AsGrad: A Sharp Unified Analysis of Asynchronous-SGD Algorithms

Rustem Islamov[1,2,3] Mher Safaryan[2] Dan Alistarh[2]

[1]University of Basel [2]Insitute of Science and Technology Austria (ISTA) [3]Insitut Polytechnique de Paris (IP Paris)

Problem Formulation

We want to solve the finite-sum optimization problem

$$\min_{x\in\mathbb{R}^d}\left\{f(x)\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^{n}f_i(x)\right\}$$

$f(x)\geq f^*$ for all $x\in\mathbb{R}^d$ → Lower bounded non-convex

clients/devices

model parameters

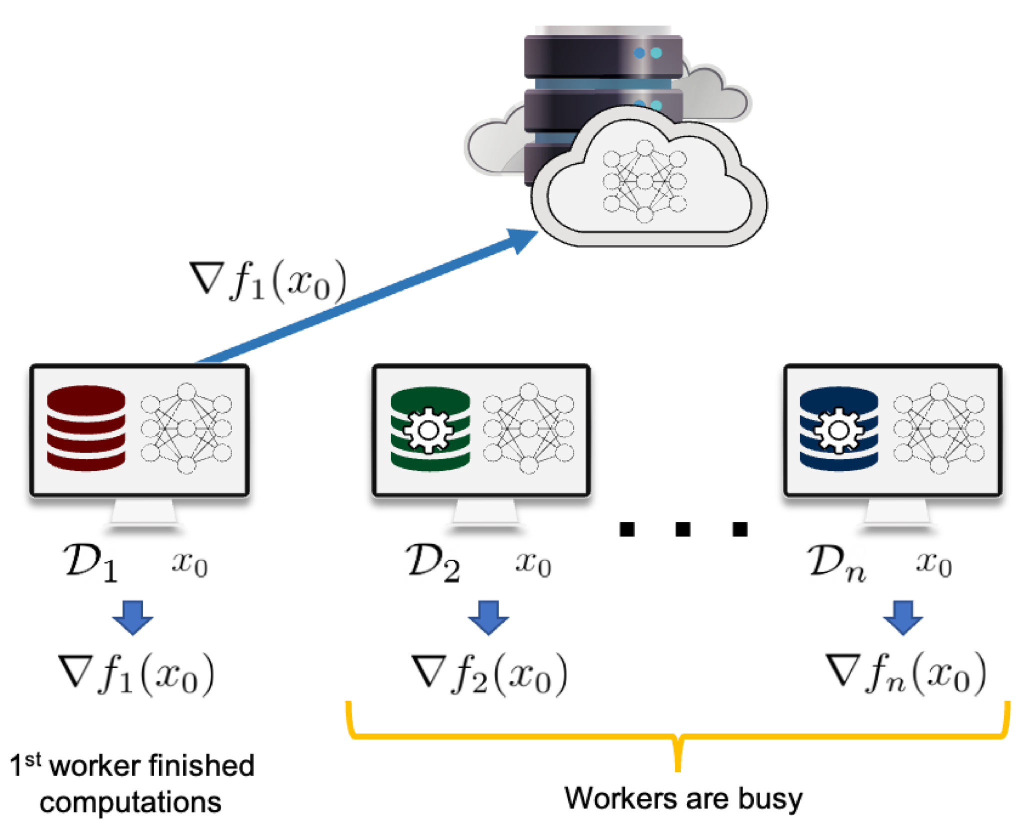
Empirical risk/loss

Local training data

Local loss function $f_i(x)=\mathbb{E}_{\xi\sim\mathcal{D}_i}[f_\xi(x)]$

- This problem has many applications in machine learning, data science and engineering.
- We focus on the regime when n **and** d **are very large**. This is typically the case in the big data settings (e.g., massively distributed and federated learning).

Asynchronous Communication



The source of asynchrony might be:

- Workers may have different computation powers or communication channels.
- Message-passing failures.
- Workers might be inactive.

Why we need asynchronous communication:

- *Synchronized* communication may drastically slow down the training if workers' computation powers significantly differ from each other.
- Asynchronous communication decreases *communication bottleneck.*

Main Contributions

- Unified framework, AsGrad, to analyze asynchronous SGD-type methods.
- As a byproduct of the analysis, we design and analyze a new asynchronous method, called *shuffled asynchronous SGD*, which can outperform existing ones both theoretically and practically.
- Our framework recovers popular synchronous variants of SGD with the best-known convergence guarantees.
- All of our results have better or similar dependencies on the maximum delay. we remove entirely dependencies on maximum delay used by prior works.

Table 1:Asynchronous algorithms whose convergence analysis is covered by our framework. For shuffled asynchronous SGD $\tau_C=n$. **BG** = requires **B**ounded **G**radients.

