Convergence Analysis of Decentralized Hessian-/Jacobian-Free Algorithm for Nonconvex Stochastic Bi-Level Optimization

Yihan Zhang[1], Xinwen Zhang[1], My T. Thai[2], Jie Wu[1], Hongchang Gao[1]

1. Temple University
2. University of Florida

Outline

- Problem Definition
- Algorithm Design
- Convergence Rate
- Experiments

Problem Definition

- Decentralized stochastic bilevel optimization

$$\min_{x\in\mathbb{R}^{d_x}, y\in Y^*(x)} f(x,y) = \frac{1}{K}\sum_{k=1}^{K} f^{(k)}(x,y)$$

$$s.t. \quad Y^*(x) = \arg\min_{y\in\mathbb{R}^{d_y}} g(x,y) = \frac{1}{K}\sum_{k=1}^{K} g^{(k)}(x,y)\,,$$

  - The lower-level loss function is nonconvex in y but satisfies the PL condition

Problem Definition

- Limitations of existing methods
  - All decentralized bilevel optimization methods require that the lower-level loss function is **strongly-convex in y**.

  - Existing methods in the single-machine setting requires a regularization term to make the lower-level loss function become strongly convex

$$\min_{z \in \mathbb{R}^{d_y}} g(x, z) + \frac{1}{2\beta}\|z - w\|^2$$

regularization

Outline

- Problem Definition
- **Algorithm Design**
- Convergence Rate
- Experiments

Algorithm Design

- Minimax Reformulation

$$\min_{x\in\mathbb{R}^{d_x}, y\in\mathbb{R}^{d_y}} \max_{z\in\mathbb{R}^{d_y}} \mathcal{L}_\rho(x,y,z)$$
$$\triangleq \frac{1}{K}\sum_{k=1}^{K}\left(f^{(k)}(x,y)+\frac{1}{\rho}(g^{(k)}(x,y)-g^{(k)}(x,z))\right)$$

  - No need to compute second-order Hessian or Jacobian matrices
  - Approximate the original bilevel optimization well given appropriate ρ

Algorithm

Algorithm 1 Decentralized stochastic first-order gradient descent algorithm (FO-DSVRBGD)

Input: $\eta_x > 0$, $\eta_y > 0$, $\eta_z > 0$, $\gamma_x > 0$, $\gamma_y > 0$, $\gamma_z > 0$, $\rho > 0$. Initialization on each worker $k \in \{1, \cdots, K\}$:

$$x_0^{(k)} = x_0, \quad y_0^{(k)} = y_0, \quad z_0^{(k)} = z_0,$$
$$m_{x,1,0}^{(k)} = \nabla_1 f^{(k)}(x_0^{(k)}, y_0^{(k)}; \xi_0^{(k)}),$$
$$m_{x,2,0}^{(k)} = \nabla_1 g^{(k)}(x_0^{(k)}, y_0^{(k)}; \zeta_0^{(k)}),$$
$$m_{x,3,0}^{(k)} = \nabla_1 g^{(k)}(x_0^{(k)}, z_0^{(k)}; \zeta_0^{(k)}),$$
$$m_{y,1,0}^{(k)} = \nabla_2 f^{(k)}(x_0^{(k)}, y_0^{(k)}; \xi_0^{(k)}),$$
$$m_{y,2,0}^{(k)} = \nabla_2 g^{(k)}(x_0^{(k)}, y_0^{(k)}; \zeta_0^{(k)}),$$
$$m_{z,1,0}^{(k)} = \nabla_2 g^{(k)}(x_0^{(k)}, z_0^{(k)}; \zeta_0^{(k)}),$$
$$U_{x,0} = M_{x,0}, \quad U_{y,0} = M_{y,0}, \quad U_{z,0} = M_{z,0}.$$

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

2: Update three variables:

$$X_{t+1} = X_t W - \eta_x U_{x,t},$$
$$Y_{t+1} = Y_t W - \eta_y U_{y,t},$$
$$Z_{t+1} = Z_t W - \eta_z U_{z,t},$$

