Stochastic Bi-level Problems: Variants, Algorithms, and Implementation

Combinatorial Optimization and Machine Learning | Lecture 6

Sanjeeb Dash / Parikshit Ram

June 27, 2023

JPOC'13
Les 20 ans !

Plan

1. Bi-level Problem Variants & Algorithms
   - Handling Constraints
   - Non-Singleton LL
   - Specialized solvers
   - Multi-objective Bi-level
2. Implementation Details
3. Further Reading

Outline

Constrained UL + Unconstrained Singleton LL

UL constrained $\theta \in \Theta \subset \mathbb{R}^{d_u}$ + LL unconstrained, unique solution

$$\min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{1}$$

Handle UL constrained via **projected gradient descent** for UL update (TTSA [Hong et al., 2020, 2023], STABLE [Chen et al., 2022])

Constrained UL + Constrained Singleton LL

UL constrained + LL constrained $\phi \in \Phi(\theta) \subset \mathbb{R}^{d_l}$, unique solution

$$\min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg \min_{\phi \in \Phi(\theta) \subset \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{2}$$

We can obtain IG based UL gradient with additional assumptions [Giovannelli et al., 2021].

Main modification. How do we compute the **IG** ${d\phi^\star(\theta)^\top}/{d\theta}$?

IG for Unconstrained LL

Since $\phi^\star(\theta)$ is a LL solution, the stationarity condition in the unconstrained case gives us

$$\nabla_\phi f_l(\theta, \phi^\star(\theta)) = 0 \tag{3}$$

Taking the derivative w.r.t. θ we have (by Implicit Function Theorem):

$$\nabla^2_{\theta\phi} f_l(\theta, \phi^\star(\theta)) + \frac{d\phi^\star(\theta)}{d\theta}^\top \underbrace{\nabla^2_\phi f_l(\theta, \phi^\star(\theta))}_{\text{Hessian}H} = 0. \tag{4}$$

Assuming the Hessian H is invertible,

$$\frac{d\phi^\star(\theta)}{d\theta}^\top = -\underbrace{\nabla^2_{\theta\phi} f_l(\theta, \phi^\star(\theta))}_{D_u \times D_l} \cdot \underbrace{\nabla^2_\phi f_l(\theta, \phi^\star(\theta))^{-1}}_{D_l \times D_l}. \tag{5}$$

LL Constraints

LL feasible set

$$\Phi(\theta) = \left\{ \phi \in \mathbb{R}^{d_l} : P_l(\theta, \phi) \leq 0, Q_l(\theta, \phi) = 0 \right\} \tag{6}$$

At $\phi^\star(\theta) \in \Phi(\theta)$, the following assumptions need to hold

- ⇛ The gradients of the active constraints are linearly independent (LICQ – linearly independent constraint qualification)
- ⇛ Strict complimentarity condition holds
- ⇛ Sufficient second-order optimality conditions are satisfied

Define the Lagrangian function with Lagrange multipliers λ_P, λ_Q as

$$\mathscr{L}(\theta, \phi, \lambda_P, \lambda_Q) = f_l(\theta, \phi) + \lambda_P^\top P(\theta, \phi) + \lambda_Q^\top Q(\theta, \phi), \tag{7}$$

Constrained LL: KKT conditions

Assuming unique Lagrange multipliers $\lambda_P(\theta), \lambda_Q(\theta)$ at $\phi^\star(\theta)$ under LICQ, this is the first-order KKT system for the LL problem:

$$\nabla_\phi f_l(\theta, \phi^\star(\theta)) + \nabla_\phi P_l(\theta, \phi^\star(\theta))^\top \cdot \lambda_P(\theta) + \nabla_\phi Q_l(\theta, \phi^\star(\theta))^\top \cdot \lambda_Q(\theta) = 0 \tag{8}$$

$$\lambda_P(\theta) \odot P(\theta, \phi^\star(\theta)) = 0 \tag{9}$$

$$Q(\theta, \phi^\star(\theta)) = 0 \tag{10}$$

With $\varphi^\star(\theta) = [\phi^\star(\theta)^\top, \lambda_P(\theta)^\top, \lambda_Q(\theta)^\top]^\top \in \mathbb{R}^{(d_l+p_l+q_l)}$,
we can rewrite the above as $G(\theta, \varphi^\star(\theta)) = 0$.
Applying Implicit Function Theorem (under appropriate assumptions)

$$\nabla_\theta G^\top + \frac{d\varphi^\star(\theta)^\top}{d\theta} \nabla_\varphi G^\top = 0 \quad \Rightarrow \quad \frac{d\varphi^\star(\theta)^\top}{d\theta} = -\nabla_\theta G^\top \cdot \nabla_\varphi (G^\top)^{-1} \tag{11}$$

