Rate of Convergence Analysis of Constant Step-Size Distributed Stochastic Gradient Descent

3rd KuVS Fachgespräch "Machine Learning & Networking"
6th October 2022

Adrian Redder, Paderborn University

Arunselvan Ramaswamy, Karlstad University
Holger Karl, Hasso-Plattner-Institut Potsdam

Agenda

Part 1

Distributed asynchronous stochastic gradient descent (DASGD) for machine learning.

Part 2

A rate of convergence result for constant step-size DASGD.

Part 3

Discussion of open problems from a resource scheduling and computer networks perspective: **training quality vs. network resources**

Agenda

Part 1

Distributed asynchronous stochastic gradient descent (DASGD) for machine learning.

Part 2

A rate of convergence result for constant step-size DASGD.

Part 3

Discussion of open problems from a resource scheduling and computer networks perspective: **training quality vs. network resources**

Introduction

Empirical Risk Minimization (ERM)

Objective: Given a data memory/set $\mathcal{R}$ containing N data points ,

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

where $f_i(x)$ is the loss associated with the i-th data point.

Introduction

Empirical Risk Minimization (ERM)

Objective: Given a data memory/set $\mathcal{R}$ containing N data points ,

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

where $f_i(x)$ is the loss associated with the i-th data point.

Problem

- Large-scale data setting: N is typically very large (ImageNet etc.)
- Global optimization approaches and even plain gradient descent are impractical.

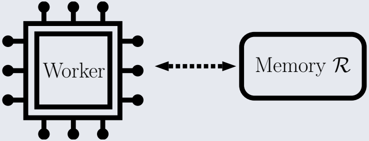
Introduction

Batch SGD solution for ERM

Initialize $x \in \mathbb{R}^d$.
Fix a regularizer $r(\cdot)$.
Fix step-size $\alpha > 0$.
repeat:

1: Sample $M \ll N$ data points i_m from $\mathcal{R}$.

2: $x \longleftarrow x - \alpha \left(\frac{1}{M} \sum_{m=1}^{M} \nabla_x f_{i_m}(x) + \nabla_x r(x) \right)$



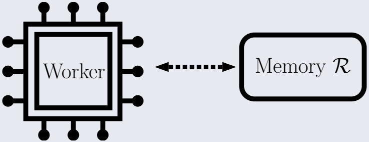
Introduction

Batch SGD solution for ERM

Initialize $x \in \mathbb{R}^d$.
Fix a regularizer $r(\cdot)$.
Fix step-size $\alpha > 0$.
repeat:

1: Sample $M \ll N$ data points i_m from $\mathcal{R}$.

2: $x \longleftarrow x - \alpha \left(\frac{1}{M} \sum_{m=1}^{M} \nabla_x f_{i_m}(x) + \nabla_x r(x) \right)$



Advantage

- For "small" M (depends on memory and $f_i(x)$) all $\nabla_x f_{i_m}(x) + \nabla_x r(x)$ can be computed efficiently on a single machine/node/worker/agent.

Underlying problem setup in ERM

Unconstrained Stochastic Optimization Problem

Objective:

$$\min_{x \in \mathbb{R}^d} f(x),$$

with objective $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(x) := \mathbb{E}\left[F(x;\omega)\right] = \int_{\omega \in \Omega} F(x;\omega) d\mathbb{P}(\omega)$$

for some **random function** $F : \mathbb{R}^d \times \Omega \to \mathbb{R}$ and underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

Underlying problem setup in ERM

Unconstrained Stochastic Optimization Problem

Objective:

$$\min_{x \in \mathbb{R}^d} f(x),$$

with objective $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(x) := \mathbb{E}\left[F(x;\omega)\right] = \int_{\omega \in \Omega} F(x;\omega) d\mathbb{P}(\omega)$$

for some **random function** $F : \mathbb{R}^d \times \Omega \to \mathbb{R}$ and underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

Stochastic gradient descent (SGD) with constant step-size $\alpha > 0$

At every discrete time-step n do:

1. Observe **sample** $\omega_n \in \Omega$;
2. $x_{n+1} = x_n - \alpha \nabla_x F(x_n; \omega_n)$;

Asynchronous multi-processor SGD

SGD on processor $i \in V := \{1, \ldots, D\}$

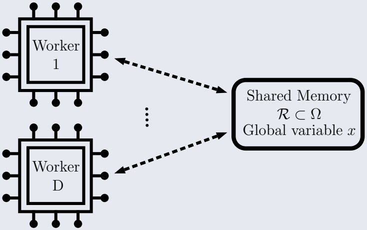
repeat:

1: Read current iterate x from shared memory.

