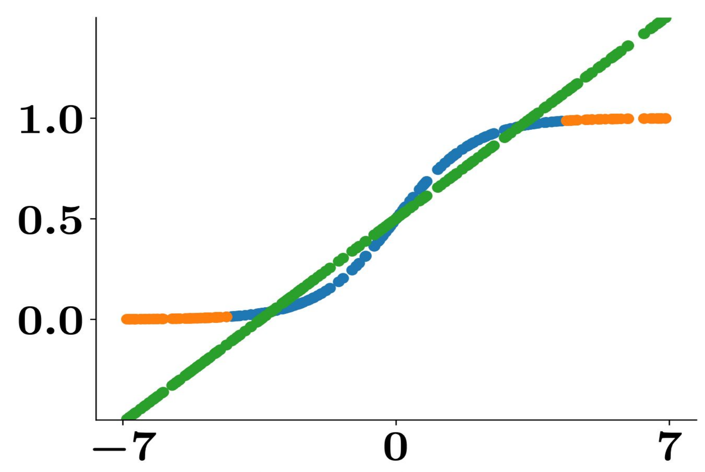
Learning to Switch Among Agents in a Team

Vahid Balazadeh Meresht[1] Abir De[2] Adish Singla[3] Manuel Gomez Rodriguez[3]

[1] Sharif University of Technology [2] Indian Institute of Technology Bombay [3] Max Planck Institute for Software Systems

1. Motivation

- Reinforcement learning agents, in comparison to humans, are:
  - Better in simulated environments like video games
  - Worse in cyber-physical systems like autonomous driving
- We may deploy RL agents under lower automation levels, and switch to the human in difficult states.
- When should we switch control among agents? We should also be able to:
  - Control the level of automation
  - Control the number of switches
  - Learn the unknown human driver



De et al. 2020

2. Setting

- MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, C, L)$, set of agents $\mathcal{D}$ with action policy $p_d(a_t \mid s_t)$
- Cost of trajectory $\{(s_t, d_t, a_t)\}_{t=1}^{L}$: $\sum_{t=1}^{L} c'(s_t, a_t) + c_c(d_t) + c_x(d_t, d_{t-1})$
- Switching policy depends on the current state and previous controller:

$$d_t = \pi_t(s_t, d_{t-1})$$

- Value function:

$$V^{\pi}_{t|P_{\mathcal{D}},P}(s,d) = \mathbb{E}\left[\sum_{\tau=t}^{L} c'(s_\tau, a_\tau) + c_c(d_\tau) + c_x(d_\tau, d_{\tau-1}) \,\Big|\, s_t = s . d_{t-1} = d\right]$$

- Goal:

$$\pi^* = \operatorname*{argmin}_{\pi} V^{\pi}_{1|P_{\mathcal{D}},P}(s_1, d_0)$$

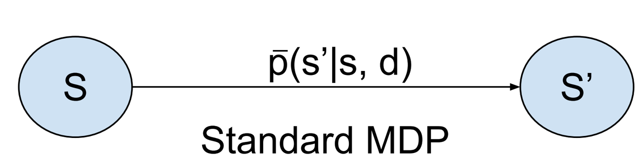
3. A simple solution

- Consider the set of agents $\mathcal{D}$ as action space and construct a new MDP $\mathcal{M} = (\mathcal{S} \times \mathcal{D}, \mathcal{D}, \bar{P}, \bar{C}, L)$ with the following transition/costs:

$$\bar{p}(s_{t+1}, d_t | (s_t, d_{t-1}), d_t) = \sum_a p(s_{t+1}|s_t, a) p_{d_t}(a|s_t)$$

$$\bar{c}((s_t, d_{t-1}), d_t) = \mathbb{E}_{a \sim p_{d_t}(\cdot|s_t)}[c'(s_t, a)] + c_c(d_t), c_x(d_t, d_{t-1})$$

- Now, we can use any standard algorithm like UCRL2, PSRL, Q-learning, policy gradient methods, etc to learn the optimal policy
- However, this approach does not learn the environment and needs to restart for each team of new agents



Standard MDP

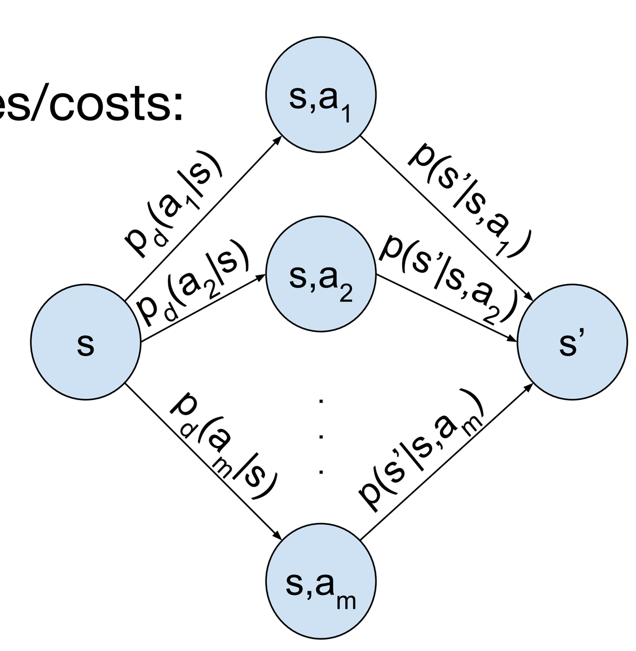
4. 2-layer MDP

- We can separate the environment transition probabilities from agents' policies
- In the left layer we have the agents' policies/costs:

  $p_d(a|s) \in P_{\mathcal{D}}$

  $c((s,d),d') = c_c(d') + c_x(d,d') \in C_{\mathcal{D}}$
- In the right layer we have the environment transition probabilities/costs:

  $p(s'|s,a) \in P$ and $c'(s,a) \in C_e$



2-layer MDP

5. Algorithm

- To find the optimal policy, we construct multiple L[1] confidence sets over agents' action policies and environment transition probabilities:

$$\mathcal{P}^k_{\cdot\,|\,d,s,t}(\delta) = \left\{ p_d : ||p_d(\cdot\,|\,s,t) - \hat{p}^k_d(\cdot\,|\,s)||_1 \leq \beta^k_{\mathcal{D}}(s,d,\delta) \right\},$$

$$\mathcal{P}^k_{\cdot\,|\,s,a,t}(\delta) = \left\{ p : ||p(\cdot\,|\,s,a,t) - \hat{p}^k(\cdot\,|\,s,a)||_1 \leq \beta^k(s,a,\delta) \right\}$$

- Then we apply *Optimism in the Face of Uncertainty* (OFU) to find the optimal value functions, i.e.,

$$v^k_t(s,d) = \min_{\pi} \min_{P_{\mathcal{D}} \in \mathcal{P}^k_{\mathcal{D}}(\delta)} \min_{P \in \mathcal{P}^k(\delta)} V^{\pi}_{t|P_{\mathcal{D}},P}(s,d)$$

Email balazadehvahid@gmail.com for further questions

5. Algorithm (cont.)

- **Theorem 1**. For any episode k, the optimal value function satisfies the following recursive equations:

$$v^k_t(s,d) = \min_{d_t \in \mathcal{D}} \left[c_{d_t}(s,d) + \min_{p_{d_t} \in \mathcal{P}^k_{\cdot\,|\,d_t,s,t}} \sum_{a \in \mathcal{A}} p_{d_t}(a\,|\,s,t) \times \left(c_e(s,a) + \min_{p \in \mathcal{P}^k_{\cdot\,|\,s,a,t}} \mathbb{E}_{s' \sim p(\cdot\,|\,s,a,t)}[v^k_{t+1}(s', d_t)] \right) \right]$$

- Here is our algorithm, UCRL2-MC, that uses Theorem. 1 to find a sequence of switching policies π^k for each episode k:

Algorithm 1 UCRL2-MC

```
Require: Cost functions C_D and C_e, δ
  N ← InitializeCounts()
  for k = 1, ..., K do
      {p̂_d^k}, p̂^k ← UpdateDistribution(N)
      P_D^k, P^k ← UpdateConfidenceSets({p̂_d^k}, p̂^k, δ)
      π^k ← GetOptimal(P_D^k, P^k, C_D, C_e),
      (s_1, d_0) ← InitializeConditions()
      for t = 1, ..., L do
          d_t ← π_t^k(s_t, d_{t-1})
          a_t ~ p_{d_t}(·|s_t)
          s_{t+1} ~ P(·|s_t, a_t)
          N ← UpdateCounts((s_t, d_t, a_t, s_{t+1}), N)
      end for
  end for
  Return π^K
```

6. Regret results

Setting	UCRL2-MC Regret	UCRL2 Regret
Single team of agents	$\tilde{\mathcal{O}}(L\lvert\mathcal{S}\rvert\sqrt{\mathcal{A}T})$	$\tilde{\mathcal{O}}(L\lvert\mathcal{S}\rvert\sqrt{\mathcal{D}T})$
Multiple teams of agents	$\tilde{\mathcal{O}}(L\lvert\mathcal{S}\rvert\sqrt{\mathcal{A}TN} + NL\sqrt{\lvert\mathcal{A}\rvert\lvert\mathcal{S}\rvert\lvert\mathcal{D}\rvert T})$	$\tilde{\mathcal{O}}(NL\lvert\mathcal{S}\rvert\sqrt{\mathcal{D}T})$

- Multiple teams of agents setting:
  - Multiple switching policies for N independent teams of agents
  - Maintain shared confidence bounds for the environment transition probabilities (e.g., a centralized setting)
  - Regret is defined as the sum of regrets for each team

7. Experiment results

- Obstacle avoidance task with teams of agents $\mathcal{D}_i = \{\mathbb{H}_i, \mathbb{M}\}$
- In the lane driving environment, each row has a traffic level in {*no-car*, *light*, *heavy*} and cell types are sampled based on that

Fig. 1: Performance of the machine policy, a human policy, and the optimal policy in terms of total cost. In panel (a), the episodes start with initial traffic level *no-car*, and in panel (b), the episodes start with an initial traffic level in {*light*, *heavy*}.

