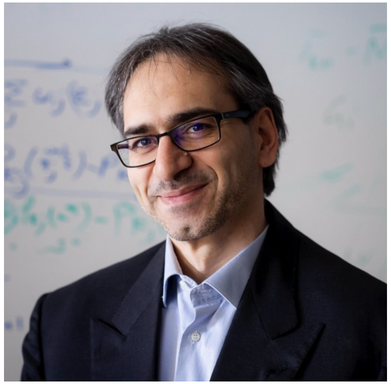
Distributed Saddle-Point Problems Under Data Similarity

Aleksandr Beznosikov
MIPT, HSE and Yandex



Gesualdo Scutari
Purdue University

Alexander Rogozin
MIPT and HSE

Alexander Gasnikov
MIPT and HSE

NeurIPS 2021

1. Problem

Distributed Saddle Point Problem

$$\min_{x \in X} \max_{y \in Y} f(x, y) := \frac{1}{M} \sum_{m=1}^{M} f_m(x, y)$$

Distributed Saddle Point Problem

$$\min_{x \in X} \max_{y \in Y} f(x, y) := \frac{1}{M} \sum_{m=1}^{M} f_m(x, y)$$

μ-strongly-convex-strongly-concave

L-smooth and convex-concave

Distributed Saddle Point Problem

$$\min_{x \in X} \max_{y \in Y} f(x, y) := \frac{1}{M} \sum_{m=1}^{M} f_m(x, y)$$

μ-strongly-convex-strongly-concave

L-smooth and convex-concave

f_m on local devices

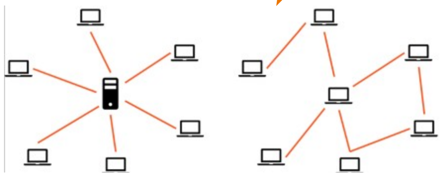
Distributed Saddle Point Problem

$$\min_{x \in X} \max_{y \in Y} f(x, y) := \frac{1}{M} \sum_{m=1}^{M} f_m(x, y)$$

μ-strongly-convex-strongly-concave

L-smooth and convex-concave

f_m on local devices



both centralized and decentralized cases

Distributed Saddle Point Problem

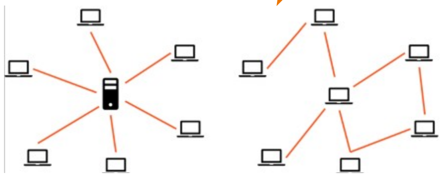
$$\min_{x \in X} \max_{y \in Y} f(x,y) := \frac{1}{M} \sum_{m=1}^{M} f_m(x,y)$$

μ-strongly-convex-strongly-concave

L-smooth and convex-concave

f_m on local devices

Communication bottleneck!



both centralized and decentralized cases

Distributed Saddle Point Problem

$$\min_{x\in X}\max_{y\in Y} f(x,y) := \frac{1}{M}\sum_{m=1}^{M} f_m(x,y)$$

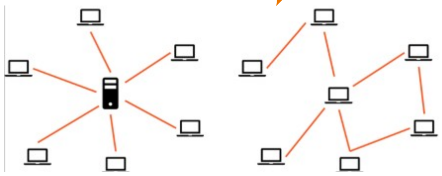
μ-strongly-convex-strongly-concave

L-smooth and convex-concave

f_m on local devices

Communication bottleneck!

