Rajul Ramchandani & Conan Zhang

Professor Thomas Henderson

CS 4300 - 001

1 December 2016

Assignment A7: Value Iteration

1. INTRODUCTION: RAJUL

Markov Decision Problems are situations where an agent needs to take an action with uncertainty. This is because realistic situations are usually non-deterministic and require programs to make decisions based on probabilities. For the Wumpus World artificial intelligence, each state it can be in has a computable utility which indicates how useful the state is for the given task. The agent can then map these states to actions based on their utility in the form of a policy. Value Iteration is an algorithm used to compute an optimal policy based on a given transition model.

In our model, the Wumpus World agent's actions consists of:

$$A = \{UP, LEFT, DOWN, RIGHT\}$$

We seek to gather data and figures that:

1. Display comparable results to figure 17.2 on p. 648 of R&N for the generated policies where some lead to death and some lead to gold.
2. Display comparable results to figure 17.3 on p. 651 of R&N for the calculated utilities.
3. Show plots comparable to figure 17.5a on p. 653 of R&N for utility estimations over iterations for gamma values of 0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999.

2. METHOD: CONAN

The two algorithms developed for value iteration are a utility calculator CS4300_MDP_value_iteration and a policy generator CS4300_MDP_policy. The utility calculator allows us to calculate the utility of each space on a given board while the policy generator creates an optimal policy for the agent to follow.

The utility calculator starts out by calculating utilities for every state on the board. It does this by checking every possible action for a state and seeing which action is optimal with a given transition board multiplied by previous utilities. The transition board is based on the uncertainties set up for our agent. In our current model this is a probability of 0.8 for moving the selected direction and 0.1 for going to either side. After the most optimal action is selected, the utility is calculated with this action multiplied by a given discount factor and adding the reward for the current state. This discount factor simulates the agent favoring future rewards rather than current ones. In our model the discount factor is 0.999999. This loops for a given maximum number of iterations or until the difference between the currently calculated utilities and the previous calculated utilities are less than a given threshold. In our model the maximum number of iterations was 1000 and our threshold was set to 0.1. The policy generator then simply uses the utilities calculated from the utility calculator to determine the most optimal actions at every state on the board.

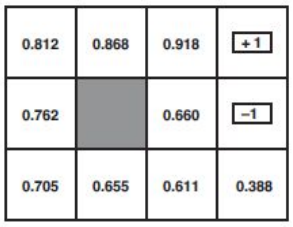
3. VERIFICATION OF PROGRAM: RAJUL

The table below represents the output of the Utilities from the Value_Iteration method(left) .

This is very close to the board provided in the textbook with the utilities.

0.8115	0.8678	0.9178	1
0.7615	0	0.6602	-1
0.7053	0.6553	0.6114	0.3879

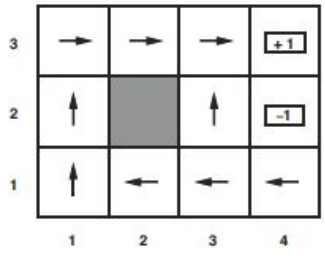
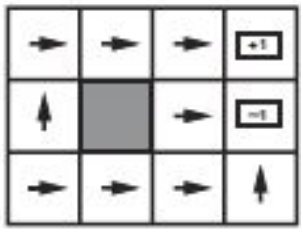
Table 1 : Value_Iteration values



This shows the correctness of the method.

The Table below shows the policy generation for varied values of the reward structure. Again, these values are compared to images from the textbook having number 1-up, 2-left, 3-down, 4-right.

