*Md. Mahin and Christoph F. Eick*

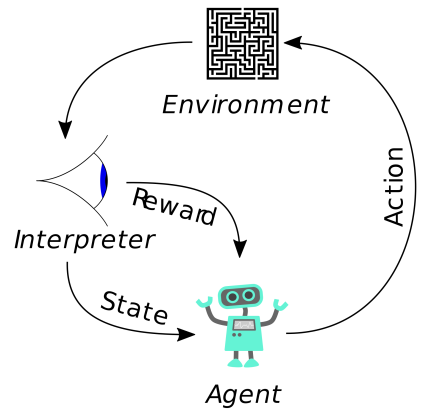
COSC 4368 Project Spring 2021

*Using Reinforcement Learning*

*To Discover Paths in a Transportation World*

Group Project (usually 4-5 Students per Group)

Version 2



Last updated: April 7, 2021

Deadlines: Submit project report and source code via MS Teams by Th., April 15

In this project we will use reinforcement to learn and adapt “promising paths” in robot-style world. Learning objectives of the COSC 4368 Group Project include:

* Understanding basic reinforcement learning concepts such as utilities, policies, learning rates, discount rates and their interactions.
* Obtain experience in designing agent-based systems that explore and learn in initially unknown environment and which are capable to adapt to changes.
* Learning how to conduct experiments that evaluate the performance of reinforcement learning systems and learning to interpret such results.
* Development of visualization techniques summarizing how the agent moves, how the world and the q-table changes, and the system performance.
* Development of path visualization and analysis techniques to interpret and evaluate the behavior of agent-based path-learning systems.
* Learning to develop AI software in a team.

A picture containing shoji, crossword puzzle, building, clipart

Description automatically generated

Figure 1: Visualization of the PD-World.



Figure 2: An Urban Grid World.

In particular in Project2 you will use *Q-learning/SARSA[[1]](#footnote-1) for the PD-Word(*[*http://www2.cs.uh.edu/~ceick/ai/2021-World.pptx*](http://www2.cs.uh.edu/~ceick/ai/2021-World.pptx) *)*, conducting four experiments using different parameters and policies, and summarize and interpret the experimental results. Moreover, you will develop path visualization techniques that are capable to shed light on what paths the learning system actually has learnt from obtained Q-Tables—we call such paths *attractive paths* in the remainder of this document.

In experiments we assume that q values are initialized with 0 at the beginning of the experiment. The following 3 policies will be used in the experiments:

* **PRANDOM**: If pickup and dropoff is applicable, choose this operator; otherwise, choose an applicable operator randomly.
* **PEXPLOIT**: If pickup and dropoff is applicable, choose this operator; otherwise, apply the applicable operator with the highest q-value (break ties by rolling a dice for operators with the same q-value) with probability 0.8 and choose a different applicable operator randomly with probability 0.2.
* **PGREEDY**: If pickup and dropoff is applicable, choose this operator; otherwise, apply the applicable operator with the highest q-value (break ties by rolling a dice for operators with the same q-value).

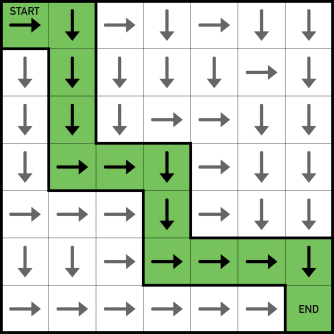


Figure 3: Visualization of an Attractive Path for a Search Problem

Please conduct the following four experiments:

1. In Experiment 1 you use α=0.3 and γ=0.5, and run the traditional Q-learning algorithm for 6000 steps; initially you run the policy PRANDOM for 500 steps, then
   1. Continue running PRANDOM for 5500 more steps
   2. Run PGREEDY for the remaining 5500 steps
   3. Run PEXPLOIT for the remaining 5500 steps

Summarize and interpret the different results you obtain by running these three strategies. Also report one of the final Q-Table of experiment 1.c.

1. Experiment 2 is the same as experiment 1.c except you run the SARSA q-learning variation for 6000 steps. When analyzing Experiment 2 center on comparing the performance of Q-learning and SARSA. Also report one of the final Q-tables of this experiment.
2. In Experiment 3 you rerun either[[2]](#footnote-2) Experiment 1.c or 2 but with learning rates α=0.15 and α=0.45. When interpreting the results focus on analyzing the effects of using the 3 different learning rate on the system performance.
3. Experiment 4 is the somewhat similar to Experiment 1c or 2; you use α=0.3 and γ=0.5 in conjunction with either Q-learning or SARSA[[3]](#footnote-3) as follows: you run PRANDOM for the first 500 steps; next, you run PEXPLOIT; however, after the agent reaches a terminal state the third time, change the pickup locations to (3,1) and (1,3); the drop off locations and the Q-table remain unchanged; finally, you continue running PEXPLOIT with the “new” pickup locations until the agent reaches a terminal state the sixth time. When interpreting the results of this experiment center on analyzing on how well the learning strategy was able to adapt to the change of the pickup locations and to which extend it was able to learn “new” paths and unlearn “old” paths which became obsolete.

