CoMo
Computational Modeling Lab

Multi-Agent Learning Systems
Assignment 4: Q-Learning

Deadline: 9 December 2010

General remarks:

- Mail your solutions to mmihaylo@vub.ac.be, or hand-in a paper copy to Prof. Nowé.
- For electronic versions please provide a single (self-contained) *.PDF or *.DOC file.
- Put your name **both** on the document and in the file name.

Windy Gridworld

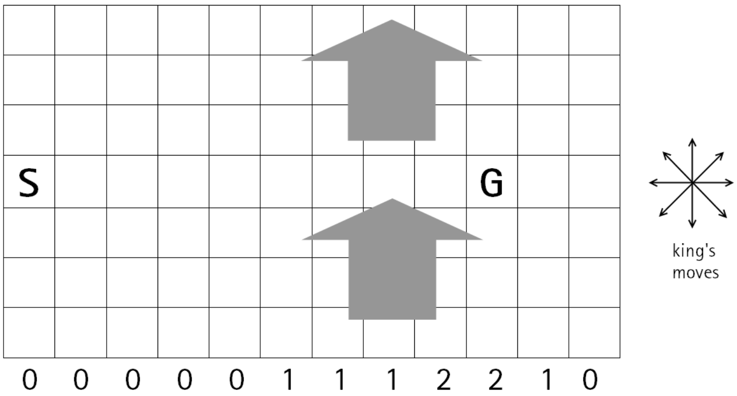

Figure 1: Gridworld in which movement is altered by a location-dependent, upward stochastic "wind".

Figure 1 shows a standard gridworld, with start (S) and goal (G) cells, but with one difference: there is a crosswind upward through the middle of the grid. The available actions in each cell are the king's moves — 8 actions in total for each cell. If any action would bring you outside the gridworld, you end up in the nearest cell (e.g. going northeast in the top left cell will bring you one cell to the right). In the middle region the resultant next cells are shifted upward by a stochastic "wind", the mean strength of which varies column by column. The mean strength of the wind is given below each column, in number of cells shifted upward. Due to stochasticity, the wind sometimes varies by 1 from the mean values given for each column. That is, a third of the time you are shifted upwards exactly according to the values indicated below the column, a third of the time you are shifted one cell above that, and another third of the time you are shifted one cell below that. For example, if you are one cell to the right of the goal (wind=1) and you move *left*, then one-third of the time you move one cell above the goal, one-third of the time you move two cells above the goal, and one-third of the time you move to the goal.

1 Q-learning

Implement the Q-learning algorithm[1] in the above problem with $\alpha = 0.1$, $\gamma = 0.9$ and initial $Q(s, a) = 0$ for all s, a. Each action generates a reward of $r_s = -1$, except for the actions that lead immediately to the goal cell ($r_g = 10$). Use the ϵ-Greedy action selection method with $\epsilon = 0.2$.

1.1 Plotting

Run the Q-learning algorithm for 5000 episodes and at the end of the learning plot the following on a single graph:

- the grid of the Windy gridworld
- arrow(s) in each cell indicating the best action(s)[2]
- one run from start to goal with the Greedy action selection (no exploration) on the learned policy showing the action selected in each cell (use here a different colour)
- for the same run above, plot a line displaying the path taken (i.e. all the visited cells) during that run

Figure 2 shows an example of a learned policy together with a sample run following that policy (applied to a slightly modified scenario).

In addition, plot the following two graphs:

- Total collected reward per episode versus the episode number
- Number of steps to reach the goal per episode versus the episode number

Discuss your results!

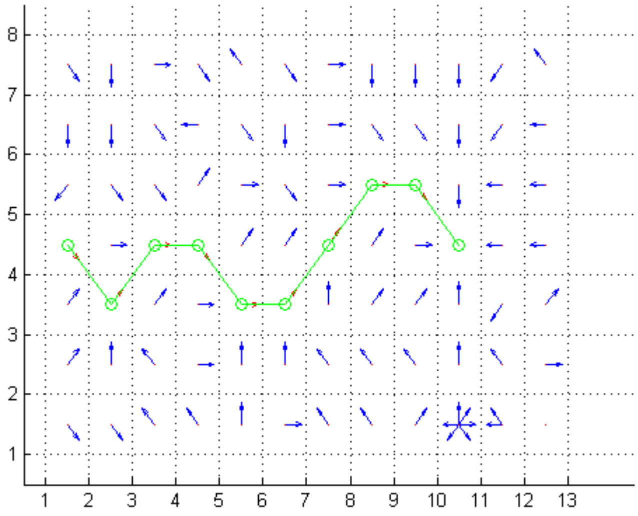

Figure 2: An example of a learned policy (arrows = best action) and a sample run following that policy (green path = visited cells). Note that in this example the scenario has been modified.

[1] The pseudo-code can be found at http://webdocs.cs.ualberta.ca/~sutton/book/ebook/node65.html

[2] In Matlab you can use the *quiver* function for this.

1.2 Discussion

Discuss the following:

1. In what way will the policy change if we set the reward for reaching the goal to $r_g = 1000$ instead of 10, while keeping the other parameters the same? What can be changed[3] in order to obtain a similar policy to the one you had before?

2. In what way will the policy change if we set the reward for each step to $r_s = 1$ instead of -1, while keeping the other parameters the same?

3. How will the learning process change if we make $\epsilon = 1$, while keeping the other parameters the same?

4. How does the stochasticity of the wind influence the learning?

5. Consider we would like to incorporate planning in our learning process in order to speed up the learning. How can you extend/change the Dyna-Q algorithm[4] to non-deterministic environments like our Windy Gridworld?

[3] except for changing the r_g back to 10 :)

[4] The pseudo-code can be found at http://webdocs.cs.ualberta.ca/~sutton/book/ebook/node96.html