Optimization with Access To Auxiliary Information

El Mahdi Chayti [1] Sai Praneeth Karimireddy [2]

[1]Swiss Federal Technology Institute of Lausanne

[2]University of Southern California

March 31, 2025

ICLR International Conference On Learning Representations

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

USC University of Southern California

Introduction

We are interested in the following problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) := \mathbb{E}_{\xi_f}\left[f(\boldsymbol{x};\xi_f)\right] \text{ given } h(\boldsymbol{x}) := \mathbb{E}_{\xi_f}\left[h(\boldsymbol{x};\xi_h)\right]$$

We are interested in the following problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) := \mathbb{E}_{\xi_f}\left[f(\boldsymbol{x}; \xi_f)\right] \text{ given } h(\boldsymbol{x}) := \mathbb{E}_{\xi_f}\left[h(\boldsymbol{x}; \xi_h)\right]$$

Question:
How can we leverage an auxiliary $h(\boldsymbol{x})$ to speed up the optimization of our target loss function $f(\boldsymbol{x})$?

Examples

The helper h can be much "cheaper": cheap, more accessible, ...

Examples

The helper h can be much "cheaper": cheap, more accessible, ...
we can have multiple helpers: $h_1, \cdots, h_n$.

Examples

The helper h can be much "cheaper": cheap, more accessible, ...
we can have multiple helpers: $h_1, \cdots, h_n$.

Examples:

Examples

The helper h can be much "cheaper": cheap, more accessible, ...
we can have multiple helpers: $h_1, \cdots, h_n$.

Examples:

- Federated Learning: $f \leftarrow$ average, $h_i \leftarrow \{\text{clients}\}$.

Examples

The helper h can be much "cheaper": cheap, more accessible, ...
we can have multiple helpers: $h_1, \cdots, h_n$.

Examples:

- Federated Learning: $f \leftarrow$ average, $h_i \leftarrow \{$clients$\}$.
- Personalized Learning: $f \leftarrow$ client$_0$, $h \leftarrow \{$other clients$\}$.

Examples

The helper h can be much "cheaper": cheap, more accessible, ...
we can have multiple helpers: $h_1, \cdots, h_n$.

Examples:

- Federated Learning: $f \leftarrow$ average, $h_i \leftarrow \{$clients$\}$.
- Personalized Learning: $f \leftarrow$ client$_0$, $h \leftarrow \{$other clients$\}$.
- Semi-supervised Learning: $f \leftarrow$ Labeled, $h \leftarrow$ unlabeled.

Examples

The helper h can be much "cheaper": cheap, more accessible, ...
we can have multiple helpers: $h_1, \cdots, h_n$.

Examples:

- Federated Learning: $f \leftarrow$ average, $h_i \leftarrow$ {clients}.
- Personalized Learning: $f \leftarrow$ client$_0$, $h \leftarrow$ {other clients}.
- Semi-supervised Learning: $f \leftarrow$ Labeled, $h \leftarrow$ unlabeled.
- Core-sets: $f \leftarrow$ large dataset, $h \leftarrow$ core-set.

Approach

We write f as

$$f(\boldsymbol{z}) := \underbrace{h(\boldsymbol{z})}_{\text{cheap}} + \underbrace{f(\boldsymbol{z}) - h(\boldsymbol{z})}_{\text{expensive}} .$$

Approach

We write f as

$$f(\boldsymbol{z}) := \underbrace{h(\boldsymbol{z})}_{\text{cheap}} + \underbrace{f(\boldsymbol{z}) - h(\boldsymbol{z})}_{\text{expensive}} .$$

Let $\boldsymbol{x}$ be the global state (slow) and $\boldsymbol{y}$ the local state (fast).

We write f as

$$f(\boldsymbol{z}) := \underbrace{h(\boldsymbol{z})}_{\text{cheap}} + \underbrace{f(\boldsymbol{z}) - h(\boldsymbol{z})}_{\text{expensive}} .$$

Let $\boldsymbol{x}$ be the global state (slow) and $\boldsymbol{y}$ the local state (fast).

Main idea: Linearize h around $\boldsymbol{y}$ and $f - h$ around $\boldsymbol{x}$.

