Computer Engineering Department

Exploration in RL

Mohammad Hossein Rohban, Ph.D.

Spring 2024

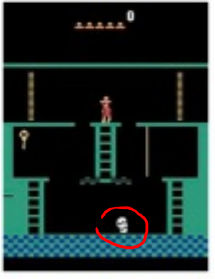
Courtesy: Most of slides are adopted from CS 285, UC Berkeley.

What's the problem?

this is easy (mostly)

Why?

this is impossible

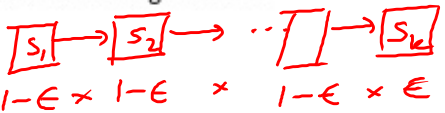


Montezuma's revenge

- Getting key = reward
- Opening door = reward
- Getting killed by skull = nothing (is it good? bad?)
- Finishing the game only weakly correlates with rewarding events
- We know what to do because we understand what these sprites mean!

Why exploration can be difficult?

- **Temporally extended** tasks like Montezuma's revenge become increasingly difficult based on
  - How extended the task is
  - How little you know about the rules



- Lets' assume a **complex task**
  - Consisting of multiple sub-task, each is a prerequisite for the next sub-task.
  - Each should be solved in a sequence to get a high reward
  - Epsilon greedy does not obviously help:
  - Suppose you mastered up to the k^{th} sub-task.
  - You have to exploit up to the k^{th} task and then explore onwards.
  - Now the chance to only explore in the sub-task (k+1) is $(1-\varepsilon)^{O(k)}\varepsilon^{O(1)}$.
  - For eps = 0.1, this is ~ 6%. For eps = 0.5, this is ~ 2%.

as $k \to \infty$ $\to 0$

Exploration and exploitation

- Two potential definitions of exploration problem:
  - How can an agent **discover** **high-reward strategies** that require a temporally **extended sequence of complex behaviors** that, **individually, are not rewarding**?
  - How can an agent decide whether to **attempt new behaviors** (to discover ones with higher reward) or **continue to do the best thing it knows so far**?

Optimal Exploration?

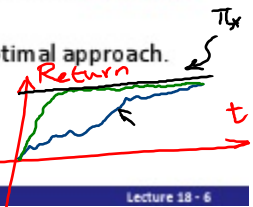
- Bayesian model of the environment. (POMDP with belief state)
- Optimize the expected reward under all uncertainties.
- Requires knowledge of state dynamic distribution class, the prior, and maintaining the belief state.
- Here we seek simpler solutions which could be extended to more complex scenarios.
- Compare the regret in such models against the Bayes' optimal approach.

Regret

$$\mathrm{Reg}(T) = TE[r(a^*)] - \sum_{t=1}^{T} r(a_t)$$

expected reward of best action (the best we can hope for in expectation)

actual reward of action actually taken



$$\mathbb{E}\left[\sum_{t=0}^{T} r_t^{\pi_{a^*}} \quad \sum_{t=0}^{T} r_t^{\pi}\right]$$

Bandits

assume $r(a_i) \sim p_{\theta_i}(r_i)$

e.g., $p(r_i = 1) = \theta_i$ and $p(r_i = 0) = 1 - \theta_i$

$\theta_i \sim p(\theta)$, but otherwise unknown

this defines a POMDP with $\mathbf{s} = [\theta_1, \ldots, \theta_n]$

belief state is $p(\theta_1, \ldots, \theta_n)$

- solving the POMDP yields the optimal exploration strategy
- but that's overkill: belief state is huge!
- we can do very well with much simpler strategies

how do we measure goodness of exploration algorithm?

regret: difference from optimal policy at time step T: $\mathrm{Reg}(T) = TE[r(a^*)] - \sum_{t=1}^{T} r(a_t)$

expected reward of best action (the best we can hope for in expectation)

actual reward of action actually taken

Optimistic exploration

keep track of average reward $\hat{\mu}_a$ for each action a

exploitation: pick $a = \arg\max \hat{\mu}_a$

optimistic estimate $a = \arg\max \hat{\mu}_a + C\hat{\sigma}_a$

some sort of variance estimate

intuition: try each arm until you are *sure* it's not great

example (Auer et al. Finite-time analysis of the multiarmed bandit problem):

