Implementation Assignment 5

Rong Yu and Finn Womack

June 6, 2017

1 Implementing the iterative value algorithm

First, we loaded the data into matlab using the dlmread function which returned a matrix of all the values appending zeros where needed. (I.e. The first row was m and n followed by several zeros, followed by the next several rows containing the action matrices smashed together, and the last row containing the reward vector.) We then reformatted the data into 4 variables and ran it through the iterative value function.

2 Results

2.1 $\beta = 0.1$

Using a discount factor of $\beta = 0.1$ we found the following optimal utility vector:

$$U = \begin{bmatrix} 0.1001 \\ 0.0090 \\ 0.0088 \\ 0.0090 \\ 1.0010 \\ 0.0068 \\ 0.0683 \\ 0.0086 \\ 0.0100 \\ 0.0896 \end{bmatrix} \tag{1}$$

as well as an optimal policy of:

$$V = \begin{bmatrix} 4 \\ 4 \\ 3 \\ 1 \\ 1 \\ 1 \\ 2 \\ 3 \\ 2 \\ 4 \end{bmatrix} \tag{2}$$

where the ith entry of each vector corresponds to the optimal policy/utility of the ith state.

2.2 $\beta = 0.9$

Next, using a discount factor of $\beta = 0.9$ we found the following optimal utility vector:

$$U = \begin{bmatrix} 3.3210 \\ 2.9234 \\ 2.8914 \\ 2.9234 \\ 3.6900 \\ 2.8407 \\ 3.1564 \\ 2.9071 \\ 2.9889 \\ 3.2482 \end{bmatrix} \tag{3}$$

as well as an optimal policy of:

$$V = \begin{bmatrix} 4 \\ 4 \\ 3 \\ 1 \\ 1 \\ 1 \\ 2 \\ 3 \\ 2 \\ 4 \end{bmatrix} \tag{4}$$

where the ith entry of each vector corresponds to the optimal policy/utility of the ith state. Something interesting to note is that while the discount factors gave different optimal utility vectors they both returned the same optimal policy.