Use similarity of local functions!



both centralized and decentralized cases

Similarity

$$\|\nabla^2_{xx} f_m(x,y) - \nabla^2_{xx} f(x,y)\| \leq \delta$$

$$\|\nabla^2_{xy} f_m(x,y) - \nabla^2_{xy} f(x,y)\| \leq \delta$$

$$\|\nabla^2_{yy} f_m(x,y) - \nabla^2_{yy} f(x,y)\| \leq \delta$$

local

global

Similarity

$$\|\nabla^2_{xx} f_m(x,y) - \nabla^2_{xx} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{xy} f_m(x,y) - \nabla^2_{xy} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{yy} f_m(x,y) - \nabla^2_{yy} f(x,y)\| \leq \delta$$

local

global

For uniform data similarity parameter is **small**

$$\delta = \tilde{O}(1/\sqrt{n})$$

n – number of local samples

Similarity

$$\|\nabla^2_{xx} f_m(x,y) - \nabla^2_{xx} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{xy} f_m(x,y) - \nabla^2_{xy} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{yy} f_m(x,y) - \nabla^2_{yy} f(x,y)\| \leq \delta$$

local　　　global

For uniform data similarity parameter is **small**

$$\delta = \tilde{O}(1/\sqrt{n})$$

n – number of local samples

Similarity for minimization:

- Lower bounds: Arjevani and Shamir, Communication complexity of distributed convex learning and optimization.

Similarity

$$\|\nabla^2_{xx} f_m(x,y) - \nabla^2_{xx} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{xy} f_m(x,y) - \nabla^2_{xy} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{yy} f_m(x,y) - \nabla^2_{yy} f(x,y)\| \leq \delta$$

local global

For uniform data similarity parameter is **small**

$$\delta = \tilde{O}(1/\sqrt{n})$$

n – number of local samples

Similarity for minimization:

- Lower bounds: Arjevani and Shamir, Communication complexity of distributed convex learning and optimization.
- Methods: DANE (Shamir et al.), DiSCO (Zhang and Lin), AIDE (Reddi et al.), GIANT (Wang et al.), SPAG (Hendrikx et al.), SONATA (Sun et al.).

Similarity

$$\|\nabla^2_{xx} f_m(x,y) - \nabla^2_{xx} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{xy} f_m(x,y) - \nabla^2_{xy} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{yy} f_m(x,y) - \nabla^2_{yy} f(x,y)\| \leq \delta$$

local global

For uniform data similarity parameter is **small**

$$\delta = \tilde{O}(1/\sqrt{n})$$

n – number of local samples

Similarity for minimization:

- Lower bounds: Arjevani and Shamir, Communication complexity of distributed convex learning and optimization.
- Methods: DANE (Shamir et al.), DiSCO (Zhang and Lin), AIDE (Reddi et al.), GIANT (Wang et al.), SPAG (Hendrikx et al.), SONATA (Sun et al.).

Optimal method for minimization is still not known!

Similarity

$$\|\nabla^2_{xx} f_m(x,y) - \nabla^2_{xx} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{xy} f_m(x,y) - \nabla^2_{xy} f(x,y)\| \leq \delta$$
$$\|\nabla^2_{yy} f_m(x,y) - \nabla^2_{yy} f(x,y)\| \leq \delta$$

local global

For uniform data similarity parameter is **small**

$$\delta = \tilde{O}(1/\sqrt{n})$$

n – number of local samples

Similarity for minimization:

- Lower bounds: Arjevani and Shamir, Communication complexity of distributed convex learning and optimization.
- Methods: DANE (Shamir et al.), DiSCO (Zhang and Lin), AIDE (Reddi et al.), GIANT (Wang et al.), SPAG (Hendrikx et al.), SONATA (Sun et al.).

Optimal method for minimization is still not known!

We present both lower bounds and optimal methods for SPPs!

2. Lower bounds

Class of Algorithms

- On local devices we can compute local gradeints and second derivatives in any reached points and solve any local subproblem.

Class of Algorithms

- On local devices we can compute local gradeints and second derivatives in any reached points and solve any local subproblem.
- Devices can communicate with neighbors (in the centralized case, neighbor = server).

Class of Algorithms

- On local devices we can compute local gradeints and second derivatives in any reached points and solve any local subproblem.
- Devices can communicate with neighbors (in the centralized case, neighbor = server).

This class of algorithms is fairly standard.
All algorithms used in practice belong to the proposed oracle.

