



Bilevel Reinforcement Learning via the Development of Hyper-gradient without Lower-Level Convexity

Yan Yang

May 1, 2025

Academy of Mathematics and Systems Science
Chinese Academy of Sciences
University of Chinese Academy of Sciences

Joint work with **Dr. Bin Gao** and **Prof. Ya-xiang Yuan**

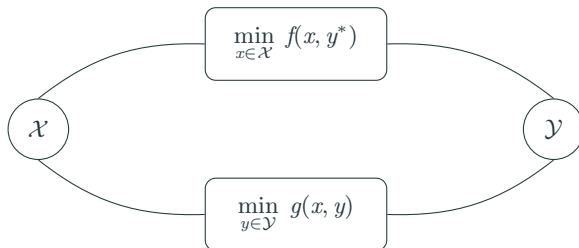
1. Bilevel Optimization and Hyper-gradient
2. Bilevel Reinforcement Learning (BiRL)
3. Model-based Soft BiRL
4. Model-free Soft BiRL
5. Theoretical Analysis
6. Numerical Experiments

Bilevel Optimization and Hyper-gradient

Bilevel Optimization (BiO)

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x, y^*) \\ \text{s.t.} \quad & y^* \in \arg \min_{y \in \mathcal{Y}} g(x, y) \end{aligned}$$

The nested structure **couples** the **upper level** and **lower level**



Model selection [Kunapuli et al., 2008; Giovannelli et al., 2021]

Hyper-parameters optimization [Franceschi et al., 2018; Bao et al., 2021]

Data poisoning [Liu et al., 2024]

Reinforcement learning [Hong et al., 2023; Chakraborty et al., 2024; Shen et al., 2024]

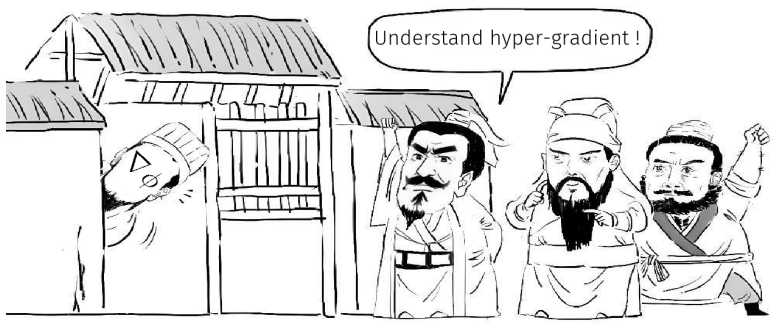
...

Three visits to hyper-gradient

Recall the BiO problem

$$\begin{aligned} \min_{x \in \mathbb{R}^{d_x}} \quad & \phi(x) := f(x, y^*(x)) \\ \text{s.t.} \quad & y^*(x) = \arg \min_{y \in \mathbb{R}^{d_y}} g(x, y) \end{aligned}$$

How to study the hyper-gradient $\nabla \phi$?



Approximate implicit differentiation (AID)

Scenario of interest: $g(x, \cdot)$ is **strongly convex**

$$y^*(x) = \arg \min_{y \in \mathbb{R}^{d_y}} g(x, y)$$

By optimality condition & implicit function theorem

$$0 \equiv \nabla_y g(x, y^*(x)) \Rightarrow \nabla_{xy}^2 g(x, y^*(x)) + \nabla_x y^*(x)^\top \nabla_{yy}^2 g(x, y^*(x)) = 0$$

Hyper-gradient computation

$$\begin{aligned} \nabla \phi(x) &:= \nabla f(x, y^*(x)) \\ &= \nabla_x f(x, y^*(x)) + \nabla_x y^*(x)^\top \nabla_y f(x, y^*(x)) \\ &= \nabla_x f(x, y^*) - \nabla_{xy}^2 g(x, y^*) \left[\nabla_{yy}^2 g(x, y^*) \right]^{-1} \nabla_y f(x, y^*) \end{aligned}$$

