Algorithms and Analysis of Stochastic Bi-level Optimization Problems

Combinatorial Optimization and Machine Learning | Lecture 5

Sanjeeb Dash / Parikshit Ram

June 27, 2023

 © 2023 IBM Corporation

Plan

1. “Simple” Problem
   - Problem Definition
   - ML Applications
2. Algorithms
   - Meta-algorithm & Implicit Gradient
   - Implicit-Gradient Based Algorithms
   - Standard Schemes and Enhancements
   - Non-IG-based Solutions
3. Analysis
   - Setup
   - Analysis & Results

© 2023 IBM Corporation

Digression: Neural Networks

© 2023 IBM Corporation

Outline

© 2023 IBM Corporation

A Simple Bi-level Problem

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

- $\Rrightarrow$ f_u, f_l smooth, continuous in both θ, ϕ
- $\Rrightarrow$ $f_l(\theta, \cdot)$ is strongly convex in ϕ for all $\theta \Rightarrow$ singleton LL solution

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

© 2023 IBM Corporation

A Simple Bi-level Problem

⇛ Singleton LL solution + No constraints

$$\min_{\theta \in \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{3}$$

ML applications

⇛ Hyperparameter optimization

$$f_u(\theta, \phi) = \sum_{i \in [n]} \ell_{\text{val}}(y_i, M_\phi(x_i)), \quad f_l(\theta, \phi) = \sum_{j \in [m]} \ell(\mathrm{y}_i, M_\phi(\mathrm{x}_i)) + \underbrace{\|\theta \odot \phi\|_2^2}_{\star\star}. \tag{4}$$

© 2023 IBM Corporation

A Simple Bi-level Problem

⇛ Singleton LL solution + No constraints

$$\min_{\theta \in \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{5}$$

ML applications

⇛ Representation learning (for multi-task learning, meta-learning)

$$f_u(\theta, \phi) = \sum_{i \in [n]} \ell(y_i, M_\phi(M_\theta(x_i))), \quad f_l(\theta, \phi) = \sum_{j \in [m]} \ell(\mathrm{y}_i, M_\phi(M_\theta(\mathrm{x}_i))) + \underbrace{\rho \|\phi\|_2^2}_{\star\star}. \tag{6}$$

© 2023 IBM Corporation

A Simple Bi-level Problem

⇛ Singleton LL solution + No constraints

$$\min_{\theta \in \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{7}$$

ML applications

⇛ Model pruning / compression

$$f_u(\theta, \phi) = \sum_{i \in [n]} \ell(y_i, M_{\theta \odot \phi}(x_i)), \quad f_l(\theta, \phi) = \sum_{j \in [m]} \ell(\mathrm{y}_i, M_{\theta \odot \phi}(\mathrm{x}_i)) + \underbrace{\rho \|\phi\|_2^2}_{\star\star}. \tag{8}$$

© 2023 IBM Corporation

A Simple Bi-level Problem

⇛ Singleton LL solution + No constraints

$$\min_{\theta \in \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{9}$$

ML applications

⇛ Data subset selection (condensation / compression / cleaning)** – HPO

⇛ Neural Architecture Search [Liu et al., 2019]** – HPO

⇛ Reinforcement Learning [Hong et al., 2020, 2023]

⇛ Personalized Federated Learning [Fallah et al., 2020]

⇛ Learning parametric loss function

⇛ Learning to optimize [Andrychowicz et al., 2016]

© 2023 IBM Corporation

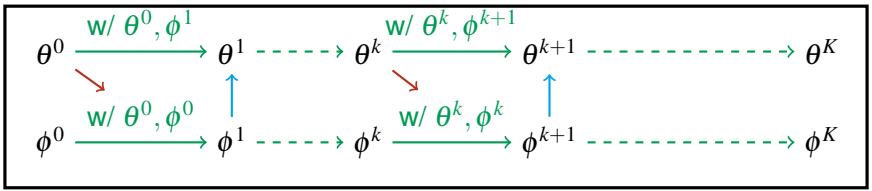
Outline

 © 2023 IBM Corporation

Meta-algorithm

Alternating optimization

- For certain number of iteration (until convergence)
  - Update the LL variable ϕ (using LL objective and current iterate of UL variable)
  - Update the UL variable θ (using UL objective and current iterate of LL variable)



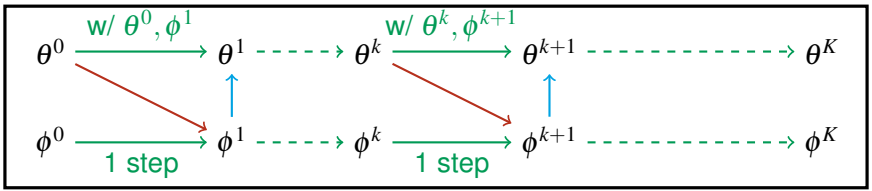
© 2023 IBM Corporation

Meta-algorithm: Single loop

Alternating optimization

⇛ For certain number of iteration (until convergence)

- ⇛ Update the LL variable ϕ with **a single descent step**
- ⇛ Update the UL variable θ with **a single descent step**

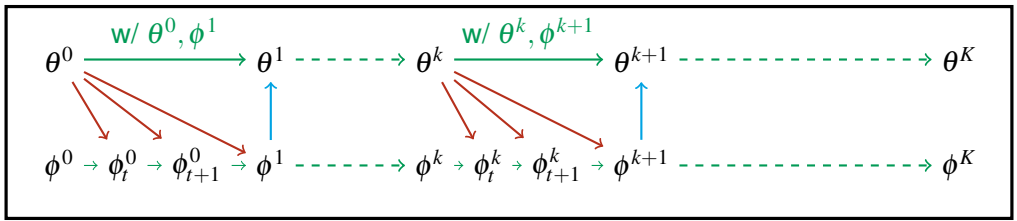


© 2023 IBM Corporation

Meta-algorithm: Double loop

Alternating optimization

- ⇛ For certain number of iteration (until convergence)
  - ⇛ Update the LL variable ϕ
    - ⇛ **For certain number of iterations take descent steps**
  - ⇛ Update the UL variable θ with **a single descent step**



© 2023 IBM Corporation

Single loop vs double loop

- ⇛ Best choice application dependent
- ⇛ Single loop more applicable for sequential learning problems (such as reinforcement learning)
- ⇛ Double loop more communication efficient in distributed optimization
- ⇛ Single loop easier to optimize – less hyperparameters (for example, no need to decide how many LL steps to take)
- ⇛ If properly tuned, double loop can have faster convergence

