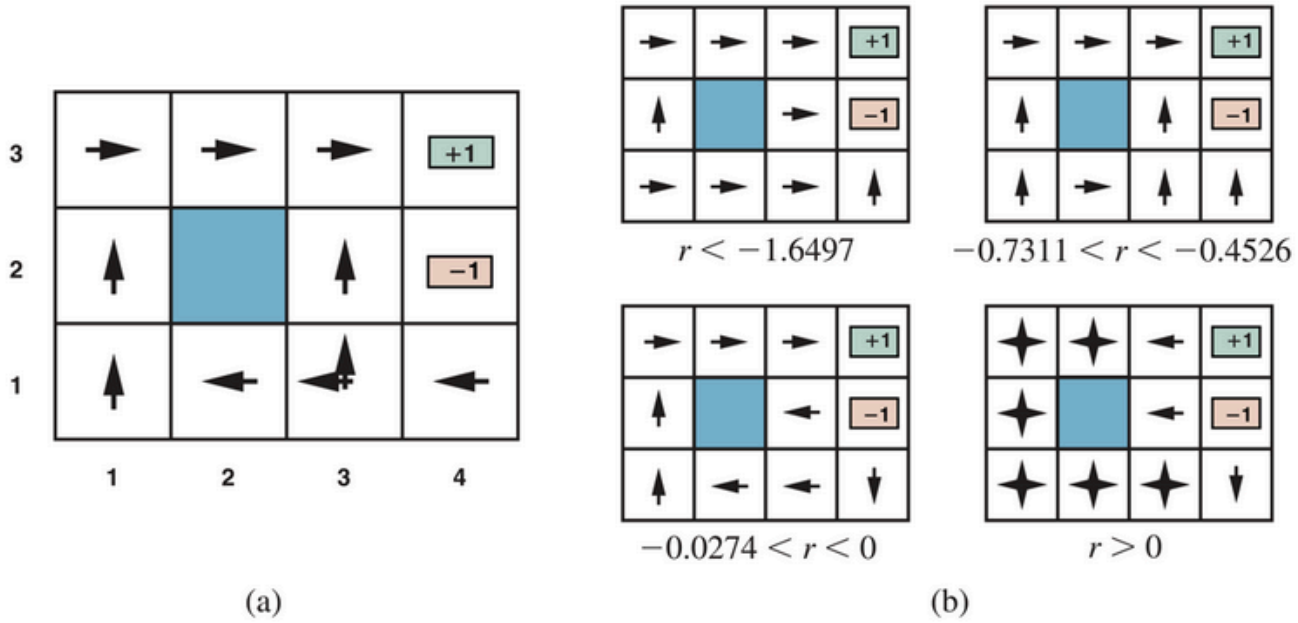
definition of optimal policy:

An optimal policy is a policy that yields the highest expected utility. (May not be unique)

We use π^* to denote an optimal policy. Given π^*, the agent decides what to do by consulting its current percept, which tells it the current state s, and then executing the action $\pi^*(s)$. A policy represents the agent function explicitly and is therefore a description of a simple reflex agent, computed from the information used for a utility-based agent.

The optimal policies for the world of Figure 17.1 are shown in Figure 17.2(a). There are two policies because the agent is exactly indifferent between going left and going up from $(3, 1)$: going left is safer but longer, while going up is quicker but risks falling into $(4, 2)$ by accident. In general there will often be multiple optimal policies.

Figure 17.2



(a) The optimal policies for the stochastic environment with $r = -0.04$ for transitions between nonterminal states. There are two policies because in state (3,1) both *Left* and *Up* are optimal. (b) Optimal policies for four different ranges of r.

The balance of risk and reward changes depending on the value of $r = R(s, a, s')$ for transitions between nonterminal states.

- The policies shown in Figure 17.2(a) are optimal for $-0.0850 < r < -0.0273$.
- Figure 17.2(b) shows optimal policies for four other ranges of r . When $r < -1.6497$, life is so painful that the agent heads straight for the nearest exit, even if the exit is worth -1 .
- When $-0.7311 < r < -0.4526$, life is quite unpleasant; the agent takes the shortest route to the +1 state from $(2, 1)$, $(3, 1)$, and $(3, 2)$, but from $(4, 1)$ the cost of reaching +1 is so high that the agent prefers to dive straight into -1 .
- When life is only slightly dreary $(-0.0274 < r < 0)$, the optimal policy takes no risks at all. In $(4, 1)$ and $(3, 2)$, the agent heads directly away from the -1 state so that it cannot fall in by accident, even though this means banging its head against the wall quite a few times.
- Finally, if $r > 0$, then life is positively enjoyable and the agent avoids both exits. As long as the actions in $(4, 1)$, $(3, 2)$, and $(3, 3)$ are as shown, every policy is optimal, and the agent obtains infinite total reward because it never enters a terminal state. It turns out that there are nine optimal policies in all for various ranges of r;

discussion of MDP

The introduction of uncertainty brings MDPs closer to the real world than deterministic search problems. For this reason, MDPs have been studied in several fields,

- including AI,
- operations research,
- economics, and
- control theory.

