[†]**IBM Research**, [¶]MIT, [§]UCSD

Robust Multi-Objective Bilevel Optimization

Applications in Machine Learning

Pari Ram[†], Alex Gu[¶], Songtao Lu[†], Lily Weng[†§]

October 18, 2022

Bilevel Stochastic Optimization

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} F(x) \triangleq \mathbb{E}_\xi \left[f(x, y^\star(x); \xi) \right]$$

$$\text{subject to} \quad y^\star(x) \in \arg\min_{y \in \mathcal{Y} = \mathbb{R}^d} G(x, y) \triangleq \mathbb{E}_\zeta \left[g(x, y; \zeta) \right]$$

Bilevel Stochastic Optimization

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} F(x) \triangleq \mathbb{E}_{\xi}\left[f(x, y^\star(x); \xi)\right]$$

$$\text{subject to} \quad y^\star(x) \in \operatorname*{arg\,min}_{y \in \mathcal{Y} = \mathbb{R}^d} G(x, y) \triangleq \mathbb{E}_{\zeta}\left[g(x, y; \zeta)\right]$$

Stochastic Objectives

- **Strongly convex** lower-level ($\mathtt{LL}$) objective $G(x, \cdot)$
- **Weakly convex** upper-level ($\mathtt{UL}$) objective $F(x)$

Bilevel Stochastic Optimization

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} F(x) \triangleq \mathbb{E}_\xi \left[f(x, y^\star(x); \xi) \right]$$

$$\text{subject to} \quad y^\star(x) \in \arg\min_{y \in \mathcal{Y} = \mathbb{R}^d} G(x, y) \triangleq \mathbb{E}_\zeta \left[g(x, y; \zeta) \right]$$

Constraints

- **Unconstrained** LL problem
- **Constrained** UL problem

Robust Multi-Objective Bilevel Problem

We have n pairs of stochastic objectives $F_i, G_i, i \in [n] \triangleq \{1, \dots, n\}$

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} \max_{i \in [n]} F_i(x) \triangleq \mathbb{E}_{\xi_i}\left[f_i(x, y_i^\star(x); \xi_i)\right]$$

$$\text{subject to} \quad y_i^\star(x) \in \underset{y_i \in \mathcal{Y}_i = \mathbb{R}^{d_i}}{\arg\min}\ G_i(x, y_i) \triangleq \mathbb{E}_{\zeta_i}\left[g_i(x, y_i; \zeta_i)\right]$$

$$\forall i \in [n]$$

Robust Multi-Objective Bilevel Problem

We have n pairs of stochastic objectives $F_i, G_i, i \in [n] \triangleq \{1, \ldots, n\}$

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} \max_{i \in [n]} F_i(x) \triangleq \mathbb{E}_{\xi_i}\left[f_i(x, y_i^\star(x); \xi_i)\right]$$

$$\text{subject to} \quad y_i^\star(x) \in \arg\min_{y_i \in \mathcal{Y}_i = \mathbb{R}^{d_i}} G_i(x, y_i) \triangleq \mathbb{E}_{\zeta_i}\left[g_i(x, y_i; \zeta_i)\right]$$

$$\forall i \in [n]$$

Multiple objectives

- Each obj pair has their UL and LL stoc oracles $\xi_i, \zeta_i, i \in [n]$
- UL variable x **shared across all objectives**
- Each pair has its **own specific** LL **variable** $y_i, i \in [n]$

Pari Ram[†], Alex Gu[¶], Songtao Lu[†], Lily Weng[†§] – Robust Multi-Objective Bilevel Optimization

Robust Multi-Objective Bilevel Problem

We have n pairs of stochastic objectives $F_i, G_i, i \in [n] \triangleq \{1, \ldots, n\}$

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} \max_{i \in [n]} F_i(x) \triangleq \mathbb{E}_{\xi_i}\left[f_i(x, y_i^\star(x); \xi_i)\right]$$

$$\text{subject to} \quad y_i^\star(x) \in \operatorname*{arg\,min}_{y_i \in \mathcal{Y}_i = \mathbb{R}^{d_i}} G_i(x, y_i) \triangleq \mathbb{E}_{\zeta_i}\left[g_i(x, y_i; \zeta_i)\right]$$

$$\forall i \in [n]$$

Additional Features

- **Flexible**: LL vars $\{y_i\}_{i \in [n]}$ can have diff domains ($\mathcal{Y}_i \neq \mathcal{Y}_j$)
- **Robust**: Shared UL var x optimizes worst-case obj F_i

Application: Multi-task Representation Learning

Learn a robust representation useful for many tasks.