© 2023 IBM Corporation

LL Descent Step

Strongly convex LL + Singleton LL solution + No constraints

$$\min_{\theta \in \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{10}$$

$$\phi^{k+1} \leftarrow \phi^k - \beta^k h_l^k \tag{11}$$

- $\Rrightarrow$ $h_l^k = \nabla_\phi f_l(\theta, \phi)$ (or a stochastic estimate)
- $\Rrightarrow$ A good candidate, but not the only option

© 2023 IBM Corporation

UL Descent Step

Strongly convex LL + Singleton LL solution + No constraints

$$\min_{\theta \in \mathbb{R}^{d_u}} f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{12}$$

What is the gradient of $F(\theta) = f_u(\theta, \phi^\star(\theta))$?

$$\tilde{\nabla}_\theta F(\theta) = \frac{d}{d\theta} F(\theta) = \underbrace{\frac{\partial}{\partial \theta}}_{\nabla_\theta} f_u(\theta, \phi^\star(\theta)) + \underbrace{\frac{d\phi^\star(\theta)^\top}{d\theta}}_{\textbf{IG} \in \mathbb{R}^{d_u \times d_l}} \cdot \underbrace{\frac{\partial}{\partial \phi}}_{\nabla_\phi} f_u(\theta, \phi^\star(\theta)) \tag{13}$$

Implicit Gradient or **IG**:

- ⇛ Gradient of the LL solution w.r.t. the UL variable
- ⇛ Gradient flow from LL back to UL
- ⇛ Alternating GD ignores the second term involving the **IG**

© 2023 IBM Corporation

Implicit Gradient

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

$$\nabla_\phi f_l(\theta, \phi^\star(\theta)) = 0 \tag{14}$$

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

$$\nabla^2_{\theta\phi} f_l(\theta, \phi^\star(\theta)) + \frac{d\phi^\star(\theta)}{d\theta}^\top \underbrace{\nabla^2_\phi f_l(\theta, \phi^\star(\theta))}_{\text{Hessian } H} = 0. \tag{15}$$

Assuming the Hessian H is invertible at $\phi^\star(\theta)$,

$$\frac{d\phi^\star(\theta)}{d\theta}^\top = -\underbrace{\nabla^2_{\theta\phi} f_l(\theta, \phi^\star(\theta))}_{d_u \times d_l} \cdot \underbrace{\nabla^2_\phi f_l(\theta, \phi^\star(\theta))^{-1}}_{d_l \times d_l}. \tag{16}$$

© 2023 IBM Corporation

Challenges with IG

$$\frac{d\phi^\star(\theta)^\top}{d\theta} = -\nabla^2_{\theta\phi} f_l(\theta, \phi^\star(\theta)) \cdot \nabla^2_{\phi} f_l(\theta, \phi^\star(\theta))^{-1}. \tag{17}$$

- ⇛ Needs Jacobian, involving a Hessian inverse and another second-order derivative!
- ⇛ Assumptions needed
  - ⇛ LL unconstrained stationarity
  - ⇛ LL unique / singleton solution $\phi^\star(\theta)$ (for any given θ)
  - ⇛ LL Hessian at stationarity exists and is invertible

© 2023 IBM Corporation

Bi-level Approximation (BA) Algorithm

Algorithm 1 Bilevel Approximation Algorithm [Ghadimi and Wang, 2018]

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.)  for current` θ^k

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

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

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

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

  `// UL descent step with` **IG**

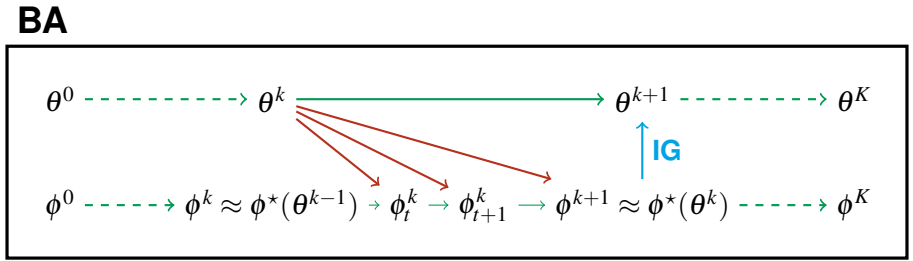
  $\theta^{k+1} \leftarrow \theta^k - \alpha^k \cdot \left[\nabla_\theta f_u(\theta, \phi) - \bar{\nabla} f_u(\theta, \phi)\right]\big|_{\theta=\theta^k, \phi=\phi^{k+1}}$

return θ^{K+1}, ϕ^{K+1}

$$\bar{\nabla} f_u(\theta, \phi) = \underbrace{\nabla^2_{\theta\phi} f_l(\theta, \phi) \cdot \nabla^2_{\phi} f_l(\theta, \phi)^{-1}}_{-\mathbf{IG}} \cdot \nabla_\phi f_u(\theta, \phi) \tag{18}$$

© 2023 IBM Corporation

BA Algorithm



- ⇛ Double loop
- ✓ Convergence guarantees (for strongly convex LL, smooth UL)
- ✗ Needs explicit Hessian inverse
- ✗ Needs (approx) LL solution in each UL iteration

© 2023 IBM Corporation

Stochastic variant: Bi-level Stochastic Approximation (BSA)

Algorithm 2 Bilevel Approximation Algorithm [Ghadimi and Wang, 2018]

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.)  for current` θ^k

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

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

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

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

  `// UL descent step with` **IG**

  $\theta^{k+1} \leftarrow \theta^k - \alpha^k \cdot \left[\nabla_\theta f_u(\theta, \phi) - \nabla^2_{\theta\phi} f_l(\theta, \phi) \cdot \nabla^2_\phi f_l(\theta, \phi)^{-1} \cdot \nabla_\phi f_u(\theta, \phi) \right] \Big|_{\substack{\theta=\theta^k, \\ \phi=\phi^{k+1}}}$

return θ^{K+1}, ϕ^{K+1}

$\Rightarrow$ $\nabla_\phi f_l$, $\nabla_\theta f_u$, $\nabla^2_{\theta\phi} f_l$, $\nabla_\phi f_u$ – replace with stochastic estimates

$\Rightarrow$ $\nabla^2_\phi f_l(\theta, \phi)^{-1}$ – replace Hessian inverse with stochastic Neumann approximation**