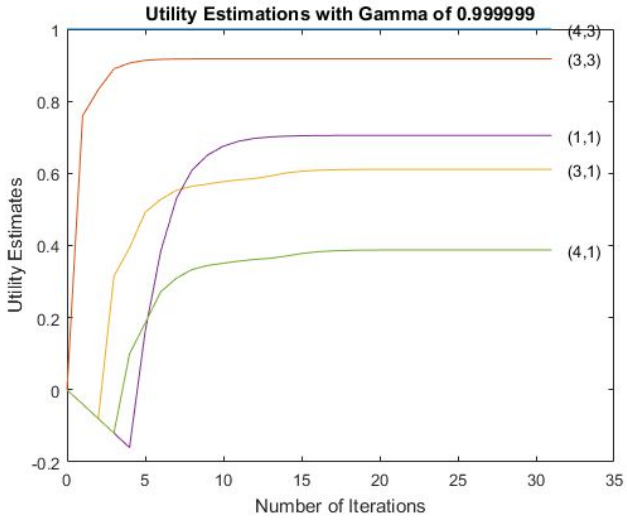
The values below show the variation of the behavior with different values of R. We see on the left that the the reward is fairly greater than the pit reward(-1) and so the policies show more probability of reaching the gold state. Contrastly, the board on the right have a much lesser reward than the pit square, hence it is seen that it is easier to reach the (-1) state at (4,2)

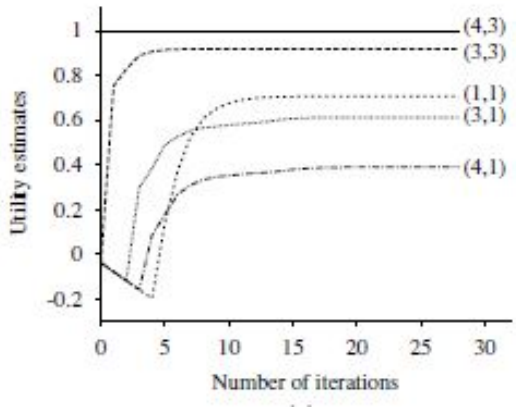
	For R(s) = (-0.04)	For R(s) < (-1.6284):
Optimal Policies in Figure 17.2 on p. 648 of R&N		

Optimal Policies Generated by our algorithm	4 4 4 1 1 1 1 1 1 2 2 2	4 4 4 1 1 1 4 1 4 4 4 1

Table 2: Policies with varying R

The graphs below shows the Utility Estimations with Gamma 0.9999999. This, when compared to the graphs provided in the textbook, look very similar.





Graphs: (left) Value_Iteration U estimates, (right) Textbook graph

4. DATA AND ANALYSIS: CONAN

The Wumpus World our agent will be generating a policy for is:

0	0	0	G
0	0	W	P
0	0	P	0
0	0	0	0

1) 4x4 R(s) = -1 :	**2)** 4x4 R(s) = -1700	**3)** 4x4 R(s) = -180	**4)** 4x4 R(s) = 1
4 4 1 1 1 2 1 1 1 2 1 3 1 2 3 3	4 4 4 1 4 4 1 1 4 4 1 1 4 4 1 1	4 4 4 1 1 1 1 1 1 1 1 2 1 1 2 2	3 2 2 1 1 2 1 1 1 2 1 3 1 2 3 3

The above tables represent policies with varying Reward values. There are 4 different cases for the policy behaviours. These cases were picked to showcase the variation on behaviour based on the various R.

These were run for a gamma of 0.999999 and 1000 max iterations. Also, the default policy was arbitrarily chosen as 1 for spots with Wumpus and pits and gold.

