

Reinforcement Learning: Lab Exercises Submission Konstantin Preußner

Description:

This submission includes my work on Exercise 4, with a focus on implementing the Semi-Gradient SARSA and Q-Learning algorithms for the continuous version of the Windy Grid World problem. I also made attempts at the basic requirements of Exercises 1, 2, and 3, which were completed during class. Below is a summary of my work on Exercise 4, where I encountered both successes and challenges.

Solution Overview:

For Exercise 4, I implemented the Semi-Gradient SARSA algorithm using different feature encodings (one-hot, tile, and RBF). The problem involved solving the Windy Grid World with these feature representations. The solution can be found in `contin_grid_world.ipynb` and is structured as follows:

- **Class WindyGridWorld:**
Represents the environment. It includes the step function (for state transitions), the reset function (to restart the environment), and a constructor to initialize the environment.
- **Class SemiGradientSARSA:**
Defines the agent. It includes feature functions for RBF, one-hot encoding, and tile coding. Other functions include `choose_action`, `update` (for learning from experience), and a getter function for the Q-values. Additionally, functions to train the agent and plot the results are included.

Challenges and Observations:

- **Feature Encoding Challenges:**
While one-hot encoding produced good results, both the tile and RBF feature representations were too compute-heavy for my machine to handle effectively.
 - **Tile Encoding:** Despite using only 8 tiles (resulting in 560 features), the calculations caused significant instability. Training took over 7000 steps to reach the goal in some cases, which was unexpected.
 - **RBF Encoding:** Similarly, RBF features also led to high computational demands, and I was unable to optimize the performance for my machine.
- **Instability and Solutions:**
 - **Step Distribution:** The default (1, 0.33) normal distribution for choosing step distances caused instability. Reducing the variance to 0.1 and 0.05 significantly improved the results, preventing the algorithm from learning noisy patterns.
 - **Q-value Instability:** To address numerical instability in Q-values, I added normalization during calculations, which resolved the issue.

Optional Exercise 4:

I also attempted the optional exercise, which modified the Windy Grid World to use angles instead of fixed directions (north, east, south, west). This posed additional challenges, as the agent had to navigate in continuous space with step distances chosen from a normal distribution. Despite using an SGD-based policy, the agent did not appear to learn effectively over 1000 episodes, with high step counts observed in some episodes. This could either be due to the complexity of the continuous problem or a bug in the code that I could not find. The code can be found in `contin_grid_world_optional.ipynb`.

Conclusion:

I successfully implemented the Semi-Gradient SARSA algorithm for the Windy Grid World problem, but faced significant challenges related to feature encoding and computational instability. I made improvements by adjusting the step distribution and normalizing Q-values. The optional exercise, which introduced continuous angles, was more difficult to solve, and further analysis is needed to confirm the learning behavior.