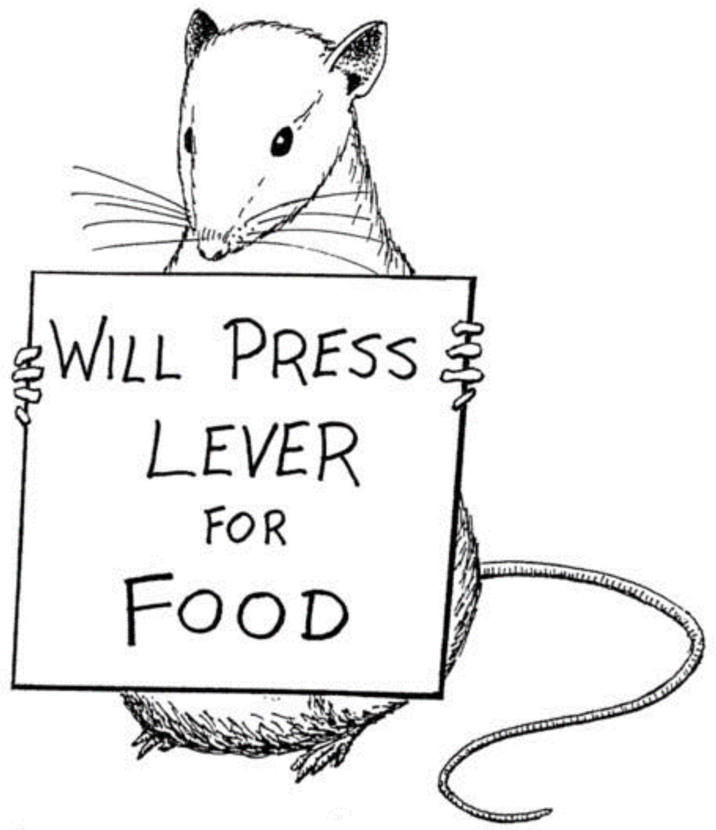
University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2021

Lecture 13: Exploration

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



Today's lecture

Sparse rewards

Multi-Armed Bandits

Exploration in Deep RL

Relation between RL and IL

- What makes RL harder than IL?
  - IL: teacher policy $\pi_e(a \mid s)$ indicates a good action to take in s
  - RL: $r(s, a)$ does not indicate a globally good action; $Q^*(s, a)$ does, but it's nonlocal
- But didn't we see an equivalence between RL and IL?
  - NLL loss in BC: $\nabla_\theta \mathbb{E}[\log \pi_\theta(a \mid s)]$
    - s and a sampled from teacher distribution (this makes IL harder than RL...)
  - PG loss: $\nabla_\theta \mathbb{E}[\log \pi_\theta(a \mid s) R]$
    - s and a sampled from learner distribution

Informational quantities: refresher

- Entropy: $\mathbb{H}[p(a)] = -\mathbb{E}_{a\sim p}[\log p(a)] = -\sum_a p(a)\log p(a)$
- Conditional entropy: $\mathbb{H}[\pi|s] = -\mathbb{E}_{a\sim\pi}[\log \pi(a|s)]$
- Expected conditional entropy: $\mathbb{H}[\pi] = \mathbb{E}_{s\sim p_\pi}[\mathbb{H}[\pi|s]] = -\mathbb{E}_{s,a\sim p_\pi}[\log \pi(a|s)]$
- Expected relative entropy: $\mathbb{D}[\pi\|\pi'] = \mathbb{E}_{s,a\sim p_\pi}\left[\log \frac{\pi(a|s)}{\pi'(a|s)}\right]$
- Expected cross entropy (aka NLL): $-\mathbb{E}_{s,a\sim p_\pi}[\log \pi'(a|s)]$
  - $\mathbb{D}[\pi\|\pi'] = \text{NLL} - \mathbb{H}[\pi]$

IL as sparse-reward RL

- NLL BC: maximize $\mathbb{E}_{s,a\sim p_e}[\log \pi_\theta(a\,|\,s)] = -\mathbb{D}[\pi_e \| \pi_\theta] - \mathbb{H}[\pi_e]$

  constant in θ

  - Experience from teacher distribution p_e
    - RL: experience from learner distribution p_θ
  - “Return” $R = 1_{\text{success}}$ for successful trajectory
    - RL: $r_t = r(s_t, a_t)$ in every step
- Sparse reward = most rewards are 0 $\implies$ rare learning signal
  - $R = 1$ on success = very sparse; but doesn't IL provide dense learning signal?