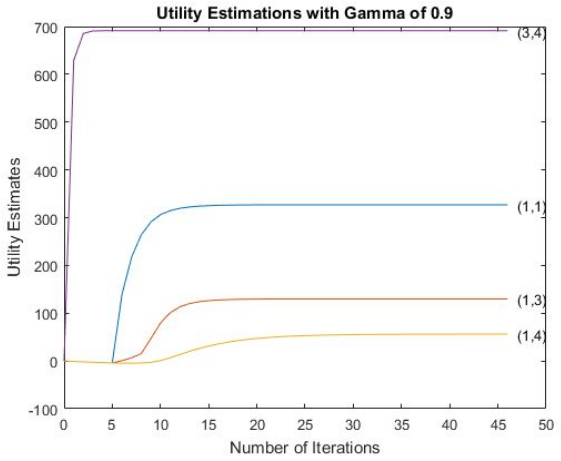
985.1198	986.5428	988.2664	1000
983.7458	982.7636	-1000	-1000
982.3437	981.13677	-1000	-377.9248
980.9560	979.8638	959.8447	949.8352

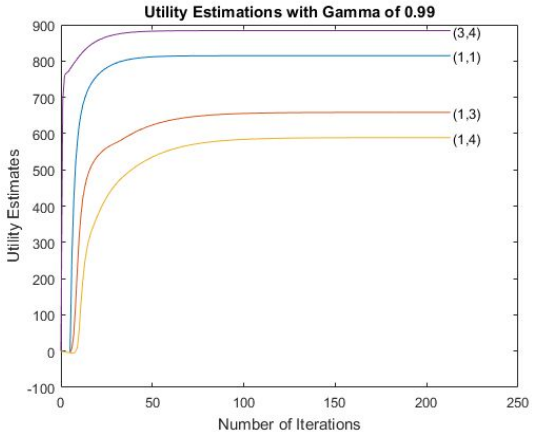
Table : Utilities for the given 4x4 board

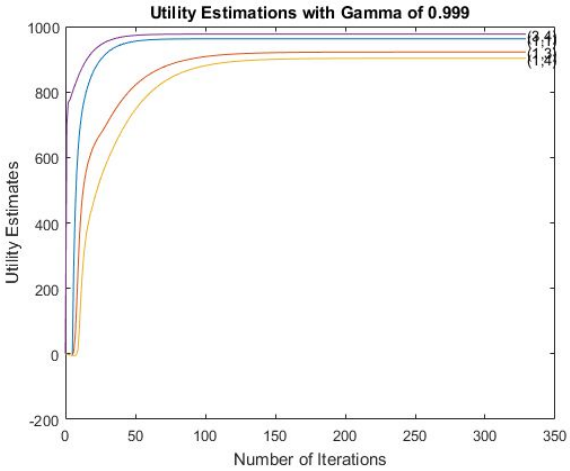
The above table shows the Utility estimates produced for the given 4x4 board.

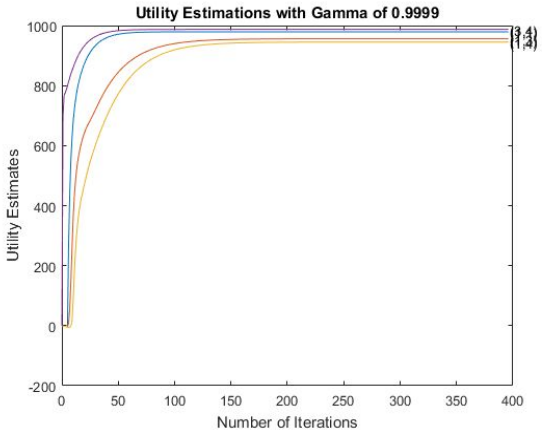
Finally, the next page shows 6 graphs of utility estimations with different gamma (discount factor) values of 0.9, 0.99, 0.999, 0.9999, 0.99999, and 0.999999.

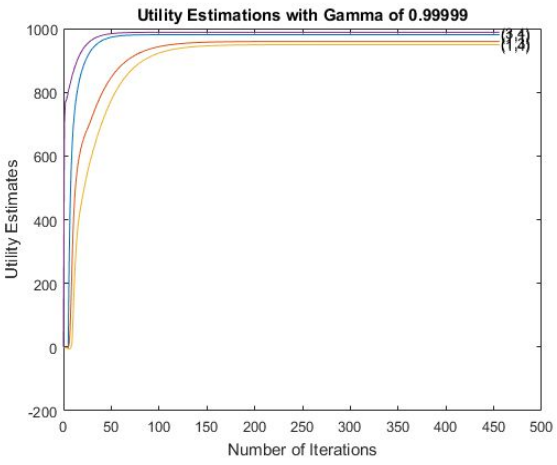
Below are the graphs for the 4x4 boards with varying γ values. As γ increases, the chosen lines start to converge.

