Acceleration in Distributed Optimization under Similarity

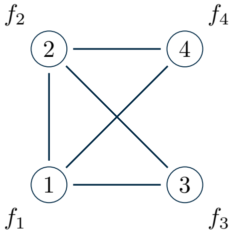
Ye Tian[*], Gesualdo Scutari[*], Tianyu Cao[*], and Alexander Gasnikov[†]

[*] Purdue University
[†] MIPT, ISP RAS Research Center for Trusted Artificial Intelligence

The 25th International Conference on Artificial Intelligence and Statistics

Distributed Optimization over Networks

$$\min_x \quad \underbrace{\frac{1}{m}\sum_{i=1}^{m} f_i(x)}_{F(x)} + G(x) \qquad \text{(P)}$$



- Each agent i locally owns only f_i and G
- $G : \mathbb{R}^d \to (-\infty, +\infty]$ is nonsmooth & convex, known to all the agents
- Communication among nodes is modeled as a general connected graph

Distributed algorithms: each agent performs computations locally and communicates only to its immediate neighbors.

Case Study: Empirical Risk Minimization (ERM) in Network

With $\mathcal{D} := \{Z_1, \ldots Z_N\} \sim \mathbb{P}$, compute

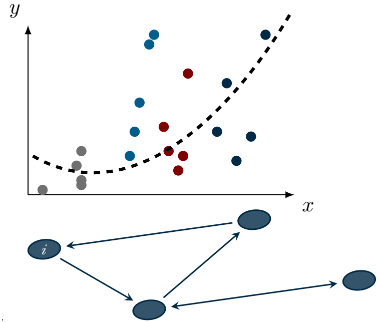
$$\widehat{x} = \underset{x \in \Theta}{\text{argmin}}\, F(x) \triangleq \frac{1}{N}\sum_{i=1}^{N} \ell(x; Z_i)$$

regression

logistic

svm

. . .



Case Study: Empirical Risk Minimization (ERM) in Network

With $\mathcal{D} := \{Z_1, \ldots Z_N\} \sim \mathbb{P}$, compute

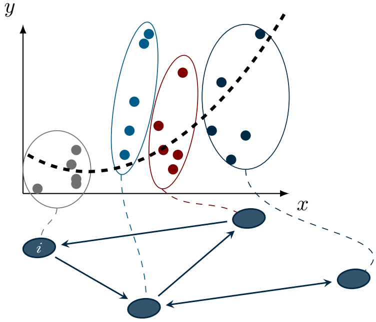
$$\widehat{x} = \underset{x \in \Theta}{\text{argmin}}\, F(x) \triangleq \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{n} \sum_{j \in \mathcal{D}_i} \ell(x; Z_j)}_{f_i(x)}$$

regression

logistic

svm

...



Communication Complexity of First Order Methods

Algorithm	Rate (# comm.)
`SSDA/MSDA` [Sca-Bach-Bub'17] `OPAPC` [Kov-Sal-Ric'20] `Accelerated Dual Ascent` [Uri-Lee-Gas'20]	$\mathcal{O}\left(\sqrt{\kappa_{local}}\sqrt{\frac{1}{1-\rho}}\log\frac{1}{\varepsilon}\right)$
`APM-C` [Li-Fang-Yin-Lin'18]	$\mathcal{O}\left(\sqrt{\kappa_{local}}\sqrt{\frac{1}{1-\rho}}\log^2\frac{1}{\varepsilon}\right)$
`Accelerated EXTRA` [Li-Lin'20]	$\widetilde{O}\left(\sqrt{\kappa_{local}}\sqrt{\frac{1}{1-\rho}}\log\frac{1}{\varepsilon}\right)$
`Mudag, DPAG` [Ye-Luo-Zhou-Zha'20]	$\widetilde{O}\left(\sqrt{\kappa_{global}}\sqrt{\frac{1}{1-\rho}}\log\frac{1}{\varepsilon}\right)$
`DAccGD` [Rog-Luk-Gas'20]	$\widetilde{O}\left(\sqrt{\kappa_{global}}\frac{1}{1-\rho}\log^2\frac{1}{\varepsilon}\right)$

$$\kappa_{global} = \frac{L}{\mu}, \qquad \kappa_{local} = \frac{L_{mx}}{\mu_{mn}}$$