Dozens of solution algorithms have been proposed.

First, however, we spell out in more detail the definitions of utilities, optimal policies, and models for MDPs.

Utilities over time

In the MDP example in Figure 17.1 the performance of the agent was measured by a sum of rewards for the transitions experienced. This choice of performance measure is not arbitrary, but it is not the only possibility for the utility function on environment histories, which we write as

$U_h\left([s_0, a_0, s_1, a_1 \ldots, s_n]\right)$.

The first question to answer is whether there is a

- finite horizon or
- an infinite horizon

for decision making. A finite horizon means that there is a fixed time N after which nothing matters-the game is over, so to speak. Thus,

$$U_h\left([s_0, a_0, s_1, a_1, \ldots, s_{N+k}]\right) = U_h\left([s_0, a_0, s_1, a_1, \ldots, s_N]\right) \quad (1)$$

- Finite horizon, if running out of time, may take a different policy (End game)
  - so $\pi = \pi(s, t)$

  (Could make the time part of the state, but this obscures the special nature of time (which only goes forward)
- Infinite horizon, no reason to behave differently in the same state (at different times)
  - so $\pi = \pi(s)$
  - This means the optimal policy is *stationary*
  - Thus policies for infinite horizon case are simpler. So we deal with infinite horizon case.
  - This does not necessarity mean that all state sequences are infinitte, just that there is no fixed deadline. So can still have terminal states.
  - Example Tennis Set: Play until first player wins 6 games, but must win by two.

How to calculate the utility of state sequences.

Additive discounted rewards:

$$U_h\left([s_0, a_0, s_1, a_1, \ldots, s_{N+k}]\right) = R(s_0, a_0, S_1) + \gamma R(s_1, a_1, S_2) + \gamma^2 R(s_2, a_2, S_3) + \ldots \quad (2)$$

For a discount factor $\gamma \in (0, 1]$

- When γ near 0, rewards in distant future are viewed as insignificant
- When γ near 1, agent is more willing to wait for long-term rewards
- When $\gamma = 1$, the discounted case becomes the special case of purely additive rewards. (However, can have an infinite reward if no termination!)

Reasons for additive rewards:

1. Empirical: humans and animal prefer near term rewards.
2. Economic: money is discounted, because can invest. Equivalent interest rate: $1/\gamma - 1$ E.g. .9 becomes 11 percent.
3. Uncertainty about the true rewards: they may not arrive. Under certain conditions, equivalent to a probabilty $1 - \gamma$ of accidental termination (so no rewards)
4. Fourth: in utility theory, idea of stationary preferences leads to additive discounted rewards.
5. Fifth: Mathematical, to avoid infinite rewards.

Consequences/Analysis of Additive discounted rewards

finite rewards

If environment history is infinite, but rewards bounded by $|r| \leq \mathbb{R}_{\max}$

$$U_h\left([s_0, a_0, s_1, a_1, \ldots, s_{N+k}]\right) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, S_{t+1}) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = \frac{R_{\max}}{1-\gamma} \quad (3)$$

proper policites: finite histories

If the environment contains terminal states and if the agent is guaranteed to get to one eventually, then we will never need to compare infinite sequences. A policy that is guaranteed to reach a terminal state is called a **proper policy**.

With proper policies, we can use $\gamma = 1$ (i.e., additive undiscounted rewards). The first three policies shown in Figure 17.2(b) are proper, but the fourth is improper. It gains infinite total reward by staying away from the terminal states when the reward for transitions between nonterminal states is positive.

The existence of improper policies can cause the standard algorithms for solving MDPs to fail with additive rewards, and so provides a good reason for using discounted rewards.

average rewards (nice but problematic)

Infinite sequences can be compared in terms of the average reward obtained per time step. Suppose that transitions to square $(1,1)$ in the 4×3 world have a reward of 0.1 while transitions to other nonterminal states have a reward of 0.01 . Then a policy that does its best to stay in $(1,1)$ will have higher average reward than one that stays elsewhere. Average reward is a useful criterion for some problems, but the analysis of average-reward algorithms is complex.

Additive discounted rewards present the fewest difficulties in evaluating histories, so we shall use them henceforth.

