

# Similarity in non-euclidian setup

Gasnikov A.V. (MIPT, SkolTech, IITP)

[gasnikov@yandex.ru](mailto:gasnikov@yandex.ru)

*Based on joint work with Alexander Beznosikov, Darina Dvinskikh, Dmitry Kovalev et al.*

AISTATS2022, NeurIPS 2022, AISTATS2023

*November 2, 2022*

# Sum type target convex function

$$f(x) = \mathbb{E}[f(x, \xi)] \rightarrow \min_{x \in Q \subseteq \mathbb{R}^n}.$$

How to choose  $m$  in:

$$\min_{x \in Q \subseteq \mathbb{R}^n} \frac{1}{m} \sum_{k=1}^m f(x, \xi^k) + \frac{\varepsilon}{2R^2} \|x - x^0\|_2^2$$

$\|x^0 - x_*\|_2$

Answer (up to a log-factor):

$$m = \min \left\{ O \left( \frac{M^2 R^2}{\varepsilon^2} \right), O \left( \frac{M^2}{\mu \varepsilon} \right) \right\}$$

Where:

$$\mathbb{E}[\|\nabla f(x, \xi)\|_2^2] \leq M^2$$

Strong convexity constant of  $f$

S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization. In *COLT*, 2009.

# Too many terms!

$$\min_{x \in Q \subseteq \mathbb{R}^n} \frac{1}{m} \sum_{k=1}^m f(x, \xi^k) + \frac{\varepsilon}{2R^2} \|x - x^0\|_2^2$$

We skip this term  
for simplicity

Problem: How to store data in memory?

Answer: To use distributed approach

Problem: Too many communications and oracle calls required

Answer: To store at each node a lot of data. And rewrite optimization problem

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M \left( \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i}) \right)$$

$$F_k(x) = \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i})$$

This problem has specific structure

1)  $F_k(x)$  has sum-type structure and variance reduction is possible

2) Statistical Similarity  $\|\nabla^2 F_k(x) - \nabla^2 F(x)\| = O\left(\sqrt{\frac{L_2^2 n}{r}}\right)$

# Variance reduction

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M \left( \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i}) \right)$$

*This problem has specific structure* →

$$F_k(x) = \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i})$$

*$F_k(x)$  has sum-type structure and variance reduction is possible*

---

## Variance Reduced EXTRA and DIGing and Their Optimal Acceleration for Strongly Convex Decentralized Optimization

---

Huan Li<sup>1</sup> Zhouchen Lin<sup>2</sup> Yongchun Fang<sup>1</sup>

### Abstract

We study stochastic decentralized optimization for the problem of training machine learning models with large-scale distributed data. We extend the widely used EXTRA and DIGing methods with variance reduction (VR), and propose two methods: VR-EXTRA and VR-DIGing. The proposed VR-EXTRA requires the time of  $\mathcal{O}((\kappa_s + n) \log \frac{1}{\epsilon})$  stochastic gradient evaluations and  $\mathcal{O}((\kappa_b + \kappa_c) \log \frac{1}{\epsilon})$  communication rounds to reach precision  $\epsilon$ , which are the best complexities among the non-accelerated gradient-type methods, where  $\kappa_s$  and  $\kappa_b$  are the stochastic condition number and batch condition number for strongly convex and smooth problems, respectively,  $\kappa_c$  is the condition number of the communication network, and  $n$  is the sample size on each distributed node. The proposed VR-DIGing has a little higher communication cost of  $\mathcal{O}((\kappa_b + \kappa_c^2) \log \frac{1}{\epsilon})$ . Our stochastic gradient computation complexities are the same as the ones of single-machine VR methods, such as SAG, SAGA, and SVRG, and our communication complexities keep the same as those of EXTRA and DIGing, respectively. To further speed up the convergence, we also propose the accelerated VR-EXTRA and VR-DIGing with both the optimal  $\mathcal{O}((\sqrt{n\kappa_s} + n) \log \frac{1}{\epsilon})$  stochastic gradient computation complexity and  $\mathcal{O}(\sqrt{\kappa_b \kappa_c} \log \frac{1}{\epsilon})$  communication complexity. Our stochastic gradient computation complexity is also the same as the ones of single-machine accelerated VR methods, such as Katyusha, and our communication complexity keeps the same as those of accelerated full batch decentralized methods, such as MSDA. To the best of our knowledge, our accelerated methods are the first to achieve both the optimal stochastic gradient computation complexity and communication complexity in the class of gradient-type methods.

