CS4200/CS5200, On-line Machine Learning

Class 9: Reinforcement Learning

Yuri Kalnishkan

Department of Computer Science
Royal Holloway, University of London

2018/19

Class Outline

1. Motivation and Preliminaries

References

[TM] T. M. Mitchell, “Machine Learning”, McGraw-Hill, 1998, Chapter 13.

[SB] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction”, 2nd edition, The MIT Press, 2018

[CS] C. Szepesvári “Algorithms for Reinforcement Learning”, Morgan & Claypool, 2010

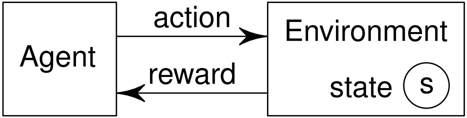
[JT] J. N. Tsitsiklis, On the Convergence of Optimistic Policy Iteration, JMLR 3 (2002) 59-72

[WD] C. J. C. H. Watkins and P. Dayan, Technical Note: Q-Learning, Machine Learning, 8, 279-292 (1992)

1. Motivation and Preliminaries

Principal Setup

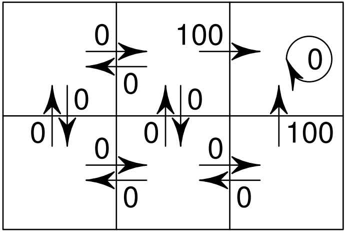
- a learning agent interacts with an environment
  — the agent takes an action
  — the environment gives a reward and changes its state



- we make the Markov assumption: all activity (choice of the action, reward, state to go to) depends on the current state rather then the whole history
  — given the current state, the future is independent of the past
- the current state is visible to the agent

Gridworlds

- gridworlds are toy examples used to illustrate principles of reinforcement learning



(after [TM], Fig. 13.2)

- the agent can move from a square to a neighbouring square; the rewards are shown on the diagram
- here the top right corner is a goal state (or terminal state); further moves from it are neither possible nor needed

Deterministic Environment

- this example is a deterministic environment
— the reward and the state we move into are functions of the current state and action taken
- let S be the set of all states and A be the set of all actions; if the environment is deterministic, then
— reward $r = \text{Reward}(s, a)$, where $\text{Reward} : S \times A \to R$ is a function
— state we move into $s = \text{State}(s, a)$, where $\text{State} : S \times A$ is a function
- the environment can be described by two functions

Stochastic Environment

- suppose we are controlling a robot in a real-life situation
- there is uncertainty as to what happens after an action — the reward we get and the state we move into after taking an action a in a state s can be modelled by random variables $\mathrm{Reward}_{s,a}$ and $\mathrm{State}_{s,a}$
- the environment may be described by a collection of distributions on $R \times S$ — there is one distribution for each pair (s, a)
- this is called a Markov Decision Process (MDP)
- we assume the MDP is stationary, i.e., if we return to state s and choose the same action a, we are faced with the same possibilities