Approach

We write f as

$$f(\boldsymbol{z}) := \underbrace{h(\boldsymbol{z})}_{\text{cheap}} + \underbrace{f(\boldsymbol{z}) - h(\boldsymbol{z})}_{\text{expensive}}.$$

Gradient: $\nabla h(\boldsymbol{y}) + \nabla f(\boldsymbol{x}) - \nabla h(\boldsymbol{x})$.

Approach

We write f as

$$f(\boldsymbol{z}) := \underbrace{h(\boldsymbol{z})}_{\text{cheap}} + \underbrace{f(\boldsymbol{z}) - h(\boldsymbol{z})}_{\text{expensive}} .$$

Gradient: $\nabla h(\boldsymbol{y}, \xi_h) + \underbrace{\boldsymbol{m}_{f-h}}_{\approx \nabla f(\boldsymbol{x}) - \nabla h(\boldsymbol{x})} .$

We write f as

$$f(\boldsymbol{z}) := \underbrace{h(\boldsymbol{z})}_{\text{cheap}} + \underbrace{f(\boldsymbol{z}) - h(\boldsymbol{z})}_{\text{expensive}}.$$

Gradient: $\nabla h(\boldsymbol{y}, \xi_h) + \underbrace{\boldsymbol{m}_{f-h}}_{\approx \nabla f(\boldsymbol{x}) - \nabla h(\boldsymbol{x})}.$

AuxMOM : $\boldsymbol{m}_{f-h} \leftarrow (1-a)\boldsymbol{m}_{f-h} + a\nabla(f-h)(\boldsymbol{x}; \xi_{f-h})$

Algorithm: AuxMOM

All in all:

All in all:

1) $\boldsymbol{m}_{f-h} \leftarrow (1-a)\boldsymbol{m}_{f-h} + a\nabla(f-h)(\boldsymbol{x};\xi_{f-h})$

Algorithm: AuxMOM

All in all:

1) $\boldsymbol{m}_{f-h} \leftarrow (1-a)\boldsymbol{m}_{f-h} + a\nabla(f-h)(\boldsymbol{x};\xi_{f-h})$
2) $\{\boldsymbol{y} \leftarrow \boldsymbol{y} - \eta(\nabla h(\boldsymbol{y},\xi_h) + \boldsymbol{m}_{f-h})\}$ repeat K times.

Algorithm: AuxMOM

All in all:

1) $\boldsymbol{m}_{f-h} \leftarrow (1-a)\boldsymbol{m}_{f-h} + a\nabla(f-h)(\boldsymbol{x};\xi_{f-h})$
2) $\{\boldsymbol{y} \leftarrow \boldsymbol{y} - \eta(\nabla h(\boldsymbol{y},\xi_h) + \boldsymbol{m}_{f-h})\}$ repeat K times.
3) $\boldsymbol{x} \leftarrow \boldsymbol{y}$.

(Smoothness.)

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2 \,.$$

(Smoothness.)

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2 .$$

(Variance.) Unbiasedness + bounded variance

$$\mathbb{E}_{\zeta_J}\|\nabla J(\boldsymbol{x};\zeta_J) - \nabla J(\boldsymbol{x})\|_2^2 \leq \sigma_J^2 , J \in \{f, h, f - h\} .$$

(Smoothness.)

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2 \,.$$

(Variance.) Unbiasedness + bounded variance

$$\mathbb{E}_{\zeta_J}\|\nabla J(\boldsymbol{x};\zeta_J) - \nabla J(\boldsymbol{x})\|_2^2 \leq \sigma_J^2\,, J \in \{f, h, f - h\}\,.$$

Hessian similarity. $\exists \delta \in [0, 2L]$ we have

$$\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 h(\boldsymbol{x})\|_2 \leq \delta\,.$$

AuxMOM iteration complexity:

To get $\mathbb{E}[\|\nabla f(\hat{\boldsymbol{x}})\|_2^2] \le \varepsilon$, AuxMOM needs at most

$$\mathcal{O}\Big(\frac{\delta F^0 \sigma_{f-h}^2}{\varepsilon^2} + \frac{\delta F^0}{\varepsilon} + \frac{\sigma_{f-h}^2}{\varepsilon}\Big)$$

(stochastic) gradient calls of f.

