Mischievous Sibling's Grid World

Aykut C. Satici[a]

[a]*Boise State University, Mechanical and Biomedical Engineering, Boise, 83725, Idaho, USA*

Abstract

This is a technical report solving an interesting question posed by Gökhan Atınç, a good friend of mine. The question starts out exactly like the well-known Grid World problem, but it has a twist. We solve the problem in two different ways: first using a direct probability calculation and then using reinforcement learning on a judiciously designed Markov Decision Process. The results are compared and discussed.

Keywords: Probability, Expectations, Reinforcement learning, Gymnasium, Policy iteration

1. Introduction

The question is interesting and the solutions are just as interesting. Let us delve right into it.

2. Problem Statement

Alice has a 5×5 chess board as shown in Figure 1, where each square is described by its coordinates $(i, j) \in \{0, 1, 2, 3\}^2$. Her remote-controlled robot starts on the square $(2, 2)$. The remote-controller has 4 buttons that can move the robot east, north, west, and south by 1 square with each press. Bob, her mischievous sibling, has tampered with the wiring of these buttons. Assuming that the buttons are fixed after having shuffled once, what is the expected value of the minimum number of times Alice needs to press the buttons in order to move the robot to the goal square $(4, 4)$?

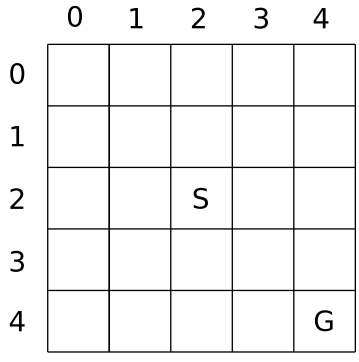

Figure 1: Schematic of the problem.

For future reference, if we linearly index the squares of the chessboard from 0 to 24 in a row-major fashion, then the mapping from the linear index to the (i, j) coordinates and back is given by the following bijections:

$$5j + i = k \leftrightarrow \left(i = k - 5\left\lfloor \frac{k}{5} \right\rfloor, j = \left\lfloor \frac{k}{5} \right\rfloor\right) \tag{1}$$

3. Solutions

We present two solutions to the problem: a probabilistic solution and a reinforcement learning (RL) solution.

3.1. Probability Solution

We invoke a celebrated theorem from probability theory that will help us solve the problem. This theorem is called the *law of total expectation* or *tower rule of probability theory* in the literature [1]. In this section, we use the correspondence $\{1, 2, 3, 4, 5\} \mapsto \{0, 1, 2, 3, 4\}$ and $\{A, B, C, D, E\} \mapsto \{0, 1, 2, 3, 4\}$ for legacy reasons.

Theorem 1 (Tower rule). *Let X be a random variable and $\{A_i\}_i^m$ be a finite partition of the sample space. Then,*

$$\mathbb{E}[X] = \sum_i^m \mathbb{E}\left[X \mid A_i\right] \mathbb{P}(A_i).$$

We denote the square on which the robot resides at the k^{th} step by s_k and the goal square by g. Consider the following family of random variables.

$$C_n := \sum_{k=n+1}^{\infty} r_k, \qquad r_k = \begin{cases} 1 & \text{if } s_k \neq g \\ 0 & \text{if } s_k = g \end{cases}. \tag{2}$$

The sum in this definition is well-defined because once the robot reaches g, all the remaining r_k's take the value zero. We are being asked the value of $\mathbb{E}[C_0]$. To that end, we will use the following repercussion of definition (2).

$$C_m = \sum_{i=m+1}^{n} r_i + C_n, \qquad m \leq n.$$

Using the properties of expectation, if $s_i \neq g$ for $m < i < n$, we can deduce the following result.

$$\mathbb{E}[C_m] = \sum_{i=m+1}^{n} \mathbb{E}[r_i] + \mathbb{E}[C_n] = n - m + \mathbb{E}[C_n].$$

Using the properties of expectation, if $s_i \neq g$ for $m < i < n$, we can deduce the following result.

$$\mathbb{E}[C_m] = \sum_{i=m+1}^{n} \mathbb{E}[r_i] + \mathbb{E}[C_n] = n - m + \mathbb{E}[C_n].$$

Similar identities to the above expression hold for conditional expectations as well. Using the tower rule 1, we start to compute.

$$\begin{aligned}\mathbb{E}[C_0] = \mathbb{E}\left[C_0 \mid s_1 \in \{C4, D3\}\right] \underbrace{\mathbb{P}(s_1 \in \{C4, D3\})}_{=\frac{1}{2}} \\ + \mathbb{E}\left[C_0 \mid s_1 \in \{B3, C2\}\right] \underbrace{\mathbb{P}(s_1 \in \{B3, C2\})}_{=\frac{1}{2}}.\end{aligned} \tag{3}$$

From now on, we will assume $s_1 = C4$ in the first term on the right-hand side and $s_1 = C2$ in the second. Notice that, because of the inherent symmetry of the problem, while the state $s_1 = D3$ is equivalent to $s_1 = C4$; $s_1 = B3$ is equivalent to $s_1 = C2$.

