Solving a Class of Non-Convex Minimax Optimization in Federated Learning

Xidong Wu*[1], Jianhui Sun*[2], Zhengmian Hu[3], Aidong Zhang[2], Heng Huang[3]

[1] University of Pittsburgh [2] University of Virginia [3] University of Maryland

Background

The minimax problems arise in many machine learning applications such as,

- Adversarial Training

$$\min_{w} \sum_{i=1}^{n} \max_{y_i \in \mathcal{Y}} \mathcal{L}\left(f\left(a_i + y_i; w\right), b_i\right), \mathcal{Y} = \{y_i \in \mathbb{R}^d \mid \|y_i\|_\infty \leq \varepsilon, i \in [n]\}$$

- Generative adversarial network

$$\min_{G} \max_{D} \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Policy Evaluation

$$\min_{\theta \in \Theta} \max_{\omega \in \mathbb{R}^d} \mathbb{E}_{s,a,s'} \left[\langle \delta \nabla_\theta V(s;\theta), \omega \rangle - \frac{1}{2} \omega^T \left(\nabla_\theta V(s;\theta) \nabla_\theta V(s;\theta)^T \right) \omega \right]$$

- Online AUPRC maximization

$$\min_{\mathbf{w} \in \mathbb{R}^d, (a,b) \in \mathbb{R}^2} \max_{\alpha \in \mathbb{R}} f(\mathbf{w}, a, b, \alpha)$$

$$f(\mathbf{w}, a, b, \alpha) = (1-p)(h(\mathbf{w};\mathbf{x}) - a)^2 \mathbb{I}_{[y=1]} + p(h(\mathbf{w};\mathbf{x}) - b)^2 \mathbb{I}_{[y=-1]} + 2(1+\alpha)\left(p h(\mathbf{w};\mathbf{x}) \mathbb{I}_{[y=-1]} - (1-p) h(\mathbf{w};\mathbf{x}) \mathbb{I}_{[y=1]}\right) - p(1-p)\alpha^2$$

- Fair Classification

$$\min_{w} \max_{y \in \mathcal{Y}} \left\{ \sum_{i=1}^{3} y_i \mathcal{L}_i(w) + g(w) - h(y) \right\}, \mathcal{Y} = \left\{ y_i \geq 0, \sum_{i=1}^{3} y_i = 1 \right\}$$

Challenges

To address the large-scale distributed data challenges across multiple clients with communication-efficient distributed training, federated learning (FL) is gaining popularity. Many optimization algorithms for minimax problems have been developed in the centralized setting. Nonetheless, the algorithm for minimax problems under FL is still underexplored. In this paper, we study a class of federated nonconvex minimax optimization problems.

We focus on the most common minimax settings and propose novel FL algorithms (FedSGDA+ and FedSGDA-M) and obtain the best known complexity results for various settings. For nonconvex-concave problems, we propose FedSGDA+ and reduce the communication complexity to $O(\varepsilon^{-6})$. Under nonconvex-strongly-concave and nonconvex-PL minimax settings, we prove that FedSGDA-M has the best-known sample complexity of $O(\kappa^3 N^{-1} \varepsilon^{-3})$ and the best-known communication complexity of $O(\kappa^2 \varepsilon^{-2})$. FedSGDA-M is the first algorithm to match the best sample complexity $O(\varepsilon^{-3})$.

Existing Methods

Complexity comparison of existing nonconvex federated minimax algorithms for finding an ε-stationary point. Sample complexity is the total number of the First-order Oracle (IFO) to reach an ε-stationary point. Communication complexity denotes the total number of back-and-forth communication times between clients and the server. Here, N is the number of clients, and $\kappa = L_f/\mu$ is the condition number.

S1 - Nonconvex Concave, S2 - Nonconvex Strongly Concave
S3 - Nonconvex PL

Type	Algorithm	Sample	Communication
S1	Local SGDA+[1]	$O\left(N^{-1}\varepsilon^{-8}\right)$	$O(\varepsilon^{-7})$
S1	FedSGDA+[1]	$O\left(N^{-1}\varepsilon^{-8}\right)$	$O\left(\varepsilon^{-6}\right)$
S2	Local SGDA[2]	$O\left(\kappa^4 N^{-1}\varepsilon^{-4}\right)$	$O(\kappa^3\varepsilon^{-3})$
S2	Momentum SGDA[2]	$O\left(\kappa^4 N^{-1}\varepsilon^{-4}\right)$	$O(\kappa^3\varepsilon^{-3})$
S2	FEDNEST[2]	$O\left(\kappa^3\varepsilon^{-4}\right)$	$O\left(\kappa^2\varepsilon^{-4}\right)$
S2	FedSGDA[2]	$O\left(\kappa^3 N^{-1}\varepsilon^{-3}\right)$	$O\left(\kappa^2\varepsilon^{-2}\right)$
S3	Local SGDA[3]	$O\left(\kappa^4 N^{-1}\varepsilon^{-4}\right)$	$O(\kappa^3\varepsilon^{-3})$
S3	Momentum SGDA[3]	$O\left(\kappa^4 N^{-1}\varepsilon^{-4}\right)$	$O(\kappa^3\varepsilon^{-3})$
S3	SAGDA[3]	$O\left(N^{-1}\varepsilon^{-4}\right)$	$O(\varepsilon^{-2})$
S3	FedSGDA[3]	$O\left(\kappa^3 N^{-1}\varepsilon^{-3}\right)$	$O\left(\kappa^2\varepsilon^{-2}\right)$

