1 Example: Markov Chain with Rewards

1.1 Model

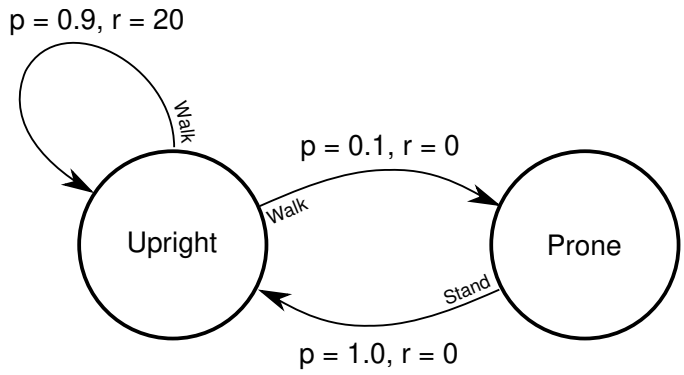
Imagine a small automation toy which, when you start it, will start walking, and if it falls over it can get itself upright again.

For a very simple model of its behaviour we can say that it has two **states**: *upright* and *prone*, let's call them s_1 and s_2. It also has two **actions**: *walk* and *stand*. If the automaton *walks* it might either be unlucky and fall over *prone* or it keeps standing *upright*. If it has fallen over it can not take any further steps, but have to *stand* to get back into an *upright* state before taking further steps.

In other words, the robot can't *walk* while *prone*, and it can't *stand* if it is already *upright*, so there's only one applicable action per state - it can modelled as a Markov Chain.

Assume the following **transitioning probabilities**: If the automation *walks* it will remain *upright* (transition from state s_1 to s_1) with probability 0.9, in all other cases it will fall *prone* (that is transition from state s_1 to s_2) with probability 0.1. If the robot *stands* it will always succeed and become *upright*, so the transition probability from s_2 to s_1 is 1.0.

Moreover, it is always better to keep standing when taking a step, so the **reward** for transitioning from s_1 to s_1 is 20. Let the rewards for all other transitions be 0. From this description we can draw a figure:



1.2 Analytic solution

Now we want to express the *value* of the respective states in this chain, that is the expected reward of each state if the automation keeps running forever - sometimes falling over and sometimes standing.

This can be done as you've seen on the lectures using the *Bellman Equation*, which in its general form looks like

$$v(s) = \max_{a \in A} \sum_{s' \in S} P(s, a, s') \left[R(s, a, s') + \gamma v(s') \right].$$

The above version is also the one used in the programming exercise by the value iteration functions.

For this simple example, we wouldn't have to use value iteration however. It is quite possible to solve the system analytically as follows.

As we only have one applicable action per state we don't have to find the maximum action and can simplify the Bellman equation:

$$v(s) = \sum_{s' \in S} P(s, s') \left[R(s, s') + \gamma v(s') \right].$$

Now, $P(s, s')$ is the probability to go from state s to state s', and $R(s, s')$ is the associated reward. You can see these values in the text and figure above. For instance, in our case $P(s_1, s_2) = 0.1, R(s_1, s_2) = 0$, that is to transition from *Standing* to *Fallen over* by *Taking a step*. Note that the action is implicit, as there is only one way to go from standing to fallen.

γ is the *discount factor* which tells us how much weight to put in future states.

Now, let's express the value of the two states given the rewards and probabilities:

$$
\begin{aligned}
v(s_1) &= \sum_{s' \in \{s_1, s_2\}} P(s_1, s') \left[R(s_1, s') + \gamma v(s') \right] \\
&= P(s_1, s_1) \left[R(s_1, s_1) + \gamma v(s_1) \right] + P(s_1, s_2) \left[R(s_1, s_2) + \gamma v(s_2) \right] \\
&= 0.9 \left[20 + \gamma v(s_1) \right] + 0.1 \left[0 + \gamma v(s_2) \right] \\
&= 0.9 \left[20 + \gamma v(s_1) \right] + 0.1 \gamma v(s_2) \\
v(s_2) &= \sum_{s' \in \{s_1\}} P(s_2, s') \left[R(s_2, s') + \gamma v(s') \right] \\
&= P(s_2, s_1) \left[R(s_2, s_1) + \gamma v(s_1) \right] \\
&= 1.0 \left[0 + \gamma v(s_1) \right] \\
&= \gamma v(s_1)
\end{aligned}
$$

For simplicity, let's call $v(s_1) = v_1, v(s_2) = v_2$ and write out the above results again:

$$
\begin{aligned}
v_1 &= 0.9 \left[20 + \gamma v_1 \right] + 0.1 \gamma v_2 \\
v_2 &= \gamma v_1
\end{aligned}
$$

Now this is a quite straight forward system of equations, which can be solved if we know a value for γ. So, say that **we want to find out the value of** s_1 (*upright*), **given** $\gamma = 0.5$:

We'd first plug in $v_2 = \gamma v_1$ in the first equation, giving us

$$
v_1 = 0.9 \left[20 + \gamma v_1 \right] + 0.1 \gamma^2 v_1.
$$

Solving for v_1 and inserting $\gamma = 0.5$ gives us

$$
\begin{aligned}
v_1 &= \frac{0.9 \times 20}{1 - 0.9\gamma - 0.1\gamma^2} \\
&= \frac{0.9 \times 20}{1 - 0.9 \times 0.5 - 0.1 \times 0.5^2} \\
&\approx 34.3.
\end{aligned}
$$

So, the value of state s_1 (upright) is approximately 34.3, given the parameters of the question.