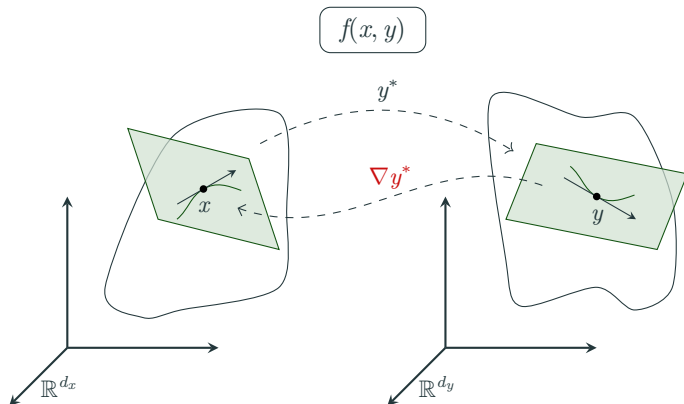
Vanilla update rule

$$\mathbf{1} \times : x^+ = x - \beta \left(\nabla_x f(x, y) - \nabla_{xy}^2 g(x, y) \left[\nabla_{yy}^2 g(x, y) \right]^{-1} \nabla_y f(x, y) \right)$$

$$\mathbf{N} \times : y^+ = y - \alpha \nabla_y g(x, y)$$

First visit to hyper-gradient

Geometry



Algebra

$$\nabla \phi(x) = \nabla_x f(x, y^*) - \nabla_{xy}^2 g(x, y^*) \left[\nabla_{yy}^2 g(x, y^*) \right]^{-1} \nabla_y f(x, y^*)$$

Bilevel Reinforcement Learning

Reinforcement learning (RL)

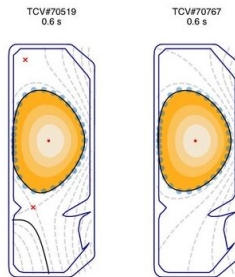
Agents learn to make sequential decisions, by **interacting** with environments



Silver et al., 2016
Nature

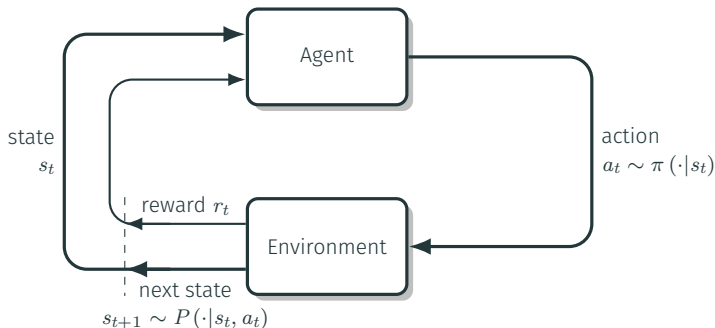


Samvelyan et al., 2019
DeepMind



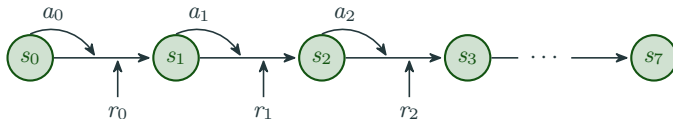
Degrave et al., 2022
Nature

Markov decision process (MDP)



- Environment $\mathcal{M}_\tau = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \tau)$
- State space $\mathcal{S}$, action space $\mathcal{A}$
- Transition matrix $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$
- Reward function $r \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$
- Policy $\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, which induces $P^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$: $P_{ss'}^\pi = \sum_a \pi_{sa} P_{sas'}$

Soft value function and Q-value function



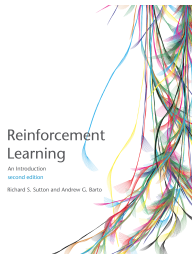
$$\forall s \in \mathcal{S} : \quad V_s^\pi := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (r_{s_t a_t} + \tau h(\pi_{s_t})) \mid s_0 = s, \pi, \mathcal{M}_\tau \right]$$

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A} : \quad Q_{sa}^\pi := r_{sa} + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} [V_{s'}^\pi]$$

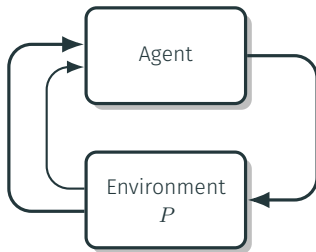
Given the initial state distribution ρ , RL aims to solve

$$\begin{aligned} \max_{\pi} \quad & V^\pi(\rho) := \mathbb{E}_{s \sim \rho} [V_s^\pi] \\ &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (r_{s_t a_t} + \tau h(\pi_{s_t})) \mid s_0 \sim \rho, \pi, \mathcal{M}_\tau \right] \end{aligned}$$

Two scenarios

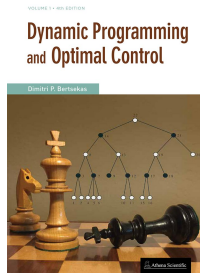


Model-free



unknown

known



Model-based

Parameterized MDP: $\mathcal{M}_\tau(x) = (\mathcal{S}, \mathcal{A}, P, r(x), \gamma, \tau)$

