REINFORCEMENT LEARNING: THEORY AND PRACTICE

Exploration and Intrinsic Motivation I

Profs. Amy Zhang and Peter Stone

TEXAS
The University of Texas at Austin

Logistics Questions?

Logistics Questions?

My office hours this week are moved to **today 2-3PM**

Last week

1. State and Temporal Abstractions
2. Options and Hierarchical Reinforcement Learning

This week

Exploration vs. Exploitation
What is the right metric for exploration?
General classes of exploration methods
How those exploration methods generalize to function approximation

How can abstractions help exploration?
How can exploration help abstractions?

What's the problem?

this is easy (mostly)

Why?

this is impossible

Slide credit: Sergey Levine CS 285

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!

Slide credit: Sergey Levine CS 285

Put yourself in the algorithm's shoes

Mao

- "the only rule you may be told is this one"
- Incur a penalty when you break a rule
- Can only discover rules through trial and error
- Rules don't always make sense to you

- Temporally extended tasks like Montezuma's revenge become increasingly difficult based on
  - How extended the task is
  - How little you know about the rules
- Imagine if your goal in life was to win 50 games of Mao…
- (and you didn't know this in advance)

Slide credit: Sergey Levine CS 285

Exploration vs. Exploitation Exercise

What are some examples of exploration vs. exploitation that occur in real life?

Exploration and exploitation examples

- Restaurant selection
  - Exploitation: go to your favorite restaurant
  - Exploration: try a new restaurant
- Online ad placement
  - Exploitation: show the most successful advertisement
  - Exploration: show a different random advertisement
- Oil drilling
  - Exploitation: drill at the best known location
  - Exploration: drill at a new location

Examples from D. Silver lecture notes: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/XX.pdf

Exploration is hard

Can we derive an **optimal** exploration strategy?

what does optimal even mean?

multi-armed bandits (1-step stateless RL problems)	contextual bandits (1-step RL problems)	small, finite MDPs (e.g., tractable planning, model-based RL setting)	large, infinite MDPs, continuous spaces

theoretically tractable ⟷ theoretically intractable

Slide credit: Sergey Levine CS 285

How do we define a *good* exploration strategy?

Discuss!

How do we define a *good* exploration strategy?

Let's start from the simpler bandit setting.

Regret:

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

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

actual reward of action
actually taken

Three broad classes of exploration approaches:

1. Optimistic Exploration
2. Posterior Sampling
3. Information Gain

Go over the basic idea
How do we implement this for large environment/continuous state-action spaces/function approximation?

Optimistic exploration

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

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

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

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

Slide credit: Sergey Levine CS 285

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 $\underbrace{\hat{p}(\theta_1, \ldots, \theta_n)}_{\text{this is a } \textit{model} \text{ of our bandit}}$

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."

Information gain

Bayesian experimental design:

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

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

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

Slide credit: Sergey Levine CS 285

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

Upper Confidence Bound (UCB) Algorithm

$$A_t = \text{argmax}_a \left(Q_t(a) + c\sqrt{\frac{\ln(t)}{N_t(a)}}\right)$$

Exploit

Explore

t = timesteps

$N_t(a)$ = no. of times action (a) is taken

Upper Confidence Bound (UCB) Algorithm

$$A_t = \text{argmax}_a \left(Q_t(a) + c \sqrt{\frac{\ln(t)}{N_t(a)}} \right)$$

Exploit **Explore**

Discussion: How does this objective affect our policy over time ($t \rightarrow \infty$)?

How does small vs. large c affect our policy?

Optimistic exploration in RL

UCB: $$a = \arg\max \hat{\mu}_a + \underbrace{\sqrt{\frac{2 \ln T}{N(a)}}}_{\text{“exploration bonus”}}$$

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

can we use this idea with MDPs?

Slide credit: Sergey Levine CS 285

Optimistic exploration in RL

UCB: $a = \arg\max \hat{\mu}_a + \underbrace{\sqrt{\frac{2 \ln T}{N(a)}}}_{\text{"exploration bonus"}}$

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

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

Slide credit: Sergey Levine CS 285

Optimistic exploration in RL

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

"exploration bonus"

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

See an issue with this?

