Eric Komperud

A7 – CS4300 Lab Report –11/21/17
Markov Decision Processes and Value Iteration

1. Introduction

It this assignment, I was implementing an algorithm for value iteration which is a somewhat natural way for an agent to determine how to traverse a world. Note that I did not implement an agent for the assignment, only an algorithm. The algorithm operates by taking in parameters that tell it how rewarding (or not) each state of the world is and how likely each state/action pair will land it in another state. Once it iterates over those, it will produce an array of state/utility pairs which show a general trend of which states are most optimal to be in. A policy generator algorithm can then further iterate and determine which action in each state will lead it to an optimal score based on the values that were returned by the value iteration algorithm. This result of state/action pairs is known as a "policy". The worlds I will be operating on are a Wumpus world board and a board taken from Russell and Norvig's *Artificial Intelligence: A Modern Approtach 2nd ed.*, but only one specific board of Wumpus World this time:

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

Fig 1. Wumpus Board

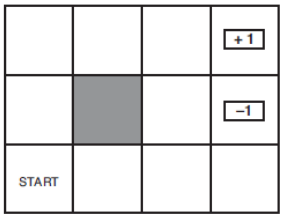

Fig 2. R&N board

One of the nuances of value iteration is that it accounts for the magnitude of various rewards. One question I intend to answer based on this fact is how having radically different living rewards affects generated policies. Another nuance is that value iteration can operate until two successive iterations yield the same exact utilities. This is unrealistic though and we will usually set a cutoff point. I want to see how the yielded utilities are affected by making the cutoff point harder and harder to reach.

2. Method

In my code, I first define all the necessary parameters to be used:

- S: A vector of states
- A: A vector of action
- P: A transition model. This defines how likely it is to get from one state to any other state given an action.
- R: A vector of state rewards.
- Gamma: A discount factor to multiply into all iterations of rewards
- Eta: A constant for helping to determine the termination cutoff point
- Max_iter: Given that the cutoff point isn't reached, the code will cutoff after this many iterations

I then ran my value iteration algorithm on each board according to the following pseudocode:

Function: Value_Iteration(S,A,P,R,Gamma,Eta,Max_iter) **returns** U

locals: U, U', vectors of utilities for states in S, initially zero

δ, the maximum change in the utility of any state in an iteration

repeat:

U ←U'; $\delta \leftarrow 0$

For each state s **in** S, **do**

U'[s] = R(s) + Gamma * (max of all actions Σ(P(s'|s,a)*U[s']))

If U'[s] – U[s] > δ **then** δ = U'[s] – U[s]

Until δ < (eta * (1 – Gamma)) / Gamma

Return U

(credit: Russell & Norvig, *Artificial Intelligence: A Modern Approach 2nd ed.* Pg. 653)

In essence, each time the algorithm repeats, it assigns a utility to each state based on the max utility achieved by picking an optimal action from the set of actions. Each successive iteration used the generated utilities from the last iteration. Once that function ran, I ran the policy generation algorithm on U:

For each state s **in** S, **do**

P[s] = The action that yields: (max of all actions Σ(P(s'|s,a)*U[s']))

Return P

This algorithm chooses an action for each state that will most likely yield it the maximum utility.

To gather data, I chose several different living reward values to used for the R&N board that would theoretically change the optimal policy. My first living reward value was -2 which should I hypothesized should yield suicidal policy results. My second living reward value was -0.04 which I hypothesized should yield an optimistic policy for attempting to reach the gold. My last living reward value was +2 which I hypothesized should yield a policy apathetic to reaching the gold.

I also ran the algorithm on a living reward of -0.04 with several different values of gamma to see how changing the cutoff point affected the policies (gamma = 0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999).

Finally, using the data I gathered from the R&N board, I ran the algorithm with what I thought were optimal parameters to generate a policy for the Wumpus board.

3. Verification of Program

I verified my program by running it with the same parameters of a problem shown in Russell & Norvig's book that I also sourced earlier. The problem is on page 651.