Constrained Singleton LL: Implicit Gradient

$$\frac{d\varphi^\star(\theta)^\top}{d\theta} = -\nabla_\theta G(\theta, \varphi^\star(\theta))^\top \cdot \nabla_\varphi (G(\theta, \varphi^\star(\theta))^\top)^{-1} \tag{12}$$

$$\nabla_\theta G(\theta, \varphi^\star(\theta))^\top = \left[\underbrace{\nabla^2_{\theta\phi} \mathscr{L}(\theta, \varphi^\star(\theta))}_{\mathbb{R}^{d_u \times d_l}} \quad \underbrace{\nabla_\theta P_l(\theta, \phi^\star(\theta))^\top \odot \lambda_P(\theta)^\top}_{\mathbb{R}^{d_u \times p_l}} \quad \underbrace{\nabla_\theta Q_l(\theta, \phi^\star(\theta))^\top}_{\mathbb{R}^{d_u \times q_l}} \right] \tag{13}$$

Constrained Singleton LL: Implicit Gradient

$$\frac{d\varphi^\star(\theta)^\top}{d\theta} = -\nabla_\theta G(\theta, \varphi^\star(\theta))^\top \cdot \nabla_\varphi (G(\theta, \varphi^\star(\theta))^\top)^{-1} \tag{14}$$

$$\nabla_\varphi (G(\theta, \varphi^\star(\theta))^\top) = \begin{bmatrix} \underbrace{\nabla^2_\phi \mathscr{L}(\theta, \varphi^\star(\theta))}_{\mathbb{R}^{d_l \times d_l}} & \underbrace{\nabla_\phi P_l(\theta, \phi^\star(\theta))^\top \odot \lambda_P(\theta)^\top}_{\mathbb{R}^{d_l \times p_l}} & \underbrace{\nabla_\phi Q_l(\theta, \phi^\star(\theta))^\top}_{\mathbb{R}^{d_l \times q_l}} \\ \underbrace{\left(\nabla_\phi P_l(\theta, \phi^\star(\theta))^\top\right)^\top}_{\mathbb{R}^{p_l \times d_l}} & \underbrace{\mathtt{diag}(P_l(\theta, \phi^\star(\theta)))}_{\mathbb{R}^{p_l \times p_l}} & \underbrace{0}_{\mathbb{R}^{p_l \times q_l}} \\ \underbrace{\left(\nabla_\phi Q_l(\theta, \phi^\star(\theta))^\top\right)^\top}_{\mathbb{R}^{q_l \times d_l}} & \underbrace{0}_{\mathbb{R}^{q_l \times p_l}} & \underbrace{0}_{\mathbb{R}^{q_l \times q_l}} \end{bmatrix} \tag{15}$$

Constrained Singleton LL: Implicit Gradient

$$\frac{d\varphi^\star(\theta)^\top}{d\theta} = -\nabla_\theta G(\theta, \varphi^\star(\theta))^\top \cdot \nabla_\varphi (G(\theta, \varphi^\star(\theta))^\top)^{-1} \tag{16}$$

$$\frac{d\phi^\star(\theta)^\top}{d\theta} = \underbrace{\frac{d\varphi^\star(\theta)^\top}{d\theta} \cdot \begin{bmatrix} I_{d_l} \\ 0 \end{bmatrix}}_{\substack{\text{extract first } d_l \text{ columns} \\ \text{of } d\varphi^\star(\theta)/d\theta}} \tag{17}$$

Non-Singleton LL

LL optimal set is non-singleton

$$S(\theta) = \arg \min_{\phi \in \Phi(\theta) \subset \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{18}$$

Non-singleton LL

Implicit Gradient not available

Optimistic version.

$$\min_{\substack{\theta \in \Theta \subset \mathbb{R}^{d_u}, \\ \phi \in S(\theta)}} f_u(\theta, \phi) \quad \texttt{subject to} \quad S(\theta) = \arg \min_{\phi \in \Phi(\theta) \subset \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{19}$$

