

Q-learning and K-means



Course: Machine Learning
Instructor: Dr. Mirela Popa

Student name: Giuseppe Lorenzo Pompigna Student ID: i6233748

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 hand-writing.

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 graphs at the end of the task section show the results of the search, Q is improved at each iteration, then explored with a **roll-forward search**: The agent is again put in a random location from where it starts applying **the best action obtained from the highest Q value** in the neighborhood of the current location, accessible via single valid actions. If the exploration is successful e.g., the terminal state is reached, then the testing procedure assign value **1** to the current instance, else **0**. Then we collect the **success rate** (y axis) over 100 trials, per instance, of roll-forward search from starting random locations in the grid.

As we can see from the results of the tests the parameters don't really affect the effectiveness of the pre-computed policy-based search which is indeed successful in most of the cases especially with num. iterations parameter (which has a success rate of at least 70%).

However, some change can be seen in the Q table:

The higher alpha value tends to assign lower Q value to the current position, even if it is very close to the terminal state. This is to prevent to achieve a fast local optimum without enough exploration. We can see that a 0 alpha value will achieve a successful search only in 30% of the test cases. Indeed, the Q table looks like a 4x4 grid of value max score (100).

The lower gamma value instead is used to penalize more the states which are not terminal by assigning them a lower Q value.

This is to make the search more effective by discouraging the agent following wrong paths.

Indeed, we can see there are more failures when gamma is close to 1.

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

Q table:

Terminal states = [0,0], [3,3]

alpha = 0.6

gamma = 0.6

num. iterations = 50

100.00	88.410	36.939	25.184
88.410	9.8410	-0.956	1.0260
60.982	32.538	9.8410	53.840
1.3725	9.8410	88.410	100.00

Policy test:

Trials = 100

Success = 100

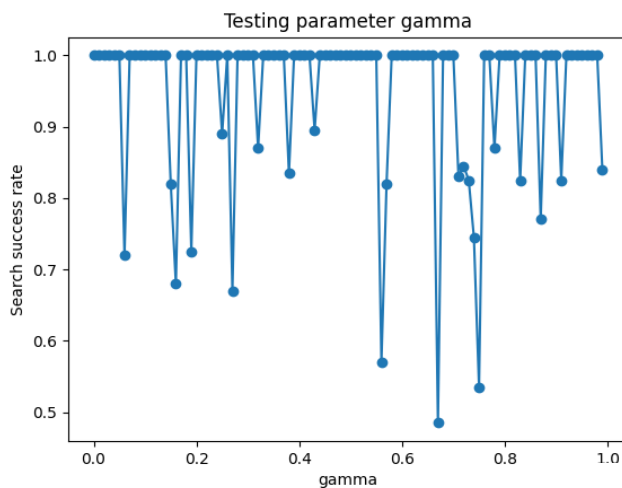
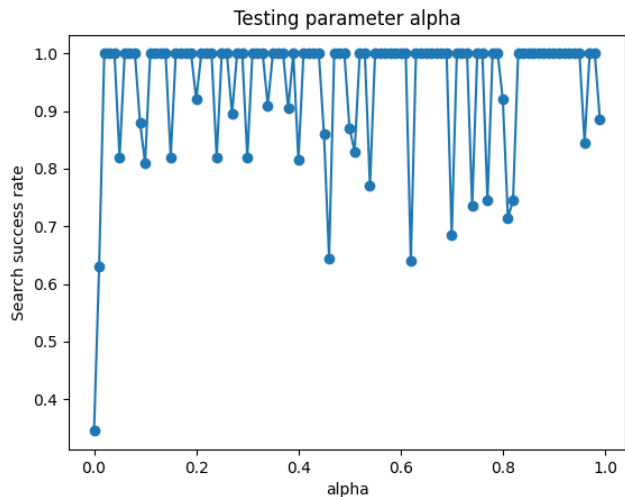
*This one was just picked because being nice and well explored by the agent, but there were many configurations which achieved 100% success rate in the policy test search, as shown in figures.

Explanation of the role of the parameters:

The alpha value is tested over a range [0,1] with step (0,01)

Gamma=0.115

Num. iterations=150.