Idea

- «Bad» functions – block bilinear (Zhang et al. On lower iteration complexity bounds for the saddle point problems)

$$f(x,y) := x^T A y, \quad f_m(x,y) := x^T A_m y$$

Idea

- «Bad» functions – block bilinear (Zhang et al. On lower iteration complexity bounds for the saddle point problems)

$$f(x, y) := x^T A y, \quad f_m(x, y) := x^T A_m y$$

- «Disjoint» split – odd and even blocks

$$A_1 = \begin{pmatrix} 1 & 0 & & & & & & \\ & 1 & -2 & & & & & \\ & & 1 & 0 & & & & \\ & & & 1 & -2 & & & \\ & & & & \cdots & \cdots & & \\ & & & & & 1 & -2 & \\ & & & & & & 1 & 0 \\ & & & & & & & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & -2 & & & & & & \\ & 1 & 0 & & & & & \\ & & 1 & -2 & & & & \\ & & & 1 & 0 & & & \\ & & & & \cdots & \cdots & & \\ & & & & & 1 & 0 & \\ & & & & & & 1 & -2 \\ & & & & & & & 1 \end{pmatrix}$$

Idea

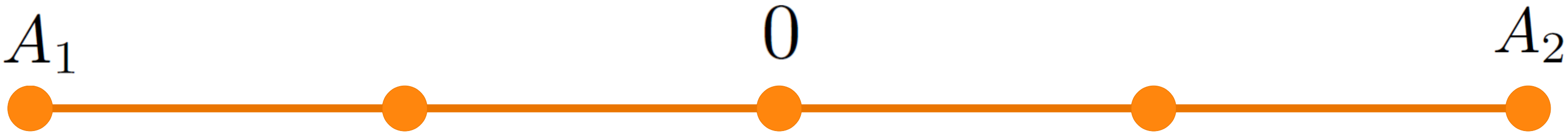
- «Bad» functions – block bilinear (Zhang et al. On lower iteration complexity bounds for the saddle point problems)

$$f(x,y) := x^T A y, \qquad f_m(x,y) := x^T A_m y$$

- «Disjoint» split – odd and even blocks

$$A_1 = \begin{pmatrix} 1 & 0 & & & & & & \\ & 1 & -2 & & & & & \\ & & 1 & 0 & & & & \\ & & & 1 & -2 & & & \\ & & & & \dots & \dots & & \\ & & & & & 1 & -2 & \\ & & & & & & 1 & 0 \\ & & & & & & & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & -2 & & & & & & \\ & 1 & 0 & & & & & \\ & & 1 & -2 & & & & \\ & & & 1 & 0 & & & \\ & & & & \dots & \dots & & \\ & & & & & 1 & 0 & \\ & & & & & & 1 & -2 \\ & & & & & & & 1 \end{pmatrix}$$

- «Long» network – chain (Scaman et al. Optimal algorithms for smooth and strongly convex distributed optimization in networks)



Idea

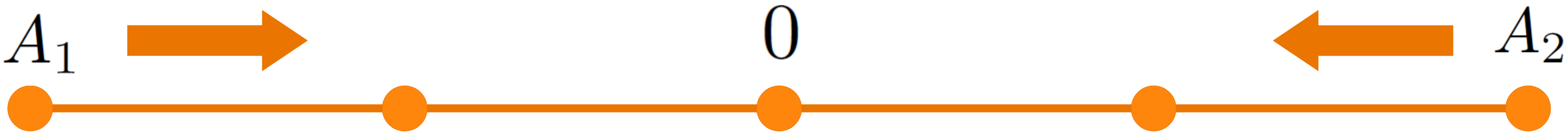
- «Bad» functions – block bilinear (Zhang et al. On lower iteration complexity bounds for the saddle point problems)

