Multi-Agent Systems

RESIT Final Homework Assignment

MSc AI, VU

E.J. Pauwels

Version: February 10, 2021— **Deadline: Friday, 26 Feb 2021 (23h59)**

IMPORTANT

- This project is an **individual assignment.** So everyone should hand in their own copy.
- Both of the questions below require programming. In addition to the report (addressing the questions and discussing the results of your experiments), also provide the code on which your results are based (e.g. as Python notebooks). However, make sure that the report is self-contained, i.e. that it can be read and understood without the need to refer to the code. Store the pdf-report and the code in a zipped folder that you upload via canvas.
- This assignment will be **graded.** The max score is 4 and will count towards your final grade.
- Your **final grade (on 10)** will be computed as follows:

  Assignments 1 thru 5 (max 1) + Individual assignment (max 4) + Final exam (max 5)

1 Multi-Armed Bandits

1.1 Thompson Sampling for Single bandit

Consider a bandit that for each pull of the arm, produces a binary reward: $r = 1$ (with probability p) or $r = 0$ (with probability $1 - p$). We model our uncertainty about the actual (but unknown) value p using a beta-distribution (cf. `https://en.wikipedia.org/wiki/Beta_distribution`). This is a probability distribution on the interval $[0, 1]$ which depends on two parameters: $\alpha, \beta \geq 1$. The explicit distribution is given by (for α, β integers!):

$$B(x;\, \alpha, \beta) = \frac{(\alpha + \beta - 1)!}{(\alpha - 1)!\,(\beta - 1)!} x^{\alpha - 1}(1 - x)^{\beta - 1} \qquad \text{(for } 0 \leq x \leq 1\text{)}.$$

The parameters α and β determine the shape of the distribution:

- If $\alpha = \beta = 1$ then we have the uniform distribution;
- If $\alpha = \beta$ the distribution is symmetric about $x = 1/2$.
- If $\alpha > \beta$ the density is right-leaning (i.e. concentrated in the neighbourhood of 1). In fact, one can compute the mean explicitly:

$$X \sim B(x;\, \alpha, \beta) \quad \Longrightarrow \quad EX = \frac{\alpha}{\alpha + \beta}.$$

- Larger values of α and β produce a more peaked distribution. This follows from the formula for the variance:

$$X \sim B(x;\, \alpha, \beta) \quad \Longrightarrow \quad Var(X) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$

Thompson update rule Assume that we don't know the success probability p for the bandit. The Thompson update rule for a single bandit proceeds as follows:

- Initialise $\alpha = \beta = 1$ (resulting in a uniform distribution, indicating that all possible values for p are equally likely). Now repeat the following loop:
    1. Sample from the bandit and get reward r (either 1 or 0);
    2. Update the values for α and β as follows:

$$\alpha \leftarrow \alpha + r \qquad\qquad \beta \leftarrow \beta + (1 - r)$$

Questions

1. Make several plots of the Beta-density to illustrate the properties (dependence on the parameters) outlined above;
2. Implement the Thompson update rule and show experimentally that the Beta-density increasingly peaks at the correct value for p. Plot both the evolution of the mean and variance over (iteration)time.

1.2 Thompson sampling for Two Bandit Problem

Suppose we have a 2-bandit problem. The first bandit delivers a reward $r = 1$ with (unknown!) probability p_1 (and hence $r = 0$ with probability $1 - p_1$), while the second one does the same with (unknown!) probability p_2. For each bandit ($k = 1, 2$), the uncertainty about the corresponding p_k is modelled using a Beta-distribution with coefficients α_k, β_k. Thompson sampling now tries to identify the bandit that delivers the maximal output and proceeds as follows:

- Initialise all parameters to 1: $\alpha_k = 1 = \beta_k$;
  Now repeat the following loop:
    1. Sample a value u_k from each of the two Beta-distributions:

$$u_k \sim B(x; \alpha_k, \beta_k)$$

2. Determine the max: $k_m = \arg\max\{u_1, u_2\}$
3. Sample the corresponding bandit and get reward r (either 1 or 0);
4. Update the corresponding parameters: α_{k_m} and β_{k_m} as follows:

$$\alpha_{k_m} \leftarrow \alpha_{k_m} + r \qquad \text{and} \qquad \beta_{k_m} \leftarrow \beta_{k_m} + (1 - r)$$

Questions

1. Write code to implement Thompson sampling for the above scenario;

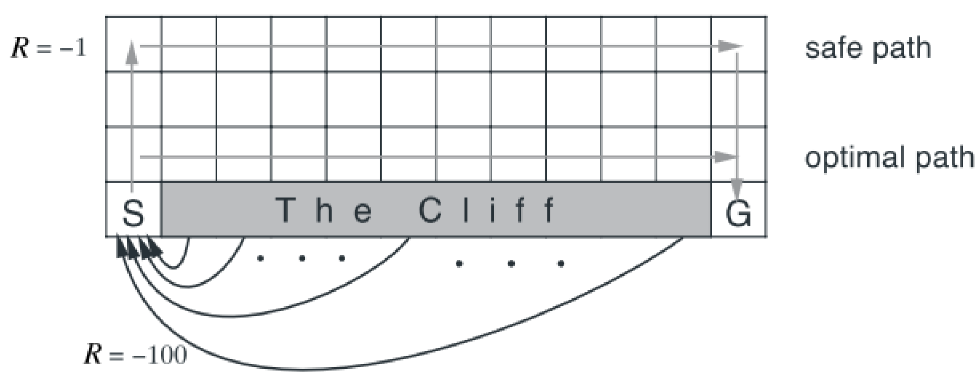
2. Perform numerical experiments in which you compare Thompson sampling with the UCB and ϵ-greedy approach. Define some performance measures (e.g. cumulative reward, percentage best arm, or total regret) and discuss the performance of each algorithm with respect to these measures.

2 Reinforcement Learning: Cliff Walking

Consider the cliff-walking example (Sutton & Barto, ex. 6.6. p.108). Assume that the grid has 10 columns and 5 rows (above or in addition to the cliff). This is a standard undiscounted, episodic task, with start (S) and goal)G) states, and the usual actions causing movement up, down, right, and left. Reward is -1 on all transitions except:

- the transition to the terminal goal state (G) which has an associated reward of $+20$;

- transitions into the region marked *The Cliff*. Stepping into this region incurs a "reward" of -100 and also terminates the episode.



Questions

1. Use both SARSA and Q-Learning to construct an appropriate policy. Do you observe the difference between the SARSA and Q-learning policies mentioned in the text (safe versus optimal path)? Discuss.

2. Try different values for ϵ (parameter for ϵ-greedy policy). How does the value of ϵ influence the result? Discuss.