Reinforcement Learning
Lecture 1

Gillian Hayes

8th January 2007

School of informatics

Admin 1

Lecturer: Gillian Hayes, IPAB, School of Informatics

Email: gmh@inf.ed.ac.uk

Office: JCMB room 2107C, ext. 513440

Course Activities:

- Lectures: Monday 12:10, Thursday 12:10, JCMB 3317
  Weeks 1–10 (week 10 and maybe 11 for overflow/revision)
  Revision tutorial in Easter break

- Assessment: Homework/Exam 20%/80%
  Two exercises worth 10% each
  Out Thursday weeks 3 and 6
  Due Monday 10pm week 7, Friday 10pm week 10

Admin 2

- Reading: Set book: R. Sutton and A. Barto, *Reinforcement Learning*, MIT Press, 1998, £35.95 on Amazon
  Online link on webpage, MIT CogNet
- Webpage: For notes, slides, information.
  `www.informatics.ed.ac.uk/teaching/courses/rl/`
- Class Rep
- Registration: Everyone must register using the Course Registration Form reachable via `www.informatics.ed.ac.uk/teaching`
- Background: Everyone done LFD? Maths, Matlab

Syllabus

- Reinforcement learning
  - The RL problem
  - Bandit problems
  - Dynamic programming
  - Monte Carlo methods
  - Q-learning, eligibility traces
  - Temporal difference learning
  - Environment modelling
  - Function approximation for generalisation
  - Actor-critic, applications in robotics, etc.
  - Collective Intelligence (COIN)
  - Planning

If time permits we may also cover some of:

- Unsupervised, self-organising networks
- Constructive methods – nets that grow
- Radial basis functions and other local functions
- Using classifiers
- Pre/postprocessing of data
- Evaluating performance – how well did it learn?

Learning from Interaction

– with environment

– to achieve some goal

- Baby playing. No teacher. Sensorimotor connection to environment.
  - Cause – effect
  - Action – consequences
  - How to achieve goals

- Learning to drive car, hold conversation, etc.
  - Environment's response affects our subsequent actions
  - We find out the effects of our actions later

Reinforcement Learning

Learning a mapping from situations to actions in order to maximise a scalar reward/reinforcement signal

HOW?

- Try out actions to learn which produces highest reward – *trial-and-error search*
- Actions affect immediate reward + next situation + all subsequent rewards – *delayed effects, delayed reward*

Situations, Actions, Goals

Sense situations, choose actions TO achieve goals

Environment uncertain

Exploration/Exploitation Tradeoff

High rewards from trying previously-well-rewarded actions – **EXPLOITATION**

BUT

Which actions are best? Must try ones not tried before – **EXPLORATION**

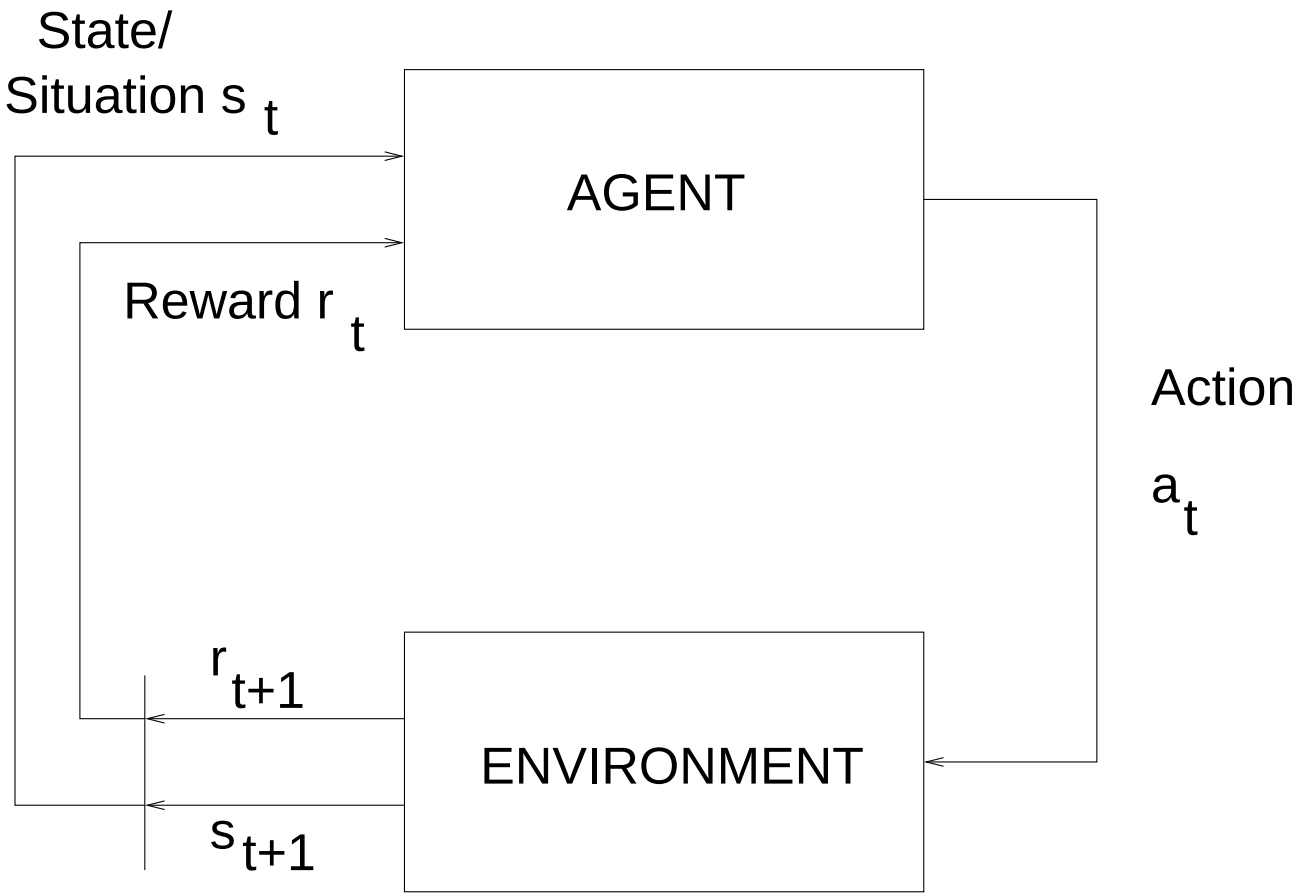
MUST DO BOTH

Especially if task stochastic, try each action many times per situations to get reliable estimate of reward.

Gradually prefer those actions that prove to lead to high reward.

(Doesn't arise in supervised learning – where you get told the correct answer and you have to try to produce an answer more like it the next time)

Framework


State/
Situation s t
AGENT
Reward r t
Action
a t
r t+1
ENVIRONMENT
s t+1

Agent in situation/state s_t chooses action a_t
World changes to situation/state s_{t+1}
Agent perceives situation s_{t+1} and gets reward r_{t+1}

Telling the agent what to do is its

POLICY $\pi_t(s, a) = Pr\{a_t = a | s_t = s\}$

Given the situation at time t is s, the policy gives the probability the agent's action will be a.

For example: $\pi_t(s, \text{goforward}) = 0.5$, $\pi_t(s, \text{gobackward}) = 0.5$.

Reinforcement learning $\Rightarrow$ Get/find/learn the policy