# Variance reduction

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M \left( \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i}) \right)$$

This problem has specific structure 

$F_k(x)$  has sum-type structure and variance reduction is possible

Assumptions:

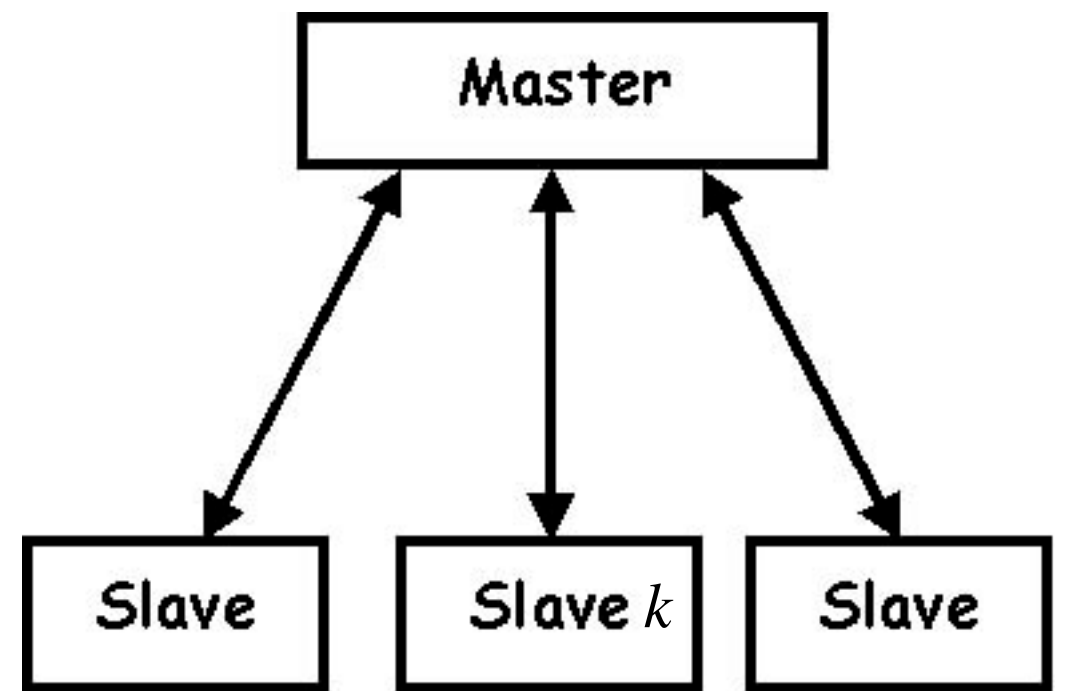
$f(x, \xi)$  is  $L$ -smooth in  $x$  for all  $\xi$


$f(x, \xi)$  is  $\lambda$ -strongly convex in  $x$  for all  $\xi$

Optimal bounds for general  $f_{k,i}(x) \neq f(x, \xi^{k,i})$ :

$$O\left(\sqrt{\frac{L}{\lambda}} \log\left(\frac{LR^2}{\varepsilon}\right)\right) \quad \text{Communication rounds}$$

$$O\left(\left(r + \sqrt{r \frac{L}{\lambda}}\right) \log\left(\frac{LR^2}{\varepsilon}\right)\right) \quad \text{Oracle calls per node}$$



$$F_k(x) = \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i})$$


# Statistical Similarity

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M \left( \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i}) \right)$$

This problem has specific structure

$F_k(x)$  has sum-type structure

Assumptions:

$f(x, \xi)$  is  $L$ -smooth in  $x$  for all  $\xi$

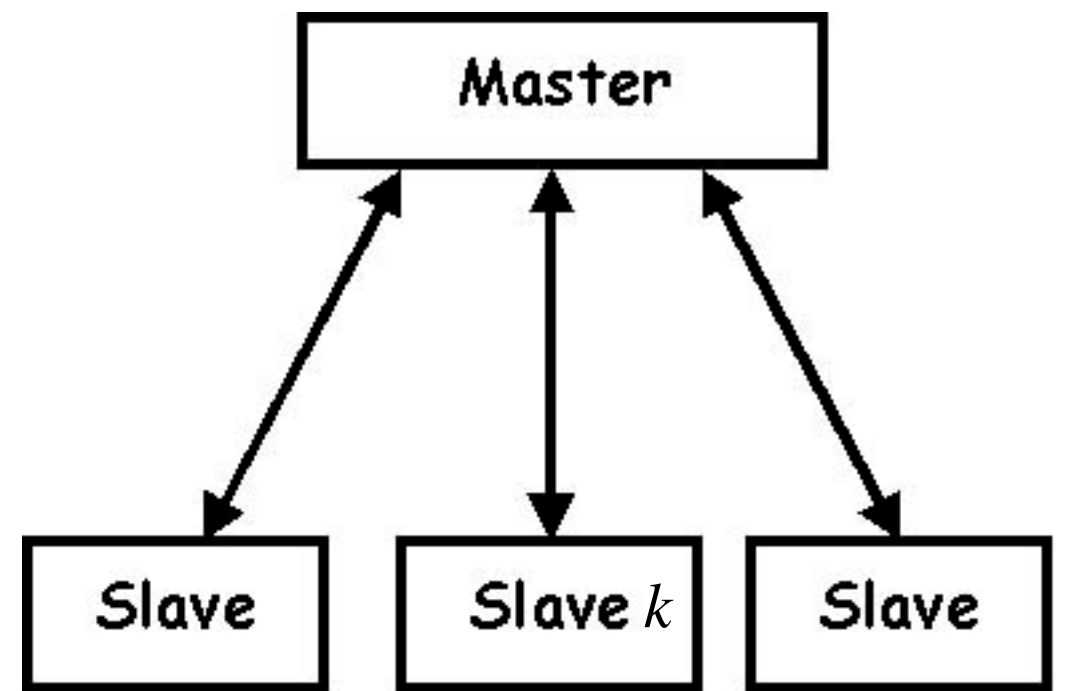
$f(x, \xi)$  is  $\lambda$ -strongly convex in  $x$  for all  $\xi$

