A fully first-order method for stochastic bilevel optimization

Dohyun Kwon

Department of Mathematics, University of Seoul / Center for AI and Natural Sciences, KIAS

Nov 8, 2024

This talk is based on joint work with Jeongyeol Kwon, Hanbaek Lyu, Stephen Wright, and Robert Nowak (UW-Madison, USA).

Table of Contents

Table of Contents

Bilevel optimization

- Bilevel optimization (Colson et al., 2007) is a fundamental optimization problem that abstracts various applications characterized by two-level hierarchical structures.
- Consider the minimization problem:

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

  where $f, g : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ are continuously-differentiable functions.
- There are various applications, including adversarial networks (Goodfellow et al., 2020; Gidel et al., 2018), game theory (Stackelberg et al., 1952), hyper-parameter optimization (Franceschi et al., 2018; Bao et al., 2021), model selection (Kunapuli et al., 2008; Giovannelli et al., 2021) and reinforcement learning (Konda & Tsitsiklis, 1999; Sutton & Barto, 2018).

Bilevel optimization

- Bilevel optimization (Colson et al., 2007) is a fundamental optimization problem that abstracts various applications characterized by two-level hierarchical structures.
- Consider the minimization problem:

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

where $f, g : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ are continuously-differentiable functions.

- There are various applications, including adversarial networks (Goodfellow et al., 2020; Gidel et al., 2018), game theory (Stackelberg et al., 1952), hyper-parameter optimization (Franceschi et al., 2018; Bao et al., 2021), model selection (Kunapuli et al., 2008; Giovannelli et al., 2021) and reinforcement learning (Konda & Tsitsiklis, 1999; Sutton & Barto, 2018).

Bilevel optimization

- Bilevel optimization (Colson et al., 2007) is a fundamental optimization problem that abstracts various applications characterized by two-level hierarchical structures.
- Consider the minimization problem:

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

  where $f, g : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ are continuously-differentiable functions.
- There are various applications, including adversarial networks (Goodfellow et al., 2020; Gidel et al., 2018), game theory (Stackelberg et al., 1952), hyper-parameter optimization (Franceschi et al., 2018; Bao et al., 2021), model selection (Kunapuli et al., 2008; Giovannelli et al., 2021) and reinforcement learning (Konda & Tsitsiklis, 1999; Sutton & Barto, 2018).

Bilevel optimization

$$\min_{x \in \mathbb{R}^{d_x}} \quad F(x) := f(x, y^*(x))$$
$$\text{s.t.} \quad y^*(x) \in \arg\min_{y \in \mathbb{R}^{d_y}} g(x, y). \qquad \textbf{(P)}$$

- The *hyperobjective* $F(x)$ depends on x both directly and indirectly via $y^*(x)$.
- $y^*(x)$ is a solution for the lower-level problem of minimizing another function g.
- Typically, we assume that the lower-level problem is strongly convex: $g(\bar{x}, y)$ is strongly convex in y for all $\bar{x} \in \mathbb{R}^{d_x}$.

Bilevel optimization

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

- The *hyperobjective* $F(x)$ depends on x both directly and indirectly via $y^*(x)$.
- $y^*(x)$ is a solution for the lower-level problem of minimizing another function g.
- Typically, we assume that the lower-level problem is strongly convex: $g(\bar{x}, y)$ is strongly convex in y for all $\bar{x} \in \mathbb{R}^{d_x}$.

Problem

$$\min_{x \in \mathbb{R}^{d_x}} \quad F(x) := f(x, y^*(x))$$
$$\text{s.t.} \quad y^*(x) \in \arg\min_{y \in \mathbb{R}^{d_y}} g(x, y). \qquad \textbf{(P)}$$

Problem

Find an ϵ-*stationary point*: a point x satisfying $\|\nabla F(x)\| \leq \epsilon$.

- The explicit expression of $\nabla F(x)$ can be derived from the implicit function theorem:

$$\nabla F(x) := \nabla_x f(x, y^*(x)) - \nabla^2_{xy} g(x, y^*(x)) \left(\nabla^2_{yy} g(x, y^*(x))\right)^{-1} \nabla_y f(x, y^*(x)).$$

- Prior approaches require an explicit extraction of second-order information from g with a major focus on estimating the Jacobian and inverse Hessian efficiently with stochastic noises.
- Algorithms are not applicable to nonconvex objectives g and are hard to extend to the constrained case.

