Meta Reinforcement Learning
Learning to Explore

CS 330
Course Reminders

Homework 3, due tonight.

Optional Homework 4 out today.

Project milestone due in two weeks: Wednesday 11/10.
Plan for Today

Recap: Meta-RL Basics

Learning to Explore <- focus of HW4

End-to-End Optimization of Exploration Strategies

Alternative Exploration Strategies

Decoupled Exploration & Exploitation

Lecture goals:

- Understand the challenge of *end-to-end optimization* of exploration
- Understand the basics of using alternative exploration strategies in meta-RL
- Understand & be able to implement *decoupled exploration & exploitation*
Recall: Meta-RL Maze Navigation Example

By learning how to learn many tasks:

\( T_1 \quad T_2 \quad \ldots \quad \text{meta-training tasks} \)

Given a small amount of experience

\( D_{\text{train}} \)

Learn to solve the task

\( S_t \rightarrow a_t \)

diagram adapted from Duan et al. ‘17
Recall: The Meta Reinforcement Learning Problem

Meta Reinforcement Learning:

Inputs: $\mathcal{D}_{\text{train}}$, $s_t$

Outputs: $a_t$

Data: $\{\mathcal{D}_i\}$

dataset of datasets collected for each task

Design & optimization of $f$ *and* collecting appropriate data
(learning to explore)

This lecture: How to learn to collect $\mathcal{D}_{\text{train}}$

Recall: Black-Box Meta-RL

**Black-box network**
(LSTM, NTM, Conv, …)

\[ a_t = f(D_{\text{train}}, s_t; \theta) \]

+ general & expressive
+ a variety of design choices in architecture
-- hard to optimize
~ inherits sample efficiency from outer RL optimizer
Plan for Today

Recap: Meta-RL Basics

Learning to Explore
End-to-End Optimization of Exploration Strategies
Alternative Exploration Strategies
Decoupled Exploration & Exploitation <- focus of HW4
Taking a step back...

In RL, should we be using the same exploration algorithm for:

- Learning to navigate an environment
- Learning to make recommendations to users
- Learning a policy for computer system caching
- Learning to physically operate a new tool or machine

This is how we currently approach exploration.

Can we learn exploration strategies based on experience from other tasks in that domain?
A simple, running example

Hallway 1

Hallway 2

Hallway N

Different tasks: navigating to the ends of different hallways

**Question:** What is one strategy for exploring and learning the task?
How Do We Learn to Explore?

**Solution #1: Optimize for Exploration & Exploitation *End-to-End* w.r.t. Task Reward**

(Duan et al., 2016, Wang et al., 2016, Mishra et al., 2017, Stadie et al., 2018, Zintgraf et al., 2019, Kamienny et al., 2020)

\[ \nabla_\theta J(\theta) = E_{T \sim \pi_\theta, T_i} \left[ \left( \sum_t \nabla_\theta \log \pi_\theta(a_t | s_t, D_i^{tr}) \right) \left( \sum_t r_i(s_t, a_t) \right) \right] \]

**Example episodes during meta-training:**

- agent goes to the end of the correct hallway
  - gets positive reward for current task, but \( D_i^{tr} \) won’t be different than for any other task

- agent goes to wrong hallway then correct hallway
  - +/- provides signal on a **suboptimal** exploration + exploitation strategy

- agent looks at the instructions
  - good exploratory behavior, but won’t get any reward for this behavior

*It’s hard to learn exploration & exploitation at the same time!*
How Do We Learn to Explore?

Solution #1: Optimize for Exploration & Exploitation *End-to-End* w.r.t. Task Reward

(Duan et al., 2016, Wang et al., 2016, Mishra et al., 2017, Stadie et al., 2018, Zintgraf et al., 2019, Kamienny et al., 2020)

+ simple
+ leads to optimal strategy in principle

-- challenging optimization when exploration is hard
Another Example of a Hard Exploration Meta-RL Problem

Learned cooking tasks in previous kitchens

Goal: Quickly learn tasks in a new kitchen.

meta-training

meta-testing
Why is End-to-End Training Hard in This Example?

