Federated Multi-Objective Learning

Haibo Yang[1], Zhuqing Liu[2], Jia Liu[2], Chaosheng Dong[3], Michinari Momma[3]

[1]Rochester Institute of Technology, [2]The Ohio State University, [3]Amazon

Overview

In this work, we propose a new **federated multi-objective learning (FMOL)** framework with multiple clients distributively and collaboratively solving an MOO problem while keeping their training data private.

- Our FMOL framework allows a different set of objective functions across different clients to support a wide range of applications, which advances and generalizes the MOO formulation to the federated learning paradigm.
- For this FMOL framework, we propose two new federated multi-objective optimization (FMOO) algorithms called federated multi-gradient descent averaging (FMGDA) and federated stochastic multi-gradient descent averaging (FSMGDA).
- Both algorithms allow local updates to significantly reduce communication costs, while achieving the same convergence rates as those of their algorithmic counterparts in centralized learning.

Problem Formulation

For a system with M clients and S tasks (objectives) in total, our FMOL framework can be written as follows:

$$\min_{\mathbf{x}} \quad \mathrm{Diag}(\mathbf{F}\mathbf{A}^\top),$$

$$\mathbf{F} \triangleq \begin{bmatrix} f_{1,1} & \cdots & f_{1,M} \\ \vdots & \ddots & \vdots \\ f_{S,1} & \cdots & f_{S,M} \end{bmatrix}_{S\times M}, \mathbf{A} \triangleq \begin{bmatrix} a_{1,1} & \cdots & a_{1,M} \\ \vdots & \ddots & \vdots \\ a_{S,1} & \cdots & a_{S,M} \end{bmatrix}_{S\times M},$$

where **F** groups all potential objectives $f_{s,i}(x)$ for each task s at each client i, and $A \in \{0,1\}^{S\times M}$ is a binary objective indicator matrix, with each element $a_{s,i} = 1$ if task s is of client i's interest and $a_{s,i} = 0$ otherwise.

1. Each client has only one distinct objective: $A = \mathrm{I}_M, S = M, Diag(FA^T) = [f_1(x), f_2(x), \ldots, f_S(x)]$. E.g., multi-task learning, classic federated learning as MOO problem.
2. All clients share the same S objectives: A is an all-one matrix. E.g., distributed MOO with decentralized data.
3. Each client has a different subset of objectives.

Algorithm

Algorithm 1 Federated (Stochastic) Multiple Gradient Descent Averaging (FMGDA/FSMGDA).

At Each Client i:

1. Synchronize local models $\mathbf{x}_{s,i}^{t,0} = \mathbf{x}_t, \forall s \in S_i$.
2. Local updates: for all $s \in S_i$, for $k = 1, \ldots, K$,
   (FMGDA): $\mathbf{x}_{s,i}^{t,k} = \mathbf{x}_{s,i}^{t,k-1} - \eta_L \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k-1})$.
   (FSMGDA): $\mathbf{x}_{s,i}^{t,k} = \mathbf{x}_{s,i}^{t,k-1} - \eta_L \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k-1}, \xi_i^{t,k})$.
3. Return accumulated updates to server $\{\Delta_{s,i}^t, s \in S_i\}$:
   (FMGDA): $\Delta_{s,i}^t = \sum_{k\in[K]} \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k})$.
   (FSMGDA): $\Delta_{s,i}^t = \sum_{k\in[K]} \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k}, \xi_i^{t,k})$.

At the Server:

4. Receive accumulated updates $\{\Delta_{s,i}^t, \forall s \in S_i, \forall i \in [M]\}$.
5. Compute $\Delta_s^t = \frac{1}{|R_s|}\sum_{i\in R_s} \Delta_{s,i}^t, \forall s \in [S]$, where $R_s = \{i : a_{s,i} = 1, i \in [M]\}$.
6. Compute $\boldsymbol{\lambda}_t^* \in [0,1]^S$ by solving
$$\min_{\boldsymbol{\lambda}_t \geq \mathbf{0}} \left\| \sum_{s\in[S]} \lambda_s^t \Delta_s^t \right\|^2, \quad \text{s.t.} \sum_{s\in[S]} \lambda_s^t = 1. \tag{3}$$
7. Let $\mathbf{d}_t = \sum_{s\in[S]} \lambda_s^{t,*} \Delta_s^t$ and update the global model as: $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{d}_t$, with a global learning rate η_t.