Eric Komperud

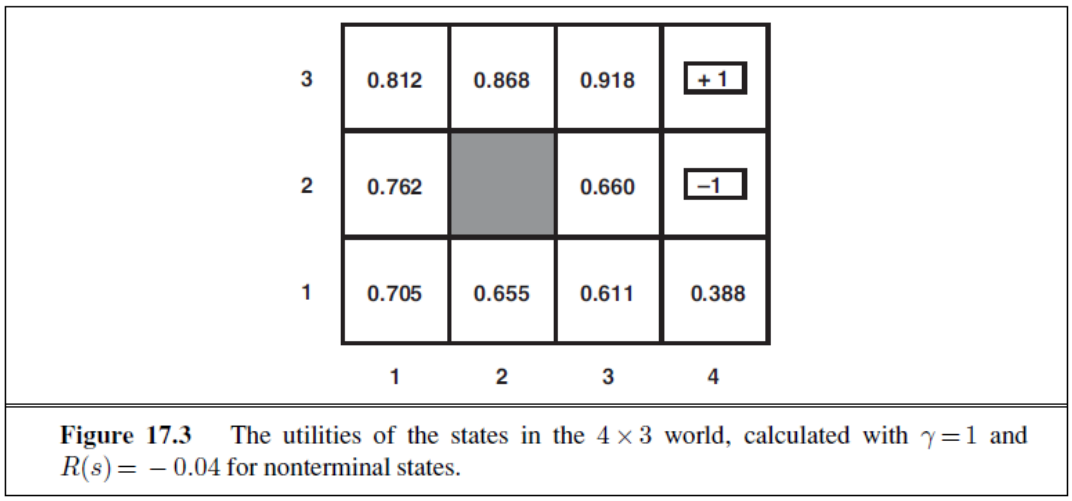

Figure 17.3 The utilities of the states in the 4×3 world, calculated with $\gamma = 1$ and $R(s) = -0.04$ for nonterminal states.

Fig 3.

The eta used in this problem is 0.1. The transition model dictates that given a movement action (North, West, South, East), an agent would have a 0.8 chance of successfully performing moving that direction, a 0.1 chance of moving in clockwise direction, and a 0.1 chance of moving in the counter-clockwise direction. If the agent bumps into a wall or the block in (2,2), it would not change states. An agent in the terminal squares of (4,3) or (4,2) would always take the climb action. Given these parameters, my value iteration algorithm should yield similar results:

0.811558	0.867808	0.917808	1
0.761558	-0.04	0.660274	-1
0.705308	0.655308	0.611416	0.387925

Fig 4.

The results from my program indicated that it is decidedly close to the algorithm used in the book.

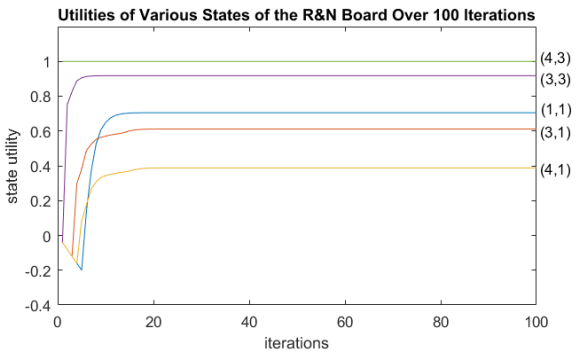

Fig 5.

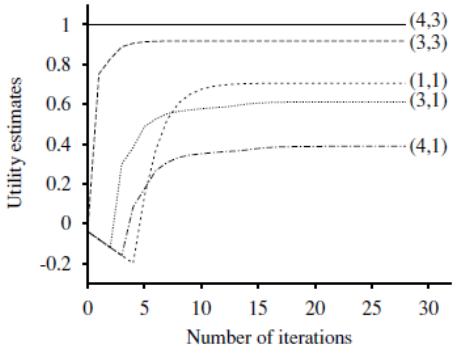

Fig 6. (Russel & Norvig, pg 653)

In addition, the utility values of the various states over time seem to match those found in the R&N book. Thus, I conclude that my program is correct for the problem at hand.

4. Data

My policies for the living reward values of -2, -0.4, and 2 are shown below. Note that 1's on the unreachable/terminal states are only placeholders.

4 ->	4 ->	4 ->	1
1 ^	1	4 ->	1
4 ->	4 ->	4 ->	1 ^

Fig 7. Living reward of -2

4 ->	4 ->	4 ->	1
1 ^	1	1 ^	1
1 ^	2 <-	2 <-	2 <-

Fig 8. Living reward of -0.04

3 v	1 ^	2 <-	1
1 ^	1	2 <-	1
1 ^	1 ^	2 <-	3 v

Fig 9. Living reward of +2

The utilities yielded from my various gamma levels are shown below. Note that the unreachable square has the living reward value for its utility as a placeholder.

