Lecture outline

- recap: policy gradient RL and how it can be used to build meta-RL algorithms
- the exploration problem in meta-RL
- an approach to encourage better exploration

break

- meta-RL as a POMDP
- an approach for off-policy meta-RL and a different way to explore
Recap: meta-reinforcement learning

“Hula Beach”, “Never grow up”, “The Sled” - by artist Matt Spangler, mattspangler.com
Recap: meta-reinforcement learning

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^{n} \mathcal{L}(\phi_i, D^{ts}_i)$$

where $\phi_i = f_{\theta}(D^{tr}_i)$

Fig adapted from Ravi and Larochelle 2017
Recap: meta-reinforcement learning

Meta-training / outer loop

\[ \theta^* = \arg \min_{\theta} \sum_{i=1}^{n} \mathcal{L}(\phi_i, D_{i}^{ts}) \]

\[ \theta^* = \arg \max_{\theta} \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)] \]

where \( \phi_i = f_{\theta}(D_{i}^{tr}) \)

→ gradient descent

Adaptation / inner loop

→ lots of options

where \( \phi_i = f_{\theta}(M_i) \)

MDP for task \( i \)

“Scooterrific!” by artist Matt Spangler
What’s different in RL?

\[ \theta^* = \arg \min_{\theta} \sum_{i=1}^{n} \mathcal{L}(\phi_i, D_{ts}^i) \]

where \( \phi_i = f_\theta(D_{tr}^i) \)

Adaptation data is given to us!

dalmation  german shepherd  pug

\[ \theta^* = \arg \max_{\theta} \sum_{i=1}^{n} E_{\pi, \phi_i(\tau)} [R(\tau)] \]

where \( \phi_i = f_\theta(M_i) \)

Agent has to collect adaptation data!

“Loser” by artist Matt Spangler
Recap: policy gradient RL algorithms

Direct policy search on $\pi_\theta(a_t|s_t)$

**REINFORCE algorithm:**

1. sample $\{\tau^i\}$ from $\pi_\theta(a_t|s_t)$ (run it on the robot)
2. $\nabla_\theta J(\theta) \approx \sum_i \left( \sum_t \nabla_\theta \log \pi_\theta(a^i_t|s^i_t) \right) \left( \sum_t r(s^i_t, a^i_t) \right)$
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Good stuff is made more likely

Bad stuff is made less likely

Formalizes the idea of “trial and error”
PG meta-RL algorithms: recurrent

Implement the policy as a recurrent network, train across a set of tasks

Persist the hidden state across episode boundaries for continued adaptation!

\[ \theta^* = \arg \max_\theta \sum_{i=1}^n E_{\pi_{\phi_i}}(\tau)[R(\tau)] \]

where \( \phi_i = f_\theta(M_i) \)

Pro: general, expressive
Con: not consistent

PG meta-RL algorithms: gradients

\[ \theta^* = \arg \max_{\theta} \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)] \]

where \( \phi_i = f_\theta(M_i) \)

Pro: consistent!
Con: not expressive

Q: Can you think of an example in which recurrent methods are more expressive?

Finn et al. 2017. Fig adapted from Finn et al. 2017
How these algorithms learn to explore

Pre-update parameters receive credit for producing good exploration trajectories

Causal relationship between pre and post-update trajectories is taken into account

Credit assignment

Figure adapted from Rothfuss et al. 2018
How well do they explore?

Recurrent approach explores in a new maze (goal is to navigate from blue to red square)

Gradient-based approach explores in a point robot navigation task

Fig adapted from RL2. Duan et al. 2016

Fig adapted from ProMP Rothfuss et al. 2017
How well do they explore?

Here gradient-based meta-RL fails to explore in a sparse reward navigation task

Fig adapted from MAESN. Gupta et al. 2018
What’s the problem?
What’s the problem?

Exploration requires stochasticity, optimal policies don’t.

Typical methods of adding noise are time-invariant.
Temporally extended exploration

\[ \theta^* = \arg \max_{\theta} \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)] \]

where \( \phi_i = f_{\theta}(M_i) \)

Sample \( z \), hold constant during episode

Adapt \( z \) to a new task with gradient descent

Pre-adaptation: good exploration
Post-adaptation: good task performance

Figure adapted from Gupta et al. 2018
Temporally extended exploration with MAESN

MAML Exploration

MAESN exploration

MAESN, Gupta et al. 2018
Meta-RL desiderata

<table>
<thead>
<tr>
<th>recurrent</th>
<th>gradient</th>
<th>structured exp</th>
</tr>
</thead>
</table>

Fig adapted from Chelsea Finn
<table>
<thead>
<tr>
<th></th>
<th>recurrent</th>
<th>gradient</th>
<th>structured exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>consistent</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
## Meta-RL desiderata