$$f(x,y) := x^T A y, \qquad f_m(x,y) := x^T A_m y$$

- «Disjoint» split – odd and even blocks

$$A_1 = \begin{pmatrix} 1 & 0 & & & & & & \\ & 1 & -2 & & & & & \\ & & 1 & 0 & & & & \\ & & & 1 & -2 & & & \\ & & & & \dots & \dots & & \\ & & & & & 1 & -2 & \\ & & & & & & 1 & 0 \\ & & & & & & & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & -2 & & & & & & \\ & 1 & 0 & & & & & \\ & & 1 & -2 & & & & \\ & & & 1 & 0 & & & \\ & & & & \dots & \dots & & \\ & & & & & 1 & 0 & \\ & & & & & & 1 & -2 \\ & & & & & & & 1 \end{pmatrix}$$

- «Long» network – chain (Scaman et al. Optimal algorithms for smooth and strongly convex distributed optimization in networks)



Theorems

Theorem. For any $L \geq \mu > 0$, $\delta > 0$, there exists a distributed SPP (satisfying our Assumptions from the 3d and 4th slides) such that the number of communication rounds required to obtain a ε-solution is lower bounded by

$$\Omega\left(\left(1 + \frac{\delta}{\mu}\right) \cdot \log\left(\frac{\|y^*\|^2}{\varepsilon}\right)\right).$$

Theorems

Theorem. For any $L \geq \mu > 0$, $\delta > 0$, there exists a distributed SPP (satisfying our Assumptions from the 3d and 4th slides) such that the number of communication rounds required to obtain a ε-solution is lower bounded by

$$\Omega\left(\left(1 + \frac{\delta}{\mu}\right) \cdot \log\left(\frac{\|y^*\|^2}{\varepsilon}\right)\right).$$

Theorem. For any $L \geq \mu > 0$, $\delta > 0$ and $\rho \in (0; 1]$, there exists a distributed SPP (satisfying our Assumptions from the 3d and 4th slides) and a gossip matrix W (over the connected network $\mathcal{G}$) with eigengap ρ, such that the number of communication rounds required to obtain a ε-solution is lower bounded by

$$\Omega\left(\frac{1}{\sqrt{\rho}}\left(1 + \frac{\delta}{\mu}\right) \cdot \log\left(\frac{\|y^*\|^2}{\varepsilon}\right)\right).$$

Theorems

Theorem. For any $L \geq \mu > 0$, $\delta > 0$, there exists a distributed SPP (satisfying our Assumptions from the 3d and 4th slides) such that the number of communication rounds required to obtain a ε-solution is lower bounded by

$$\Omega\left(\left(1 + \frac{\delta}{\mu}\right) \cdot \log\left(\frac{\|y^*\|^2}{\varepsilon}\right)\right).$$

Theorem. For any $L \geq \mu > 0$, $\delta > 0$ and $\rho \in (0; 1]$, there exists a distributed SPP (satisfying our Assumptions from the 3d and 4th slides) and a gossip matrix W (over the connected network $\mathcal{G}$) with eigengap ρ, such that the number of communication rounds required to obtain a ε-solution is lower bounded by

$$\Omega\left(\frac{1}{\sqrt{\rho}}\left(1 + \frac{\delta}{\mu}\right) \cdot \log\left(\frac{\|y^*\|^2}{\varepsilon}\right)\right).$$

Theorems state that for small similarity parameter the number of communications may not depend on the parameters of the functions.

3. Algorithms

Cenralized Algorithm

Algorithm 1 (Star Min-Max Data Similarity Algorithm)

Parameters: stepsize γ, accuracy e;

Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, for all $m \in [M]$;

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Each worker m computes $F_m(z^k)$ and sends it to the master;

3: The master node:

(i) computes $v^k = z^k - \gamma \cdot \left(F(z^k) - F_1(z^k)\right)$;

(ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \le e$, where $\hat{u}^k$ is the solution of:

$$\min_{u_x \in \mathcal{X}} \max_{u_y \in \mathcal{Y}} \left[\gamma f_1(u_x, u_y) + \frac{1}{2}\|u_x - v_x^k\|^2 - \frac{1}{2}\|u_y - v_y^k\|^2\right];$$