$$a = \arg\max \hat{\mu}_a + \sqrt{\frac{2 \ln T}{N(a)}}$$

number of times we picked this action

$\mathrm{Reg}(T)$ is $O(\log T)$, provably as good as any algorithm

$$\hat{\mu}(a) = \frac{1}{N(a)} \sum_{i=1}^{N(a)} R_i$$

$$\frac{1}{T}\text{Regret} = \frac{1}{T} \sum_{t=1}^{T} Q(a^*) - Q(a_t)$$

(MAB)

sublinear

$$Q(a_i) = \mathbb{E}(r(a_i)) \qquad a^* = \underset{a}{\text{argmax}}\, Q(a)$$

$\forall X \geq 0$: $\mathbb{P}(X \geq \epsilon) \leq \frac{\mathbb{E}(X)}{\epsilon}$

Markov Inequality

proof:

$$= \frac{\int_{\epsilon}^{\infty} \epsilon f_X(x)\,dx}{\epsilon}$$

$$\leq \int_{\epsilon}^{\infty} x f_X(x)\,dx \Big/ \epsilon \leq \int_{0}^{\infty} \frac{x f_X(x)\,dx}{\epsilon}$$

$$= \mathbb{E}(X)/\epsilon$$

$X \longrightarrow e^{tX}$ $\quad t > 0$ $\quad Y$

$\forall t \geq 0 \quad \mathbb{P}(X \geq \epsilon) = \mathbb{P}(\underbrace{e^{tX}}_{>0} \geq \underbrace{e^{t\epsilon}}_{\epsilon'})$

$\leq \mathbb{E}(Y)/\epsilon' = \mathbb{E}(e^{tX})/e^{t\epsilon}$

$\Rightarrow \quad \mathbb{P}(X \geq \epsilon) \leq \inf_{t>0} \dfrac{\mathbb{E}(e^{tX})}{e^{t\epsilon}}$

Chernoff Bound

Suppose $E(X) = 0$, $X \in [a, b]$

$E(e^{tX})$

$X = \alpha b + (1-\alpha) a$

$0 \le \alpha \le 1$

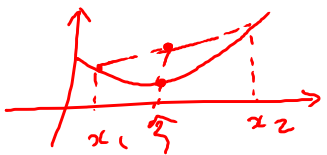
$\alpha (b-a) = X - a$

$\alpha = \frac{X-a}{b-a}$,

$E(\alpha) = \frac{-a}{b-a}$

for any convex function f

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2)$$



Let $f(x) = e^{tx}$ $\qquad x = \alpha b + (1-\alpha)a$

$$e^{tx} \leq \alpha e^{tb} + (1-\alpha) e^{ta}$$

$$g''(u) = \frac{(-\gamma e^u)(-\gamma e^u + 1 + \gamma) + \gamma e^u(-\gamma e^u)}{(-\gamma e^u + 1 + \gamma)^2} = p(1-p)$$

$$\Rightarrow \mathbb{E}\left(e^{tX}\right) \leq \mathbb{E}\left(\alpha e^{tb} + (1-\alpha)e^{ta}\right)$$

$$= \frac{-a}{b-a} e^{tb} + \frac{b}{b-a} e^{ta}$$

$$= e^{ta}\left(\frac{-a}{b-a} e^{t(b-a)} + \frac{b}{b-a}\right)$$

$-\gamma$

$u := t(b-a) \qquad e^{g(u)} \qquad \gamma := \frac{a}{b-a}$

$$g(u) = \gamma u + \log\left(-\gamma e^u + 1 + \gamma\right)$$

$$g'(u) = \gamma + \frac{-\gamma e^u}{-\gamma e^u + 1 + \gamma} \rightarrow g'(u) = 0$$

$$g(u) = 0 + \frac{0 \times u}{1!} + \frac{g''(\xi)}{2!} u^2$$

$\xi \in [0, u]$

$$\underbrace{g''(\xi)}_{p(1-p)} \leq 1/4$$

$$g(u) \leq \frac{1}{8} u^2$$

$$\Rightarrow \underline{E\left(e^{t(X)}\right)} \leq e^{g(u)} \leq e^{\frac{1}{8}u^2}$$