2: Sample data $\omega \in \mathcal{R}$ from shared memory.

3: Compute stochastic gradient $\nabla_x F(x; \omega)$

4: Overwrite current global iterate with $x - \alpha \nabla_x F(x; \omega)$ in shared memory.

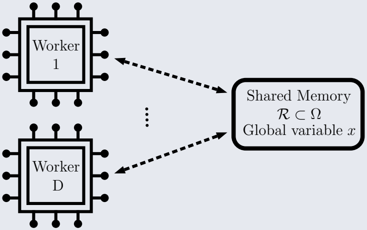


Asynchronous multi-processor SGD

SGD on processor $i \in V := \{1, \ldots, D\}$

repeat:

1: Read current iterate x from shared memory.
2: Sample data $\omega \in \mathcal{R}$ from shared memory.
3: Compute stochastic gradient $\nabla_x F(x; \omega)$
4: Overwrite current global iterate with $x - \alpha \nabla_x F(x; \omega)$ in shared memory.



Advantage

Parallel computation of multiple stochastic gradients.

Asynchronous multi-processor SGD

SGD on processor $i \in V := \{1, \ldots, D\}$

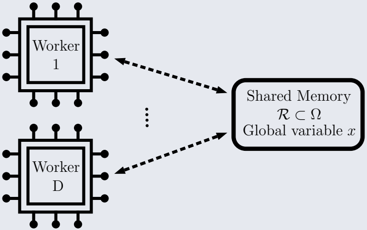
repeat:

1: Read current iterate x from shared memory.

2: Sample data $\omega \in \mathcal{R}$ from shared memory.

3: Compute stochastic gradient $\nabla_x F(x;\omega)$

4: Overwrite current global iterate with $x - \alpha \nabla_x F(x;\omega)$ in shared memory.



Advantage

Parallel computation of multiple stochastic gradients.

Problem

- Step 3 is the computational bottleneck.
- Step 2 and 3 take different time for **heterogeneous machines**.

Asynchronous multi-processor SGD (continued)

Example

- While processor 1 runs step 2 & 3, the global variable may be updated K times. $\implies$ When processor 1 applies SGD step (step 4), the stochastic gradient is K time steps old!

Asynchronous multi-processor SGD (continued)

Example

- While processor 1 runs step 2 & 3, the global variable may be updated K times. $\Longrightarrow$ When processor 1 applies SGD step (step 4), the stochastic gradient is K time steps old!

Hypothetical global clock n

- n runs faster than every local iterate counter at each worker.
- E.g. union over all local time-steps in $[0, \infty)$, where workers read and write on the memory. Then enumerate.

Example (continued)

The global iterate experiences the gradient error:

$$\nabla_x F(x_n; \omega) - \nabla_x F(x_{n-K}; \omega)$$

Asynchronous coordinate-wise SGD

SGD on processor $i \in V$ for coordinate $x^i \in \mathbb{R}^{d^i}$, $d = \sum_{i=1}^{D} d^i$

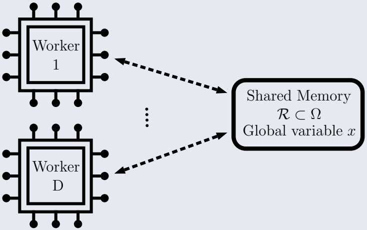
repeat:

1: Read current iterate x from shared memory.

2: Sample data $\omega \in \mathcal{R}$ from shared memory.

3: Compute stochastic gradient $\nabla_{x^i} F(x;\omega)$

4: Overwrite i-th coordinate of global iterate with $x^i - \alpha \nabla_{x^i} F(x;\omega)$ in shared memory.



Asynchronous coordinate-wise SGD

SGD on processor $i \in V$ for coordinate $x^i \in \mathbb{R}^{d^i}$, $d = \sum_{i=1}^{D} d^i$

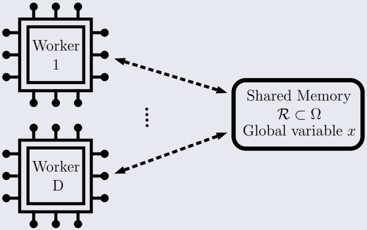
repeat:

1: Read current iterate x from shared memory.

