BLG456E
Robotics
Intro to Reactive Robot Learning

<u>Lecture Contents:</u>

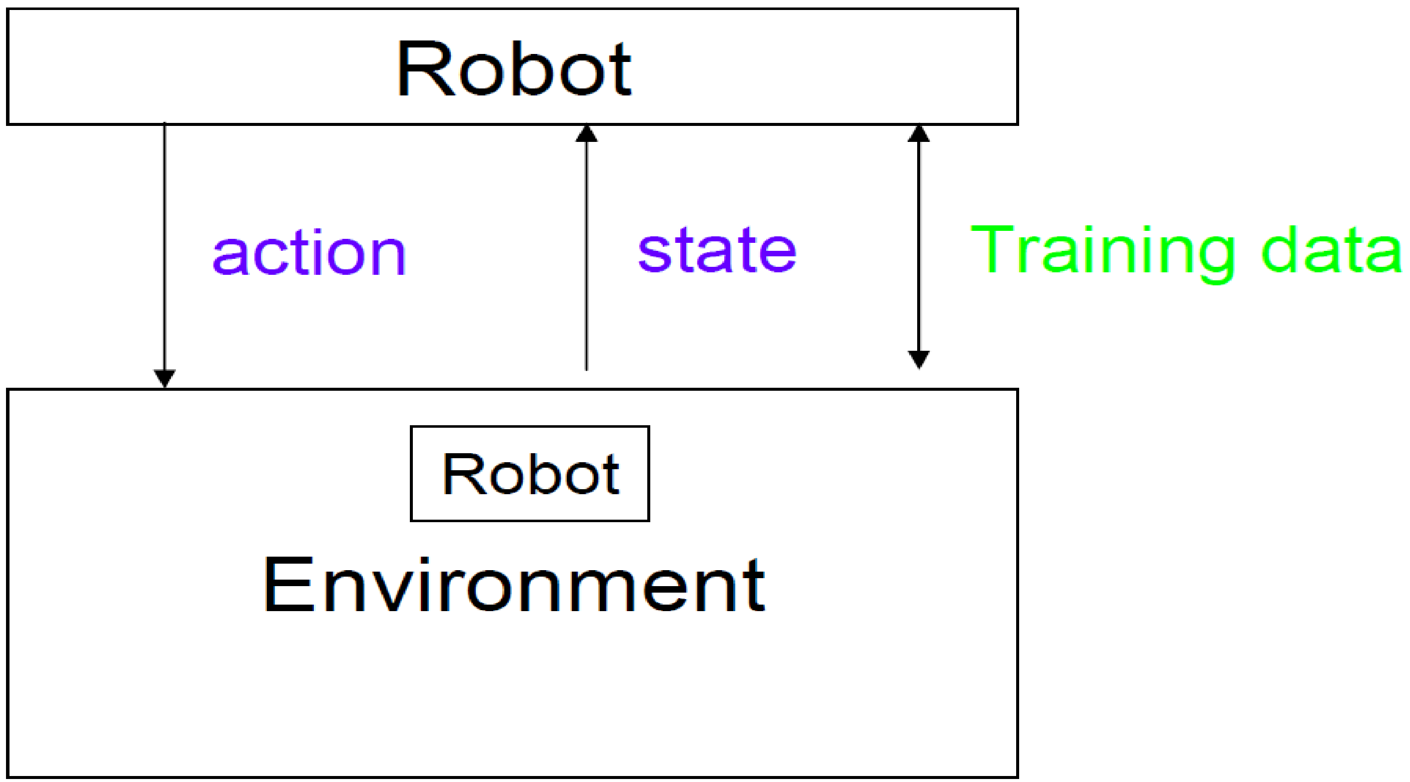
- Kinds of learning.
- Introducing time.
- Value functions.
- Bootstrap learning of value functions.
- Exploration vs. exploitation.