Best handled the surrogate optimal value function [Sow et al., 2022a]

Non-Singleton LL

Pessimistic version.

$$\min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} \max_{\phi \in S(\theta)} f_u(\theta, \phi) \quad \texttt{subject to} \quad S(\theta) = \arg \min_{\phi \in \Phi(\theta) \subset \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{20}$$

Robustness via Pessimism

- $\Rrightarrow$ *min-min*: $\min_\theta \min_{\phi \in S(\theta)} f_u$ requires **just a single** $\phi \in S(\theta)$ to be "good"
- $\Rrightarrow$ *min-max*: $\min_\theta \max_{\phi \in S(\theta)} f_u$ requires **all** $\phi \in S(\theta)$ to be "good"
- $\Rrightarrow$ If $\min_{\phi \in S(\theta)}$ hard, inefficient *min-min* solution + approximations can be "bad"
  - $\Rrightarrow$ *min-max* does not require the $\min_{\phi \in S(\theta)}$ to be solved well
  - $\Rrightarrow$ Puts onus on $\min_\theta$ to find a "generally good" $S(\theta)$
  - $\Rrightarrow$ Makes solution "robust" to $\min_{\phi \in S(\theta)}$ quality
- $\Rrightarrow$ Cost of robustness: generally $\min_\theta \min_{\phi \in S(\theta)} f_l < \min_\theta \max_{\phi \in S(\theta)} f_l$ (for exact sols)

Non-Singleton LL

Pessimistic version.

$$\min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} \max_{\phi \in S(\theta)} f_u(\theta, \phi) \quad \texttt{subject to} \quad S(\theta) = \arg \min_{\phi \in \Phi(\theta) \subset \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{21}$$

$\Rrightarrow$ With multiple ML model types, $S(\theta)$ is the non-singleton set of "LL optimal" models for hyperparameter θ in **bi-level hyperparameter optimization**.

$\Rrightarrow$ With "overparameterized" models, non-singleton $S(\theta)$ appear even in **bi-level representation learning** and other **bi-level adversarial training**

$\Rrightarrow$ More situations exist ...

Non-singleton LL

Robust Learning

In all such cases, we have easy access to some $\phi \in S(\theta)$, but $\min_{\phi \in S(\theta)} f_u$ is not feasible – it is more robust to find θ such that $\max_{\phi \in S(\theta)} f_u$ is optimized.

Not much attention in ML literature, so lots of opportunity!

Specialized LL solvers + Gradient Unrolling [Fan et al., 2021]

Unconstrained singleton strongly convex LL

$$\min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} f_l(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{22}$$

Main idea.

⇛ Solve LL with Sign-SGD [Bernstein et al., 2018]

⇛ Unrolled gradient simplifies UL descent step to GD

Gradient Unrolling:
For any general $t > 1$

$$\underbrace{\frac{d\varphi^{t+1}}{d\theta}}_{Z_{t+1} \in \mathbb{R}^{d_l \times d_u}} = \underbrace{\frac{\partial \varphi^{t+1}}{\partial \varphi^{t}}}_{A_{t+1} \in \mathbb{R}^{d_l \times d_l}} \cdot \underbrace{\frac{d\varphi^{t}}{d\theta}}_{Z_t} + \underbrace{\frac{\partial \varphi^{t+1}}{\partial \theta}}_{B_{t+1} \in \mathbb{R}^{d_l \times d_u}} \tag{23}$$

Recursively compute $Z_{t+1} = A_{t+1} Z_t + B_{t+1}$ by “unrolling” the gradient [Franceschi et al., 2017].

Specialized LL solvers + Gradient Unrolling

LL update with SignSGD:

$$\varphi^{t+1} \leftarrow \varphi^t - \beta \cdot \mathtt{sign}\left(\nabla_\phi f_l(\theta^k, \varphi^t)\right) \tag{24}$$

Main simplification:

$$\frac{\partial}{\partial x}\mathtt{sign}(x) = 0 \qquad \text{(almost surely)} \tag{25}$$

This gives us

$$A_{t+1} = \frac{\partial \varphi^{t+1}}{\partial \varphi^t} = I_{d_l} \qquad B_{t+1} = \frac{\partial \varphi^{t+1}}{\partial \theta} = 0_{d_l \times d_u} \tag{26}$$

Then the $Z_{t+1} = A_{t+1}Z_t + B_{t+1}$ recursion implies $Z_T = 0$.