Quantities in bilevel context

$$\forall s \in \mathcal{S} : \quad V_s^\pi(x) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (r_{s_t a_t}(x) + \tau h(\pi_{s_t})) \mid s_0 = s, \pi, \mathcal{M}_\tau(x) \right]$$

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A} : \quad Q_{sa}^\pi(x) := r_{sa}(x) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} [V_{s'}^\pi(x)]$$

Bilevel RL problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s. t.} \quad & \pi^*(x) = \arg \min_{\pi} -V_{\mathcal{M}_\tau(x)}^\pi(\rho) \end{aligned}$$

Reward shaping [Hu et al., 2020; Devidze et al., 2022]

$$\min_x -V_{\bar{\mathcal{M}}_{\bar{\tau}}}^{\pi^*(x)}(\bar{\rho}), \text{ s. t. } \pi^*(x) = \arg \min_{\pi \in \Delta^{\mathcal{A}}} -V_{\mathcal{M}_{\tau(x)}}^{\pi}(\rho)$$

RL from human feedback (RLHF) [Christiano et al., 2017]

$$\begin{aligned} \min_x \quad & \mathbb{E}_{y, d_1, d_2 \sim \rho(d; \pi^*(x))} [l_h(d_1, d_2, y; x)] \\ \text{s. t.} \quad & \pi^*(x) = \arg \min_{\pi} -V_{\mathcal{M}_{\tau(x)}}^{\pi}(\rho) \end{aligned}$$

- Trajectory sampled by $\pi^*(x)$: $d_i = \{(s_h^i, a_h^i)\}_{h=0}^{H-1}$ ($i = 1, 2$)
- Preference for d_1 over d_2 : $y \in \{0, 1\}$

More applications

- Apprenticeship learning [Arora and Doshi, 2021]
- Contract design [Wu et al., 2024]
- Robot navigation [Chakraborty et al., 2024]

Problem formulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s. t.} \quad & \pi^*(x) = \arg \min_{\pi} -V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho) \end{aligned}$$

where $V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (r_{s_t a_t}(x) + \tau h(\pi_{s_t})) \mid s_0 \sim \rho, \pi, \mathcal{M}_{\tau}(x) \right]$

Main difficulties

- Lower-level inherent **non-convexity** [Agarwal et al., 2020; Lan, 2023 MP]
- Only the **non-uniform PL** condition [Mei et al., 2020]
- **Existence** of hyper-gradient $\nabla \phi(x)$? [Shen et al., 2024]

Related work

- An AID-based bilevel RL framework, PARL [Chakraborty et al., ICLR 2024]
- A penalty-based method, PBRL [Shen et al., ICML 2024]
- A stochastic bilevel RL method, HPGD [Thoma et al., NeurIPS 2024]

Comparison among provable bilevel RL algorithms

Algorithm	Deter. or Stoch.	Conv. Rate	Inner Iter.	Oracle
PARL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log \epsilon^{-1})$	1st+2nd
PBRL	Deter.	$\mathcal{O}(\lambda \epsilon^{-1})$	$\mathcal{O}(\log \lambda^2 \epsilon^{-1})$	1st
HPGD	Stoch.	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\log \epsilon^{-1})$	1st
M-SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(1)$	1st
SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log \epsilon^{-1})$	1st
Stoc-SoBiRL	Stoch.	$\tilde{\mathcal{O}}(\epsilon^{-1.5})$	$\mathcal{O}(\log \epsilon^{-1})$	1st

Model-based Soft BiRL

Revisit the strong convexity requirement in AID

- Unique lower-level solution $\implies y^*(x)$
- Nonsingular Hessian $\implies \nabla y^*(x)$

Softmax temporal value consistency [Nachum et al., 2017]

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}: \quad \pi_{sa}^*(x) = \exp \left(\tau^{-1} \left(r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V_{s'}^*(x) - V_s^*(x) \right) \right)$$
$$\forall s \in \mathcal{S}: \quad V_s^*(x) = \tau \log \left(\sum_a \exp \left(\frac{r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V_{s'}^*(x)}{\tau} \right) \right)$$

Take the derivative

$$\nabla \pi^*(x) = \tau^{-1} \text{diag}(\pi^*(x)) (\nabla r(x) - U \nabla V^*(x))$$

The fixed-point equation

Revisit the strong convexity requirement in AID

- Unique lower-level solution $\implies y^*(x)$
- Nonsingular Hessian $\implies \nabla y^*(x)$

Softmax temporal value consistency [Nachum et al., 2017]

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}: \quad \pi_{sa}^*(x) = \exp \left(\tau^{-1} \left(r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V_{s'}^*(x) - V_s^*(x) \right) \right)$$
$$\forall s \in \mathcal{S}: \quad V_s^*(x) = \tau \log \left(\sum_a \exp \left(\frac{r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V_{s'}^*(x)}{\tau} \right) \right)$$

Take the derivative

$$\nabla \pi^*(x) = \tau^{-1} \text{diag}(\pi^*(x)) (\nabla r(x) - U \nabla V^*(x))$$

How to tackle $\nabla V^*(x)$?