2: Sample data $\omega \in \mathcal{R}$ from shared memory.

3: Compute stochastic gradient $\nabla_{x^i} F(x; \omega)$

4: Overwrite i-th coordinate of global iterate with $x^i - \alpha \nabla_{x^i} F(x; \omega)$ in shared memory.



Problem

- Additional asynchronous updates of each x^i.
- Gradient errors:

$$\nabla_{x^i} F(x_n; \omega) - \nabla_{x^i} F(x^1_{n-\Delta_{i1}(n)}, \ldots, x^D_{n-\Delta_{iD}(n)}; \omega)$$

- $\Delta_{ij}(n)$ can be viewed as **Age of Information** (AoI) random variables (from the perspective of the hypothetical global clock)

Fully distributed heterogeneous workers

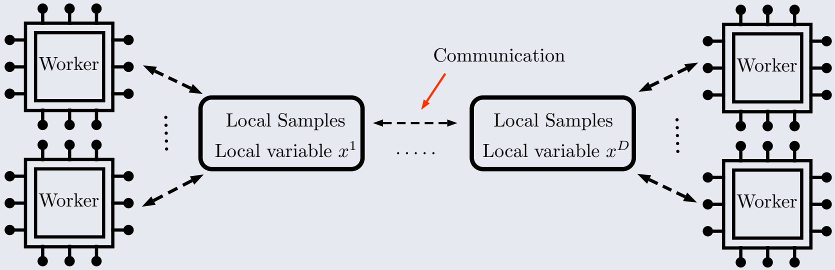
- As of now: AoI/Delay was due to heterogeneous computing resources.

Fully distributed heterogeneous workers

- As of now: AoI/Delay was due to heterogeneous computing resources.
- Shared memory (or samples from environment) may be distributed.
- Example:
  Agents locally store a coordinate x^i and gets local samples $\omega^i \in \Omega$ from its local environment.

Fully distributed heterogeneous workers

- As of now: AoI/Delay was due to heterogeneous computing resources.
- Shared memory (or samples from environment) may be distributed.
- Example:
  Agents locally store a coordinate x^i and gets local samples $\omega^i \in \Omega$ from its local environment.



Distributed Asynchronous Stochastic Gradient Descent

AoI from the perspective of the i-th coordinate

- Define
$$x_{\Delta_i(n)} := (x^1_{n-\Delta_{i1}(n)}, \dots, x^D_{n-\Delta_{iD}(n)}).$$
- Approximation of the global variable x_n w.r.t hypothetical global clock n.
- At each tick of the global clock, at least one coordinate gets updated.

Distributed Asynchronous Stochastic Gradient Descent

AoI from the perspective of the *i*-th coordinate

- Define
$$x_{\Delta_i(n)} \coloneqq (x^1_{n-\Delta_{i1}(n)}, \dots, x^D_{n-\Delta_{iD}(n)}).$$
- Approximation of the global variable x_n w.r.t hypothetical global clock n.
- At each tick of the global clock, at least one coordinate gets updated.

DASGD iteration

$$x^i_{n+1} = x^i_n + \alpha^i \mathbb{1}_{Y_n}(i) \nabla_{x_i} F(x_{\Delta_i(n)}; \omega^i_n),$$

with $Y_n \subset V$ for all $n \geq 0$.

Distributed Asynchronous Stochastic Gradient Descent

AoI from the perspective of the *i*-th coordinate

- Define
$$x_{\Delta_i(n)} := (x^1_{n-\Delta_{i1}(n)}, \ldots, x^D_{n-\Delta_{iD}(n)}).$$
- Approximation of the global variable x_n w.r.t hypothetical global clock n.
- At each tick of the global clock, at least one coordinate gets updated.

DASGD iteration

$$x^i_{n+1} = x^i_n + \alpha^i \mathbb{1}_{Y_n}(i) \nabla_{x_i} F(x_{\Delta_i(n)}; \omega^i_n),$$

with $Y_n \subset V$ for all $n \geq 0$.

Remarks

- The AoI processes $\Delta_{ij}(n)$ contain information delay due to heterogeneous updates **and/or** communication.
- Main question: How do the AoI processes affect the **rate of convergence**?

Agenda

Part 1

Distributed asynchronous stochastic gradient descent (DASGD) for machine learning.

