CS 295: Optimal Control and Reinforcement Learning

Winter 2020

Lecture 13: Exploration

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine

Today's lecture

- Sparse and dense rewards
- Sandbox for exploration vs. exploitation: Multi-Armed Bandits (MAB)
- Count-based exploration
- Thompson sampling

Relation between RL and IL

- Why is RL so much harder than IL?
  - IL: $\pi_T(a|s)$ indicates a good action to take in s
  - RL: $r(s, a)$ does not indicate a good action, $Q^*(s, a)$ does but it's nonlocal
- But didn't we see an equivalence between RL and IL?
  - Isn't $\nabla_\theta \mathbb{E}[\log \pi_\theta(a|s)]$ in IL like $\nabla_\theta \mathbb{E}[\log \pi_\theta(a|s)R]$ in RL?
  - Yes, except for the distribution: teacher demonstrations in IL, vs. learner in RL
  - Can the learner prefer good episodes? Well, that's the entire point...

IL as dense-reward RL

- In cross-entropy Behavior Cloning we maximize

$$\mathbb{E}_{s,a\sim p_T}[\log \pi_\theta(a|s)] = -\mathbb{D}[\pi_T\|\pi_\theta] - \mathbb{H}[\pi_T]$$

  - Like RL, but with teacher distribution and extremely sparse reward $R = 1_{\text{success}}$

- What if instead we minimize the other KL divergence?

$$\mathbb{D}[\pi_\theta\|\pi_T] = -\mathbb{E}_{s,a\sim p_\theta}[\log \pi_T(a|s)] - \mathbb{H}[\pi_\theta]$$

  - This is exactly RL with $r(s,a) = \log \pi_T(a|s)$ and entropy regularizer
  - Now $r(s,a)$ does give global information
  - In fact, with deterministic teacher, $r(s,a) = -\infty$ for any suboptimal action

Reward shaping

- One advantage of RL over IL is that rewards can be given programmatically
  - Allowing automatic supervision of many episodes
- Sparse reward functions may be easier to program than dense ones
  - Easier to identify good goal states and safety violations after the fact
- Reward shaping: the practice of adjusting the reward function for easier RL
  - More art than science, partly because "easy to program" is hard to quantify
- General tips:
  - Reward "bottleneck states": subgoals that are likely to help the bigger goals
  - To guide exploration, break down long sequences of coordinated actions
    - e.g. place reward beacons on long narrow paths, such that exploration from each can stumble on next

Learning with sparse rewards

- Montezuma's Revenge
  - Key = 100 points
  - Door = 500 points
  - Skull = 0 points
    - Is it good? Bad? Does something off-screen? Opens up an easter egg?
  - Humans have a head start with transfer from known objects
- Exploration before learning:
  - Random walk until you get some points — could take a while!