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

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

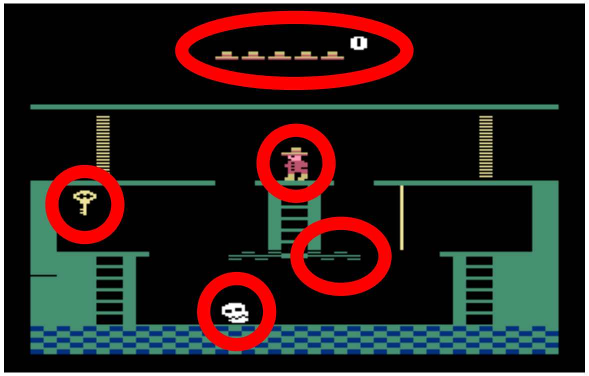
- need to tune bonus weight

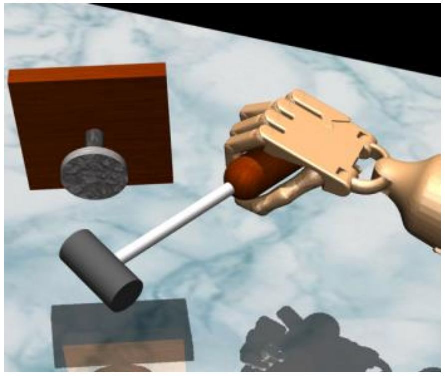
Slide credit: Sergey Levine CS 285

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?



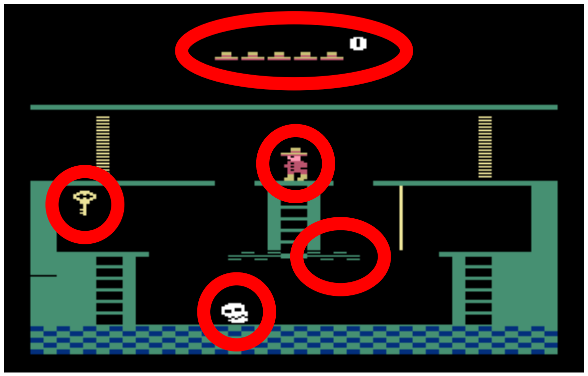


Uh oh… we never see the same thing twice!

But some states are more similar than others

Slide credit: Sergey Levine CS 285

Fitting generative models

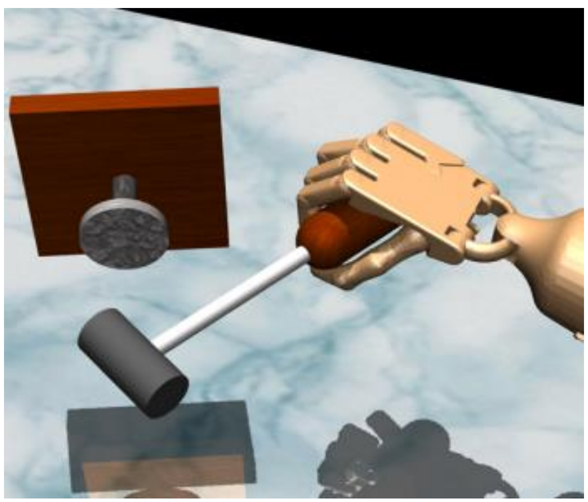


idea: fit a density model $p_\theta(\mathbf{s})$ (or $p_\theta(\mathbf{s}, \mathbf{a})$)

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

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

count

probability/density

total states visited

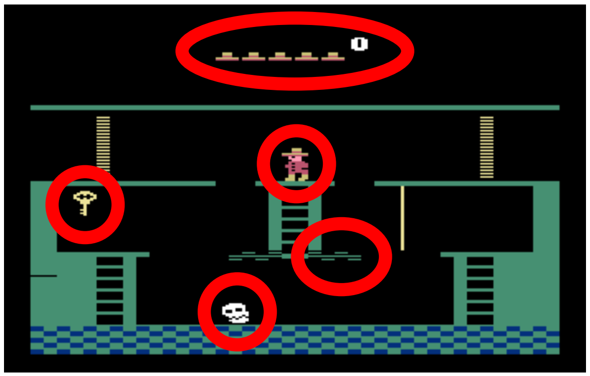
after we see $\mathbf{s}$, we have:

