Multi-agent RL

A4.1: Implement exploration for MountainCar **(7 points)**

In this question you will implement *deep exploration* to solve environments like `MountainCar-v0`, which are impossible to learn with random exploration. The question can be completed in the Jupyter notebook `explore.ipynb`, that can be found on Brightspace.

(a) *[1 point]* The `Mountaincar-v0` environment is challenging because of the exploration behavior, random episodes will rarely see any reward before the maximum length is reached, but also because of its extremely unevenly scaled state space: while the car's position is between -1.2 and $+0.6$, the car's velocity is bounded between -0.07 and $+0.07$. Run a DQN experiment, which changes this state space by rescaling each dimension to be between -1 and 1 using the provided `RescaledEnv` wrapper class, for $200k$ environmental steps. To make sure you propagate rewards fast enough to see a result in that time frame (if there is one), run 10 gradient updates for every sampled episode.

(b) *[2 points]* To solve the `Mountaincar-v0` environment, you will need to use some deep exploration technique. To test this we will "cheat" a bit at first: use the provided `CountUncertainty` class to produce intrinsic reward during training. The class must `observe()` states from newly sampled episodes (make sure each transition in your replay buffer has only been observed once) and provides a scaled uncertainty estimate with a `()` method call, for example, `u(state)` for `u = CountUncertainty(...)`. This particular uncertainty class divides the state space into `uncertainty_resolution`^{m} bins, where m is the number of state-dimensions, counts how often observed states fall into these bins and estimates the resulting uncertainty as the standard deviation of an empirical estimator: `uncertainty_scale / sqrt(n)` for `n` observations of the same state. Complete the implementation of `ExplorationDQNExperiment` and run the `Mountaincar-v0` environment with intrinsic rewards for $200k$ steps. Make sure that the experiment only uses intrinsic rewards if the `intrinsic_reward` parameter is **`True`**.

Hint: there are multiple ways to implement intrinsic reward, but the most intuitive is to use the *uncertainty* of having seen the next state of a transition as additional reward. Make sure that you have observed those states before you compute their uncertainty, though, as states with observation-counts of 0 produce extremely large uncertainties that can destabilize learning.

(c) *[2 points]* Your above implementation of `ExplorationDQNExperiment` should be able to solve the `Mountaincar-v0` now. However, the `CountUncertainty` class does not scale to other environments. For example, the `Acrobot-v1` environment has 6-dimensional states and would induce over 15 billion bins if we would count with the same resolution as above. To use intrinsic rewards in high dimensional state spaces, you will complete the implementation of Random Network Distillation (slide 19 of Lecture 7) in the `RNDUncertainty` class. The RND uncertainty estimate shall use 3 linear layers with hidden dimension 1024, ReLU's between them and an output dimension of 256. Test your implementation on the `Mountaincar-v0` environment with intrinsic rewards for $200k$ steps.

Hint: useful `unertainty_scale` parameters depend a lot on your exact implementation of RND. It is recommended to print the average intrinsic reward in `Mountaincar-v0` achieved using `CountUncertainty` (e.g. in the previous question) and then change `unertainty_scale` for this question until the uncertainty of `RNDUncertainty` yields similar intrinsic rewards at the beginning of training. A good value for average intrinsic rewards of an initial episode is ≈ 0.1.

(d) *[2 points]* The `Acrobot-v1` environment requires the agent to learn how to swing up a chain of two connected links. The joint between the two links is (under-) actuated, and swinging the acrobot can exhibit chaotic behavior. Normally this environment allows episodes of up to 500 steps to ensure the agent sees at least some rewards using random exploration. We make this here a bit harder by restricting episodes to 200 steps (like in `Mountaincar-v0`). First run double DQN without intrinsic rewards for $200k$ steps to evaluate whether random exploration is enough to learn in this environment. Next run `Acrobot-v1` with `RNDUncertainty` intrinsic reward.

Hint: If you do not see similar learning as in `Mountcar-v0`, try to adjust `uncertainty_scale` with the same techniques as in A4.1c.