Computation of the first conditional expectation in (3). First of all, we consider the first expectation term on the right-hand side of equation (3).

$$\begin{aligned}\mathbb{E}\left[C_0 \mid s_1 = C4\right] = \overbrace{r_1 + r_2}^{=2} \\ + \overbrace{\mathbb{E}[C_2 \mid s_1 = C4, s_3 = D5]}^{=2} \overbrace{\mathbb{P}(s_3 = D5 \mid s_1 = C4)}^{=\frac{1}{3}} \\ + \mathbb{E}[C_2 \mid s_1 = s_3 = C4] \underbrace{\mathbb{P}(s_3 = C4 \mid s_1 = C4)}_{=\frac{1}{3}} \\ + \mathbb{E}[C_2 \mid s_1 = C4, s_3 = B5] \underbrace{\mathbb{P}(s_3 = B5 \mid s_1 = C4)}_{=\frac{1}{3}}.\end{aligned} \tag{4}$$

We delve further into the computation of the two expected values in equation (4), whose values are not immediately apparent.

$$\begin{aligned}\mathbb{E}[C_2 \mid s_1 = s_3 = C4] = \underbrace{r_3 + r_4}_{=2} \\ + \underbrace{\mathbb{E}[C_4 \mid s_1 = s_3 = C4, s_5 = D5]}_{=2} \underbrace{\mathbb{P}(s_5 = D5 \mid s_1 = s_3 = C4)}_{=\frac{1}{2}} \\ + \underbrace{\mathbb{E}[C_4 \mid s_1 = s_3 = C4, s_5 = B5]}_{=4} \underbrace{\mathbb{P}(s_5 = B5 \mid s_1 = s_3 = C4)}_{=\frac{1}{2}}.\end{aligned}$$

There is only one expected value that we have left to compute in equation (4):

$$\begin{aligned}\mathbb{E}[C_2 \mid\ s_1 = C4, s_3 = B5] = \\ \underbrace{\mathbb{E}[C_2 \mid s_1 = C4, s_3 = B5, s_4 = C5]}_{=4} \\ \times \underbrace{\mathbb{P}(s_4 = C5 \mid s_1 = C4, s_3 = B5)}_{=\frac{1}{2}} \\ + \underbrace{\mathbb{E}[C_2 \mid s_1 = C4, s_3 = B5, s_4 = B4]}_{=6} \\ \times \underbrace{\mathbb{P}(s_4 = B4 \mid s_1 = C4, s_3 = B5)}_{=\frac{1}{2}}.\end{aligned}$$

The computations above allows us to determine the first expected value in equation (3) using equation (4): $\boxed{\mathbb{E}[C_0 \mid s_1 = C4] = 6}$.

Computation of the second conditional expectation in (3). We use similar techniques to compute the second expected value on the right-hand side of equation (3).

$$\begin{aligned}\mathbb{E}\left[C_0 \mid s_1 = C2\right] = \\ \underbrace{\mathbb{E}[C_0 \mid s_1 = C2, s_2 = B2]}_{=8} \underbrace{\mathbb{P}(s_2 = B2 \mid s_1 = C2)}_{=\frac{1}{3}} \\ + \mathbb{E}[C_0 \mid s_1 = C2, s_2 = C3] \underbrace{\mathbb{P}(s_2 = C3 \mid s_1 = C2)}_{=\frac{1}{3}} \\ + \mathbb{E}[C_0 \mid s_1 = C2, s_2 = D2] \underbrace{\mathbb{P}(s_2 = D2 \mid s_1 = C2)}_{=\frac{1}{3}}.\end{aligned} \tag{5}$$

Once more, we utilize the tower rule to compute the expected values in equation (5) whose values are not immediately apparent.

