

Q-learning and K-means



Course: Machine Learning
Instructor: Dr. Mirela Popa

Student name: **Martin Michaux**

Student ID: **i6220118**

Handing in

Upload a single report in form of a PDF. E.g. make a scan. Hand in code in form of a single zip file. Submissions by email or other types of archives are not accepted. Thank you for your understanding.

For the first part (a) include in the report a short description of your result, the best policy and your interpretation of the role of the two parameters α and γ . For the second part (b) include the required explanations.

Filling in

You can use this Word file to answer your questions in a digital form. Alternatively, you can print the document, fill it in, and upload a scan. Make sure that we can read your handwriting.

Graded: Code and Paper assignment: Q-learning

Your task is to implement the SARSA algorithm for a simple single player game, in which an agent explores the environment, collects rewards and eventually arrives in the destination state, finishing the game (e.g. snake game, PacMan). Your goal is to maximize the final score (which is obtained by arriving in the shortest time to the destination state), while also exploring the environment. The grid is 4x4 and the set of valid actions are move up, down, right, left, except for the boundary walls, where only specific actions are possible. All the other values are currently initialized, but you can adjust them as you consider. A part of the code is provided for you in Canvas (tutorial6.ipynb); your task is to complete the missing steps, including the update of the value function.

The algorithm is the following:

For each s, a initialize the state $Q(s,a)$ to zero

Start from a random state s

Do forever:

- Select an action a randomly and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $Q(s,a)$ as follows

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma Q(s', a'))$$

- Make the transition $s \leftarrow s'$
- If s' is the destination state, then stop

Include in this report your observations about the process, the obtained Q matrix and your interpretation about the role of the two parameters alpha and gamma and how do they affect the final policy.

The process of that algorithm simply consists of the following:

- First initialize random state in the 4x4 grid
- For X (=100 for us) number of iterations:
 - Until we reach a terminal:
 - 1) Initialize new random action from the current state
 - 2) Calculate the next state and the new reward of our current state
 - 3) Update the Q table at the current state

What I observed in the process is that we make 100 iterations in order to “train” (and update) our Q values the most as possible in all the possible ways.

Those iterations represent a “simulation” of the agent on the grid in order to get the quality of specific actions in specific states.

At the end of the process, we can compute our final Q table to make the decision of the agent.

Best policy for maximizing the score (include it as a matrix/drawing)

After several(100) computations (iterations) of the Q table update from a random state, I computed the final update of that table and got the following output:

```
[ [ 0.          -0.50537616 -0.85636023 -0.96748114]
  [-0.97399145 -1.04706077 -1.023284   -0.9886986 ]
  [-1.10085934 -1.10147687 -1.09733602 -1.06914887]
  [-1.11111108 -1.11111046 -1.11109876  0.          ] ]
```

The aspect of the table can be explained easily. In fact, our terminal points were the two coordinates (0,0) and (4,4), those states contain the quality value of 0 because they are our point and each other reward are negative because the initial reward was -1.

Now, let's analyze the other states. From each point of our table (state), there exists an "optimal" path that follows the best rewards on the table. Indeed, since the SARSA algorithm chooses the action that was actually chosen randomly, the agent explores the environment, collects rewards and arrives in the destination state, finishing the game with a maximized final score.

Explanation of the role of the parameters:

The main role of the alpha parameters, aka the learning rate, is to handle the update of the Q value (if alpha was equal to 0, the quality would not be updated and the Q table would be null). Indeed, it manages the proportion of the new quality compared to the old one.

For the gamma, aka the discount factor, its point is to “scale” the new rewards as better or worth than the immediate rewards. If that value was equal to 0, the agent would consider only the future reward, which would make it too optimistic.

The grid size modifies the complexity of our algorithm. The bigger the grid is, the more we have to iterate and update the Q table to be more accurate.

The state terminals:

- The number of state terminals changes the number needed of iterations. Indeed, the more there are terminals, the faster we will reach one of them.
- The location of state terminals also changes the iterations number. As above, the more the location of the state terminals is in the middle, the faster we will reach one of them.

The valid actions have not a big impact on the number of iterations since either way, it's random so, until we reach a terminal, we will still use a random. At least the actions need to be consistent with our problem.

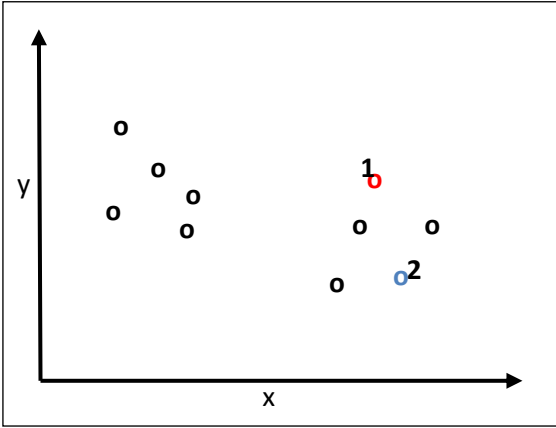
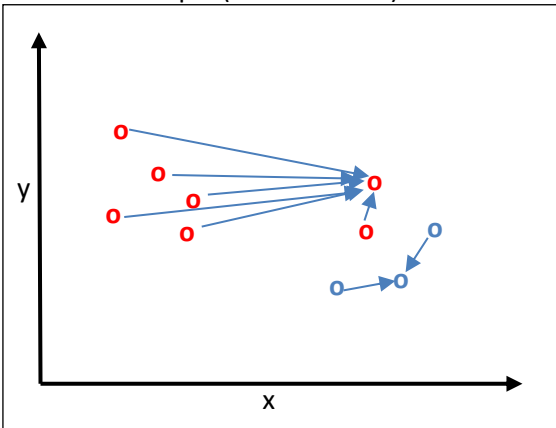
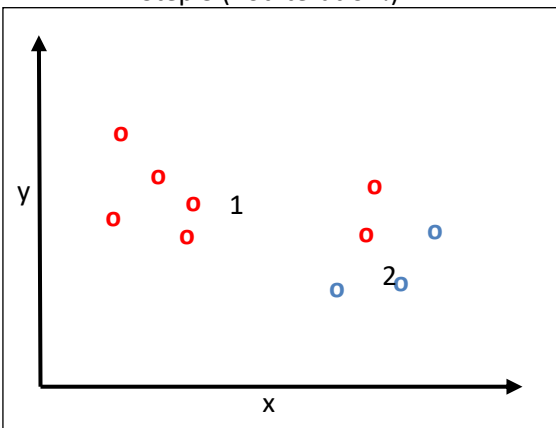
The current reward value is important at the beginning of the process since each first update of the Q value are computed from it, but its value doesn't really matter as soon as it is not null.