Solution A sample solution is provided in the Jupyter notebook `explore_solution.ipynb` on Brightspace.

A4.2: Nash equilibria **(2 points)**

Let in the following a Nash equilibrium (NE) of a state-less general-sum game of n agents be defined as a joint action $\boldsymbol{a}_* \in \mathcal{A} := \mathcal{A}^1 \times \ldots \times \mathcal{A}^n$ for which holds $r^i(\boldsymbol{a}_*^{-i}, a_*^i) \geq r^i(\boldsymbol{a}_*^{-i}, a^i), \forall a^i \in \mathcal{A}^i, \forall i$.

(a) *[1 point]* Which inequalities of r^1 hold in a state-less two-player zero-sum game?

(b) *[1 point]* Prove analytically that all NE in such a game yield exactly the same reward.

Solution

(a) In a two-player zero-sum game, we have $n = 2$ and $r^1(\boldsymbol{a}) = -r^2(\boldsymbol{a}), \forall \boldsymbol{a} \in \mathcal{A}$. For a given NE $\boldsymbol{a}_*$ holds:
$r^1(\boldsymbol{a}_*) \overset{(1)}{\geq} r^1(a^1, a_*^2)\,, \;\; \forall a^1 \in \mathcal{A}^1\,,$ and $r^2(\boldsymbol{a}_*) \geq r^2(a_*^1, a^2) \Leftrightarrow r^1(\boldsymbol{a}_*) \overset{(2)}{\leq} r^1(a_*^1, a^2)\,, \; \forall a^2 \in \mathcal{A}^2\,.$

(b) Now let us assume there exist a second NE $\boldsymbol{a}'_* \in \mathcal{A}$, for which also holds:
$r^1(a^1, a_*'^2) \overset{(1)}{\leq} r^1(\boldsymbol{a}'_*) \overset{(2)}{\leq} r^1(a_*'^1, a^2)\,, \; \forall \boldsymbol{a} \in \mathcal{A} \;\Rightarrow\; r^1(a_*^1, a_*'^2) \leq r^1(\boldsymbol{a}'_*) \leq r^1(a_*'^1, a_*^2)\,,$ as $\boldsymbol{a}_* \in \mathcal{A}$.

Now we substitute the NE defintion of $\boldsymbol{a}_*$, where $\boldsymbol{a}'_*$ takes the role of the "other" action:

$$\Rightarrow \qquad r^1(\boldsymbol{a}_*) \overset{(2)}{\leq} r^1(a_*^1, a_*'^2) \leq r^1(\boldsymbol{a}'_*) \leq r^1(a_*'^1, a_*^2) \overset{(1)}{\leq} r^1(\boldsymbol{a}_*) \qquad \Rightarrow \qquad r^1(\boldsymbol{a}_*) = r^1(\boldsymbol{a}'_*)\,.$$

A4.3: Solving cyclic games **(3 points)**

Let the following matrix define the rewards $r^1(a^1, a^2)$ of a state-less two-player zero-sum game:

P1\P2	a_1^2	a_2^2
a_1^1	+1	-1
a_2^1	-1	+1

(a) *[1 point]* Simulate a cycle of max-min decisions to show the game does *not* have a Nash equilibrium.

(b) *[1 point]* Compute the analytical expected reward $\mathbb{E}[r^1]$ for a pair of stochastic policies

$$\pi^i(a^i) := \begin{cases} \theta_i & , \text{if } a^i = a_1^i \\ (1-\theta_i) & , \text{if } a^i = a_2^i \end{cases} .$$

(c) *[1 point]* Compute the parameters $\boldsymbol{\theta} = [\theta_1, \theta_2]$ of a mixed Nash equilibrium for π^1 and π^2.

Hint: a mixed Nash equilibrium is a saddle point of $\mathbb{E}[r^1]$ w.r.t. $\boldsymbol{\theta}$.

Bonus-question: can you do the same analysis for the rock-paper-scissors game?

