A5

Monte Carlo Problem

Trung Le and Johnny Le

10/28/2016

1. Introduction - Trung Le

The assignment studies the performance of an agent which uses the Monte Carlo approach to solve for likelihoods of pits and wumpi in each cell. The agent then uses this information to deduce the safest possible cell to move to next. If there are no safe cells to go to the agent attempts to create a safe square by shooting and attempting to kill the wumpus. If after this action there are still no safe cells the agent takes a risk and finds a path to the safest cells i.e. the cell with lowest probability of both a wumpus and a pit. The agent does not give up and will die trying to retrieve the gold.

Questions:

1. On average, do larger MC samples provide better scores?
2. Do larger MC samples provide more likelihood of success?
3. Is there an improvement in successes when compared to the hybrid agent?

2. Method - Johnny Le

The A5 Monte Carlo experiment was conducted with the CS4300_A5_driver. Here, a majority of the experimental statistics are gathered such as the mean, variance, and confidence interval when iterating through the various sample levels and clearing each function prior to experimentation.

CS4300_WW1, the function for the Wumpus World, took in the new CS4300_MC_agent for its logical deduction. Utilizing a number of helper methods, the core functionality of the Monte Carlo algorithm is completed here with an initialization of the agent subsequent conditional statements for certain percepts.

Every time the frontier is accessed, CS4300_WP_estimates was run to provide an up to date analysis of the board without accruing too much runtime. In addition, if the wumpus is slain, the estimates are re-analyzed as well.

CS4300_WP_estimates will create a random board which is checked for validity and then added to the number of pits or wumpi in each room after the requisite number of valid boards have been reached. Each room total is then divided by the number of valid boards to calculate the percentage of each room having a pit or wumpus for the given conditions.

The CS4300_check_board method checks for validity by iterating through all breezes and stenches to ensure there was or isn't a pit or wumpus as a neighbor for each spot. In addition, it verifies that for the room with a percept taken, there was no pit or wumpus for that location.

Returning to CS4300_MC_agent, the agent will then decide amongst the frontier to go to a safe location; however, if a stench has been found and no plans have been made with the safe frontier, the agent used the CS4300_plan_shot function. This function analyzes the most likely location of the wumpus and will navigate to the nearest square in which the corresponding room can be shot.

Upon needing to take a risky decision, the agent decided upon a room by taking the mean of the chance of a pit and the chance of a wumpus for each possible option. Then the lowest option was selected.

This process continued until the agent either encountered a pit/wumpus or found the gold and returned.

3. Verification - Trung Le

The verification process will be done by stepping through the agent's actions and behavior for the following board:

0	0	0	3
2	0	0	0
0	1	1	0
0	0	0	0

To save space, since the agent will not encounter a stench during, the stenches board will always be the priors i.e. a board with roughly 1/15 chance of Wumpus except where the agent has visited, in which case it that cell will be replaced with a 0.

Initial iteration

<table>
<tr><th>Board</th><th>Breezes</th><th>Frontier</th><th>Pits</th></tr>
<tr>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>></td><td>0</td><td>0</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>0</td><td>-1</td><td>-1</td><td>-1</td></tr>
</table></td>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>0.19</td><td>0.20</td><td>0.17</td><td>0.22</td></tr>
<tr><td>0.18</td><td>0.20</td><td>0.21</td><td>0.19</td></tr>
<tr><td>0</td><td>0.19</td><td>0.23</td><td>0.22</td></tr>
<tr><td>0</td><td>0</td><td>0.21</td><td>0.19</td></tr>
</table></td>
</tr>
</table>

The agent is able to deduce that (2,1) and (1,2) are both safe because it does not detect a breeze or stench at (1,1). It takes both (2,1) and (1,2) as candidate rows and arbitrarily picks to navigate to (1,2).
Its plan is to turn left, then go forward.

Now it needs to make another decision.

<table>
<tr><th>Board</th><th>Breezes</th><th>Frontier</th><th>Pits</th></tr>
<tr>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>^</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>0</td><td>-1</td><td>-1</td><td>-1</td></tr>
</table></td>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>0.19</td><td>0.20</td><td>0.17</td><td>0.22</td></tr>
<tr><td>0.51</td><td>0.20</td><td>0.21</td><td>0.19</td></tr>
<tr><td>0</td><td>0.52</td><td>0.23</td><td>0.22</td></tr>
<tr><td>0</td><td>0</td><td>0.21</td><td>0.19</td></tr>
</table></td>
</tr>
</table>

The agent detected a breeze at (1,2) and notes that (1,3) and (2,2) have 50% probability of a pit. It backtracks, and decides to navigate to the other safe cell: (2,1).

Its plan is to turn left, turn left, go forward, turn left, go forward.

<table>
<tr><th>Board</th><th>Breezes</th><th>Frontier</th><th>Pits</th></tr>
<tr>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>></td><td>0</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>0</td><td>1</td><td>-1</td><td>-1</td></tr>
</table></td>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>0.19</td><td>0.20</td><td>0.17</td><td>0.22</td></tr>
<tr><td>0.34</td><td>0.20</td><td>0.21</td><td>0.19</td></tr>
<tr><td>0</td><td>0.86</td><td>0.23</td><td>0.22</td></tr>
<tr><td>0</td><td>0</td><td>0.31</td><td>0.19</td></tr>
</table></td>
</tr>
</table>

It detects another breeze at (2,1) and deduces that there is high probability of a pit in (2,2). Since there are no more safe cells, the agent takes a risk. It calculate the average probability of danger in the three frontier cells: (1,3), (2,2) and (3,1) and finds that (3,1) has the lowest probability of danger.
Its plan is to navigate to (3,1) by going forward.