Problem

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

Problem

Find an ϵ-*stationary point*: a point x satisfying $\|\nabla F(x)\| \leq \epsilon$.

- The explicit expression of $\nabla F(x)$ can be derived from the implicit function theorem:

$$\nabla F(x) := \nabla_x f(x, y^*(x)) - \nabla^2_{xy} g(x, y^*(x)) \left(\nabla^2_{yy} g(x, y^*(x))\right)^{-1} \nabla_y f(x, y^*(x)).$$

- Prior approaches require an explicit extraction of second-order information from g with a major focus on estimating the Jacobian and inverse Hessian efficiently with stochastic noises.
- Algorithms are not applicable to nonconvex objectives g and are hard to extend to the constrained case.

Problem

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

Problem

Find an ϵ-*stationary point*: a point x satisfying $\|\nabla F(x)\| \leq \epsilon$.

- The explicit expression of $\nabla F(x)$ can be derived from the implicit function theorem:

$$\nabla F(x) := \nabla_x f(x, y^*(x)) - \nabla^2_{xy} g(x, y^*(x)) \left(\nabla^2_{yy} g(x, y^*(x))\right)^{-1} \nabla_y f(x, y^*(x)).$$

- Prior approaches require an explicit extraction of second-order information from g with a major focus on estimating the Jacobian and inverse Hessian efficiently with stochastic noises.
- Algorithms are not applicable to nonconvex objectives g and are hard to extend to the constrained case.

Our goal

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

Goal

Develop a fully first-order approach for stochastic bilevel optimization. Find an ϵ-stationary solution of F using only first-order gradients of f and g.

- Some works only use first-order information, but these works either lack a complete finite-time analysis or are applicable only to deterministic functions.

Stochastic bilevel optimization

$$\min_{x \in \mathbb{R}^{d_x}} \quad F(x) := f(x, y^*(x))$$
$$\text{s.t.} \quad y^*(x) \in \arg\min_{y \in \mathbb{R}^{d_y}} g(x, y), \qquad \textbf{(P)}$$

- We consider the first-order algorithm class that accesses functions through *first-order oracles* that return estimators of first-order derivatives $\hat{\nabla} f(x, y; \zeta), \hat{\nabla} g(x, y; \xi)$ for a given query point (x, y).

We assume that

- The estimators are unbiased:

$$\mathbb{E}[\hat{\nabla} f(x, y; \zeta)] = \nabla f(x, y),$$
$$\mathbb{E}[\hat{\nabla} g(x, y; \xi)] = \nabla g(x, y),$$

- The variance of the estimators are bounded:

$$\mathbb{E}[\|\hat{\nabla} f(x, y; \zeta) - \mathbb{E}[\nabla f(x, y; \zeta)]\|^2] \leq \sigma_f^2,$$
$$\mathbb{E}[\|\hat{\nabla} g(x, y; \xi) - \mathbb{E}[\nabla g(x, y; \xi)]\|^2] \leq \sigma_g^2.$$

for constants $\sigma_f^2 > 0$ and $\sigma_g^2 > 0$.

Stochastic bilevel optimization

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

- We consider the first-order algorithm class that accesses functions through *first-order oracles* that return estimators of first-order derivatives $\hat{\nabla} f(x, y; \zeta), \hat{\nabla} g(x, y; \xi)$ for a given query point (x, y).

We assume that

- The estimators are unbiased:

$$\mathbb{E}[\hat{\nabla} f(x, y; \zeta)] = \nabla f(x, y),$$
$$\mathbb{E}[\hat{\nabla} g(x, y; \xi)] = \nabla g(x, y),$$

- The variance of the estimators are bounded:

$$\mathbb{E}[\|\hat{\nabla} f(x, y; \zeta) - \mathbb{E}[\nabla f(x, y; \zeta)]\|^2] \le \sigma_f^2,$$
$$\mathbb{E}[\|\hat{\nabla} g(x, y; \xi) - \mathbb{E}[\nabla g(x, y; \xi)]\|^2] \le \sigma_g^2.$$

for constants $\sigma_f^2 > 0$ and $\sigma_g^2 > 0$.

Stochastic bilevel optimization

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

- We consider the first-order algorithm class that accesses functions through *first-order oracles* that return estimators of first-order derivatives $\hat{\nabla} f(x, y; \zeta), \hat{\nabla} g(x, y; \xi)$ for a given query point (x, y).

