Riemannian Federated Learning via Averaging Gradient Stream

Speaker: Wen Huang

Xiamen University

Nov. 14. 2024

Joint work with Zhenwei Huang, Pratik Jawanpuria, and Bamdev Mishra

- 1. Federated Learning Review
- 2. Riemannian Federated Learning Averaging Gradient Stream
- 3. Convergence Analysis
- 4. Numerical Experiments
- 5. Summary

Outline

1. Federated Learning Review

- 2. Riemannian Federated Learning Averaging Gradient Stream
- 3. Convergence Analysis
- 4. Numerical Experiments
- 5. Summary

General Federated Learning Optimization:

$$\min_{x\in\mathbb{R}^n}F(x)=\sum_{i=1}^{S}p_if_i(x), \text{ with } p_i\geq 0 \text{ and } \sum_{i=1}^{S}p_i=1, \tag{1.1}$$

- S is the number of agents;
- *f_i* is the local objective of agent *i*, and covers

$$f_i(x) = \begin{cases} \mathbb{E}_{\xi \sim \mathcal{D}_i}[f_i(x;\xi)] & \text{with } \mathcal{D}_i \text{ being a local data distribution} \\ \frac{1}{S_i} \sum_{i=1}^{S_i} f_i(x;z_{i,j}) & \text{with } \mathcal{D}_i = \{z_{i,1}, \dots, z_{i,S_i}\} \text{ being a local dataset;} \end{cases}$$

General Federated Learning Optimization:

$$\min_{x \in \mathcal{M}} F(x) = \sum_{i=1}^{S} p_i f_i(x), \text{ with } p_i \ge 0 \text{ and } \sum_{i=1}^{S} p_i = 1,$$
 (1.1)

- S is the number of agents;
- f_i is the local objective of agent i, and covers

 $f_i(x) = \begin{cases} \mathbb{E}_{\xi \sim \mathcal{D}_i}[f_i(x;\xi)] & \text{with } \mathcal{D}_i \text{ being a local data distribution} \\ \frac{1}{S_i} \sum_{i=1}^{S_i} f_i(x;z_{i,i}) & \text{with } \mathcal{D}_i = \{z_{i,1}, \dots, z_{i,S_i}\} \text{ being a local dataset;} \end{cases}$

Riemannian Federated Learning considers (1.1) with x in a manifold \mathcal{M}



General Federated Learning Optimization:

$$\min_{x \in \mathcal{M}} F(x) = \sum_{i=1}^{S} p_i f_i(x), \text{ with } p_i \ge 0 \text{ and } \sum_{i=1}^{S} p_i = 1,$$
(1.1)

- S is the number of agents;
- f_i is the local objective of agent i, and covers

 $f_i(x) = \begin{cases} \mathbb{E}_{\xi \sim \mathcal{D}_i}[f_i(x;\xi)] & \text{with } \mathcal{D}_i \text{ being a local data distribution} \\ \frac{1}{S_i} \sum_{i=1}^{S_i} f_i(x;z_{i,j}) & \text{with } \mathcal{D}_i = \{z_{i,1}, \dots, z_{i,S_i}\} \text{ being a local dataset;} \end{cases}$

Riemannian Federated Learning considers (1.1) with x in a manifold \mathcal{M}

Applications:

- Matrix completion
- Principal component analysis
- Online learning
- Taxonomy embedding
- etc





Figure 1: Flowchart of a federated learning algorithm

Euclidean version:

Algorithm: A representative federated averaging algorithm [McM+17]

- 1. for t = 0, 1, ..., T 1 do
- 2. The server uniformly selects a subset S_t of *s* agents at random;
- 3. The server upload global parameter \tilde{x}_t to all agents in S_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$;
- 4. for $j \in S_t$ in parallel do
- 5. Agent *j* updates a local parameter $x_{t,K}^j$ by *K*-step SGD with \tilde{x}_t being initial iterate; Sent $x_{t,K}^j$ to the correct $\min_{x_i \in \mathbb{R}^n} f_i(x)$
- 6. Sent $x_{t,K}^{j}$ to the server;
- 7. end for
- 8. Server aggregates the received local parameters $\{x_{t,K}^j\}_{j \in S_t}$ by averaging

$$\tilde{x}_{t+1} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} x_{t,K}^j;$$

9. end for

- Sever: Steps 2, 3, and 8;
- Agents: Steps 5 and 6;

Euclidean version:

Algorithm: A representative federated averaging algorithm [McM+17]

1. for t = 0, 1, ..., T - 1 do

- 2. The server uniformly selects a subset S_t of *s* agents at random;
- 3. The server upload global parameter \tilde{x}_t to all agents in \mathcal{S}_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$;
- 4. for $j \in S_t$ in parallel do
- 5. Agent *j* updates a local parameter $x_{t,K}^{j}$ by *K*-step SGD with \tilde{x}_{t} being initial iterate; Sent $x_{t,K}^{j}$ to the converse $\min_{x_{t} \in \mathbb{R}^{n}} f_{i}(x)$
- 6. Sent $x_{t,K}^{j}$ to the server;
- 7. end for
- 8. Server aggregates the received local parameters $\{x_{t,K}^j\}_{j \in S_t}$ by averaging

$$\tilde{x}_{t+1} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} x_{t,\kappa}^j;$$

9. end for

- Sever: Steps 2, 3, and 8;
- Agents: Steps 5 and 6;

[McM+17] B. McMahan, E. Moore, D. Ramage, B. A. y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. Proceedings of Machine Learning Research, 54, P.1273-1282, 2017.

Euclidean to Riemannian

Algorithm: A Riemannian federated learning algorithm

- 1. for $t = 0, 1, \ldots, T 1$ do
- 2. The server uniformly selects a subset S_t of *s* agents at random;
- 3. The server upload global parameter \tilde{x}_t to all agents in S_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$;
- 4. for $j \in S_t$ in parallel do
- Agent *j* updates a local parameter $x_{t,K}^{j}$ by *K*-step Riemannian SGD with \tilde{x}_{t} 5. being initial iterate;
- Sent $x_{t \kappa}^{J}$ to the server; 6.
- 7. end for
- Server aggregates the received local parameters $\{x_{t,K}^j\}_{j\in S_t}$ by averaging 8.

$$\tilde{x}_{t+1} \leftarrow \operatorname{ave}(x_{t,K}^j \mid j \in \mathcal{S}_j);$$

9. end for

- Agents: Riemannian SGD [Bon13]
- Sever: Aggregation

6

$$\min_{x_j\in\mathcal{M}}f_i(x)$$

$$\check{x}_{t+1} \leftarrow \operatorname{ave}(x_{t,K}^j \mid j \in \mathcal{S}_j);$$