3: Compute three gradient estimators:

$$M_{x,t+1} = M_{x,1,t+1} + \frac{1}{\rho}(M_{x,2,t+1} - M_{x,3,t+1}),$$
$$M_{y,t+1} = M_{y,1,t+1} + \frac{1}{\rho} M_{y,2,t+1},$$
$$M_{z,t+1} = \frac{1}{\rho} M_{z,1,t+1},$$

4: Perform gradient tracking:

$$U_{x,t+1} = U_{x,t} W + M_{x,t+1} - M_{x,t},$$
$$U_{y,t+1} = U_{y,t} W + M_{y,t+1} - M_{y,t},$$
$$U_{z,t+1} = U_{z,t} W + M_{z,t+1} - M_{z,t},$$

5: **end for**

Update three variables

Only first-order gradients

Gradient tracking step

Outline

- Problem Definition
- Algorithm Design
- **Convergence Rate**
- Experiments

Convergence Rate

- Convergence rate

Theorem 4.6. *Suppose Assumptions 4.1-4.5 hold, when* $\gamma_x = O\left(\frac{\kappa^2}{(1-\lambda)^4\rho^2 K}\right)$, $\gamma_y = O\left(\frac{1}{(1-\lambda)^4\rho^2 K}\right)$, $\gamma_z = O\left(\frac{1}{(1-\lambda)^4\rho^2 K}\right)$, $\eta_x = O\left(\frac{K(1-\lambda)^2\epsilon^3}{\kappa^9}\right)$, $\eta_y = O\left(\frac{K(1-\lambda)^2\epsilon^3}{\kappa^7}\right)$, $\eta_z = O\left(\frac{K(1-\lambda)^2\epsilon^3}{\kappa^7}\right)$, $S = O\left(\frac{\kappa^7}{\epsilon^3}\right)$, $T = O\left(\frac{\kappa^9}{K(1-\lambda)^2\epsilon^5}\right)$, $\rho = O\left(\frac{\epsilon}{\kappa^3}\right)$, *Algorithm 1 can achieve the* ϵ*-accuracy solution:* $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\bar{x}_t)\|^2] \leq O(\epsilon^2)$, *where* $\epsilon > 0$, S *is the batch size in the initial iteration.*

  - Linear speedup
  - Better dependence on the condition number

Convergence Analysis

- Novelties when measuring the optimality of y and z
  - Our method use

$$\frac{1}{\rho}\mathbb{E}[h_\rho(\bar{x}_t, \bar{y}_t) - \hat{h}_\rho(\bar{x}_t)] ,$$
$$\frac{1}{\rho}\mathbb{E}[g(\bar{x}_t, \bar{z}_t) - \hat{g}(\bar{x}_t)] ,$$

  - Existing methods in the single-machine setting use

$$\|y_t - y_\rho^*(x)\|^2$$
$$\|z_t - y^*(x)\|^2$$

Convergence Analysis

- Key Lemmas

Lemma 4.10. *Suppose Assumptions 4.1-4.5 hold, when* $\eta_x \le \eta_y \frac{\mu^2}{4L_{h_\rho}^2}$ *and* $\eta_y \le \frac{\rho}{2L_{h_\rho}}$, *where* $L_{h_\rho} = \rho L_f + L_g$, *we have that*

$$\frac{1}{\rho}\mathbb{E}[h_\rho(\bar{x}_{t+1},\bar{y}_{t+1}) - \hat{h}_\rho(\bar{x}_{t+1})] \le \frac{1}{\rho}\mathbb{E}[h_\rho(\bar{x}_t,\bar{y}_t) - \hat{h}_\rho(\bar{x}_t)]$$

