Randomized Optimization, Unsupervised Learning, and Dimensionality Reduction

Writeup for Assignment 02 - CS 6741

Magahet Mendiola

ABSTRACT

An empirical analysis of planning and reinforcement learning algorithms in the context of marchov decision processes.

1. PLANNING METRICS

A number of simple metrics were used to evaluate the performance of our planning algorithms. These include the number of iterations and elapsed time required to find the optimal utility values for each state. However, the goal of planning is to find the optimal policy for an MDP, and not the optimal set of utility values. Therefore, we can also compare how long it takes each to find the optimal policy for each state.

There is also the rate at which a policy converges toward the optimum. This can be seen with the hamming distance between the current policy and the optimal policy at each iteration. If a partial plan can be created quickly, it may be good enough for practical use on the MDP.

Another derivation of the hamming distance metric is the time/iterations required to find the policy which will successfully guide a deterministic agent to the goal state. The idea behind this metric is that there may be many states that are never visited. Running value or policy iteration until enough of the states have optimal policies to guide the agent could greatly reduce planning time.

Creating a planning algorithm with this stopping criteria would be similar to q-learning in regards to limiting exploration. It would also be similar to a* path finding, with search areas radiating omni-directionally from non-zero reward states.

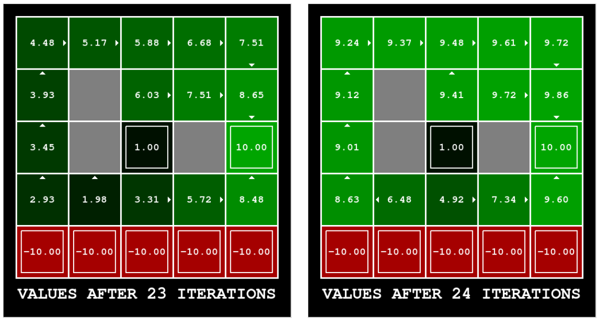
2. GRID WORLDS

Each algorithm was tested against contrived gridworld type MDPs. These worlds were setup to accentuate the strengths and weaknesses of each algorithm. They also illustrate the impact of rewards, discounts, and and to point out opportunities for modifications to each.

2.1 Discount Grid

This grid world, titled Discount Grid [1], is useful for illustrating the effects of utility discounting, transition function probabilities, and state rewards on optimal policies. It includes two terminal states with positive rewards (+1 closer, +10 farther) and five negative terminal states at the bottom edge of the grid. This setup allows us to explore what set of problem and solution parameters affect the goals defined by an optimal policy.

Figure 1a shows the resulting values of the MDP after running value iteration with a γ of 0.9 and a reward of 0 in all the non-terminal states. The transition probabilities are 0.8 for desired action and a combined 0.2 for all orthogonal movements. In this case, the policy directs the agent to seek out the farther, higher valued, terminal state. It also avoids the high penalty cliff except when the alternative is moving into the lower reward terminal state.



(a) 0.9

(b) 0.99

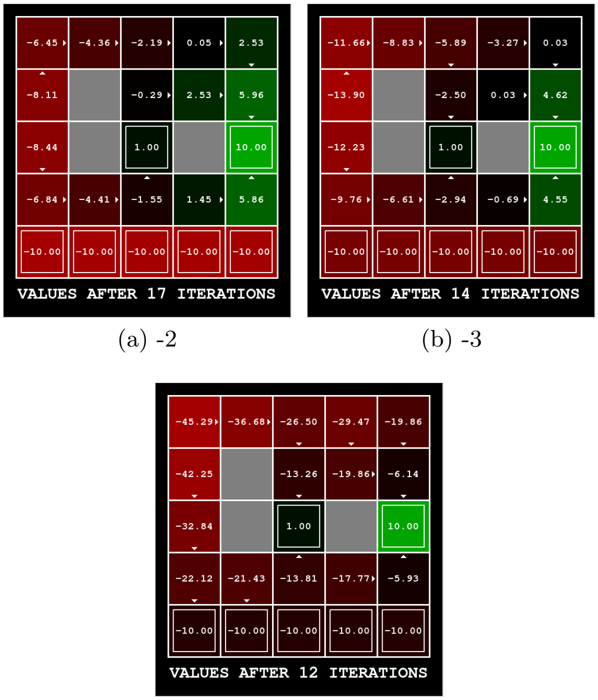
Figure 1: discount grid - various γ settings

Time matters.

With a γ of 0.99, which reduces the effects of discounting, the policy shifts slightly (Figure 1b). The policy in state (2,2) now risks a direct movement to the east, given the higher expected utility of that action. This is due to the higher relative value of state (1,2) due to the reduced discounting. Also, state (3,6) avoids the risk of falling into the lower reward for the same reason. The expected value of attempting a northward movement is now relatively more valuable than with a move east policy, despite the indirect route to the terminal state. This shows the implications of adjusting delayed reward and how an agent could be lead to take greater risks or follow an otherwise sub-optimal route to gain a larger reward farther in the distance.

Rewards matter.

With a shift in rewards, we can affect how the resulting policy directs the agent. Figure 2a shows the result of changing the reward in all non-terminal states to -2. The policy now believes that the closer terminal state is a better alternative at state (3,2). The negative rewards, and risk of falling off the cliff, push the agent toward the closer end point. With the non-terminal rewards set to -3, this early termination policy extends to the northern path as well (Figure 2b). With a sufficiently negative non-terminal reward, we can even entice the agent to end the process as quickly as possible, by intentionally jumping off the cliff (Figure 2c).