$$\begin{aligned}\mathbb{E}[C_0 \mid s_1 = C2, s_2 = C3] = \overbrace{r_1 + r_2 + r_3 + r_4}^{=4} \\ + \underbrace{\mathbb{E}[C_4 \mid s_1 = C2, s_2 = C3, s_5 = D5]}_{=2} \\ \times \underbrace{\mathbb{P}(s_5 = D5 \mid s_1 = C2, s_2 = C3)}_{=\frac{1}{2}} \\ + \underbrace{\mathbb{E}[C_4 \mid s_1 = C2, s_2 = C3, s_5 = B5]}_{=4} \\ \times \underbrace{\mathbb{P}(s_5 = B5 \mid s_1 = C2, s_2 = C3)}_{=\frac{1}{2}}.\end{aligned}$$

There is only one expected value that we have left to compute in equation (5):

$$\begin{aligned}\mathbb{E}[C_0 \mid s_1 = C2, s_2 = D2] = \overbrace{r_1 + r_2 + r_3}^{=3} \\ + \underbrace{\mathbb{E}[C_3 \mid s_1 = C2, s_2 = D2, s_4 = E3]}_{=3} \\ \times \underbrace{\mathbb{P}(s_4 = E3 \mid s_1 = C2, s_2 = D2)}_{=\frac{1}{2}} \\ + \underbrace{\mathbb{E}[C_3 \mid s_1 = C2, s_2 = s_4 = D2]}_{=5} \\ \times \underbrace{\mathbb{P}(s_4 = D2 \mid s_1 = C2, s_2 = D2)}_{=\frac{1}{2}}.\end{aligned}$$

We can now use equation (5) in order to compute the second expected value on the right-hand side of equation (3): $\boxed{\mathbb{E}[C_0 \mid s_1 = C2] = \frac{22}{3}}$. We have computed every quantity that we are interested in. Now, we go back to equation (3):

$$\begin{aligned}\mathbb{E}[C_0] = &\underbrace{\mathbb{E}\left[C_0 \mid s_1 \in \{C4, D3\}\right]}_{=6}\underbrace{\mathbb{P}(s_1 \in \{C4, D3\})}_{=\frac{1}{2}} \\ &+ \underbrace{\mathbb{E}\left[C_0 \mid s_1 \in \{B3, C2\}\right]}_{=\frac{22}{3}}\underbrace{\mathbb{P}(s_1 \in \{B3, C2\})}_{=\frac{1}{2}} \\ &= \frac{20}{3} = 6.\bar{6}.\end{aligned}$$

3.2. RL solution

We want to be able to programmatically solve this toy problem so that we can find the solution for any given initial state. We can use reinforcement learning (RL) to solve this problem.

The reasoning goes as follows: we want to apply RL techniques, such as double Q-learning [2] to find the action-value function and from it the state-value function. However, in order to do this, we need to define the state, action, transition, and reward functions. The key to solving this problem efficiently with RL methods is to observe that the state does not only comprise the position of the robot on the chessboard, but also the "world belief". Here, we define world belief as our current belief of mapping of the arrow keys to the actual directions.

Let $\underline{N} = \{0, 1, \ldots, N-1\}$ denote the set of nonnegative integers ranging from 0 to $N-1$. We define the Markov decision process (MDP) as follows:

- **State space**: The product $\mathcal{S} = \underline{4}\times\underline{4}\times\underline{24}$. The first two factors product stands for the position of the robot in the grid world and the third factor to the ID number of the world, of which there are $4! = 24$. These 24 worlds correspond to the possible ways to permute the four key directions.
- **Action space**: The product $\mathcal{A} = \underline{4} \times \underline{24}$. The first factor corresponds to a particular direction to move the robot. The second factor corresponds to which of the 24 worlds to move the current belief to. Of course, any direction chosen in this set would be the actual direction the robot would move only if there is overlap between the assumed and correct directions.
- **Transition function**: This is a deterministic function of the true world. If the robot is at position (i, j) and the world belief is w, then1
- **Reward function**: A negative reward (penalty) of -10 is incurrent whenever $s \in \mathcal{S}$ is nonterminal. Otherwise, this reward is 0. Furthermore, for each overlap between the assumed and correct directions the agent is rewarded $+1$.

Our implementation [3] of this MDP is carried out using Gymnasium [4], a Python library for RL.

4. Conclusions

Interesting problem solved.

References

[1] Bertsekas, D.P., Tsitsiklis, J.N., 2002. Introduction to probability. volume 1. Athena Scientific Belmont, MA.

[2] Morales, M., 2020. Grokking Deep Reinforcement Learning. Manning Publications.

[3] Satici, A.C., 2024. Convoluted questions. URL: https://github.com/Symplectomorphism/miscellaneous/tree/master/dolambac.

[4] Towers, M., Kwiatkowski, A., Terry, J., Balis, J.U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al., 2024. Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint arXiv:2407.17032 .