Offline Meta Reinforcement Learning

Ron Dorfman
M.Sc student, EE Department
Technion - Israel Institute of Technology

Joint work with Prof. Aviv Tamar
How to reach the goal?
What if goal location is unknown, but we have data from agents trained to reach different goals, all lie on a semi-circle?
Sample possible goal
1. Sample possible goal
2. Pretend goal is correct and plan
1. Sample possible goal
2. Pretend goal is correct and plan
3. Execute, observe evidence and update possible goals
Thompson Sampling

1. Sample possible goal
2. Pretend goal is correct and plan
3. Execute, observe evidence and update possible goals
Is that the optimal thing to do?
Is that the optimal thing to do? No!
Search optimally across semi-circle
1. Search optimally across semi-circle
2. Go to found goal
Bayes-optimal exploration

1. Search optimally across semi-circle
2. Go to found goal
Can we use collected data to learn Bayes-optimal behavior?
Can we use collected data to learn Bayes-optimal behavior?

Suppose we can.. Why is it important?
Exploration generally requires **online** data collection.
Exploration generally requires **online** data collection.

Data collection can be expensive/unsafe: Robotics, Healthcare, AD, ...
Exploration generally requires **online** data collection.

Data collection can be expensive/unsafe: Robotics, Healthcare, AD, ...

Learn to explore from **offline** data
Reinforcement Learning (RL)

- Markov Decision Process (MDP) $\mathcal{M} = (S, A, R, P)$.

- $S$ – state space
- $A$ – action space
- $R$ – reward function
- $P$ – transition function

**Goal:** Find policy $\pi$ that maximizes $E\left[\sum_{t=0}^{H} R(s_t, a_t)\right]$.

- There exists an optimal policy $\pi^*$ which is Markov, i.e., $\pi^* : S \rightarrow A$. 
How to discover high reward strategies?
How to discover high reward strategies?

No prior information

- UCRL
- $E^3$
- R-max
- Exploration bonuses
- Count-based
- ...
How to discover high reward strategies?

No prior information

- UCRL
- $E^3$
- R-max
- Exploration bonuses
- Count-based
- ...

Efficiently search state space

Regret bounds, PAC bounds, ...
How to discover high reward strategies?

**No prior information**
- UCRL
- $E^3$
- R-max
- Exploration bonuses
- Count-based
- ...

Efficiently search state space

**Prior over MDPs**
- Bayesian RL

Optimal exploration

Regret bounds, PAC bounds, ...
Bayesian RL (BRL)

- Prior distribution over MDP parameters, $p(R, P)$.
- **Goal:** Find policy $\pi$ that maximizes
  \[
  \mathbb{E}_{R, P \sim p(\cdot, \cdot)} \left[ \sum_{t=0}^{H} R(s_t, a_t) \right].
  \]

- In general, the optimal policy is **history-dependent**.
- Optimally balance exploration-exploitation: *An optimal agent takes actions that reduce its uncertainty, only if such leads to higher rewards.*
BRL as Partially-Observed MDP

- \( R, P \) are unobserved variables.
- Collect samples:
  \[
  h_t = (s_0, a_0, r_1, s_1, \ldots, r_t, s_t)
  \]
- Maintain belief:
  \[
  b_{t+1}(R, P) = P(R, P|h_{t+1}) \propto P(s_{t+1}, r_{t+1}|h_t, R, P)b_t(R, P),
  \]
  \[
  b_0(R, P) = p(R, P).
  \]
- **Bayes-optimal policy** is of the form \( \pi^*(s, b) \).
Bayes-Adaptive MDP (BAMDP)

- **Hyper-state space:**
  \[ S^+ = S \times B \]

- **Transition function:**
  \[
P^+(s^+_{t+1} | s^+_t, a_t) = \mathbb{E}_{b_t} \left[ P(s_{t+1} | s_t, a_t) \right] \delta(b_{t+1} = P(R, P|h_{t+1}))
  \]

- **Reward function:**
  \[
  R^+(s^+_t, a_t) = \mathbb{E}_{b_t} \left[ R(s_t, a_t) \right]
  \]

- **Maximizing the BRL objective amounts to solving the BAMDP!**

---


- Intractable posterior update
- Intractable planning in belief space
Variational Bayes-Adaptive Deep RL (VariBAD)

Approximate using Meta-RL and Variational Inference.

- Train variational autoencoder to approximate belief.
- Train on-policy RL agent, conditioned on belief.
Given access to train tasks $\mathcal{M}_1, \ldots, \mathcal{M}_N \sim p(\mathcal{M}) = p(\mathcal{R}, \mathcal{P})$.