Specialized LL solvers + Gradient Unrolling

Then the UL gradient reduces to:

$$\nabla_\theta F(\theta) = \nabla_\theta f_u(\theta, \phi^\star(\theta)) + \boxed{\frac{d\phi^\star(\theta)^\top}{d\theta}} \cdot \nabla_\phi f_u(\theta, \phi^\star(\theta)) \tag{27}$$

$$= \nabla_\theta f_u(\theta, \phi^\star(\theta)) \tag{28}$$

Specialized LL solvers + Gradient Unrolling

Algorithm 1 Sign-SGD based Bi-level Optimization [Fan et al., 2021]

Input: Initialization θ^0, ϕ^0, initial learning rates α^0, β^0 for UL and LL resp.

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

  `// Solve LL (approx.) with SignSGD for current` θ^k

  $\varphi^0 \leftarrow \phi^k$

  for $t = 1, 2, \ldots, T$ **do**

    $\varphi^{t+1} \leftarrow \varphi^t - \beta^t \cdot \mathtt{sign}\left(\nabla_\phi f_l(\theta, \phi)\big|_{\theta=\theta^k, \phi=\varphi^t} \right)$

  $\phi^{k+1} \leftarrow \varphi^{T+1}$

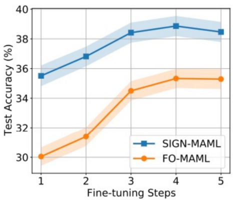
  `// UL descent step with simple partial gradient`

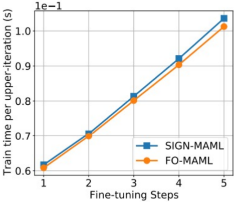
  $\theta^{k+1} \leftarrow \theta^k - \alpha^k \cdot \nabla_\theta f_u(\theta, \phi)\big|_{\theta=\theta^k, \phi=\phi^{k+1}}$

return θ^{K+1}, ϕ^{K+1}

Specialized LL solvers + Gradient Unrolling

- ✓ Easy to compute UL descent step (without explicitly ignoring the IG)
- ✓ Improved performance compared to methods that ignore IG, but just as fast
- ✗ SignSGD in the LL might slow convergence
- ✗ No theoretical convergence guarantees (open question!)





Digression: Multi-objective Optimization

$$\min_{\theta \in \Theta} \{f_1(\theta), \ldots, f_n(\theta)\} \tag{29}$$

- ⇛ Multiple objectives $f_i : \Theta \to \mathbb{R}$
- ⇛ Connected by single decision variable θ
- ⇛ Challenging: (usually) Trades-off between objectives
- ⇛ Dominance among solutions $\theta, \vartheta \in \Theta$:

$$\theta \succ \vartheta \Rightarrow \forall i \in [n], f_i(\theta) \leq f_i(\vartheta), \exists i \in [n], f_i(\theta) < f_i(\vartheta). \tag{30}$$

- ⇛ Pareto optimal solution θ: $\nexists \vartheta \in \Theta : \vartheta \succ \theta$
- ⇛ Pareto front: Set of (all) Pareto optimal solutions
  - ⇛ Provides a set of solutions; we might have to pick one

Digression: Multi-objective

$$\min_{\theta \in \Theta} \{ f_1(\theta), \ldots, f_n(\theta) \} \tag{31}$$

Options other than Pareto optimality

$\Rrightarrow$ Lexicographic ordering among objective – obj 1 $\succ$ obj 2 $\succ \cdots$

$\Rrightarrow$ A domain specific scalarization with scalars $\lambda_1, \ldots, \lambda_n$:

$$\min_{\theta \in \Theta} \sum_{i \in [n]} \lambda_i f_i(\theta) \qquad \text{single objective optimization} \tag{32}$$

Digression: Robust Multi-objective

- ⇛ We do not know the best scalarization
- ⇛ Optimize θ for the worst-case scalarization

$$\min_{\theta \in \Theta} \max_{\lambda_i, i \in [n]} \sum_{i \in [n]} \lambda_i f_i(\theta) \qquad \text{minmax optimization} \tag{33}$$