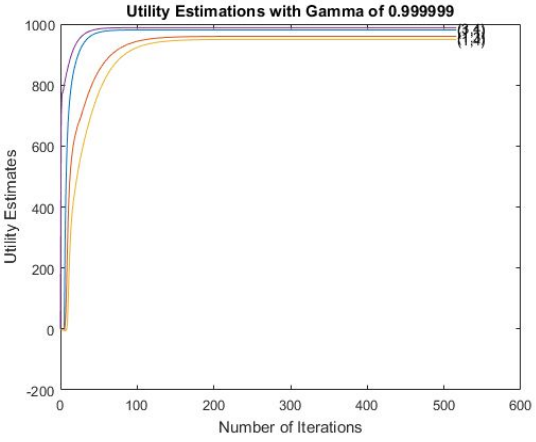

Utility Estimations with Gamma of 0.9
Utility Estimates
Number of Iterations
(3,4)
(1,1)
(1,3)
(1,4)


Utility Estimations with Gamma of 0.99
Utility Estimates
Number of Iterations
(3,4)
(1,1)
(1,3)
(1,4)


Utility Estimations with Gamma of 0.999
Utility Estimates
Number of Iterations


Utility Estimations with Gamma of 0.9999
Utility Estimates
Number of Iterations


Utility Estimations with Gamma of 0.99999
Utility Estimates
Number of Iterations


Utility Estimations with Gamma of 0.999999
Utility Estimates
Number of Iterations

5. INTERPRETATION: RAJUL

The 4 cases shown in the previous section show 4 major cases.

1) The first case shown in the table is the normal case where each step costs 1/1000 of the danger states. This shows that if the idea policy is taken, upon reaching right next to the Goal in the top right in state 11. Upon reaching here (3,4) the desired direction is up. This is due to the placement of the wumpus at (2,4). Therefore, the desired policy is to move away from it, where going to it has probability zero.
2) The second case is when every step is much worse than the lose-state. Every step costs -1700 and the lose state is -1000. Here, the better option would be to move towards the state with greater cost, i.e the lose state. If these policies were followed, it would lead straight to the pit.
3) This third next case, is creates a riskless behavior. It' seen as the safest path and a situation where the path total gets close to the lose state total.
4) The fourth case, is one where taking a step is rewarded with positive cost. This makes it such that it is ideal to keep walking and taking steps. No risk, more reward.

Along with the utilities, the graphs present in the previous section show various cases with varying Gamma values which represents the discount factor. This is clearly shown in the graphs. The discount factor describes the preference of an agent for current rewards over future rewards. When γ is close to 0, rewards in the distant future are viewed as insignificant.

When γ is 1, discounted rewards are exactly equivalent to additive rewards, so additive rewards are a special case of discounted rewards. As the γ value increases in the graphs, the utility increases to higher utility and starts to converge to show this behavior.

6. CRITIQUE: CONAN

We learned a great deal about the implementation of the value iteration process. To see Utilities change with the variation of variables like gamma and rewards was interesting. We now have a better understanding of the policy iteration process as well and how it changes.

It might be more interesting and better to observe this information over a bigger board or even a larger data set that could help with accuracy. The addition of an agent and insertion of this process into the Wumpus world will be interesting especially if it was combined with other traversal ideologies.

7. LOG: RAJUL & CONAN

Rajul: I spent 12 hours on implementation and 5 hours on the Lab Report.

Conan: I spent 10 hours on implementation and 6 hours on the Lab Report.