Describe MDP $\mathcal{M}_i$ with learned latent variable $m_i$:

\[
\mathcal{P}_i(s_{t+1}|s_t, a_t) \approx \mathcal{P}(s_{t+1}|s_t, a_t, m_i) \\
\mathcal{R}_i(s_t, a_t) \approx \mathcal{R}(s_t, a_t|m_i)
\]

Infer $m_i$ by interaction with $\mathcal{M}_i$:

\[
p(m_i|\tau^i) = P(m_i|s^i_0, a^i_0, r^i_1, s^i_1, \ldots, r^i_t, s^i_t).
\]

Model trajectories using variational autoencoder.

- Generative model:

\[
P(\tau_s^r, a_{H-1} | a_{H-1}) = \int p_\theta(m) p_\theta(\tau_s^r | m, a_{H-1}) dm
\]

- Approximate posterior:

\[
q_\phi(m | \tau:t) = \mathcal{N}(\mu(\tau:t), \Sigma(\tau:t))
\]

- Variational lower bound (ELBO):

\[
\log P(\tau_s^r, a_{H-1}) \geq \mathbb{E}_{q_\phi(m | \tau:t)} \left[ \log p_\theta(\tau_s^r | m, a_{H-1}) \right] - D_{KL}(q_\phi(m | \tau:t) \| p_\theta(m)) \\
\equiv ELBO_t
\]
For $i = 1, \ldots, N$:
Collect trajectories from $\mathcal{M}_i$

Optimize $\mathcal{L}_{RL} + \lambda \mathcal{L}_{VAE}$

Can we use this to solve BRL offline?

1. For $i = 1, \ldots, N$: Collect trajectories from $\mathcal{M}_i$
2. Optimize $\mathcal{L}_{RL} + \lambda \mathcal{L}_{VAE}$
In this work, offline data is **entire** training histories of RL agents.
- Train VAE using trajectories in data.
Train VAE using trajectories in data.

Relabel states:

1. Run encoder on every partial trajectory $\tau_t$. Obtain $b_t \approx (\mu_t, \Sigma_t)$.
2. Replace each $s_t$ in data with $s_t^+ = (s_t, \mu_t, \Sigma_t)$. 
Can’t use on-policy RL in offline setting! For off-policy, need tuples \((s, a, r, s')\)

**Ideal**
\[
\begin{align*}
    r & \sim \mathcal{R}^+ = \mathbb{E}_b\mathcal{R} \\
    s' & \sim \mathcal{P}^+ = \mathbb{E}_b\mathcal{P}
\end{align*}
\]

**Reality**
\[
\begin{align*}
    r & \sim \mathcal{R}_M \\
    s' & \sim \mathcal{P}_M, M \sim p(M)
\end{align*}
\]
Can’t use on-policy RL in offline setting! For off-policy, need tuples \((s, a, r, s')\)

**Ideal**

\[
\begin{align*}
r &\sim \mathcal{R}^+ = \mathbb{E}_b\mathcal{R} \\
s' &\sim \mathcal{P}^+ = \mathbb{E}_b\mathcal{P}
\end{align*}
\]

**Reality**

\[
\begin{align*}
r &\sim \mathcal{R}_{\mathcal{M}} \\
s' &\sim \mathcal{P}_{\mathcal{M}}, \mathcal{M} \sim p(\mathcal{M})
\end{align*}
\]
Can’t use on-policy RL in offline setting! For off-policy, need tuples \((s, a, r, s')\)

**Ideal**
- \(r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}\)
- \(s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}\)

**Reality**
- \(r \sim \mathcal{R}_M\)
- \(s' \sim \mathcal{P}_M\)
- \(\mathcal{M} \sim p(\mathcal{M})\)
Can’t use on-policy RL in offline setting! For off-policy, need tuples \((s, a, r, s')\)

**Ideal**

\[
\begin{align*}
  r &\sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R} \\
  s' &\sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}
\end{align*}
\]

**Reality**

\[
\begin{align*}
  r &\sim \mathcal{R_M} \\
  s' &\sim \mathcal{P_M} , \mathcal{M} \sim p(\mathcal{M})
\end{align*}
\]
Can’t use on-policy RL in offline setting! For off-policy, need tuples $(s, a, r, s')$

### Ideal

- $r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$
- $s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$

### Reality

- $r \sim \mathcal{R}_M$
- $s' \sim \mathcal{P}_M$
- $M \sim p(M)$

Time

MDPs
Can’t use on-policy RL in offline setting! For off-policy, need tuples \((s, a, r, s')\)

**Ideal**
\[
\begin{align*}
    r & \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R} \\
    s' & \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}
\end{align*}
\]

**Reality**
\[
\begin{align*}
    r & \sim \mathcal{R}_M \\
    s' & \sim \mathcal{P}_M , M \sim p(M)
\end{align*}
\]
Can’t use on-policy RL in offline setting! For off-policy, need tuples \((s, a, r, s')\)