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$

$$\varphi(x, v) = \tau \log \left(\exp \left(\frac{r(x) + \gamma P v}{\tau} \right) \cdot \mathbf{1} \right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}$$

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$

$$\varphi(x, v) = \tau \log \left(\exp \left(\frac{r(x) + \gamma P v}{\tau} \right) \cdot \mathbf{1} \right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}$$

$$\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1 \implies \text{unique solution } V^*(x)$$

Investigating $\nabla V^*(x)$

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|S|} \mapsto \mathbb{R}^{|S|}$

$$\varphi(x, v) = \tau \log \left(\exp \left(\frac{r(x) + \gamma P v}{\tau} \right) \cdot \mathbf{1} \right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}$$

$$\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1 \implies \text{unique solution } V^*(x)$$

- Differentiate both sides of $\varphi(x, V^*(x)) = V^*(x)$

$$\nabla V^*(x) = (I - \nabla_v \varphi(x, V^*(x)))^{-1} \nabla_x \varphi(x, V^*(x))$$

Investigating $\nabla V^*(x)$

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$

$$\varphi(x, v) = \tau \log \left(\exp \left(\frac{r(x) + \gamma P v}{\tau} \right) \cdot \mathbf{1} \right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp(\tau^{-1} (r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}{\sum_a \exp(\tau^{-1} (r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}$$

$$\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1 \implies \text{unique solution } V^*(x)$$

- Differentiate both sides of $\varphi(x, V^*(x)) = V^*(x)$

$$\nabla V^*(x) = (I - \nabla_v \varphi(x, V^*(x)))^{-1} \nabla_x \varphi(x, V^*(x))$$

$$\text{An observation in MDP } \nabla_v \varphi(x, V^*(x)) = \gamma P^{\pi^*(x)}$$

Model-based hyper-gradient

$$\begin{aligned}\nabla \phi(x) &= \nabla_x f(x, \pi^*(x)) + \nabla \pi^*(x)^\top \nabla_{\pi} f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} (\nabla r(x) - U \nabla V^*(x))^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \\ &\quad - \tau^{-1} \nabla_x \varphi(x, V^*(x))^\top \left(I - \gamma P^{\pi^*(x)} \right)^{-\top} U^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x))\end{aligned}$$

Model-based hyper-gradient

$$\begin{aligned}\nabla \phi(x) &= \nabla_x f(x, \pi^*(x)) + \nabla \pi^*(x)^\top \nabla_{\pi} f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} (\nabla r(x) - U \nabla V^*(x))^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \\ &\quad - \tau^{-1} \nabla_x \varphi(x, V^*(x))^\top \left(I - \gamma P^{\pi^*(x)} \right)^{-\top} U^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x))\end{aligned}$$

Computing $\nabla \phi(x)$ is expensive!

Model-based hyper-gradient

$$\begin{aligned}\nabla \phi(x) &= \nabla_x f(x, \pi^*(x)) + \nabla \pi^*(x)^\top \nabla_{\pi} f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} (\nabla r(x) - U \nabla V^*(x))^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \\ &\quad - \tau^{-1} \nabla_x \varphi(x, V^*(x))^\top \left(I - \gamma P^{\pi^*(x)} \right)^{-\top} U^\top \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x))\end{aligned}$$

Computing $\nabla \phi(x)$ is expensive!

Inexact strategy

- $V_k^* := V^*(x_k) \leftarrow V_k$ $Q_k^* := Q^*(x_k) \leftarrow Q_k$ $\pi_k^* := \pi^*(x_k) \leftarrow \pi_k$
- $A_k^{-1} b_k \leftarrow w_k$ with $A_k := (I - \gamma P^{\pi_k})^\top$ $b_k := U^\top \text{diag}(\pi_k) \nabla_{\pi} f(x_k, \pi_k)$

Model-based hyper-gradient estimator

$$\begin{aligned}\widehat{\nabla} \phi(x_k, \pi_k, V_k, w_k) &:= \nabla_x f(x_k, \pi_k) + \frac{1}{\tau} \nabla r(x_k)^\top \text{diag}(\pi_k) \nabla_{\pi} f(x_k, \pi_k) \\ &\quad - \frac{1}{\tau} \nabla_x \varphi(x_k, V_k)^\top w_k\end{aligned}$$