**End-to-end approach:** optimize exploration and execution episode behaviors end-to-end to maximize reward of execution

**Coupling problem:** learning exploration and execution depend on each other

---

Liu, Raghunathan, Liang, Finn. *Decoupling Exploration and Exploitation for Meta- Reinforcement Learning without Sacrifices*. ICML 2021
Plan for Today

Recap: Meta-RL Basics

Learning to Explore

End-to-End Optimization of Exploration Strategies

Alternative Exploration Strategies

Decoupled Exploration & Exploitation

<- focus of HW4
Solution #2: Leverage Alternative Exploration Strategies

2a. Use posterior sampling
(also called Thompson sampling)

Method Sketch:
1. Learn to solve & collect data for all of the training tasks
   (independent of learning to explore)

2. Form a latent representation $z$ of each task, and a task-conditioned policy $\pi(a | s, z_i)$

3. Learn to infer a distribution over the task: $p(z)$ and $q(z | D_{tr}^i)$

4. Alternate between sampling from current task distribution $\hat{z}_i \sim q(z | D_{tr}^i)$
   and collecting data according to that task with $\pi(a | s, \hat{z}_i)$
   (posterior sampling)
Solution #2: Leverage Alternative Exploration Strategies

2a. Use posterior sampling (also called Thompson sampling)

**Method Sketch:**

1. Learn to solve & collect data for all of the training tasks (independent of learning to explore)

2. Form a latent representation $\mathbf{z}$ of each task, and a task-conditioned policy $\pi(\mathbf{a} \mid \mathbf{s}, z_i)$

3. Learn to infer a distribution over the task: $p(z)$ and $q(z \mid \mathcal{D}_i^{tr})$

4. Alternate between sampling from current task distribution $\hat{z}_i \sim q(z \mid \mathcal{D}_i^{tr})$ and collecting data according to that task with $\pi(\mathbf{a} \mid \mathbf{s}, \hat{z}_i)$ (posterior sampling)
Solution #2: Leverage Alternative Exploration Strategies

2a. Use posterior sampling  
(also called Thompson sampling)

Method Sketch:  
1. Learn to solve & collect data for all of the training tasks  
   (independent of learning to explore)

2. Form a latent representation $\mathbf{z}$ of each task, and a task-conditioned policy $\pi(a \mid s, z_i)$

Use a particular form of black box architecture

$$D_{i}^{tr} \rightarrow \mathbf{z}_i \rightarrow Q_{\theta}(s, a, z) \rightarrow \mathcal{L}_{\text{critic}}$$

$$\pi_{\theta}(a \mid s, z) \rightarrow \mathcal{L}_{\text{actor}}$$

3. Learn to infer a distribution over the task: $p(z)$ and $q(z \mid D_{i}^{tr})$

Impose a distribution on the task encoder:

$$D_{KL} \left( q(z \mid D_{i}^{tr}) \| \mathcal{N}(0, I) \right)$$

Complete objective: $\max_{\theta, \psi} \sum_{i} E_{\tau \sim \pi_{\theta}} [R_i(\tau)] - D_{KL} \left( q_{\psi}(z \mid D_{i}^{tr}) \| \mathcal{N}(0, I) \right)$

2a. Use posterior sampling  
(also called Thompson sampling)

1-3. Learn distribution over latent task variable $\mathbf{z}$ and task-conditioned policy $\pi(a \mid s, z_i)$

4. Sample $\mathbf{z}$ from current posterior $q(\mathbf{z} \mid \mathcal{D}_{tr})$, sample from policy $\pi(a \mid s, \mathbf{z})$, add experience to $\mathcal{D}_{tr}$

**Question:** In what situations might posterior sampling be bad?

eg. Goals far away & sign on wall that tells you the correct goal.
**Solution #2: Leverage Alternative Exploration Strategies**

2a. Use posterior sampling
(also called Thompson sampling)

PEARL (Rakelly, Zhou, Quillen, Finn, Levine. ICML ‘19)

1-3. Learn distribution over latent task variable $z$ and task-conditioned policy $\pi(a | s, z_i)$

4. Sample $z$ from current posterior $q(z | D_{tr})$, sample from policy $\pi(a | s, z)$, add experience to $D_{tr}$

2b. Task dynamics & reward prediction

MetaCURE (Zhang, Wang, Hu, Chen, Fan, Zhang. ‘20)

**Key idea:** Train model $f(s', r | s, a, D_{tr})$ & collect $D_{tr}$ so that model is accurate.

Iteratively, for each task:

1. Collect data $D_{tr}$ with exploration policy $\pi_{exp}$

2. Collect data $D_{ts}$ with policy $\pi_{exe}(a | s, D_{tr})$

3. Train policy $\pi_{exp}$ w.r.t. reward of $r_{exp} = -\| (\hat{s}', \hat{r}) - (s', r) \|^2$
   where $\hat{s}', \hat{r} \sim f(\cdot, \cdot | s, a, D_{tr})$

4. Update policy $\pi_{exe}$ w.r.t. task reward
Solution #2: Leverage Alternative Exploration Strategies

2a. Use posterior sampling
(also called Thompson sampling)

1-3. Learn distribution over latent task variable \( z \) and task-conditioned policy \( \pi(a \mid s, z_i) \)

4. Sample \( z \) from current posterior \( q(z \mid D_{tr}) \), sample from policy \( \pi(a \mid s, z) \), add experience to \( D_{tr} \)

2b. Task dynamics & reward prediction

Key idea: Train model \( f(s', r \mid s, a, D_{tr}) \) & collect \( D_{tr} \) so that model is accurate.