Communication Complexity of First Order Methods

Algorithm	Rate (# comm.)
SSDA/MSDA [Sca-Bach-Bub'17] OPAPC [Kov-Sal-Ric'20] Accelerated Dual Ascent [Uri-Lee-Gas'20]	$\mathcal{O}\left(\sqrt{\kappa_{local}}\sqrt{\frac{1}{1-\rho}}\log\frac{1}{\varepsilon}\right)$
APM-C [Li-Fang-Yin-Lin'18]	$\mathcal{O}\left(\sqrt{\kappa_{local}}\sqrt{\frac{1}{1-\rho}}\log^2\frac{1}{\varepsilon}\right)$
Accelerated EXTRA [Li-Lin'20]	$\widetilde{O}\left(\sqrt{\kappa_{local}}\sqrt{\frac{1}{1-\rho}}\log\frac{1}{\varepsilon}\right)$
Mudag, DPAG [Ye-Luo-Zhou-Zha'20]	$\widetilde{O}\left(\sqrt{\kappa_{global}}\sqrt{\frac{1}{1-\rho}}\log\frac{1}{\varepsilon}\right)$
DAccGD [Rog-Luk-Gas'20]	$\widetilde{O}\left(\sqrt{\kappa_{global}}\frac{1}{1-\rho}\log^2\frac{1}{\varepsilon}\right)$

$$\kappa_{global} = \frac{L}{\mu}, \qquad \kappa_{local} = \frac{L_{mx}}{\mu_{mn}}$$

For ill-conditioned problems with extremely large κ_{global} or κ_{local}, can we further improve communication complexity?

Leveraging Function Similarity

- **Statistical similarity:** i.i.d. data + assump. [Arj-Sha'05] [Hen-Xiao-Bub-Bach'20]

$$\|\nabla^2 f_i - \nabla^2 F\| \leq \beta = \mathcal{O}_d\left(\sqrt{1/n}\right) \quad \text{(on } \Theta\text{)} \;\; \text{w.h.p.}$$

ERM with optimal regularization: $\kappa = \mathcal{O}(\sqrt{m \cdot n}), \quad \beta/\mu = \mathcal{O}(\sqrt{m})$

Leveraging Function Similarity

- **Statistical similarity:** i.i.d. data + assump. [Arj-Sha'05] [Hen-Xiao-Bub-Bach'20]

$$\|\nabla^2 f_i - \nabla^2 F\| \leq \beta = \mathcal{O}_d\left(\sqrt{1/n}\right) \quad \text{(on } \Theta\text{)} \;\; \text{w.h.p.}$$

ERM with optimal regularization: $\kappa = \mathcal{O}(\sqrt{m \cdot n})$, $\beta/\mu = \mathcal{O}(\sqrt{m})$

- **Exploiting function similarity to reduce communication complexity**
  - State-of-the-arts over star network

Algorithm	Rate (# comm.)
DANE (quadratic) [Sha-Sre-Zha'14] CEASE [Fan-Guo-Wang'19]	$\widetilde{\mathcal{O}}\left(\left(\frac{\beta}{\mu}\right)^2 \log\frac{1}{\varepsilon}\right)$
[Lu-Fre-Nes'18]	$\widetilde{\mathcal{O}}\left(\frac{\beta}{\mu} \log\frac{1}{\varepsilon}\right)$
DiSCO (self-concordant loss) [Zha-Lin'15]	$\widetilde{O}\left(\left(1+\sqrt{\frac{\beta}{\mu}}\right) \log\frac{1}{\varepsilon}\right)$
SPAG [Hen-Xiao-Bub'20]	$O\left(\frac{\beta}{\mu} \log\frac{1}{\varepsilon}\right)$, asymptotically

Leveraging Function Similarity

- **Statistical similarity:** i.i.d. data + assump. [Arj-Sha'05] [Hen-Xiao-Bub-Bach'20]

$$\|\nabla^2 f_i - \nabla^2 F\| \leq \beta = \mathcal{O}_d\left(\sqrt{1/n}\right) \quad \text{(on } \Theta\text{)} \;\; \text{w.h.p.}$$

ERM with optimal regularization: $\kappa = \mathcal{O}(\sqrt{m \cdot n}), \quad \beta/\mu = \mathcal{O}(\sqrt{m})$

- **Exploiting function similarity to reduce communication complexity**
  - State-of-the-arts over star network

Algorithm	**Rate (# comm.)**
`DANE` (quadratic) [Sha-Sre-Zha'14] `CEASE` [Fan-Guo-Wang'19]	$\widetilde{\mathcal{O}}\left(\left(\frac{\beta}{\mu}\right)^2 \log\frac{1}{\varepsilon}\right)$
[Lu-Fre-Nes'18]	$\widetilde{\mathcal{O}}\left(\frac{\beta}{\mu}\log\frac{1}{\varepsilon}\right)$
`DiSCO` (self-concordant loss) [Zha-Lin'15]	$\widetilde{O}\left(\left(1+\sqrt{\frac{\beta}{\mu}}\right)\log\frac{1}{\varepsilon}\right)$
`SPAG` [Hen-Xiao-Bub'20]	$O\left(\frac{\beta}{\mu}\log\frac{1}{\varepsilon}\right)$, asymptotically

  - Can we achieve $\sqrt{\frac{\beta}{\mu}}$ complexity dependency on mesh networks?
    Accelerate SONATA algorithm!

