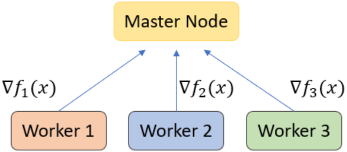
On Stochastic Variation of Optimal Gradient Sliding Algorithm for Strongly Convex Case

Aleksandr Beznosikov, Vladimir Smirnov

May 23, 2023

Data-parallelism

- Only gradients are communicated
- Whole model on each device



- $\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$
- $x^{k+1} = x^k - \frac{\eta^k}{n} \sum_{i=1}^{n} \nabla f_i(x)$

Faster Optimization Using a Cheaper Proxy

- Consider a distributed optimization problem over a network of n agents:

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

Faster Optimization Using a Cheaper Proxy

- Consider a distributed optimization problem over a network of n agents:

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

- Reformulate the problem in the following form:

$$\min_{x \in \mathbb{R}^d} r(x) := q(x) + p(x)$$

Faster Optimization Using a Cheaper Proxy

- Consider a distributed optimization problem over a network of n agents:

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

- Reformulate the problem in the following form:

$$\min_{x \in \mathbb{R}^d} r(x) := q(x) + p(x)$$

- In the case of n agent network, we get:

$$\min_{x \in \mathbb{R}^d} r(x) = \underbrace{f_1(x)}_{:=q(x)} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} [f_i(x) - f_1(x)]}_{:=p(x)}$$

For more information, see: [4], [11]

Convexity and Lipshitz gradients bounds

Let's further assume the following conditions for given partition:

$$\min_{x \in \mathbb{R}^d} r(x) := q(x) + p(x)$$

Assumption (1)

$r(x)\colon \mathbb{R}^d \to \mathbb{R}$ *is* μ*-strongly convex on* $\mathbb{R}^d$

Convexity and Lipshitz gradients bounds

Let's further assume the following conditions for given partition:

$$\min_{x \in \mathbb{R}^d} r(x) := q(x) + p(x)$$

Assumption (1)

$r(x)\colon \mathbb{R}^d \to \mathbb{R}$ *is* μ*-strongly convex on* $\mathbb{R}^d$

Assumption (2)

$q(x)\colon \mathbb{R}^d \to \mathbb{R}$ *is convex and* L_q*-smooth on* $\mathbb{R}^d$

Convexity and Lipshitz gradients bounds

Let's further assume the following conditions for given partition:

$$\min_{x \in \mathbb{R}^d} r(x) := q(x) + p(x)$$

Assumption (1)

$r(x)\colon \mathbb{R}^d \to \mathbb{R}$ *is* μ*-strongly convex on* $\mathbb{R}^d$

Assumption (2)

$q(x)\colon \mathbb{R}^d \to \mathbb{R}$ *is convex and* L_q*-smooth on* $\mathbb{R}^d$

Assumption (3)

$p(x)\colon \mathbb{R}^d \to \mathbb{R}$ *is* L_p*-smooth on* $\mathbb{R}^d$

Original Algorithm

Algorithm Accelerated Extragradient

Input: $x^0 = x_f^0 \in \mathbb{R}^d$
Parameters: $\tau \in (0,1]$, $\eta,\theta,\alpha > 0$, $K \in \{1,2,\dots\}$
for $k = 0,1,2,\dots,K-1$ **do**
    $x_g^k = \tau x^k + (1-\tau)x_f^k$
    $x_f^{k+1} \approx \arg\min_{x\in\mathbb{R}^d} \left[A_\theta^k(x) := p(x_g^k) + \langle \nabla p(x_g^k), x - x_g^k\rangle + \frac{1}{2\theta}\|x - x_g^k\|^2 + q(x)\right]$
    $x^{k+1} = x^k + \eta\alpha(x_f^{k+1} - x^k) - \eta\nabla r(x_f^{k+1})$
end for
Output: x^K

Algorithm source: [4]

Additional Convergence Conditions

- $\min_{x\in\mathbb{R}^d} r(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$
- $\min_{x\in\mathbb{R}^d} r(x) = \underbrace{f_1(x)}_{:=q(x)} + \underbrace{\frac{1}{n}\sum_{i=1}^n [f_i(x) - f_1(x)]}_{:=p(x)}$
- $r(x)\colon \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex.

Additional Convergence Conditions

- $\min_{x\in\mathbb{R}^d} r(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$
- $$\min_{x\in\mathbb{R}^d} r(x) = \underbrace{f_1(x)}_{:=q(x)} + \underbrace{\frac{1}{n}\sum_{i=1}^n [f_i(x) - f_1(x)]}_{:=p(x)}$$
- $r(x)\colon \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex.
- Each $f_i(x)\colon \mathbb{R}^d \to \mathbb{R}$ is convex and L-smooth.