Part 2

A rate of convergence result for constant step-size DASGD.

Part 3

Discussion of open problems from resource scheduling and computer networks perspective.

Recall problem formalization

Unconstrained Stochastic Optimization Problem

Let $x^i \in \mathbb{R}^{d_i}$ be a **local variable** with $x = (x^1, \dots, x^D)$. Objective:

$$\min_{x \in \mathbb{R}^d} f(x),$$

with objective $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(x) := \mathbb{E}\left[F(x;\omega)\right] = \int_{\omega \in \Omega} F(x;\omega) d\mathbb{P}(\omega)$$

for some **random function** $F : \mathbb{R}^d \times \Omega \to \mathbb{R}$.

Recall problem formalization

Unconstrained Stochastic Optimization Problem

Let $x^i \in \mathbb{R}^{d_i}$ be a **local variable** with $x = (x^1, \ldots, x^D)$. Objective:

$$\min_{x \in \mathbb{R}^d} f(x),$$

with objective $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(x) := \mathbb{E}\left[F(x;\omega)\right] = \int_{\omega \in \Omega} F(x;\omega) d\mathbb{P}(\omega)$$

for some **random function** $F : \mathbb{R}^d \times \Omega \to \mathbb{R}$.

DASGD iteration

$$x^i_{n+1} = x^i_n + \alpha^i \mathbb{1}_{Y_n}(i) \nabla_{x_i} F(x_{\Delta_i(n)}; \omega^i_n),$$

with $Y_n \subset V$ for all $n \geq 0$.

Recall problem formalization

Unconstrained Stochastic Optimization Problem

Let $x^i \in \mathbb{R}^{d_i}$ be a **local variable** with $x = (x^1, \ldots, x^D)$. Objective:

$$\min_{x \in \mathbb{R}^d} f(x),$$

with objective $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(x) := \mathbb{E}\left[F(x; \omega)\right] = \int_{\omega \in \Omega} F(x; \omega) d\mathbb{P}(\omega)$$

for some **random function** $F : \mathbb{R}^d \times \Omega \to \mathbb{R}$.

DASGD iteration

$$x^i_{n+1} = x^i_n + \alpha^i \mathbb{1}_{Y_n}(i) \nabla_{x_i} F(x_{\Delta_i(n)}; \omega^i_n),$$

with $Y_n \subset V$ for all $n \geq 0$.

Number of times the i-th coordinate gets updated

For every $n \geq 0$, define

$$\nu(n, i) = \sum_{k=0}^{n} \mathbb{1}_{Y_n}(i).$$

Assumptions

(A1) $F(x;\omega)$ is differentiable in x for $\mathbb{P}$-almost all $\omega \in \Omega$.

(A2) $\inf_{x\in\mathbb{R}^d} f(x) > -\infty$.

(A3) The random processes w_n^i are $i.i.d.$

(A4) $\nabla_x F(x;\omega)$ has finite second moment: $\sup_{x\in\mathbb{R}^d} \mathbb{E}\left[\|\nabla_x F(x;\omega)\|_2^2\right] < \infty$.

(A5) $\mathbb{E}\left[\nabla_x F(x;\omega)\right]$ is Lipschitz continuous.

(A6) $\liminf\limits_{n\to\infty} \frac{\nu(n,i)}{n} > 0$ for all $i \in V$.

Assumptions

(A1) $F(x;\omega)$ is differentiable in x for $\mathbb{P}$-almost all $\omega \in \Omega$.

(A2) $\inf_{x\in\mathbb{R}^d} f(x) > -\infty$.

(A3) The random processes w_n^i are $i.i.d.$

(A4) $\nabla_x F(x;\omega)$ has finite second moment: $\sup_{x\in\mathbb{R}^d} \mathbb{E}\left[\|\nabla_x F(x;\omega)\|_2^2\right] < \infty$.

(A5) $\mathbb{E}\left[\nabla_x F(x;\omega)\right]$ is Lipschitz continuous.

(A6) $\liminf\limits_{n\to\infty} \frac{\nu(n,i)}{n} > 0$ for all $i \in V$.