Proposed Approach: Accelerated SONATA

- update local cost:

$$f_i^{k+1}(x) = f_i(x) + \frac{\beta - \mu}{2} \left\| x - x_i^k \right\|^2$$

- execute SONATA for T rounds:

$$\left\{ z_i^{k+1} \right\}_{i \in [m]} \approx \text{SONATA} \left(\operatorname*{argmin}_x \sum_{i=1}^{m} f_i^{k+1}(x) + G(x) \right)$$

- extrapolation:

$$x_i^{k+1} = z_i^{k+1} + \frac{1 - \alpha}{1 + \alpha} (z_i^{k+1} - z_i^k)$$

Communication Complexity

Theorem (Star Network)

The total # communication rounds needed by the Accelerated SONATA-star algorithm to obtain $\frac{1}{m}\sum_{i=1}^{m}\left(U(x_i^k) - U^\star\right) \leq \epsilon$ *reads*

$$\mathcal{O}\left(\sqrt{\frac{\beta}{\mu}}\log\left(\frac{\beta}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

Theorem (Mesh Network)

The total # communication rounds needed by the Accelerated SONATA algorithm to obtain $\frac{1}{m}\sum_{i=1}^{m}\left(U(x_i^k) - U^\star\right) \leq \epsilon$ *reads*

$$\mathcal{O}\left(\sqrt{\frac{\beta/\mu}{1-\rho}}\log\left(1+\frac{\kappa-1}{\beta/\mu}\right)\log\left(\frac{\beta}{\mu}\right)\log\frac{1}{\varepsilon}\right).$$

Communication Complexity

Theorem (Star Network)

The total # communication rounds needed by the Accelerated SONATA-star algorithm to obtain $\frac{1}{m}\sum_{i=1}^{m}\left(U(x_i^k) - U^\star\right) \leq \epsilon$ *reads*

$$\mathcal{O}\left(\sqrt{\frac{\beta}{\mu}}\log\left(\frac{\beta}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

Theorem (Mesh Network)

The total # communication rounds needed by the Accelerated SONATA algorithm to obtain $\frac{1}{m}\sum_{i=1}^{m}\left(U(x_i^k) - U^\star\right) \leq \epsilon$ *reads*

$$\mathcal{O}\left(\sqrt{\frac{\beta/\mu}{1-\rho}}\log\left(1+\frac{\kappa-1}{\beta/\mu}\right)\log\left(\frac{\beta}{\mu}\right)\log\frac{1}{\varepsilon}\right).$$

First result matching the lower bound $\Omega\left(\sqrt{\frac{\beta/\mu}{1-\rho}}\log\left(\frac{1}{\epsilon}\right)\right)$ (up to log-factors)

Distributed Hinge Loss Minimization

$$\min_{x\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n}\sum_{j=1}^{n}\ell_s(b_i^j\cdot\langle x, a_i^j\rangle)+\frac{\lambda}{2}\|x\|^2,$$

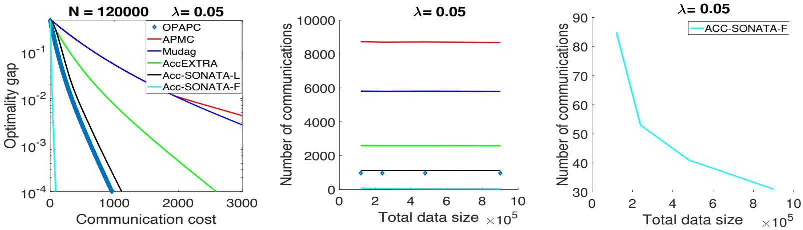


Figure: Hinge loss minimization, `HIGGS` dataset. **(left panel)**: optimality gap versus total number of communications; **(mid panel)**: number of communications to reach a precision of 10^{-4} versus (total) sample; **(right panel)**: the mid panel on a different scale of the y-axes.

Thank you!