© 2023 IBM Corporation

Approximate Implicit Differentiation Bi-level Opt (AID-BiO)

Algorithm 3 Approx Implicit Differentiation Bi-level Optimization [Ji 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.)  for current` θ^k

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

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

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

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

  `// UL descent step with IG`

  $\theta^{k+1} \leftarrow \theta^k - \alpha^k \cdot \left[\nabla_\theta f_u(\theta, \phi) - \nabla^2_{\theta\phi} f_l(\theta, \phi) \cdot \nabla^2_\phi f_l(\theta, \phi)^{-1} \cdot \nabla_\phi f_u(\theta, \phi) \right] \Big|_{\substack{\theta=\theta^k, \\ \phi=\phi^{k+1}}}$

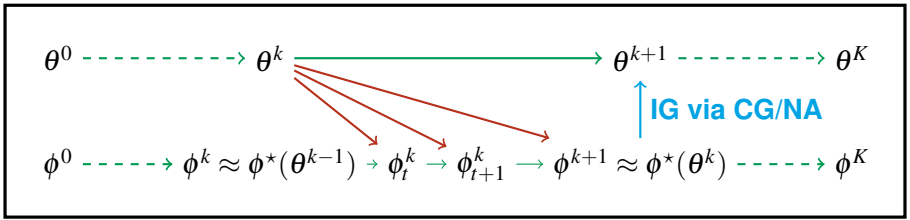
return θ^{K+1}, ϕ^{K+1}

$\Rrightarrow$ $\nabla^2_\phi f_l^{-1} \cdot \nabla_\phi f_u$ – approx inverse Hessian-gradient product with Conjugate Gradient**

© 2023 IBM Corporation

AID-BiO Algorithm

BSA/AID-BiO



⇛ Double loop

✓ Convergence guarantees (for strongly convex LL, smooth UL)

✗ Needs (approx) LL solution in each UL iteration

© 2023 IBM Corporation

Two-Timescale Stochastic Approximation (TTSA)

Algorithm 4 Two-Timescale Stoc Approx [Hong et al., 2020, 2023]

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**

// Single LL descent step

$$\phi^{k+1} \leftarrow \phi^k - \beta^k \left. \nabla_\phi f_l(\theta, \phi) \right|_{\theta=\theta^k, \phi=\phi^k}$$

// UL descent step with IG

$$\theta^{k+1} \leftarrow \mathscr{P}_\Theta\Big(\theta^k - \alpha^k \big[\nabla_\theta f_u(\theta,\phi) - \nabla^2_{\theta\phi} f_l(\theta,\phi) \cdot \nabla^2_\phi f_l(\theta,\phi)^{-1} \cdot \nabla_\phi f_u(\theta,\phi) \big] \big|_{\substack{\theta=\theta^k, \\ \phi=\phi^{k+1}}} \Big)$$

return θ^{K+1}, ϕ^{K+1}

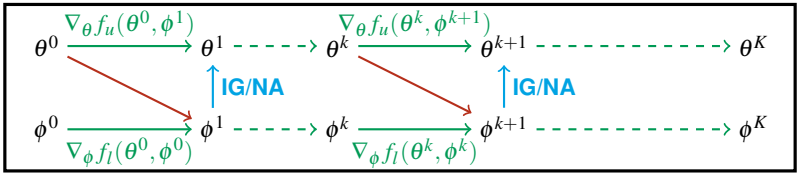
$\Rrightarrow$ $\nabla_\phi f_l$, $\nabla_\theta f_u$, $\nabla^2_{\theta\phi} f_l$, $\nabla_\phi f_u$ – stochastic estimates

$\Rrightarrow$ $\nabla^2_\phi f_l(\theta,\phi)^{-1}$ – Hessian inverse with stochastic Neumann approximation**

© 2023 IBM Corporation

TTSA Algorithm

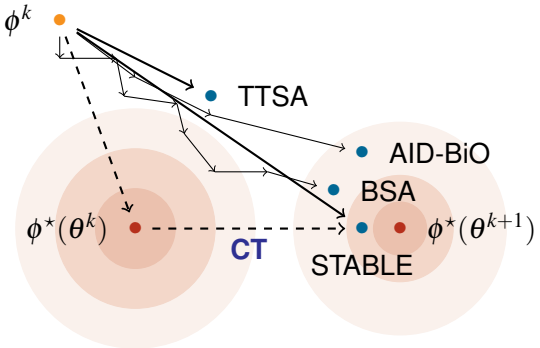
TTSA



- ✓ Single loop – no need to solve LL at each UL iteration
- ✓ Convergence guarantees if $\alpha^k \ll \beta^k$ and $\alpha^k/\beta^k \to 0$ as $k \to \infty$
  - ⇛ LL optimizes faster than UL, thus, *two-timescale*
- ✓ Handles UL csts with projected SGD in the UL update: $\theta^{k+1} \leftarrow \mathscr{P}_\Theta\left(\theta^k - \alpha^k \cdot h_u^k\right)$
- ✗ Needs to use small α^k, which slows the UL convergence

© 2023 IBM Corporation

Understanding LL updates [Chen et al., 2022]



Main Idea

Leverage an estimate of $\phi^\star(\theta^{k+1}) - \phi^\star(\theta^k)$ – the correction term (**CT**)

© 2023 IBM Corporation

Single-Timescale Stochastic Bi-level Optimization (STABLE)

Algorithm 5 Single-Timescale Stoc Bi-level Optimization [Chen et al., 2022]

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**

`// UL descent step with IG`

$$\theta^{k+1} \leftarrow \mathscr{P}_\Theta\left(\theta^k - \alpha^k \cdot \left[\nabla_\theta f_u(\theta,\phi) - \nabla^2_{\theta\phi} f_l(\theta,\phi) \cdot \nabla^2_\phi f_l(\theta,\phi)^{-1} \cdot \nabla_\phi f_u(\theta,\phi)\right]\Big|_{\substack{\theta=\theta^k,\\ \phi=\phi^k}}\right)$$

`// Single LL descent step with correction term`

$$\phi^{k+1} \leftarrow \phi^k - \beta^k \nabla_\phi f_l(\theta,\phi) \overbrace{-\nabla^2_\phi f_l(\theta,\phi)^{-1} \cdot \nabla^2_{\theta\phi} f_l(\theta,\phi)^\top\Big|_{\substack{\theta=\theta^k,\\ \phi=\phi^k}} \cdot (\theta^{k+1} - \theta^k)}$$

return θ^{K+1}, ϕ^{K+1}

© 2023 IBM Corporation

Single-Timescale Stochastic Bi-level Optimization (STABLE)