$$= \underline{e^{\frac{1}{8}t^2(b-a)^2}}$$

$$\mathbb{P}(X \geq \epsilon) \leq \inf_{t \geq 0} \frac{\mathbb{E}(e^{tX})}{e^{t\epsilon}}$$

$$\leq \inf_{t \geq 0} \frac{e^{1/8\, t^2 (b-a)^2}}{e^{t\epsilon}}$$

$$e^{1/8\, t^2 (b-a)^2 - t\epsilon}$$

$$\frac{1}{4} t (b-a)^2 - \epsilon = 0 \rightarrow t = \frac{4\epsilon}{(b-a)^2}$$

$$\Rightarrow e^{-2\epsilon^2/(b-a)^2}$$

Hoeffding Bound

$$\mathbb{P}\left(\frac{X_1 + \cdots + X_n}{n} - \mathbb{E}(X) \geq \epsilon\right) \leq \inf_t \mathbb{E}\left(e^{tZ}\right) / e^{t\epsilon} = e^{-2n\epsilon^2/c}$$

$$\frac{\delta}{t^2}$$

$$c = (b-a)^2$$

$$\epsilon = \sqrt{\frac{2\log t^2/\delta}{n}}$$

w.h.p. $1 - \delta/t^2$

$$\hat{\mu} \leq \mathbb{E}(X) + \sqrt{\frac{2\log t^2/\delta}{n}}$$

$$U(a_t) := \hat{Q}(a_t) + \sqrt{\frac{2 \log t/\delta}{N(a_t)}}$$

$$\text{Regret} = \sum_{t=1}^{T} Q(a^*) - Q(a_t)$$

$$= \sum_{t=1}^{T} \underbrace{Q(a^*) - U(a_t)}_{< 0 \text{ w.h.p}} - \underbrace{Q(a_t) - U(a_t)}_{\geq}$$

$$a_t = \arg\max_a U(a) \qquad a^* = \arg\max_a Q(a)$$

Assumption $\forall t, \forall a, \quad Q(a) - \hat{Q}(a) \leq \sqrt{\frac{2\log t/\delta}{N(a)}} \epsilon$

(1) $a_t = a^*$ $\quad Q(a^*) \leq U(a^*)$ w.h.p

$$Q(a^*) - \hat{Q}(a^*) \leq \epsilon$$

$$\Rightarrow Q(a^*) \leq \underbrace{\hat{Q}(a^*) + \epsilon}_{U(a^*)}$$

(2) $a_t \neq a^*$ $\quad U(a_t) \geqslant U(a^*) \geqslant Q(a^*)$ w.h.p

$$\Rightarrow U(a_t) \geqslant Q(a^*)$$

$$\Rightarrow \text{Regret} \leq \sum_{t=1}^{T} U(a_t) - Q(a_t)$$

$$\hat{Q}(a_t) + \sqrt{\frac{2\log t/\delta}{N(a_t)}}$$

$$\hat{Q}(a_t) - Q(a_t) \leq \sqrt{\frac{2\log t/\delta}{N(a_t)}} \quad \text{with prob. of at least } 1 - \delta/t^2$$

$$\Rightarrow \text{Regret} \leq \sum_{t=1}^{T} 2\sqrt{\frac{2\log t/\delta}{N(a_t)}}$$

w.p. of at least $\mathbb{P}(\text{Assumptions})$

$1 - 2m\delta$

$$\mathbb{P}(\text{Assumptions : } \forall t \forall a \text{ Confid. Bounds are satisfied})$$

$$= 1 - \mathbb{P}(\exists t \exists a \text{ Conf. Bound is violated})$$

$$= 1 - \mathbb{P}\Big(\underbrace{\bigcup_{t=1}^{T} \bigcup_{a=a_1}^{a_m}}_{} Q(a_i^{(t)}) - \hat{Q}(a_i^{(t)}) \geq \sqrt{\frac{2\log \delta/t}{N_t(a_i)}}\Big)$$

Union Bound $\quad \mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$

$$\mathbb{P}(\text{Assumption}) \geq 1 - \sum_{t,a} \underbrace{\mathbb{P}(Q - \hat{Q} \geq \epsilon)}_{\delta / t^2}$$

$$\sum_{t=1}^{T} 1/t^2 \leq \sum_{t=1}^{\infty} 1/t^2 = \pi^2/6 < 2$$

$$\Rightarrow \quad \mathbb{P}(\text{Assumption}) \geq \underline{1 - 2m\delta}$$