We assume that

- The estimators are unbiased:

$$\mathbb{E}[\hat{\nabla} f(x, y; \zeta)] = \nabla f(x, y),$$
$$\mathbb{E}[\hat{\nabla} g(x, y; \xi)] = \nabla g(x, y),$$

- The variance of the estimators are bounded:

$$\mathbb{E}[\|\hat{\nabla} f(x, y; \zeta) - \mathbb{E}[\nabla f(x, y; \zeta)]\|^2] \leq \sigma_f^2,$$
$$\mathbb{E}[\|\hat{\nabla} g(x, y; \xi) - \mathbb{E}[\nabla g(x, y; \xi)]\|^2] \leq \sigma_g^2.$$

for constants $\sigma_f^2 > 0$ and $\sigma_g^2 > 0$.

Table of Contents

Penalty method

$$\min_{x\in\mathbb{R}^{d_x}} \quad F(x) := f(x, y^*(x)) \quad \text{s.t.} \quad y^*(x) \in \arg\min_{y\in\mathbb{R}^{d_y}} g(x,y), \qquad \textbf{(P)}$$

- The starting point of our approach is to convert **(P)** to an equivalent constrained single-level version:

$$\min_{x\in X,\ y\in\mathbb{R}^{d_y}} \quad f(x,y) \quad \text{s.t.} \quad g(x,y) - g^*(x) \leq 0,$$

  where $g^*(x) := g(x, y^*(x))$.
- The Lagrangian $\mathcal{L}_\lambda$ with multiplier $\lambda > 0$ is

$$\mathcal{L}_\lambda(x,y) := f(x,y) + \lambda(g(x,y) - g^*(x)).$$

- The gradient of $\mathcal{L}_\lambda$ can be computed only with gradients of f and g, and thus the entire procedure can be implemented using only first-order derivatives. This reformulation has been attempted by (Liu et al., 2021; Sow et al., 2022; Ye et al., 2022)).

Penalty method

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

- The starting point of our approach is to convert (**P**) to an equivalent constrained single-level version:

$$\min_{x \in X,\ y \in \mathbb{R}^{d_y}} \quad f(x, y) \quad \text{s.t.} \quad g(x, y) - g^*(x) \leq 0,$$

  where $g^*(x) := g(x, y^*(x))$.
- The Lagrangian $\mathcal{L}_\lambda$ with multiplier $\lambda > 0$ is

$$\mathcal{L}_\lambda(x, y) := f(x, y) + \lambda(g(x, y) - g^*(x)).$$

- The gradient of $\mathcal{L}_\lambda$ can be computed only with gradients of f and g, and thus the entire procedure can be implemented using only first-order derivatives. This reformulation has been attempted by (Liu et al., 2021; Sow et al., 2022; Ye et al., 2022)).

Difficulties in penalty method

$$\min_{x\in\mathbb{R}^{d_x}} \quad F(x) := f(x, y^*(x)) \quad \text{s.t.} \quad y^*(x) \in \arg\min_{y\in\mathbb{R}^{d_y}} g(x, y), \tag{P}$$

$$\mathcal{L}_\lambda(x, y) := f(x, y) + \lambda(g(x, y) - g^*(x)).$$

- The challenge is to find an appropriate value of the multiplier λ. Unfortunately, the desired solution $x^* = \arg\min_x F(x)$ can only be obtained at $\lambda = \infty$.
- With $\lambda = \infty$, $\mathcal{L}_\lambda(x, y)$ has unbounded smoothness, which prevents us from employing gradient-descent style approaches.
- None of the previously proposed algorithms can obtain a complete finite time analysis for the original problem $\min_x F(x)$ without access to second derivatives of g.

Difficulties in penalty method

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

$$\mathcal{L}_\lambda(x, y) := f(x, y) + \lambda(g(x, y) - g^*(x)).$$

- The challenge is to find an appropriate value of the multiplier λ. Unfortunately, the desired solution $x^* = \arg\min_x F(x)$ can only be obtained at $\lambda = \infty$.
- With $\lambda = \infty$, $\mathcal{L}_\lambda(x, y)$ has unbounded smoothness, which prevents us from employing gradient-descent style approaches.
- None of the previously proposed algorithms can obtain a complete finite time analysis for the original problem $\min_x F(x)$ without access to second derivatives of g.