Euclidean to Riemannian

Algorithm: A Riemannian federated learning algorithm

- 1. for t = 0, 1, ..., T 1 do
- 2. The server uniformly selects a subset S_t of *s* agents at random;
- 3. The server upload global parameter \tilde{x}_t to all agents in \mathcal{S}_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$;
- 4. **for** $j \in S_t$ in parallel **do**
- 5. Agent *j* updates a local parameter $x_{t,K}^j$ by *K*-step Riemannian SGD with \tilde{x}_t being initial iterate;
- 6. Sent $x_{t,K}^j$ to the server;
- 7. end for
- 8. Server aggregates the received local parameters $\{x_{t,K}^{j}\}_{j\in S_{t}}$ by averaging

$$\tilde{x}_{t+1} \leftarrow \operatorname{ave}(x_{t,K}^j \mid j \in \mathcal{S}_j);$$

9. end for

- Agents: Riemannian SGD [Bon13]
- Sever: Aggregation

6

 $\min_{x_j\in\mathcal{M}}f_i(x)$

Euclidean to Riemannian

Algorithm: A Riemannian federated learning algorithm

- 1. for $t = 0, 1, \ldots, T 1$ do
- 2. The server uniformly selects a subset S_t of *s* agents at random;
- 3. The server upload global parameter \tilde{x}_t to all agents in S_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$;
- 4. for $i \in S_t$ in parallel do
- Agent *j* updates a local parameter $x_{t,K}^{j}$ by *K*-step Riemannian SGD with \tilde{x}_{t} 5. being initial iterate;
- Sent $x_{t \kappa}^{J}$ to the server; 6.
- 7. end for
- Server aggregates the received local parameters $\{x_{t,K}^j\}_{i \in S_t}$ by averaging 8.

$$\tilde{x}_{t+1} \leftarrow \operatorname{ave}(x_{t,K}^j \mid j \in \mathcal{S}_j);$$

9. end for

- Agents: Riemannian SGD [Bon13]
- Sever: Aggregation

6

$$\min_{x_j\in\mathcal{M}}f_i(x)$$

$$\tilde{x}_{t+1} \leftarrow \operatorname{ave}(x_{t,K}^j \mid j \in \mathcal{S}_j);$$

Euclidean to Riemannian

Algorithm: A Riemannian federated learning algorithm

- 1. for $t = 0, 1, \ldots, T 1$ do
- 2. The server uniformly selects a subset S_t of *s* agents at random;
- The server upload global parameter \tilde{x}_t to all agents in S_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$; 3.
- 4. for $j \in S_t$ in parallel do
- Agent *j* updates a local parameter $x_{t,K}^{j}$ by *K*-step Riemannian SGD with \tilde{x}_{t} 5. being initial iterate;
- Sent $x_{t \kappa}^{J}$ to the server; 6.
- 7. end for
- Server aggregates the received local parameters $\{x_{t,K}^{j}\}_{j\in S_{t}}$ by averaging 8.

$$\tilde{x}_{t+1} \leftarrow \operatorname{ave}(x_{t,K}^j \mid j \in \mathcal{S}_j);$$

9. end for

- Agents: Riemannian SGD [Bon13]
- Sever: Aggregation

How to aggregates $\{x_{t,\kappa}^j\}_{i \in S_t}$ on a manifold?

$$\min_{x_j\in\mathcal{M}}f_i(x)$$

$$\min_{x_j\in\mathcal{M}}f_i(x)$$

Euclidean to Riemannian (Aggregation: an existing approach):

• Naive generalization:

$$\tilde{x}_{t+1} \leftarrow \sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} x_{t,\kappa}^j
ightarrow \mathsf{Riemannian setting}$$

Euclidean to Riemannian (Aggregation: an existing approach):

• Naive generalization:

$$ilde{x}_{t+1} \leftarrow \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} x^j_{t,\kappa}
eq \mathsf{Riemannian setting}$$

• An alternative approach:

$$\begin{split} \tilde{x}_{t+1} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} x_{t,K}^j \iff \tilde{x}_{t+1} = \arg\min_{x} \sum_{j \in \mathcal{S}_j} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \|x - x_{t,K}^j\|_F^2 \\ \iff \tilde{x}_{t+1} = \arg\min_{x} \sum_{j \in \mathcal{S}_j} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \operatorname{dist}^2(x, x_{t,K}^j) \Longrightarrow \text{Riemannian setting}; \end{split}$$

Euclidean to Riemannian (Aggregation: an existing approach):

• Naive generalization:

$$ilde{x}_{t+1} \leftarrow \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} x_{t,K}^j
eq \mathsf{Riemannian setting}$$

An alternative approach:

$$\tilde{x}_{t+1} \leftarrow \sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} x_{t,K}^j \iff \tilde{x}_{t+1} = \arg\min_x \sum_{j \in S_j} \frac{p_j}{\sum_{j \in S_t} p_j} ||x - x_{t,K}^j||_F^2$$
$$\iff \tilde{x}_{t+1} = \arg\min_x \sum_{j \in S_j} \frac{p_j}{\sum_{j \in S_t} p_j} \operatorname{dist}^2(x, x_{t,K}^j) \Longrightarrow \text{Riemannian setting};$$

Euclidean to Riemannian (Aggregation: an existing approach):

Naive generalization:

$$ilde{x}_{t+1} \leftarrow \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} x^j_{t,\kappa}
eq \mathsf{Riemannian setting}$$

An alternative approach:

$$\begin{split} \tilde{x}_{t+1} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} x_{t,K}^j \iff \tilde{x}_{t+1} = \arg\min_{x} \sum_{j \in \mathcal{S}_j} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \|x - x_{t,K}^j\|_F^2 \\ \iff \tilde{x}_{t+1} = \arg\min_{x} \sum_{j \in \mathcal{S}_j} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \operatorname{dist}^2(x, x_{t,K}^j) \Longrightarrow \text{Riemannian setting}; \end{split}$$

- $\tilde{x}_{t+1} = \arg \min_{x} \sum_{j \in S_j} \frac{p_j}{\sum_{j \in S_t} p_j} \operatorname{dist}^2(x, x_{t,K}^j)$: computationally expensive;
- One step of Riemannian gradient descent (called tangent mean) [LM23]:

$$\tilde{x}_{t+1} \leftarrow \operatorname{Exp}_{\tilde{x}_t}\left(\sum_{j \in \mathcal{S}_t} \frac{\rho_j}{\sum_{i \in \mathcal{S}_t} \rho_i} \operatorname{Exp}_{\tilde{x}_t}^{-1}(x_{t,K}^j)\right);$$

[[]LM23] Jiaxiang Li and Shiqian Ma. Federated learning on Riemannian manifolds. Applied Set-Valued Analysis and Optimization, 5(2), 2023.

Existing Riemannian Federated Learning:

- Federated Learning on Riemannian Manifolds [LM23]
 - Integrate SVRG technique within Riemannian federated learning
 - Use tangent mean as the server aggregation
 - Requirements for convergence
 - Full agent participation, and one step of local update;
 - One agent participates, and multiple steps of local update.