<table>
<thead>
<tr>
<th></th>
<th>recurrent</th>
<th>gradient</th>
<th>structured exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>consistent</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>expressive</td>
<td>✔️</td>
<td>❌</td>
<td>❌</td>
</tr>
</tbody>
</table>

Fig adapted from Chelsea Finn
## Meta-RL desiderata

<table>
<thead>
<tr>
<th></th>
<th>recurrent</th>
<th>gradient</th>
<th>structured exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>consistent</td>
<td>✘</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>expressive</td>
<td>✔</td>
<td>✘</td>
<td>✘</td>
</tr>
<tr>
<td>structured exploration</td>
<td>~</td>
<td>~</td>
<td>✔</td>
</tr>
</tbody>
</table>

Fig adapted from Chelsea Finn
### Meta-RL Desiderata

<table>
<thead>
<tr>
<th></th>
<th>recurrent</th>
<th>gradient</th>
<th>structured exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>consistent</td>
<td>✘</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>expressive</td>
<td>✔</td>
<td>✘</td>
<td>✘</td>
</tr>
<tr>
<td>structured exploration</td>
<td>~</td>
<td>~</td>
<td>✔</td>
</tr>
<tr>
<td>efficient &amp; off-policy</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
</tr>
</tbody>
</table>

In single-task RL, off-policy algorithms 1-2 orders of magnitude more efficient! Huge difference for real-world applications (1 month -> 10 hours)

Fig adapted from Chelsea Finn
Why is off-policy meta-RL difficult?

Key characteristic of meta-learning: the conditions at meta-training time should closely match those at test time!

\[ \theta^* = \arg \max_{\theta} \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)] \]

where \( \phi_i = f_{\theta}(M_i) \)

Note: this is very much an unresolved question
Break
PEARL
Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables

Kate Rakelly*, Aurick Zhou*, Deirdre Quillen, Chelsea Finn, Sergey Levine
Aside: POMDPs

observation gives incomplete information about the state

state is unobserved (hidden)

Example: incomplete sensor data

“That Way We Go” by Matt Spangler
The POMDP view of meta-RL

Can we leverage this connection to design a new meta-RL algorithm?
Model belief over latent task variables

**POMDP for unobserved state**

- Goal state
- Where am I?
- $p(h|c)$
- $s = S_0$
- $a = \text{“left”}, s = S_0, r = 0$

**POMDP for unobserved task**

- Goal for MDP 0
- Goal for MDP 1
- Goal for MDP 2
- What task am I in?
- $p(z|c)$
- $s = S_0$
- $a = \text{“left”}, s = S_0, r = 0$
Model belief over latent task variables

POMDP for unobserved state

- Goal state
- Where am I?
- \( p(h|c) \)
- Goal for MDP 0
- Goal for MDP 1
- Goal for MDP 2
- What task am I in?
- \( p(z|c) \)
- a = “left”, s = S0, r = 0
- s = S0

POMDP for unobserved task

- s = S0
- sample
- a = “left”, s = S0, r = 0
RL with task-belief states

How do we learn this in a way that generalizes to new tasks?

“Task” can be supervised by reconstructing states and rewards

OR

By minimizing Bellman error
Meta-RL with task-belief states

\[ \theta^* = \arg \max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)] \]

where \( \phi_i = f_\theta(M_i) \)

Stochastic encoder
Posterior sampling in action
Meta-RL with task-belief states

\[ \theta^* = \arg \max_{\theta} \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)] \]

where \( \phi_i = f_{\theta}(M_i) \)

Stochastic encoder

Variational approximations to posterior and prior

“Likelihood” term (Bellman error)

\[ \mathbb{E}_{T} \left[ \mathbb{E}_{z \sim q_{\phi}(z|c^T)} \left[ R(T, z) + \beta D_{KL}(q_{\phi}(z|c^T) \parallel p(z)) \right] \right] \]

“Regularization” term / information bottleneck

See Control as Inference (Levine 2018) for justification of thinking of Q as a pseudo-likelihood.
Encoder design

Don’t need to know the order of transitions in order to identify the MDP (Markov property)

Use a permutation-invariant encoder for simplicity and speed
Aside: Soft Actor-Critic (SAC)

“Soft”: Maximize rewards *and* entropy of the policy (higher entropy policies explore better)

\[
J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t) \sim \rho_\pi} [r(s_t, a_t) + \alpha H(\pi(\cdot|s_t))]
\]

“Actor-Critic”: Model *both* the actor (aka the policy) and the critic (aka the Q-function)

\[
J_Q(\theta) = \mathbb{E}_{(s_t,a_t) \sim D} \left[ \frac{1}{2} \left( Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t) \right)^2 \right]
\]

\[
J_\pi(\phi) = \mathbb{E}_{s_t,a_t} [Q_\theta(s_t, a_t) + \alpha H(\pi(\cdot|s_t))]
\]