$m = |A|$

$$\text{Regret} \leq \sum_{t=1}^{T} 2\sqrt{\frac{2\log t/\delta}{N(a_t)}}$$

$$\leq 2\sqrt{2\log T/\delta} \sum_{t=1}^{T} \sqrt{\frac{1}{N(a_t)}}$$

$$\sum_{a=a_1}^{a_m} \sum_{i=1}^{n(a)} \sqrt{\frac{1}{i}}$$

maximized when $n(a)$ are all equal.

$$\leq 2\sqrt{\log T/\delta} \sum_{a=a_1}^{am} \sum_{i=1}^{T/m} \sqrt{\frac{1}{i}}$$

$$\sum_{i=1}^{M} \frac{1}{\sqrt{i}} \leq 2\sqrt{M}$$

$$2\sqrt{T/m}$$

$$\leq 4\sqrt{2Tm \log T/\delta} \quad \text{w.h.p}$$

↘ Sublinear

Probability matching/posterior sampling

assume $r(a_i) \sim p_{\theta_i}(r_i)$

this defines a POMDP with $\mathbf{s} = [\theta_1, \ldots, \theta_n]$

belief state is $\hat{p}(\theta_1, \ldots, \theta_n)$

this is a *model* of our bandit

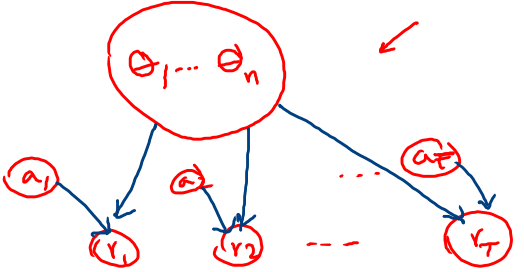
idea: sample $\theta_1, \ldots, \theta_n \sim \hat{p}(\theta_1, \ldots, \theta_n)$

pretend the model $\theta_1, \ldots, \theta_n$ is correct

take the optimal action

update the model

- This is called posterior sampling or Thompson sampling
- Harder to analyze theoretically
- Can work very well empirically
- See: Chapelle & Li, "An Empirical Evaluation of Thompson Sampling."


$\theta_1 \cdots \theta_n$
a_1
a_2
a_T
r_1
r_2
r_T

Information gain

Bayesian experimental design:

VIME

say we want to determine some latent variable z (e.g., z might be the optimal action, or its value)

which action do we take?

let $\mathcal{H}(\hat{p}(z))$ be the current entropy of our z estimate

let $\mathcal{H}(\hat{p}(z)|y)$ be the entropy of our z estimate after observation y (e.g., y might be $r(a)$)

the lower the entropy, the more precisely we know z

s_{t+1} s_t, a_t

$$\mathrm{IG}(z, y) = E_y[\mathcal{H}(\hat{p}(z)) - \mathcal{H}(\hat{p}(z)|y)]$$

max a

typically depends on action, so we have $\mathrm{IG}(z; y|a)$

Information gain example

$$\mathrm{IG}(z, y|a) = E_y[\mathcal{H}(\hat{p}(z)) - \mathcal{H}(\hat{p}(z)|y)|a]$$

how much we learn about z from action a, given current beliefs

Example bandit algorithm:

Russo & Van Roy "Learning to Optimize via Information-Directed Sampling"

$y = r_a$, $z = \theta_a$ (parameters of model $p(r_a)$)

$g(a) = \mathrm{IG}(\theta_a, r_a|a)$ – information gain of a

$\Delta(a) = E[r(a^\star) - r(a)]$ – expected suboptimality of a

choose a according to $\arg\min_a \frac{\Delta(a)^2}{g(a)}$

don't take actions that you're sure are suboptimal

don't bother taking actions if you won't learn anything

$\min_a \frac{-\mathbb{E}^2(r(a))}{g(a)}$

General themes

UCB:

reward

$$a = \arg\max \hat{\mu}_a + \sqrt{\frac{2 \ln T}{N(a)}}$$

Thompson sampling:

$$\theta_1, \ldots, \theta_n \sim \hat{p}(\theta_1, \ldots, \theta_n)$$

$$a = \arg\max_a E_{\theta_a}[r(a)]$$

Info gain:

$$\mathrm{IG}(z, y|a)$$

- Most exploration strategies require some kind of uncertainty estimation (even if it's naïve)
- Usually assumes some value to new information
  - Assume unknown = good (optimism)
  - Assume sample = truth
  - Assume information gain = good