[[]LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds. Applied Set-Valued Analysis and Optimization, 2023.

Existing Riemannian Federated Learning:

- Federated Learning on Riemannian Manifolds [LM23]
 - Integrate SVRG technique within Riemannian federated learning
 - Use tangent mean as the server aggregation
 - Requirements for convergence
 - · Full agent participation, and one step of local update;
 - One agent participates, and multiple steps of local update.
- Federated Learning on Riemannian Manifolds with Differential Privacy [Hua+24]
 - Use differential privacy to enhance the privacy of federated learning;
 - Use tangent mean as the server aggregation.
 - Requirements for convergence similar to that in [LM23].

[[]LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds. Applied Set-Valued Analysis and Optimization, 2023.

[[]Hua+24] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra. Federated Learning on Riemannian Manifolds with Differential Privacy. arxiv:2404.10029, 2024.

Existing Riemannian Federated Learning:

- Federated Learning on Riemannian Manifolds [LM23]
 - Integrate SVRG technique within Riemannian federated learning
 - Use tangent mean as the server aggregation
 - Requirements for convergence
 - · Full agent participation, and one step of local update;
 - One agent participates, and multiple steps of local update.
- Federated Learning on Riemannian Manifolds with Differential Privacy [Hua+24]
 - Use differential privacy to enhance the privacy of federated learning;
 - Use tangent mean as the server aggregation.
 - Requirements for convergence similar to that in [LM23].
- Riemannian Federated Learning on Compact Submanifolds with Heterogeneous Data [Zha+24]
 - Use projection onto the manifold
 - Allow multiple agents and multiple local updates

[Hua+24] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra. Federated Learning on Riemannian Manifolds with Differential Privacy. arxiv:2404.10029, 2024.

[Zha+24] J. Zhang and J. Hu and A. M.-C. So and M. Johansson. Nonconvex Federated Learning on Compact Smooth Submanifolds With Heterogeneous Data. arxiv:2406.08465, 2024.

[[]LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds. Applied Set-Valued Analysis and Optimization, 2023.

Limitations

- Full agent participation and one step of local update [LM23; Hua+24]
- Compact submanifolds embedded in Euclidean spaces [Zha+24]

[[]LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds. Applied Set-Valued Analysis and Optimization, 2023.

[[]Hua+24] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra. Federated Learning on Riemannian Manifolds with Differential Privacy. arxiv:2404.10029, 2024.

[[]Zha+24] J. Zhang and J. Hu and A. M.-C. So and M. Johansson. Nonconvex Federated Learning on Compact Smooth Submanifolds With Heterogeneous Data. arxiv:2406.08465, 2024.

Limitations

- Full agent participation and one step of local update [LM23; Hua+24]
- Compact submanifolds embedded in Euclidean spaces [Zha+24]

Proposed Riemannian federated learning algorithm overcomes these limitations!

[[]LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds. Applied Set-Valued Analysis and Optimization, 2023.

[[]Hua+24] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra. Federated Learning on Riemannian Manifolds with Differential Privacy. arxiv:2404.10029, 2024.

[[]Zha+24] J. Zhang and J. Hu and A. M.-C. So and M. Johansson. Nonconvex Federated Learning on Compact Smooth Submanifolds With Heterogeneous Data. arxiv:2406.08465, 2024.

2. Riemannian Federated Learning Averaging Gradient Stream

- 3. Convergence Analysis
- 4. Numerical Experiments

5. Summary

Euclidean aggregation:
$$\tilde{x}_{t+1} = \sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} x_{t,K}^j$$

 $x_{t,K}^j = x_{t,K-1}^j - \frac{\alpha_{t,K-1}}{B_{t,K-1}} \sum_{b \in B_{t,K-1}^j} \nabla f_j(x_{t,K-1}^j; \xi_{t,K-1,b}^j)$
RSGD for instance
 $= x_{t,K-2}^j - \frac{\alpha_{t,K-2}}{B_{t,K-2}} \sum_{b \in B_{t,K-2}^j} \nabla f_j(x_{t,K-2}^j; \xi_{t,K-2,b}^j) - \frac{\alpha_{t,K-1}}{B_{t,K-1}} \sum_{b \in B_{t,K-1}^j} \nabla f_j(x_{t,K-1}^j; \xi_{t,K-1,b}^j)$
 $= \cdots = x_{t,0}^j - \sum_{k=0}^{K-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in B_{t,k}^j} \nabla f_j(x_{t,k}^j; \xi_{t,k,b}^j)$
 $\Longrightarrow \tilde{x}_{t+1} - \tilde{x}_t = -\sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} \sum_{k=0}^{K-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in B_{t,k}^j} \nabla f_j(x_{t,k}^j; \xi_{t,k,b}^j).$

Euclidean aggregation:
$$\tilde{x}_{t+1} = \sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} x_{t,K}^j$$

 $-d_t = \tilde{x}_{t+1} - \tilde{x}_t = -\sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} \sum_{k=0}^{K-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in B_{t,k}^j} \nabla f_j(x_{t,k}^j; \xi_{t,k,b}^j)$

Tangent mean:
$$\tilde{x}_{t+1} = \operatorname{Exp}_{\tilde{x}_t} \left(-\sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} \operatorname{Exp}_{\tilde{x}_t}^{-1}(x_{t,K}^j) \right)$$

$$\mathbf{x}_{t,K}^{j} = \operatorname{Exp}_{\mathbf{x}_{t,K-1}^{j}} \left(-\frac{\alpha_{t,K-1}}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^{j}} \operatorname{grad} f_{j}(\mathbf{x}_{t,K-1}^{j}; \xi_{t,K-1,b}^{j}) \right)$$

$$\boldsymbol{x}_{t,1}^{j} = \operatorname{Exp}_{\tilde{\boldsymbol{x}}_{t}} \left(-\frac{\alpha_{t,0}}{\boldsymbol{B}_{t,0}} \sum_{\boldsymbol{b} \in \mathcal{B}_{t,0}^{j}} \operatorname{grad} f_{j}(\boldsymbol{x}_{t,0}^{j}; \boldsymbol{\xi}_{t,0,b}^{j}) \right)$$

. . .

