

A framework for bilevel optimization that enables stochastic and global variance reduction algorithms



Mathieu Dagr  ou¹ Pierre Ablin² Samuel Vaiter³ Thomas Moreau¹

¹Univ. Paris-Saclay, Inria, Mind team, Saclay, France.

²CNRS, Universit   Paris-Dauphine, PSL-University, Paris, France.

³CNRS & Universit   C  te d'Azur, LJAD, Nice, France.



1. Problem Statement and applications

Bilevel Optimization Problems

Let $F, G : \mathbb{R}^p \times \mathbb{R}^d \rightarrow \mathbb{R}$

► Goal:

$$\min_{x \in \mathbb{R}^d} h(x) \triangleq F(z^*(x), x), \text{ s.t. } z^*(x) \in \arg \min_{z \in \mathbb{R}^p} G(z, x).$$

► ERM setting:

$$F(z, x) = \frac{1}{m} \sum_{j=1}^m F_j(z, x), \quad G(z, x) = \frac{1}{n} \sum_{i=1}^n G_i(z, x)$$

Examples of applications

- Hyperparameter selection
- Deep Equilibrium Models
- Neural architecture search
- Data Augmentation

2. First order methods

Gradient descent

$$x^{t+1} = x^t - \gamma \nabla h(x^t)$$

Implicit differentiation

$$\nabla h(x) = \nabla_2 F(z^*(x), x) + \nabla_{21}^2 G(z^*(x), x) v^*(x)$$

with

$$v^*(x) = -[\nabla_{11}^2 G(z^*(x), x)]^{-1} \nabla_1 F(z^*(x), x)$$

Bottlenecks

- One optimization problem to solve at each iteration
- One linear system to solve at each iteration
- Computation of the full batch derivatives when m and n are large

3. General framework

Idea: Maintain the variables z, v, x by alternating steps in the following directions

- $D_z(z, v, x) = \nabla_1 G(z, x)$: gradient step towards $z^*(x)$
- $D_v(z, v, x) = \nabla_{11}^2 G(z, x) v + \nabla_1 F(z, x)$: gradient step towards $v^*(x)$
- $D_x(z, v, x) = \nabla_{21}^2 G(z, x) v + \nabla_2 F(z, x)$: gradient step towards x^*

Input: initializations $z_0 \in \mathbb{R}^p, x_0 \in \mathbb{R}^d, v_0 \in \mathbb{R}^p$, number of iterations T , step size sequences $(\rho^t)_{t < T}$ and $(\gamma^t)_{t < T}$.

for $t = 0, \dots, T - 1$ **do**

Update z : $z^{t+1} = z^t - \rho^t D_z^t$,

Update v : $v^{t+1} = v^t - \rho^t D_v^t$,

Update x : $x^{t+1} = x^t - \gamma^t D_x^t$,

where D_z^t, D_v^t and D_x^t are unbiased estimators of $D_z(z^t, v^t, x^t), D_v(z^t, v^t, x^t)$ and $D_x(z^t, v^t, x^t)$.

end

Output: (z^T, v^T, x^T)

4. Stochastic Bilevel Algorithm (SOBA)

Chosen directions

Pick $i \in [n]$ and $j \in [m]$ and take

- $D_z^t = \nabla_1 G_i(z^t, x^t)$
- $D_v^t = \nabla_{11}^2 G_i(z^t, x^t) v^t + \nabla_1 F_j(z^t, x^t)$
- $D_x^t = \nabla_{21}^2 G_i(z^t, x^t) v^t + \nabla_2 F_j(z^t, x^t)$

Convergence rate

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla h(x^t)\|^2] = \mathcal{O}(T^{-\frac{1}{2}})$$

Same sample complexity as SGD for non-convex single level problems!

5. Aside: SAGA for single level problem

Single level problem

$$\min_{x \in \mathbb{R}^p} f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$$

Initialization Compute and store $m[i] = \nabla f_i(x^0)$ for any $i \in \{1, \dots, n\}$ and $S[m] = \frac{1}{n} \sum_{i=1}^n m[i]$.

At iteration t :

1 Pick $i \in \{1, \dots, n\}$

2 Update x

$$x^{t+1} = x^t - \rho (\nabla f_i(x^t) \underbrace{- m[i] + S[m]}_{\text{variance reduction}})$$

3 Update the memory

$$m[i] \leftarrow \nabla f_i(x^t)$$

6. Stochastic Averaged Bilevel Algorithm (SABA)

Chosen directions

Estimate the five quantities $\nabla_1 G(z^t, x^t), \nabla_1 F(z^t, x^t), \nabla_2 F(z^t, x^t), \nabla_{12}^2 G(z^t, x^t) v^t, \nabla_{11}^2 G(z^t, x^t) v^t$ on the principle of SAGA and plug these estimates in $D_z(z^t, v^t, x^t), D_v(z^t, v^t, x^t), D_x(z^t, v^t, x^t)$.