Our approach

Recall

$$\min_{x\in\mathbb{R}^{d_x}} \quad F(x) := f(x, y^*(x)) \quad \text{s.t.} \quad y^*(x) \in \arg\min_{y\in\mathbb{R}^{d_y}} g(x,y), \qquad \textbf{(P)}$$

$$\mathcal{L}_\lambda(x,y) := f(x,y) + \lambda(g(x,y) - g(x, y^*(x))).$$

Set $\mathcal{L}_\lambda^*(x) := \min_y \mathcal{L}_\lambda(x,y)$.

Lemma (J. Kwon-D. Kwon-Wright-Nowak, ICML 2023 Oral)

F can be approximated by $\mathcal{L}_\lambda^*(x)$ in the sense that

$$\|\nabla F(x) - \nabla \mathcal{L}_\lambda^*(x)\| \le O(1/\lambda)$$

where

$$\nabla \mathcal{L}_\lambda^*(x) = \nabla_x f(x, y_\lambda^*(x)) + \lambda(\nabla_x g(x, y_\lambda^*(x)) - \nabla_x g(x, y^*(x))),$$

and $y_\lambda^*(x) := \arg\min_y \left(\lambda^{-1} f(x,y) + g(x,y)\right)$.

- Therefore, we can find an ϵ-stationary point of $\mathcal{L}_\lambda^*(x)$, by running a stochastic gradient descent (SGD) style method on $\mathcal{L}_\lambda^*(x)$ with $\lambda = O(\epsilon^{-1})$.

Our approach

Recall

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

$$\mathcal{L}_\lambda(x, y) := f(x, y) + \lambda(g(x, y) - g(x, y^*(x))).$$

Set $\mathcal{L}_\lambda^*(x) := \min_y \mathcal{L}_\lambda(x, y)$.

Lemma (J. Kwon-D. Kwon-Wright-Nowak, ICML 2023 Oral)

F can be approximated by $\mathcal{L}_\lambda^*(x)$ in the sense that

$$\|\nabla F(x) - \nabla \mathcal{L}_\lambda^*(x)\| \leq O(1/\lambda)$$

where

$$\nabla \mathcal{L}_\lambda^*(x) = \nabla_x f(x, y_\lambda^*(x)) + \lambda(\nabla_x g(x, y_\lambda^*(x)) - \nabla_x g(x, y^*(x))),$$

and $y_\lambda^*(x) := \arg\min_y \left(\lambda^{-1} f(x, y) + g(x, y)\right)$.

- Therefore, we can find an ϵ-stationary point of $\mathcal{L}_\lambda^*(x)$, by running a stochastic gradient descent (SGD) style method on $\mathcal{L}_\lambda^*(x)$ with $\lambda = O(\epsilon^{-1})$.

Our approach

Recall

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

$$\mathcal{L}_\lambda(x, y) := f(x, y) + \lambda(g(x, y) - g(x, y^*(x))).$$

Set $\mathcal{L}^*_\lambda(x) := \min_y \mathcal{L}_\lambda(x, y)$.

Lemma (J. Kwon-D. Kwon-Wright-Nowak, ICML 2023 Oral)

F can be approximated by $\mathcal{L}^*_\lambda(x)$ in the sense that

$$\|\nabla F(x) - \nabla \mathcal{L}^*_\lambda(x)\| \le O(1/\lambda)$$

where

$$\nabla \mathcal{L}^*_\lambda(x) = \nabla_x f(x, y^*_\lambda(x)) + \lambda(\nabla_x g(x, y^*_\lambda(x)) - \nabla_x g(x, y^*(x))),$$

and $y^*_\lambda(x) := \arg\min_y \left(\lambda^{-1} f(x, y) + g(x, y)\right)$.

- Therefore, we can find an ϵ-stationary point of $\mathcal{L}^*_\lambda(x)$, by running a stochastic gradient descent (SGD) style method on $\mathcal{L}^*_\lambda(x)$ with $\lambda = O(\epsilon^{-1})$.

Our approach

Recall

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

$$\mathcal{L}_\lambda(x, y) := f(x, y) + \lambda(g(x, y) - g(x, y^*(x))).$$

Set $\mathcal{L}_\lambda^*(x) := \min_y \mathcal{L}_\lambda(x, y)$.