Euclidean aggregation:
$$\tilde{x}_{t+1} = \sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} x_{t,K}^j$$

 $-d_t = \tilde{x}_{t+1} - \tilde{x}_t = -\sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} \sum_{k=0}^{K-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in B_{t,k}^j} \nabla f_j(x_{t,k}^j; \xi_{t,k,b}^j)$

Tangent mean:
$$\tilde{x}_{t+1} = \operatorname{Exp}_{\tilde{x}_t} \left(-\sum_{j \in S_t} \frac{p_j}{\sum_{j \in S_t} p_j} \operatorname{Exp}_{\tilde{x}_t}^{-1}(x_{t,K}^j) \right)$$

$$\mathbf{x}_{t,K}^{j} = \operatorname{Exp}_{\mathbf{x}_{t,K-1}^{j}} \left(-\frac{\alpha_{t,K-1}}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^{j}} \operatorname{grad} f_{j}(\mathbf{x}_{t,K-1}^{j}; \xi_{t,K-1,b}^{j}) \right)$$

$$\boldsymbol{x}_{t,1}^{j} = \operatorname{Exp}_{\tilde{\boldsymbol{x}}_{t}} \left(-\frac{\alpha_{t,0}}{\boldsymbol{B}_{t,0}} \sum_{\boldsymbol{b} \in \mathcal{B}_{t,0}^{j}} \operatorname{grad} f_{j}(\boldsymbol{x}_{t,0}^{j}; \boldsymbol{\xi}_{t,0,b}^{j}) \right)$$

Exp and Exp^{-1} are short of linearity!

. . .

Back to the Euclidean aggregation, note that

$$\Delta_{t,\kappa}^{j} := \tilde{x}_{t} - x_{t,\kappa}^{j} = \sum_{k=0}^{\kappa-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^{j}} \nabla f_{j}(x_{t,k}^{j};\xi_{t,k,b}^{j}).$$

Back to the Euclidean aggregation, note that

$$\Delta_{t,\kappa}^{j} := \tilde{x}_{t} - x_{t,\kappa}^{j} = \sum_{k=0}^{\kappa-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^{j}} \nabla f_{j}(x_{t,k}^{j};\xi_{t,k,b}^{j}).$$

Then one has

$$ilde{x}_{t+1} = ilde{x}_t - d_t, ext{ with } d_t = \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \Delta^j_{t,\kappa}.$$

Back to the Euclidean aggregation, note that

$$\Delta_{t,\kappa}^j := \tilde{x}_t - x_{t,\kappa}^j = \sum_{k=0}^{\kappa-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^j} \nabla f_j(x_{t,k}^j; \xi_{t,k,b}^j).$$

Then one has

$$ilde{x}_{t+1} = ilde{x}_t - d_t, ext{ with } d_t = \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \Delta^j_{t,\kappa}.$$

In the Euclidean setting:

- agent *j* sends $\Delta_{t,k}^{j}$ to the server
- the server averages these $\Delta_{t,K}^{j}$
- the server generates a new global parameter *x*_{t+1}

Back to the Euclidean aggregation, note that

$$\Delta_{t,\kappa}^{j} := \tilde{x}_{t} - x_{t,\kappa}^{j} = \sum_{k=0}^{\kappa-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^{j}} \nabla f_{j}(x_{t,k}^{j};\xi_{t,k,b}^{j}).$$

Then one has

$$ilde{x}_{t+1} = ilde{x}_t - d_t, ext{ with } d_t = \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \Delta^j_{t,\kappa}.$$

In the Euclidean setting:

- agent *j* sends $\Delta_{t,k}^{j}$ to the server
- the server averages these $\Delta_{t,K}^{j}$
- the server generates a new global parameter *x*_{t+1}

In existing works [Kar+20; Red+21], sending $\Delta_{t,K}^{j}$ is to use acceleration technique in the server aggregation.

Back to the Euclidean aggregation, note that

$$\Delta_{t,\kappa}^{j} := \tilde{x}_{t} - x_{t,\kappa}^{j} = \sum_{k=0}^{\kappa-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^{j}} \nabla f_{j}(x_{t,k}^{j};\xi_{t,k,b}^{j}).$$

Then one has

$$ilde{x}_{t+1} = ilde{x}_t - d_t, ext{ with } d_t = \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \Delta^j_{t,\kappa}.$$

In the Euclidean setting:

- agent *j* sends $\Delta_{t,k}^{j}$ to the server
- the server averages these $\Delta_{t,K}^{j}$
- the server generates a new global parameter *x*_{t+1}

In the Riemannian setting, we proposed a similar aggregation

- agent *j* sends the "Δ^j_{t,K}" to the server;
- the server averages these "Δ^j_{t,K}";
- the server retracts the average into the manifold;

Back to the Euclidean aggregation, note that

$$\Delta_{t,\kappa}^j := \tilde{x}_t - x_{t,\kappa}^j = \sum_{k=0}^{\kappa-1} \frac{\alpha_{t,k}}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^j} \nabla f_j(x_{t,k}^j; \xi_{t,k,b}^j).$$

Then one has

$$ilde{x}_{t+1} = ilde{x}_t - d_t, ext{ with } d_t = \sum_{j \in \mathcal{S}_t} rac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \Delta^j_{t,\kappa}.$$

In the Euclidean setting:

- agent *j* sends $\Delta_{t,k}^{j}$ to the server
- the server averages these $\Delta_{t,K}^{j}$
- the server generates a new global parameter *x*_{t+1}

In the Riemannian setting, we proposed a similar aggregation

- agent *j* sends the "Δ^j_{t,K}" to the server;
- the server averages these "Δ^j_{t,K}";
- the server retracts the average into the manifold;

What is " $\Delta_{t,K}^{j}$ " in the Riemannian manifold?

Construct the " $\Delta_{t,K}^{j}$ ", which is dented by $\zeta_{t,K}^{j}$ in the Riemannian setting:

• The local mini-batch gradients, $\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^{j}} \operatorname{grad} f_{j}(x_{t,0}^{j}; \xi_{t,0,b}^{j}), \dots, \frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^{j}} \operatorname{grad} f_{j}(x_{t,K-1}^{j}; \xi_{t,K-1,b}^{j})$ are inside different tangent spaces.

Construct the " $\Delta_{t,K}^{j}$ ", which is dented by $\zeta_{t,K}^{j}$ in the Riemannian setting:

- The local mini-batch gradients, $\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^{j}} \operatorname{grad} f_{j}(x_{t,0}^{j}; \xi_{t,0,b}^{j}), \dots, \frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^{j}} \operatorname{grad} f_{j}(x_{t,K-1}^{j}; \xi_{t,K-1,b}^{j})$ are inside different tangent spaces.
- Transport the local mini-batch gradients to the tangent space $T_{\tilde{X}_t}\mathcal{M}$, i.e., $\Gamma_{x_{t,0}^j}^{\tilde{X}_t}\left(\frac{1}{B_{t,0}}\sum_{b\in\mathcal{B}_{t,0}^j} {}^{\operatorname{grad}f_j(x_{t,0}^j;\,\xi_{t,0,b}^j)}\right), \dots, \Gamma_{x_{t,1}^j}^{\tilde{X}_t}\left(\frac{1}{B_{t,K-1}}\sum_{b\in\mathcal{B}_{t,K-1}^j} {}^{\operatorname{grad}f_j(x_{t,K-1}^j;\,\xi_{t,K-1,b}^j)}\right),$