Algorithm 6 Single-Timescale Stoc Bi-level Optimization [Chen et al., 2022]

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**

`// UL descent step with IG`

$$\theta^{k+1} \leftarrow \mathscr{P}_\Theta\left(\theta^k - \alpha^k \cdot \left[\nabla_\theta f_u(\theta,\phi) - \nabla^2_{\theta\phi} f_l(\theta,\phi) \cdot \nabla^2_{\phi} f_l(\theta,\phi)^{-1} \cdot \nabla_\phi f_u(\theta,\phi)\right]\Big|_{\substack{\theta=\theta^k,\\ \phi=\phi^k}}\right)$$

`// Single LL descent step with correction term`

$$\phi^{k+1} \leftarrow \phi^k - \beta^k \nabla_\phi f_l(\theta,\phi) \overbrace{- \nabla^2_{\phi} f_l(\theta,\phi)^{-1} \cdot \nabla^2_{\theta\phi} f_l(\theta,\phi)^\top\Big|_{\substack{\theta=\theta^k,\\ \phi=\phi^k}} \cdot (\theta^{k+1} - \theta^k)}$$

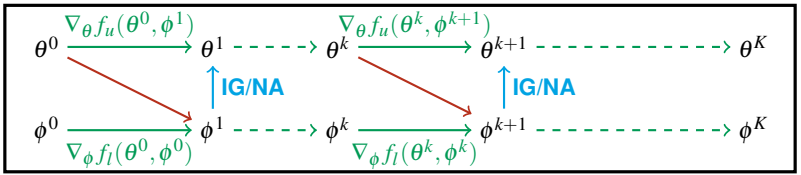
return θ^{K+1}, ϕ^{K+1}

- $\Rightarrow$ $\nabla_\phi f_l$, $\nabla_\theta f_u$, $\nabla_\phi f_u$ – stochastic estimates
- $\Rightarrow$ $\nabla^2_{\phi,\phi}{}^{-1}$ – Hessian inverse with stoc Neumann approximation** & variance reduction
- $\Rightarrow$ $\nabla^2_{\theta\phi} f_l$ – variance reduced stochastic estimate

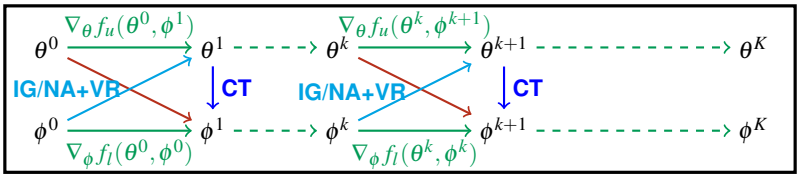
© 2023 IBM Corporation

TTSA vs STABLE

TTSA



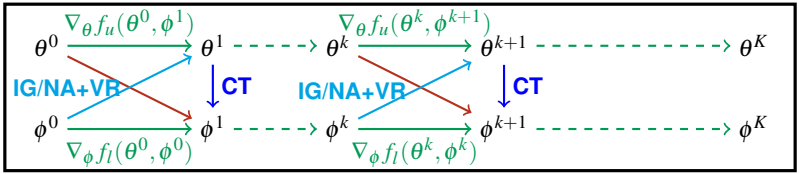
STABLE



© 2023 IBM Corporation

TTSA Algorithm

STABLE



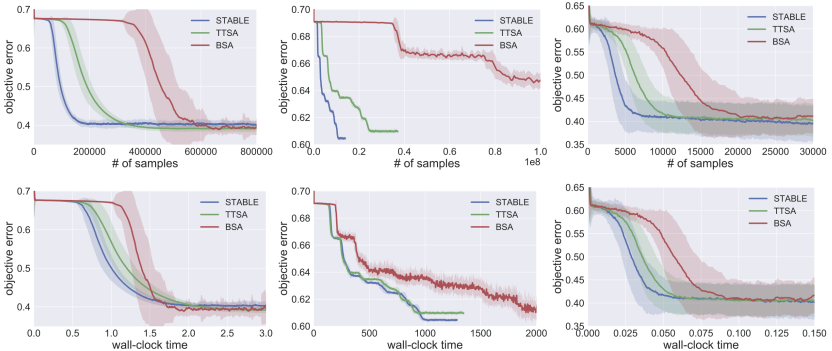
- ✓ Single loop
- ✓ Handles UL constraints via projected gradient descent
- ✓ Single timescale – $\alpha^k, \beta^k \sim O(1/\sqrt{K})$ for convergence
- ✗ More expensive LL update

© 2023 IBM Corporation

Empirical comparison [Chen et al., 2022]

Bi-level hyperparameter optimization

$$\min_{\theta \in \mathbb{R}^{d_u}} \sum_{i \in [n]} \ell(y_i, M_{\phi^\star(\theta)}(x_i)) \quad \texttt{s.t.} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} \sum_{j \in [m]} \ell(\mathrm{y}_i, M_\phi(\mathrm{x}_i)) + \theta^\top (\phi \odot \phi)$$



© 2023 IBM Corporation

Other Modifications and Enhancements (incomplete)

Techniques

- ⇛ Leverage momentum acceleration for UL and LL descent steps for faster convergence guarantees [Khanduri et al., 2021]
- ⇛ Avoid IG approximation completely by Hessian-free approaches [Sow et al., 2022a]
- ⇛ Variance Reduction

© 2023 IBM Corporation

Vanilla gradient descent

For a function $f(\theta)$:

$$\theta^{k+1} \leftarrow \theta^k - \alpha^k \cdot \nabla_\theta f(\theta^k), \quad \text{where} \quad \nabla_\theta f(\theta^k) = \nabla_\theta f(\theta)|_{\theta=\theta^k} \tag{19}$$

$\nabla_\theta f(\theta^k)$ can also be a stochastic estimate of the gradient of $f(\theta)$ at θ^k.