(iii) updates $z^{k+1} = \mathrm{proj}_{\mathcal{Z}}\left[u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k))\right]$ and broadcasts z^{k+1} to the workers

4: **end for**

Cenralized Algorithm

Algorithm 1 (Star Min-Max Data Similarity Algorithm)

Parameters: stepsize γ, accuracy e;
Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, for all $m \in [M]$;

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Each worker m computes $F_m(z^k)$ and sends it to the master;
3: The master node:
(i) computes $v^k = z^k - \gamma \cdot \left(F(z^k) - F_1(z^k)\right)$;
(ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \leq e$, where $\hat{u}^k$ is the solution of:

$$\min_{u_x \in \mathcal{X}} \max_{u_y \in \mathcal{Y}} \left[\gamma f_1(u_x, u_y) + \frac{1}{2}\|u_x - v_x^k\|^2 - \frac{1}{2}\|u_y - v_y^k\|^2\right];$$

(iii) updates $z^{k+1} = \text{proj}_{\mathcal{Z}}\left[u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k))\right]$ and broadcasts z^{k+1} to the workers
4: **end for**

$$\boxed{F_m(z) := \begin{pmatrix} \nabla_x f_m(x,y) \\ -\nabla_y f_m(x,y) \end{pmatrix}}$$

Cenralized Algorithm

each worker need to compute and send $F_m(z^k), F_m(u^k)$ to the master node

Algorithm 1 (Star Min-Max Data Similarity Algorithm)

Parameters: stepsize γ, accuracy e;

Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, for all $m \in [M]$;

for $k = 0, 1, 2, \dots$ **do**

Each worker m computes $F_m(z^k)$ and sends it to the master;

The master node:

(i) computes $v^k = z^k - \gamma \cdot (F(z^k) - F_1(z^k))$;

(ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \le e$, where $\hat{u}^k$ is the solution of:

$$\min_{u_x \in \mathcal{X}} \max_{u_y \in \mathcal{Y}} \left[\gamma f_1(u_x, u_y) + \frac{1}{2}\|u_x - v_x^k\|^2 - \frac{1}{2}\|u_y - v_y^k\|^2 \right];$$

(iii) updates $z^{k+1} = \text{proj}_{\mathcal{Z}}\left[u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k))\right]$ and broadcasts z^{k+1} to the workers

end for

$$\boxed{F_m(z) := \begin{pmatrix} \nabla_x f_m(x,y) \\ -\nabla_y f_m(x,y) \end{pmatrix}}$$

Cenralized Algorithm

each worker need to compute and send $F_m(z^k), F_m(u^k)$ to the master node

Algorithm 1 (Star Min-Max Data Similarity Algorithm)

Parameters: stepsize γ, accuracy e;
Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, for all $m \in [M]$;
1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Each worker m computes $F_m(z^k)$ and sends it to the master;
3: The master node:
(i) computes $v^k = z^k - \gamma \cdot (F(z^k) - F_1(z^k))$;
(ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \le e$, where $\hat{u}^k$ is the solution of:

$$\boxed{\min_{u_x \in \mathcal{X}} \max_{u_y \in \mathcal{Y}} \left[\gamma f_1(u_x, u_y) + \frac{1}{2}\|u_x - v_x^k\|^2 - \frac{1}{2}\|u_y - v_y^k\|^2\right];}$$

(iii) updates $z^{k+1} = \text{proj}_{\mathcal{Z}}\left[u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k))\right]$ and broadcasts z^{k+1} to the workers
4: **end for**

$$\boxed{F_m(z) := \begin{pmatrix} \nabla_x f_m(x,y) \\ -\nabla_y f_m(x,y) \end{pmatrix}}$$

main computations on server

Cenralized Algorithm

each worker need to compute and send $F_m(z^k), F_m(u^k)$ to the master node

Algorithm 1 (Star Min-Max Data Similarity Algorithm)

