Imitation Learning

Learning from demonstrations

- RL = learning by trail & error without a teacher
- What if there is a teacher/expert?
- Basic setting: data = trajectories generated by π^* (or just a reasonably good policy)
- Break into $\{(s, a)\}$ pairs and apply supervised learning?
  - in particular, a multi-class classification problem (s is feature, a is a multi-class label)
  - also known as “behavior cloning”

Example: Super Mario
Expert data

credit: Ross et al, taken from Hal Daume's slides

Example: Super Mario
Behavior Cloning

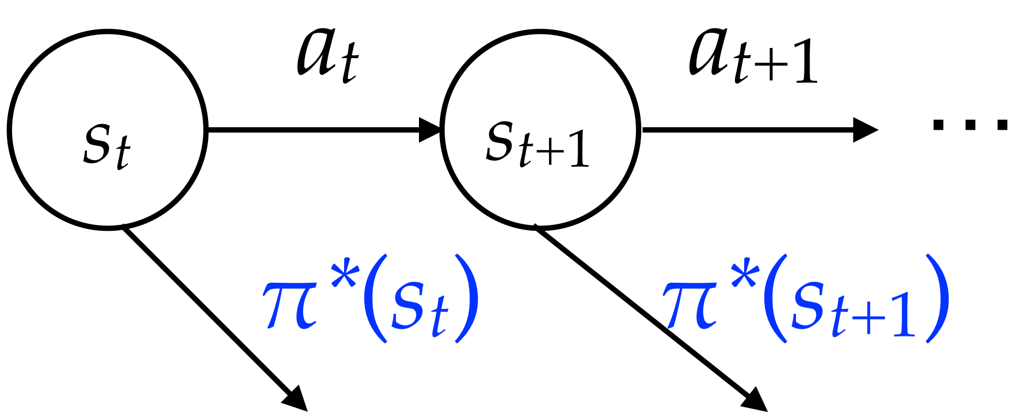
credit: Ross et al, taken from Hal Daume’s slides

What's going wrong?

- In the ideal world, if we can perfectly mimic expert's every action, then nothing goes wrong
- Reality: our imitation is imperfect, and we run into situations (states) that expert never encounters
  - No data about how to behave in this case
  - Result in poor performance
- How to handle this issue?
  - Intuition: let the expert tell us what to do in those new situations!

Interactive Imitation Learning

- Protocol:
  - Initialize the agent by behavior cloning
  - Generate new trajectories using the **learned policy**
  - Ask expert to provide demonstrations (e.g., optimal action) for the states in the new trajectories
  - Update the policy with the union of old and new data, and iterate
  - Representative algorithm: DAgger



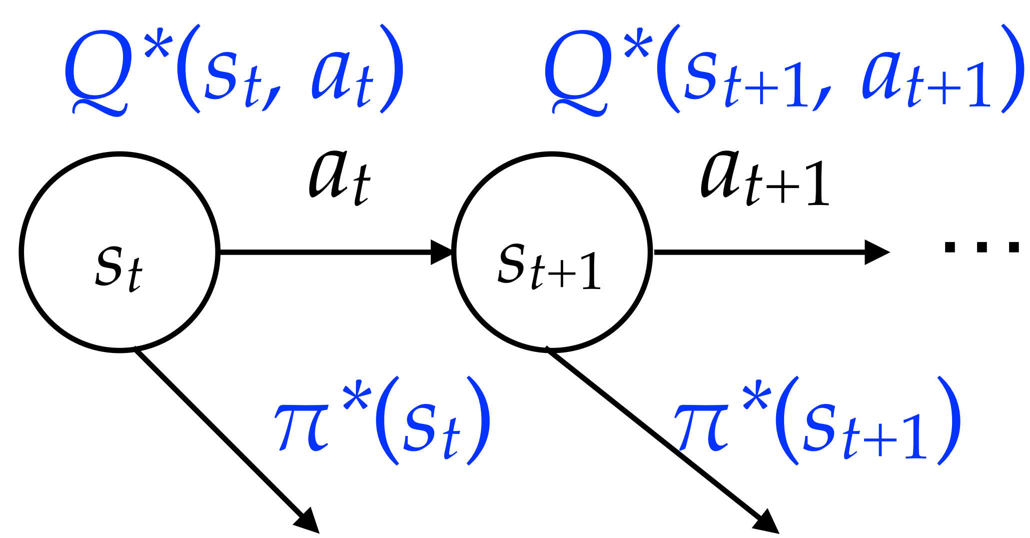
- Note: this requires a stronger (and perhaps less natural) expert. The expert has to be available when you run the algorithm.

Example: Super Mario
DAgger

credit: Ross et al, taken from Hal Daume's slides

Issue with learning from actions

- What if learner cannot represent π^* with its function approximation (but can represent some other good policies)
    - Error will be large regardless of what you do, if expert only provides action feedback
- Ask the expert more!
- One solution: ask for Q* value



- How to obtain Q* (if the expert does not know math)?
    - Let expert take over and finish the trajectory
    - The random return is a MC estimate of Q*
    - Learner needs to do some (simple) exploration
- Solve a "cost-sensitive classification" problem $\arg\max_{\pi \in \Pi} \mathbb{E}[Q^\star(s, \pi(s))]$

Lessons from interactive imitation learning

- Distribution matters (a lot!) in RL
  - The distribution of states you train on may be different from the distribution of states you “test” on (i.e., the distribution induced by the learned policy)
  - Can cause degenerate performance if not taken care of
- Leverage side information available in practice
  - Side information = structural knowledge, feedback signals, etc
  - In most cases you don’t care about solving RL (this is my job); you care about solving the problem
  - Think about what side information you can leverage