BLG456E
Robotics
Intro to Reinforcement Learning

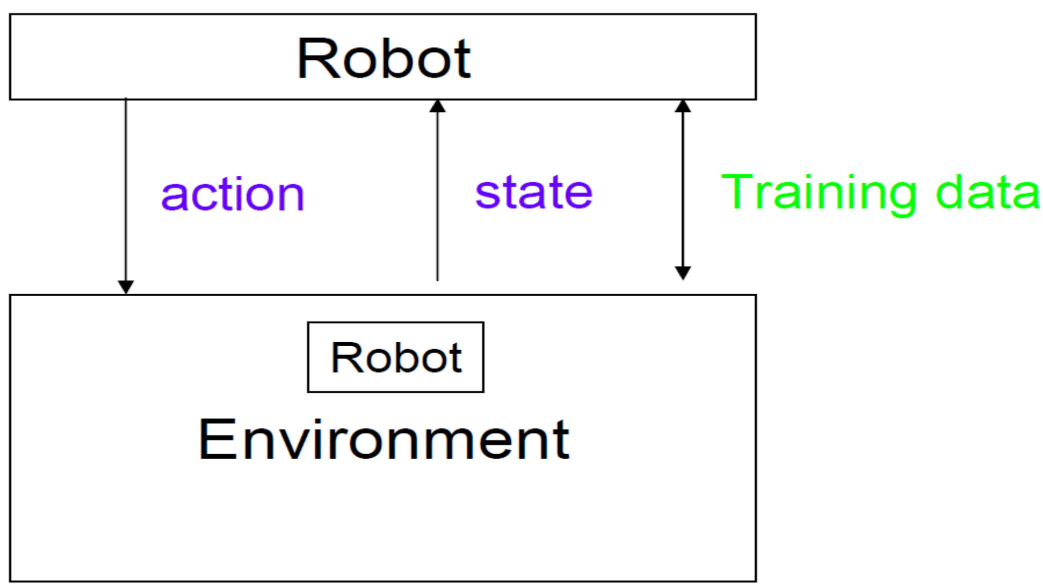
Lecture Contents:
- Kinds of learning.
- Introducing time.
- Value functions.
- Bootstrap learning of value functions.
- Exploration vs. exploitation.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Coordination:	http://ninova.itu.edu.tr/Ders/4709

Some slides taken/adapted from Dr. Sanem Sarıel-Talay

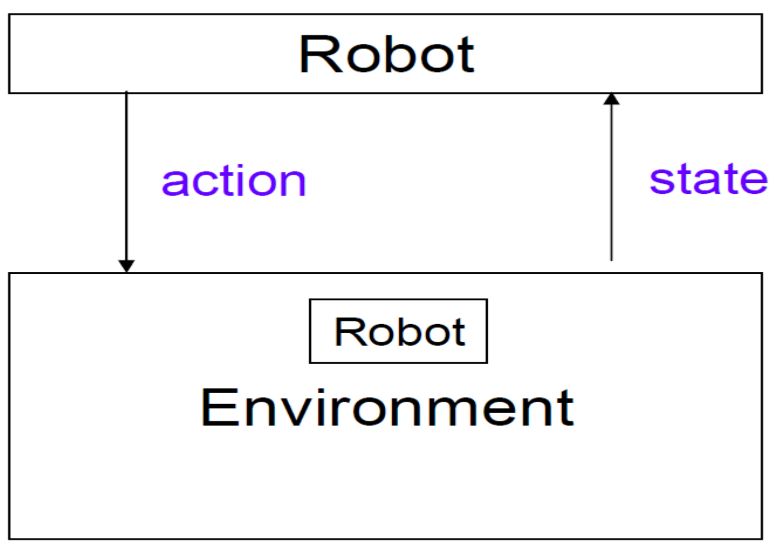
Supervised learning

- Pairs of state-action (s, a) given as training data.
- robot learns appropriate action for a state.