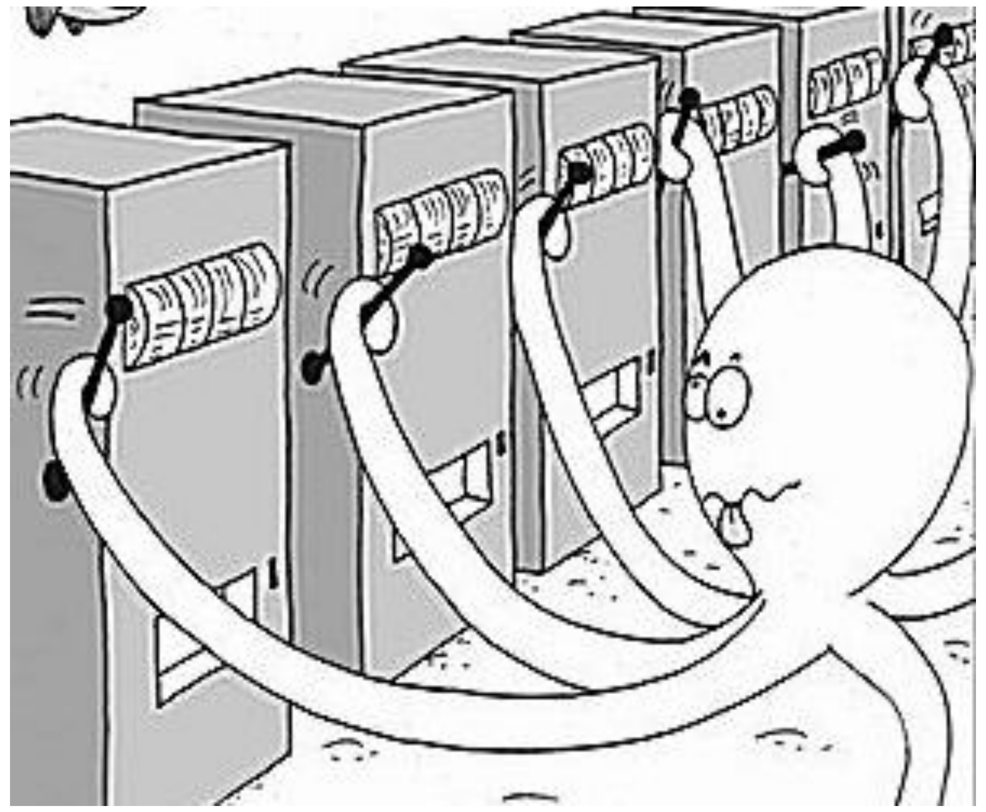
Optimal exploration in simplified settings

- Multi-Arm Bandits (MAB): single state, one-step horizon
  - Exploration–exploitation tradeoff very well understood
- Contextual bandits: random state, one-step horizon
  - Also has good theory; part of the exciting field of Online Learning
- Tabular RL
  - Some good heuristics, recent theoretical guarantees
- Deep RL
  - Only few exploratory ideas and heuristics

Multi-Arm Bandits (MABs)

- "One-arm bandit":

- Multi-arm bandit:



- States: $\{s_0\}$
- Actions: $\{\text{pull}_1, \ldots, \text{pull}_k\}$
- One-step, no transitions
- Rewards: $p(r|\text{pull}_i)$

Let's play!

- http://iosband.github.io/2015/07/28/Beat-the-bandit.html

Exploration vs. exploitation

- We can choose actions that seemed good so far (exploitation)
- But we could be missing out on even better ones (exploration)
- Algorithms we saw before would try everything enough times — trivial
- What if we care about rewards while we learn
- Regret: how much worse our return is than an optimal action

$$\rho(T) = T\,\mathbb{E}[r|a^*] - \sum_{t=0}^{T-1} r_t$$

- Can we get the regret to grow sub-linearly with T; average regret tends to 0

Optimism under uncertainty

- Let's be more conservative than E³ in out optimism
- Track the mean reward for each arm $\hat{\mu}_i = \frac{1}{N_i} \sum_{t_i} r_{t_i}$
- By the central limit theorem, the distribution $\hat{\mu}_i$ of tends quickly to Gaussian
  - with standard deviation $O\left(\frac{1}{\sqrt{N_i}}\right)$
- Let's be optimistic by a slowly-growing number of standard deviations

$$a = \operatorname*{argmax}_i \hat{\mu}_i + \sqrt{\frac{2 \ln T}{N_i}}$$

  - Has to grow because we don't know the constant in the variance
  - But not too fast, or we fail to exploit what we do know
- Regret: $\rho(T) = O(\log T)$, provably optimal

Learning as POMDP planning

- We can frame the learning problem as a POMDP planning problem
- Extend the state with the model parameters $\tilde{s}_t = (s_t, \theta)$
  - Uncontrollable, unobservable
- Now we "know" the dynamics: $p((s', \theta)|(s, \theta), a) = p_\theta(s'|s, a)$
- For the rewards: $p(r|(s, \theta), a) = p_\theta(r|s, a)$
- This is a special case of POMDP planning
  - POMDP planning in parameter state space is at least as hard as MDP learning
  - Too hard to solve with POMDP methods, even in the bandits case

Thompson sampling

- In the bandits case: $p_{\theta_i}(r|a_i)$
- Consider the belief = posterior over θ (note: distribution over distributions)
- Computing the belief value function: optimal experiment design; challenging
- Approximation:
  - Sample $\theta|(a_t, r_t)_t \sim b_t$ from the belief
  - Take the optimal action
  - Update the belief
  - Repeat

RL exploration is more complicated...