Function value

$$- \frac{1}{\rho^2}\frac{\eta_y}{8}\mathbb{E}[\|\nabla_2 h_\rho(\bar{x}_t,\bar{y}_t)\|^2] - \frac{\eta_y}{4}\mathbb{E}[\|\bar{m}_{y,t}\|^2]$$

$$+ \left(\frac{\eta_x}{2} + \frac{1}{\rho}\frac{\eta_x^2 L_{h_\rho}}{2} + \frac{1}{\rho}\frac{3\eta_x^2 L_{h_\rho}}{2} + \frac{1}{\rho}\frac{\eta_x^2 L_{\hat{h}_\rho}}{2}\right)\mathbb{E}[\|\bar{m}_{x,t}\|^2]$$

$$+ 2\eta_y \frac{L_{h_\rho}^2}{\rho^2}\frac{1}{K}\mathbb{E}[\|X_t - \bar{X}_t\|_F^2] + 2\eta_y \frac{L_{h_\rho}^2}{\rho^2}\frac{1}{K}\mathbb{E}[\|Y_t - \bar{Y}_t\|_F^2]$$

$$+ 4\eta_y \mathbb{E}[\|\frac{1}{K}\sum_{k=1}^{K}\nabla_2 f^{(k)}(x_t^{(k)},y_t^{(k)}) - \frac{1}{K}\sum_{k=1}^{K} m_{y,1,t+1}^{(k)}\|^2]$$

$$+ 4\eta_y \frac{1}{\rho^2}\mathbb{E}[\|\frac{1}{K}\sum_{k=1}^{K}\nabla_2 g^{(k)}(x_t^{(k)},y_t^{(k)}) - \frac{1}{K}\sum_{k=1}^{K} m_{y,2,t+1}^{(k)}\|^2]$$

Convergence Analysis

- Key Lemmas

Lemma 4.11. *Suppose Assumptions 4.1-4.5 hold, when* $\eta_x \leq \frac{\mu^2}{4L_g^2}\eta_z$ *and* $\eta_z \leq \frac{\rho}{2L_g}$, *we have that*

$$\frac{1}{\rho}\mathbb{E}[g(\bar{x}_{t+1}, \bar{z}_{t+1}) - \hat{g}(\bar{x}_{t+1})] \leq \frac{1}{\rho}\mathbb{E}[g(\bar{x}_t, \bar{z}_t) - \hat{g}(\bar{x}_t)]$$

Function value

$$- \frac{1}{\rho^2}\frac{\eta_z}{8}\mathbb{E}[\|\nabla_2 g(\bar{x}_t, \bar{z}_t)\|^2] - \frac{\eta_z}{4}\mathbb{E}[\|\bar{m}_{z,t}\|^2]$$

$$+ \left(\frac{\eta_x}{2} + 2\eta_x^2 L_g \frac{1}{\rho} + \frac{\eta_x^2 L_{\hat{g}}}{2}\frac{1}{\rho}\right)\mathbb{E}[\|\bar{m}_{x,t}\|^2]$$

$$+ 2\eta_z \frac{L_g^2}{\rho^2}\frac{1}{K}\mathbb{E}[\|X_t - \bar{X}_t\|_F^2] + 2\eta_z \frac{L_g^2}{\rho^2}\frac{1}{K}\mathbb{E}[\|Z_t - \bar{Z}_t\|_F^2]$$

$$+ \frac{2\eta_z}{\rho^2}\mathbb{E}[\|\frac{1}{K}\sum_{k=1}^{K}\nabla_2 g^{(k)}(x_t^{(k)}, z_t^{(k)}) - \frac{1}{K}\sum_{k=1}^{K} m_{z,1,t}^{(k)}\|^2] .$$

Outline

- Problem Definition
- Algorithm Design
- Convergence Rate
- **Experiments**

Experiments

- Hyperparameter Optimization
  - Lower-level learns model parameters
  - Upper-level learns hyperparameters