Construct the " $\Delta_{t,K}^{j}$ ", which is dented by $\zeta_{t,K}^{j}$ in the Riemannian setting:

- The local mini-batch gradients, $\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^{j}} \operatorname{grad} f_{j}(x_{t,0}^{j}; \xi_{t,0,b}^{j}), \dots, \frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^{j}} \operatorname{grad} f_{j}(x_{t,K-1}^{j}; \xi_{t,K-1,b}^{j})$ are inside different tangent spaces.
- Transport the local mini-batch gradients to the tangent space $T_{\tilde{x}_t}\mathcal{M}$, i.e., $\Gamma_{x_{t,0}^{\tilde{x}_t}}^{\tilde{x}_t} \left(\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^j} \operatorname{grad}_j(x_{t,0}^j; \xi_{t,0,b}^j) \right), \dots, \Gamma_{x_{t,1}^j}^{\tilde{x}_t} \left(\frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^j} \operatorname{grad}_j(x_{t,K-1}^j; \xi_{t,K-1,b}^j) \right),$
- Add these transported together to get to $\zeta_{t,K}^{j}$:

$$\zeta_{t,K}^{j} = \sum_{k=0}^{K-1} \alpha_{t,k} \Gamma_{x_{t,k}^{j}}^{\tilde{x}_{t}} \left(\frac{1}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^{j}} \operatorname{grad} f_{j}(x_{t,k}^{j}; \xi_{t,k,b}^{j}) \right);$$

Construct the " $\Delta_{t,K}^{j}$ ", which is dented by $\zeta_{t,K}^{j}$ in the Riemannian setting:

- The local mini-batch gradients, $\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^{j}} \operatorname{grad} f_{j}(x_{t,0}^{j}; \xi_{t,0,b}^{j}), \dots, \frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^{j}} \operatorname{grad} f_{j}(x_{t,K-1}^{j}; \xi_{t,K-1,b}^{j})$ are inside different tangent spaces.
- Transport the local mini-batch gradients to the tangent space $T_{\tilde{x}_t}\mathcal{M}$, i.e., $\Gamma_{x_{t,0}^j}^{\tilde{x}_t} \left(\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^j} {}^{\operatorname{grad}f_j(x_{t,0}^j; \xi_{t,0,b}^j)} \right), \dots, \Gamma_{x_{t,1}^j}^{\tilde{x}_t} \left(\frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^j} {}^{\operatorname{grad}f_j(x_{t,K-1}^j; \xi_{t,K-1,b}^j)} \right),$
- Add these transported together to get to $\zeta_{t \kappa}^{j}$:

$$\zeta_{t,K}^{j} = \sum_{k=0}^{K-1} \alpha_{t,k} \Gamma_{x_{t,k}^{j}}^{\tilde{x}_{t}} \left(\frac{1}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^{j}} \operatorname{grad} f_{j}(x_{t,k}^{j}; \xi_{t,k,b}^{j}) \right);$$

The proposed server aggregation is given by

$$\begin{aligned} \widetilde{\mathbf{x}}_{t+1} &= \mathbf{R}_{\widetilde{\mathbf{x}}_t} \left(-\sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \zeta_{t,K}^j \right) \\ &= \mathbf{R}_{\widetilde{\mathbf{x}}_t} \left(-\sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \sum_{k=0}^{K-1} \alpha_{t,k} \Gamma_{\mathbf{x}_{t,k}^j}^{\widetilde{\mathbf{x}}_t} \left(\frac{1}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^j} \operatorname{grad} f_j(\mathbf{x}_{t,k}^j; \boldsymbol{\xi}_{t,k,b}^j) \right) \right). \end{aligned}$$
A new server aggregation: average of gradient stream

Construct the " $\Delta_{t,K}^{j}$ ", which is dented by $\zeta_{t,K}^{j}$ in the Riemannian setting:

- The local mini-batch gradients, $\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^{j}} \operatorname{grad} f_{j}(x_{t,0}^{j}; \xi_{t,0,b}^{j}), \dots, \frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^{j}} \operatorname{grad} f_{j}(x_{t,K-1}^{j}; \xi_{t,K-1,b}^{j})$ are inside different tangent spaces.
- Transport the local mini-batch gradients to the tangent space $T_{\tilde{x}_t}\mathcal{M}$, i.e., $\Gamma_{x_{t,0}^{\tilde{x}_t}}^{\tilde{x}_t} \left(\frac{1}{B_{t,0}} \sum_{b \in \mathcal{B}_{t,0}^j} \operatorname{grad}_j(x_{t,0}^j; \xi_{t,0,b}^j) \right), \dots, \Gamma_{x_{t,1}^j}^{\tilde{x}_t} \left(\frac{1}{B_{t,K-1}} \sum_{b \in \mathcal{B}_{t,K-1}^j} \operatorname{grad}_j(x_{t,K-1}^j; \xi_{t,K-1,b}^j) \right),$
- Add these transported together to get to $\zeta_{t,K}^{j}$:

$$\zeta_{t,K}^{j} = \sum_{k=0}^{K-1} \alpha_{t,k} \Gamma_{x_{t,k}^{j}}^{\tilde{x}_{t}} \left(\frac{1}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^{j}} \operatorname{grad} f_{j}(x_{t,k}^{j}; \xi_{t,k,b}^{j}) \right);$$

The proposed server aggregation is given by

$$\begin{split} \tilde{\mathbf{x}}_{t+1} &= \mathbf{R}_{\tilde{\mathbf{x}}_t} \left(-\sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \zeta_{t,K}^j \right) \\ &= \mathbf{R}_{\tilde{\mathbf{x}}_t} \left(-\sum_{j \in \mathcal{S}_t} \frac{p_j}{\sum_{j \in \mathcal{S}_t} p_j} \sum_{k=0}^{K-1} \alpha_{t,k} \Gamma_{\mathbf{x}_{t,k}^j}^{\tilde{\mathbf{x}}_t} \left(\frac{1}{B_{t,k}} \sum_{b \in \mathcal{B}_{t,k}^j} \operatorname{grad} f_j(\mathbf{x}_{t,k}^j; \xi_{t,k,b}^j) \right) \right). \end{split}$$

The proposed aggregation is another generalization of the Euclidean aggregation.