0.507518	0.649447	0.79533	1
0.392833	-0.04	0.48635	-1
0.283197	0.248752	0.34284	0.125976

Fig 10. Gamma = 0.9

0.776167	0.843935	0.905096	1
0.716572	-0.04	0.641327	-1
0.650229	0.591491	0.559707	0.337358

Fig 11. Gamma = 0.99

0.807963	0.865399	0.916532	1
0.756965	-0.04	0.658363	-1
0.69967	0.648784	0.604641	0.381339

Fig 12. Gamma = 0.999

0.811198	0.867567	0.917681	1
0.761098	-0.04	0.660083	-1
0.704743	0.654653	0.610735	0.387261

Fig 13. Gamma = 0.9999

0.811522	0.867784	0.917795	1
0.761512	-0.04	0.660255	-1
0.705252	0.655243	0.611348	0.387859

Fig 14. Gamma = 0.99999

0.811555	0.867806	0.917807	1
0.761554	-0.04	0.660272	-1
0.705303	0.655302	0.611409	0.387918

Fig 15. Gamma = 0.999999

The policy generated for my Wumpus board is shown below. Note that the

4 ->	4 ->	1 ^	G
1 ^	2 <-	P	4 ->
1 ^	2 <-	W	4 ->
1 ^	2 <-	P	4 ->

Fig 16. Wumpus policy

5. Analysis

It appears that the various living rewards that I defined did in fact dramatically change the generated policies. In the 1st, the living reward was so harsh that the policy determined that reaching any terminal state was of utmost importance, no matter the reward on that terminal state. In the 2nd, the living reward was at a point where the policy had plenty of motivation to try to reach the +1 state while still avoiding the -1 state. In the 3rd, the living reward was so nice that reaching the terminal rewards, even the "good" state, was considered a bad thing.

As for the various gamma levels, the difference between the 0.9 level and the 0.999999 level was subtle but noticeable. Both showed a trend of increasing the closer to the +1 state they were, but 0.9 had its 2nd lowest utility in (2,1) whereas 0.999999 had its 2nd lowest utility in (3,1). While this would probably not make an agent following policies based on these utilities act significantly differently, it seems that there is a point where one would want to set gamma to keep it as close as possible to gamma = 1

The policy generated for the Wumpus board makes a lot of sense when one considers that the reward for landing on a Pit of Wumpus state is -1000. The algorithm considers this large negative reward and generates a policy that will never allow an agent to fall land in those states, but still have a good chance of getting the gold after some time. Note taking action North in state (3,4) will land an agent in the (2,4) for a -1 reward 10% of the time, will land an agent in the (3,4) for a -1 reward 80% of the time, and will land an agent in (4,4) for a +1000 reward 10% of the time. Taking action East in state (3,4) will land an agent in (3,4) for a -1 reward 10% of the time, will land an agent in (4,4) for a +1000 reward 80% of the time, and will land an agent in (3,3) for a -1000 reward 10% of the time.

6. Interpretation

It seems to me that AI imitates real life. In the case of the varying living rewards, policies will change depending on how bad or good it is to achieve a set goal in comparison to the rewards for remaining

neutral. If an agent were to follow these policies, we would likely want small negative living rewards that pushed it towards achieving the goal.

As mentioned earlier, there appears to be a certain point in varying gamma levels that the gamma matches the result of gamma = 1. While one might think gamma = 0.9 is reasonably high, it was not enough to match the utility patterns generated at 0.999999.

7. Critique

Doing this assignment gave me not only helped me understand value iteration a lot better, but also offered a bit of insight into the human mind as well. In life, we have many things which we consider benchmark achievements in an adult life such as getting a well-paying job, graduating college, having a family, etc. These achievements all have different values for different people though. In addition, different people's life situations give them different living rewards such that they may not be motivated to do anything with their life at all, or that they may be motivated to simply reach any state that would offer a change to their current living reward.

Further research on varying levels of gamma might include doing testing on many different boards to determine the average gamma level that matches the utility pattern of gamma = 1 for most Wumpus World problems.

8. Log

I spent about 4 hours studying various videos online to get a solid understanding of the concepts I was looking at. I spent about 8 hours writing my code and debugging it to a working state. I spent about 4 hours gathering data and writing this lab report.