PEARL (Rakelly, Zhou, Quillen, Finn, Levine. ICML ’19)

MetaCURE (Zhang, Wang, Hu, Chen, Fan, Zhang. ’20)

Question: In what situations might this be bad?

Lots of task-irrelevant distractors, or complex, high-dim state dynamics
Solution #2: Leverage Alternative Exploration Strategies

2a. Use posterior sampling
(Also called Thompson sampling)

1-3. Learn distribution over latent task variable $z$ and task-conditioned policy $\pi(a | s, z_i)$

4. Sample $z$ from current posterior $q(z | D_{tr})$, sample from policy $\pi(a | s, z)$, add experience to $D_{tr}$

2b. Task dynamics & reward prediction

Key idea: Train model $f(s', r | s, a, D_{tr})$ & collect $D_{tr}$ so that model is accurate.

PEARL (Rakelly, Zhou, Quillen, Finn, Levine. ICML ’19)

MetaCURE (Zhang, Wang, Hu, Chen, Fan, Zhang. ’20)

Overall

+ **easy** to optimize
+ **many** based on **principled** strategies

-- **suboptimal** by arbitrarily large amount in some environments.
Can we decouple exploitation and exploration *without* sacrificing optimality? (best of both worlds?)
Plan for Today

Recap: Meta-RL Basics

Learning to Explore

End-to-End Optimization of Exploration Strategies

Alternative Exploration Strategies

Decoupled Exploration & Exploitation

<- focus of HW4
Solution #3

Idea from solution #2b: Train model \(f(s', r \mid s, a, \mathcal{D}_{tr})\) & collect \(\mathcal{D}_{tr}\) so that model is accurate.

Do we have to learn a *full dynamics & reward model*?

Idea 3.0: Label each training task with a unique ID \(\mu\)

**Meta training**

**Exploration policy**: train policy \(\pi^{\text{exp}}(a \mid s)\) and task identification model \(q(\mu \mid \mathcal{D}_{tr})\) such that \(\mathcal{D}_{tr} \sim \pi^{\text{exp}}\) allows accurate task prediction from \(f\).

**Execution policy**: train ID-conditioned policy \(\pi^{\text{exec}}(a \mid s, \mu_i)\)

**Meta testing**

Explore: \(\mathcal{D}_{tr} \sim \pi^{\text{exp}}(a \mid s)\)  
Infer task: \(\hat{\mu} \sim q(\mu \mid \mathcal{D}_{tr})\)  
Perform task: \(\pi^{\text{exec}}(a \mid s, \hat{\mu})\)

+ no longer need to model dynamics, rewards  
— may not generalize well for one-hot \(\mu\)
Solution #3: **Decouple** by acquiring representation of task relevant information

1) Learn execution & identify key information

- MDP identifier $\mu$
- Bottlenecked representation
- Execution policy $\pi^{\text{exec}}$

2) Learn to explore by recovering that information

- Exploration policy $\pi^{\text{exp}}$
- Exploration episode $\mathcal{T}$
- Information recovery reward $\text{MI}(z; \mathcal{T})$

---

Liu, Raghunathan, Liang, Finn. *Decoupling Exploration and Exploitation for Meta-Reinforcement Learning without Sacrifices*. ICML 2021
Solution #3: **Decouple** by acquiring representation of task relevant information

1) Learn execution & identify key information

- MDP identifier \( \mu \)
- Wall color [2]
- Ingredients [0]
- Decorations [1]

2) Learn to explore by recovering that information

- Bottlenecked representation \( F(z_i | \mu_i) \)
- Exploration policy \( \pi^{\text{exp}} \)

In practice: (1) and (2) can be trained simultaneously.

Train \( \pi^{\text{exec}}(a | s, z_i) \) and encoder \( F(z_i | \mu_i) \) to:

\[
\max \sum_i \mathbb{E}_{\pi^{\text{exec}}} [r_i] - D_{\text{KL}} (F(z_i | \mu_i) \| \mathcal{N}(0, 1))
\]

Train \( \pi^{\text{exp}} \) such that collected \( \mathcal{D}_{\text{tr}} \) is predictive of \( z_i \).