Lemma (J. Kwon-D. Kwon-Wright-Nowak, ICML 2023 Oral)

F can be approximated by $\mathcal{L}_\lambda^*(x)$ in the sense that

$$\|\nabla F(x) - \nabla \mathcal{L}_\lambda^*(x)\| \leq O(1/\lambda)$$

where

$$\nabla \mathcal{L}_\lambda^*(x) = \nabla_x f(x, y_\lambda^*(x)) + \lambda(\nabla_x g(x, y_\lambda^*(x)) - \nabla_x g(x, y^*(x))),$$

and $y_\lambda^*(x) := \arg\min_y \left(\lambda^{-1} f(x, y) + g(x, y)\right)$.

- Therefore, we can find an ϵ-stationary point of $\mathcal{L}_\lambda^*(x)$, by running a stochastic gradient descent (SGD) style method on $\mathcal{L}_\lambda^*(x)$ with $\lambda = O(\epsilon^{-1})$.

Our proposed algorithm

Recall $y^*(x) := \arg\min_y g(x,y)$, $y^*_\lambda(x) := \arg\min_y \left(\lambda^{-1} f(x,y) + g(x,y)\right)$, and

$$\nabla \mathcal{L}^*_\lambda(x) = \nabla_x f(x, y^*_\lambda(x)) + \lambda(\nabla_x g(x, y^*_\lambda(x)) - \nabla_x g(x, y^*(x))).$$

1. Outer-loop updates x^k using $\nabla \mathcal{L}^*_\lambda(x^k)$: $x^{k+1} = x^k - \alpha \hat{G}_k$ where

$$G_k := \nabla_x f(x^k, y^{k+1}) + \lambda(\nabla_x g(x^k, y^{k+1}) - \nabla_x g(x^k, z^{k+1})).$$

2. Inner-loop solves $y^*_{\lambda_k}(x^k)$, and $y^*(x^k)$ (approximately): y^{k+1} and z^{k+1} are the estimates of $y^*_\lambda(x^k)$ and $y^*(x^k)$ at the k^{th} iteration, respectively

Our proposed algorithm

Recall $y^*(x) := \arg\min_y g(x,y)$, $y^*_\lambda(x) := \arg\min_y \left(\lambda^{-1} f(x,y) + g(x,y)\right)$, and

$$\nabla \mathcal{L}^*_\lambda(x) = \nabla_x f(x, y^*_\lambda(x)) + \lambda(\nabla_x g(x, y^*_\lambda(x)) - \nabla_x g(x, y^*(x))).$$

1. Outer-loop updates x^k using $\nabla \mathcal{L}^*_\lambda(x^k)$: $x^{k+1} = x^k - \alpha \hat{G}_k$ where

$$G_k := \nabla_x f(x^k, y^{k+1}) + \lambda(\nabla_x g(x^k, y^{k+1}) - \nabla_x g(x^k, z^{k+1})).$$

2. Inner-loop solves $y^*_{\lambda_k}(x^k)$, and $y^*(x^k)$ (approximately): y^{k+1} and z^{k+1} are the estimates of $y^*_\lambda(x^k)$ and $y^*(x^k)$ at the k^{th} iteration, respectively

Our proposed algorithm

Recall $y^*(x) := \arg\min_y g(x,y)$, $y^*_\lambda(x) := \arg\min_y \left(\lambda^{-1} f(x,y) + g(x,y)\right)$, and

$$\nabla \mathcal{L}^*_\lambda(x) = \nabla_x f(x, y^*_\lambda(x)) + \lambda(\nabla_x g(x, y^*_\lambda(x)) - \nabla_x g(x, y^*(x))).$$

1. Outer-loop updates x^k using $\nabla \mathcal{L}^*_\lambda(x^k)$: $x^{k+1} = x^k - \alpha \hat{G}_k$ where

$$G_k := \nabla_x f(x^k, y^{k+1}) + \lambda(\nabla_x g(x^k, y^{k+1}) - \nabla_x g(x^k, z^{k+1})).$$

2. Inner-loop solves $y^*_{\lambda_k}(x^k)$, and $y^*(x^k)$ (approximately): y^{k+1} and z^{k+1} are the estimates of $y^*_\lambda(x^k)$ and $y^*(x^k)$ at the k^{th} iteration, respectively

Our main results

Theorem (J. Kwon-D. Kwon-Wright-Nowak, ICML 2023 Oral)

Under suitable assumptions and step-sizes, the following convergence results hold.

