Reinforcement Learning

Environments

- Fully-observable vs partially-observable
- Single agent vs multiple agents
- Deterministic vs stochastic
- Episodic vs sequential
- Static or dynamic
- Discrete or continuous

What is reinforcement learning?

- Three machine learning paradigms:
  - Supervised learning
  - Unsupervised learning (overlaps w/ data mining)
  - Reinforcement learning
- In reinforcement learning, the agent receives incremental pieces of feedback, called rewards, that it uses to judge whether it is acting correctly or not.

Examples of real-life RL

- Learning to play chess.
- Animals learning to walk.
- Driving to school or work in the morning.

- **Key idea**: Most RL tasks are *episodic*, meaning they repeat many times.
  - So unlike in other AI problems where you have one shot to get it right, in RL, it's OK to take time to try different things to see what's best.

Episodes, exploration, and exploitation

RL problems

- Every RL problem is structured similarly.
- We have an ***environment***, which consists of a set of ***states***, and ***actions*** that can be taken in various states.
  - Environment is often stochastic (there is an element of chance).
- Our RL agent wishes to learn a ***policy***, π, a function that maps states to actions.

What is the goal in RL?

- In other AI problems, the "goal" is to get to a certain state. Not in RL!
- A RL environment gives feedback every time the agent takes an action. This is called a reward.
  - Rewards are usually numbers.
  - Goal: Agent wants to maximize the amount of reward it gets over time.
  - Critical point: Rewards are given by the environment, not the agent.

Mathematics of rewards

- Assume our rewards are r_0, r_1, r_2, …
- What expression represents our total rewards?
- How do we maximize this? Is this a good idea?