- Need to consider states and dynamics
- Need coordinated behavior to get *anywhere*
  - Cross a bridge to get the game started
  - Random exploration will kill us with high probability
    - Structured exploration
- How to define regret?
  - With respect to constant action? We can outperform it
  - With respect to optimal policy? May be too hard to learn, linear regret
  - Most approaches are heuristic, no regret guarantees

Count-based exploration

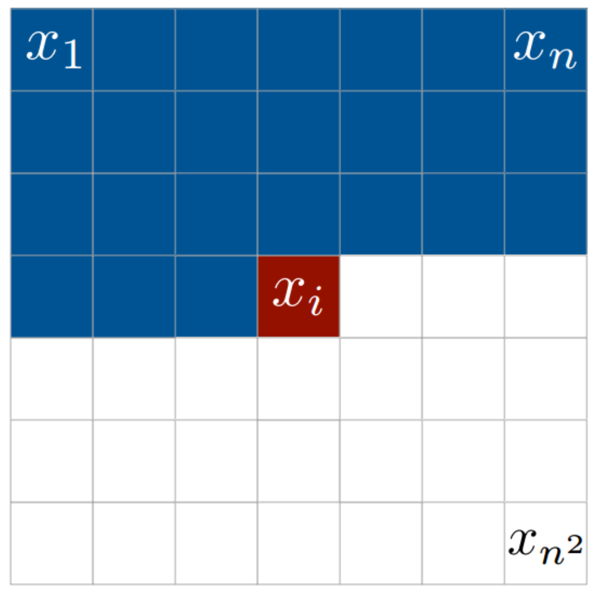
- Generalizing $a = \operatorname*{argmax}_i \hat{\mu}_i + \sqrt{\frac{2 \ln T}{N_i}}$ to RL
- Count visitations to each state $N(s)$ (or state-action $N(s, a)$)
- Optimism under uncertainty, add exploration bonus to scarcely-visited states

$$\tilde{r} = r + r_e(N(s))$$

  - r_e should be monotonic decreasing in $N(s)$
  - Need to tune its weight

Density model for count-based exploration

- How to represent "counts" in large state spaces?
  - We may never see the same state twice
  - If a state is very similar to ones we've seen often, is it new?
- Train a density model $p_\phi(s)$ over past experience
- Unlike generative models, we care about getting the density correctly
  - But not about the quality of samples
- Density models for images:
  - CTS, PixelRNN, PixelCNN, etc.



Pseudo-counts

- How to infer pseudo-counts from a density model?