Liu, Raghunathan, Liang, Finn. *Decoupling Exploration and Exploitation for Meta-Reinforcement Learning without Sacrifices*. ICML 2021
Solution #3: **Decouple** by acquiring representation of task relevant information

Meta-training

1) Learn execution & identify key information

2) Learn to explore by recovering that information

Train $\pi^{\text{exec}}(a | s, z_i)$ and encoder $F(z_i | \mu_i)$ to:

$$\max \sum_i \mathbb{E}_{\pi^{\text{exec}}}[r_i] - D_{\text{KL}}(F(z_i | \mu_i) \| \mathcal{N}(0, 1))$$

Train $\pi^{\exp}$ such that collected $\mathcal{D}_{\text{tr}}$ is predictive of $z_i$.

How to formulate the reward function for $\pi^{\exp}$?

- (a) Train model $q(z_i | \mathcal{D}_{\text{tr}})$
- (b) $r_t = \text{per-step information gain}$

$$r_t = \text{prediction error from } \tau_{1:t-1} - \text{prediction error from } \tau_{1:t}$$

**Decoupled Reward-free Exploration and Execution in Meta-Reinforcement Learning (DREAM)**

Liu, Raghunathan, Liang, Finn. *Decoupling Exploration and Exploitation for Meta-Reinforcement Learning without Sacrifices*. ICML 2021
Solution #3: **Decouple** by acquiring representation of task relevant information

(Informal) Theoretical Analysis

(1) **DREAM** objective is *consistent* with end-to-end optimization. -> can in principle recover the optimal exploration strategy [under mild assumptions]

(2) Consider a bandit-like setting with $|\mathcal{A}|$ arms.

In MDP $i$, arm $i$ yields reward. In all MDPs, arm 0 reveals the rewarding arm.

**RL**\(^2\) requires $\Omega(|\mathcal{A}|^2 \log |\mathcal{A}|)$ samples for meta-optimization.

**DREAM** requires $O(|\mathcal{A}| \log |\mathcal{A}|)$ samples for meta-optimization.

[assuming Q-learning with uniform outer-loop exploration]

---

**Liu, Raghunathan, Liang, Finn.** *Decoupling Exploration and Exploitation for Meta-Reinforcement Learning without Sacrifices.* ICML 2021
How Do We Learn to Explore?

<table>
<thead>
<tr>
<th>End-to-End</th>
<th>Alternative Strategies</th>
<th>Decoupled Exploration &amp; Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ leads to optimal strategy in principle</td>
<td>+ easy to optimize</td>
<td>+ leads to optimal strategy in principle</td>
</tr>
<tr>
<td></td>
<td>+ many based on principled strategies</td>
<td>+ easy to optimize in practice</td>
</tr>
<tr>
<td>- challenging optimization when exploration is hard</td>
<td>- suboptimal by arbitrarily large amount in some environments.</td>
<td>- requires task identifier</td>
</tr>
</tbody>
</table>

Empirical Comparison: Sparse Reward 3D Visual Navigation Problem

- Task: go to the (key / block / ball), color specified by the sign
- Agent starts on other side of barrier, must walk around to read the sign
- Pixels observations (80 x 60 RGB)
- Sparse binary reward

More challenging variant of task from Kamienny et al., 2020

Liu, Raghunathan, Liang, Finn. *Explore then Execute: Adapting without Rewards via Factorized Meta-RL*. ICML 2021
Quantitative Comparison

- **End-to-end algorithms** (*RL²*, *IMPORT*, *VARIBAD*) perform poorly due to coupling.
- **PEARL-UB**: Upper-bound on PEARL: optimal policy and Thompson-Sampling exploration, does not learn the optimal exploration strategy.
- **DREAM** achieves near-optimal reward.

RL² (Duan et al., 2016), IMPORT (Kamienny et al., 2020), VARIBAD (Zintgraf et al., 2019), PEARL (Rakelly, et. al., 2019), Thompson, 1933

Qualitative Results for DREAM

Exploration episode

Execution episode
Goal: Go to key
Plan for Today

Recap: Meta-RL Basics

Learning to Explore <- focus of HW4

End-to-End Optimization of Exploration Strategies

Alternative Exploration Strategies

Decoupled Exploration & Exploitation

Lecture goals:
- Understand the challenge of end-to-end optimization of exploration
- Understand the basics of using alternative exploration strategies in meta-RL
- Understand & be able to implement decoupled exploration & exploitation
Roadmap for Remaining Lectures

Next week:
* Offline Multi-Task RL
* Hierarchical RL
* final RL topics

Following week:
* Guest lectures by Colin Raffel, Jascha Sohl-Dickstein
  * multi-task NLP
  * learned optimizers

Week after that:
* Lifelong Learning
* Frontiers & Open Challenges
  * learning tasks sequentially

Thank you for being an engaging student group!