Optimal bound ?:

$$O \left( \sqrt{\frac{L}{\lambda}} \log \left( \frac{LR^2}{\varepsilon} \right) \right) \quad \text{Communication rounds}$$

Is this bound optimal?

Answer: No, if we use similarity:  $f_{k,i}(x) = f(x, \xi^{k,i})$ ,  $\xi^{k,i}$  i.i.d.



$$F_k(x) = \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i})$$

# Statistical Similarity

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M \left( \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i}) \right)$$

This problem has specific structure

$$F_k(x) = \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i})$$

$F_k(x)$  has sum-type structure

Assumptions:

$f(x, \xi)$  is  $L$ -smooth in  $x$  for all  $\xi \Rightarrow \|\nabla^2 F_k(x)\| \leq L$

$f(x, \xi)$  is  $\lambda$ -strongly convex in  $x$  for all  $\xi$

Communication rounds

$$O\left(\sqrt{\frac{L}{\lambda}} \log\left(\frac{LR^2}{\varepsilon}\right)\right)$$

Variance reduction

$$O\left(\sqrt{\frac{\delta}{\lambda}} \log\left(\frac{LR^2}{\varepsilon}\right)\right)$$

$$\|\nabla^2 F_k(x) - \nabla^2 F(x)\| \leq \delta$$

$\delta$ -Similarity

Total oracle calls

$$\tilde{O}\left(M + M^{3/4} \sqrt{\frac{\delta}{\lambda}}\right)$$

Khaled A., Jin C. Faster federated optimization under second-order similarity //arXiv preprint arXiv:2209.02257. – 2022.



# Gradient method with relative smoothness and strong convexity

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M F_k(x)$$

$d(x)$  - Smooth convex function

$V(y, x) = d(y) - d(x) - \langle \nabla d(x), y - x \rangle$  - Bregman divergence

SIAM J. OPTIM.  
Vol. 28, No. 1, pp. 333–354

© 2018 Society for Industrial and Applied Mathematics

## RELATIVELY SMOOTH CONVEX OPTIMIZATION BY FIRST-ORDER METHODS, AND APPLICATIONS\*

HAIHAO LU<sup>†</sup>, ROBERT M. FREUND<sup>‡</sup>, AND YURII NESTEROV<sup>§</sup>

**Abstract.** The usual approach to developing and analyzing first-order methods for smooth convex optimization assumes that the gradient of the objective function is uniformly smooth with some Lipschitz constant  $L$ . However, in many settings the differentiable convex function  $f(\cdot)$  is not uniformly smooth—for example, in  $D$ -optimal design where  $f(x) := -\ln \det(HXH^T)$  and  $X := \text{Diag}(x)$ , or even the univariate setting with  $f(x) := -\ln(x) + x^2$ . In this paper we develop a notion of “relative smoothness” and relative strong convexity that is determined relative to a user-specified “reference function”  $h(\cdot)$  (that should be computationally tractable for algorithms), and we show that many differentiable convex functions are relatively smooth with respect to a correspondingly fairly simple reference function  $h(\cdot)$ . We extend two standard algorithms—the primal gradient scheme and the dual averaging scheme—to our new setting, with associated computational guarantees. We apply our new approach to develop a new first-order method for the  $D$ -optimal design problem, with associated computational complexity analysis. Some of our results have a certain overlap with the recent work [H. H. Bauschke, J. Bolte, and M. Teboulle, *Math. Oper. Res.*, 42 (2017), pp. 330–348].