Additional Convergence Conditions

- $\min_{x\in\mathbb{R}^d} r(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$
- $\min_{x\in\mathbb{R}^d} r(x) = \underbrace{f_1(x)}_{:=q(x)} + \underbrace{\frac{1}{n}\sum_{i=1}^n [f_i(x) - f_1(x)]}_{:=p(x)}$
- $r(x)\colon \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex.
- Each $f_i(x)\colon \mathbb{R}^d \to \mathbb{R}$ is convex and L-smooth.
- $f_1(x), \ldots, f_n(x)$ are δ-related: $\|\nabla^2 f_i(x) - \nabla^2 f_j(x)\| \leq \delta$, for all $i \neq j$ and $x \in \mathbb{R}^d$, and some $\delta > 0$.

Quality (Upper and Lower Bounds) of the Original Algorithm

		Reference	Communication complexity	Local gradient complexity	Order	Limitations
Minimization	Upper	DANE [8]	$\mathcal{O}\left(\frac{\delta^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\sqrt{\frac{\delta^3}{\mu^3}}\log^2\frac{1}{\varepsilon}\right)$ (2)	1st	quadratic
		DANE-HB [12]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$ (5)	1st/2nd	quadratic (6)
		SONATA [10]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\sqrt{\frac{\delta}{\mu}}\log^2\frac{1}{\varepsilon}\right)$ (2)	1st	decentralized
		SPAG [3]	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$ (1)	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\sqrt{\frac{L}{\delta}}\log^2\frac{1}{\varepsilon}\right)$ (1,2)	1st	M - Lipshitz hessian
		Acc. ExtraGD	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	1st	
	Lower	[1]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$	—		
		[6]	—	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$		non-distributed
Saddles	Upper	SMMDSA [2]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\log\frac{L}{\mu}\right)$	1st	
		Acc. ExtraGD	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$	1st	
	Lower	[2]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	—		
		[7]	-	$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$		non-distributed

Table 1: Existing convergence results for distributed (saddle point) optimization under δ-similarity. *Notation:* δ = similarity parameter, L=smoothness constant of f_i, μ = strong convexity constant of r, ε =accuracy of the solution.

Table source: [4]

Stochastic Algorithm

Algorithm Accelerated Stochastic Extragradient

Input: $x^0 = x_f^0 \in \mathbb{R}^d$
Parameters: $\tau \in (0,1]$, $\eta,\theta,\alpha > 0$, $K \in \{1,2,\ldots\}$
for $k = 0,1,2,\ldots,K-1$ **do**
  $x_g^k = \tau x^k + (1-\tau)x_f^k$
  $x_f^{k+1} \approx \arg\min_{x\in\mathbb{R}^d} \left[\bar{A}_\theta^k(x) := p(x_g^k) + \langle s_k, x - x_g^k\rangle + \frac{1}{2\theta}\|x - x_g^k\|^2 + q(x)\right]$
  $x^{k+1} = x^k + \eta\alpha(x_f^{k+1} - x^k) - \eta t_k$
end for
Output: x^K

Stochastic Gradient Limitations

Assumption (4, Unbiased p gradient oracle)

Almost surely,

$$\mathbb{E}[s_k|\mathcal{F}_k] = \nabla p(x_g^k), \quad \mathbb{E}[\|s_k - p(x_g^k)\|^2|\mathcal{F}_k] \leq \sigma_1^2, \quad \forall k \in \mathbb{N}$$

Stochastic Gradient Limitations

Assumption (4, Unbiased p gradient oracle)

Almost surely,

$$\mathbb{E}[s_k|\mathcal{F}_k] = \nabla p(x_g^k), \quad \mathbb{E}[\|s_k - p(x_g^k)\|^2|\mathcal{F}_k] \leq \sigma_1^2, \quad \forall k \in \mathbb{N}$$

Assumption (5, Unbiased r gradient oracle)

Almost surely,

$$\mathbb{E}[t_k|\mathcal{F}_k] = \nabla r(x_f^{k+1}), \quad \mathbb{E}[\|t_k - r(x_f^{k+1})\|^2|\mathcal{F}_k] \leq \sigma_2^2, \quad \forall k \in \mathbb{N}$$

More examples: [9], [5]

Original Algorithm Convergence

Theorem

Consider Algorithm 2 under previous Assumptions with the following tuning:

$$\tau = \min\left\{1, \frac{\sqrt{\mu}}{3\sqrt{L_p}}\right\}, \quad \theta = \frac{1}{3L_p}, \quad \eta = \min\left\{\frac{1}{3\mu}, \frac{1}{3\sqrt{\mu L_p}}\right\}, \quad \alpha = \mu;$$

and let x_f^{k+1} *in line 2 satisfy*

$$\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 \leq \frac{9L_p^2}{11}\|x_g^k - \arg\min_{x\in\mathbb{R}^d} \bar{A}_\theta^k(x)\|^2.$$