Soft Bellman operator solves lower-level problem

$$\mathcal{T}_{\mathcal{M}_\tau(x)}(Q)(s, a) = r_{sa}(x) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} [\tau \log (\|\exp (Q(s', \cdot) / \tau)\|_1)]$$

Softmax temporal value consistency points to V and π

$$\begin{aligned}V_s^*(x) &= \tau \log \left(\sum_a \exp (Q_{sa}^*(x) / \tau) \right) \\ \pi_{sa}^*(x) &= \frac{\exp \{Q_{sa}^*(x) / \tau\}}{\sum_{a'} \exp (Q_{sa'}^*(x) / \tau)}\end{aligned}$$

Least squared perspective

$$w_k \approx \min_{w \in \mathbb{R}^{|S|}} \frac{1}{2} \|A_k w - b_k\|^2$$

M-SoBiRL

$$\mathbf{1} \times : w_k = w_{k-1} - \xi \left(\left(A_k^\top A_k \right) w_{k-1} - A_k^\top b_k \right)$$

$$\mathbf{1} \times : V_k(s) = \tau \log \left(\sum_a \exp(Q_k(s, a) / \tau) \right)$$

$$\mathbf{1} \times : x_{k+1} = x_k - \beta \hat{\nabla} \phi(x_k, \pi_k, V_k, w_k)$$

$$\mathbf{N} \times : Q_{k,n} = \mathcal{T}_{\mathcal{M}_\tau(x_{k+1})}(Q_{k,n-1})$$

$$\mathbf{1} \times : \pi_{k+1} = \text{softmax}(Q_{k+1} / \tau)$$

Model-free Soft BiRL

Problem formulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s. t.} \quad & \pi^*(x) = \arg \min_{\pi} -V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho) \end{aligned}$$

with $f(x, \pi) = \mathbb{E}_{d_i \sim \rho(d; \pi)} [l(d_1, d_2, \dots, d_I; x)]$

Main difficulties

$$\begin{aligned} \nabla \phi(x) = & \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^{\top} \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \\ & - \tau^{-1} \nabla_x \varphi(x, V^*(x))^{\top} \left(I - \gamma P^{\pi^*(x)} \right)^{-\top} U^{\top} \text{diag}(\pi^*(x)) \nabla_{\pi} f(x, \pi^*(x)) \end{aligned}$$

- ☹ Relying explicitly on the **black-box P**
- ☹ Involving large-scale **matrix multiplications**

Absorb P into an expectation

$$\begin{aligned}\nabla V_s^*(x) &= (I - \nabla_v \varphi(x, V^*(x)))_s^{-1} \nabla_x \varphi(x, V^*(x)) \\&= \sum_{t=0}^{\infty} \gamma^t \sum_{s'} P(s_t = s' | s_0 = s, \pi^*(x)) \sum_a \nabla r_{s'a}(x) \pi_{s'a}^*(x) \\&= \sum_{t=0}^{\infty} \gamma^t \sum_{s', a} P(s_t = s', a_t = a | s_0 = s, \pi^*(x)) \nabla r_{s'a}(x) \\&= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_x r_{s_t a_t}(x) \mid s_0 = s, \pi^*(x), \mathcal{M}_\tau(x) \right]\end{aligned}$$

Absorb P into an expectation

$$\begin{aligned}\nabla V_s^*(x) &= (I - \nabla_v \varphi(x, V^*(x)))_s^{-1} \nabla_x \varphi(x, V^*(x)) \\&= \sum_{t=0}^{\infty} \gamma^t \sum_{s'} P(s_t = s' | s_0 = s, \pi^*(x)) \sum_a \nabla r_{s'a}(x) \pi_{s'a}^*(x) \\&= \sum_{t=0}^{\infty} \gamma^t \sum_{s', a} P(s_t = s', a_t = a | s_0 = s, \pi^*(x)) \nabla r_{s'a}(x) \\&= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_x r_{s_t a_t}(x) \mid s_0 = s, \pi^*(x), \mathcal{M}_\tau(x) \right]\end{aligned}$$

☺ Estimate via fully first-order information!