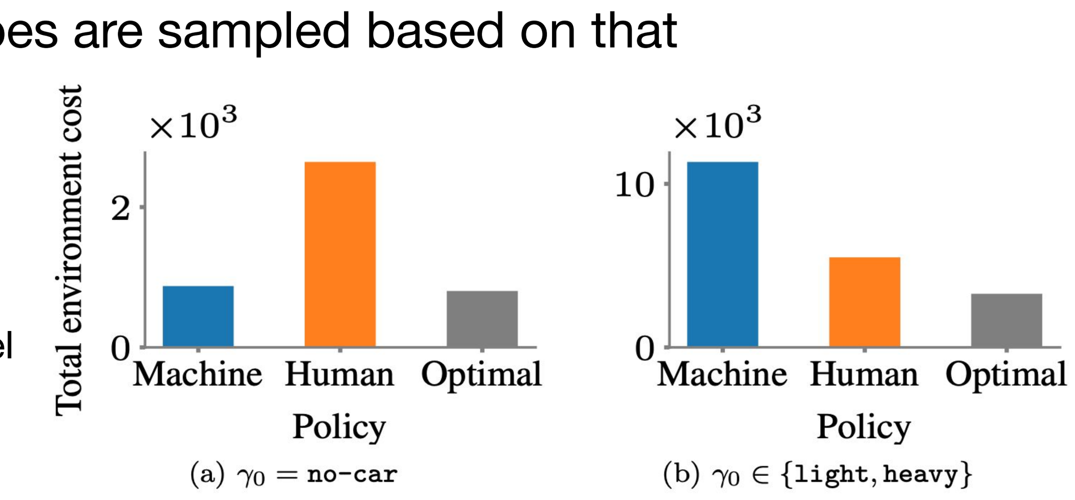


Fig. 2 (a): Total regret of the trajectories induced by the switching policies found by UCRL2-MC and those induced by a variant of UCRL2 in comparison with the trajectories induced by a machine driver and a human driver in a setting with a single team of agents, in K = 20,000 episodes.

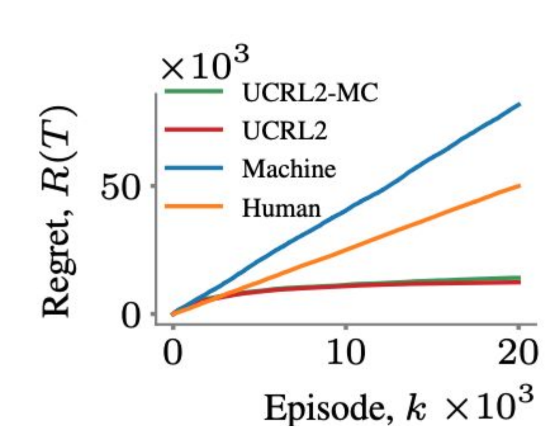
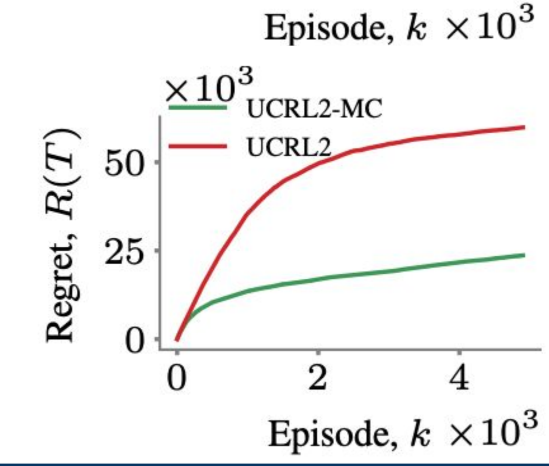


Fig. 2 (b): Total regret of the trajectories induced by the switching policies found by N instances of UCRL2-MC and those induced by N instances of a variant of UCRL2 in a setting with N team of agents, in K = 5,000 episodes. The sequence of policies found by UCRL2-MC outperform those found by the variant of UCRL2 in terms of total regret.



8. Future work

- We assumed agents' policies are fixed. Can we simultaneously optimize both the agents' action policies and the switching policy?
- Can we use function approximation or model-free algorithms for large MDPs?
- Human policy may change over time or before/after switching.
- Interventional experiments on a real-world semi-autonomous system.