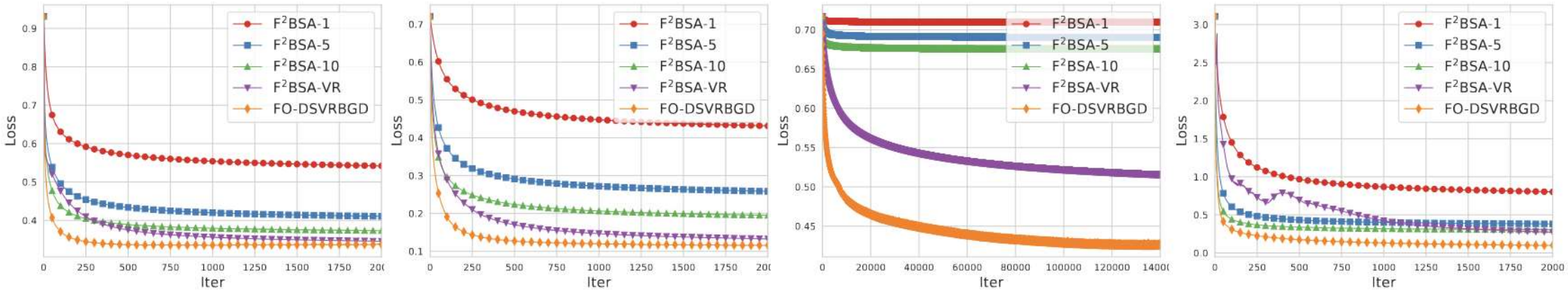
$$\min_{x} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \ell(y^*(x); a_{v,i}^{(k)}, b_{v,i}^{(k)})$$

$$s.t.\ y^*(x) = \arg\min_{y} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \ell(y; a_{t,i}^{(k)}, b_{t,i}^{(k)})$$

$$+ \frac{1}{d_1 d_2} \sum_{p=1}^{d_1} \sum_{q=1}^{d_2} \exp(x_{1,q}) y_{1,pq}^2 + \frac{1}{d_2 d_3} \sum_{p=1}^{d_2} \sum_{q=1}^{d_3} \exp(x_{2,q}) y_{2,pq}^2$$

Experiments

- Loss function value versus iterations

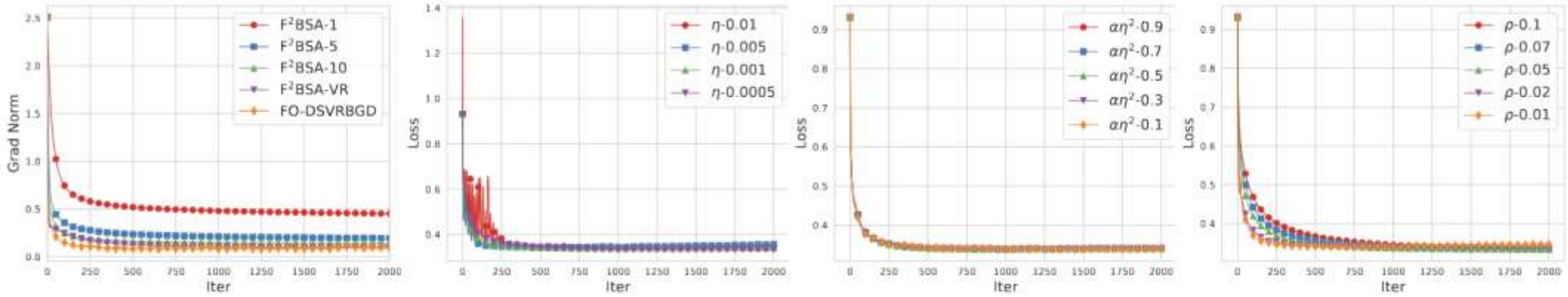


(a) a9a (b) w8a (c) covtype (d) MNIST

Figure 1. The upper-level loss function value with respect to the number of iterations.

Experiments

- Ablation Studies



(a) Gradient Norm (b) Learning rate (c) Momentum coefficient (d) Penalty parameter

Figure 2. (a) The gradient norm of upper-level loss function versus iterations for a9a dataset. (b-c) The upper-level loss function value versus iterations under different settings for a9a dataset.

Conclusions

- Algorithm Design
  - Fully first-order gradient
  - No need to compute second-order Hessian or Jacobian matrices

- Convergence Analysis
  - Measure lower-level optimality based on function values
  - Only require the pure PL condition
  - No need to use the regularization term to introduce strong convexity