Why expected Utilities?

Because of uncertainty there is no best policy!

Example,

fair coin toss, with reward $+10$ for heads, and reward of -1 for tails,

The policy woth best expected rewards is is $\pi = H$, to bet heads.

Then for one toss, expected reward is

$$E[R(\pi)] = +10/2 - 1/2 = 5.5 \tag{4}$$

Over many tosses, discounted future rewards is :

$$U^{\pi} = \sum \gamma^t E(R(\pi)) = \frac{5.5}{1-\gamma} \tag{5}$$

Which is finite.

Optimal policies and the utilities of states

Having decided that the utility of a given history is the sum of discounted rewards, we can

> *compare policies by comparing the expected utilities obtained when executing them.*

We assume the agent is in some initial state s and

define

define random state varaible

define S_t (a random variable) to be the state the agent reaches at time t when executing a particular policy π. (Obviously, $S_0 = s$, the state the agent is in now.)

The probability distribution over state sequences $S_1, S_2, \ldots$, is determined by

- the initial state s,
- the policy π, and
- the transition model for the environment.

The expected utility obtained by executing π starting in s is given by

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R\left(S_t, \pi\left(S_t\right), S_{t+1}\right)\right] \tag{6}$$

where the expectation E is with respect to the probability distribution over state sequences determined by s and π.

Now, out of all the policies the agent could choose to execute starting in s, one (or more) will have higher expected utilities than all the others. We use π_s^* to denote one of these policies:
(17.3)

$$\pi_s^* \in \underset{\pi}{\operatorname{argmax}}\, U^{\pi}(s) \tag{7}$$

Remember that π_s^* is a policy,

- so it recommends an action for every state
- its connection with s in particular is that it's an optimal policy when s is the starting state.

Theorem

A remarkable consequence of using discounted utilities with infinite horizons is that the optimal policy is independent of the starting state.

(Of course, the action sequence won't be independent; remember that a policy is a function specifying an action for each state.)

Intuition:

This fact seems intuitively obvious: if policy π_a^* is optimal starting in a and policy π_b^* is optimal starting in b, then, when they reach a third state c, there's no good reason for them to disagree with each other, or with π_c^*, about what to do next. So we can simply write π^* for an optimal policy.

The proof follows directly from the uniqueness of the utility function on states, below.

Utility of a state

Utility of a state

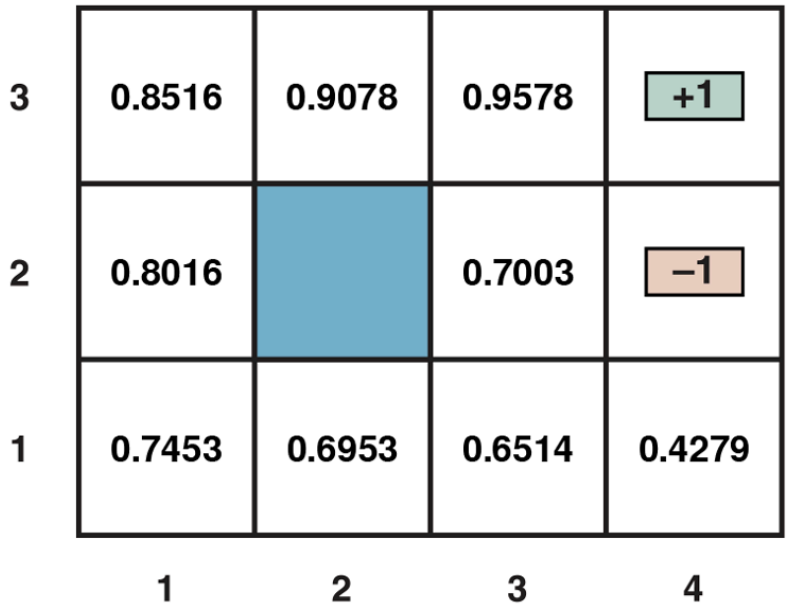
Given this definition, the true utility of a state is just $U^{\pi^*}(s)$-that is, the expected sum of discounted rewards if the agent executes an optimal policy. We write this as $U(s)$,

$$U(s) = U^{\pi^*}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R\left(S_t, \pi^*\left(S_t\right), S_{t+1}\right)\right] \tag{8}$$

(For any fixed choice of optimal policy)

Figure 17.3 shows the utilities for the 4×3 world. Notice that the utilities are higher for states closer to the +1 exit, because fewer steps are required to reach the exit.

FIGURE 17.3



The utilities of the states in the 4×3 world with $\gamma = 1$ and $r = -0.04$ for transitions to nonterminal states.