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

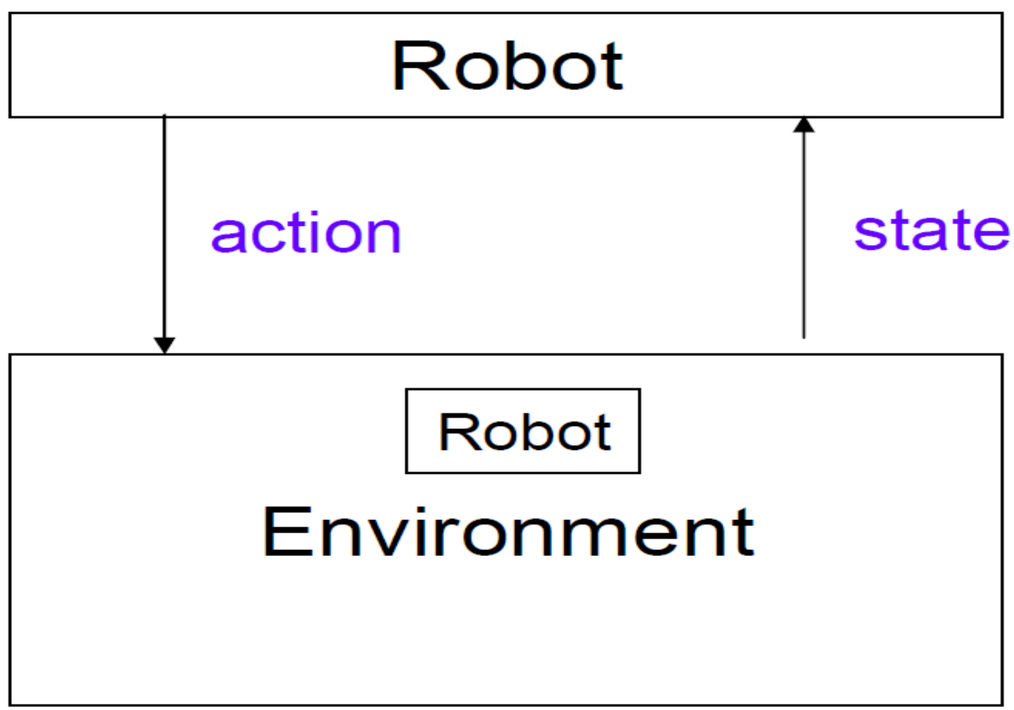
Supervised learning

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



Unsupervised learning

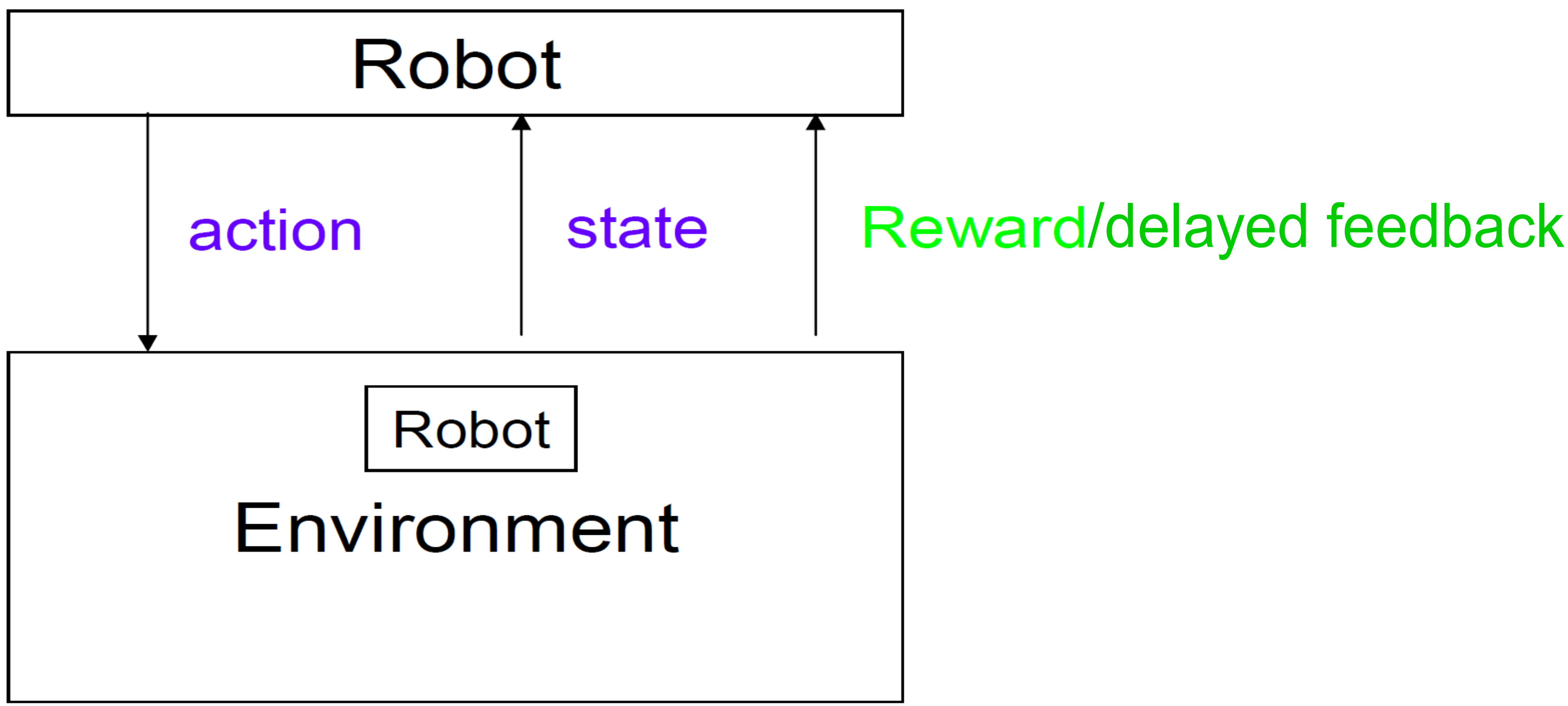
- The robot gets input data $x_1, x_2, \ldots x_n$
- Construct a representation of x for reasoning, decision making, classification, …