(S)GD with momentum

With momentum parameters $\eta^k \in (0,1), \forall k \in [K]$

$$\theta^{k+1} \leftarrow \theta^k - \alpha^k \cdot h^k, \quad \text{where} \quad h^k \leftarrow \eta^k h^{k-1} + (1-\eta^k)\nabla_\theta f(\theta^k). \tag{20}$$

© 2023 IBM Corporation

Enhancement: Bi-level Momentum

Bi-level momentum [Khanduri et al., 2021]

⇛ LL update with momentum with $\eta_l^k \in (0,1)$

$$\tilde{h}_l^k \leftarrow (1-\eta_l^k)\nabla_\phi f_l(\theta^k,\phi^k) + \eta_l^k\left(\tilde{h}_l^{k-1} + \nabla_\phi f_l(\theta^k,\phi^k) - \nabla_\phi f_l(\theta^{k-1},\phi^{k-1})\right) \tag{21}$$

⇛ UL update with momentum with $\eta_u^k \in (0,1)$

$$\begin{aligned}
&\tilde{h}_u^k \leftarrow (1-\eta_u^k)\bar{\nabla} f_u(\theta^k,\phi^k) + \eta_u^k\left(\tilde{h}_u^{k-1} + \bar{\nabla} f_u(\theta^k,\phi^k) - \bar{\nabla} f_u(\theta^{k-1},\phi^{k-1})\right)\\
&\bar{\nabla} f_u(\theta^k,\phi^k) \approx \left[\nabla_\theta f_u(\theta,\phi) - \nabla^2_{\theta\phi} f_l(\theta,\phi)\cdot\nabla^2_\phi f_l(\theta,\phi)^{-1}\cdot\nabla_\phi f_u(\theta,\phi)\right]\Big|_{\substack{\theta=\theta^k,\\ \phi=\phi^k}}
\end{aligned} \tag{22}$$

© 2023 IBM Corporation

Details: Conjugate Gradient

Obtain inverse-Hessian vector product $H^{-1}v$ by solving the following quadratic program:

$$\min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top H x - v^\top x, \qquad H - d \times d\text{-Hessian matrix}, v - \text{ a vector.} \tag{23}$$

Solve via Conjugate Gradient method [Nazareth, 2009].

© 2023 IBM Corporation

Details: Conjugate Gradient

Algorithm 7 Conjugate Gradient Algorithm

Input: Positive definite symmetric $H \in \mathbb{R}^{d \times d}$, vector $v \in \mathbb{R}^d$

Input: Initial $x_0 \in \mathbb{R}^d$, precision $\varepsilon > 0$, max iters n

$d_0 = r_0 \leftarrow v - Hx_0$

for $i \leftarrow 0, 1, \ldots, n$ **do**

- $\alpha_i \leftarrow (d_i^\top r_i)/(d_i^\top Hd_i)$ `// Hessian-vector product**`
- $x_{i+1} \leftarrow x_i + \alpha_i d_i$
- $r_{i+1} \leftarrow r_i - \alpha_i Hd_i$
- $\beta_{i+1} \leftarrow (r_{i+1}^\top r_{i+1})/(r_i^\top r_i)$
- $d_{i+1} \leftarrow r_{i+1} + \beta_{i+1} d_i$
- **if** $r_{i+1}^\top r_{i+1} \leq \varepsilon$ **then**
  - **return** x_i

return x_{n+1}

© 2023 IBM Corporation

Details: Neumann Series Approximation

Neumann series expansion:

$$H^{-1} = \sum_{n=0}^{\infty} [I - H]^n \tag{24}$$

Stochastic approximation for IG:

- ⇛ Sample $p \in [0, t_{\max}]$
- ⇛ Compute $C_1 \prod_{i=0}^{p} [I - C_2 \nabla^2_{\phi} f_l(\theta^k, \phi^k)]$ using p different stochastic estimates of $\nabla^2_{\phi} f_l$
- ⇛ For appropriately set scalars C_1, C_2, this provides a **biased but sufficiently accurate** (for convergence) estimate of the IG

© 2023 IBM Corporation

Non-IG based Solutions

Solutions that do not make use of the Implicit Gradient

⇛ Optimal value function based techniques

⇛ Gradient unrolling based techniques

© 2023 IBM Corporation

Optimal Value Function

$$\min_{\theta\in\Theta,\phi\in\Phi} f_u(\theta,\phi) \quad \texttt{subject to} \quad f_l(\theta,\phi) \leq v(\theta) = \min_{\phi\in\Phi} f_l(\theta,\phi) \tag{25}$$

- $v : \mathbb{R}^{d_u} \to \mathbb{R}$ is called the Optimal Value Function
- $\Rightarrow$ No closed form available
- $\Rightarrow$ Maybe non-convex, non-differentiable
- $\Rightarrow$ May not be strictly feasible and regularity conditions may not hold

© 2023 IBM Corporation

Optimal Value Function

Smooth upper bound on the optimal value function

$$\tilde{v}_a(\theta) = \min_{\phi \in \Phi} f_l(\theta, \phi) + \frac{a_1}{2}\|\phi\|_2^2 + a_2, a = \{a_1, a_2\}, a_1, a_2 > 0. \tag{26}$$

⇛ Smooth

⇛ strictly feasible

© 2023 IBM Corporation

Optimal Value Function

Now solve

$$\min_{\theta\in\Theta,\phi\in\Phi} f_u(\theta,\phi) \quad \texttt{subject to} \quad f_l(\theta,\phi) \leq \tilde{v}_a(\theta) \tag{27}$$

Penalty based single-level optimization (for a large enough $\rho > 0$)

$$\min_{\theta\in\Theta,\phi\in\Phi} f_u(\theta,\phi) + \rho \cdot \max\{0, f_l(\theta,\phi) - \tilde{v}_a(\theta)\} \tag{28}$$

Sow et al. [2022b] solves the following using primal-dual methods

$$\max_{\rho\geq 0} \min_{\theta\in\Theta,\phi\in\Phi} f_u(\theta,\phi) + \rho \cdot (f_l(\theta,\phi) - \tilde{v}_a(\theta)) \tag{29}$$

© 2023 IBM Corporation

Gradient Unrolling

- ⇛ Double-loop setting
- ⇛ Access to the LL optimizer and the iterates $\phi^k = \varphi^0 \to \varphi^1 \to \cdots \varphi^T \approx \phi^\star(\theta^k)$
- ⇛ Can we compute $d\phi^\star(\theta^k)/d\theta$ via *chain-rule*?