Parameters: stepsize γ, accuracy e;

Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, for all $m \in [M]$;

for $k = 0, 1, 2, \ldots$ **do**

  Each worker m computes $F_m(z^k)$ and sends it to the master;

  The master node:

  (i) computes $v^k = z^k - \gamma \cdot (F(z^k) - F_1(z^k))$;

  (ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \leq e$, where $\hat{u}^k$ is the solution of:

$$\boxed{\min_{u_x \in \mathcal{X}} \max_{u_y \in \mathcal{Y}} \left[\gamma f_1(u_x, u_y) + \frac{1}{2}\|u_x - v_x^k\|^2 - \frac{1}{2}\|u_y - v_y^k\|^2 \right];}$$

  (iii) updates $z^{k+1} = \text{proj}_{\mathcal{Z}} \left[u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k)) \right]$ and broadcasts z^{k+1} to the workers

end for

$$\boxed{F_m(z) := \begin{pmatrix} \nabla_x f_m(x, y) \\ -\nabla_y f_m(x, y) \end{pmatrix}}$$

main computations on server

Ideas:

1) Sliding for composite problem

$$\boxed{f_1(x, y)} + \boxed{\frac{1}{M} \sum_{m=1}^{M} [f_m(x, y) - f_1(x, y)]}$$

Cenralized Algorithm

each worker need to compute and send $F_m(z^k), F_m(u^k)$ to the master node

Algorithm 1 (Star Min-Max Data Similarity Algorithm)

Parameters: stepsize γ, accuracy e;
Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, for all $m \in [M]$;
for $k = 0, 1, 2, \ldots$ **do**
  Each worker m computes $F_m(z^k)$ and sends it to the master;
  The master node:
   (i) computes $v^k = z^k - \gamma \cdot (F(z^k) - F_1(z^k))$;
   (ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \le e$, where $\hat{u}^k$ is the solution of:

$$\boxed{\min_{u_x \in \mathcal{X}} \max_{u_y \in \mathcal{Y}} \left[\gamma f_1(u_x, u_y) + \frac{1}{2}\|u_x - v_x^k\|^2 - \frac{1}{2}\|u_y - v_y^k\|^2\right];}$$

   (iii) updates $z^{k+1} = \text{proj}_{\mathcal{Z}}\left[u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k))\right]$ and broadcasts z^{k+1} to the workers
end for

$$\boxed{F_m(z) := \begin{pmatrix} \nabla_x f_m(x, y) \\ -\nabla_y f_m(x, y) \end{pmatrix}}$$

main computations on server

Ideas:
1) Sliding for composite problem

$$\boxed{f_1(x, y)} + \boxed{\frac{1}{M}\sum_{m=1}^{M}[f_m(x, y) - f_1(x, y)]}$$

δ-smooth and non-convex-non-concave

Cenralized Algorithm

each worker need to compute and send $F_m(z^k), F_m(u^k)$ to the master node

Algorithm 1 (Star Min-Max Data Similarity Algorithm)

Parameters: stepsize γ, accuracy e;
Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, for all $m \in [M]$;
1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Each worker m computes $F_m(z^k)$ and sends it to the master;
3: The master node:
(i) computes $v^k = z^k - \gamma \cdot (F(z^k) - F_1(z^k))$;
(ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \le e$, where $\hat{u}^k$ is the solution of:

$$\boxed{\min_{u_x \in \mathcal{X}} \max_{u_y \in \mathcal{Y}} \left[\gamma f_1(u_x, u_y) + \frac{1}{2}\|u_x - v_x^k\|^2 - \frac{1}{2}\|u_y - v_y^k\|^2\right];}$$