Method	Algorithm	Citation	BG	Rate (a)
Pure Asynchronous SGD	$(k_{t+1},\alpha_{t+1})=(i_t,t+1)$	[1]	No	$\frac{\tau_C}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\zeta^2$ (b)
		Ours	No	$\frac{\sqrt{\tau_{\max}\tau_C}}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\zeta^2$
		Ours	Yes	$\frac{\tau_C}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\left(\frac{G\tau_C}{T}\right)^{2/3}+\zeta^2$
Pure Asynchronous SGD with waiting	$(k_{t+1},\alpha_{t+1})=\left(i_t,\lfloor\frac{t+1}{b}\rfloor b\right)$	Ours	No	$\frac{\sqrt{\tau_{\max}\tau_C}}{T\sqrt{b}}+\left(\frac{\sigma^2}{Tb}\right)^{1/2}+\zeta^2$
		Ours	Yes	$\frac{LF_0\tau_C}{Tb}+\left(\frac{\sigma^2}{Tb}\right)^{1/2}+\left(\frac{G\tau_C}{Tb}\right)^{2/3}+\zeta^2$
Random Asynchronous SGD	$k_{t+1}\sim\mathrm{Unif}[n],\alpha_{t+1}=t+1$	[2]	No	$\frac{LF_0\sqrt{\tau_{\max}\tau_C}}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\left(\frac{\zeta^2}{T}\right)^{1/2}+\left(\frac{\tau_C\zeta}{T}\right)^{2/3}$
		[2]	Yes	$\frac{\tau_C}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\left(\frac{\zeta^2}{T}\right)^{1/2}+\left(\frac{\tau_C G}{T}\right)^{2/3}$
		Ours	Yes	$\frac{\tau_C}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\left(\frac{\zeta^2}{T}\right)^{1/2}+\left(\frac{\tau_C G}{T}\right)^{2/3}$
Random Asynchronous SGD with waiting (FedBuff)	$k_{t+1}\sim\mathrm{Unif}[n],\alpha_{t+1}=\lfloor\frac{t+1}{b}\rfloor$	[3]	Yes	$\frac{1}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\left(\frac{\zeta\tau_{\max}}{T}\right)^{2/3}+\left(\frac{G\tau_{\max}}{T}\right)^{2/3}$ (c)
		Ours	Yes	$\frac{\tau_C}{T}+\left(\frac{\zeta^2}{Tb}\right)^{1/2}+\left(\frac{\sigma^2}{Tb}\right)^{1/2}+\left(\frac{\tau_C G}{Tb}\right)^{2/3}$
Shuffled Asynchronous SGD **[NEW]**	$k_{t+1}=\chi(j),\alpha_{t+1}=t+1$; $j-1=t(\mathrm{mod}\ n)$; χ is a permutation of $[n]$	Ours	Yes	$\frac{n}{T}+\left(\frac{\sigma^2}{T}\right)^{1/2}+\left(\frac{\sqrt{n}\zeta}{T}\right)^{2/3}+\left(\frac{Gn}{T}\right)^{2/3}$

(a) We present the best-known rates under the same set of assumptions as we use in the analysis in $\mathcal{O}$-notation.
(b) [1] uses delay adaptive stepsizes to get rid of the dependency on $\tau_{\max}$.
(c) If we set $\eta_l=\frac{\gamma}{b},\eta_g=b,Q=1$ in Theorem 1 [3]. The analysis is done under the unrealistic assumption that $\{i_t\}_{t=0}^{T-1}$ are distributed uniformly at random.

Assumptions

A1 Smoothness. Each function f_i is L-smooth, namely

$$\|\nabla f_i(x)-\nabla f_i(y)\|\leq L\|x-y\|\quad\forall x,y\in\mathbb{R}^d.$$

A2 Bounded variance. Stochastic gradients $g_i(x):=\nabla f_i(x,\xi)$ are unbiased and have bounded variance, i.e.

$$\mathbb{E}_{\xi\sim\mathcal{D}_i}[\|\nabla f_i(x,\xi)-\nabla f_i(x)\|^2]\leq\sigma^2\quad\forall x\in\mathbb{R}^d.$$

A3 Bounded heterogeneity. Each local gradient $\nabla f_i(x)$ satisfies the bounded heterogeneity condition

$$\|\nabla f_i(x)-\nabla f(x)\|^2\leq\zeta^2,\quad\forall x\in\mathbb{R}^d.$$

For some results, we also need the boundedness of local gradients.

