*Christoph F. Eick and Romita Banerjee*

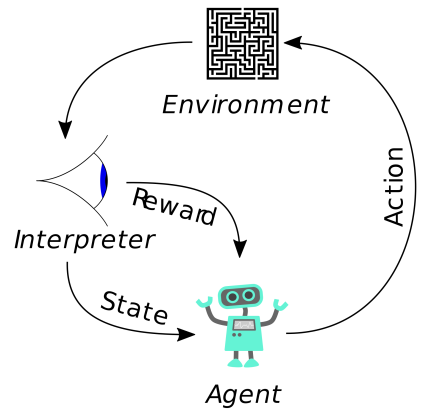
COSC 4368 Project Spring 2019

*Using Reinforcement Learning*

*To Discover Paths in a Transportation World*

Group Project (usually 4-5 Students per Group)

Version 5



Deadlines: Groups 5-13: Deliverables: April 18, 11p, 7-minute presentation on: Monday, April 22; Groups 1-4: Deliverables: April 20, **11a**, 7-minute presentation on Wednesday, April 17.

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.

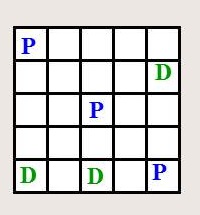


Figure 1: Visualization of the PD-World.



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/2019-World.pptx>), conducting five 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 you conduct, the learning rate is α=0.3 and the discount rate is assumed to be γ=0.5 (except Experiment 4) and 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.
* **PEPLOIT**: 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).

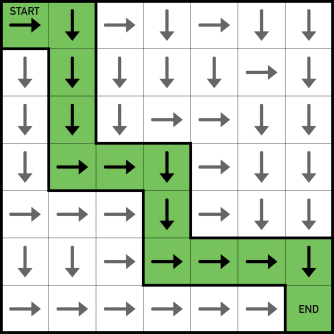


Figure 3: Visualization of an Attractive Path for a Search Problem

The five experiments you conduct are as follows:

1. In Experiment 1 you use α=0.3, and run the Q-learning algorithm[[2]](#footnote-2) for 4000 steps with policy PRANDOM; then run PGREEDY for 4000 steps[[3]](#footnote-3). Display and interpret the Q-table you obtained in the middle and the end of the experiment.
2. In Experiment 2 you use α=0.3, and run the Q-learning algorithm for 8000 steps with policy PEXPLOIT—however, use policy PRANDOM for the first 200 steps of the experiment, and then switch to PEXPLOIT for the remainder of the experiment. Analyze the performance variables and summarize what was learnt by analyzing the q-table at different stages of the experiment.
3. In Experiment 3 you use α=0.3, and run the SARSA q-learning variation for 8000 steps with policy PEXPLOIT—however, use policy PRANDOM for the first 200 steps of the experiment, and then switch to PEXPLOIT for the remainder of the experiment. When analyzing Experiment 3 center on comparing the performance of Q-learning and SARSA. Also report the final q-table of this experiment.
4. Experiment 4 is the same as Experiment3, except you use a discount rate of γ=1.0 (instead of 0.5). However, just focus on analyzing the impact of the discount rate γ in this experiment on the performance variables and not on other matters[[4]](#footnote-4).
5. Experiment 5 is the same as Experiment 2; that is α=0.3 and γ=0.5 and you use PEXPLOIT with Q-learning except for the first 200 operator applications you use PRANDOM; however, after the agent reaches a terminal state the second time, you will swap pickup and drop-off locations. When analyzing the results of this experiment focus how well and quickly the q-learning approach was able to adapt to this change.

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[[5]](#footnote-5) and interpret the results; e.g. utilities computed, rewards obtained in various stages of each experiment.

Assess which experiment obtained the best results[[6]](#footnote-6). Next, analyze the various q-tables you created and try identify attractive paths[[7]](#footnote-7) 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 q-learning state space recommended in the PD World slide show[[8]](#footnote-8)!
* 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 25% 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. More detailed project submission guidelines will be added to this specification no later than March 21, 2019.

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 the option to use the SARSA algorithm instead, if you prefer that! Make clear in your report which algorithm you used in Experiment1. [↑](#footnote-ref-2)
3. Do **not** **reset** the q-table after 4000 steps! [↑](#footnote-ref-3)
4. For groups with only 4 students, conducting experiment 4 is optional! [↑](#footnote-ref-4)
5. 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-5)
6. Provide graphs that show, how the algorithm’s performance variables changed over the duration of the experiment in the three experiments. [↑](#footnote-ref-6)
7. 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-7)
8. See PD-2019World slide show; if you want to use a different state space, consult with Dr. Eick. [↑](#footnote-ref-8)