(iii) updates $z^{k+1} = \text{proj}_{\mathcal{Z}}\left[u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k))\right]$ and broadcasts z^{k+1} to the workers
4: **end for**

$$\boxed{F_m(z) := \begin{pmatrix} \nabla_x f_m(x,y) \\ -\nabla_y f_m(x,y) \end{pmatrix}}$$

main computations on server

Ideas:
1) Sliding for composite problem

$$\boxed{f_1(x,y)} + \boxed{\frac{1}{M}\sum_{m=1}^{M}[f_m(x,y) - f_1(x,y)]}$$

δ-smooth and non-convex-non-concave

2) ExtraGradient + preconditioning

Convergence

Theorem. Consider distributes SPP from the 3d slide under Assumptions from the 3d and 4th slides. Let $\{z^k\}$ be the sequence generated by Algorithm. Then, given $\varepsilon > 0$, the number of communication rounds for $\|z^k - z^*\|^2 \leq \varepsilon$ is

$$\mathcal{O}\left(\left(1 + \frac{\delta}{\mu}\right) \cdot \log\left(\frac{1}{\varepsilon}\right)\right).$$

Convergence

Theorem. Consider distributes SPP from the 3d slide under Assumptions from the 3d and 4th slides. Let $\{z^k\}$ be the sequence generated by Algorithm. Then, given $\varepsilon > 0$, the number of communication rounds for $\|z^k - z^*\|^2 \le \varepsilon$ is

$$\mathcal{O}\left(\left(1 + \frac{\delta}{\mu}\right) \cdot \log\left(\frac{1}{\varepsilon}\right)\right).$$

Upper bound matches lower bound!

4. Experiments

Robust Linear Regression

Robust Linear Regression (or Linear Regression with Adversarial noise):

$$\min_{w} \max_{\|\rho\| \leq R_\rho} \frac{1}{N} \sum_{n=1}^{N} (w^T(x_n + \rho) - y_n)^2 + \frac{\lambda}{2}\|w\|^2 - \frac{\beta}{2}\|\rho\|^2$$

Robust Linear Regression

Robust Linear Regression (or Linear Regression with Adversarial noise):

data sample

$$\min_{w} \max_{\|\rho\| \leq R_\rho} \frac{1}{N} \sum_{n=1}^{N} (w^T (x_n + \rho) - y_n)^2 + \frac{\lambda}{2} \|w\|^2 - \frac{\beta}{2} \|\rho\|^2$$

weights

Robust Linear Regression

Robust Linear Regression (or Linear Regression with Adversarial noise):

data sample

$$\min_{w} \max_{\|\rho\| \leq R_\rho} \frac{1}{N} \sum_{n=1}^{N} (w^T (x_n + \rho) - y_n)^2 + \frac{\lambda}{2} \|w\|^2 - \frac{\beta}{2} \|\rho\|^2$$

weights

adversarial noise

Robust Linear Regression

Robust Linear Regression (or Linear Regression with Adversarial noise):

data sample

$$\min_{w} \max_{\|\rho\| \leq R_\rho} \frac{1}{N} \sum_{n=1}^{N} \left(w^T (x_n + \rho) - y_n\right)^2 + \frac{\lambda}{2} \|w\|^2 - \frac{\beta}{2} \|\rho\|^2$$

weights

adversarial noise

control the noise

regularizers

Robust Linear Regression

Robust Linear Regression (or Linear Regression with Adversarial noise):

data sample

$$\min_{w} \max_{\|\rho\| \le R_\rho} \frac{1}{N} \sum_{n=1}^{N} (w^T(x_n + \rho) - y_n)^2 + \frac{\lambda}{2}\|w\|^2 - \frac{\beta}{2}\|\rho\|^2$$

weights

adversarial noise

control the noise

regularizers

For comparison, we use Distributed ExtraGradient method (SOTA and optimal for general SPPs)