Riemannian Federated Learning via Averaging Gradient Stream

Algorithm: Riemannian Federated Learning Averaging Gradient Stream

1. for t = 0, 1, ..., T - 1 do

- 2. The server uniformly selects a subset S_t of s agents at random;
- 3. The server upload global parameter \tilde{x}_t to all agents in S_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$;
- 4. for $j \in S_t$ in parallel do
- 5. Set $\zeta_{t,0}^j \leftarrow \mathbf{0}_{\tilde{x}_t}$;
- 6. for k = 1, 2, ..., K do
- 7. Agent *j* randomly samples an i.i.d. mini-batch $B_{t,k-1}^{j}$ of size $B_{t,k-1}$;
- 8. Set $\eta_{t,k-1}^j \leftarrow -\frac{\alpha_{t,k-1}}{B_{t,k-1}} \sum_{b \in \mathcal{B}_{t,k-1}^j} \operatorname{grad} f_j(x_{t,k-1}^j; \xi_{t,k-1,b}^j);$

9. Set
$$x_{t,k}^{j} \leftarrow \mathsf{R}_{x_{t,k-1}^{j}}(\eta_{k-1}^{j})$$
, and $\zeta_{t,k}^{j} \leftarrow \zeta_{t,k-1}^{j} + \Gamma_{x_{t,k-1}^{\tilde{k}_{t}}}^{\tilde{k}_{t}}(\eta_{t,k-1}^{j})$

- 10. end for
- 11. Sent $\zeta_{t \kappa}^{j}$ to the server;
- 12. end for
- 13. Server aggregates the received local parameter difference $\{\zeta_{t,K}^j\}_{j\in\mathcal{S}_t}$ by averaging

$$\tilde{x}_{t+1} \leftarrow \mathtt{R}_{\tilde{x}_{t}} \left(-\sum_{j \in \mathcal{S}_{t}} \frac{p_{j}}{\sum_{j \in \mathcal{S}_{t}} p_{j}} \zeta_{t,K}^{j} \right);$$

14.end for

Riemannian Federated Learning via Averaging Gradient Stream

Algorithm: Riemannian Federated Learning Averaging Gradient Stream

1. for $t = 0, 1, \ldots, T - 1$ do

- 2. The server uniformly selects a subset S_t of s agents at random;
- 3. The server upload global parameter \tilde{x}_t to all agents in S_t , i.e., $x_{t,0}^j \leftarrow \tilde{x}_t$;
- 4. for $j \in S_t$ in parallel do
- 5. Set $\zeta_{t,0}^j \leftarrow \mathbf{0}_{\tilde{x}_t};$
- 6. for k = 1, 2, ..., K do
- 7. Agent *j* randomly samples an i.i.d. mini-batch $B_{t,k-1}^{j}$ of size $B_{t,k-1}$;

8. Set
$$\eta_{t,k-1}^j \leftarrow -\frac{\alpha_{t,k-1}}{B_{t,k-1}} \sum_{b \in \mathcal{B}_{t,k-1}^j} \operatorname{grad}_f(x_{t,k-1}^j; \xi_{t,k-1,b}^j);$$

9. Set
$$x_{t,k}^j \leftarrow \mathsf{R}_{x_{t,k-1}^j}^j(\eta_{k-1}^j)$$
, and $\zeta_{t,k}^j \leftarrow \zeta_{t,k-1}^j + \Gamma_{x_{t,k-1}^j}^{\tilde{x}_t}(\eta_{t,k-1}^j)$

- 10. end for
- 11. Sent $\zeta_{t,K}^{J}$ to the server;
- 12. end for
- 13. Server aggregates the received local parameter difference $\{\zeta_{t,K}^{j}\}_{j \in S_{t}}$ by averaging

$$\tilde{x}_{t+1} \leftarrow \mathtt{R}_{\tilde{x}_{t}} \left(-\sum_{j \in \mathcal{S}_{t}} \frac{p_{j}}{\sum_{j \in \mathcal{S}_{t}} p_{j}} \zeta_{t,K}^{j} \right);$$

14.end for

- The communication cost remains unchanged.
- The computational cost of the server remains unchanged.
- K 1 times more transport calculations on the agent.
- The algorithm works for general manifolds.

- 1. Federated Learning Review
- 2. Riemannian Federated Learning Averaging Gradient Stream
- 3. Convergence Analysis
- 4. Numerical Experiments
- 5. Summary

Assumptions:

- (Full Participation) Full agents participate in local updates at each communication round.
- (I.I.D. Data) Agent's data are subjected to an independently identical distribution.

Assumptions:

- (Full Participation) Full agents participate in local updates at each communication round.
- (I.I.D. Data) Agent's data are subjected to an independently identical distribution.

We focus on expected risk minimization.

Assumptions:

- (Full Participation) Full agents participate in local updates at each communication round.
- (I.I.D. Data) Agent's data are subjected to an independently identical distribution.

We focus on expected risk minimization.

$$\min_{x \in \mathcal{M}} F(x) := \frac{1}{S} \sum_{i=1}^{S} \mathbb{E}_{\xi \sim \mathcal{D}_i}[f_i(x;\xi)]$$
$$= \mathbb{E}[f(x;\xi)]$$

Each agent only has access to f(x; ξ) and gradf(x; ξ).

Convergence Analysis

Assumption 3.1

We assume that:

- x* = arg min_{x∈M} F(x), the outer iterates {x_t}_{t≥1} and the inner iterates {{x_t}_{t≥1} s = 1} generated by FedAGS remain in a compact and connected subset W ⊆ M;
- (2) the compact and connected subset W is totally retractive with respect to the retraction R;
- (3) for each realization of ξ, the component f(·; ξ) are continuously differentiable;
- (4) the vector transport Γ is isometric;
- (5) the cost function F is L-retraction smooth and L-Lipchitz continuous differentiable with respect to Γ on W; and
- (6) the step sizes $\alpha_{t,k}$ are upper bounded, i.e., there exists A > 0 such that $\alpha_{t,k} \leq A$ for all t and k.

Assumption 3.2

For any fixed parameter $x \in M$, the Riemannian stochastic gradient gradf($x; \xi$) is an unbiased estimator of the true gradient corresponding to the parameter x, *i.e.*,

 $\mathbb{E}_{\xi}[\operatorname{grad} f(x;\xi)] = \operatorname{grad} F(x)$

Assumption 3.3

For any fixed parameter $x \in M$, there exists a scalar $\sigma > 0$ such that for any mini-batch indices set \mathcal{B} of the realizations of random variable ξ , the following holds

$$\mathbb{E}_{\mathcal{B}}\left[\left\|\frac{1}{B}\sum_{b\in\mathcal{B}}\operatorname{grad} f(x;\xi_b)-\operatorname{grad} F(x)\right\|^2\right]\leq \frac{\sigma^2}{B}$$

where B is the size of \mathcal{B} .