# Gradient method with relative smoothness and strong convexity

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M F_k(x)$$

$$\lambda \nabla^2 d(x) \prec \nabla^2 F(x) \prec L \nabla^2 d(x)$$

$$V(y, x) = d(y) - d(x) - \langle \nabla d(x), y - x \rangle$$

$$x^{k+1} = \arg \min_{x \in Q} \left\{ F(x^k) + \langle \nabla F(x^k), x - x^k \rangle + L V(x, x^k) \right\}$$

$$F(x^N) - \min_{x \in Q} F(x) \leq \varepsilon$$

$$N = O \left( \frac{L}{\lambda} \log \left( \frac{\Delta F}{\varepsilon} \right) \right)$$

# Gradient method with relative smoothness and strong convexity and Similarity

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M F_k(x)$$

$$d(x) = F_1(x) + \frac{\delta}{2} \|x\|^2$$

Available at master  
node (1)

$$\frac{\lambda}{\lambda + 2\delta} \nabla^2 d(x) \prec \nabla^2 F(x) \prec \nabla^2 d(x)$$

$$x^{k+1} = \arg \min_{x \in Q} \left\{ F(x^k) + \langle \nabla F(x^k), x - x^k \rangle + V(x, x^k) \right\}$$

Available at master  
node (1) via  
communications

$$F(x^N) - \min_{x \in Q} F(x) \leq \varepsilon$$

$$N = O \left( \frac{\max\{\delta, \lambda\}}{\lambda} \log \left( \frac{\Delta F}{\varepsilon} \right) \right)$$

# Gradient method with relative smoothness and strong convexity and Similarity

$$N = O \left( \frac{\max\{\delta, \lambda\}}{\lambda} \log \left( \frac{\Delta F}{\varepsilon} \right) \right) = O \left( \frac{\delta}{\lambda} \log \left( \frac{\Delta F}{\varepsilon} \right) \right)$$

*Warning: Unfortunately, this rate is not optimal (accelerated)!*

*But! Accelerated method with relative smoothness and strong convexity is in principle impossible in general set up.*

Full Length Paper | [Published: 21 April 2021](#)

## Optimal complexity and certification of Bregman first-order methods

[Radu-Alexandru Dragomir](#) , [Adrien B. Taylor](#), [Alexandre d'Aspremont](#) & [Jérôme Bolte](#)

[Mathematical Programming](#) (2021) | [Cite this article](#)

**169** Accesses | **0** Altmetric | [Metrics](#)

# Statistical Similarity

$$\min_{x \in Q} F(x) := \frac{1}{M} \sum_{k=1}^M \left( \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i}) \right)$$

This problem has specific structure

$$F_k(x) = \frac{1}{r} \sum_{i=1}^r f(x, \xi^{k,i})$$

$F_k(x)$  has sum-type structure

Assumptions:

$f(x, \xi)$  is  $L$ -smooth in  $x$  for all  $\xi \Rightarrow \|\nabla^2 F_k(x)\| \leq L$

$f(x, \xi)$  is  $\lambda$ -strongly convex

Communication rounds

$$O\left(\sqrt{\frac{L}{\lambda}} \log\left(\frac{LR^2}{\varepsilon}\right)\right) \quad \text{Variance reduction}$$

$$O\left(\sqrt{\frac{\delta}{\lambda}} \log\left(\frac{LR^2}{\varepsilon}\right)\right) \quad \begin{array}{l} \|\nabla^2 F_k(x) - \nabla^2 F(x)\| \leq \delta \\ \delta\text{-Similarity} \end{array}$$

Is this bound tight? That is, can we propose such algorithm that works according to this bound? What is a lower bound?

The answer is - this is the lower bound (up to a log-factors), but is this bound tight or not? - until 2022 it was an open question!

# Scheme of the prove

Lower bound (Arjevani-Shamir, 2015):

$$\Omega \left( \sqrt{\frac{\delta}{\lambda}} \log \left( \frac{\lambda R^2}{\varepsilon} \right) \right)$$

Upper bound (Optimal Sliding, 2022):

$$O \left( \sqrt{\frac{\delta}{\lambda}} \log \left( \frac{\lambda R^2}{\varepsilon} \right) \right)$$

Kovalev, D., Beznosikov, A., Borodich, E., Gasnikov, A., & Scutari, G. (2022). Optimal Gradient Sliding and its Application to Distributed Optimization Under Similarity. arXiv preprint arXiv:2205.15136.

$$\min_{x \in Q} \left[ \bar{f}(x) := \frac{1}{m} \sum_{k=1}^m \bar{f}_k(x) := \frac{1}{m} \sum_{k=1}^m \frac{1}{s} \sum_{j=1}^s f(x, \xi^{k,j}) \right].$$

To use similarity we describe **Accelerated gradient sliding** for unconstrained composite optimization problem:

$$\min_{x \in \mathbb{R}^n} [\bar{f}(x) := g(x) + h(x)],$$

where  $g(x)$  has  $L_g$ -Lipschitz continuous gradient,  $h(x)$  is convex and has  $L_h$ -Lipschitz continuous gradient ( $L_g \leq L_h$ );  $\bar{f}(x)$  is  $\mu$ -strongly convex function in 2-norm. Note that we do not assume  $g(x)$  to be convex!