Dclaw robot turns valve from pixels

Soft Actor-Critic
Integrating task-belief with SAC

\[ \theta^* = \arg \max_{\theta} \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)] \]

SAC where \( \phi_i = f_\theta(M_i) \)

Stochastic encoder

\[ q_\phi(z|c) \]

\[ N(0, I) \]

\[ D_{KL} \]

\[ L_{critic} \]

\[ L_{actor} \]
Meta-RL experimental domains

Simulated via MuJoCo (Todorov et al. 2012), tasks proposed by (Finn et al. 2017, Rothfuss et al. 2019)
ProMP (Rothfuss et al. 2019), MAML (Finn et al. 2017), RL2 (Duan et al. 2016)
ProMP (Rothfuss et al. 2019), MAML (Finn et al. 2017), RL2 (Duan et al. 2016)

20-100X more sample efficient!
Separate task-Inference and RL data

- **On-policy**
  - Replay buffer $S_C$
  - Function $\phi$
  - Distribution $q_\phi(z|c)$
  - KL Divergence $D_{KL}$
  - Normal distribution $\mathcal{N}(0, I)$

- **Off-policy**
  - Train tasks
  - Function $\pi_\theta(a|s, z)$
  - Function $Q_\theta(s, a, z)$
  - Loss functions $\mathcal{L}_{\text{critic}}$ and $\mathcal{L}_{\text{actor}}$
Limits of posterior sampling

Posterior sampling exploration strategy

Optimal exploration strategy
Limits of posterior sampling

MAESN (pre-adapted z constrained)  PEARL (post-adapted z constrained)

Prior distribution (pre-adaptation)
Posterior distribution (post-adaptation)
Summary

- Building on policy gradient RL, we can implement meta-RL algorithms via a recurrent network or gradient-based adaptation.
- Adaptation in meta-RL includes both exploration as well as learning to perform well.
- We can improve exploration by conditioning the policy on latent variables held constant across an episode, resulting in temporally-coherent strategies.

Break

- Meta-RL can be expressed as a particular kind of POMDP.
- We can do meta-RL by inferring a belief over the task, explore via posterior sampling from this belief, and combine with SAC for a sample efficient alg.
Explicitly Meta-Learn an Exploration Policy

Instantiate separate teacher (exploration) and student (target) policies

Train the exploration policy to maximize the increase in rewards earned by the target policy after training on the exploration policy’s data

\[ \hat{R}(\pi, D_0) = \hat{R}_{\pi'} - \hat{R}_\pi \]

State visitation for student and teacher

Learning to Explore via Meta Policy Gradient, Xu et al. 2018
Fast Reinforcement Learning via Slow Reinforcement Learning (RL2) (Duan et al. 2016), Learning to Reinforcement Learn (Wang et al. 2016), Memory-Based Control with Recurrent Neural Networks (Heess et al. 2015) - recurrent meta-RL

Model-Agnostic Meta-Learning (MAML) (Finn et al. 2017), Proximal Meta-Policy Gradient (ProMP) (Rothfuss et al. 2018) - gradient-based meta-RL (see ProMP for a breakdown of the gradient terms)

Meta-Learning Structured Exploration Strategies (MAESN) (Gupta et al. 2018) - temporally extended exploration with latent variables and MAML

Efficient Off-Policy Meta-RL via Probabilistic Context Variables (PEARL) (Rakelly et al. 2019) - off-policy meta-RL with posterior sampling

Soft Actor-Critic (Haarnoja et al. 2018) - off-policy RL in the maximum entropy framework

Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review (Levine 2018) - a framework for control as inference, good background for understanding SAC

(More) Efficient Reinforcement Learning via Posterior Sampling (Osband et al. 2013) - establishes a worse-case regret bound for posterior sampling that is similar to optimism-based exploration approaches
Further Reading

**Stochastic Latent Actor-Critic (SLAC)** (arXiv 2019) - do SAC in a latent state space inferred from image observations

**Meta-Learning as Task Inference** (arXiv 2019) - similar idea to PEARL and investigates different objectives to use for training the latent task space

**VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning** (arXiv 2019) - similar idea to PEARL and updates the latent state at every timestep rather than every trajectory, learns latent space a bit differently

**Deep Variational Reinforcement Learning for POMDPs** (Igl. et al. 2018) - variational inference approach for solving general POMDPs

**Some Considerations on Learning to Explore with Meta-RL** (Stadie et al. 2018) - does MAML but treats the adaptation step as part of the unknown dynamics of the environment (see ProMP for a good explanation of this difference)

**Learning to Explore via Meta-Policy Gradient** (Xu et al. 2018) - a different problem statement of learning to explore in a *single* task, an interesting approach of training the exploration policy based on differences in rewards accrued by the target policy