By the *L*-retraction smoothness of *F*, we have

$$F(\tilde{x}_{t+1}) - F(\tilde{x}_t) \leq \left\langle \operatorname{grad} F(\tilde{x}_t), \operatorname{R}_{\tilde{x}_t}^{-1}(\tilde{x}_{t+1}) \right\rangle + \frac{L}{2} \|\operatorname{R}_{\tilde{x}_t}^{-1}(\tilde{x}_{t+1})\|^2.$$

Taking expectation over the randomness at the *t*-th outer iteration conditioned on \tilde{x}_t yields

$$\mathbb{E}_{t}[F(\tilde{x}_{t+1})] - F(\tilde{x}_{t}) \leq \mathbb{E}_{t}[\left\langle \operatorname{grad} F(\tilde{x}_{t}), \operatorname{R}_{\tilde{x}_{t}}^{-1}(\tilde{x}_{t+1}) \right\rangle] + \frac{L}{2} \mathbb{E}_{t}[\left\|\operatorname{R}_{\tilde{x}_{t}}^{-1}(\tilde{x}_{t+1})\right\|^{2}].$$
(3.1)

where $\mathbb{E}_t[\cdot]$ means the expectation over the randomness of the *t*-outer iteration.

By the *L*-retraction smoothness of *F*, we have

$$F(\tilde{x}_{t+1}) - F(\tilde{x}_t) \leq \left\langle \operatorname{grad} F(\tilde{x}_t), \operatorname{R}_{\tilde{x}_t}^{-1}(\tilde{x}_{t+1}) \right\rangle + \frac{L}{2} \|\operatorname{R}_{\tilde{x}_t}^{-1}(\tilde{x}_{t+1})\|^2.$$

Taking expectation over the randomness at the *t*-th outer iteration conditioned on \tilde{x}_t yields

$$\mathbb{E}_{t}[F(\tilde{x}_{t+1})] - F(\tilde{x}_{t}) \leq \mathbb{E}_{t}[\left\langle \operatorname{grad} F(\tilde{x}_{t}), \operatorname{R}_{\tilde{x}_{t}}^{-1}(\tilde{x}_{t+1}) \right\rangle] + \frac{L}{2} \mathbb{E}_{t}[\left\|\operatorname{R}_{\tilde{x}_{t}}^{-1}(\tilde{x}_{t+1})\right\|^{2}].$$
(3.1)

where $\mathbb{E}_t[\cdot]$ means the expectation over the randomness of the *t*-outer iteration.

We focus on bounding the terms on the right-hand side of (3.1)

The second term of the right-hand side

Lemma 1

The iterates $\{\tilde{x}_t\}$ generated by RFedAGS satisfy that

$$\mathbb{E}_{t}[\|\mathbf{R}_{\tilde{x}_{t}}^{-1}(\tilde{x}_{t+1})\|^{2}] \leq \sum_{k=0}^{K-1} K \alpha_{t,k}^{2} \mathbb{E}_{t}[\|\operatorname{grad} F(x_{t,k}^{j})\|^{2}] + \sum_{k=0}^{K-1} \frac{K \alpha_{t,k}^{2} \sigma^{2}}{SB_{t,k}},$$

where the expectation is taken over the randomness at the t-th outer iteration conditioned on \tilde{x}_t .

Convergence Analysis

The first term of the right-hand side

Lemma 2

At the t-th outer iteration of RFedAGS with a fixed step size $\bar{\alpha}_{t,k} = \bar{\alpha}_t$ within the inner iteration of each agent, we have that

$$egin{aligned} &\mathbb{E}_t[\left\langle ext{grad} F(ilde{x}_t), \mathbb{R}_{ ilde{x}_t}^{-1}(ilde{x}_{t+1})
ight
angle] \ &\leq -rac{(K+1)ar{lpha}_t}{2} \| ext{grad} F(ilde{x}_t)\|^2 - rac{ar{lpha}_t}{2} \sum_{k=1}^{K-1} \mathbb{E}_t[\| ext{grad} F(ilde{x}_{t,k}^j)\|^2 \ &+ rac{ar{lpha}_t \mathcal{L}^2}{2} \sum_{k=1}^{K-1} \mathbb{E}_t[\| ext{R}_{ ilde{x}_t}^{-1}(ilde{x}_{t,k}^j)\|^2], \end{aligned}$$

and, in particular, for K = 1,

$$\mathbb{E}_t[\left\langle \operatorname{grad} \mathcal{F}(\tilde{x}_t), \mathbb{R}_{\tilde{x}_t}^{-1}(\tilde{x}_{t+1}) \right\rangle] = -\bar{\alpha}_t \|\operatorname{grad} \mathcal{F}(\tilde{x}_t)\|^2,$$

where the expectation is taken over the randomness at the t-th outer iteration conditioned on \tilde{x}_t .

The first term of the right-hand side

Lemma 3

At the k-th inner iteration of the t-th outer iteration of RFedAGS, for each agent j = 1, 2, ..., S and k = 1, 2, ..., K - 1, we have

$$\mathbb{E}_{t}[\|\mathbf{R}_{\tilde{x}_{t}}^{-1}(x_{t,k}^{j})\|^{2}] \leq 2kM \sum_{\tau=0}^{k-1} \alpha_{t,\tau}^{2} \mathbb{E}_{t}[\|\mathrm{grad}F(x_{t,\tau}^{j})\|^{2}] + 2kM\sigma^{2} \sum_{\tau=0}^{k-1} \frac{\alpha_{t,\tau}^{2}}{B_{t,\tau}}$$

where the expectation is taken over the randomness at the t-th outer iteration conditioned on \tilde{x}_t , $M = (C_2^2 + A^2 C_1^2 C_3^2)$ is a positive constant, A is stated in Assumption 3.1(6), C_1 is a constant such that $\|\text{grad}F(x)\| \leq C_1$ for all $x \in W$ (as Assumption 3.1(1)), and C_2 and C_3 are two constants related with the manifold and retraction.

Based on Lemmas 2 and 3, if one uses $\alpha_{t,k} = \bar{\alpha}_t$ and $B_{t,k} = \bar{B}_t$, then the first term of the right-hand side of (3.1) is bounded as

$$\begin{split} \mathbb{E}_t \left\langle \operatorname{grad} F(\tilde{x}_t), \operatorname{R}_{\tilde{x}_t}^{-1}(\tilde{x}_{t+1}) \right\rangle &\leq -\frac{\bar{\alpha}_t}{2} (K+1 - ML^2 \bar{\alpha}_t^2 K(K-1)) \|\operatorname{grad} F(\tilde{x}_t)\|^2 \\ &- \frac{\bar{\alpha}_t}{2} (1 - ML^2 \bar{\alpha}_t^2 (K+1)(K-2)) \sum_{k=1}^{K-1} \mathbb{E}_t [\|\operatorname{grad} F(x_{t,k}^j)\|^2] \\ &+ \frac{(2K-1)K(K-1)M\sigma^2 L^2 \bar{\alpha}_t^3}{6\bar{B}_t}. \end{split}$$

Convergence Analysis

Theorem 4

wi th

We run RFedAGS with a fixed step size $\alpha_{t,k} = \bar{\alpha}_t$ and a fixed batch size $B_{t,k} = \bar{B}_t$ within parallel steps.