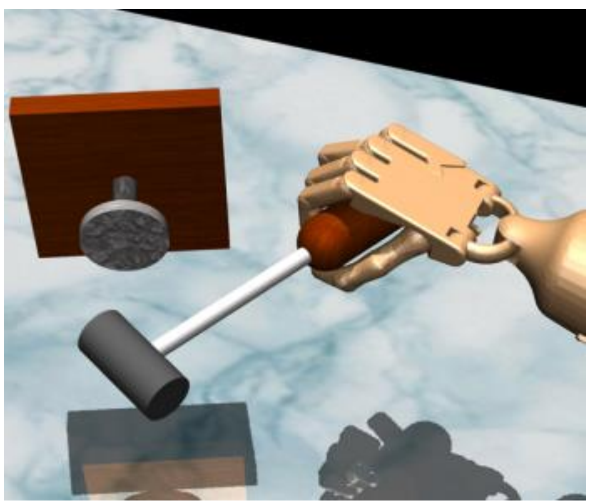
$$P'(\mathbf{s}) = \frac{N(\mathbf{s}) + 1}{n + 1}$$

can we get $p_\theta(\mathbf{s})$ and $p_{\theta'}(\mathbf{s})$ to obey these equations?

Slide credit: Sergey Levine CS 285

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"

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

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

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…"

Slide credit: Sergey Levine CS 285

What kind of bonus to use?

Lots of functions in the literature, inspired by optimal methods for bandits or small MDPs

UCB: $\mathcal{B}(N(\mathbf{s})) = \sqrt{\frac{2 \ln n}{N(\mathbf{s})}}$

MBIE-EB (Strehl & Littman, 2008): $\mathcal{B}(N(\mathbf{s})) = \sqrt{\frac{1}{N(\mathbf{s})}}$

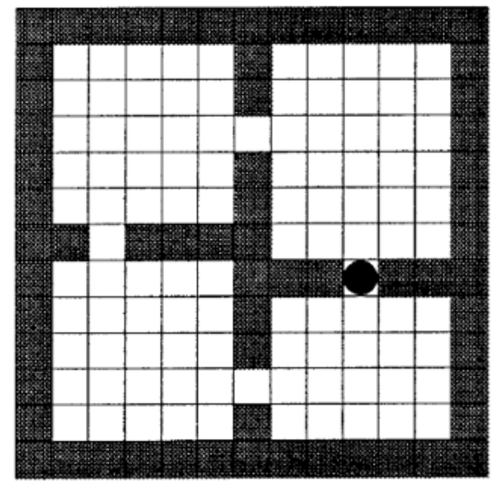
this is the one used by Bellemare et al. '16

BEB (Kolter & Ng, 2009): $\mathcal{B}(N(\mathbf{s})) = \frac{1}{N(\mathbf{s})}$

Slide credit: Sergey Levine CS 285

Count-based exploration exercise

What happens for each of the reward bonuses?