- Local update: communication efficient
- Two-sided learning rates: local learning rate controls the derivations and noise, global learning rate manage the learning progress
- Convex quadratic optimization for common direction

Convergence Rates

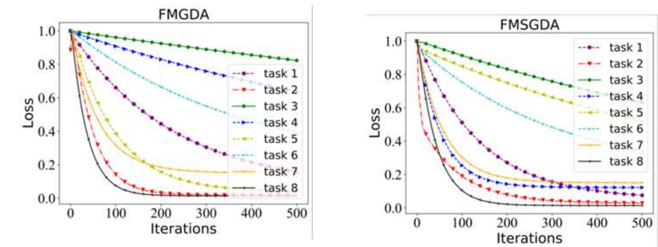
Table 1: Convergence rate results (shaded parts are our results) comparisons.

Methods	Strongly Convex		Non-convex	
	Rate	Assumption*	Rate	Assumption*
MGD [7]	$\mathcal{O}(r^T)$ #	Linear search & sequence convergence	$\mathcal{O}(1/T)$	Linear search & sequence convergence
SMGD [8]	$\mathcal{O}(1/T)$	First moment bound & Lipschitz continuity of λ	Not provided	Not provided
FMGDA	$\mathcal{O}(\exp(-\mu T))$ #	Not needed	$\mathcal{O}(1/T)$	Not needed
FSMGDA	$\tilde{\mathcal{O}}(1/T)$	(α, β)-Lipschitz continuous stochastic gradient	$\mathcal{O}(1/\sqrt{T})$	(α, β)-Lipschitz continuous stochastic gradient

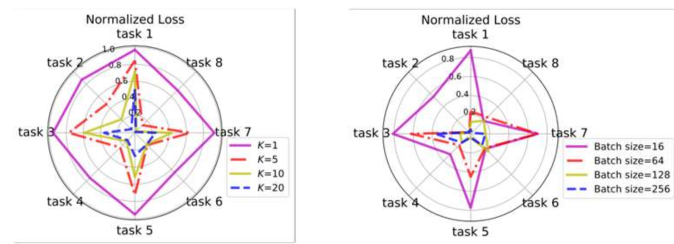
#Notes on constants: μ is the strong convexity modulus; r is a constant depends on μ, s.t., $r \in (0,1)$.
*Assumption short-hands: "Linear search": learning rate linear search [7]; "Sequence convergence": $\{\mathbf{x}_t\}$ converges to $\mathbf{x}^*$ [7]; "First moment bound" (Asm. 5.2(b) [8]): $\mathbb{E}[\|\nabla f(\mathbf{x}, \xi) - \nabla f(\mathbf{x})\|] \leq \eta(a + b\|\nabla f(\mathbf{x})\|)$;"Lipschitz continuity of λ" (Asm. 5.4 [8]): $\|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_t\| \leq \beta \left\| [(\nabla f_1(\mathbf{x}_k) - \nabla f_1(\mathbf{x}_t))^T, \ldots, (\nabla f_S(\mathbf{x}_k) - \nabla f_S(\mathbf{x}_t))^T] \right\|$; "$(\alpha, \beta)$-Lipschitz continuous stochastic gradient": see Asm. 4.

Assumption 4 ((α, β)-Lipschitz Continuous Stochastic Gradient). *A function f has (α, β)-Lipschitz continuous stochastic gradients if there exist two constants $\alpha, \beta > 0$ such that, for any two independent training samples ξ and ξ', $\mathbb{E}[\|\nabla f(\mathbf{x}, \xi) - \nabla f(\mathbf{y}, \xi')\|^2] \leq \alpha\|\mathbf{x} - \mathbf{y}\|^2 + \beta\sigma^2$.*

Numerical Results



Effectiveness of FMOL algorithms: River Flow dataset with 8 tasks.



Ablation study: local steps and batch size

Conclusion

- Proposed a general Federated Multi-Objective Learning (FMOL) framework..
- Proposed two federated (stochastic) multi-gradient descent averaging algorithms with theoretical guarantees.

Acknowledgments

This work has been supported in part by NSF grants CAREER CNS-2110259 and CNS-2112471.