Digression: Robust Multi-objective

$$\min_{\theta \in \Theta} \max_{\lambda_i, i \in [n]} \sum_{i \in [n]} \lambda_i f_i(\theta) \qquad \text{minmax optimization} \tag{34}$$

Meaningful to put constraints on $\{\lambda_i, i \in [n]\}$

$\Rrightarrow$ Non-negative $\lambda_i \geq 0 \forall i \in [n]$ – corresponds to weighted (importance) sum of objs

$\Rrightarrow$ Simplex cst $\sum_{i \in [n]} \lambda_i = 1$ – weights induce a (discrete) distribution over objs

$\Rrightarrow$ Gives us the n-simplex $\Delta_n = \left\{\lambda = [\lambda_1, \ldots, \lambda_n] \in \mathbb{R}^n, \lambda_i \in [0,1] \forall i \in [n], \mathbf{1}_n^\top \lambda = 1\right\}$

$$\min_{\theta \in \Theta} \max_{\substack{\lambda = [\lambda_1, \ldots, \lambda_n] \\ \lambda \in \Delta_n}} \sum_{i \in [n]} \lambda_i f_i(\theta) \quad \equiv \quad \min_{\theta \in \Theta} \max_{i \in [n]} f_i(\theta) \tag{35}$$

Digression: Robust Multi-objective

$$\min_{\theta \in \Theta} \max_{\substack{\lambda = [\lambda_1, \ldots, \lambda_n] \\ \lambda \in \Delta_n}} \sum_{i \in [n]} \lambda_i f_i(\theta) \quad \equiv \quad \min_{\theta \in \Theta} \max_{i \in [n]} f_i(\theta) \tag{36}$$

$\Rrightarrow$ Optimize for the worst-case obj

$\Rrightarrow$ Find θ so that all objs are "good" – robust solution

$\Rrightarrow$ Worst-case scalarization – solution robust to scalarization

Robust Multi-objective Bi-Level

$$\begin{aligned} \min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} \max_{i \in [n]} f_{u,i}(\theta, \phi_i^\star(\theta)) \quad &\equiv \quad \min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} \max_{\lambda \in \Delta_n} \sum_{i \in [n]} \lambda_i f_{u,i}(\theta, \phi_i^\star(\theta)) \\ \texttt{subject to} \quad \forall i \in [n],\ \phi_i^\star(\theta) &= \arg \min_{\phi_i \in \Phi_i = \mathbb{R}^{d_{l,i}}} f_{l,i}(\theta, \phi_i) \end{aligned} \tag{37}$$

- ⇛ Multiple $n > 1$ UL/LL obj pairs $\{f_{u,i}, f_{l,i}\}$
- ⇛ Shared UL variable θ
- ⇛ Per-objective pair LL variable ϕ_i
- ⇛ LL variables can have different domains – that is, $\Phi_i \neq \Phi_j$
- ⇛ UL constrained
- ⇛ LL unconstrained + singleton solution

Single-loop algorithm: MORBiT

Algorithm 2 Multi-obj Robust Bi-level Two-timescale Alg [Gu et al., 2022, 2023]

Input: Initialization $\theta^0, \lambda^0, \phi_i^0, i \in [n]$

Input: Initial learning rates $\alpha^0, \beta^0, \gamma^0$ for UL, LL and simplex vars resp.

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

`// Single LL descent step per LL objective`

$$\forall i \in [n],\ \phi_i^{k+1} \leftarrow \phi_i^k - \beta^k \left. \nabla_{\phi_i} f_{l,i}(\theta, \phi_i) \right|_{\theta=\theta^k, \phi_i=\phi_i^k}$$

`// UL descent step with per-objective pair IG`

$$\theta^{k+1} \leftarrow \mathscr{P}_\Theta \left(\theta^k - \alpha^k \cdot \sum_{i \in [n]} \lambda_i^k \cdot \left. \left[\nabla_\theta f_{u,i}(\theta, \phi_i) - \bar{\nabla} f_{u,i}(\theta, \phi_i) \right] \right|_{\theta=\theta^k, \phi_i=\phi_i^{k+1}} \right)$$