Remarks

- (A3) can be weakened to dependent or Martingale noise.
- (A1) and (A4) $\implies \nabla_x f(x) = \nabla_x \mathbb{E}\left[F(x;\omega)\right] = \mathbb{E}\left[\nabla_x F(x;\omega)\right]$.
- (A5) $\implies$

$$\|\nabla_{x^i} f(x^1,\ldots,x^j,\ldots x^D) - \nabla_{x^i} f(x^1,\ldots,y^j,\ldots x^D)\| \leq L^{ij}\|x^j - y^j\|$$

for all $x \in \mathbb{R}^d$ and $y^j \in \mathbb{R}^{d_j}$, with Lipschitz constant L^{ij} for all $(i,j) \in V \times V$.

Rate of convergence analysis

Theorem

Define $\kappa^i := \{\liminf_{n\to\infty} \frac{\nu(n,i)}{n}\} \in (0,1]$ for any $i \in V$, then

$$\frac{1}{n}\sum_{k=0}^{n-1} \mathbb{E}\left[\|\nabla f(x_k)\|_2^2\right] \leq$$

$$\mathcal{O}\left(\frac{1}{n}\right) + \underbrace{\mathcal{O}\left(\frac{\max_i \alpha^i}{\min_i \kappa^i}\right) + \mathcal{O}\left(\sum_{i\in V} \underbrace{\frac{\alpha^i}{\kappa^i}\left[\sum_{j\neq i} L^{ij}\left(\frac{1}{n}\sum_{k=0}^{n-1}\Delta_{ij}(k)\right)\right]}_{\textbf{local quantity}}\right)}_{:=\varepsilon^n}.$$

Rate of convergence analysis

Theorem

Define $\kappa^i := \{\liminf_{n\to\infty} \frac{\nu(n,i)}{n}\} \in (0,1]$ for any $i \in V$, then

$$\frac{1}{n}\sum_{k=0}^{n-1} \mathbb{E}\left[\|\nabla f(x_k)\|_2^2\right] \leq$$

$$\mathcal{O}\left(\tfrac{1}{n}\right) + \underbrace{\mathcal{O}\left(\frac{\max_i \alpha^i}{\min_i \kappa^i}\right) + \mathcal{O}\left(\sum_{i\in V} \underbrace{\frac{\alpha^i}{\kappa^i}\left[\sum_{j\neq i} L^{ij}\left(\frac{1}{n}\sum_{k=0}^{n-1}\Delta_{ij}(k)\right)\right]}_{\textbf{local quantity}}\right)}_{:=\varepsilon^n}.$$

Remarks

- No separate bounds for individual $\nabla f(x_k)\big|_i$ as a function of Δ_{ij} for $j \neq i$.
- The slowest update rate is a bottleneck for the limiting neighborhood radius
$$\varepsilon := \lim_{n\to\infty} \varepsilon^n.$$

Rate of convergence analysis (continued)

Corollary

There exists a subsequence $\{n_k\}_{k \geq 0}$, such that

$$\nabla f(x_{n_k}) \to \mathcal{B}_\varepsilon(0) \text{ with high probability.}$$

Rate of convergence analysis (continued)

Corollary

There exists a subsequence $\{n_k\}_{k\geq 0}$, such that

$$\nabla f(x_{n_k}) \to \mathcal{B}_\varepsilon(0) \text{ with high probability.}$$

Another pespective

At every time step n sample $\tilde{x}_n$ from $\{x_0, \dots, x_{n-1}\}$ uniformly at random, then

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_n)\|_2^2\right] \to \mathcal{B}_\varepsilon(0).$$

Agenda

Part 1

Distributed asynchronous stochastic gradient descent (DASGD) for machine learning.

Part 2

A rate of convergence result for constant step-size DASGD.

Part 3

Discussion of open problems from a resource scheduling and computer networks perspective: **training quality vs. network resources**

Discussion of problems

Training quality vs. network and computing resources

$$\sum_{i\in V} \underbrace{\frac{\alpha^i}{\kappa^i}\left[\sum_{j\neq i} L^{ij}\left(\frac{1}{n}\sum_{k=0}^{n-1}\Delta_{ij}(k)\right)\right]}_{\textbf{local quantity}}$$

Discussion of problems

Training quality vs. network and computing resources

$$\sum_{i \in V} \underbrace{\frac{\alpha^i}{\kappa^i} \left[\sum_{j \neq i} L^{ij} \left(\frac{1}{n} \sum_{k=0}^{n-1} \Delta_{ij}(k) \right) \right]}_{\textbf{local quantity}}$$

(+) Effect of peak AoI gets averaged.