-6.45	-4.36	-2.19	0.05	2.53
-8.11		-0.29	2.53	5.96
-8.44		1.00		10.00
-6.84	-4.41	-1.55	1.45	5.86
-10.00	-10.00	-10.00	-10.00	-10.00

VALUES AFTER 17 ITERATIONS

(a) -2

-11.66	-8.83	-5.89	-3.27	0.03
-13.90		-2.50	0.03	4.62
-12.23		1.00		10.00
-9.76	-6.61	-2.94	-0.69	4.55
-10.00	-10.00	-10.00	-10.00	-10.00

VALUES AFTER 14 ITERATIONS

(b) -3

-45.29	-36.68	-26.50	-29.47	-19.86
-42.25		-13.26	-19.86	-6.14
-32.84		1.00		10.00
-22.12	-21.43	-13.81	-17.77	-5.93
-10.00	-10.00	-10.00	-10.00	-10.00

VALUES AFTER 12 ITERATIONS

(c) -11

Figure 2: discount grid - various non-terminal reward settings

The problem matters.

Policies are not only affected by planning algorithm parameters, but also by the problem definition, such as the probabilities in the transition function. If movement in the MDP is deterministic, or the intentional action probabilities are sufficiently high, otherwise risky paths would be considered optimal. Figure 3a shows a policy of cliff-walking given only a 1% change of falling off the ledge. Conversely, the policy in Figure 3b shows how our planning algorithm would approach a high risk environment (inebriated agent). In the later case, the agent will avoid any east-west movement, even at the expense of exiting at the low reward terminal state.

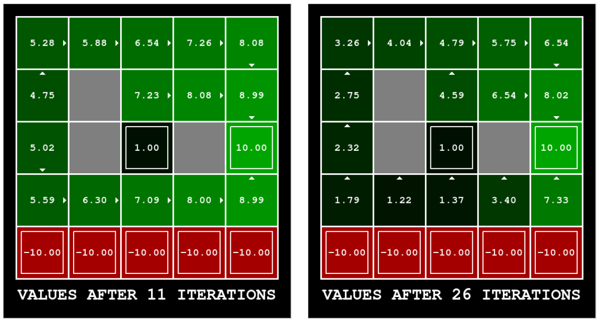
2.2 Tunnel Grid

Tunnel grid is a one dimensional grid with the starting state on one end and the only terminal state on the other. Although this grid is simplistic, it

	Elapsed Time	Iterations/Episodes
Value	0.04	29
Policy	0.04	19
Q-Learning	0.390	18

Table 1: tunnel grid - performance

3. VALUE ITERATION



5.28	5.88	6.54	7.26	8.08
4.75		7.23	8.08	8.99
5.02		1.00		10.00
5.59	6.30	7.09	8.00	8.99
-10.00	-10.00	-10.00	-10.00	-10.00

VALUES AFTER 11 ITERATIONS

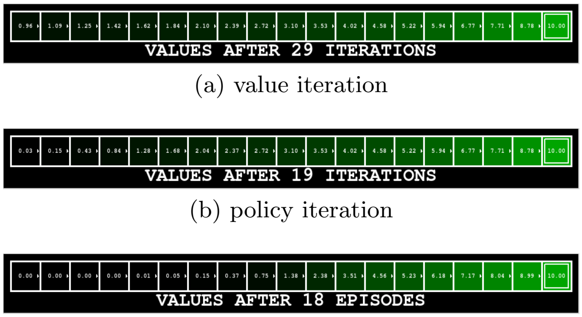
(a) 0.01

3.26	4.04	4.79	5.75	6.54
2.75		4.59	6.54	8.02
2.32		1.00		10.00
1.79	1.22	1.37	3.40	7.33
-10.00	-10.00	-10.00	-10.00	-10.00

VALUES AFTER 26 ITERATIONS

(b) 0.4

Figure 3: discount grid - various unintentional movement probabilities



0.96	1.09	1.25	1.42	1.62	1.84	2.10	2.39	2.72	3.10	3.53	4.02	4.58	5.22	5.94	6.77	7.71	8.78	10.00

VALUES AFTER 29 ITERATIONS

(a) value iteration

0.03	0.15	0.43	0.84	1.28	1.68	2.04	2.37	2.72	3.10	3.53	4.02	4.58	5.22	5.94	6.77	7.71	8.78	10.00

VALUES AFTER 19 ITERATIONS

(b) policy iteration

0.00	0.00	0.00	0.00	0.01	0.05	0.15	0.37	0.75	1.38	2.38	3.51	4.56	5.23	6.18	7.17	8.04	8.99	10.00

VALUES AFTER 18 EPISODES

(c) q-learning

Figure 4: tunnel grid

4. POLICY ITERATION

5. Q-LEARNING

metrics elapsed time hamming distance

lazy vs. eager learning.

planning performance

policy iteration implementation issues get's stuck with only policy change check takes too long with epsilon check works well with value update check

q-learning performnace q-learning strategies

6. REFERENCES

[1] D. Klein. *Project 3: Reinforcement Learning*, 2009 (accessed April, 2014). http://inst.eecs.berkeley.edu/ cs188/fa09/ projects/reinforcement/reinforcement.html.

[2] S. J. Russell, P. Norvig, J. F. Candy, J. M. Malik, and D. D. Edwards. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.