	Problem mapping
$i \in [n]$	Tasks
UL var x	Shared representation network Φ_x params
LL var y_i	Per-task model w_{y_i} params
UL obj $\mathbb{E}_{\xi_i} f_i(x, y_i; \xi_i)$	$\mathcal{L}(w_{y_i} \circ \Phi_x; D_i^{\texttt{val}})$
LL obj $\mathbb{E}_{\zeta_i} g_i(x, y_i; \zeta_i)$	$\mathcal{L}(w_{y_i} \circ \Phi_x; D_i^{\texttt{train}}) + \rho \cdot \Omega(y_i)$

Application: **Fair** Representation Learning

Learn a robust representation ensuring fairness across different groups.

	Problem mapping
$i \in [n]$	Demographic groups
UL var x	Shared representation network Φ_x params
LL var y_i	Per-task model w_{y_i} params
UL obj $\mathbb{E}_{\xi_i} f_i(x, y_i; \xi_i)$	$\mathcal{L}(w_{y_i} \circ \Phi_x; D_i^{\mathtt{val}})$
LL obj $\mathbb{E}_{\zeta_i} g_i(x, y_i; \zeta_i)$	$\mathcal{L}(w_{y_i} \circ \Phi_x; D_i^{\mathtt{train}}) + \rho \cdot \Omega(y_i)$

Pari Ram[†], Alex Gu[¶], Songtao Lu[†], Lily Weng[†§] – Robust Multi-Objective Bilevel Optimization

Application: Meta-learning for Hyperparameter Opt

Learn a robust set of hyperparameters (HP) that works well for multiple problems.

	Problem mapping
$i \in [n]$	Different HPO tasks
UL var x	Shared HP x
LL var y_i	Per-task model w_{y_i} params for HP x
UL obj $\mathbb{E}_{\xi_i} f_i(x, y_i; \xi_i)$	$\mathcal{L}(w_{y_i}; D_i^{\mathtt{val}})$
LL obj $\mathbb{E}_{\zeta_i} g_i(x, y_i; \zeta_i)$	$\mathcal{L}(w_{y_i}; D_i^{\mathtt{train}}), w_{y_i} \in \mathcal{F}(x)$

Application: "SuperNet" training for DARTS

Learn a shared SuperNet such that different application specific sub-networks obtained via Differentiable Architecture Search (DARTS) have robust performance.

	Problem mapping
$i \in [n]$	Different applications
UL var x	SuperNet params W_x
LL var y_i	Per-application subnetwork w_{y_i} params
UL obj $\mathbb{E}_{\xi_i} f_i(x, y_i; \xi_i)$	$\mathcal{L}(w_{y_i}; D_i^{\mathtt{val}})$
LL obj $\mathbb{E}_{\zeta_i} g_i(x, y_i; \zeta_i)$	$\mathcal{L}(w_{y_i}; D_i^{\mathtt{train}}) + \rho \cdot \Omega(W_x, w_{y_i})$

Cai, H., et al. *Once for All: Train One Network and Specialize it for Efficient Deployment.* ICLR 2020.

Application: Multi-agent Reinforcement Learning

Learn a shared model of the environment allowing robust performance across all agents.

	Problem mapping
$i \in [n]$	Different agents in the environment
UL var x	Environment model params E_x
LL var y_i	Per-agent params A_{y_i}
UL obj $\mathbb{E}_{\xi_i} f_i(x, y_i; \xi_i)$	$-\mathcal{R}(A_{y_i}, E_x; R^{\texttt{val}})$
LL obj $\mathbb{E}_{\zeta_i} g_i(x, y_i; \zeta_i)$	$-\mathcal{R}(A_{y_i}, E_x; R^{\texttt{train}})$

Flexibility: Diff agents can have diff action spaces (land, air, water).