IL as dense-reward RL

- What if instead we minimize the other relative entropy?

teacher labeling of learner states/actions as in DAgger

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

  - This is exactly the RL objective, with $r(s,a) = \log \pi_e(a|s)$ and entropy regularizer
  - Now $r(s,a)$ does give global information on optimal action
  - In fact, with deterministic teacher, $r(s,a) = -\infty$ for any suboptimal action
- The same return can be viewed as sum of sparse or dense rewards
  - Can we do the same in proper RL?

Reward shaping

- Ideal reward: $r(s, a) = -\infty$ for any suboptimal action $\implies$ as hard to provide as π^*
  - We need supervision signal that's sufficiently easy to program $\implies$ generate more data
- Sparse reward functions may be easier than dense ones
  - E.g., may be easy to identify good goal states, safety violations, etc.
- Reward shaping: art of adjusting the reward function for easier RL; some tips:
  - Reward "bottleneck states": subgoals that are likely to lead to bigger goals
  - Break down long sequences of coordinated actions $\implies$ better exploration
    - E.g. reward beacons on long narrow paths, for exploration to stumble upon

Learning with sparse rewards

- Montezuma's Revenge
  - Key = 100 points
  - Door = 500 points
  - Skull = 0 points
    - Is it good? Bad? Affects 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: simple settings

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

Today's lecture

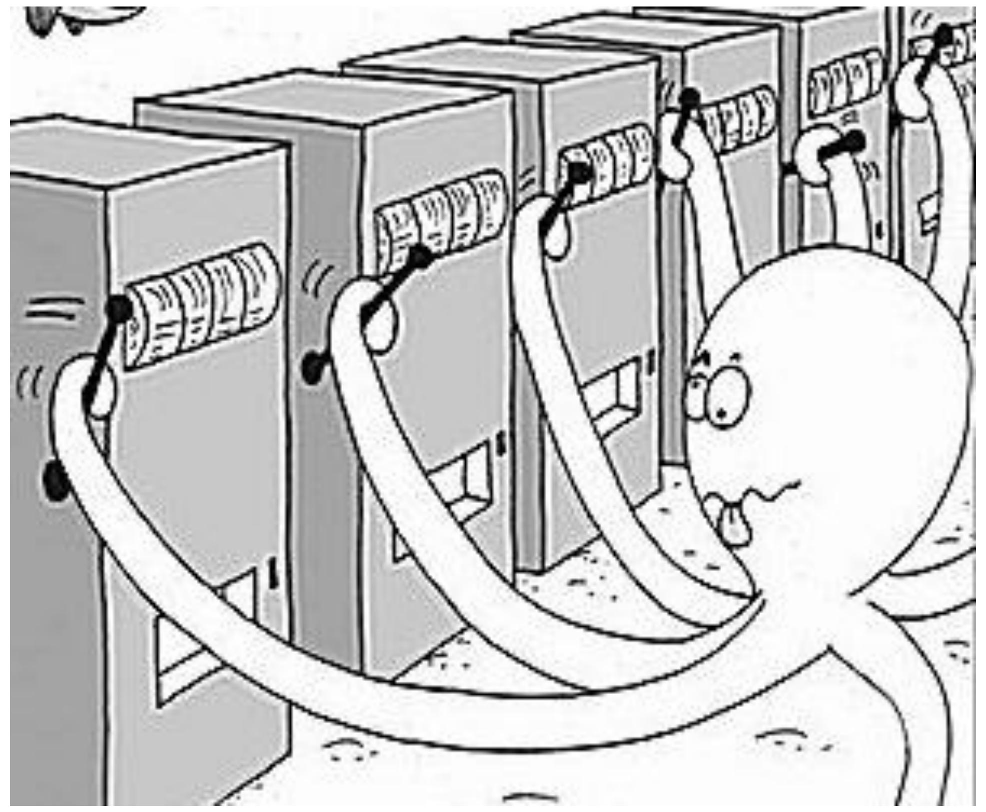
Sparse rewards

Multi-Armed Bandits

Exploration in Deep RL

Multi-Armed Bandits (MABs)