<table>
<tr><th>Board</th><th>Breezes</th><th>Frontier</th><th>Pits</th></tr>
<tr>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>></td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>1</td><td>-1</td><td>-1</td><td>-1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>-1</td></tr>
</table></td>
<td><table>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table></td>
<td><table>
<tr><td>0.19</td><td>0.20</td><td>0.17</td><td>0.22</td></tr>
<tr><td>0.27</td><td>0.19</td><td>0.21</td><td>0.19</td></tr>
<tr><td>0</td><td>1</td><td>0.58</td><td>0.22</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0.5</td></tr>
</table></td>
</tr>
</table>

<table><tr><td></td><td></td><td></td><td><table><tr><td></td><td></td><td></td><td>3</td></tr></table></td></tr></table>

Again, the agent detects a breeze and there are no safe squares. It infers that there is definitely a pit at (2,2) and high probability of a pit in (3,2) and (4,1). It takes a risk and decides that the safest cell is (1,3). It decides to navigate to (1,3).
Its plan is to turn left, turn left, go forward, go forward, turn right, go forward, go forward to (1,3).

At (1,3) it perceives a glitter indicating that the gold is in the square. It then grabs and calls A* to track back to (1,1).
Its plan is to grab, turn left, turn left, go forward, go forward then climb out.

The full trace is:

Action	Score
3	-1
1	-2
3	-3
3	-4
1	-5
3	-6
1	-7
1	-8
3	-9
3	-10
1	-11
1	-12
2	-13
1	-14

1	-15
4	-16
3	-17
3	-18
1	-19
1	-20
6	-21
SUCCESS	979

4. Data and Analysis - Johnny Le

MC samples	Mean Score	Number Successes	Number Failures	Variance	Confidence Interval
No MC	184.88	152	98	9.5704e05	[63.61, 306.15]
50	268.95	163	87	9.1142e05	[150.60, 387.29]
100	284.13	165	85	8.9433e05	[166.90, 401.36]
250	355.64	174	76	8.3955e05	[242.06, 469.22]

5. Interpretation - Trung Le

According to the data, there seems to be a direct positive correlation between the mean score, number of successes and MC samples. This means that for larger MC samples, the agent has a better chance for success than lower MC samples. The agent is also able to reach the gold in fewer actions as it has more data to compute safer cells. The larger the data, the more reliable it can deduce that a cell has low probability of a pit or wumpus thus providing a lower probability of walking into a pit.

The variance is large because the data points are more spread out from the mean. This is due to the scoring system in place, where a good and short successful run would be score around the upper 900s and deaths would range in the upper -1000s. However there is an inverse relationship between the number of MC samples and the variance. Meaning that for larger MC samples, the data points range closer to the mean. A smaller variance would also provide a more tighter and more accurate confidence interval. We used a 95% confidence interval and the bounds of the interval become tighter for larger MC samples.

If the agent is going to take a risk, it chooses the safest cell based on the lowest average probability of danger. We computed this by adding the probability of a pit and the probability of a wumpus and average the result. The performance of the agent is highly dependent on it choosing a safer cell to navigate to and this is dependent on the number of MC samples.
The MC agent is an improvement on the hybrid agent. The hybrid agent is able to use logic to prove whether a cell is safe or is unable to prove it. However, the MC agent uses probability to deduce the most dangerous cell and attempts to avoid it at all costs.

For the board

0	0	0	0
3	1	0	0
0	0	1	0
0	0	2	0

Our hybrid agent fails and dies whereas the MC agent succeeds and achieves a score of 982.

There are scenarios when the MC agent and hybrid agent perform similarly, such as the following board

0	0	0	0
0	0	0	0
3	0	0	0
0	0	2	0

The MC agent can only deduce that there is 50% probability of a wumpus in (2,1) and (1,2) and makes a guess on where to shoot. The hybrid agent cannot prove that (2,1) and (1,2) are safe and has detected a stench so also attempts to make a guess.

6. Critique - Johnny Le

The Monte Carlo algorithm was a more effective method for an agent to determine the best path towards locating the gold. Unfortunately, there were many failures enroute as in some situations, the probabilities calculated were not enough to determine the safest path. To supplement Monte Carlo, another method of probabilistic deduction should be applied such as the most likely location of the gold. There is a chance that it could be beneficial which would make it an interesting sequence to run.

We also modified the the MC agent to only attempt to make a safe square via shooting an arrow if it has detected a stench. This is because, if the agent had no safe square, it would always shoot an arrow blindly in attempt to make a safe square. We feel that this is wasteful and only have the agent shoot if it has detected a stench. Our plan shot method navigates the agent to the cell where it detected the stench and then shoots the wumpus.

Another drawback of Monte Carlo is the time efficiency of the deductions as large sample sizes are required for more accurate probabilities. A possible method that could remedy that is to use some form of memoization to reduce the time complexity of the algorithm and avoid recalculating known areas.

7. Log

Monte Carlo Implementation:

- Trung Le: 15 hour
- Johnny Le: 15 hour

Analysis and Lab Report Write Up:

- Trung Le: 3 hours
- Johnny Le: 3 Hours