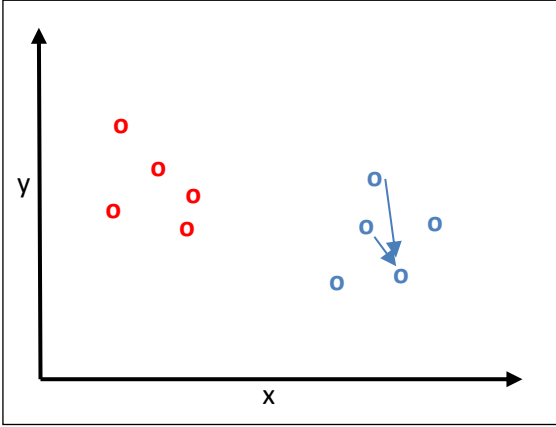
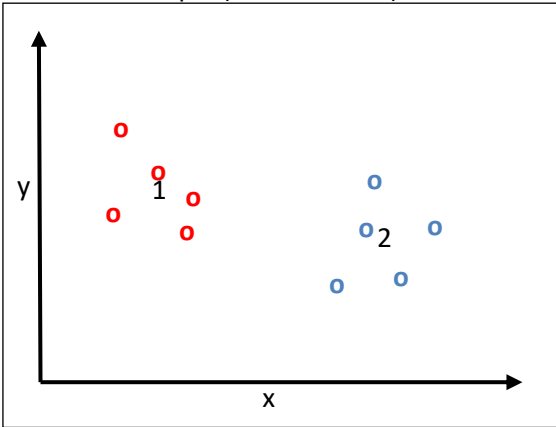
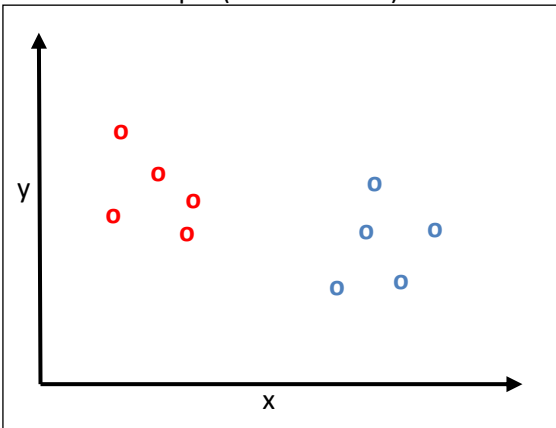
The number of iterations modifies the accuracy of our Q table, the higher that number is, the more accurate are our Q values.

Graded: Paper assignment: K-Means

Given the following data set, show (with drawings) and explain (with your own words) the different steps of a k-means algorithm when $k=2$. Show and explain individual steps of the algorithm – not just full iterations.

(Explanation of symbols: o = data points; 1 = marker for first centroid, 2 = marker second centroid)

<p>Step 1 (not iteration!):</p> 	<p>Explanation:</p> <p><i>Initialization: Centroids get assigned to random locations. Here two random points from the data set are picked as initial seeds.</i></p>
<p>Step 2 (not iteration!):</p> 	<p>Explanation:</p> <p>For each instance in the model, assign (not reassign because it's the first iteration) to that instance the cluster of the nearest centroid.</p>
<p>Step 3 (not iteration!):</p> 	<p>Explanation:</p> <p>Now, simply initialize the new centroids in the model corresponding to the average center of each cluster (the centroids could be instances or locations in the model).</p>

<p>Step 4 (not iteration!):</p>  <p>A scatter plot on an x-y coordinate system. There are two clusters of points: one cluster of red points on the left and one cluster of blue points on the right. Two blue arrows point from the blue points towards a central point within the blue cluster, representing the centroid.</p>	<p>Explanation:</p> <p>One more time, for each instance in the model, reassign to that instance the cluster of the nearest centroid.</p>
<p>Step 5 (not iteration!):</p>  <p>A scatter plot on an x-y coordinate system. There are two clusters of points: one cluster of red points on the left and one cluster of blue points on the right. The red cluster has a point labeled '1' and the blue cluster has a point labeled '2', representing the new centroids.</p>	<p>Explanation:</p> <p>Again, initialize the new centroids in the model corresponding to the average center of each cluster (the centroids could be instances or locations in the model).</p>
<p>Step 6 (not iteration!):</p>  <p>A scatter plot on an x-y coordinate system. There are two clusters of points: one cluster of red points on the left and one cluster of blue points on the right. The points are now fully assigned to their respective clusters.</p>	<p>Explanation:</p> <p>Finally, we can see that we can't neither reassign nor re-initializing new centroids because the clustering has converged.</p>