# Optimal Sliding Algorithm

$$\min_{x \in \mathbb{R}^n} [\bar{f}(x) := g(x) + h(x)]$$

$$\tilde{x}^t = \tau x^t + (1 - \tau)x_f^t,$$

$$x_f^{t+1} \approx \operatorname{argmin}_{x \in \mathbb{R}^n} [A^t(x) := g(\tilde{x}^t) + \langle \nabla g(\tilde{x}^t), x - \tilde{x}^t \rangle + L_g \|x - \tilde{x}^t\|_2^2 + h(x)],$$

which means

$$\|\nabla A^t(x_f^{t+1})\|_2^2 \leq \frac{L_g^2}{3} \left\| \tilde{x}^t - \arg \min_{x \in \mathbb{R}^n} A^t(x) \right\|_2^2,$$

$$x^{t+1} = x^t + \eta \mu (x_f^{t+1} - x^t) - \eta \nabla \bar{f}(x_f^{t+1}),$$

where

$$\tau = \min \left\{ 1, \frac{\sqrt{\mu}}{2\sqrt{L_g}} \right\}, \quad \eta = \min \left\{ \frac{1}{2\mu}, \frac{1}{2\sqrt{\mu}L_g} \right\}.$$

# Properties of the Algorithm

$$\min_{x \in \mathbb{R}^n} [\bar{f}(x) := g(x) + h(x)]$$

This algorithm (with output point  $x^N$ ) has an iteration complexity

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right)$$

and solves several tasks at once:

- **(simple acceleration)** If  $h(x) \equiv 0$  this algorithm becomes an ordinary accelerated method with

$$x_f^{t+1} = \tilde{x}^t - \frac{1}{2L_p} \nabla g(\tilde{x}^t);$$

- **(Catalyst)** If  $g(x) \equiv 0$  this algorithm becomes a Catalyst-type proximal envelop [72], but less sensitive to the accuracy of the solution of (1.59)

$$x_f^{t+1} \approx \operatorname{argmin}_{x \in \mathbb{R}^n} [A^t(x) := g(\tilde{x}^t) + \langle \nabla g(\tilde{x}^t), x - \tilde{x}^t \rangle + L_g \|x - \tilde{x}^t\|_2^2 + h(x)], \quad (1.59)$$



# Properties of the Algorithm

$$\min_{x \in \mathbb{R}^n} [\bar{f}(x) := g(x) + h(x)]$$

$$x_f^{t+1} \approx \operatorname{argmin}_{x \in \mathbb{R}^n} [A^t(x) := g(\tilde{x}^t) + \langle \nabla g(\tilde{x}^t), x - \tilde{x}^t \rangle + L_g \|x - \tilde{x}^t\|_2^2 + h(x)], \quad (1.59)$$

- **(Sliding)** If we apply to (1.59) Accelerated gradient sliding with  $g(x) := h(x)$  then obtain the total complexity of  $\nabla h(x)$  oracle as

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right) \cdot O\left(\sqrt{\frac{L_g + L_h}{L_g}}\right) = \tilde{O}\left(\sqrt{\frac{L_h}{\mu}}\right).$$

That is, we have split the complexity of considered composite problem to the complexities correspond to the separate problems:

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right) \quad \text{for } \# \nabla g(x) \quad \text{and} \quad \tilde{O}\left(\sqrt{\frac{L_h}{\mu}}\right) \quad \text{for } \# \nabla h(x).$$

# Main idea

$$\min_{x \in Q} \left[ \bar{f}(x) := \frac{1}{m} \sum_{k=1}^m \bar{f}_k(x) := \frac{1}{m} \sum_{k=1}^m \frac{1}{s} \sum_{j=1}^s f(x, \xi^{k,j}) \right].$$

Let us rewrite the empirical problem as follows

$$\min_{x \in \mathbb{R}^n} [\bar{f}(x) := (\bar{f}(x) - \bar{f}_1(x)) + \bar{f}_1(x)].$$

That is, we have split the complexity of considered composite problem to the complexities correspond to the separate problems:

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right) \quad \text{for } \#\nabla g(x) \quad \text{and} \quad \tilde{O}\left(\sqrt{\frac{L_h}{\mu}}\right) \quad \text{for } \#\nabla h(x).$$

Denoting the first sum as  $g(x)$  and the second one as  $h(x)$  we can use Sliding trick to split the complexities. Note that we significantly use the fact that in this scheme  $g(x)$  is not necessarily convex! So it remains only to notice that described Accelerated gradient sliding under this choice of  $g(x)$  and  $h(x)$  has a natural distributed interpretation. It gives at the end a distributed algorithm that works according to the lower bounds for communications and oracle calls per node complexities under similarity [7]. Due to the statistical (i.i.d.) nature of  $\{\xi^{k,j}\}$  (statistical similarity) one may expect that [51]:  $L_g \propto s^{-1/2}$ .

# Distributed interpretation

$$\min_{x \in Q} \left[ \bar{f}(x) := \frac{1}{m} \sum_{k=1}^m \bar{f}_k(x) := \frac{1}{m} \sum_{k=1}^m \frac{1}{s} \sum_{j=1}^s f(x, \xi^{k,j}) \right].$$

Let us rewrite the empirical problem as follows

$$\min_{x \in \mathbb{R}^n} [\bar{f}(x) := (\bar{f}(x) - \bar{f}_1(x)) + \bar{f}_1(x)].$$

That is, we have split the complexity of considered composite problem to the complexities correspond to the separate problems:

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right) \quad \text{for } \#\nabla g(x) \quad \text{and} \quad \tilde{O}\left(\sqrt{\frac{L_h}{\mu}}\right) \quad \text{for } \#\nabla h(x).$$

Indeed, we can assign the node number 1 to be a master node that minimize at each iteration (1.59) with  $g(x) := \bar{f}(x) - \bar{f}_1(x)$  and  $h(x) := \bar{f}_1(x)$ . It is obvious, that  $h(x)$  is available to the master node and  $\nabla g(\tilde{x}^t)$  can be available due to communications of the master node with the other ones. At each round of communications  $k$ -th node sends  $\nabla \bar{f}_k(\tilde{x}^t)$  to the master node and receive in return  $x_f^{t+1}$ , which is calculated at the master node.

$$x_f^{t+1} \approx \operatorname{argmin}_{x \in \mathbb{R}^n} [A^t(x) := g(\tilde{x}^t) + \langle \nabla g(\tilde{x}^t), x - \tilde{x}^t \rangle + L_g \|x - \tilde{x}^t\|_2^2 + h(x)], \quad (1.59)$$

# Variational inequalities in the general prox-setup

We are particularly interested in constrained VI problem formulated as follows

$$\text{find } z^* \in \mathcal{Z} : \quad \langle F(z^*), z - z^* \rangle \geq 0 \quad \forall z \in \mathcal{Z}, \quad (1)$$

where  $\mathcal{Z} \subseteq \mathbb{R}^d$  is a convex set, and  $F : \mathcal{Z} \rightarrow \mathbb{R}^d$  is a monotone and  $L$ -Lipschitz operator. In this paper, we consider the operator  $F$  of the form

$$F(z) := \frac{1}{m} \sum_{i=1}^m F_i(z), \quad (2)$$

where each  $F_i$  is accessed only by machine  $i$  ( $i = 1, \dots, m$ ) locally.

# Variational inequalities in the general prox-setup

**Notation.** For vectors,  $\|z\|$  is a general norm on space  $\mathcal{Z}$ , and  $\|s\|_*$  is its dual norm on the dual space  $\mathcal{Z}^*$ :  $\|s\|_* = \max_{z \in \mathcal{Z}} \{\langle s, z \rangle : \|z\| = 1\}$ . For matrices,  $\|A\|$  is the matrix norm induced by vector norm  $\|z\|$ :  $\|A\| = \sup_{z \in \mathcal{Z}} \{\|Az\| : \|z\| = 1\}$ .

**Definition 2.1** *We say that  $w : \mathcal{Z} \rightarrow \mathbb{R}$  is a distance generating function (DGF), if  $w$  is 1-strongly convex with respect to  $\|\cdot\|$ , i.e., for all  $u, v \in \mathcal{Z}$ :  $w(v) \geq w(u) + \langle \nabla w(u), v - u \rangle + \frac{1}{2}\|v - u\|^2$ . The corresponding Bregman divergence is*

$$V(u, v) = w(u) - w(v) - \langle \nabla w(v), u - v \rangle, \quad \forall u, v \in \mathcal{Z}.$$

The property of distance generating function ensures  $V(u, u) = 0$  and  $V(u, v) \geq \frac{1}{2}\|u - v\|^2$ .