Solution

(a) We begin with $a^2 = a_1^2$. The best (maximimal) choice for player 1 is now $a^1 = a_1^1$. The best (minimal) response to this of player 2 is $a^2 = a_2^2$. The next best responses are $a^1 = a_2^1$, followed by $a^2 = a_1^2$, which closes the cycle. One can see that starting with $a^2 = a_2^2$ would have yielded the same cycle.

(b) $\bar{r} := \mathbb{E}\Big[r^1(a^1, a^2)\Big|_{a^2 \sim \pi^2}^{a^1 \sim \pi^1}\Big] := \theta_1\theta_2 - \theta_1(1-\theta_2) - (1-\theta_1)\theta_2 + (1-\theta_1)(1-\theta_2) = 4\theta_1\theta_2 - 2\theta_1 - 2\theta_2 + 1$

(c) The mixed Nash equilibrium is a saddle point of $\bar{r}$ in θ_1 and θ_2. All extreme points (including saddle points) must have a gradient of zero:

$$\frac{\partial \bar{r}}{\partial \theta_1} = 4\theta_2 - 2 \stackrel{!}{=} 0 \quad \Rightarrow \quad \theta_2 = \tfrac{1}{2}\,, \qquad \frac{\partial \bar{r}}{\partial \theta_2} = 4\theta_1 - 2 \stackrel{!}{=} 0 \quad \Rightarrow \quad \theta_1 = \tfrac{1}{2}\,, \qquad \boldsymbol{\theta} = [\tfrac{1}{2}, \tfrac{1}{2}]\,.$$

Saddle points differ from minima and maxima by a Hessian $\mathbf{H}$ that is neither positive nor negative definite, i.e., some eigenvalues are positive and others are negative:

$$\frac{\partial^2 \bar{r}}{\partial^2 \theta_1} = \frac{\partial^2 \bar{r}}{\partial^2 \theta_2} = 0\,, \qquad \frac{\partial^2 \bar{r}}{\partial \theta_1 \partial \theta_2} = \frac{\partial^2 \bar{r}}{\partial \theta_2 \partial \theta_1} = 4\,, \qquad \Rightarrow \qquad \mathbf{H} := \begin{bmatrix} 0 & 4 \\ 4 & 0 \end{bmatrix} .$$

The Hessian $\mathbf{H}$ has two eigenvalues, 4 and -4, which shows that it is neither positive nor negative definite, and that $\boldsymbol{\theta} = [\frac{1}{2}, \frac{1}{2}]$ is therefore a saddle point.

Bonus-question: In rock-paper-scissors the actions are $a_1 =$ rock, $a_2 =$ paper and $a_3 =$ scissors. This yields:

P1 \P2	a_1^2	a_2^2	a_3^2
a_1^1	0	-1	+1
a_2^1	+1	0	-1
a_3^1	-1	+1	0

The stochastic policies can be written as

$$\pi^i(a^i) = \begin{cases} \theta_{i1} & , \text{if } a^i = a_1^i \\ \theta_{i2} & , \text{if } a^i = a_2^i \\ 1 - \theta_{i1} - \theta_{i2} & , \text{if } a^i = a_3^i \end{cases} .$$

We can compute the average reward as

$$\bar{r} = \theta_{11} - \theta_{12} - \theta_{21} + \theta_{22} - 3\theta_{11}\theta_{22} + 3\theta_{12}\theta_{21}\,,$$

and therefore

$$\frac{\partial \bar{r}}{\partial \theta_{11}} = 1 - 3\theta_{22}\,, \quad \frac{\partial \bar{r}}{\partial \theta_{22}} = 1 - 3\theta_{11}\,, \quad \frac{\partial \bar{r}}{\partial \theta_{12}} = 3\theta_{21} - 1\,, \quad \frac{\partial \bar{r}}{\partial \theta_{21}} = 3\theta_{12} - 1\,.$$

