# Reinforcement Learning Homework 1 - Annotations

Damian Valle, Martin Schuck

November 11, 2020

## Machine Replacement

### Solution by backwards induction

The problem can be solved by backwards induction. In general, the solution is given by the equation

$$u_k = \underset{a_k}{max}[r_k(s,a) + P(a)J_{k+1}].$$

With the given transition probabilities P(replace) and P(broken), $\theta = 0.5$, a replacement reward of -8 and a cost reward of -6, this can be written as

$$u_k = \underset{a_k}{max}\left(\begin{pmatrix} 0.5u_{k+1}(1) + 0.5u_{k+1}(2) \\ 0.5u_{k+1}(1) + 0.5u_{k+1}(2) - 3 \\ u_{k+1}(3) - 6 \end{pmatrix}, \begin{pmatrix} -8 + u_{k+1}(1) \\ -8 + u_{k+1}(1) \\ -8 + u_{k+1}(1) \end{pmatrix}\right).$$

The expected rewards and optimal actions are then given as follows for a time horizon T=2:

| | T=0 | T=1 | T=2 |
|---|---|---|---|
| $u_k$ | $\begin{pmatrix} -\frac{3}{2} \\ -\frac{15}{2} \\ -8 \end{pmatrix}$ | $\begin{pmatrix} 0 \\ -3 \\ -6 \end{pmatrix}$ | $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ |
| $a_k$ | $\begin{pmatrix} c \\ c \\ r \end{pmatrix}$ | $\begin{pmatrix} c \\ c \\ c \end{pmatrix}$ | $\begin{pmatrix} c \\ c \\ c \end{pmatrix}$ |

with "c" standing for "continue" and "r" for "replace".

## Optimal stopping

### a) Modelling the problem

A naive way to model the problem such that the rewards become stationary is to simply include the number of tosses into the state. We therefore have $S = (\{1, ..., T\} \times \{1, ..., T\} \cup \emptyset)$, with $(t, n)$ denoting the number of throws and heads, and $\emptyset$ as the state after stopping the experiment. We therefore have $T^2 + 1$ states. The action space is given as $\{c, s\}$, c standing for continuing and s for stopping. The list of non-null transitions is as follows:

- $P(\emptyset|\emptyset, x) = 1$
- $P(\emptyset|(t, n), s) = 1$
- $P(\emptyset|(T, n), x) = 1$
- $P((t + 1, n)|(t, n), c) = 0.5$
- $P((t + 1, n + 1)|(t, n), c) = 0.5$

and the reward is zero for all states, except $r((t, n), s) = \frac{n}{t}$.
The Bellman equation in this case can be written as

$$V(s) = \underset{a}{max}\left\{r(s), \sum_{j\in S} P(j|s,a) * V(j)\right\} = \underset{a}{max}\left\{r(t,n), \frac{V((t+1,n)) + V((t+1,n+1))}{2}\right\}$$

## b) Induction proof

Intuitively, proposition A is correct. In order to prove the statement, we can show that it holds for $V_T$ and make a backwards induction. In the case of $V_T$, we have $V_T(T,n) = \underset{a}{max}\{r(T,n)\} = \frac{n}{T}$. The inequality $V_T(n+1) \geq V_T(n)$ therefore holds, since $\frac{n+1}{T} > \frac{n}{T}$. For the general case $V_t$, we have

$$\underset{a}{max}\left\{r(t,n+1), \frac{V_{t+1}(n+2)+V_{t+1}(n+1)}{2}\right\} \geq \underset{a}{max}\left\{r(t,n), \frac{V_{t+1}(n+1)+V_{t+1}(n)}{2}\right\}.$$

In case $r(t,n) > \frac{V_{t+1}(n+1)+V_{t+1}(n)}{2}$, the reward is the maximum and the inequality is obviously correct, since $r(t,n+1) > r(t,n)$. If not, then the inequality is also correct, since we have that

$$\frac{V_{t+1}(n+2)+V_{t+1}(n+1)}{2} \geq \frac{V_{t+1}(n+1)+V_{t+1}(n)}{2}$$
$$V_{t+1}(n+2) \geq V_{t+1}(n).$$

The correctness of the last inequality is established by backwards induction, since for all times ¿ t, the inequality is assumed to hold.

## d) Off-policy RL

Reinforcement learning relies on the exploration of the state space, both for on- and off-policy algorithms. Since option B) does not explore this space by only ever trying a single action, it is unsuitable for RL. We therefore have to choose behaviour policy A.