**Ideal**

\[
\begin{align*}
    r & \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R} \\
    s' & \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}
\end{align*}
\]

**Reality**

\[
\begin{align*}
    r & \sim \mathcal{R}_M \\
    s' & \sim \mathcal{P}_M, \mathcal{M} \sim p(\mathcal{M})
\end{align*}
\]

Equivalent!
Can’t use on-policy RL in offline setting! For off-policy, need tuples \((s, a, r, s')\)

**Ideal**

\[
\begin{align*}
    r &\sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R} \\
    s' &\sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}
\end{align*}
\]

**Reality**

\[
\begin{align*}
    r &\sim \mathcal{R}_\mathcal{M} \\
    s' &\sim \mathcal{P}_\mathcal{M}, \mathcal{M} \sim p(\mathcal{M})
\end{align*}
\]

---

**Proposition**

Sample \(\mathcal{R}, \mathcal{P} \sim p(\mathcal{R}, \mathcal{P})\). **Collect trajectory according to**

\[
\begin{align*}
    a_t &\sim \pi(\cdot|h; t) \\
    s_{t+1} &\sim \mathcal{P}(\cdot|s_t, a_t) \text{ and } r_{t+1} \sim \mathcal{R}(\cdot|s_t, a_t).
\end{align*}
\]

Then,

\[
\begin{align*}
    \mathbb{P}(s_{t+1}|s_0, a_0, r_1, \ldots, s_t, a_t) &= \mathbb{E}_{\mathcal{R}, \mathcal{P}\sim b_t} \mathcal{P}(s_{t+1}|s_t, a_t), \\
    \mathbb{P}(r_{t+1}|s_0, a_0, r_1, \ldots, s_t, a_t) &= \mathbb{E}_{\mathcal{R}, \mathcal{P}\sim b_t} \mathcal{R}(r_{t+1}|s_t, a_t).
\end{align*}
\]
Conclusion

Any off-policy RL algorithm can be used on the hyper-state tuples in our data.
Conclusion

Any off-policy RL algorithm can be used on the hyper-state tuples in our data.

Is that it?
The MDP Ambiguity Problem

Two different MDPs or a single MDP with rewards at both circles?
Two different MDPs or a single MDP with rewards at both circles?

Unique problem to the offline Meta-RL setting!
Two different MDPs or a single MDP with rewards at both circles?

Unique problem to the offline Meta-RL setting!

During the VAE training
**Problem:** For each MDP, different part of state space is visited.
Reward Relabelling

- **Problem:** For each MDP, different part of state space is visited.
- Make state distribution uniform across MDPs:

  1. Let $\tau^i = (s^i_0, a^i_0, r^i_1, s^i_1, \ldots, r^i_H, s^i_H)$ from $\mathcal{M}_i$.
  2. Sample randomly $i' \neq i$. Relabel rewards:

        $\hat{\tau}^i = (s^i_0, a^i_0, \hat{r}^i_1, s^i_1, \ldots, \hat{r}^i_H, s^i_H)$

        where $\hat{r}^i_{t+1} = R_{i'}(s^i_t, a^i_t)$

- Requires access to $R_i$ for each $\mathcal{M}_i$. 
Our Method
Our Method

(1) Data Collection

Agent 1
Agent 2
Agent N

Offline Data

Task 1
Task 2
Task N

(2) VAE Training

Reward Relabeling
Our Method

(1) Data Collection

(2) VAE Training

(3) Off-policy RL

Meta-RL agent

Offline Data

Task 1

Task 2

Task N

Reward Relabeling

State Relabeling: $s_t \rightarrow (s_t, \mu_t, \Sigma_t)$
Illustrative Domains

Gridworld

Semi-circle
Illustrative Domains - Performance

- Gridworld
- Semi-circle

![Graphs showing performance comparison between different methods (Ours, Ours, w/o RR, Thompson Sampling, Online PEARL) across train steps.](Image)
Semi-circle
Semi-circle - Belief Visualization

Belief update: when reaching the goal

Belief update: when searching for the goal

Belief update: reward relabelling ablation
MuJoCo Domains

Half-Cheetah-Vel

Ant-Semi-circle
MuJoCo Domains - Performance

Half-Cheetah-Vel

![Graph showing the performance of different algorithms in Half-Cheetah-Vel domain.]

Ant-Semi-circle

![Graph showing the performance of different algorithms in Ant-Semi-circle domain.]

- **Ours**
- **Ours, w/o RR**
- **Online PEARL**
Ant-Semi-circle

Episode 1

Episode 2

Ron Dorfman
Offline Meta Reinforcement Learning
September 25, 2020
Online Setting Performance
Summary

- Formalized offline Meta-RL as BRL.
- Demonstrated learning an approximately Bayes-optimal policy.
- Sample efficient off-policy RL optimization.