`// Simplex variable ascent step`

$$\lambda^{k+1} \leftarrow \mathscr{P}_{\Delta_n} \left(\lambda^k + \gamma^k \cdot \left[f_{u,1}(\theta^{k+1}, \phi_1^{k+1}), \cdots, f_{u,n}(\theta^{k+1}, \phi_n^{k+1}) \right]^\top \right)$$

return $\theta^{K+1}, \{\phi_i^{K+1}\}_{i \in [n]}$

Single-loop algorithm: MORBiT

Algorithm 3 Multi-obj Robust Bi-level Two-timescale Alg [Gu et al., 2022, 2023]

Input: Initialization $\theta^0, \lambda^0, \phi_i^0, i \in [n]$

Input: Initial learning rates $\alpha^0, \beta^0, \gamma^0$ for UL, LL and simplex vars resp.

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

$\forall i \in [n],\ \phi_i^{k+1} \leftarrow \phi_i^k - \beta^k \left. \nabla_{\phi_i} f_{l,i}(\theta, \phi_i) \right|_{\theta=\theta^k, \phi_i=\phi_i^k}$ `// Single LL descent step per LL objective`

$\theta^{k+1} \leftarrow \mathscr{P}_\Theta \left(\theta^k - \alpha^k \cdot \sum_{i\in[n]} \lambda_i^k \cdot \left. \left[\nabla_\theta f_{u,i}(\theta, \phi_i) - \bar{\nabla} f_{u,i}(\theta, \phi_i) \right] \right|_{\theta=\theta^k, \phi_i=\phi_i^{k+1}} \right)$ `// UL descent step with per-objective pair IG`

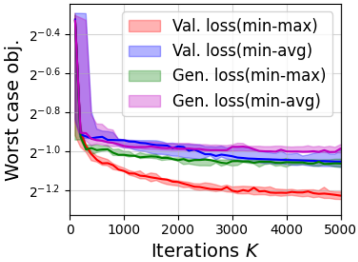
$\lambda^{k+1} \leftarrow \mathscr{P}_{\Delta_n} \left(\lambda^k + \gamma^k \cdot \left[f_{u,1}(\theta^{k+1}, \phi_1^{k+1}), \cdots, f_{u,n}(\theta^{k+1}, \phi_n^{k+1}) \right]^\top \right)$ `// Simplex variable ascent step`

return $\theta^{K+1}, \left\{ \phi_i^{K+1} \right\}_{i\in[n]}$

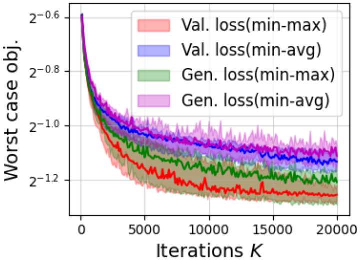
$$\bar{\nabla} f_{u,i}(\theta, \phi_i) = \nabla^2_{\theta,\phi_i} f_{l,i}(\theta, \phi_i) \cdot \nabla^2_{\phi_i,\phi_i} f_{l,i}(\theta, \phi_i)^{-1} \cdot \nabla_{\phi_i} f_{u,i}(\theta, \phi_i) \tag{38}$$

Empirical Evaluation: Multi-task Learning

Representation Learning



Hyperparameter Optimization



$\Rrightarrow$ UL var θ – shared representation

$\Rrightarrow$ LL var ϕ_i – per-task learner

$\Rrightarrow$ UL var θ – shared hyperparameter

$\Rrightarrow$ LL var ϕ_i – per-task model

Outline

Computational Bottleneck: IG-based Schemes

$$\bar{\nabla} f_u(\theta,\phi) = \nabla^2_{\theta\phi} f_l(\theta,\phi) \cdot \nabla^2_{\phi} f_l(\theta,\phi)^{-1} \cdot \nabla_{\phi} f_u(\theta,\phi) \tag{39}$$

- ⇛ Inverse Hessian-vector Product (`HvP`) $[\nabla^2_\phi f_l]^{-1} \cdot v$
  - ⇛ Conjugate Gradient needs `HvP` $\boxed{\nabla^2_\phi f_l \cdot v}$
  - ⇛ Neumann series approx needs `HvP`:

$$[I - \nabla^2_\phi f_l]^i \cdot v = [I - \nabla^2_\phi f_l]^{i-1} \cdot [I \cdot v - \boxed{\nabla^2_\phi f_l \cdot v}]$$