We will require the fulfilment of the next three assumptions to provide the convergence guarantee for our first algorithm.

# Variational inequalities in the general prox-setup

**Assumption 2.2 (Monotonicity)** *The operator  $F$  is a monotone, i.e. for all  $u, v \in \mathcal{Z}$  :*

$$\langle F(u) - F(v), u - v \rangle \geq 0.$$

A special case of monotone VIs are convex optimization problems and convex-concave SPPs.

**Assumption 2.3 (Lipschitzness)** *The operator  $F$  is  $L$ -Lipschitz continuous, i.e. for all  $u, v \in \mathcal{Z}$  :*

$$\|F(u) - F(v)\|_* \leq L\|u - v\|.$$

# Variational inequalities in the general prox-setup

$$\text{find } z^* \in \mathcal{Z} : \quad \langle F(z^*), z - z^* \rangle \geq 0 \quad \forall z \in Z, \quad (1)$$

**Assumption 2.4 ( $\delta$ -similarity)** *Operator  $F_1$  is  $\delta$ -related, that is operator  $F_1 - F$  is  $\delta$ -Lipschitz continuous, i.e., for all  $u, v \in \mathcal{Z}$  :*

$$\|F_1(u) - F(u) - F_1(v) + F(v)\|_* \leq \delta \|u - v\|.$$



# Variational inequalities in the general prox-setup

---

**Algorithm 1** PAUS

---

**Input:** parameter of similarity  $\delta$ , stepsize  $\gamma \leq 1/\delta$ ,  
number of iterations  $K$ , starting points  $z^0 = u^0 \in \mathcal{Z}$ .

- 1: **for**  $k = 0, 1, 2, \dots, K - 1$  **do**
  - 2:   All devices in parallel compute  $F_i(z^k)$  and  $F_i(u^k)$   
and send them to server ( $i = 1, \dots, m$ ).
  - 3:   Server computes  $F(z^k) = \frac{1}{m} \sum_{i=1}^m F_i(z^k)$   
and  $F(u^k) = \frac{1}{m} \sum_{i=1}^m F_i(u^k)$
  - 4:   Server solves
    - for VIs: find  $u^k$  as a solution to  
 $\langle \gamma(F_1(u^k) + F(z^k) - F_1(z^k))$   
 $+ \nabla w(u^k) - \nabla w(z^k), z - u^k \rangle \geq 0$  for all  $z \in \mathcal{Z}$ .
  - 5:   Server solves  $z^{k+1} \leftarrow \arg \min_{z \in \mathcal{Z}} \{ \gamma \langle F(u^k)$   
 $- F_1(u^k) - F(z^k) + F_1(z^k), z \rangle + V(z, u^k) \}$ .
  - 6:   Server broadcasts  $u^k$  and  $z^{k+1}$  to devices.
  - 7: **end for**
  - 8: **return**  $\tilde{u}^K = \frac{1}{K} \sum_{k=0}^{K-1} u^k$ .
-

# Variational inequalities in the general prox-setup

**Theorem 3.2** *Let Assumptions 2.2, 2.3 and 2.4 hold. Then after  $K$  communication rounds, PAUS (Algorithm 1), run with stepsize  $\gamma$  and a starting point  $z^0 \in \mathcal{Z}$ , outputs  $\tilde{u}^K$  such that*

$$\text{Gap}(\tilde{u}^K) \leq \frac{\max_{z \in \mathcal{Z}} V(z, z^0)}{K\gamma}.$$

$$\text{find } z^* \in \mathcal{Z} : \quad \langle F(z^*), z - z^* \rangle \geq 0 \quad \forall z \in \mathcal{Z}, \quad (1)$$

$$\text{Gap}(u) = \max_{z \in \mathcal{Z}} \{ \langle F(z), u - z \rangle \}.$$

# Variational inequalities in the general prox-setup

**Corollary 3.3** *Let Assumptions 2.2, 2.3 and 2.4 hold. Let  $\tilde{u}^K$  be the output after*

$$K = \frac{\delta}{\epsilon} \max_{z \in \mathcal{Z}} V(z, z^0)$$

*communication rounds of PAUS (Algorithm 1), run with stepsize  $\gamma = 1/\delta$  and a starting point  $z^0 \in \mathcal{Z}$ . Then  $\text{Gap}(\tilde{u}^K) \leq \epsilon$ .*

# Variational inequalities in the general prox-setup

**Inner problems in line 4 of Algorithm 1.** To solve the problems in line 4 of PAUS (Algorithm 1), we provide a procedure which we refer to as COMPOSITE MP. In the next theorem, we give its convergence guarantee. In what follows, we comment on the existence of closed-form solutions of inner problems of the COMPOSITE MP procedure, as e.g. in the Entropy setup.