e.g. clustering

Learning from time-delayed/inconsistent feedback.

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



Credit assignment problem

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

→ Actions are not "labelled" by supervisor.

- **Naive solution**: 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

An alternative:

learn a function for the **value** of a state or action.

$$Q(s, a)$$

Q is the value of action a at state s

Could update the function from a backtrace:

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

ϵ is action error, s_t 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 $t+1$ then I can update the value of state at time t.
- I don't need to keep an action trace.
- I am using previous experience too.

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

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)

Notes about reinforcement learning

- I just showed *SARSA (State-Action-Reward-State-Action)*.
  - Future reward dependent on the current *policy*.
    - *On-policy learning: I assume I will act as I am acting while learning.*
  - Policy = robot's current behaviour.

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

- How to act after learning:
  - One approach: choose action with greatest Q value.

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

    - *Off-policy learning.*

- State s could be *world state* or *sensory state*.

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

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.

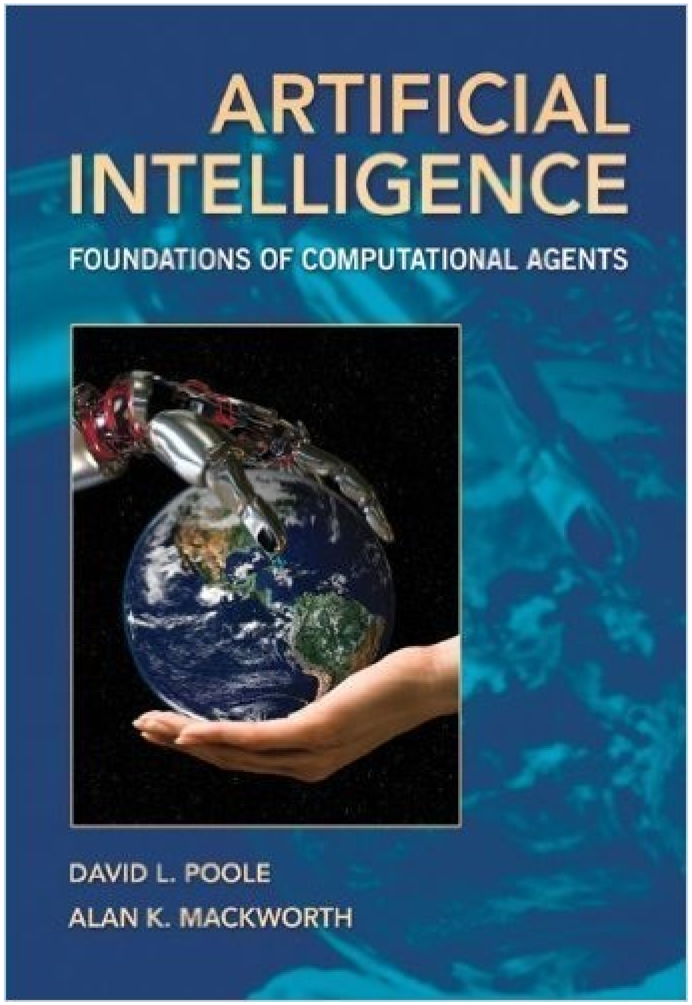
Other kinds of learning

- Evolutionary development.
- 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=ckwsvmf3slU

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