The utility function $U(s)$ allows the agent to select actions by using

the principle of maximum expected utility.

choose the action that maximizes the reward for the next step plus the expected discounted utility of the subsequent state:

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P\left(s' \mid s, a\right)\left[R\left(s, a, s'\right) + \gamma U\left(s'\right)\right] \tag{9}$$

We have defined the utility of a state, $U(s)$, as the expected sum of discounted rewards from that point onwards.

From this, it follows that there is a direct relationship between the utility of a state and the utility of its

From this, it follows that there is a direct relationship between the utility of a state and the utility of its neighbors:

- the utility of a state is the expected reward for the next transition plus the discounted utility of the next state, assuming that the agent chooses the optimal action. That is, the utility of a state is given by

$$U(s) = \max_{a \in A(s)} \sum_{s'} P\left(s' \mid s, a\right) \left[R\left(s, a, s'\right) + \gamma U\left(s'\right)\right] \tag{10}$$

This is called the Bellman equation, after Richard Bellman (1957).

The utilities of the states defined as the expected utility of subsequent state sequences-are solutions of the set of Bellman equations. In fact, they are the unique solutions,

Policy from Utility

- Given the utility function, can find the (an) optimal policy $\pi*$ as any element of the argmax in the preceding equation

Bellman equation Example

Let us look at one of the Bellman equations for the 4×3 world. The expression for $U(1,1)$ is

$$\begin{aligned} \max\{&[0.8(-0.04 + \gamma U(1,2)) + 0.1(-0.04 + \gamma U(2,1)) + 0.1(-0.04 + \gamma U(1,1))] \\ &[0.9(-0.04 + \gamma U(1,1)) + 0.1(-0.04 + \gamma U(1,2))] \\ &[0.9(-0.04 + \gamma U(1,1)) + 0.1(-0.04 + \gamma U(2,1))] \\ &[0.8(-0.04 + \gamma U(2,1)) + 0.1(-0.04 + \gamma U(1,2)) + 0.1(-0.04 + \gamma U(1,1))]\} \end{aligned} \tag{11}$$

where the four expressions correspond to Up, Left, Down and Right moves. When we plug in the numbers from Figure 17.3 , with $\gamma = 1$, we find that Up is the best action.

Action Utility Function

Another important quantity is the **action-utility function, or Q-function:**

- $Q(s, a)$ is the expected utility of taking a given action in a given state. The Q-function is related to utilities in the obvious way:

$$U(s) = \max_{a} Q(s, a) \tag{12}$$

Furthermore, the optimal policy can be extracted from the Q-function as follows:

$$\pi^*(s) = \operatorname*{argmax}_{a} Q(s, a) \tag{13}$$

We can also develop a Bellman equation for Q-functions, noting that the expected total reward for taking an

We can also develop a Bellman equation for Q-functions, noting that the expected total reward for taking an action is its immediate reward plus the discounted utility of the outcome state, which in turn can be expressed in terms of the Q-function:
(17.8)

$$\begin{aligned} Q(s,a) &= \sum_{s'} P\left(s' \mid s,a\right)\left[R\left(s,a,s'\right)+\gamma U\left(s'\right)\right] \\ &= \sum_{s'} P\left(s' \mid s,a\right)\left[R\left(s,a,s'\right)+\gamma \max_{a'} Q\left(s',a'\right)\right] \end{aligned} \tag{14}$$

Solving the Bellman equations for U (or for Q) gives us what we need to find an optimal policy. The Q-function shows up again and again in algorithms for solving MDPs, so we shall use the following definition:

$$\begin{aligned} &\text{function Q}-\text{VALUE}(mdp,s,a,U) \text{ returns a utility value} \\ &\text{return } \sum_{s'} P\left(s' \mid s,a\right)\left[R\left(s,a,s'\right)+\gamma U\left[s'\right]\right] \end{aligned} \tag{15}$$

Reward scales

an affine transformation transformation of rewards will leave the optimal policy unchanged in an MDP:

$$R'\left(s,a,s'\right) = mR\left(s,a,s'\right)+b \tag{16}$$

- Optimal policies unchanged.

It turns out, however, that the additive reward decomposition of utilities leads to significantly more freedom in defining rewards. Let $\Phi(s)$ be any function of the state s. Then, according to the **shaping theorem**, the following transformation leaves the optimal policy unchanged:
(17.9)

$$R'\left(s,a,s'\right) = R\left(s,a,s'\right)+\gamma\Phi\left(s'\right)-\Phi(s) \tag{17}$$

Shaping theorem

To show that this is true, we need to prove that two MDPs, M and M', have identical optimal policies as long as they differ only in their reward functions as specified above.

[skip proof]