AuxMOM iteration complexity:

To get $\mathbb{E}[\|\nabla f(\hat{\boldsymbol{x}})\|_2^2] \leq \varepsilon$, AuxMOM needs at most

$$\mathcal{O}\Big(\frac{\delta F^0 \sigma_{f-h}^2}{\varepsilon^2} + \frac{\delta F^0}{\varepsilon} + \frac{\sigma_{f-h}^2}{\varepsilon}\Big)$$

(stochastic) gradient calls of f.

Gain: Compare to $\mathcal{O}\Big(\frac{LF^0\sigma_f^2}{\varepsilon^2} + \frac{LF^0}{\varepsilon}\Big)$

AuxMOM iteration complexity:

To get $\mathbb{E}[\|\nabla f(\hat{\boldsymbol{x}})\|_2^2] \le \varepsilon$, AuxMOM needs at most

$$\mathcal{O}\Big(\frac{\delta F^0 \sigma_{f-h}^2}{\varepsilon^2} + \frac{\delta F^0}{\varepsilon} + \frac{\sigma_{f-h}^2}{\varepsilon}\Big)$$

(stochastic) gradient calls of f.

AuxMOM iteration complexity:

To get $\mathbb{E}[\|\nabla f(\hat{\boldsymbol{x}})\|_2^2] \leq \varepsilon$, AuxMOM needs at most

$$\mathcal{O}\Big(\frac{\delta F^0 \sigma_{f-h}^2}{\varepsilon^2} + \frac{\delta F^0}{\varepsilon} + \frac{\sigma_{f-h}^2}{\varepsilon}\Big)$$

(stochastic) gradient calls of f.

Gain: we replaced L by δ and σ_f^2 by σ_{f-h}^2

AuxMOM iteration complexity:

To get $\mathbb{E}[\|\nabla f(\hat{\boldsymbol{x}})\|_2^2] \leq \varepsilon$, AuxMOM needs at most

$$\mathcal{O}\Big(\frac{\delta F^0 \sigma_{f-h}^2}{\varepsilon^2} + \frac{\delta F^0}{\varepsilon} + \frac{\sigma_{f-h}^2}{\varepsilon}\Big)$$

(stochastic) gradient calls of f.

Gain: we replaced L by δ and σ_f^2 by σ_{f-h}^2

Small dissimilarity or positive correlation $\implies$ gain.

AuxMOM iteration complexity:

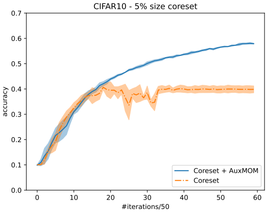
To get $\mathbb{E}[\|\nabla f(\hat{\boldsymbol{x}})\|_2^2] \leq \varepsilon$, AuxMOM needs at most

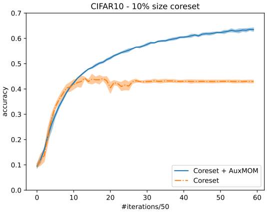
$$\mathcal{O}\Big(\frac{\delta F^0 \sigma_{f-h}^2}{\varepsilon^2} + \frac{\delta F^0}{\varepsilon} + \frac{\sigma_{f-h}^2}{\varepsilon}\Big)$$

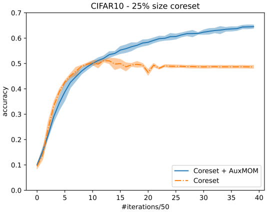
(stochastic) gradient calls of f.

What is the catch? $K = \mathcal{O}\Big(\frac{\sigma_h^2}{\varepsilon} + 1_{\delta \neq 0}\frac{L}{\delta} + 1\Big)$ inner steps of the helper h.

Core-sets


CIFAR10 - 5% size coreset
accuracy
#iterations/50
Coreset + AuxMOM
Coreset


CIFAR10 - 10% size coreset
accuracy
#iterations/50
Coreset + AuxMOM
Coreset


CIFAR10 - 25% size coreset
accuracy
#iterations/50
Coreset + AuxMOM
Coreset

Conclusion

- Introduced the framework of optimization with access to auxiliary information
- Showed how it can improve optimization without using a helper.
- The framework works on simple problems in practice.