Convergence rate

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla h(x^t)\|^2] = \mathcal{O}((n+m)^{\frac{2}{3}} T^{-1})$$

Same sample complexity as SAGA for non-convex single level problems!

6. Application to hyperparameter selection

Setting

- Task: binary classification
- IJCNN1 dataset: 49 990 training samples, 91 701 validation samples, 22 features
- Training loss:

$$G(\theta, \lambda) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i \langle x_i, \theta \rangle)) + \frac{1}{2} \sum_{k=1}^p e^{\lambda_k} \theta_k^2$$

- Validation loss: logistic loss

$$F(\theta, \lambda) = \frac{1}{m} \sum_{j=1}^m \log(1 + \exp(-y_j^{\text{val}} \langle x_j^{\text{val}}, \theta \rangle))$$

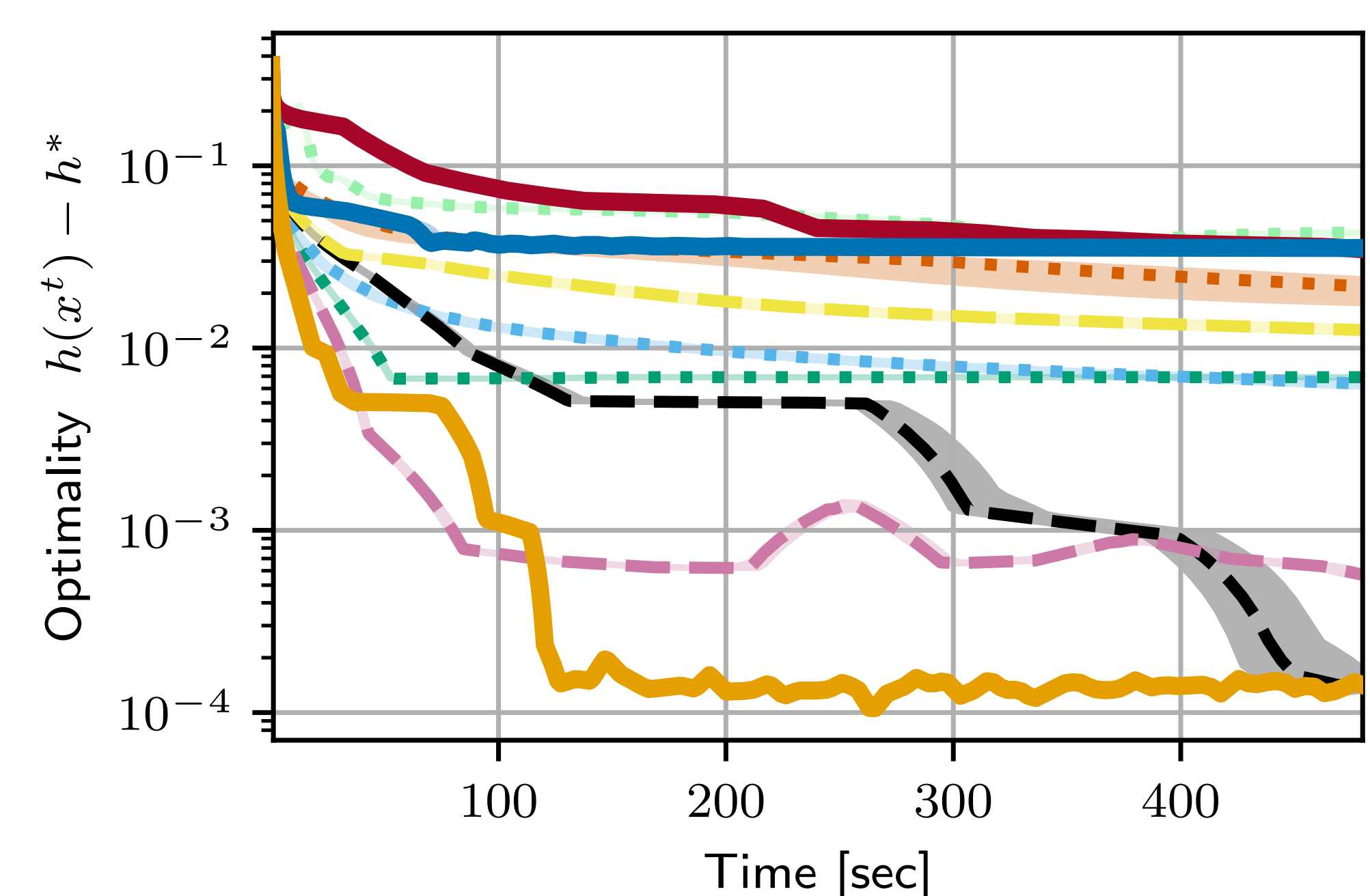


Figure: Comparison of SOBA and SABA with other stochastic bilevel optimization methods.