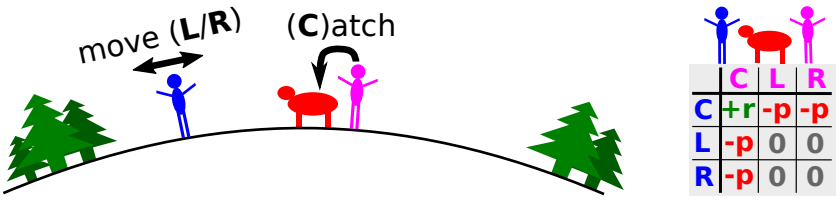
From this we can derive the exreme point $\theta_{11} = \theta_{12} = \theta_{21} = \theta_{22} = \frac{1}{3}$. The Hessian matrix is

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & -3 \\ 0 & 0 & +3 & 0 \\ 0 & +3 & 0 & 0 \\ -3 & 0 & 0 & 0 \end{bmatrix} .$$

This matrix $\mathbf{H}$ has the eigenvalues -3, -3, $+3$ and $+3$. $\boldsymbol{\theta} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ is therefore a saddle-point.

A4.4: Relative overgeneralization **(5 points)**

In this question we will analyze relative overgeneralization at the example of a predator prey game, depicted in the figure below. Here two agents (Blue and Magenta) live in a one-dimensional fully observable world, in which they can move left (L) or right (R), but get blocked by trees and the stationary prey (red). If *both* agent stand next to the prey, they have each access to a special "catch" action (C). In this special state s the collaborative reward is given by the matrix below. If both agents select the C action together in this state, the episode ends (but it continues for any other action combination). In all other states the catch action is not available, the collaborative reward is 0 and the episode continues. Assume that both agents learn with independent Q-learning (IQL) and have discount factor $\gamma \in [0, 1)$.



(a) *[1 point]* Determine analytically the value of the optimal policy when the first agent is 3 steps, and the second agent 5 steps away from the prey.

(b) *[2 points]* Assume that both agents will get stuck (never return to s) once they leave s. Determine analytically the independent Q-value of agent 1 (Blue) in state s (i.e. $q^1(s, a^1; \boldsymbol{\pi})$ from slide 10 of Lecture 9), assuming that both agents uniformly explore (i.e. $\pi^i(a^i|s) = \frac{1}{3}, \forall a \in \mathcal{A}^i, \forall i \in \{1, 2\}$).

Hint: don't give up if the terms become a bit unwieldy. The solution contains (among others) surprisingly large multiples of 3 in ugly fractions. Just simplify as far as possible and keep going!

(c) *[2 points]* For a given reward $r > 0$ in your solution of (b), which punishments $p \in \mathbb{R}$ induce in expectation *relative overgeneralization*, i.e., for which range of p holds $q^1(s, C; \boldsymbol{\pi}) < q^1(s, L/R; \boldsymbol{\pi})$?

Solution For notational simplicity, we will use $q^1(s, a) \equiv q^1(s, a; \boldsymbol{\pi})$ in the following.

(a) The optimal policy lets both agents move to the prey. As soon as state s is reached (both agents have arrived), they will both execute C, which yields a reward of $+r$. The optimal value is therefore $V^* = \gamma^5 r$.

(b) If Blue selects C, the other agent can either select: C (end episode, outcome $+r$), R (leaves s, outcome $-p$) or L (stays in s, outcome $-p + \gamma V^\pi(s)$), where $V^\pi(s)$ is the joined state-value. As all 3 possibilities happen with equal probability,

$$q^1(s, C) = \tfrac{1}{3}\big(r - 2p + \gamma V^\pi(s)\big)\,.$$

If Blue selects L, it will leave s and never return. The other agent can select C now (outcome $-p$), but never again, yielding

$$q^1(s, L) = -\tfrac{1}{3}p\,.$$

If Blue selects R, it is blocked and may remain in s if the other agent does as well. Magenta's possible actions are: C (stay in s, outcome $-p + \gamma V^\pi(s)$), L (stay in s, outcome $\gamma V^\pi(s)$) and R (leave s, outcome 0). This yields