---

```
1: procedure COMPOSITE MP( $\gamma, L_{F_1}, z^k$ )
2:   Choose stepsize  $\eta \leftarrow \frac{1}{2\gamma L_{F_1}}$ .
3:   Choose starting point  $v^0 \in \mathcal{Z}$ .
4:   for  $t = 0, 1, 2, \dots, T - 1$  do
5:     Server solves  $v^{t+\frac{1}{2}} \leftarrow \arg \min_{v \in \mathcal{Z}} \{ \gamma \eta \langle F_1(v^t) + F(z^k) - F_1(z^k), v \rangle + \eta V(v, z^k) + V(v, v^t) \}$ .
6:     Server solves  $v^{t+1} \leftarrow \arg \min_{v \in \mathcal{Z}} \{ \gamma \eta \langle F_1(v^{t+\frac{1}{2}}) + F(z^k) - F_1(z^k), v \rangle + \eta V(v, z^k) + V(v, v^t) \}$ .
7:   end for
8:   return  $v^T$ .
9: end procedure
```

---

# Variational inequalities in the general prox-setup

**Theorem 3.6** *Let Assumptions 2.2, 2.3 and 2.4 hold. Let  $v^*$  be a solution of the inner problem in line 4 (for SPPs or VIs) of Algorithm 1, and  $L_{F_1}$  be the Lipschitz constant for  $F_1$ . Let  $v^T$  be the output after*

$$T = \frac{3L_{F_1}}{\delta} \log \frac{V(v^*, v^0)}{\epsilon}.$$

*iterations of COMPOSITE MP procedure, run with step-size  $\gamma = 1/\delta$  and starting point  $v^0$ . Then  $V(v^T, v^*) \leq \epsilon$ .*

# Experiments

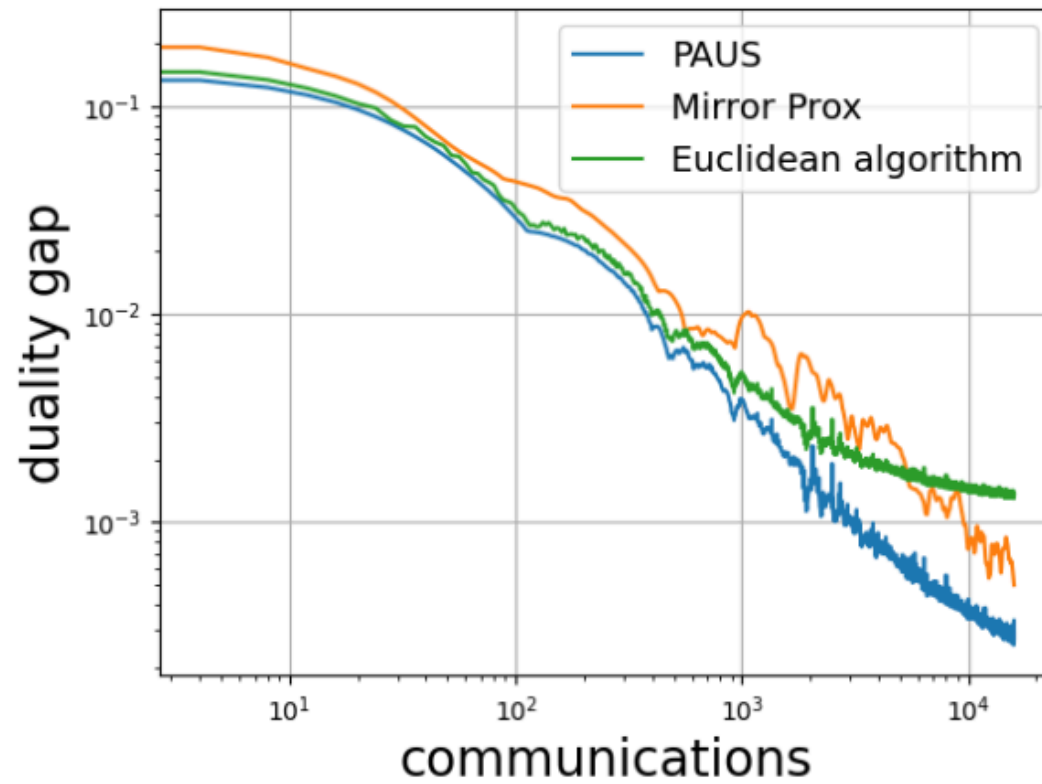


Figure 1: Comparison of PAUS with distributed Mirror Prox without similarity ([Rogozin et al., 2021](#)) and the Euclidean algorithm under similarity ([Kovalev et al., 2022](#))