Optimistic exploration in RL

UCB: $a = \arg\max \hat{\mu}_a + \sqrt{\frac{2 \ln T}{N(a)}}$

"exploration bonus"

lots of functions work, so long as they decrease with $N(a)$

can we use this idea with MDPs?

count-based exploration: use $N(\mathbf{s}, \mathbf{a})$ or $N(\mathbf{s})$ to add *exploration bonus*

use $r^+(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + B(N(\mathbf{s}))$

$B(x) = 1/x$ or $\sqrt{1/x}$

bonus that decreases with $N(\mathbf{s})$

use $r^+(\mathbf{s}, \mathbf{a})$ instead of $r(\mathbf{s}, \mathbf{a})$ with any model-free algorithm

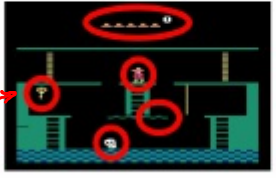
+ simple addition to any RL algorithm

- need to tune bonus weight

The trouble with counts

use $r^+(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \mathcal{B}(N(\mathbf{s}))$

But wait… what's a count?



Uh oh… we never see the same thing twice!

But some states are more similar than others

Fitting generative models

idea: fit a density model $p_\theta(\mathbf{s})$ (or $p_\theta(\mathbf{s}, \mathbf{a})$)

$p_\theta(\mathbf{s})$ might be high even for a new $\mathbf{s}$

if $\mathbf{s}$ is similar to previously seen states

can we use $p_\theta(\mathbf{s})$ to get a "pseudo-count"?

if we have small MDPs

the true probability is:

$$P(\mathbf{s}) = \frac{N(\mathbf{s})}{n}$$

$N(\mathbf{s})$: count

n: total states visited

$P(\mathbf{s})$: probability/density

after we see $\mathbf{s}$, we have:

$$P'(\mathbf{s}) = \frac{N(\mathbf{s}) + 1}{n + 1}$$

Exploring with pseudo-counts

fit model $p_\theta(\mathbf{s})$ to all states $\mathcal{D}$ seen so far

take a step i and observe $\mathbf{s}_i$

fit new model $p_{\theta'}(\mathbf{s})$ to $\mathcal{D} \cup \mathbf{s}_i$

use $p_\theta(\mathbf{s}_i)$ and $p_{\theta'}(\mathbf{s}_i)$ to estimate $\hat{N}(\mathbf{s})$

set $r_i^+ = r_i + \mathcal{B}(\hat{N}(\mathbf{s}))$ ← "pseudo-count"

$p_\theta(\mathbf{s})$

$p_{\theta'}(\mathbf{s})$

how to get $\hat{N}(\mathbf{s})$? use the equations

known $p_\theta(\mathbf{s}_i) = \frac{\hat{N}(\mathbf{s}_i)}{\hat{n}}$ $\propto \hat{N}(\mathbf{s}_i)$

known $p_{\theta'}(\mathbf{s}_i) = \frac{\hat{N}(\mathbf{s}_i)+1}{\hat{n}+1}$ $\propto \hat{N}(\mathbf{s}_i)+1$

two equations and two unknowns!

$$\hat{N}(\mathbf{s}_i) = \hat{n} p_\theta(\mathbf{s}_i) \qquad \hat{n} = \frac{1 - p_{\theta'}(\mathbf{s}_i)}{p_{\theta'}(\mathbf{s}_i) - p_\theta(\mathbf{s}_i)} p_\theta(\mathbf{s}_i)$$

Bellemare et al. "Unifying Count-Based Exploration..."

What kind of generative modeling to use?