1. *If stochastic noises are present in both upper-level objective f and lower-level objective g (i.e., $\sigma_f^2, \sigma_g^2 > 0$), then our algorithm finds an ϵ-stationary point within $O(\epsilon^{-7})$ iterations.*
2. *If we have access to exact information about f and g (i.e., $\sigma_f^2 = \sigma_g^2 = 0$), then our algorithm finds an ϵ-stationary point within $O(\epsilon^{-3})$ iterations.*

Table of Contents

Next questions

Question

1. *Are the convergence rates optimal?*
2. *Are the first-order methods necessarily slower than second-order methods?*

- Under the additional assumption, it is known that the second-order methods find the ϵ-stationary point within $O(\epsilon^{-4})$.

Deterministic case

Inner-loop

Solve $y^*_{\lambda_k}(x^k)$, and $y^*(x^k)$ (approximately).

- Indeed, these are convex optimization problems for large enough $\lambda > 0$:

$$y^*(x) \in \arg\min_{y \in \mathbb{R}^{d_y}} g(x, y) \text{ and } y^*_\lambda(x) := \arg\min_y \left(\lambda^{-1} f(x, y) + g(x, y)\right).$$

- Using this idea, (Chen et al., 2024) improves the complexity of our proposed algorithm from $O(\epsilon^{-3})$ to $O(\epsilon^{-2}\log(1/\epsilon))$.

Deterministic case

Inner-loop

Solve $y^*_{\lambda_k}(x^k)$, and $y^*(x^k)$ (approximately).

- Indeed, these are convex optimization problems for large enough $\lambda > 0$:

$$y^*(x) \in \arg\min_{y \in \mathbb{R}^{d_y}} g(x,y) \text{ and } y^*_\lambda(x) := \arg\min_y \left(\lambda^{-1} f(x,y) + g(x,y)\right).$$

- Using this idea, (Chen et al., 2024) improves the complexity of our proposed algorithm from $O(\epsilon^{-3})$ to $O(\epsilon^{-2}\log(1/\epsilon))$.

Stochastic case: optimal number of iterations

Outer-loop

Update x^k using $\nabla \mathcal{L}_\lambda^*(x^k)$:

$$x^{k+1} = x^k - \alpha \hat{G}_k.$$

Comparing

$$\nabla \mathcal{L}_\lambda^*(x^k) = \nabla_x f(x^k, y_\lambda^*(x^k)) + \lambda(\nabla_x g(x^k, y_\lambda^*(x^k)) - \nabla_x g(x^k, y^*(x^k))).$$

and

$$G_k := \nabla_x f(x^k, y^{k+1}) + \lambda(\nabla_x g(x^k, y^{k+1}) - \nabla_x g(x^k, z^{k+1}))$$

- $\|\nabla \mathcal{L}_\lambda^*(x^k) - G_k\|$ can be estimated by

$$\lambda(\|y^{k+1} - y_\lambda^*(x^k)\| + \|z^{k+1} - y^*(x^k)\|)$$

- To obtain $\|\nabla \mathcal{L}_\lambda^*(x^k) - G_k\| = O(\epsilon)$, we need $O(\epsilon/\lambda) = O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.
- $T \asymp \epsilon^{-4}$ inner-loop iterations are required to have $O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.

Stochastic case: optimal number of iterations

Outer-loop

Update x^k using $\nabla \mathcal{L}_\lambda^*(x^k)$:

$$x^{k+1} = x^k - \alpha \hat{G}_k.$$

Comparing

$$\nabla \mathcal{L}_\lambda^*(x^k) = \nabla_x f(x^k, y_\lambda^*(x^k)) + \lambda(\nabla_x g(x^k, y_\lambda^*(x^k)) - \nabla_x g(x^k, y^*(x^k))).$$

and

$$G_k := \nabla_x f(x^k, y^{k+1}) + \lambda(\nabla_x g(x^k, y^{k+1}) - \nabla_x g(x^k, z^{k+1}))$$

- $\|\nabla \mathcal{L}_\lambda^*(x^k) - G_k\|$ can be estimated by

$$\lambda(\|y^{k+1} - y_\lambda^*(x^k)\| + \|z^{k+1} - y^*(x^k)\|)$$

- To obtain $\|\nabla \mathcal{L}_\lambda^*(x^k) - G_k\| = O(\epsilon)$, we need $O(\epsilon/\lambda) = O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.
- $T \asymp \epsilon^{-4}$ inner-loop iterations are required to have $O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.