The gamma value is tested over a range [0,1] with step (0,01)

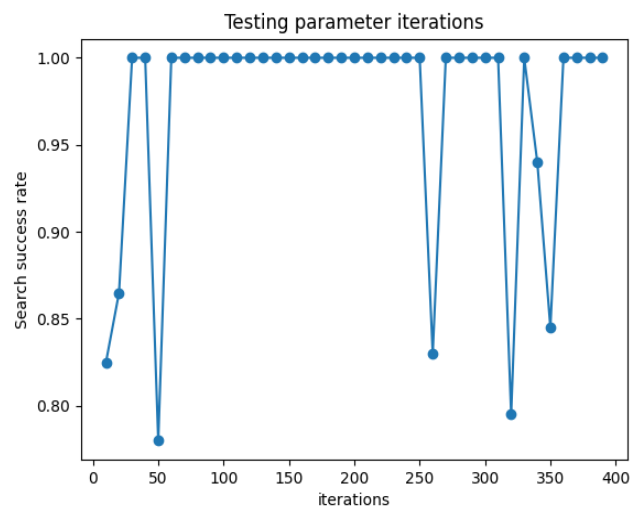
Alpha=0.19

Num. iterations=150

The num. iterations value is tested over a range [0,400] with step (10)

Gamma=0.115


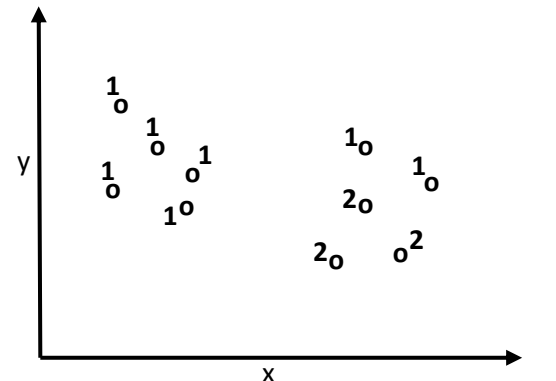
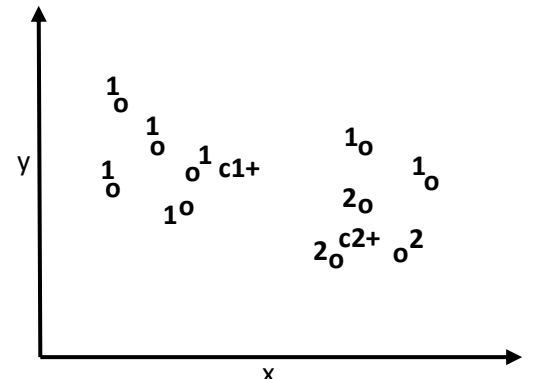
Alpha=0.19

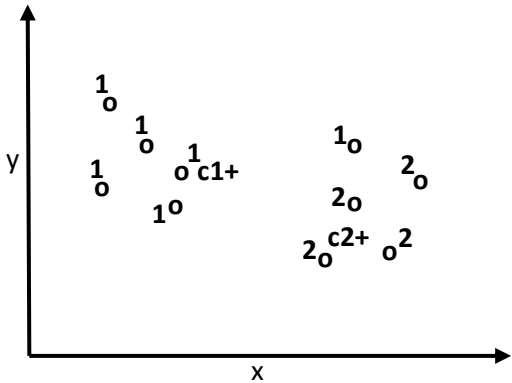
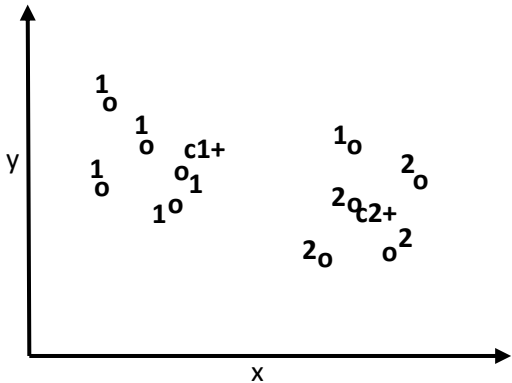
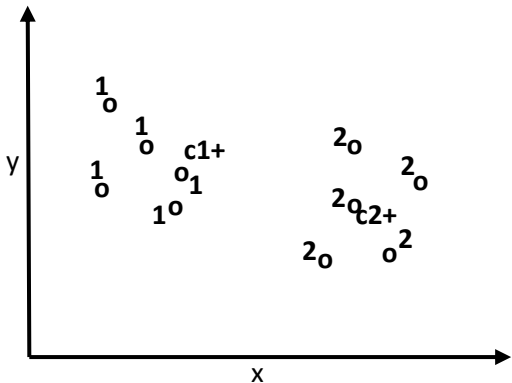


Graded: Paper assignment: K-Means

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>Measure distance from each centroid and assign to every instance the label of the closest centroid ($m1 = 0.7$, $m2 = 0.3$)</p>
<p>Step 3 (not iteration!):</p> 	<p>Explanation:</p> <p>Calculate centre of each cluster k by average of x values</p>

<p>Step 4 (not iteration!):</p> 	<p>Explanation:</p> <p>Reassign labels based on Shortest distance to centroid $c(k)$</p>
<p>Step 5 (not iteration!):</p> 	<p>Explanation:</p> <p>Calculate again centroids c_k based on average of $x(k)$ points</p>
<p>Step 6 (not iteration!):</p> 	<p>Explanation:</p> <p>Reassign labels based on Shortest distance to centroid $c(k)$</p>