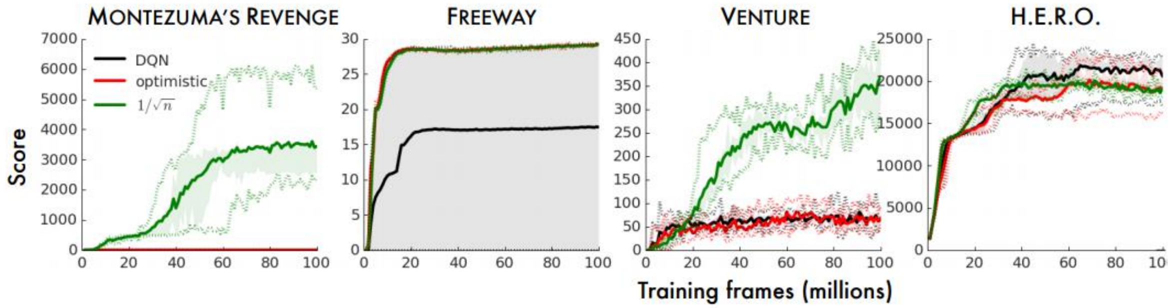


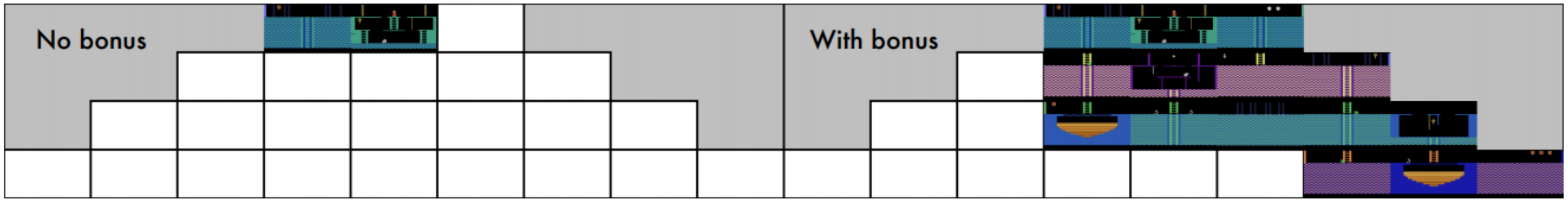
UCB: $$\mathcal{B}(N(\mathbf{s})) = \sqrt{\frac{2\ln n}{N(\mathbf{s})}}$$

MBIE-EB (Strehl & Littman, 2008): $$\mathcal{B}(N(\mathbf{s})) = \sqrt{\frac{1}{N(\mathbf{s})}}$$

BEB (Kolter & Ng, 2009): $$\mathcal{B}(N(\mathbf{s})) = \frac{1}{N(\mathbf{s})}$$

Does it work?

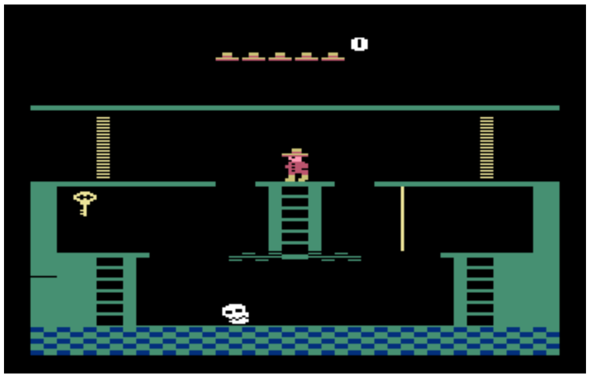


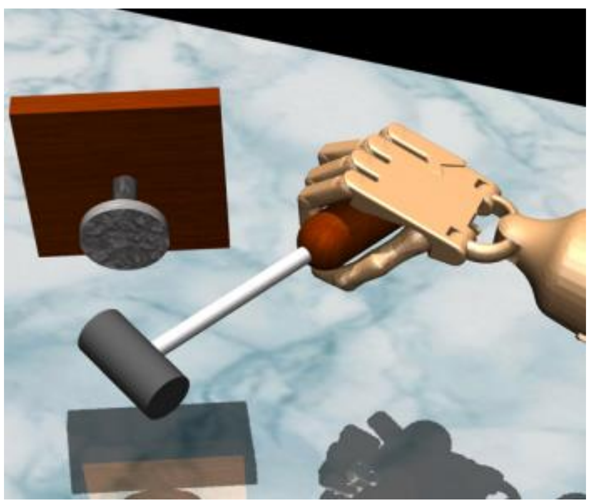


Bellemare et al. "Unifying Count-Based Exploration…"

Slide credit: Sergey Levine CS 285

What kind of model to use?



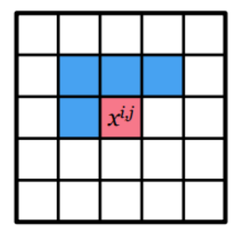


$$p_\theta(\mathbf{s})$$

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

Slide credit: Sergey Levine CS 285

Count-based exploration (Bellemare et al. 2016)