Consistency condition

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}: \quad \pi_{sa}^*(x) = \exp \left(\tau^{-1} \left(Q_{sa}^*(x) - V_s^*(x) \right) \right)$$

Hyper-objective

$$\begin{aligned} \phi(x) &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [l(d_1, d_2, \dots, d_I; x)] \\ &= \sum_{(d_1, \dots, d_I)} l(d_1, d_2, \dots, d_I; x) \prod_{i=1}^I P(d_i; \pi^*(x)) \end{aligned}$$

Log probability trick

$$\begin{aligned} &\sum_{(d_1, \dots, d_I)} l(d_1, d_2, \dots, d_I; x) \nabla \prod_{i=1}^I P(d_i; \pi^*(x)) \\ &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \nabla \left(\sum_i \log P(d_i; \pi^*(x)) \right) \right] \\ &= \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right] \end{aligned}$$

Model-free hyper-gradient

$$\begin{aligned}\nabla \phi(x) = & \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ & + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right]\end{aligned}$$

Second visit to hyper-gradient

Model-free hyper-gradient

$$\begin{aligned}\nabla \phi(x) = & \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ & + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right]\end{aligned}$$

☺ Estimate $\nabla \phi(x)$ via sampling first-order information!

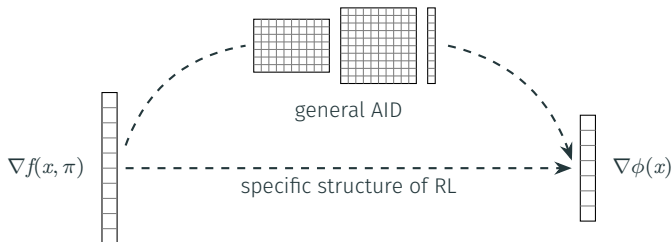
Second visit to hyper-gradient

Model-free hyper-gradient

$$\begin{aligned}\nabla\phi(x) = & \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ & + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right]\end{aligned}$$

☺ Estimate $\nabla\phi(x)$ via sampling first-order information!

Review the hyper-gradient



Step 1: evaluating the implicit differentiations

$$\begin{aligned}\tilde{\nabla} V_s(x_k, \pi_k) &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, \pi_k, \mathcal{M}_{\tau}(x_k) \right] \\ \tilde{\nabla} Q_{sa}(x_k, \pi_k) &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, a_0 = a, \pi_k, \mathcal{M}_{\tau}(x_k) \right]\end{aligned}$$

Step 2: absorbing them into the sampling process

$$\begin{aligned}\tilde{\nabla} \phi(x_k, \pi_k) &= \mathbb{E}_{d_i \sim \rho(d; \pi_k)} [\nabla l(d_1, d_2, \dots, d_I; x_k)] \\ &\quad + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi_k)} \left[l(d_1, d_2, \dots, d_I; x_k) \left(\sum_i \sum_h \tilde{\nabla} \left(Q_{s_h^i a_h^i} - V_{s_h^i} \right) (x_k, \pi_k) \right) \right]\end{aligned}$$

Algorithm SoBiRL

Input: iteration number K , step size β , initialization x_1, π_0 , accuracy ϵ

- 1: **for** $k = 1, \dots, K$ **do**
- 2: Solve the lower-level problem approximately with initial guess π_{k-1} to get π_k such that $\|\pi_k - \pi^*(x_k)\|_2^2 \leq \epsilon$
- 3: Compute the approximate hyper-gradient $\tilde{\nabla}\phi(x_k, \pi_k)$
- 4: Implement an inexact hyper-gradient descent step

$$x_{k+1} = x_k - \beta \tilde{\nabla}\phi(x_k, \pi_k)$$

- 5: **end for**

Output: (x_{K+1}, π_{K+1})

Algorithm SoBiRL

Input: iteration number K , step size β , initialization x_1, π_0 , accuracy ϵ

- 1: **for** $k = 1, \dots, K$ **do**
- 2: Solve the lower-level problem approximately with initial guess π_{k-1} to get π_k such that $\|\pi_k - \pi^*(x_k)\|_2^2 \leq \epsilon$
- 3: Compute the approximate hyper-gradient $\tilde{\nabla}\phi(x_k, \pi_k)$
- 4: Implement an inexact hyper-gradient descent step

$$x_{k+1} = x_k - \beta \tilde{\nabla}\phi(x_k, \pi_k)$$

- 5: **end for**

Output: (x_{K+1}, π_{K+1})

☹ Sampling process introduces stochasticity

Main adaptation

- Characterize the bias and variance
- Replace $\tilde{\nabla}\phi$ with its stochastic counterpart

$$\begin{aligned}\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; \mathbf{x}_k, \pi_k) &= \frac{1}{M} \sum_{m=1}^M \nabla l(\mathbf{d}_k^m; x_k) \\ &\quad + \frac{1}{\tau M} \sum_{m=1}^M l(\mathbf{d}_k^m; x_k) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^{m,i} a_h^{m,i}}^{\xi_k} - V_{s_h^{m,i}}^{\zeta_k} \right) (x_k, \pi_k)\end{aligned}$$