Consider $T = 1$, with $\varphi^1 \leftarrow \varphi^0 - \beta \nabla_\phi f_l(\theta^k, \varphi^0)$

$$\frac{d\varphi^1}{d\theta} = \frac{\partial \varphi^1}{\partial \varphi^0} \cdot \frac{d\varphi^0}{d\theta} + \frac{\partial \varphi^1}{\partial \theta} = -\beta \nabla^2_{\theta\phi} f_l(\theta^k, \varphi^0), \tag{30}$$

assuming $d\varphi^0/d\theta = 0$

© 2023 IBM Corporation

Gradient Unrolling

Consider $T = 2$, with $\varphi^2 \leftarrow \varphi^1 - \beta \nabla_\phi f_l(\theta^k, \varphi^1)$

$$\begin{aligned} \frac{d\varphi^2}{d\theta} &= \frac{\partial \varphi^2}{\partial \varphi^1} \cdot \frac{d\varphi^1}{d\theta} + \frac{\partial \varphi^2}{\partial \theta} \\ &= -\beta \left[\left(I - \nabla^2_\phi f_l(\theta^k, \varphi^1) \right) \cdot \nabla^2_{\theta\phi} f_l(\theta^k, \varphi^0) + \nabla^2_{\theta\phi} f_l(\theta^k, \varphi^1) \right], \end{aligned} \tag{31}$$

For any general $t > 1$

$$\underbrace{\frac{d\varphi^t}{d\theta}}_{Z_t \in \mathbb{R}^{d_l \times d_u}} = \underbrace{\frac{\partial \varphi^t}{\partial \varphi^{t-1}}}_{A_t \in \mathbb{R}^{d_l \times d_l}} \cdot \underbrace{\frac{d\varphi^{t-1}}{d\theta}}_{Z_{t-1}} + \underbrace{\frac{\partial \varphi^t}{\partial \theta}}_{B_t \in \mathbb{R}^{d_l \times d_u}} \tag{32}$$

Using the recursion $Z_t = A_t Z_{t-1} + B_t$, we can compute Z_T by "unrolling" the gradient. This is known as the *forward hypergradient* [Franceschi et al., 2017].

© 2023 IBM Corporation

Gradient Unrolling

Things to note:

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

$$A_t = \left(I - \beta \nabla^2_\phi f_l(\theta^k, \varphi^{t-1})\right), \quad B_t = -\beta \nabla^2_{\theta\phi} f_l(\theta^k, \varphi^{t-1})$$

- Forward mode computation of $d\varphi^T/d\theta$:
  - $Z_0 = 0$
  - for $t = 1 \to T$
    - Compute A_t, B_t, update $Z_t \leftarrow A_t Z_{t-1} + B_t$
  - Return Z_T
- Backward mode computation useful in computing $(d\varphi^T/d\theta)^\top v$ for some $v \in \mathbb{R}^{d_l}$
  - $\alpha_T \leftarrow v$, $g \leftarrow 0 \in \mathbb{R}^{d_u}$
  - for $t = (T-1) \to 1$
    - Compute A_{t+1}, B_{t+1}, update $g \leftarrow g + B_{t+1}^\top \alpha_{t+1}$, $\alpha_t \leftarrow A_{t+1}^\top \alpha_{t+1}$
  - Return g

© 2023 IBM Corporation

Gradient Unrolling

Compared to other methods:

- ✓ No Hessian-inverse required
- ✗ Relies on choice of LL optimizer and optimization path
- ✗ Memory and computation overhead increases with T
  - ⇛ Truncated unrolling – ignore earlier LL steps
  - ⇛ Select special LL optimizer**

Forward vs backward:

- ✓ Forward does not require maintaining the iterates
- ✗ Forward requires matrix-matrix multiplications
- ✗ Backward requires whole sequence of iterates
- ✓ Backward can take advantage of efficient Hessian-vector products**

© 2023 IBM Corporation

Outline

 © 2023 IBM Corporation

Setting and Goal: Unconstrained

Specific bi-level problem: No constraints + unique LL solution

$$\min_{\theta \in \mathbb{R}^{d_u}} F(\theta) = f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{33}$$

A point $\bar{\theta} \in \mathbb{R}^{d_u}$ is an ε**-stationary point** if

$$\begin{aligned} \| \nabla_\theta F(\theta)|_{\theta=\bar{\theta}} \|_2^2 &\leq \varepsilon \qquad \texttt{(deterministic)} \\ \mathbb{E}\left[\| \nabla_\theta F(\theta)|_{\theta=\bar{\theta}} \|_2^2 \right] &\leq \varepsilon \qquad \texttt{(stochastic)} \end{aligned} \tag{34}$$

Total number of (stochastic) gradient estimates $S(f_u, \varepsilon), S(f_l, \varepsilon)$ of f_u, f_l resp evaluated to reach a ε-stationary solution is called the **sample complexity**.

Example

If $\mathbb{E}\|\nabla_\theta F(\bar{\theta})\|_2^2 \leq O(K^{-r})$ for K iterations and $r \in (0,1)$, then $S(f_u, \varepsilon) \sim O(1/\varepsilon^{1/r})$

© 2023 IBM Corporation

Setting and Goal: UL constraints

Specific bi-level problem: UL constraints + unique LL solution

$$\min_{\theta \in \Theta \subset \mathbb{R}^{d_u}} F(\theta) = f_u(\theta, \phi^\star(\theta)) \quad \texttt{subject to} \quad \phi^\star(\theta) = \arg\min_{\phi \in \mathbb{R}^{d_l}} f_l(\theta, \phi) \tag{35}$$

In unconstrained case, we need $\nabla_\theta F(\theta)|_{\theta=\bar{\theta}}$ to be small.
In the constrained case, the condition is more general:

$$\left\langle \nabla_\theta F(\theta)|_{\theta=\bar{\theta}}, \theta - \bar{\theta} \right\rangle \geq 0, \forall \theta \in \Theta. \tag{36}$$

Essentially, moving from the solution increases the objective value.