Solution

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} \max_{i \in [n]} F_i(x) \quad \text{subject to} \quad y_i^\star(x) \in \operatorname*{arg\,min}_{y_i \in \mathcal{Y}_i = \mathbb{R}^{d_i}} G_i(x, y_i) \, \forall i \in [n]$$

Reformulation

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^D} \max_{\lambda \in \Delta_n} \sum_{i=1}^{n} \lambda_i F_i(x) \text{ s.t. } y_i^\star(x) \in \operatorname*{arg\,min}_{y_i \in \mathcal{Y}_i = \mathbb{R}^{d_i}} G_i(x, y_i) \, \forall i \in [n]$$

$$\Delta_n = \left\{ \lambda \in \mathbb{R}^n : \lambda_i \geqslant 0 \, \forall i \in [n], \sum_{i=1}^{n} \lambda_i = 1 \right\} \quad (n\text{-simplex})$$

Single-loop Projected Gradient Descent-Ascent

`MORBiT`: Multi-Objective Robust Bilevel Two-timescale alg

Algorithm 1: Learning rates α, β, γ for x, y, λ resp

for $k = 1, 2, \cdots, K$ **do**

$\quad y^{(k+1)} \leftarrow y^{(k)} - \beta h^{(k)}$ `(SGD)`

$\quad x^{(k+1)} \leftarrow \text{proj}_{\mathcal{X}}(x^{(k)} - \alpha h_x^{(k)})$ `(Proj SGD)`

$\quad \lambda^{(k+1)} \leftarrow \text{proj}_{\Delta_n}(\lambda^{(k)} + \gamma h_\lambda^{(k)})$ `(Proj SGA)`

Sample $\tau \sim \mathcal{U}(\{1, \cdots, K\})$

return $\bar{x} \leftarrow x^{(\tau)}, \bar{y}_i \leftarrow y_i^{(\tau-1)}, \bar{\lambda} \leftarrow \lambda^{(\tau)}$

For $n = 1$, it reduces to the TTSA algorithm

Hong, M., et al. *A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic.* arXiv 2020.

Single-loop Projected Gradient Descent-Ascent

IBM

for $k = 1, 2, \cdots, K$ **do**

$\quad \mathbf{y}^{(k+1)} \leftarrow \mathbf{y}^{(k)} - \beta \mathbf{h}^{(k)}$ (SGD)

$\quad x^{(k+1)} \leftarrow \text{proj}_{\mathcal{X}}(x^{(k)} - \alpha h_x^{(k)})$ (P-SGD)

$\quad \lambda^{(k+1)} \leftarrow \text{proj}_{\Delta_n}(\lambda^{(k)} + \gamma h_\lambda^{(k)})$ (P-SGA)

Sample $\tau \sim \mathcal{U}(\{1, \cdots, K\})$

return $\bar{x} \leftarrow x^{(\tau)}, \bar{y}_i \leftarrow y_i^{(\tau-1)}, \bar{\lambda} \leftarrow \lambda^{(\tau)}$

- $\mathbf{y}^{(k)} \triangleq \{y_1^{(k)}, \ldots, y_n^{(k)}\}$
- $\mathbf{h}^{(k)} \triangleq \{\nabla_{y_1} g_1(x^{(k)}, y_1^{(k)}; \zeta_1), \ldots, \nabla_{y_n} g_n(x^{(k)}, y_n^{(k)}; \zeta_n)$
- $h_x^{(k)} \approx \sum_{i \in [n]} \lambda_i^{(k)} \underbrace{\overline{\nabla} f_i(x^{(k)}, y_i^{(k+1)}; \xi_i)}_{**}$

$$** \triangleq \nabla_x f_i(x, y_i; \xi_i) - \nabla^2_{xy_i} g_i(x, y_i; \zeta_i) \left[\nabla^2_{y_i} g_i(x, y_i; \zeta_i)\right]^{-1} \nabla_{y_i} f_i(x, y_i; \xi_i)$$