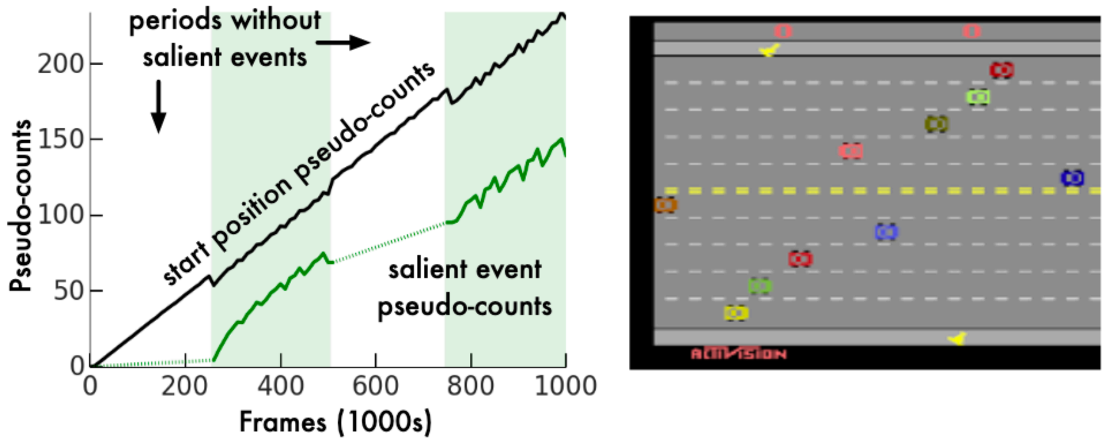


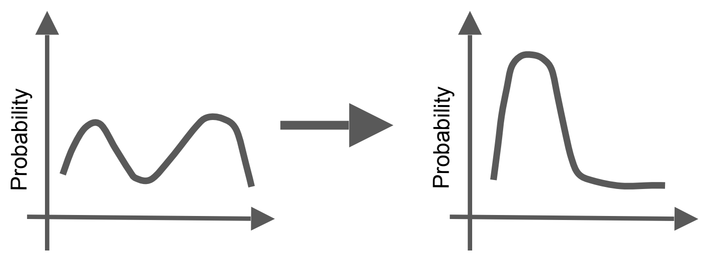
Figure 1: Pseudo-counts obtained from a CTS density model applied to FREEWAY, along with a frame representative of the salient event (crossing the road). Shaded areas depict periods during which the agent observes the salient event, dotted lines interpolate across periods during which the salient event is not observed. The reported values are 10,000-frame averages.

(Alex Chandler) I'm curious about the impact of different density models on the efficiency of exploration and whether there are ways to optimize the choice of density model for specific environments. Is this an empirical question or are there some higher level ideas that might lead to a fitting density model for each environment?

Count-based exploration (Bellemare et al. 2016)

Info gain: KL divergence between prior and posterior (in this case, of the density model) when observing new data

Intuitively: how much does the data change your beliefs?



$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right)$$

A particular choice of pseudo count-based exploration bonus is at least as exploratory as computing a (usually intractable) information gain bonus!

Posterior Sampling in Deep RL

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*

What's the MDP version?

Osband et al. "Deep Exploration via Bootstrapped DQN"

Slide credit: Sergey Levine CS 285

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"

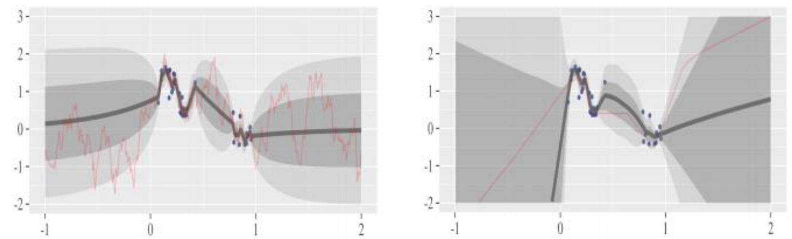
Slide credit: Sergey Levine CS 285

Bootstrap

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

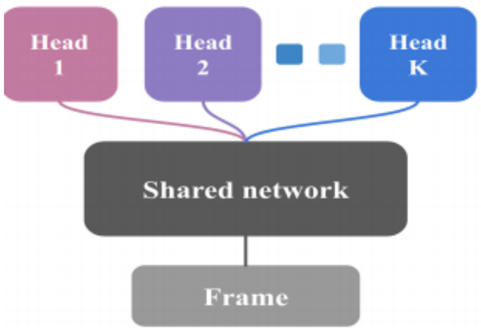
train each model f_{θ_i} on $\mathcal{D}_i$