Stochastic case: optimal number of iterations

Outer-loop

Update x^k using $\nabla\mathcal{L}_\lambda^*(x^k)$:

$$x^{k+1} = x^k - \alpha \hat{G}_k.$$

Comparing

$$\nabla\mathcal{L}_\lambda^*(x^k) = \nabla_x f(x^k, y_\lambda^*(x^k)) + \lambda(\nabla_x g(x^k, y_\lambda^*(x^k)) - \nabla_x g(x^k, y^*(x^k))).$$

and

$$G_k := \nabla_x f(x^k, y^{k+1}) + \lambda(\nabla_x g(x^k, y^{k+1}) - \nabla_x g(x^k, z^{k+1}))$$

- $\|\nabla\mathcal{L}_\lambda^*(x^k) - G_k\|$ can be estimated by

$$\lambda(\|y^{k+1} - y_\lambda^*(x^k)\| + \|z^{k+1} - y^*(x^k)\|)$$

- To obtain $\|\nabla\mathcal{L}_\lambda^*(x^k) - G_k\| = O(\epsilon)$, we need $O(\epsilon/\lambda) = O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.
- $T \asymp \epsilon^{-4}$ inner-loop iterations are required to have $O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.

Stochastic case: optimal number of iterations

Outer-loop

Update x^k using $\nabla \mathcal{L}_\lambda^*(x^k)$:

$$x^{k+1} = x^k - \alpha \hat{G}_k.$$

Comparing

$$\nabla \mathcal{L}_\lambda^*(x^k) = \nabla_x f(x^k, y_\lambda^*(x^k)) + \lambda(\nabla_x g(x^k, y_\lambda^*(x^k)) - \nabla_x g(x^k, y^*(x^k))).$$

and

$$G_k := \nabla_x f(x^k, y^{k+1}) + \lambda(\nabla_x g(x^k, y^{k+1}) - \nabla_x g(x^k, z^{k+1}))$$

- $\|\nabla \mathcal{L}_\lambda^*(x^k) - G_k\|$ can be estimated by

$$\lambda(\|y^{k+1} - y_\lambda^*(x^k)\| + \|z^{k+1} - y^*(x^k)\|)$$

- To obtain $\|\nabla \mathcal{L}_\lambda^*(x^k) - G_k\| = O(\epsilon)$, we need $O(\epsilon/\lambda) = O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.
- $T \asymp \epsilon^{-4}$ inner-loop iterations are required to have $O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.

Stochastic gradient descent

$T \asymp \epsilon^{-4}$ inner-loop iterations are required to have $O(\epsilon^2)$ accuracy of y^{k+1} and z^{k+1}.

- Let f be a L-smooth and μ-strongly convex function for some $\mu, L > 0$.
- $G(x, \xi)$ is an unbiased stochastic gradient estimator for f:

$$\mathbb{E}[G(x, \xi)] = \nabla f(x).$$

- The variance of the gradient estimation error is bounded:

$$\mathbb{E}[\|G(x, \xi) - \nabla f(x)\|^2] \leq \sigma^2.$$

Lemma

For $x_{t+1} \leftarrow x_t - \alpha G(x_t, \xi_t)$ and for all $0 \leq t \leq T$,

$$\mathbb{E}[\|x^t - x^*\|^2] \leq (1 - \mu\alpha)^t \|x^0 - x^*\|^2 + \frac{\alpha\sigma^2}{\mu}.$$

In particular, taking $\alpha = \frac{8 \log T}{\mu T}$, we have

$$\mathbb{E}[\|x^T - x^*\|^2] \leq \frac{1}{T^4}\|x^0 - x^*\|^2 + \frac{8 \log T}{\mu^2 T}\sigma^2.$$

Our main results

1. Outer-loop updates x^k using $\nabla \mathcal{L}_\lambda^*(x^k)$ with K iterations.
2. Inner-loop solves $y_{\lambda_k}^*(x^k)$, and $y^*(x^k)$ with T iterations.