- $h_\lambda^{(k)} \triangleq [f_1(x^{(k)}, y_1^{(k+1)}; \xi_1), \ldots, f_n(x^{(k)}, y_n^{(k+1)}; \xi_n)]^\top$

Single-loop Projected Gradient Descent-Ascent

IBM

Learning rates

$$\alpha \sim \mathcal{O}\left(K^{-\frac{3}{5}}\right), \beta \sim \mathcal{O}\left(K^{-\frac{2}{5}}\right), \gamma \sim \mathcal{O}\left(\sqrt{n}K^{-\frac{3}{5}}\right)$$

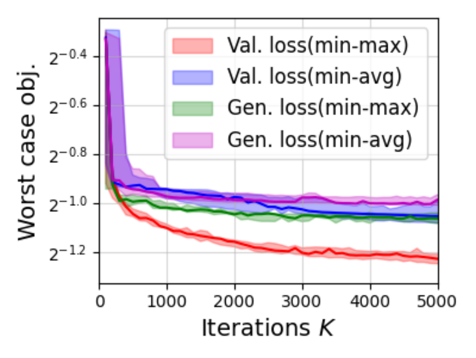
Convergence rate of `MORBiT`

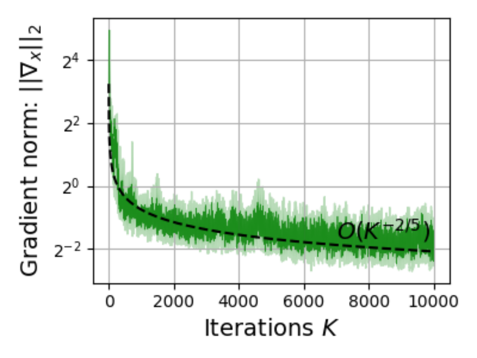
$$\begin{aligned} \mathbb{E}\left[\|\hat{x}(\bar{x}) - \bar{x}\|^2\right] &\leqslant \widetilde{\mathcal{O}}(\sqrt{n}K^{-2/5})^{\dagger}, \\ \mathbb{E}\left[\max_{i \in [n]} \|\bar{y}_i - y_i^\star(\bar{x})\|^2\right] &\leqslant \widetilde{\mathcal{O}}(\sqrt{n}K^{-2/5}), \\ \max_{\lambda} \mathbb{E}\left[F(\bar{x}, \lambda)\right] - \mathbb{E}[F(\bar{x}, \bar{\lambda})] &\leqslant \widetilde{\mathcal{O}}(\sqrt{n}K^{-2/5}). \end{aligned}$$

† $\hat{x}(\bar{x})$ is the **proximal map**

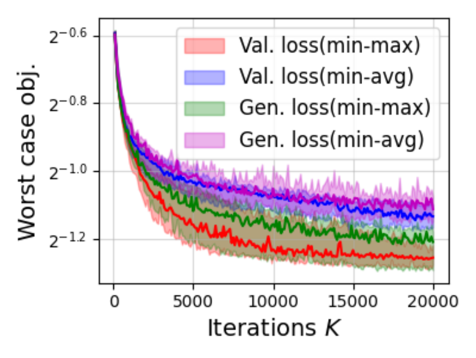
Main Proof Challenges

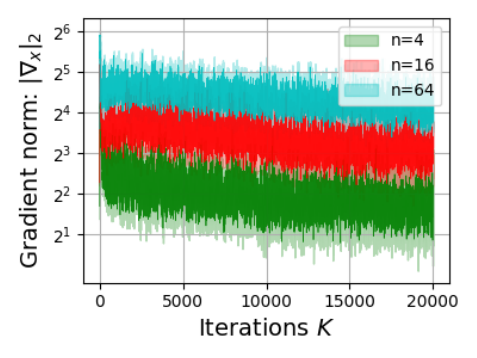
- Handling the non-smooth $\max_{i \in [n]}$
- Establishing convergence of each $\bar{y}_i$ *simultaneously* $\forall i \in [n]$
- Descent equation involving $(n+1)$ sequences
- Establishing convergence of λ


