# Reinforcement Learning HW 5

Mert Bilgin (7034879)
mert.bilgin@student.uni-tuebingen.de

Lalitha Sivakumar (6300674)
lalitha.sivakumar@student.uni-tuebingen.de

Kevin Van Le (7314700)
kevin-van.le@student.uni-tuebingen.de

November 19, 2025

## 1 Q-Learning and SARSA

### a)

Q-learning is considered an off-policy control method because, when updating the Q-value for a given state-action pair it does not use the action taken by the current policy. Instead, it updates based on the maximum possible future reward in the next state across all actions, effectively learning about the optimal policy regardless of the policy being followed during exploration.

### b)

No, Q-learning and SARSA are not the same even with greedy action selection. Q-learning is off-policy and always updates using the maximum Q-value for the next state, while SARSA updates based on the actual action taken. They can still make different updates, especially if there are ties in Q-values and the tie breaking is random.

---

In SARSA the behavior and target policy are identical. In Q-Learning the values are updated using the target policy which is greedy while the the behavior policy is used to interact with the environment. If the agents action selection (interaction with the environment) is greedy, then behavior and target policy are also identical. Therefore, SARSA and Q-learning would be identical in this case.

### c)

### a)

The optimal action is to choose **right**, since it leads to an immediate reward of 0, which is higher than the expected return of $-0.1$ from taking the left action.

**b)**

Q-learning will observe that the right action from $A$ always leads to a return of 0, while the left action leads to random rewards averaging $-0.1$. Over time, the updates will favor the right action, as it consistently provides a higher expected value. However with Q-Learning it would take longer to converge to the optimal policy (than with SARSA), because at the beginning of learning with a big action space there are bound to be state action pairs in state B where the agent received positive rewards early on. With Q-Learning the maximum Q-Value at state B is considered for the Q-Value for the state action pair A und going left, and thus for the agent to converge to the optimal policy it would need all Q-Values in State B to be less than 0, and with a large action space in State B this will take a long time.

## 2 Hands-on the Gridworld

Changed code is updated in `agent.py` and `gridworld.py` files.

### 2.1 Q-Learning

**a)**

After training the agent for 100 episodes on the `MazeGrid`, the average return from the start state was only about 0.147, and the value estimates on the grid were less smooth and generally lower than the optimal values shown in the figure.

This happens because Q-learning is a model-free method that only updates values for states and actions that it actually encounters. With just 100 episodes the agent has not explored every part of the grid consistently. However in value iteration there's full access to the transition models and all states simultaneously get updated which allows it to converge to the optimal values.

The Q-learning results become closer to the optimal values if we allow more learning and exploration. For example, increasing the number of episodes or using a higher exploration rate (larger $\epsilon$) helps the agent visit more states and refine its Q-values.

**b)**

After training the Q-learning agent on the `BridgeGrid` with no noise for 100 episodes, the learned Q-values differ from those produced by value iteration. In our run, the average return from the start state was approximately $-16.34$, with many of the states around the bridge having large negative Q-values. The agent strongly avoids the risky bridge area rather than moving toward the high reward on the right.

This happens because Q-learning relies on exploration and the states around the bridge contain large negative penalties. During early exploration the agent quickly experiences poor returns when moving in that direction. It then learns to avoid the bridge entirely and exploits safer actions instead of discovering the optimal path. On the other hand, value iteration considers every possible action in every state and therefore always converges to the optimal policy. Even running for a lot of episodes and higher value of epsilon, the agent still avoids the path on the right and chooses the safe action.

**c)**

The reason the value is higher than the expected return because of the off-policy nature of the Q-Learning algorithm. If we were to use SARSA, the expected return at start state would match with the value of that start state, but with Q-Learning the q-values are set optimistically. What that means is that the agent would update its q-values based on the greedy / current best action taken even if it falls off the cliff due to a random action being selected as the behavior policy is epsilon greedy.