Unsupervised learning

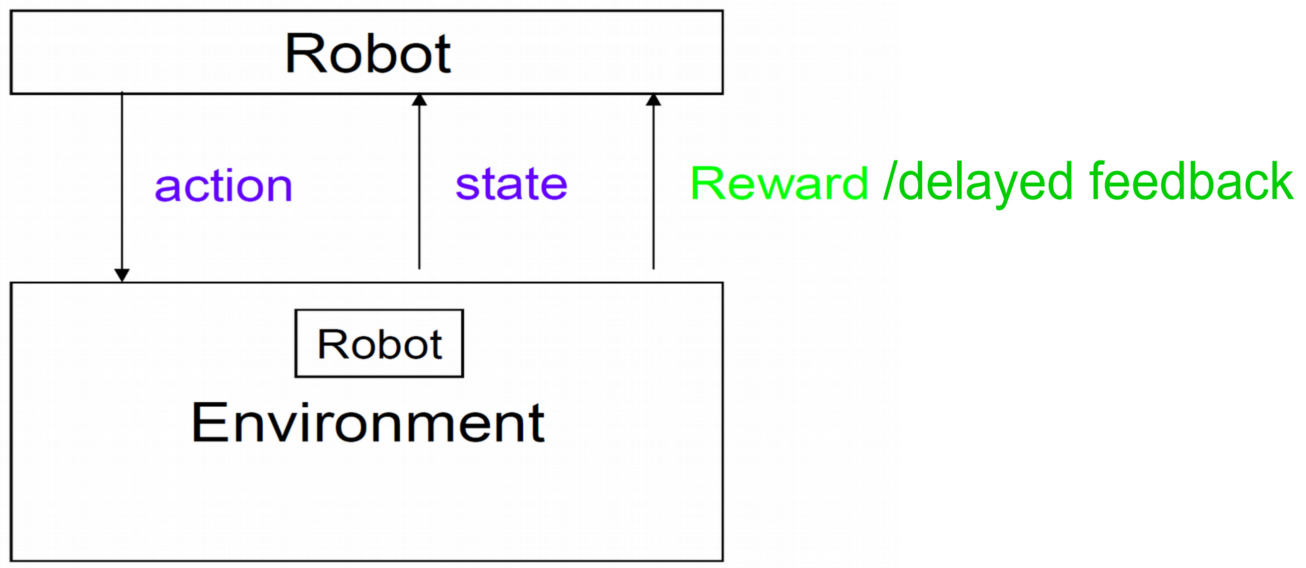
- The robot gets input data x_1, x_2, … x_n
- Construct a representation of x for reasoning, decision making, classification, …



e.g. clustering

Learning from time-delayed/inconsistent feedback.

- Delayed feedback on actions.
- Feedback may be occasional or probabilistic.



Credit assignment problem

- **Problem:** Feedback can come long after actions.
  - e.g. found food / hit wall.
  - usually: reward / punishment (reinforcement learning).

→ Actions are not “labelled” by supervisor.

Action Traces

- Solution to credit assignment problem.
- Remember *action trace*, try to recreate rewarding traces.

State/action trace: $s_o, a_0, r_0, s_1, a_1, r_1, \ldots, s_N, a_N, r_N$

s_i - state at time i

a_i - action then taken

r_i - reward from taking that action

Learning Value Functions

Learn a function for the **value** of an action in a state.

$$Q(s, a)$$

Q is the value of action a at state s

Could update the function from backtraces:

Here, the "hat" means "an estimate of".

$$\hat{Q}(s_t, a_t) \sim \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

s_t is state, a_t action at time t.

r is future reward/feedback value, γ discount value.

Bootstrapping approach to learning Value function

Bootstrapping:

- If I know the *value* of state/action at time $t+1$, I can update the value of state/action at time t.
- I don't need to keep an action trace.
- Incorporating previous experience too.

$$\hat{Q}(s_t, a_t) \leftarrow (1-\epsilon)\hat{Q}(s_t, a_t) + \epsilon\left(r_t + \gamma \hat{Q}(s_{t+1}, a_{t+1})\right)$$

Q is state-action value, s_t state, a_t action at time t.
r is reward/feedback value, γ discount value, ϵ learning rate.

(update expression assumes that we have a lookup table for the value of action a at state s, i.e. discrete states - but this can be generalised)

Using a Q function

- After learning, need to choose an action.
- State-dependent action choice is called a **policy**.

Policy:

$$\pi(s) \rightarrow a$$

- Exploitation policy:

$$\pi_{exploit}(s) = \operatorname{argmax}_a Q(s, a)$$

- Note: State s could be *world state* or *sensory state*

http://www.scholarpedia.org/article/Temporal_difference_learning

Policy during learning

- On-policy learning: I assume I will act as I am acting while learning.
  - ***SARSA** (State-Action-Reward-State-Action) **learning***.
  - Future reward dependent on the current policy.

