*Christoph F. Eick*

COSC 6368 (Fall 2016)

Project 2: Learning Paths from Feedback Using Q-Learning

Group Project ((2-)3 Students per Group)

Third Draft



Deadline: Sa., November 19, 11p (if you deliver by Th., Nov. 17 you get a 5% bonus)

Last updated: October 20, 12:35p

In this project we will use reinforcement to learn and adapt “promising paths” in robot-style world. Learning objectives of Project2 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.

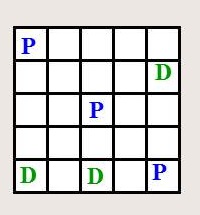


Figure 1: Visualization of the PD-World

In particular in Project2 you will use *q-learning for the PD-Word* (<http://www2.cs.uh.edu/~ceick/ai/2016-World.pptx>), conducting 6 experiments using different parameters and policies, and summarize and interpret the experimental results. Moreover, you will develop path visualization and analysis techniques that are capable to shed light on what a tested path learning system actually has learnt about attractive path. In the experiments, you conduct, the discount rate is assumed to be γ=0.3 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 operator randomly.
* **PExploit1**: 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 0.65 and choose an applicable operator randomly with probability 0.35
* **PExploit2**: 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 0.9 and choose an applicable operator randomly with probability 0.1.

The experiments are as follows:

1. In Experiment 1 you use α=0.3, and run the Q-learning algorithm for 10000 steps[[1]](#footnote-1) with policy PRANDOM. Interpret the obtained Q-Table.
2. In Experiment 2 you use α=0.3, and run the Q-learning algorithm for 10000 steps[[2]](#footnote-2) with policy PEXPLOIT1—however, use policy PRANDOM for the first 100 steps of the experiment, and then switch to PEXPLOIT1 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 Q-learning algorithm for 10000 steps2 with policy PEXPLOIT2—however, use policy PRANDOM for the first 100 steps of the experiment, and then switch to PEXPLOIT2 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.
4. Experiment 4 is the same as Experiment3, except you use a learning rate of α=0.5. However, just focus on analyzing the impact of the learning rate α in this experiment on the performance variables and the learnt paths.
5. Experiment 5 is the same as Experiment 4 (that is α=0.5 and you use PEXPLOIT2 except for the first 100 operator applications); 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.[[3]](#footnote-3)
6. Experiment 6 the learning rate is α=0.5 and but use PEXPLOIT1 except for the first 100 operator applications for which you use PRANDOM; however, after the agent reaches a terminal state the second time, you will change the pickup locations to: (2,2), (4,4), (1,5). When analyzing the results of this experiment focus how well and quickly the q-learning approach was able to adapt to this change, also compare this result with those you obtained for Experiment 5.

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; try to assess if the five tested strategies learnt something useful in the experiment. Also analyze if the performance enhanced after restarts, and identify attractive paths[[5]](#footnote-5) at various stages of the experiments. Also comment on the influence of the parameter α by comparing the results of the third and fourth experiment. Try to assess which of the 4 strategies that were used in the first 4 experiments performed best. Finally, analyze how well the approach adapted to change in the last two experiments. Also summarize experimental result of each experiment in expressive graphs that show, how the algorithm’s performance variables changed over the duration of the experiment. Also analyze to which extend the results of the two different runs agree and disagree in the 5 experiments. Finally, develop a visualization component that summarizes how the PD-worlds changes, what is learnt, and how well the system performs with respect to the different performance measures.

Moreover,

* Students that provide good methods for visualizing q-tables and good visualizations for the analysis of attractive paths obtain extra credit.
* Make sure that you use different random generator seeds in different runs of the same experiment to obtain different results—having identical results in the 2 runs of the same experiment is unacceptable.
* Parameter settings proposed in this specification are subject to change; therefore, check the course webpage and Nguyen’s webpage frequently.
* Either use reinforcement learning statespace1 or statespace2 in the project[[6]](#footnote-6); if you like to use a different state-space in the project see Dr. Eick during his office hour or after the lecture to approve the alternative state space; if you use other state spaces for q-learning without Dr. Eick’s approval you will be penalized.
* You should use the traditional Q-learning algorithm in the project and not any other Q-learning or reinforcement learning algorithm—if you like to explore additional algorithms talk to Dr. Eick.
* Allow in your implementation that the positions of dropoff and pickup states might be changing before and during a run.
* As far as counting operator applications in performance measures is concerned, you should never count operators that are not applicable in a particular state.
* 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 very well designed and visually appealing visualization component will receive extra credit for this part of the Project2.

Write a 7-11 page report that summarizes the findings of the project. Be aware of the fact that at least 30% 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 Project2 submission guidelines will be posted in Nguyen’s Webpage by November 6 the latest!

Project2 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://archive2.cra.org/Activities/craw_archive/dmp/awards/2004/Coggan/FinalReport.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. If the agent reaches a terminal state, reset the PD world to the initial state, but do not change the Q-table. [↑](#footnote-ref-1)
2. 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; however, if a terminal state is reached the fourth time terminate the experiment prematurely; in this case Q-learning was run less than 10,000 steps. [↑](#footnote-ref-2)
3. For groups with only 2 students, conducting experiment 5 is not required! [↑](#footnote-ref-3)
4. Report the following Q-tables for Experiment 1, 2, and 5 in your report: a) after 100 steps and for each run: b) when the first drop-off location is filled (the fifth block has been delivered to it) and c) when a terminal state is reached. Additionally, you must provide “screen shots” of program runs, displaying the Q-tables and also the corresponding full state for Experiments 2 and 4 for iterations 100, 200, 300,…, 2000. Screen shots will have to be submitted as a separate file.

   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 state space 2 has 25 x 2 rows and 6 columns; however, the q-values for the drop-off and pickup operators do not necessarily have to be reported. [↑](#footnote-ref-4)
5. A path is attractive if, the q-values of the motion operators going from (i,j) to (i’,j’) are high; in this analysis, keep x fixed to 1 or 0 and only analyze states with (s,t,u) fixed to (1,1,1) if you use state space2; make a separate analysis for q-values of states with x=1 and of states with x=0 [↑](#footnote-ref-5)
6. The two RL state spaces are discussed in PD-World slide show. [↑](#footnote-ref-6)