Theorem (J. Kwon-D. Kwon-Lyu, ICML 2024)

Under suitable assumptions, step-sizes, $K \asymp \epsilon^{-2}$, and $T \asymp \epsilon^{-4}$,

1. *Our algorithm finds an ϵ-stationary point within $O(\epsilon^{-6})$ iterations.*
2. *If we additionally assume the stochastic smoothness as in the second order method, then our algorithm finds an ϵ-stationary point within $O(\epsilon^{-4})$ iterations.*

- $\mathbb{E}[\|\hat{\nabla} g(x, y^1; \xi) - \hat{\nabla} g(x, y^2; \xi)\|^2] \leq C\|y^1 - y^2\|^2$

Table of Contents

Lower bound

Question

Are the convergence rates optimal?

- In (J. Kwon-D. Kwon-Lyu, ICML 2024), we provide the matching ϵ^{-6} lower bound on y^*-aware oracles with finite $r \asymp \epsilon$.
- Under the same condition, ϵ^{-6} upper bound can be shown.

Lower bound

Definition (y^*-Aware Oracle)

An oracle is y^*-*aware*, if there exists $r \in (0, \infty]$ such that for every query point (x, y), the following conditions hold.

- In addition to stochastic gradients, the oracle also returns $\hat{y}(x)$ such that $\|\hat{y}(x) - y^*(x)\| \leq r/2$
- Gradient estimators satisfy the assumptions only if $\|y - y^*(x)\| \leq r$; otherwise, the returned gradient estimators can be arbitrary.

- If we take $r = \infty$, the additional estimator $\hat{y}(x)$ is uninformative. We recover the usual first-order stochastic gradient oracle.
- The same upper bound holds for finite r.

Lower bound

Definition (y^*-Aware Oracle)

An oracle is y^*-*aware*, if there exists $r \in (0, \infty]$ such that for every query point (x, y), the following conditions hold.

- In addition to stochastic gradients, the oracle also returns $\hat{y}(x)$ such that $\|\hat{y}(x) - y^*(x)\| \leq r/2$
- Gradient estimators satisfy the assumptions only if $\|y - y^*(x)\| \leq r$; otherwise, the returned gradient estimators can be arbitrary.

- If we take $r = \infty$, the additional estimator $\hat{y}(x)$ is uninformative. We recover the usual first-order stochastic gradient oracle.
- The same upper bound holds for finite r.

Non-convex lower level

- If g is not convex, then $y^*(x)$ and $y^*_\lambda(x)$ may not be uniquely determined.
- A solution set $T(x,\lambda) := \arg\min_y \left(\lambda^{-1} f(x,y) + g(x,y)\right)$ may not be stable.
- In (J. Kwon-D. Kwon-Wright-Nowak, ICLR 2024), similar convergence results are given under the Lipschitz continuity of T.

Non-convex lower level

- If g is not convex, then $y^*(x)$ and $y^*_\lambda(x)$ may not be uniquely determined.
- A solution set $T(x,\lambda) := \arg\min_y \left(\lambda^{-1} f(x,y) + g(x,y)\right)$ may not be stable.
- In (J. Kwon-D. Kwon-Wright-Nowak, ICLR 2024), similar convergence results are given under the Lipschitz continuity of T.

Summary

- We provide a complete finite-time analysis of the first-order method for bilevel optimization.
- Under a fair comparison, our proposed method is not necessarily slower than second-order ones.
- Lower bounds and non-convex cases are open.

- Further applications in large-scale machine learning problems?

Summary

- We provide a complete finite-time analysis of the first-order method for bilevel optimization.
- Under a fair comparison, our proposed method is not necessarily slower than second-order ones.
- Lower bounds and non-convex cases are open.

+

- Further applications in large-scale machine learning problems?

References

- Jeongyeol Kwon, Dohyun Kwon, and Hanbaek Lyu. On the Complexity of First-Order Methods in Stochastic Bilevel Optimization. The 41st International Conference on Machine Learning, PMLR 235:25784-25811. (ICML 2024)
- Jeongyeol Kwon, Dohyun Kwon, Steve Wright, and Robert Nowak. On Penalty Methods for Nonconvex Bilevel Optimization and First-Order Stochastic Approximation. The Twelfth International Conference on Learning Representations. (ICLR 2024, Spotlight)
- Jeongyeol Kwon, Dohyun Kwon, Steve Wright, and Robert Nowak. A Fully First-Order Method for Stochastic Bilevel Optimization. The 40th International Conference on Machine Learning, PMLR 202:18083-18113. (ICML 2023, Oral)

Thank you for your attention!