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



(b) Gaussian process posterior (c) Bootstrapped neural nets

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



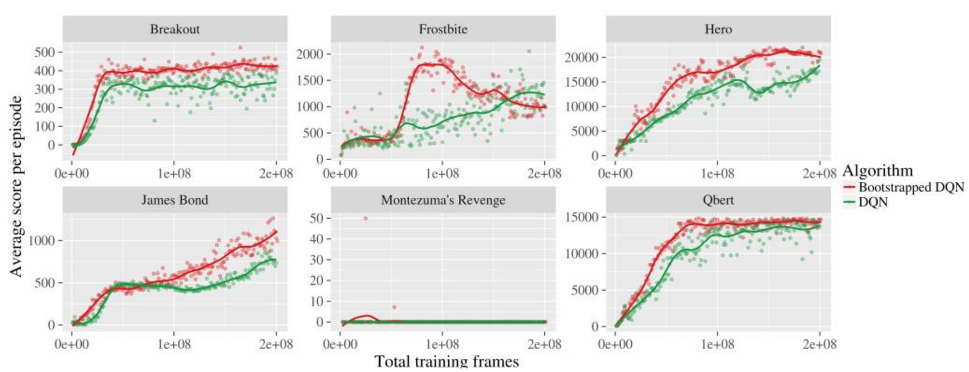
Osband et al. "Deep Exploration via Bootstrapped DQN"

Slide credit: Sergey Levine CS 285

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"

Slide credit: Sergey Levine CS 285

Information Gain in Deep RL

Reasoning about information gain (approximately)

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

information gain about *what*?

Slide credit: Sergey Levine CS 285

Reasoning about information gain (approximately)

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

information gain about *what*?

information gain about reward $r(\mathbf{s}, \mathbf{a})$?	not very useful if reward is sparse
state density $p(\mathbf{s})$?	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

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

Slide credit: Sergey Levine CS 285

Reasoning about information gain (approximately)

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

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

Slide credit: Sergey Levine CS 285

Reasoning about information gain (approximately)

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

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

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

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

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

history of all prior transitions

newly observed transition

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

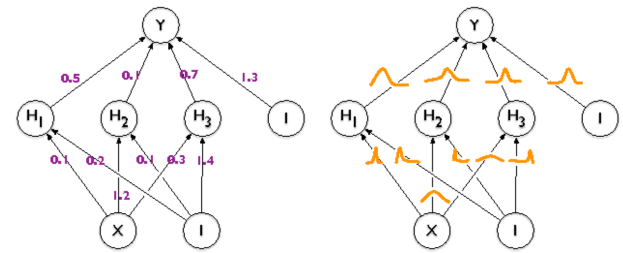
idea: use variational inference to estimate $q(\theta|\phi) \approx p(\theta|h)$

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

Slide credit: Sergey Levine CS 285

Reasoning about information gain (approximately)

VIME implementation:



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

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

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

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

ϕ

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

Houthooft et al. "VIME"

Slide credit: Sergey Levine CS 285

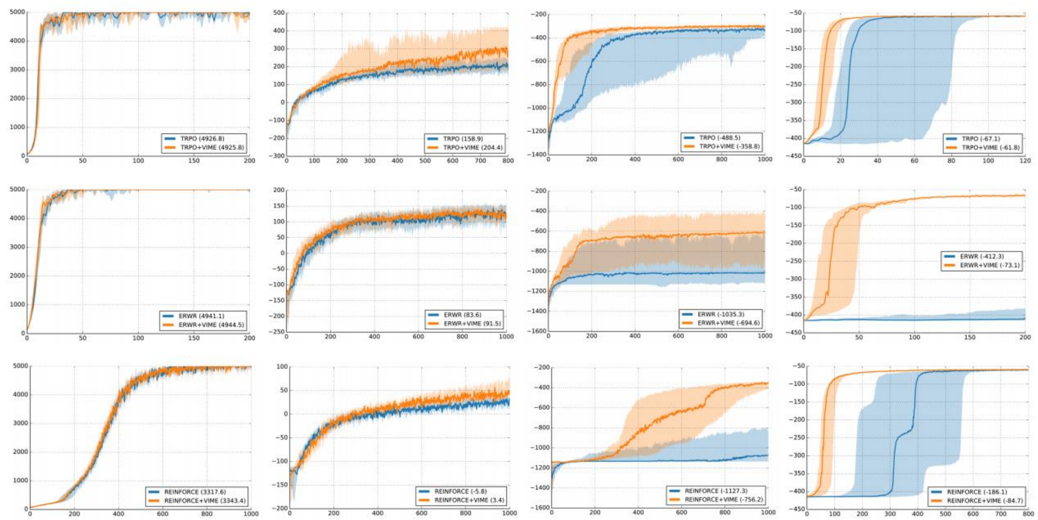
Reasoning about information gain (approximately)