  Off-policy learning: I assume I will act differently later.
  - ***Q learning.***
  - Future reward not fully dependent on current policy.

http://artint.info/html/ArtInt_268.html

Exploration vs. exploitation

What choices should a learner make in order to learn better?

- This is "**active learning**".

Exploitation vs. exploration:

- **Exploitation**: Use learnt skills to maximise reward.
- **Exploration**: Continue to act to maximise learning.

$$\pi_{\text{explore}\,\epsilon}(s) = \begin{cases} \text{argmax}_a Q(s,a) \text{ with probability } 1-\epsilon \\ \text{random } a \text{ with probability } \epsilon \end{cases}$$

SARSA algorithm

for all s_t, a_t:

$\hat{Q}(s_t, a_t) = 0$

$s \Leftarrow$ current state

$a \Leftarrow \operatorname{argmax}_{a^*} \hat{Q}(s, a^*)$

Loop:

$s', r \Leftarrow$ robot do (a)

$a \Leftarrow \operatorname{argmax}_{a^*} \hat{Q}(s', a^*)$

$\hat{Q}(s, a) \leftarrow (1-\epsilon)\hat{Q}(s, a) + \epsilon(r + \gamma \hat{Q}(s', a'))$

$s \Leftarrow s'$

$a \Leftarrow a'$

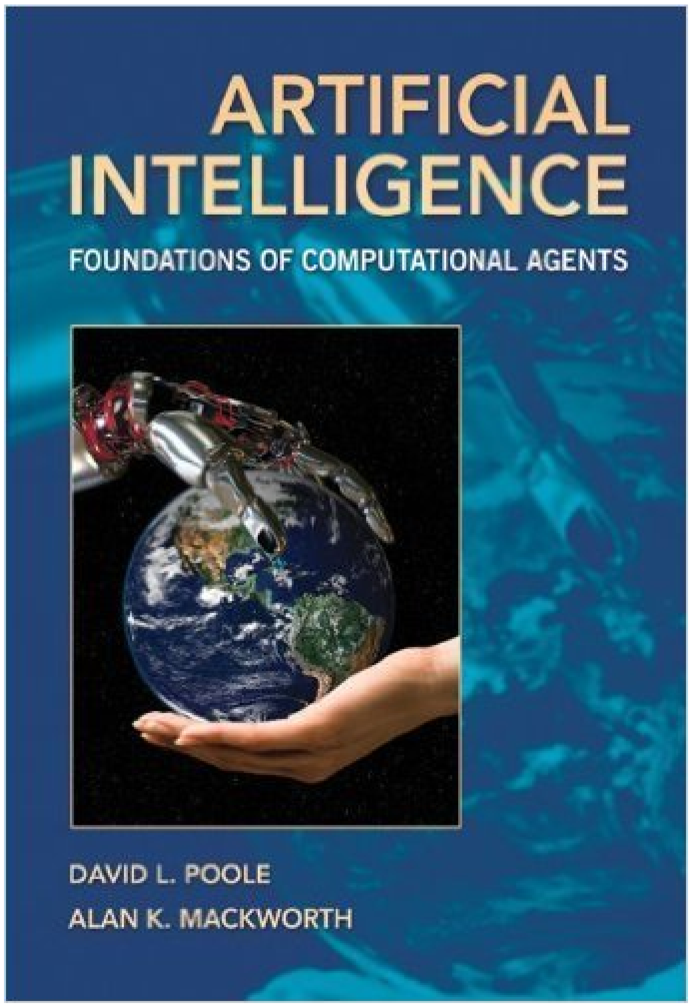
Other kinds of learning

- Evolutionary algorithms.
- Object classification / recognition.
- Object appearance models.
- Object/robot motion model learning.
- Planner learning.
- Learning learners.
- Optimisation.
- etc.

Robots that learnt to classify objects by weight & appearance:
http://www.youtube.com/watch?v=ckwsvmf3slU

Robots learning to walk during development:
http://www.youtube.com/watch?v=ckwsvmf3sl
U

Readings I



Poole & Mackworth (2010).
Artificial Intelligence: Foundations of Computational Agents:

Available from
http://artint.info/html/ArtInt_262.html
http://divit.library.itu.edu.tr/record=b1554963

Chapter 11.3:
Reinforcement Learning