© 2023 IBM Corporation

Setting and Goal: UL constraints

Moreau Envelop

For a fixed $\rho > 0$, the Moreau envelop and the proximal map are defined as

$$M_{1/\rho}(\vartheta) = \min_{\theta \in \Theta} \{F(\theta) + (\rho/2)\|\theta - \vartheta\|^2\}, \quad \hat{\theta}(\vartheta) = \arg\min_{\theta \in \Theta} \{F(\theta) + (\rho/2)\|\theta - \vartheta\|^2\} \tag{37}$$

For an $\varepsilon > 0$, a point $\bar{\theta} \in \mathbb{R}^{d_u}$ is an **ε-nearly stationary point** if $\bar{\theta}$ is an approximate fixed point of $\{\hat{\theta} - I\}(\cdot)$

$$\mathbb{E}\left[\left\|\hat{\theta}(\bar{\theta}) - \bar{\theta}\right\|_2^2\right] \leq \varepsilon/\rho^2 \qquad \texttt{(stochastic)} \tag{38}$$

© 2023 IBM Corporation

Two-Timescale Stochastic Approximation (TTSA)

Algorithm 8 Two-Timescale Stoc Approx [Hong et al., 2020, 2023]

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**

// Single LL descent step

$$\phi^{k+1} \leftarrow \phi^k - \beta^k \nabla_\phi f_l(\theta, \phi)\Big|_{\theta=\theta^k, \phi=\phi^k}$$

// UL descent step with IG

$$\theta^{k+1} \leftarrow \mathscr{P}_\Theta(\theta^k - \alpha^k [\nabla_\theta f_u(\theta, \phi) - \underbrace{\nabla^2_{\theta\phi} f_l(\theta, \phi) \cdot \nabla^2_\phi f_l(\theta, \phi)^{-1} \cdot \nabla_\phi f_u(\theta, \phi)}_{=\bar{\nabla} f_u(\theta, \phi)}]|_{\substack{\theta=\theta^k, \\ \phi=\phi^{k+1}}})$$

return θ^{K+1}, ϕ^{K+1}

$\Rrightarrow$ $\nabla_\phi f_l$, $\nabla_\theta f_u$, $\nabla^2_{\theta\phi} f_l$, $\nabla_\phi f_u$ – stochastic estimates

$\Rrightarrow$ $\nabla^2_\phi f_l(\theta, \phi)^{-1}$ – Hessian inverse with stochastic Neumann approximation**

© 2023 IBM Corporation

TTSA Analysis: Assumptions

The UL objective $f_u(\theta, \phi)$ and $F(\theta) = f_u(\theta, \phi^\star(\theta))$ satisfy the following:

- $\Rightarrow$ For any $\theta \in \mathbb{R}^{d_u}$, $\nabla_\theta f_u(\theta, \cdot)$ and $\nabla_\phi f_u(\theta, \cdot)$ are Lipschitz continuous w.r.t. $\phi \in \mathbb{R}^{d_l}$
- $\Rightarrow$ For any $\phi \in \mathbb{R}^{d_l}$, $\nabla_\phi f_u(\cdot, \phi)$ is Lipschitz continuous w.r.t. $\theta \in \Theta$
- $\Rightarrow$ For any $\theta \in \Theta, \phi \in \mathbb{R}^{d_l}$, $\|\nabla_\phi f_u(\theta, \phi)\|$ is bounded

© 2023 IBM Corporation

TTSA Analysis: Assumptions

The LL objective $f_l(\theta, \phi)$ satisfy the following

- $\Rightarrow$ For any $\theta \in \Theta, \phi \in \mathbb{R}^{d_l}$, $f_l(\theta, \phi)$ is twice continuously differentiable in (θ, ϕ)
- $\Rightarrow$ For any $\theta \in \Theta$, $\nabla_\phi f_l(\theta, \cdot)$ is Lipschitz continuous w.r.t. $\phi \in \mathbb{R}^{d_l}$
- $\Rightarrow$ For any $\theta \in \Theta$, $f_l(\theta, \cdot)$ is strongly convex in ϕ
- $\Rightarrow$ For any $\theta \in \Theta$, $\nabla^2_{\theta\phi} f_l(\theta, \cdot), \nabla^2_\phi f_l(\theta, \cdot)$ is Lipschitz continuous w.r.t. $\phi \in \mathbb{R}^{d_l}$
- $\Rightarrow$ For any $\phi \in \mathbb{R}^{d_l}$, $\nabla^2_{\theta\phi} f_l(\cdot, \phi), \nabla^2_\phi f_l(\cdot, \phi)$ is Lipschitz continuous w.r.t. $\theta \in \Theta$
- $\Rightarrow$ For any $\theta \in \Theta, \phi \in \mathbb{R}^{d_l}$, $\|\nabla^2_{\theta\phi} f_l(\theta, \phi)\|$ is bounded

© 2023 IBM Corporation

TTSA Analysis: Assumptions

Algorithm 9 Two-Timescale Stoc Approx [Hong et al., 2020, 2023]

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**

// Single LL descent step

$$\phi^{k+1} \leftarrow \phi^k - \beta^k \left. \nabla_\phi f_l(\theta, \phi) \right|_{\theta=\theta^k, \phi=\phi^k}$$

// UL descent step with IG

$$\theta^{k+1} \leftarrow \mathscr{P}_\Theta \left(\theta^k - \alpha^k \left[\nabla_\theta f_u(\theta, \phi) - \nabla^2_{\theta\phi} f_l(\theta, \phi) \cdot \nabla^2_\phi f_l(\theta, \phi)^{-1} \cdot \nabla_\phi f_u(\theta, \phi) \right] \Big|_{\substack{\theta=\theta^k, \\ \phi=\phi^{k+1}}} \right)$$

return θ^{K+1}, ϕ^{K+1}

The stochastic estimates satisfy the following:

- ⇛ The stochastic estimates of $\nabla_\phi f_l$ are unbiased with bounded variance
- ⇛ The stochastic estimates of $\bar{\nabla} f_u(\theta, \phi)$ are biased with bounded variance, and the bias nonincreasing with the number of iterations k

© 2023 IBM Corporation

TTSA Analysis: First set of Lemmas

$\Rightarrow \|\bar{\nabla}_\theta f_u(\theta,\phi) - \nabla_\theta F(\theta)\| \leq L\|\phi^\star(\theta) - \phi\|$

$\Rightarrow \|\phi^\star(\theta_1) - \phi^\star(\theta_2)\| \leq L_\phi\|\theta_1 - \theta_2\|$

$\Rightarrow \|\nabla_\theta F(\theta_1) - \nabla_\theta F(\theta_2)\| \leq L_u\|\theta_1 - \theta_2\|$

This allows us to bound the difference between the inaccurate and the exact implicit gradient.