FedSGDA+ Algorithm

Algorithm 1 FedSGDA+ Algorithm

```
for t = 0, 1, ..., T-1 do
  for i = 1, 2, ..., N do
    Local Update:
    for q = 0, 1, ..., Q-1 do
      Draw mini-batch samples B^i_{t,q} = {ξ^j_i}^b_{j=1} with |B^i_t| = b from D_i locally
      x^i_{t,q+1} = x^i_{t,q} - ĉ ∇_x f_i(x^i_{t,q}, y^i_{t,q}; B^i_{t,q})
      y^i_{t,q+1} = y^i_{t,q} + c ∇_y f_i(x̃_k, y^i_{t,q}; B^i_{t,q})
    end for
  end for
  x^i_{t+1,0} = x̄_{t+1} = x̄_t + η_x (1/N) Σ^N_{i=1} (x^i_{t,Q} - x̄_t)
  y^i_{t+1,0} = ȳ_{t+1} = ȳ_t + η_y (1/N) Σ^N_{i=1} (y^i_{t,Q} - ȳ_t)
  if mod (t+1, S) = 0 then
    k = k + 1
    x̃_k = x̄_{t+1}
  end if
end for
Output: x and y chosen uniformly random from {(x̄_t, ȳ_t)}^T_{t=1}.
```

FedSGDA-M Algorithm

Algorithm 2 FedSGDA-M Algorithm

```
for t = 1, 2, ..., T do
  for i = 1, 2, ..., N do
    if mod (t, Q) = 0 then
      Sever Update:
        u^i_t = ū_t = (1/N) Σ^N_{j=1} u^j_t,  v^i_t = v̄_t = (1/N) Σ^N_{j=1} v^j_t
        x^i_t = x̄_t = (1/N) Σ^N_{j=1} (x^j_{t-1} - ĉη u^j_t)
        y^i_t = ȳ_t = (1/N) Σ^N_{j=1} (y^j_{t-1} + cη v^j_t)
    else
        x^i_t = x^i_{t-1} - ĉη u^i_t,  y^i_t = y^i_{t-1} + cη v^i_t
    end if
    Draw mini-batch samples B^i_t = {ξ^j_i}^b_{j=1} with |B^i_t| = b
    u^i_{t+1} = ∇_x f_i(x^i_t, y^i_t; B^i_t) + (1-α)(u^i_t - ∇_x f_i(x^i_{t-1}, y^i_{t-1}; B^i_t))
    v^i_{t+1} = ∇_y f_i(x^i_t, y^i_t; B^i_t) + (1-β)(v^i_t - ∇_y f_i(x^i_{t-1}, y^i_{t-1}; B^i_t))
  end for
end for
```

