Foundations of Machine Learning II
Project: Bandits and Wumpus *

Guillaume Charpiat, Corentin Tallec & Gaétan Marceau Caron

January 8, 2017

Description of the problem The Wumpus world is a well-known toy problem in artificial intelligence popularized by the reference book of Russell and Norvig, 2003. The game consists in a grid world where an agent (Einstein) is looking for a treasure (scientific papers) while avoiding the deadly Wumpus (red monster) and some traps (black holes). We assume that the time is discretized and that the agent can take only one action per time step. The agent can explore the grid by moving in the four cardinal directions. Also, it has a flashlight with a limited number of power units that can be used to kill the Wumpus. For the project, the Wumpus world will be our benchmark problem for studying reinforcement learning (RL) algorithms. Moreover, these algorithms are general enough to be applied to different problems (environment).

In the RL framework, we define an environment, which specifies the information to be used by the agent to take some actions. For the Wumpus world, the environment is partially observable since the locations of the black holes, the Wumpus and the scientific papers are unknown to the agent. However, the environment provides some signals to the agent:

- if the agent is adjacent to the Wumpus, it receives a *smell* signal
- if the agent is adjacent to the hole, it receives a *breeze* signal
- the location of the agent is deterministically determined by the initial position and the actions
- the number of power units is known

Also, the environment provides a reward signal, which represents the immediate utility to be in a giving state. The table 1) gives the rewards associated to the events of the Wumpus world. Notice that, except for the default event, the four others lead to a terminal state.

*https://www.lri.fr/~gcharpia/machinelearningcourse/

Question 1 Why is it interesting to give a negative reward for the default event ?

However, since the Wumpus can move around the grid (it could be an agent too!), we must synchronize the updates. We assume the following:

- if the agent is adjacent to the Wumpus and if they move towards each other, there is a Wumpus event.
- if the agent is adjacent to the Wumpus and if the agent flashes in the direction of the Wumpus while the Wumpus is moving in the direction of the agent, there is a killing Wumpus event.
- the other cases are easily determined by the positions of the objects.

The agent can only interact with the environment by choosing one action per time step in the following set:

$$\begin{aligned}\mathcal{A} = (&\text{Up, Down, Left, Right,}\\ &\text{FlashUp, FlashDown, FlashLeft, FlashRight) (Action space)}\end{aligned} \tag{1}$$

Once the environment is fully specified, the goal of RL algorithms is to learn a policy that maps each state to an action. We want this policy to maximize the expected cumulative reward obtained during an episode. While the actions are specified by the environment, the state representation is internal to the agent. Indeed, we can define a mapping from the sensors of the environment to a feature space. In the Wumpus world, we can map directly the sensors to the following state space:

$$\mathcal{S} = (X, Y, B, S, F) \quad \text{(State space)} \tag{2}$$

where X, Y are integers for the position of the agent in the grid, B, S are binary variables associated to the breeze and smell signals, respectively, and F is an integer for the number of remaining power units for the flashlight. Notice that this choice is not unique but encompasses the signals returned by the environment.

Event	Reward
treasure	100
killing Wumpus	10
hole	-10
Wumpus	-10
default	-1

Table 1: Reward table

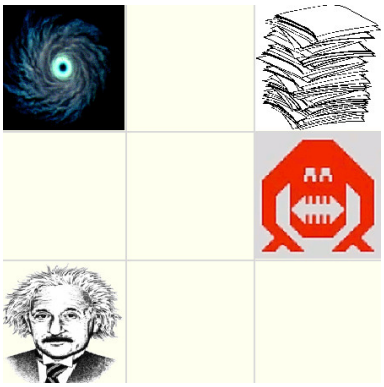

Figure 1: An instance of the Wumpus World

Description of the project The goal of the first part of the project is to learn a policy with a contextual bandit algorithm. To familiarize yourself with the code, start by running the random Agent and interpret the terminal outputs. You should see the time step number, the previous state, the chosen action, the next state and the cumulative reward.[1]

Question 2 What do you observe in terms of cumulative reward for the random agent ?

Question 3 Propose and implement a new agent and compare the result with the cumulative reward of the random agent. Discuss about the robustness of your agent with the possible behaviors of the Wumpus.

During the project, we will study RL algorithms for learning the optimal policy that maps each state to an action. The first algorithm is based on *contextual bandit algorithms*, which make very strong assumptions about the environment. Indeed, for a given state s, we are interested in the expected reward $\mathbb{E}_{p(r|s,a)}[r]$ where $p(r|s,a)$ is the probability to obtain a reward r given that the agent takes action a in state s. However, $p(r|s,a)$ is unknown to the agent, but we can still approximate it with the empirical average. Indeed, each time the agent is in state s, it chooses an action a, record the observed reward (history) and update the average $Q(s,a)$. In the contextual bandit framework, the state s gives the context and the actions are associated to the arms. Finally, we still need to specify how to choose the action with the given history. The most simple approach is to the choose the action with the highest average, which is called the *greedy policy*.

[1] You can also print only the cumulative reward at the end of the episode by fixing `self.LOGGER_TIME_STEP` to false.

Question 4 What is the main drawback of the greedy policy ?

An inherent difficulty to the RL problem concerns the exploration-exploitation trade-off. The greedy policy performs exploitation only and so, the ε-greedy policy has been proposed to promote exploration. This policy selects the best action with probability $1 - \varepsilon$ and a random action with probability ε. Finally, we can prefer to explore more on promising actions than very bad action. The *softmax action selection* policy defines a Gibbs distribution over the actions:

$$p(a|s) = \frac{\exp(Q(a,s)/\tau)}{\sum_{a'} \exp(Q(a',s)/\tau)} \tag{3}$$

where $Q(a,s)$ is the average reward of action a given the state s and τ is an hyper-parameter. If $\tau \to 0$, then the softmax policy tends to the greedy policy and if $\tau \to \infty$ then it tends to the random policy. These three policies are described in details in the RL reference book of Sutton and Barto, 2012.

Question 5 Implement the optimistic ε-greedy, softmax and UCB policies, and compare their cumulative rewards. Consider different state spaces such as $\mathcal{S} = (B, S, F)$, $\mathcal{S} = (A_{previous}, B, S, F)$ and $\mathcal{S} = (X, Y, B, S, F)$ and discuss about the relation between the grid size and the Wumpus behavior, and the number of states. In order to obtain a statistic on the performances, you can run the training algorithm many times and plot all the learning curves or simply average over the past 100 episode for a single run.

Question 6 (Bonus) Discuss about the limitations of the contextual bandit algorithm for the RL problem and propose an algorithm for solving these issues

References

Russell, Stuart J. and Peter Norvig (2003). *Artificial Intelligence: A Modern Approach*. 2nd ed. Pearson Education. ISBN: 0137903952.

Sutton, Richard S. and Andrew G. Barto (2012). *Introduction to Reinforcement Learning*. 2nd ed. Cambridge, MA, USA: MIT Press. ISBN: 0262193981.