$p_\theta(\mathbf{s})$

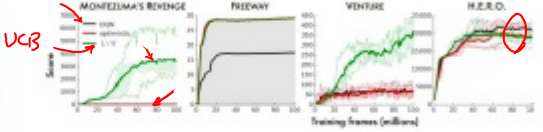
need to be able to output densities, but doesn't necessarily need to produce great samples

opposite considerations from many popular generative models in the literature (e.g., GANs)

Bellemare et al.: "CTS" model: condition each pixel on its top-left neighborhood

Other models: stochastic neural networks, compression length, EX2

Does it work?



UCB

Bellemare et al. "Unifying Count-Based Exploration..."

Counting with hashes

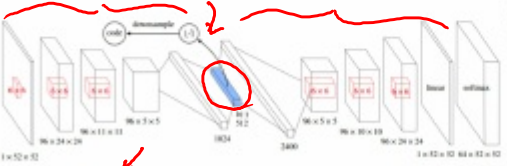
What if we still count states, but in a different space?

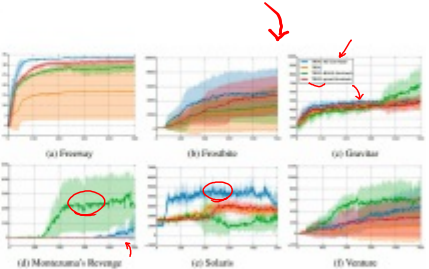
idea: compress $\mathbf{s}$ into a k-bit code via $\phi(\mathbf{s})$, then count $N(\phi(\mathbf{s}))$

shorter codes = more hash collisions

similar states get the same hash? maybe

improve the odds by *learning* a compression:





Tang et al. "#Exploration: A Study of Count-Based Exploration"

Implicit density modeling with exemplar models

$p_\theta(\mathbf{s})$ need to be able to output densities, but doesn't necessarily need to produce great samples

Can we explicitly compare the new state to past states?

Intuition: the state is **novel** if it is **easy** to distinguish from all previous seen states by a classifier

for each observed state $\mathbf{s}$, fit a classifier to classify that state against all past states $\mathcal{D}$, use classifier error to obtain density

$$p_\theta(\mathbf{s}) = \frac{1 - D_{\mathbf{s}}(\mathbf{s})}{D_{\mathbf{s}}(\mathbf{s})}$$

$D_{\mathbf{s}}(\mathbf{s})$: probability that classifier assigns that $\mathbf{s}$ is "positive"
positives: $\{\mathbf{s}\}$
negatives: $\mathcal{D}$

Fu et al. "EX2: Exploration with Exemplar Models..."

Implicit density modeling with exemplar models

hang on... aren't we just checking if $\mathbf{s} = \mathbf{s}$?

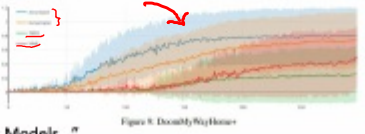
if $\mathbf{s} \in \mathcal{D}$, then the optimal $D_{\mathbf{s}}(\mathbf{s}) \neq 1$

in fact: $D^*_{\mathbf{s}}(\mathbf{s}) = \frac{1}{1 + p(\mathbf{s})}$ $\Rightarrow$ $p_\theta(\mathbf{s}) = \frac{1 - D_{\mathbf{s}}(\mathbf{s})}{D_{\mathbf{s}}(\mathbf{s})}$

in reality, each state is unique, so we *regularize* the classifier

isn't one classifier per state a bit much?

train one *amortized* model: single network that takes in exemplar as input!



Fu et al. "EX2: Exploration with Exemplar Models..."

Heuristic estimation of counts via errors

$p_\theta(\mathbf{s})$

need to be able to output densities, but doesn't necessarily need to produce great samples

...and doesn't even need to output great densities

...just need to tell if state is **novel** or not!