Experiments

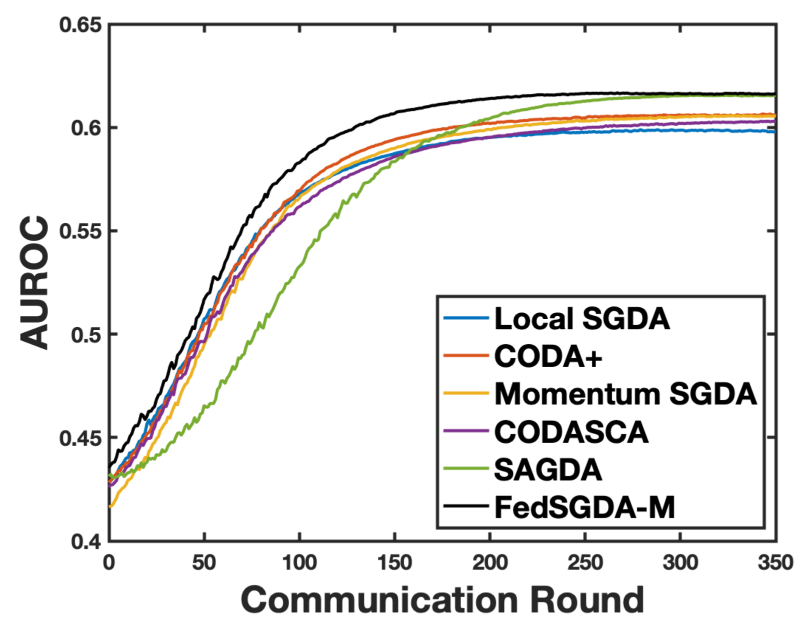

Figure 1: Cifar10 Fair Classification

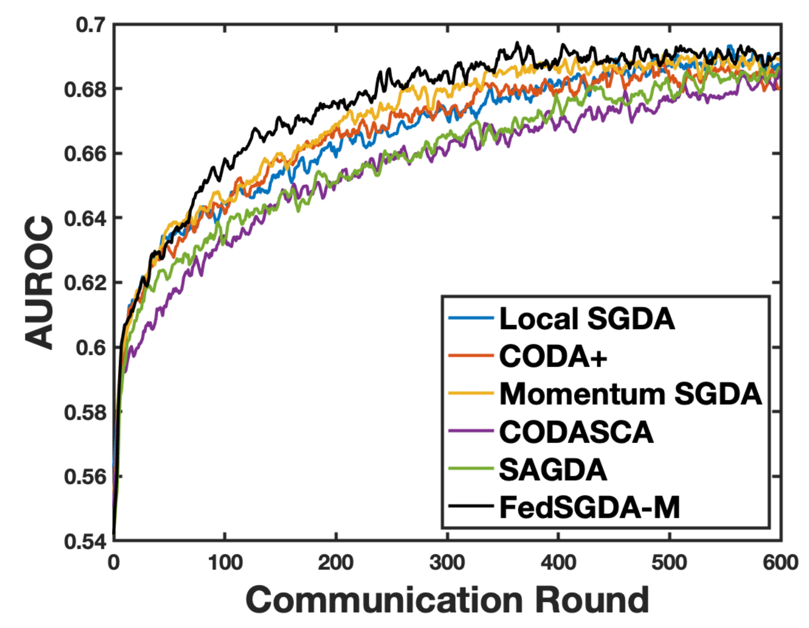

Figure 2: Tiny ImageNet

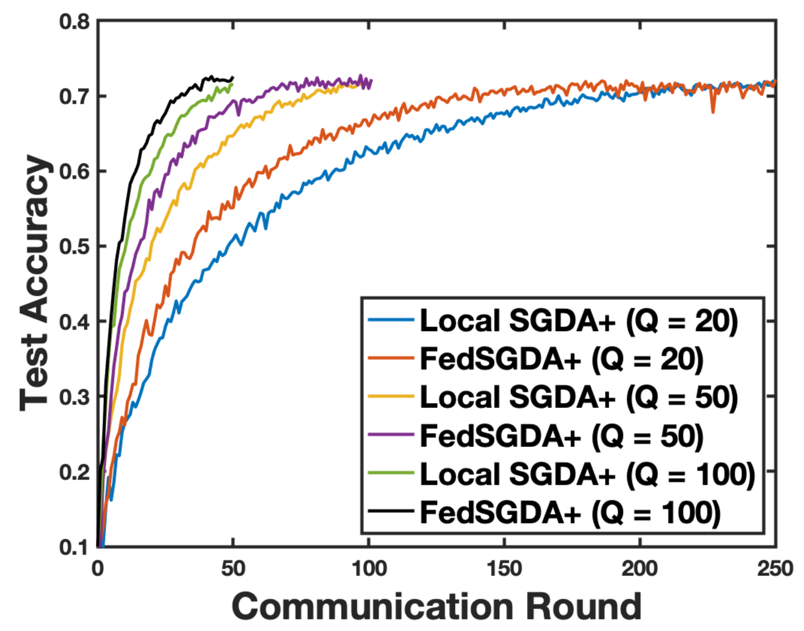

Figure 3: CIFAR10 AUROC

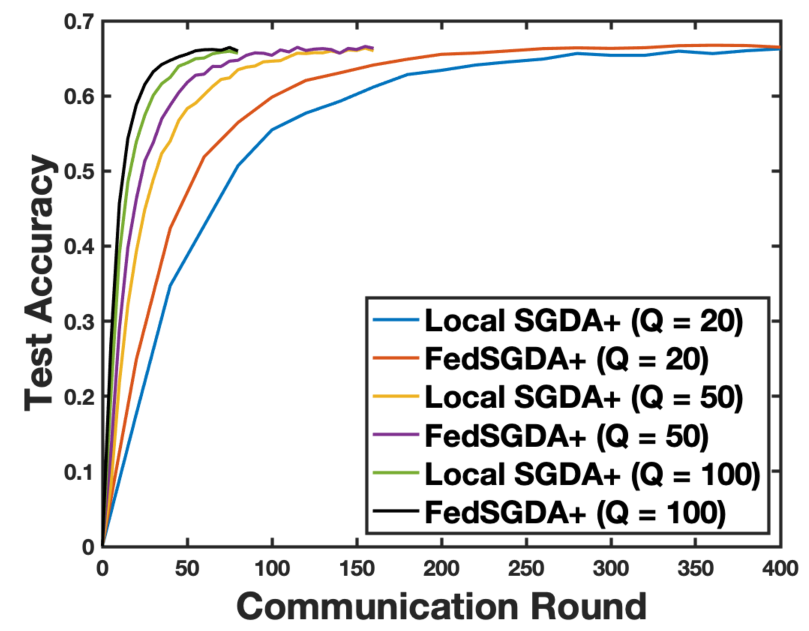

Figure 4: Tiny ImageNet