Then, for any

$$K \geq 3\max\left\{1, \sqrt{\frac{L_p}{\mu}}\right\}\log\frac{\|x^0 - x^*\| + \frac{2\eta}{\tau}[r(x^0) - r(x^*)]}{\varepsilon},$$

we have the following estimate for the distance to the solution x^**:*

$$\|x^K - x^*\| \leq \varepsilon + \frac{3K\eta\sigma_1^2}{\tau} + 3\eta^2 K\sigma_2^2.$$

Methods

- The server computes x_g^k and sends it to all the workers. Workers compute $\nabla f_i(x_g^k)$ and send it to the server. After collecting all $\nabla f_i(x_g^k)$, the server builds $\nabla p(x_g^k) = \nabla r(x_g^k) - \nabla f_1(x_g^k)$, and then solves the local problem A_θ^k. The solution x_f^{k+1} is then broadcast to the workers, which update their own receives $\nabla f_i(x_f^{k+1})$ and send back to the server, which can then evaluate $\nabla r(x_f^{k+1})$
- Ridge Regression problem:

$$\min_{w \in \mathbb{R}^d} \left[\frac{1}{2N} \sum_{i=1}^{N} (w^T x_i - y_i)^2 + \frac{\lambda}{2} \|w\|^2 \right]$$

where w is the vector of weights of the model, $\{x_i, y_i\}_{i=1}^N$ is the training dataset, and $\lambda > 0$ is the regularization parameter.

Results

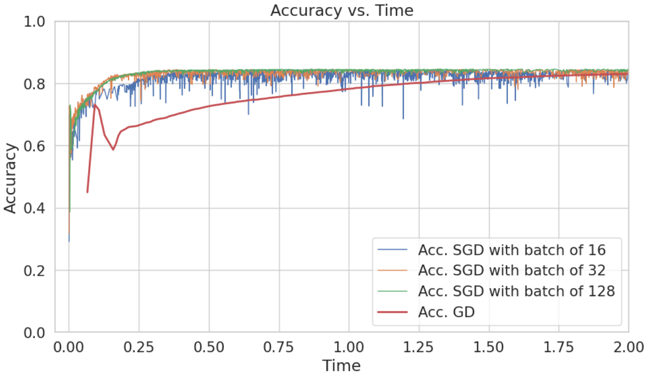


Figure: Accuracy growth in comparison with the Accelerated Extragradient in the case of an exact solution of the intermediate problem.

Results

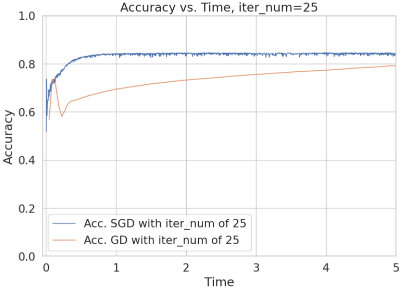


Figure: Accuracy growth in comparison with the Accelerated Extragradient in the case of an inexact(iterative) solution of the intermediate problem.

Results

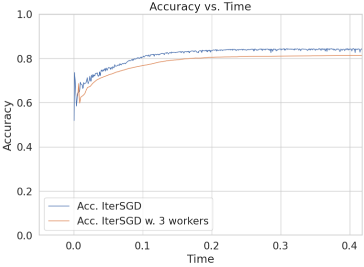


Figure: Mean Results from Multiple Workers vs Iterative Stochastic Algorithm on a Single Master Worker

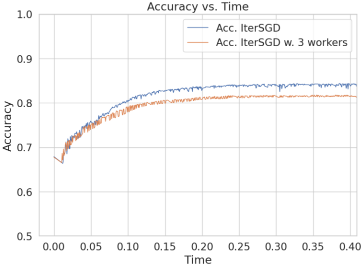


Figure: First Acquired Result from Multiple Workers vs Iterative Stochastic Algorithm on a Single Master Worker

Future Work

- Further Assessment of Algorithm Convergence Rate in the case of Inaccurate Intermediate Solution
- Try the algorithm on a distributed system
- Try to apply on NNs

References

Yossi Arjevani and Ohad Shamir.
Communication complexity of distributed convex learning and optimization.
Advances in neural information processing systems, 28, 2015.

Aleksandr Beznosikov, Gesualdo Scutari, Alexander Rogozin, and Alexander Gasnikov.
Distributed saddle-point problems under data similarity.
Advances in Neural Information Processing Systems, 34, 2021.

Hadrien Hendrikx, Lin Xiao, Sebastien Bubeck, Francis Bach, and Laurent Massoulie.
Statistically preconditioned accelerated gradient method for distributed optimization.
In *International Conference on Machine Learning*, pages 4203–4227. PMLR, 2020.

Dmitry Kovalev, Aleksandr Beznosikov, Ekaterina Borodich, Alexander Gasnikov, and Gesualdo Scutari.
Optimal gradient sliding and its application to distributed optimization under similarity.