- Maintain a momentum-instructed h_k for acceleration

$$h_k = \bar{\nabla}\phi_k + (1 - \mu_k) (h_{k-1} - \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; \mathbf{x}_{k-1}, \pi_{k-1}))$$

Main adaptation

- Characterize the bias and variance
- Replace $\tilde{\nabla}\phi$ with its stochastic counterpart

$$\begin{aligned}\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; \mathbf{x}_k, \pi_k) &= \frac{1}{M} \sum_{m=1}^M \nabla l(\mathbf{d}_k^m; x_k) \\ &\quad + \frac{1}{\tau M} \sum_{m=1}^M l(\mathbf{d}_k^m; x_k) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^m, i}^{\xi_k, a_h^m} - V_{s_h^m, i}^{\zeta_k} \right) (x_k, \pi_k)\end{aligned}$$

- Maintain a momentum-instructed h_k for acceleration

$$h_k = \bar{\nabla}\phi_k + (1 - \mu_k) (h_{k-1} - \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; \mathbf{x}_{k-1}, \pi_{k-1}))$$

☺ Misalignment: distribution of (D_k, ξ_k, ζ_k) relies on π_k

Theoretical Analysis

Upper-level assumptions

- (Model-based) $\nabla f(x, \pi)$ is L_f -Lipschitz, and $\|\nabla_{\pi} f(x, \pi^*(x))\|_2 \leq C_{f\pi}$
- (Model-free) $l(d_1, d_2, \dots, d_I; x)$ is L_l -Lipschitz, $\nabla l(d_1, d_2, \dots, d_I; x)$ is L_{l1} -Lipschitz, and $|l(d_1, d_2, \dots, d_I; x)| \leq C_l$.

Lower-level assumptions

- For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $|r_{sa}(x)| \leq C_r$, $\|\nabla r_{sa}(x)\|_2 \leq C_{rx}$, and $\nabla r_{sa}(x)$ is L_r -Lipschitz

Upper-level assumptions

- (Model-based) $\nabla f(x, \pi)$ is L_f -Lipschitz, and $\|\nabla_{\pi} f(x, \pi^*(x))\|_2 \leq C_{f\pi}$
- (Model-free) $l(d_1, d_2, \dots, d_I; x)$ is L_l -Lipschitz, $\nabla l(d_1, d_2, \dots, d_I; x)$ is L_{l1} -Lipschitz, and $|l(d_1, d_2, \dots, d_I; x)| \leq C_l$.

Lower-level assumptions

- For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $|r_{sa}(x)| \leq C_r$, $\|\nabla r_{sa}(x)\|_2 \leq C_{rx}$, and $\nabla r_{sa}(x)$ is L_r -Lipschitz

😊 Only require first-order regularity!

Theorem (model-based)

In M-SoBiRL, with constant step sizes β, ξ , and the inner iteration number $N \sim \mathcal{O}(1)$, the iterates $\{x_k\}$ satisfy

$$\frac{1}{K} \sum_{k=1}^K \|\nabla \phi(x_k)\|_2^2 = \mathcal{O}\left(\frac{1}{K}\right)$$

Proposition (hyper-gradient)

For any $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $x_1, x_2 \in \mathbb{R}^n$,

$$\|\nabla Q_{sa}^*(x_1) - \nabla Q_{sa}^*(x_2)\|_2 \leq L_{V1} \|x_1 - x_2\|_2$$

$$\|\nabla V_s^*(x_1) - \nabla V_s^*(x_2)\|_2 \leq L_{V1} \|x_1 - x_2\|_2$$

$$\|\nabla \phi(x_1) - \nabla \phi(x_2)\|_2 \leq L_\phi \|x_1 - x_2\|_2$$

with

$$L_{V1} = \frac{(1 + \gamma) C_{rx} L_\pi \sqrt{|\mathcal{A}|}}{(1 - \gamma)^2} + \frac{L_r}{1 - \gamma}$$

$$L_\phi = L_{l1} + L_l H I L_\pi \sqrt{|\mathcal{A}|} + 2\tau^{-1} H I \left(C_l L_{V1} + \frac{C_{rx}}{1 - \gamma} L_l \right) + \frac{2H^2 I^2 C_l C_{rx} L_\pi}{\tau (1 - \gamma)} \sqrt{|\mathcal{A}|}$$

Convergence analysis (model-free)

Theorem (model-free)