- ⇛ Jacobian-vector product (`JvP`) $\boxed{\nabla^2_{\theta\phi} f_l \cdot v}$

Input: PD sym $H \in \mathbb{R}^{d \times d}$, vec $v \in \mathbb{R}^d$
Input: Init $x_0 \in \mathbb{R}^d$, prec $\varepsilon > 0$, max iters n
$d_0 = r_0 \leftarrow v - Hx_0$
for $i \leftarrow 0, 1, \ldots, n$ **do**
  $\alpha_i \leftarrow (d_i^\top r_i)/(d_i^\top \boxed{Hd_i})$
  $x_{i+1} \leftarrow x_i + \alpha_i d_i$
  $r_{i+1} \leftarrow r_i - \alpha_i \boxed{Hd_i}$
  $\beta_{i+1} \leftarrow (r_{i+1}^\top r_{i+1})/(r_i^\top r_i)$
  $d_{i+1} \leftarrow r_{i+1} + \beta_{i+1} d_i$
  if $r_{i+1}^\top r_{i+1} \le \varepsilon$ **then**
    return x_i
return x_{n+1}

Computational Bottleneck: Gradient-Unrolling Schemes

Backward-mode hypergradient $\left({d\varphi^T}/{d\theta}\right)^\top v$ for some $v \in \mathbb{R}^{d_l}$

- $\Rrightarrow$ $\alpha_T \leftarrow v$, $g \leftarrow 0 \in \mathbb{R}^{d_u}$
- $\Rrightarrow$ for $t = (T-1) \rightarrow 1$
  - $\Rrightarrow$ Compute A_{t+1}, B_{t+1}
  - $\Rrightarrow$ Update $g \leftarrow g + B_{t+1} \cdot \alpha_{t+1}$
  - $\Rrightarrow$ Update $\alpha_t \leftarrow A_{t+1} \cdot \alpha_{t+1}$
- $\Rrightarrow$ Return g

With $\varphi^t \leftarrow \varphi^{t-1} - \beta \nabla_\phi f_l(\theta^k, \varphi^{t-1})$

- $\Rrightarrow$ $A_t = \left(I - \beta \nabla^2_\phi f_l(\theta^k, \varphi^{t-1})\right)$,
- $\Rrightarrow$ $B_t = -\beta \nabla^2_{\theta\phi} f_l(\theta^k, \varphi^{t-1})$
- $\Rrightarrow$ $A_t \cdot v = (I \cdot \alpha - \beta \; \boxed{\nabla^2_\phi f_l \cdot v}\;)$
- $\Rrightarrow$ $B_t \cdot v = -\beta \; \boxed{\nabla^2_{\theta\phi} f_l(\theta^k, \varphi^{t-1}) \cdot v}$

Each iteration needs a `HvP` and a `JvP`!

Computational Bottleneck: Ordering matters!

Consider following general operation (Hessian-vector if $\phi = \varphi$):

$$\begin{gathered}\nabla^2_{\varphi\phi} f(\varphi, \phi) \cdot v = \left(\nabla_\varphi \left(\nabla_\phi f(\varphi, \phi)\right)^\top\right) \cdot v, \\ f : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}, \quad \varphi \in \mathbb{R}^D, \quad \phi \in \mathbb{R}^d, \quad v \in \mathbb{R}^d\end{gathered} \tag{40}$$

Order of operations:

⇛ Compute gradient $\nabla_\phi f(\varphi, \phi) \in \mathbb{R}^d$

✗ Compute Jacobian $\nabla_\varphi \left(\nabla_\phi f(\varphi, \phi)\right)^\top \in \mathbb{R}^{D\times d}$

✗ Compute Jacobian-vector product: $\left(\nabla_\theta \left(\nabla_\phi f(\varphi, \phi)\right)^\top\right) \cdot v \in \mathbb{R}^D$

Computational Bottleneck: Ordering matters!

$$\nabla^2_{\varphi\phi} f(\varphi,\phi)\cdot v = \left(\nabla_\varphi \left(\nabla_\phi f(\varphi,\phi)\right)^\top\right)\cdot v,$$
$$f:\mathbb{R}^D\times\mathbb{R}^d\to\mathbb{R},\quad \varphi\in\mathbb{R}^D,\quad \phi\in\mathbb{R}^d,\quad v\in\mathbb{R}^d \tag{41}$$