$$p_\phi(s) = \frac{N(s)}{N}$$

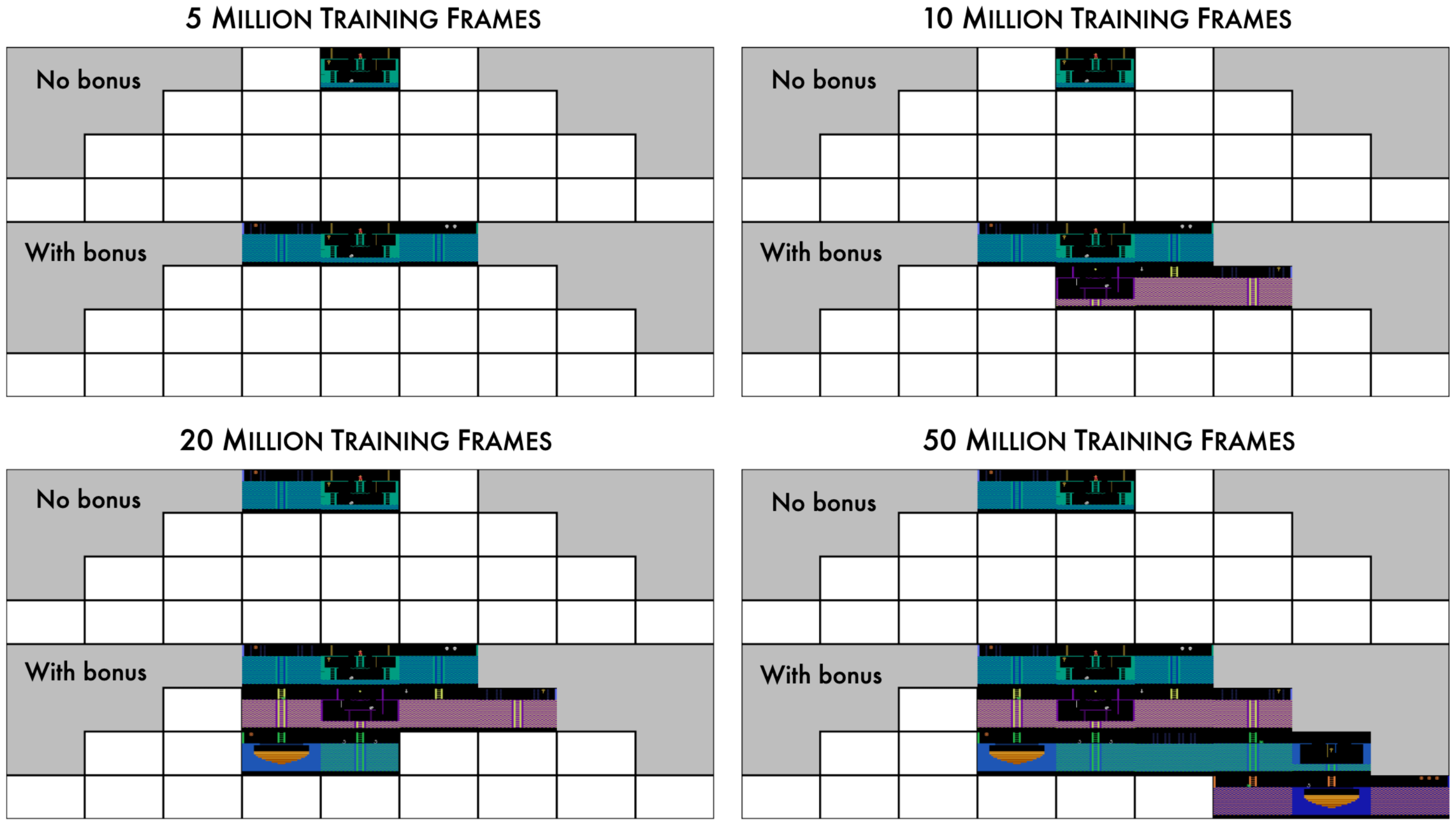
- After another visit:

$$p_{\phi'}(s) = \frac{N(s)+1}{N+1}$$

- To recover the pseudo-count:
  - $p_{\phi'} \leftarrow$ mock-update the density model with another visit of s
  - Compute $\hat{N} = \frac{1-p_{\phi'}(s)}{p_{\phi'}(s)-p_\phi(s)} p_\phi(s)$ $\qquad \hat{N}(s) = \hat{N} p_\phi(s)$

Exploration bonus

- What's a good exploration bonus?
- In bandits: Upper Confidence Bound (UCB) $$r_e(N(s)) = \sqrt{\frac{2 \ln N}{N(s)}}$$
- In RL, often: $$r_e(N(s)) = \sqrt{\frac{1}{N(s)}}$$
- [Bellemare et al., 2016]:



Thompson sampling for RL

- Keep a distribution over models
- What's our model?
  - MDP
  - Q-function
- Sample $Q \sim p_\theta$
- Roll out an episode with the greedy policy $\pi = \underset{a}{\operatorname{argmax}} Q$
- Use experience to update $p(Q)$
- Repeat

Recap

- Dense rewards help, but hard to generate
- Challenges of random exploration can be overcome with
  - Count-based exploration bonus for novelty, effective way to make rewards denser
  - Posterior sampling for coordinated exploration actions