$$q^1(s, R) = \tfrac{1}{3}\big(-p + 2\gamma V^\pi(s)\big)\,.$$

Now we can determine $V^\pi(s)$:

$$V^\pi(s) = \mathbb{E}\big[q^1(s, a)\big|a \sim \pi^1(\cdot|s)\big] = \tfrac{1}{3}\big(q^1(s, C) + q^1(s, L) + q^1(s, R)\big) = \tfrac{1}{9}\big(r - 4p + 3\gamma V^\pi(s)\big)$$

$$\Leftrightarrow \quad (1 - \tfrac{1}{3}\gamma)V^\pi(s) = \tfrac{1}{9}(r - 4p) \quad \Leftrightarrow \quad V^\pi(s) = \tfrac{r-4p}{9-3\gamma}\,.$$

Substituted back into q^1, this yields:

$$q^1(s,a) \;=\; \frac{1}{27-9\gamma}\begin{cases}(9-2\gamma)r-(18-2\gamma)p & ,a=C\\ -(9-3\gamma)p & ,a=L\,,\\ 2\gamma r-(9+5\gamma)p & ,a=R\end{cases}$$ looks ugly, but what can you do ;).

(c) Relative overgeneralization (RO) occurs when the independent Q-value for a sub-optimal action is larger than the value of an optimal action. Based on the previous solution we can say that this can happen when $q^1(s,C) < q^1(s,L/R)$. For the first case we have:

$$q^1(s,C) < q^1(s,L) \quad\Leftrightarrow\quad (9-2\gamma)r-(18-2\gamma)p < -(9-3\gamma)p \quad\Leftrightarrow\quad \tfrac{9-2\gamma}{9+\gamma}r < p\,.$$

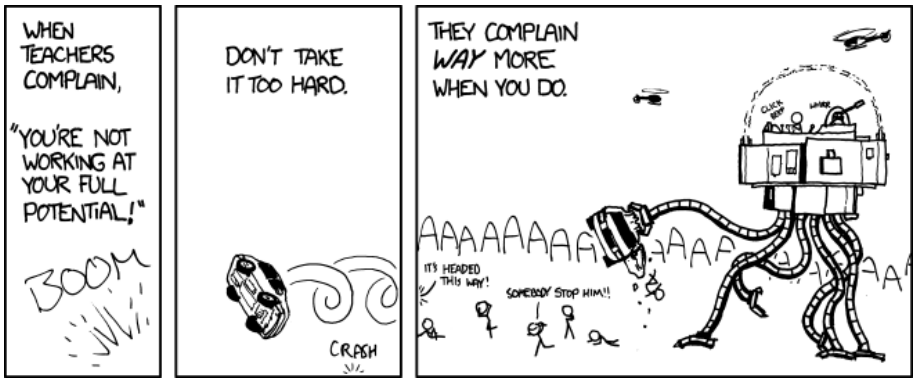
For the second case we have:

$$q^1(s,C) < q^1(s,R) \quad\Leftrightarrow\quad (9-2\gamma)r-(18-2\gamma)p < 2\gamma r-(9+5\gamma)p \quad\Leftrightarrow\quad \tfrac{9-4\gamma}{9-7\gamma}r < p\,.$$

This gives us two potential lower bounds on p that induce RO, which means that the smaller of the two is the answer we are looking for. The first threshold on p is the smaller one, which we can show for $\gamma \in [0,1)$ by

$$\tfrac{9-2\gamma}{9+\gamma} < \tfrac{9-4\gamma}{9-7\gamma} \quad\Leftrightarrow\quad \gamma < 3\,. \quad ✓$$

So agent 1 (and in extension agent 2) will exhibit relative overgeneralization when $p > \frac{9-2\gamma}{9+\gamma}r$. For small $\gamma \approx 0$, this corresponds to $p > r$, whereas for large $\gamma \approx 1$, it corresponds to $p > \frac{7}{10}r$. Note that if $p < \frac{7}{10}r$ no relative overgeneralization can occur!



https://xkcd.com/987

Total 17 points.