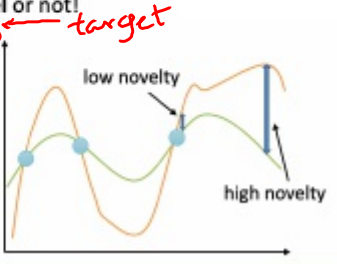
let's say we have some **target** function $f^*(\mathbf{s}, \mathbf{a})$

given our buffer $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i)\}$, fit $\hat{f}_\theta(\mathbf{s}, \mathbf{a})$

use $\mathcal{E}(\mathbf{s}, \mathbf{a}) = ||\hat{f}_\theta(\mathbf{s}, \mathbf{a}) - f^*(\mathbf{s}, \mathbf{a})||^2$ as bonus

$f^*(s_t, a_t) = s_{t+1}$

target



Heuristic estimation of counts via errors

what should we use for $f^\star(\mathbf{s}, \mathbf{a})$?

one common choice: set $f^\star(\mathbf{s}, \mathbf{a}) = \mathbf{s}'$ – i.e., next state prediction

even simpler: $f^\star(\mathbf{s}, \mathbf{a}) = f_\phi(\mathbf{s}, \mathbf{a})$, where ϕ is a *random* parameter vector

Burda et al. Exploration by random network distillation, 2018.

Posterior sampling in deep RL

Thompson sampling:

$\theta_1, \ldots, \theta_n \sim \hat{p}(\theta_1, \ldots, \theta_n)$

$a = \arg\max_a E_{\theta_a}[r(a)]$

What do we sample?

How do we represent the distribution?

bandit setting: $\hat{p}(\theta_1, \ldots, \theta_n)$ is distribution over *rewards*

MDP analog is the Q-function!

1. sample Q-function Q from $p(Q)$
2. act according to Q for one episode
3. update $p(Q)$

since Q-learning is off-policy, we don't care which Q-function was used to collect data

how can we represent a distribution over functions?

Osband et al. "Deep Exploration via Bootstrapped DQN"

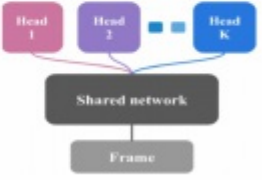
Bootstrap

given a dataset $\mathcal{D}$, resample with replacement N times to get $\mathcal{D}_1, \ldots, \mathcal{D}_N$

train each model f_{θ_i} on $\mathcal{D}_i$

to sample from $p(\theta)$, sample $i \in [1, \ldots, N]$ and use f_{θ_i}

training N big neural nets is expensive, can we avoid it?



Osband et al. "Deep Exploration via Bootstrapped DQN"

Bootstrap

$Q_1 \cdots Q_K$ are samples from the posterior dist.

Algorithm 1 Bootstrapped DQN

1: **Input:** Value function networks Q with K outputs $\{Q_k\}_{k=1}^K$. Masking distribution M.
2: Let B be a replay buffer storing experience for training.
3: **for** each episode **do**
4: Obtain initial state from environment s_0
5: Pick a value function to act using $k \sim \text{Uniform}\{1, \ldots, K\}$
6: **for** step $t = 1, \ldots$ until end of episode **do**
7: Pick an action according to $a_t \in \arg\max_a Q_k(s_t, a)$
8: Receive state s_{t+1} and reward r_t from environment, having taking action a_t
9: Sample bootstrap mask $m_t \sim M$
10: Add $(s_t, a_t, r_{t+1}, s_{t+1}, m_t)$ to replay buffer B
11: **end for**
12: **end for**

$Q_1^{(0)} \cdots Q_K^{(0)}$

$Q_1^{(t)} \cdots Q_K^{(t)}$

$P(Q) \rightsquigarrow P(Q \mid h_t)$

1100101

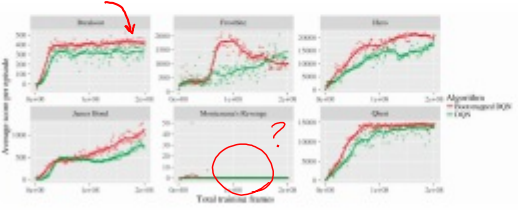
Osband et al. "Deep Exploration via Bootstrapped DQN"

$h_t = (s_0, a_0, s_1, a_1, s_2 \cdots, s_t)$

Why does this work?