Discussion of problems

Training quality vs. network and computing resources

$$\sum_{i \in V} \underbrace{\frac{\alpha^i}{\kappa^i} \left[\sum_{j \neq i} L^{ij} \left(\frac{1}{n} \sum_{k=0}^{n-1} \Delta_{ij}(k) \right) \right]}_{\textbf{local quantity}}$$

(+) Effect of peak AoI gets averaged.

How close we come to a stationary point (ε) is affected by:

- level of asynchronicity $\frac{1}{\kappa^i}$.
- problem/algorithm dynamics L^{ij}.
- average AoI $\frac{1}{n} \sum_{k=0}^{n-1} \Delta_{ij}(k)$.

Discussion of problems

Training quality vs. network and computing resources

$$\sum_{i\in V} \underbrace{\frac{\alpha^i}{\kappa^i}\left[\sum_{j\neq i} L^{ij}\left(\frac{1}{n}\sum_{k=0}^{n-1}\Delta_{ij}(k)\right)\right]}_{\textbf{local quantity}}$$

(+) Effect of peak AoI gets averaged.

How close we come to a stationary point (ε) is affected by:

- level of asynchronicity $\frac{1}{\kappa^i}$.
- problem/algorithm dynamics L^{ij}.
- average AoI $\frac{1}{n}\sum_{k=0}^{n-1}\Delta_{ij}(k)$.

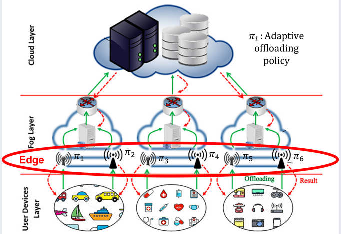
Example

We can assign multiple "bad" workers to one coordinate i:

- $\frac{1}{\kappa^i}$ small, but $\frac{1}{n}\sum_{k=0}^{n-1}\Delta_{ij}(k)$ large.

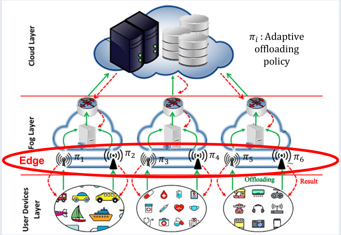
Discussion of problems

Problem 1: Online Edge Computational Task Offloading



Discussion of problems

Problem 1: Online Edge Computational Task Offloading



D synchronous workers 1 for each coordinate

Theorem shows that we should minimize

$$\sum_{i \in V} \left[\sum_{j \neq i} L^{ij} \left(\frac{1}{n} \sum_{k=0}^{n-1} \Delta_{ij}(k) \right) \right]$$

$(-)$ Non-trivial problem since L^{ij} are unknown and most likely different. $\implies$

1. Joint estimation and scheduling problem.
2. Model free scheduling problem.

Open problems

Problem 2: D asynchronous workers on whole $x \in \mathbb{R}^d$

$$\sum_{i \in V} \left[\sum_{j \neq i} L^{ij} \left(\frac{1}{n} \sum_{k=0}^{n-1} \Delta(k) \right) \right]$$

- $\Delta(n)$ is now a function of the level of asynchronicity of the workers.

Open problems

Problem 2: D asynchronous workers on whole $x \in \mathbb{R}^d$

$$\sum_{i \in V} \left[\sum_{j \neq i} L^{ij} \left(\frac{1}{n} \sum_{k=0}^{n-1} \Delta(k) \right) \right]$$

- $\Delta(n)$ is now a function of the level of asynchronicity of the workers.

Problem 3: Multiple parallel training runs

- Try to minimize multiple $f(x), g(x), h(x)$ in parallel on a set of workers.
- Objective, e.g., same make span.

Conclusions

Final remarks

- When do $\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \Delta_{ij}(k)$ exist? [1]
- Non-stationary data and constant step-size.
- Distributed Multi-Agent Reinforcement Learning [2]

Thank you for your attention!

[1] Redder, A., Ramaswamy, A., Karl, H. "Age of Information Process under Strongly Mixing Communication – Moment Bound, Mixing Rate and Strong Law", Proc. 58th Allerton Conference on Communication, Control, and Computing (2022)

[2] Redder, A., Ramaswamy, A., Karl, H. "Asymptotic Convergence of Deep Multi-Agent Actor-Critic Algorithms." https://arxiv.org/abs/2201.00570 (2022)