A4 Bounded gradients. Each local gradient $\nabla f_i(x)$ satisfies

$$\|\nabla f_i(x)\|\leq G\quad\forall x\in\mathbb{R}^d.$$

Notation and Convergence Theory

- $\mathcal{A}_{t+1}$ and $\mathcal{R}_t$ sets of **assigned** and **received** jobs at iteration t.
- τ_t (resp. $\widetilde{\tau}_t$) a **delay** of the received (resp. assigned) gradient at iteration t.
- τ_C a **maximum number** of active jobs, i.e.
$$\tau_C:=\max_{0\leq t\leq T}|\mathcal{A}_{t+1}\setminus\mathcal{R}_t|.$$
- ν^2 is a **delay variance** associated with a sequence of indices $\{i_t\}_{t\geq0}$ and defined as
$$\nu:=\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\sum_{j=\pi_t}^{t-1}\nabla f_{i_j}(x_{\pi_j})-\nabla f(x_{\pi_j})\right\|^2\right].$$
- $\tau_{\max}$ (resp. $\widetilde{\tau}_{\max}$) a **maximum delay** of received (resp. assigned) gradients during the training, i.e.
$$\tau_{\max}:=\max\left\{\max_{0\leq t\leq T}\tau_t,\max_{(i,j)\in\mathcal{A}_{T+1}\setminus\mathcal{R}_T}T-j\right\}.$$
- For any given correlation period $\tau\geq1$ we successively split the set of received gradient indices $\{i_t\}_{t\geq0}$ into $\lceil\frac{T}{\tau}\rceil$ chunks of size τ. The **sequence correlation** $\sigma^2_{k\tau}$ within k-th period is defined as
$$\sigma^2_{k\tau}:=\max_{0\leq j<\tau}\mathbb{E}\left[\left\|\sum_{t=k\tau}^{\min\{k\tau+j,T-1\}}\nabla f_{i_t}(x_{k\tau})-\nabla f(x_{k\tau})\right\|^2\right].$$

Theorem 1 (Analysis of gradient receiving process). Let Assumptions A1 and A2 hold. Let the stepsize γ satisfy inequalities $6L\gamma\leq1$ and $20L\gamma\sqrt{\tau_{\max}\tau_C}\leq1$, the correlation period $\tau=\left\lfloor\frac{1}{20L\gamma}\right\rfloor$, and quantities $\{\sigma^2_{k,\tau}\}_{k=0}^{\lfloor T/\tau\rfloor}$ and ν^2 are finite. Then

$$\mathbb{E}\left[\|\nabla f(\hat{x}_T)\|^2\right]\leq\mathcal{O}\left(\frac{1}{\gamma T}+L\gamma\sigma^2+L^2\gamma^2\Phi\right),\quad\Phi:=\frac{1}{\lfloor T/\tau\rfloor}\sum_{k=0}^{\lfloor T/\tau\rfloor}\sigma^2_{k,\tau}+\frac{1}{T}\nu^2.$$

Theorem 2 (Analysis of gradient assigning process). Let Assumptions A1, A2, and A4 hold. Let the stepsize γ satisfies inequalities $6L\gamma\leq1$ and $30L\gamma\max\{\widetilde{\tau}_{\max},\tau_C\}\leq1$, the correlation period $\tau=\left\lfloor\frac{1}{30L\gamma}\right\rfloor$, quantities $\{\widetilde{\sigma}^2_{k,\tau}\}_{k=0}^{\lfloor T/\tau\rfloor}$ and $\widetilde{\nu}^2$ are finite. Then

$$\mathbb{E}\left[\|\nabla f(\hat{x}_T)\|^2\right]\leq\mathcal{O}\left(\frac{1}{\gamma T}+L\gamma\sigma^2+L^2\gamma^2\widetilde{\Phi}+L^2\gamma^2(\tau_C-1)^2G^2\right),\quad\widetilde{\Phi}:=\frac{1}{\lfloor T/\tau\rfloor}\sum_{k=0}^{\lfloor T/\tau\rfloor}\widetilde{\sigma}^2_{k,\tau}+\frac{1}{T}\widetilde{\nu}^2.$$

Algorithm 1: AsGrad framework: General Asynchronous SGD

Input: $x^0\in\mathbb{R}^d$, stepsize $\gamma>0$, set of assigned jobs $\mathcal{A}_0=\emptyset$, set of received jobs $\mathcal{R}_0=\emptyset$

Initialization: for all jobs $(i,0)\in\mathcal{A}_1$, the server assigns worker i to compute a stochastic gradient $g_i(x_0)$

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

 Once worker i_t finishes a job $(i_t,\pi_t)\in\mathcal{A}_{t+1}$, it sends $g_{i_t}(x_{\pi_t})$ to server

 server updates $x_{t+1}=x_t-\gamma g_{i_t}(x_{\pi_t})$ and $\mathcal{R}_{t+1}=\mathcal{R}_t\cup\{(i_t,\pi_t)\}$

 server assigns worker k_{t+1} to compute a gradient $g_{k_{t+1}}(x_{\alpha_{t+1}})$

 server updates the set $\mathcal{A}_{t+2}=\mathcal{A}_{t+1}\cup\{(k_{t+1},\alpha_{t+1})\}$

end

Experiments

We consider Logistic Regression problem with non-convex regularization:

$$\min_{x\in\mathbb{R}^d}\left\{\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\lambda\sum_{j=1}^{d}\frac{x_j^2}{1+x_j^2}\right\},\quad f_i(x)=\frac{1}{m}\sum_{j=1}^{m}\log\left(1+e^{-b_{ij}a_{ij}^\top x}\right).$$