VIME implementation:

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

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

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



(a) CartPole (b) CartPoleSwingup (c) DoublePendulum (d) MountainCar

Approximate IG:

+ appealing mathematical formalism

- models are more complex, generally harder to use effectively

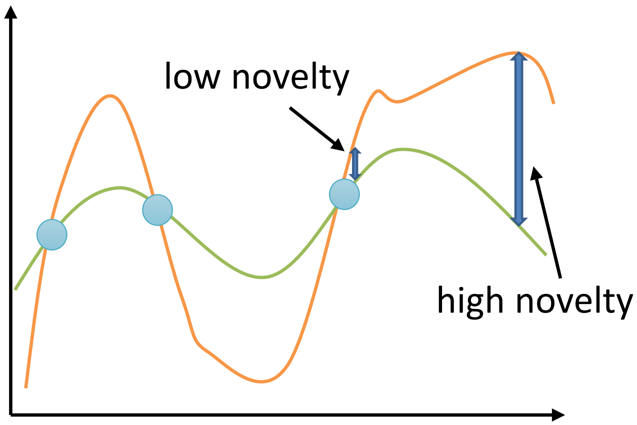
Houthooft et al. “VIME”

Slide credit: Sergey Levine CS 285

Exploration with model errors

$D_{\mathrm{KL}}(q(\theta|\phi')\|q(\theta|\phi))$ can be seen as change in network (mean) parameters ϕ

if we forget about IG, there are many other ways to measure this



Stadie et al. 2015:

- encode image observations using auto-encoder
- build predictive model on auto-encoder latent states
- use model error as exploration bonus

Schmidhuber et al. (see, e.g. "Formal Theory of Creativity, Fun, and Intrinsic Motivation):

- exploration bonus for model error
- exploration bonus for model gradient
- many other variations

Many others!

Slide credit: Sergey Levine CS 285

General themes

UCB:

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

Thompson sampling:

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

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

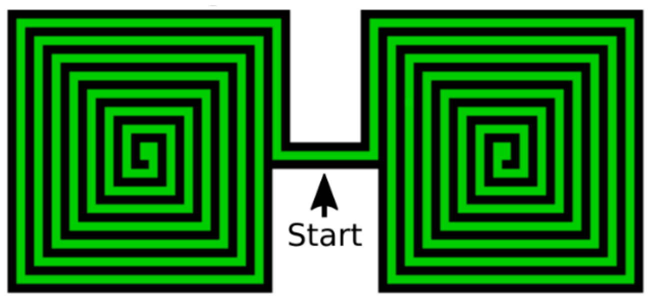
Info gain:

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

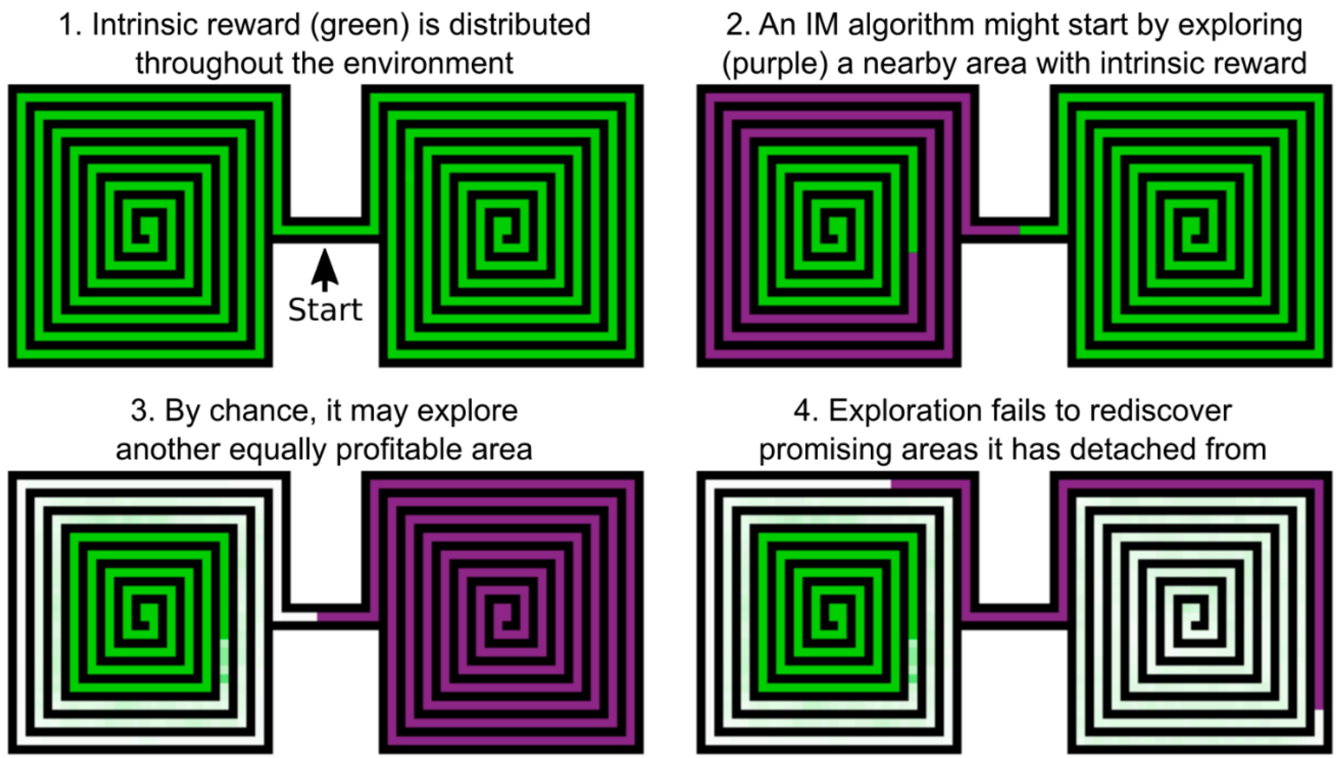
- Most exploration strategies require some kind of uncertainty estimation (even if it's naïve)
- Usually assumes some value to new information
  - Assume unknown = good (optimism)
  - Assume sample = truth
  - Assume information gain = good

Slide credit: Sergey Levine CS 285

What's a possible failure mode of intrinsic motivation?



Go-Explore (Ecoffet et al. 2019)



Go-Explore (Ecoffet et al. 2019)

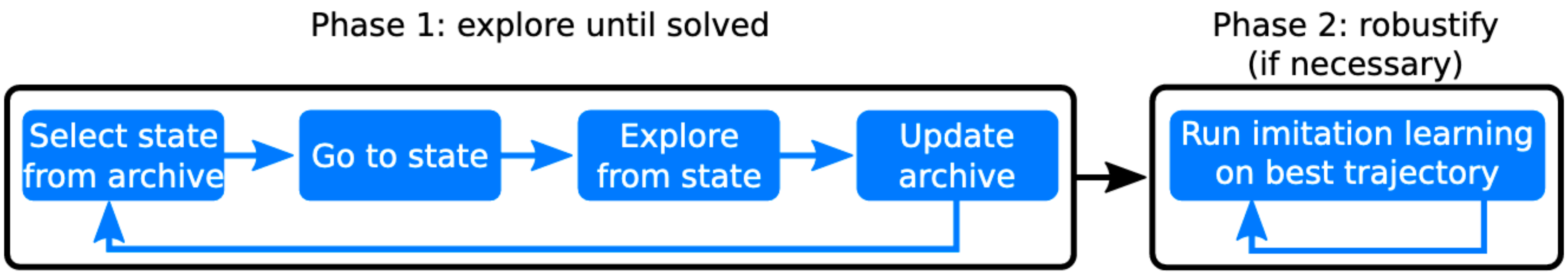


Figure 2: **A high-level overview of the Go-Explore algorithm.**

Discussion Exercise

How can state abstractions or temporal abstractions be combined with exploration?

Go back to the robot navigating to the tower example. What type of state abstraction would make exploration more efficient?

What type of temporal abstraction would make exploration more efficient?

Reading Responses

(Haoran Niu) At the start of the semester you talked about how intermediate rewards were bad because they could cause the agent to learn the wrong things and exploit that reward - why is this different here? How do we know that exploration bonuses won't lead to this exploitation?

Final Logistics

Next lecture: Exploration and Intrinsic Motivation II
We'll cover: reward shaping, DIAYN
Reading assignments due **2PM Monday**

Another reminder: **My office hours are moved to today 2-3PM for this week!**

Final project literature review due at **11:59pm on Thursday, 4/11**