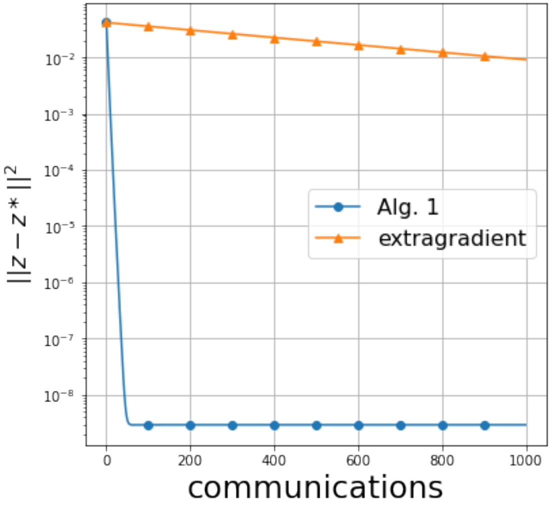
Generated data

In generated data we can control similarity parameter and observe how convergence changes depending on this parameter:

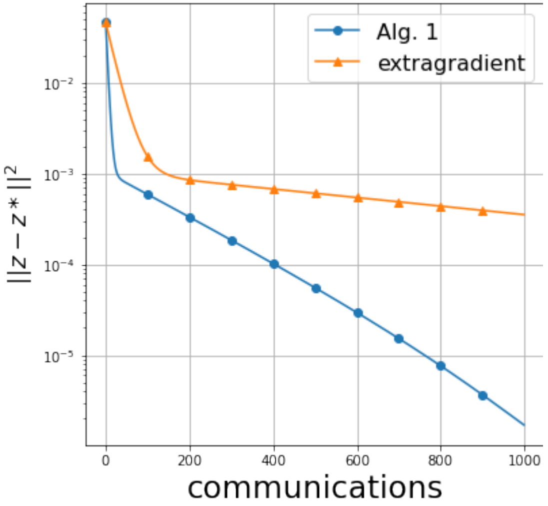
Generated data

In generated data we can control similarity parameter and observe how convergence changes depending on this parameter:

small similarity parameter



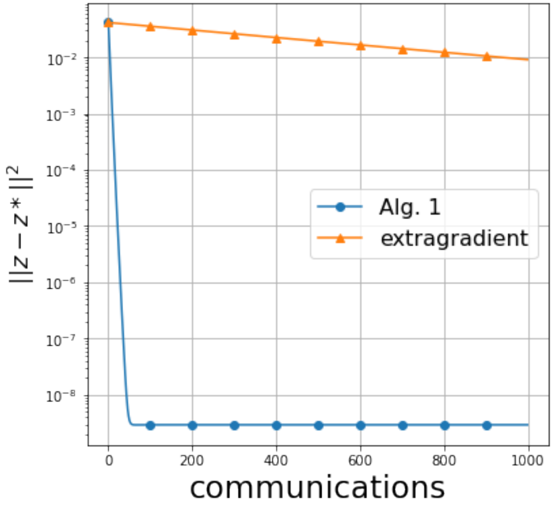
large similarity parameter



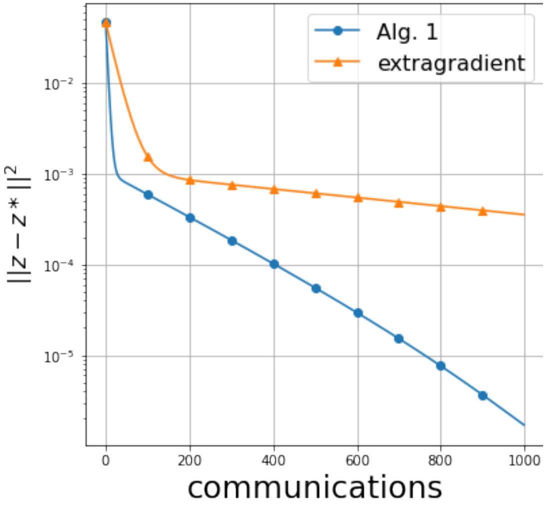
Generated data

In generated data we can control similarity parameter and observe how convergence changes depending on this parameter:

small similarity parameter



large similarity parameter



small similarity parameter = very similar data = faster convergence

The End