Operations can be re-ordered assuming v is a constant:

⇛ Compute gradient $\nabla_\phi f(\varphi,\phi)\in\mathbb{R}^d$

✓ Compute gradient-vector dot-product $\nabla_\phi f(\varphi,\phi)^\top v\in\mathbb{R}$ (scalar)

⇛ Compute gradient $\nabla_\varphi\left(\nabla_\phi f(\varphi,\phi)^\top v\right)\in\mathbb{R}^D$

$$\nabla^2_{\varphi,\phi} f(\varphi,\phi)\cdot v = \underbrace{\left(\nabla_\varphi \underbrace{\left(\nabla_\phi f(\varphi,\phi)\right)^\top}_{1\times d}\right)}_{D\times d}\cdot\underbrace{v}_{d\times 1} = \nabla_\varphi \underbrace{\left(\left(\nabla_\phi f(\varphi,\phi)\right)^\top v\right)}_{\texttt{scalar}}, \tag{42}$$

Outline

Existing Literature on Stochastic Bilevel Optimization

Method	Single-loop	$\Theta \subset \mathbb{R}^{d_u}$	$\Phi \subset \mathbb{R}^{d_l}$	Hessian-free	Multi-obj	Min-max
BSA [Ghadimi and Wang, 2018]	✗	✓	✗	✗	✗	✗
TTSA [Hong et al., 2020, 2023]	✓	✓	✗	✗	✗	✗
StocBio [Ji et al., 2021]	✗	✗	✗	✗	✗	✗
MRBO [Yang et al., 2021]	✓	✗	✗	✗	✗	✗
VRBO [Yang et al., 2021]	✗	✗	✗	✗	✗	✗
ALSET [Chen et al., 2021]	✓	✗	✗	✗	✗	✗
BSG [Giovannelli et al., 2021]	✗	✓	✓	✗	✗	✗
SignSGD+UL [Fan et al., 2021]	✗	✓	✗	✓	✗	✗
PDBO [Sow et al., 2022a]	✗	✓	✓	✓	✗	✓
STABLE [Chen et al., 2022]	✓	✓	✗	✗	✗	✗
PZOBO [Sow et al., 2022b]	✗	✗	✗	✓	✗	✗
MORBiT [Gu et al., 2022, 2023]	✓	✓	✗	✗	✓	✓
MMB [Hu et al., 2022]	✓	✗	✗	✗	✗	✓

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`.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023. URL `https://epubs.siam.org/doi/abs/10.1137/20M1387341`.

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. URL `https://proceedings.mlr.press/v151/chen22e/chen22e.pdf`.

Tommaso Giovannelli, Griffin Kent, and Luis Nunes Vicente. Inexact bilevel stochastic gradient methods for constrained and unconstrained lower-level problems. *arXiv preprint arXiv:2110.00604*, 2021. URL `https://arxiv.org/pdf/2110.00604.pdf`.

Daouda Sow, Kaiyi Ji, Ziwei Guan, and Yingbin Liang. A primal-dual approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022a. URL `https://arxiv.org/pdf/2203.01123.pdf`.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018. URL `http://proceedings.mlr.press/v80/bernstein18a/bernstein18a.pdf`.

Chen Fan, Parikshit Ram, and Sijia Liu. Sign-MAML: Efficient model-agnostic meta-learning by SignSGD. In *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2021. URL `https://arxiv.org/pdf/2109.07497.pdf`.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR, 2017. URL `http://proceedings.mlr.press/v70/franceschi17a/franceschi17a.pdf`.

Alex Gu, Songtao Lu, Parikshit Ram, and Lily Weng. Robust multi-objective bilevel optimization with applications in machine learning. In *INFORMS Annual Meeting*, 2022.

Alex Gu, Songtao Lu, Parikshit Ram, and Tsui-Wei Weng. Min-max multi-objective bilevel optimization with applications in robust machine learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/pdf?id=PvDY71zKsvP`.

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

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.

References II

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.

Daouda Sow, Kaiyi Ji, and Yingbin Liang. On the convergence theory for hessian-free bilevel algorithms. *Advances in Neural Information Processing Systems*, 35:4136–4149, 2022b. URL `https://openreview.net/pdf?id=suHUJr7dV5n`.

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.