In SoBiRL, with a constant step size $\beta < \frac{1}{2L_\phi}$, the iterates satisfy

$$\frac{1}{K} \sum_{k=1}^K \|\nabla \phi(x_k)\|^2 \leq \frac{\phi(x_1) - \phi^*}{K(\frac{\beta}{2} - \beta^2 L_\phi)} + \frac{1 + 2\beta L_\phi}{1 - 2\beta L_\phi} L_\phi^2 \epsilon$$

Theorem (stochastic)

In Stoc-SoBiRL, with appropriate sampling configurations, the iterates satisfy

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} [\|\nabla \phi(x_k)\|_2^2] = \mathcal{O}\left(\frac{\log K}{K^{2/3}}\right) = \tilde{\mathcal{O}}\left(K^{-\frac{2}{3}}\right)$$

Algorithm	Deter. or Stoch.	Conv. Rate	Inner Iter.	Oracle
PARL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log \epsilon^{-1})$	1st+2nd
PBRL	Deter.	$\mathcal{O}(\lambda \epsilon^{-1})$	$\mathcal{O}(\log \lambda^2 \epsilon^{-1})$	1st
HPGD	Stoch.	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\log \epsilon^{-1})$	1st
M-SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(1)$	1st
SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log \epsilon^{-1})$	1st
Stoc-SoBiRL	Stoch.	$\tilde{\mathcal{O}}(\epsilon^{-1.5})$	$\mathcal{O}(\log \epsilon^{-1})$	1st

Numerical Experiments

Goal: learn the intrinsic reward model and the optimal policy
only based on preference labels $y \in \{0, 1\}$

Problem formulation:

$$\begin{aligned} \min_x \quad & \mathbb{E}_{y, d_1, d_2 \sim \rho(d; \pi^*(x))} [l_h(d_1, d_2, y; x)] \\ \text{s. t.} \quad & \pi^*(x) = \arg \min_{\pi} - V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho) \end{aligned}$$

Environments: Atari games and Mujoco environments

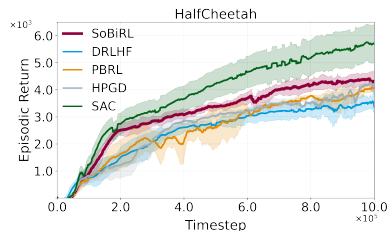
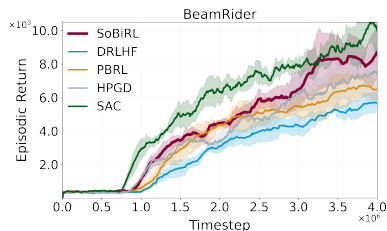
Compared methods

- DRLHF [Christiano et al., 2017]: alternating minimization of $\mathbb{E}[l_h]$ and $-V_{\mathcal{M}_\tau}^\pi(x)$
- PBRL [Shen et al., 2024]: penalty-based method
- HPGD [Thoma et al., 2024]: stochastic bilevel RL method
- SAC (baseline) [Haarnoja et al., 2018]: soft actor-critic algorithm given the ground-truth reward

Running platform

- Intel® Xeon® Gold 6330 CPUs & NVIDIA A800 GPU
- Python 3.8.0 + Pytorch 1.13.1 + Stable-Baselines 3

Numerical results



- SoBiRL is comparable to the baseline SAC
- SoBiRL obtains a higher return than other bilevel methods
- The accuracy of preference prediction of DRLHF achieves approximately 85%, while it hovers around 54% for SoBiRL

Third visit to hyper-gradient

Update rule of DRLHF [Christiano et al., 2017] alternates between minimizing $\mathbb{E}[l_h]$ and learning π

$$\text{M}\times : x^+ = x - \beta \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)]$$

$$\text{N}\times : \pi^+ = \pi + \nabla_{\pi} V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho)$$

Update rule of SoBiRL (ours)

$$\text{1}\times : x^+ = x - \beta \nabla \phi(x)$$

$$\text{N}\times : \pi^+ = \pi + \nabla_{\pi} V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho)$$

$$\begin{aligned} \nabla \phi(x) = & \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ & + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right] \end{aligned}$$

☺ $\nabla \phi(x)$ integrates exploitation and exploration!

Contributions

- Fully first-order hyper-gradient
- AID-based RL algorithm without lower-level convexity
- Stochastic extension
- Enhanced convergence results

Future works

- Extension to general strongly-convex regularization
- General settings without regularization

References

- **Yan Yang**, Bin Gao, Ya-xiang Yuan. *LancBiO: dynamic Lanczos-aided bilevel optimization via Krylov subspace*. ICLR (2025)
- **Yan Yang**, Bin Gao, Ya-xiang Yuan. *Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity*. AISTATS (2025)

Thanks for your attention!

yangyan@amss.ac.cn