For all experiments, if the agent reaches a terminal state, restart the experiment by resetting the PD world to the initial state, but do not reset the Q-table. Run each experiment twice, and report[[4]](#footnote-4) and interpret the results; e.g. utilities computed, rewards obtained in various stages of each experiment.

Assess which experiment obtained the best results[[5]](#footnote-5). Next, analyze the various q-tables you created and try identify attractive paths[[6]](#footnote-6) in the obtained q-tables, if there are any. Moreover, briefly assess if your system gets better after it solved a few PD-world problems—reached the terminal state at least once. Briefly analyze to which extend the results of the two different runs agree and disagree in the 5 experiments. Also comment on the influence of the discount rate γby comparing the results of the third and fourth experiment. Finally, analyze how well the approach adapted to change in the 5th experiment.

Moreover,

* Make sure that you use different random generator seeds in different runs of the same experiment to obtain different results—having identical results for the 2 runs of the same experiment is unacceptable. It is okay just to report and interpret the Q-tables for the better of the two runs for each experiment, but you should report the performance variables for all ten runs.
* Use the two q-learning state spaces recommended in the PD World slide show[[7]](#footnote-7)!
* You should use the traditional Q-learning and SARSA algorithm in the project and not any other Q-learning variations or reinforcement learning algorithms!
* As far as counting operator applications in performance measures is concerned, you should never count operators that are not applicable in a particular state.
* Students that provide good methods for visualizing q-tables and good visualizations for the analysis of attractive paths obtain extra credit.
* Allow in your software design that the positions of dropoff and pickup positions might be changing before and during a run; otherwise, you will need to write a lot of additional software for experiment5.
* Evidence of the running of your system has to be provided using screen shots that will be delivered in a separate document.
* Groups that develop a very well designed and visually appealing visualization component will receive extra credit for this part of the 4368 Group Project!

Write a 8-12 page report that summarizes the findings of the project. Be aware of the fact that at least 15% of the points available for this project are allocated to the interpretation of the experimental results. Finally, submit the source code of the software you wrote in addition to your project report and be ready to demo the system you developed. Be also prepared to demo the system you developed.

Project Links

<http://www-all.cs.umass.edu/rlr/domains.html>

<http://courses.cs.washington.edu/courses/cse473/15sp/assignments/project3/project3.html>

<http://ai.berkeley.edu/project_overview.html>

<https://github.com/kristofvanmoffaert/Gridworld>

<http://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_td.html>

<https://mediatum.ub.tum.de/doc/1238753/1238753.pdf>

<http://www2.econ.iastate.edu/tesfatsi/RLUsersGuide.ICAC2005.pdf>

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.7978&rep=rep1&type=pdf>

1. SARSA is a variation of Q-learning that uses the q-value of the actually chosen action and not the q-value of the best action! [↑](#footnote-ref-1)
2. You have a choice here! [↑](#footnote-ref-2)
3. Again you have a choice here. [↑](#footnote-ref-3)
4. Additionally, report the following Q-tables for Experiments 2 (or Experiment 3 if you prefer that, in this case you will only need to report the final Q-Table of Experiment 2) in your report a) when the first drop-off location is filled (the fifth block has been delivered to it) and b) when a terminal state is reached and c) the final Q-table of each experiment.

   The Q-table in the screenshot should be presented as a matrix, with *s* rows (states) and *t* columns (operators).  Thus, the Q-table for recommended state space has 25 x 2 rows and 6 columns; however, the q-values for the drop-off and pickup operators do not need to be reported. [↑](#footnote-ref-4)
5. Provide graphs that show, how the algorithm’s performance variables changed over the duration of the experiment in the three experiments. [↑](#footnote-ref-5)
6. A path going from (i,j) to (i’,j’) is attractive if, the q-values of the motion operators are high in comparison to other directions. Be aware of the fact that different paths are attractive for agents who hold a block and for agents how do not hold a block. [↑](#footnote-ref-6)
7. See PD-2021World slide show; if you want to use a different state space, consult with Mahin or Dr. Eick. [↑](#footnote-ref-7)