© 2023 IBM Corporation

TTSA Analysis: Tracking errors

The following are terms we need to track in each iteration

- ⇛ UL optimality gap: $\Delta_\theta^k = \mathbb{E}[\|\theta^k - \theta^\star\|]$
- ⇛ LL optimality gap: $\Delta_\phi^k = \mathbb{E}[\|\phi^k - \phi^\star(\theta^{k-1})\|]$
  - ⇛ Tracks how inaccurate the LL solution is for current θ^{k-1}
  - ⇛ Helps track the error introduced by the IG computed with ϕ^k instead of $\phi^\star(\theta^{k-1})$
- ⇛ UL constraint proximal gap $\tilde{\Delta}_\theta^k = \mathbb{E}[\|\hat{\theta}(\theta^k) - \theta^k\|]$
  - ⇛ $\hat{\theta}(\theta) = \arg\min_{\vartheta \in \Theta}\{F(\vartheta) + (\rho/2)\|\theta - \vartheta\|^2\}$
  - ⇛ Tracks distance from fixed point of $\{\hat{\theta} - I\}(\cdot)$

© 2023 IBM Corporation

TTSA Analysis: Strongly convex $F(\theta)$

Strong convexity of UL $F(\theta) = f_u(\theta, \phi^\star(\theta))$

$$F(\theta) \geq F(\theta') + \langle \nabla_\theta F(\theta), \theta - \theta' \rangle + c_F \|\theta - \theta'\|_2^2, \forall \theta, \theta' \in \Theta, c_F > 0 \tag{39}$$

Coupling equations:

$$\Delta_\phi^{k+1} \leq \prod_{j=0}^{k} (1 - c_1 \beta^j) \Delta_\phi^0 + c_2 \beta^k \tag{40}$$

$$\Delta_\theta^{k+1} \leq \prod_{j=0}^{k} (1 - d_1 \alpha^k) \Delta_\theta^0 + d_2 \alpha^k + d_3 \sum_{j=0}^{k} \alpha^j \prod_{i=j+1}^{k} (1 - d_1 \alpha^k) \Delta_\phi^{k+1} \tag{41}$$

© 2023 IBM Corporation

Coupling equations:

$$\Delta_\phi^{k+1} \leq \prod_{j=0}^{k}(1 - c_1\beta^j)\Delta_\phi^0 + c_2\beta^k \tag{42}$$

$$\Delta_\theta^{k+1} \leq \prod_{j=0}^{k}(1 - d_1\alpha^k)\Delta_\theta^0 + d_2\alpha^k + d_3\sum_{j=0}^{k}\alpha^j\prod_{i=j+1}^{k}(1 - d_1\alpha^k)\Delta_\phi^{k+1} \tag{43}$$

Considering the dominating terms above, with $\beta^j \sim \tilde{O}((\alpha^j)^{2/3})$

$$\Delta_\phi^{k+1} \sim \tilde{O}(\beta^k) \sim \tilde{O}((\alpha^k)^{2/3}) \tag{44}$$

$$\sum_{j=0}^{k}\alpha^j\prod_{i=j+1}^{k}(1 - d_1\alpha^k)\Delta_\phi^{k+1} = \sum_{j=0}^{k}\tilde{O}((\alpha^j)^{5/3})\prod_{i=j+1}^{k}(1 - d_1\alpha^k) \sim \tilde{O}((\alpha^k)^{2/3}) \tag{45}$$

© 2023 IBM Corporation

TTSA Analysis: Strongly convex $F(\theta)$

Considering the dominating terms, with $\beta^j \sim \tilde{O}((\alpha^j)^{2/3})$

$$\Delta_\phi^{k+1} \sim \tilde{O}((\alpha^k)^{2/3}) \tag{46}$$

$$\Delta_\theta^{k+1} \sim \tilde{O}((\alpha^k)^{2/3}) \tag{47}$$

Setting $\alpha^k \sim O(1/k)$ and $\beta^k \sim O(1/k^{2/3})$, we can establish that the optimality gap

$$\Delta_\phi^{k+1} \sim \tilde{O}(k^{-2/3}) \qquad \Delta_\theta^{k+1} \sim \tilde{O}(k^{-2/3}) \tag{48}$$

With K iterations of TTSA, we converge to a $\tilde{O}(K^{-2/3})$-stationary point, with a sample complexity of $\tilde{O}(1/\varepsilon^{3/2})$ for both the UL and LL objectives.

© 2023 IBM Corporation

TTSA Analysis: Weakly convex $F(\theta)$

Weak convexity of UL $F(\theta) = f_u(\theta, \phi^\star(\theta))$

$$F(\theta) \geq F(\theta') + \langle \nabla_\theta F(\theta), \theta - \theta' \rangle + c_F \|\theta - \theta'\|_2^2, \forall \theta, \theta' \in \Theta, \tag{49}$$

where the modulus of convexity, c_F, might be negative.

© 2023 IBM Corporation

TTSA Analysis: Weakly convex $F(\theta)$

Using a different set of recursive coupling equations, and a more involved analysis, we can establish the following with $\alpha^k \sim O(K^{-3/5})$ and $\beta^k \sim O(K^{-2/5})$ for all $k \in [K]$.

$$\frac{1}{K}\sum_{k=1}^{K} \Delta_\phi^k \leq \tilde{O}(K^{-2/5}) \tag{50}$$

$$\frac{1}{K}\sum_{k=1}^{K} \|\theta^k - \theta^{k-1}\|^2 \leq \tilde{O}(K^{-6/5}) \tag{51}$$

$$\frac{1}{K}\sum_{k=0}^{K} \tilde{\Delta}_\theta^k \leq \tilde{O}(K^{-2/5}), \tag{52}$$

These can be used to show that TTSA with K iterations converges to a $\tilde{O}(K^{-2/5})$-nearly stationary point, giving us a sample complexity of $\tilde{O}(1/\varepsilon^{5/2})$.

© 2023 IBM Corporation

Other results and comparison

Algorithm	Loop	UL cst	$S(f_u, \varepsilon)$	$S(f_l, \varepsilon)$
BSA [Ghadimi and Wang, 2018]	Double	✗	$O(1/\varepsilon^2)$	$O(1/\varepsilon^3)$
TTSA [Hong et al., 2023]	Single	✓	$O(1/\varepsilon^{5/2})$	$O(1/\varepsilon^{5/2})$
STABLE [Chen et al., 2022]	Single	✓	$O(1/\varepsilon^2)$	$O(1/\varepsilon^2)$

© 2023 IBM Corporation

References

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eYHoC5FX.

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.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

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.

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.

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in neural information processing systems*, 34:30271–30283, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/fe2b421b8b5f0e7c355ace66a9fe0206-Supplemental.pdf.

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, 2022a. URL https://openreview.net/pdf?id=suHUJr7dV5n.

John L Nazareth. Conjugate gradient method. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):348–353, 2009.

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*, 2022b. URL https://arxiv.org/pdf/2203.01123.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.

 © 2023 IBM Corporation