Worst case obj.
$2^{-0.4}$
$2^{-0.6}$
$2^{-0.8}$
$2^{-1.0}$
$2^{-1.2}$
0
1000
2000
3000
4000
5000
Iterations K
Val. loss(min-max)
Val. loss(min-avg)
Gen. loss(min-max)
Gen. loss(min-avg)


Gradient norm: $||\nabla_x||_2$
2^4
2^2
2^0
2^{-2}
0
2000
4000
6000
8000
10000
Iterations K
$O(K^{-2/5})$

Meta-learning for HPO





Concluding Remarks

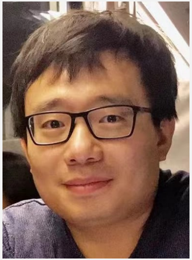
Caveats:

- Useful when gap between max and mean is large
- From a learning perspective, x should have enough capacity to simultaneously optimize all $\mathtt{UL}$ objectives
- Room for improvement for large n

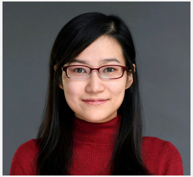
Arxiv Preprint: `https://arxiv.org/pdf/2203.01924.pdf`

Gu, Alex, et al. *Min-Max Bilevel Multi-objective Optimization with Applications in Machine Learning.* arXiv 2022.

Alex Gu



Songtao Lu



Tsui-Wei (Lily) Weng

Thank You!

Positioning

Problem/Method	Bilevel	Multi-obj	Min-max	Single-loop	$\mathcal{X} \subset \mathbb{R}^{d_x}$	$\mathcal{Y}_i \subset \mathbb{R}^{d_{y_i}}$
Distributionally Robust Learning	†	✗	✓	-	✓	✗
Adversarially Robust Learning	†	✗	✓	-	✗	✓
Multi-task Learning (MTL)	†	☐	✗	-	✗	✗
Robust MTL [Mehta et al., 2012]	†	✓	✓	-	✗	✗
Meta-learning	†	☐	✗	-	✗	✗
HiBSA [Lu et al., 2020]	✗	✗	✓	✓	✓	-
GDA [Lin et al., 2020]	✗	✗	✓	✓	✗	-
TR-MAML [Collins et al., 2020]	✗	✓	✓	✓	✓	-
BSA [Ghadimi and Wang, 2018]	✓	✗	✗	✗	✓	✗
TTSA [Hong et al., 2020]	✓	✗	✗	✓	✓	✗
StocBio [Ji et al., 2021]	✓	✗	✗	✗	✗	✗
MRBO [Yang et al., 2021]	✓	✗	✗	✓	✗	✗
VRBO [Yang et al., 2021]	✓	✗	✗	✗	✗	✗
ALSET [Chen et al., 2021]	✓	✗	✗	✓	✗	✗
STABLE [Chen et al., 2022]	✓	✗	✗	✓	✓	✗
MMB [Hu et al., 2022]	✓	✗	✓	✓	✗	✗
MORBiT (Ours)	✓	✓	✓	✓	✓	✗

References I

Nishant A. Mehta, Dongryeol Lee, and Alexander G. Gray. Minimax multi-task learning and a generalized loss-compositional paradigm for mtl. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, page 2150–2158, 2012.

Songtao Lu, Ioannis Tsaknakis, Mingyi Hong, and Yongxin Chen. Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing*, 68:3676–3691, 2020.

Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 6083–6093. PMLR, 2020.

Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Task-robust model-agnostic meta-learning. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020. URL `https://arxiv.org/pdf/2007.05170.pdf`.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 4882–4892. PMLR, 2021.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. *arXiv preprint arXiv:2101.11203*, 2021.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 34:25294–25307, 2021.

Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2466–2488. PMLR, 2022.

Quanqi Hu, Yongjian Zhong, and Tianbao Yang. Multi-block min-max bilevel optimization with applications in multi-task deep auc maximization. *arXiv preprint arXiv:2206.00260*, 2022.