Each worker has a parameter s_i and spends r seconds to compute a gradient according to

- **Fixed:** $r\equiv s_i$
- **Normal:** $r=|s|+1,\ s\sim\mathcal{N}(s_i,s_i)$
- **Poisson:** $r\sim\mathrm{Po}(s_i)$
- **Uniform:** $r\sim\mathrm{Uni}(0,s_i)$

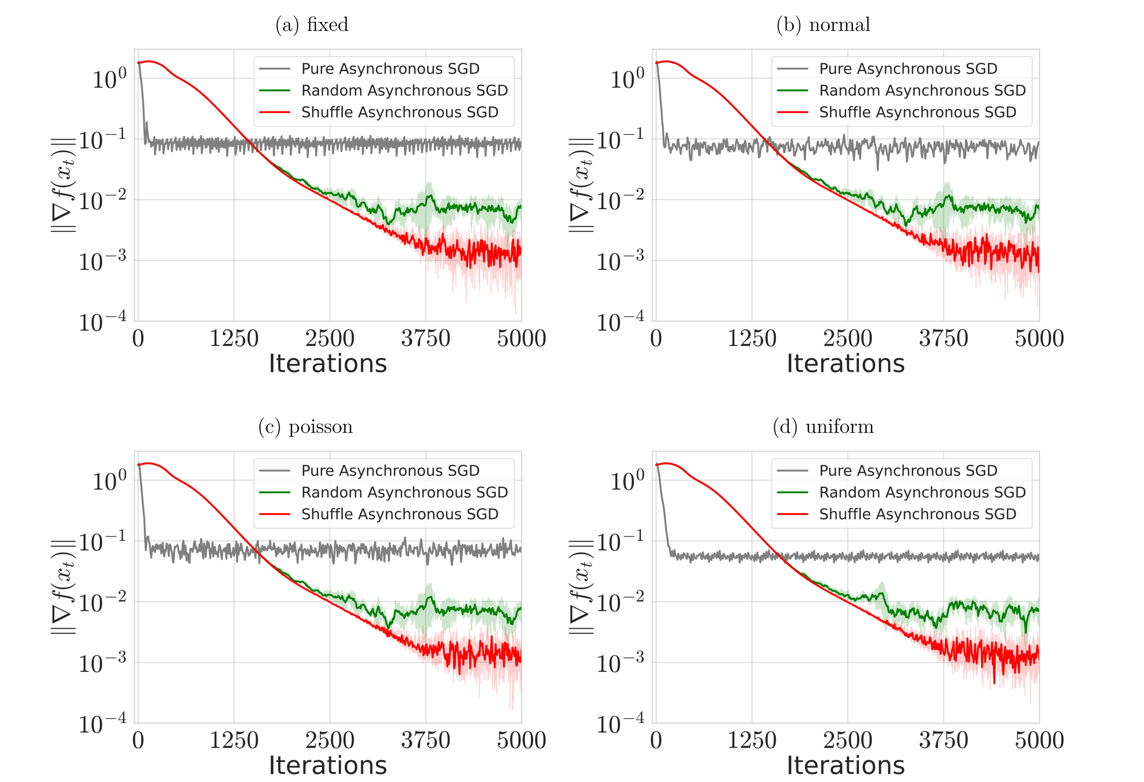


Figure 1:Comparison of pure, random, and shuffled asynchronous SGD with tuned stepsizes and full gradient computation on w7a dataset with various delay patterns. Here $n=10$, $\lambda=0.1$, $d=300$, $m=2505$.

References

[1] K. Mishchenko, F. Bach, M. Even, and B. Woodworth. *Asynchronous SGD beats minibatch SGD under arbitrary delays.* Advances in Neural Information Processing Systems, 2022.

[2] A. Koloskova, S. Stich, and M. Jaggi. *Sharper convergence guarantees for asynchronous SGD for distributed and federated learning.* Advances in Neural Information Processing Systems, 2022.

[3] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba. *Federated learning with buffered asynchronous aggregation.* International Conference on Artificial Intelligence and Statistics, 2022.

[4] A. Koloskova, N. Doikov, S. U. Stich, and M. Jaggi. *Shuffle SGD is always better than SGD: Improved analysis of SGD with arbitrary data orders.* arXiv preprint arXiv:2305.19259, 2023.