- "One-armed bandit":
- Multi-armed bandit:



- States: $\mathcal{S} = \{s_0\}$
- Actions: $\mathcal{A} = \{\text{pull}_1, \ldots, \text{pull}_k\}$
- One time step, no transitions
- Rewards: $p(r \mid \text{pull}_i)$

Exploration vs. exploitation

- Exploitation = choose actions that seems good (so far)
- Exploration = see if we're missing out on even better ones
- Model-based algorithms (E^3, R-MAX) learn r by trying every action enough times
  - Suppose we can't wait that long: 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? $\implies$ average $\frac{\rho(T)}{T} \to 0$

Let's play!

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

Optimism under uncertainty

- E^3: optimistic while the model isn't known; we need to start exploiting sooner
- 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 of $\hat{\mu}_i$ quickly $\to \mathcal{N}\left(\mu_i, O\left(\frac{1}{\sqrt{N_i}}\right)\right)$
- Be optimistic by slowly-growing number of standard deviations: $a = \arg\max_i \hat{\mu}_i + \sqrt{\frac{2 \ln T}{N_i}}$
  - 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

- MDP learning as POMDP planning:
  - Extend the state with the model parameters $\tilde{s}_t = (s_t, \phi)$
  - ϕ uncontrollable, unobservable
- Now we "know" the dynamics: $p((s', \theta) \,|\, (s, \theta), a) = p_\theta(s' \,|\, s, a)$
- For the rewards: $p(r \,|\, (s, \theta), a) = p_\theta(r \,|\, s, a)$
- POMDP planning in parameter space = 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

Today's lecture

Sparse rewards

Multi-Armed Bandits

Exploration in Deep RL

RL exploration is more complicated...

- Need to consider states and dynamics
- Need coordinated behavior to get *anywhere*
  - E.g., 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 $\Longrightarrow$ linear regret
  - Most approaches are heuristic, no regret guarantees

Count-based exploration

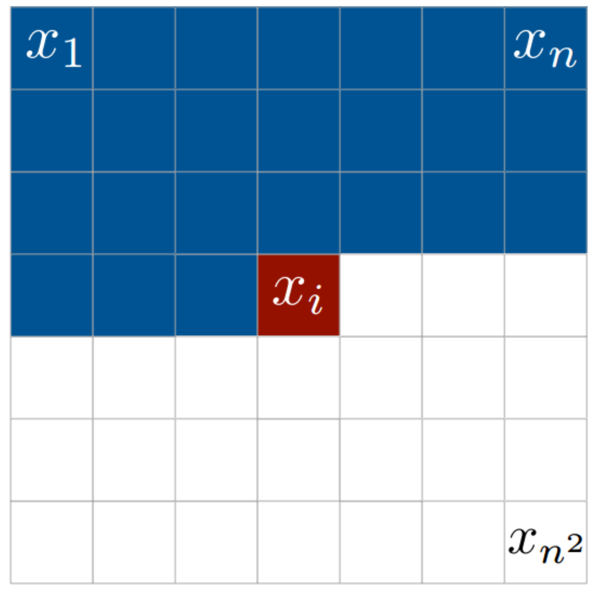
- Generalizing $a = \underset{i}{\operatorname{argmax}}\, \hat{\mu}_i + \sqrt{\frac{2 \ln T}{N_i}}$ to RL
- Count visitations to each state $N(s)$ (or state-action $N(s, a)$)
- Optimism under uncertainty: add exploration bonus to scarcely-visited states

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

  - 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 we don't care about the quality of samples
- Density models for images:
  - CTS, PixelRNN, PixelCNN, etc.



