SARSA and K-means

Course: Machine Learning

Instructor: Dr. Mirela Popa

Student name: Claudia Sanchez\_\_\_\_\_\_\_ Student ID:\_i6242189\_\_\_

**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 alpha and gamma. 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: SARSA**

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 initialize the state to zero

Start from a random state s

Do forever:

* Select an action randomly and execute it
* Receive immediate reward
* Observe the new state
* Update the table entry for as follows
* 
* Make the transition
* If 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.

Increasing gamma to 0.5 and leaving alpha to 0.1 resulted in having bigger absolute results (i.e. the negative values were smaller ) for every state given its corresponding action. (I slightly modified the grid to have a 16x4 matrix representing every possible state, rows, with every possible corresponding action, columns).

Doing vice-versa the results didn’t differ too many from the original grid obtained with alpha = 0.1, gamma = 0.1.

Increasing alpha to 1.0 rendered smaller abs results from the grid, likely when gamma was raised to 1.0 and alpha was left to 0.1 . But equaling gamma to 1.0 means that the future rewards is the same as the current rewards so the agent will just do its best action and forget about exploration.

Following I increased gamma to 0.5 and alpha to 1.0, by this I obtained a grid containing only values that were very close to 1, thus showing these values are close to the best policy.

After doing some research, I read that given a stochastic environment, which does apply to our case given that our actions chosen randomy , alpha, the learning rate, should converge to 0 as approaching infinity. Thus, I experimented with alpha being 0.01 and gamma as 0.9, so the agent goes into the best action directly, this resulted in some overall good values since all were very close to -1 .

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

Table

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| G <- | <- | <- | |> |
| ^- (up) | <- | |> | |> |
| ^- | |> (down) | |> | |> |
| -> | -> | -> | G |

**Explanation of the role of the parameters:**

**Gamma is the value that determines the weigh or importance of the future reward, enhancing exploration. Whereas alpha is the learning rate, the value of alpha determines how quick the learning occurs.**

**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)

|  |  |
| --- | --- |
| Step 1 (not iteration!):  **2**  **1**  x  y  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o** | Explanation:  *Initialization: Centroids get assigned to random locations. Here two random points from the data set are picked as initial seeds.* |
| Step 2 (not iteration!):  x  y  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o** | Explanation:  \_\_we assign all the points to the closest cluster centroid |
| Step 3 (not iteration!):  x  y  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o** | Explanation:  \_recompute the centroid of newly formed clusters by the mean of all points assign to the cluster moving the centroid along the graph |
| Step 4 (not iteration!):  x  y  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o** | Explanation:  \_\_\_reassign all the points to the closest cluster centroid \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Step 5 (not iteration!):  x  y  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o** | Explanation:  \_\_\_\_\_recompute the centroids of our new clusters \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Step 6 (not iteration!):  x  y  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o**  **o** | Explanation:  \_\_assign all the points to the closest cluster centroid\_  Recompute centroids of the new clusters, does it change? No!, are points remaining in the same cluster? Yes! We are done! |