Exploring with random actions (e.g., epsilon-greedy): oscillate back and forth, might not go to a coherent or interesting place

Exploring with random Q-functions: commit to a randomized but internally consistent strategy for an entire episode



+ no change to original reward function

- very good bonuses often do better

Osband et al. "Deep Exploration via Bootstrapped DQN"

Reasoning about information gain (approximately)

Info gain: $\mathrm{IG}(z, y|a)$ → action

$I(z; y | a)$ → a: a_t; y: s_{t+1}; z: subject of the learning

information gain about *what*?	
information gain about reward $r(\mathbf{s}, \mathbf{a})$?	not very useful if reward is sparse
state density $p(\mathbf{s})$? $p_{\pi^*}(s)$	a bit strange, but somewhat makes sense!
information gain about dynamics $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$?	good proxy for *learning* the MDP, though still heuristic

$z = p_\theta(s'|s,a)$ $\quad z = \theta$

Generally intractable to use exactly, regardless of what is being estimated!

Reasoning about information gain (approximately)

$I(X;Y) = \mathbb{E}_Y D_{KL}(P(X|Y) \| P(X))$

A few approximations:

prediction gain: $\log p_{\theta'}(\mathbf{s}) - \log p_\theta(\mathbf{s})$ (Schmidhuber '91, Bellemare '16)

intuition: if density changed a lot, the state was novel

variational inference: (Houthooft et al. "VIME")

IG can be equivalently written as $D_{\mathrm{KL}}(p(z|y)\|p(z))$

learn about *transitions* $p_\theta(s_{t+1}|s_t, a_t)$: $z = \theta$

$y = (s_t, a_t, s_{t+1})$

$r(s_t, a_t) + D_{\mathrm{KL}}(p(\theta|h, s_t, a_t, s_{t+1})\|p(\theta|h))$

$\prod_{i=1}^{d} q(\theta_i|\phi_i)$

model parameters for $p_\theta(s_{t+1}|s_t, a_t)$

history of all prior transitions

newly observed transition

intuition: a transition is more informative if it causes belief over θ to change

idea: use variational inference to estimate $q(\theta|\phi) \approx p(\theta|h)$

given new transition (s, a, s'), update ϕ to get ϕ'

$\underline{\theta} = (\theta_1, \cdots, \theta_d)$

$$P(\underline{\theta} \mid h) = \prod_{i=1}^{d} q(\theta_i \mid \Phi_i)$$

$$q(\cdot \mid (\mu_i, \sigma_i)) = \mathcal{N}(\mu_i, \sigma_i^2) \quad \text{VI} \downarrow \Phi$$

q_Φ

VI $\forall i \ \mu_i, \sigma_i$

Buffer

q_Φ

$D_{KL}\left(P(\theta \mid h, \{s_t, a_t, s_{t+1}\}) \parallel P(\theta \mid h)\right)$

q_Φ

Reasoning about information gain (approximately)

VIME implementation:

IG can be equivalently written as $D_{\mathrm{KL}}(p(\theta|h, s_t, a_t, s_{t+1})\|p(\theta|h))$

model parameters for $p_\theta(s_{t+1}|s_t, a_t)$

history of all prior transitions

newly observed transition

$q(\theta|\phi) \approx p(\theta|h)$ specifically, optimize variational lower bound $D_{\mathrm{KL}}(q(\theta|\phi)\|p(h|\theta)p(\theta))$

represent $q(\theta|\phi)$ as product of independent Gaussian parameter distributions

with mean ϕ (see Blundell et al. "Weight uncertainty in neural networks")

given new transition (s, a, s'), update ϕ to get ϕ'

i.e., update the network weight means and variances

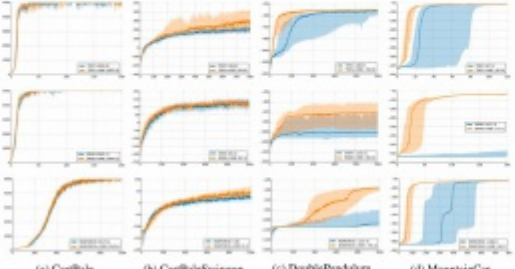
use $D_{\mathrm{KL}}(q(\theta|\phi')\|q(\theta|\phi))$ as approximate bonus

$$p(\theta|\mathcal{D}) = \prod_i p(\theta_i|\mathcal{D})$$

$$p(\theta_i|\mathcal{D}) = \mathcal{N}(\mu_i, \sigma_i)$$

ϕ

Reasoning about information gain (approximately)



Houthooft et al. "VIME"

Approximate IG:

+ appealing mathematical formalism

- models are more complex, generally harder to use effectively