Pseudo-counts

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

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

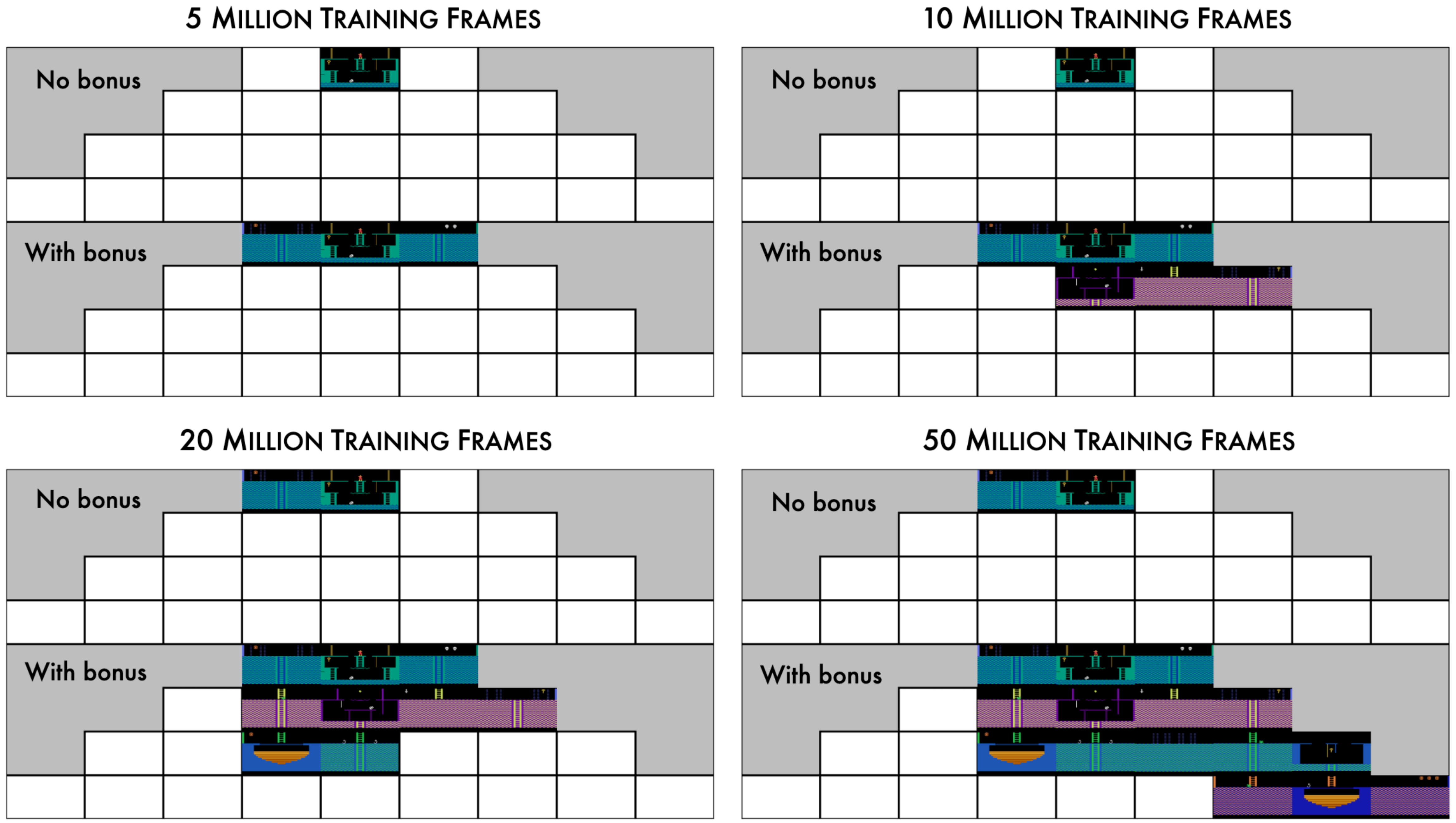
- After another visit:

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

- 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 $p_\theta(\phi)$
- What's our "model"? Idea 1: MDP; Idea 2: Q-function
- Thompson sampling over Q-functions:
  - Sample $Q \sim p_\theta$
  - Roll out an episode with the greedy policy $\pi(s) = \arg\max_a Q(s, a)$
  - Update p_θ to be more likely for Q' that gives low empirical Bellman error
  - 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