• If K = 1 with step sizes $\bar{\alpha}_t$ satisfying

$$2 - \delta \ge L\bar{\alpha}_t; \tag{3.2}$$

• or K > 1 with step sizes $\bar{\alpha}_t$ satisfying

$$\begin{cases} 1 \ge L^2 \bar{\alpha}_t^2 \mathcal{M}(\mathcal{K}+1)(\mathcal{K}-2) + \bar{\alpha}_t \mathcal{L}\mathcal{K}, \\ 1 - \delta \ge 2L^2 \bar{\alpha}_t^2 \mathcal{M}, \end{cases}$$
(3.3)

where $\delta \in (0, 1)$ is some constant, then it holds that

$$\mathbb{E}_{t}[F(\tilde{x}_{t+1})] - F(\tilde{x}_{t}) \leq -\frac{\bar{\alpha}_{t}(K-1+\delta)}{2} \| \operatorname{grad} F(\tilde{x}_{t}) \|^{2} + \frac{K \bar{\alpha}_{t}^{2} \sigma^{2} L}{2 \bar{B}_{t}} H(\bar{\alpha}_{t}, K, S),$$

here $H(\bar{\alpha}_{t}, K, S) = \frac{\bar{\alpha}_{t}(2K-1)(K-1)ML}{3} + \frac{K}{S}$, and the expectations above are taken over e randomness at the t-th outer iteration conditioned on \tilde{x}_{t} .

Crucial result for the convergence analysis

- Inspired from Euclidean results in [ZC18];
- Difficulty: $R_x^{-1}(y)$ versus y x;
- Technique: Taylor expansion of R^{-1} and further control the higher order term;

[[]ZC18] F. Zhou and G. Cong. On the convergence properties of a K-step averaging stochastic gradient descent algorithm for nonconvex optimization. International Joint Conference on Artificial Intelligence, 2018.

Theorem 5 (Nonconvex)

If we run RFedAGS with a fixed step size $\alpha_{t,k} = \bar{\alpha}$, a fixed batch size $B_{t,k} = \bar{B}$ satisfying (3.2) and (3.3). Then the resulting sequence of iterates $\{\tilde{x}_t\}_{t=1}^T$ satisfies

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \|\operatorname{grad} F(\tilde{x}_{t})\|^{2}\right] \leq \frac{2(F(\tilde{x}_{1}) - F(x^{*}))}{T(K - 1 + \delta)\bar{\alpha}} + \frac{\bar{\alpha}K\sigma^{2}L}{(K - 1 + \delta)}H(\bar{\alpha}, K, S),$$

here $x^{*} \in \operatorname{arg\,min}_{x \in \mathcal{M}}F(x).$

wł

A consequence of Theorem 5:

Corollary 6 (Nonconvex)

Under the conditions of Theorem 5, if the step size $\bar{\alpha}$ and T are given by

$$\begin{split} \bar{\alpha} &= \sqrt{\frac{(F(\tilde{x}_t) - F(x^*))S\bar{B}}{TK^2\sigma^2 L}}, \text{ and } T \geq \frac{(F(\tilde{x}_1) - F(x^*))\bar{B}LM^2S^3(2K-1)^2(K-1)^2}{9\sigma^2K^4}, \\ \text{such that (3.3) holds, the following holds that} \\ &\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \|\text{grad}F(\tilde{x}_t)\|^2\right] \leq \left(\frac{4K}{K-1+\delta}\right)\sqrt{\frac{(F(\tilde{x}_1) - F(x^*))\sigma^2L}{S\bar{B}}}\frac{1}{\sqrt{T}}. \end{split}$$

A consequence of Theorem 5:

Corollary 6 (Nonconvex)

Under the conditions of Theorem 5, if the step size $\bar{\alpha}$ and T are given by

$$\bar{\alpha} = \sqrt{\frac{(F(\tilde{x}_t) - F(x^*))S\bar{B}}{TK^2\sigma^2L}}, \text{ and } T \ge \frac{(F(\tilde{x}_1) - F(x^*))\bar{B}LM^2S^3(2K-1)^2(K-1)^2}{9\sigma^2K^4},$$
such that (3.3) holds, the following holds that
$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \|\text{grad}F(\tilde{x}_t)\|^2\right] \le \left(\frac{4K}{K-1+\delta}\right)\sqrt{\frac{(F(\tilde{x}_1) - F(x^*))\sigma^2L}{S\bar{B}}}\frac{1}{\sqrt{T}}.$$

Given small $\epsilon > 0$, ensuring $\frac{1}{T}\mathbb{E}[\sum_{t=1}^{T} \|\operatorname{grad} F(\tilde{x}_t)\|^2] \le \epsilon$ resuires $T \ge \mathcal{O}(\frac{1}{\epsilon^2})$

Theorem 7 (Riemannian Polyak-Łojasiewicz)

Under the same conditions as Theorem 5 together with assuming that the function F satisfies the Riemannian Polyak-Łojasiewicz (RPL) condition

$$F(x) - F(x^*) \leq \frac{1}{2\mu} \|\operatorname{grad} F(x)\|^2, \ \forall x \in \mathcal{W},$$

where $x^* = \arg \min_{x \in \mathcal{M}} F(x)$ and μ is a positive constant. Under Conditions (3.2) and (3.3) and $\bar{\alpha} \leq \frac{1}{\mu(K-1+\delta)}$, we have

$$\begin{split} \mathbb{E}[F(\tilde{x}_{T}) - F(x^{*})] &\leq (1 - \mu \bar{\alpha} (K - 1 + \delta))^{T-1} \mathbb{E}[F(\tilde{x}_{1}) - F(x^{*})] \\ &+ \frac{K \bar{\alpha} \sigma^{2} L}{2 \mu \bar{B} (K - 1 + \delta)} H(\bar{\alpha}, K, S). \end{split}$$

Theorem 7 (Riemannian Polyak-Łojasiewicz)

Under the same conditions as Theorem 5 together with assuming that the function F satisfies the Riemannian Polyak-Łojasiewicz (RPL) condition

$$F(x) - F(x^*) \leq \frac{1}{2\mu} \|\operatorname{grad} F(x)\|^2, \ \forall x \in \mathcal{W},$$

where $x^* = \arg \min_{x \in \mathcal{M}} F(x)$ and μ is a positive constant. Under Conditions (3.2) and (3.3) and $\bar{\alpha} \leq \frac{1}{\mu(K-1+\delta)}$, we have

$$\begin{split} \mathbb{E}[F(\tilde{x}_{T}) - F(x^{*})] &\leq (1 - \mu \bar{\alpha} (K - 1 + \delta))^{T - 1} \mathbb{E}[F(\tilde{x}_{1}) - F(x^{*})] \\ &+ \frac{K \bar{\alpha} \sigma^{2} L}{2 \mu \bar{B} (K - 1 + \delta)} H(\bar{\alpha}, K, S). \end{split}$$

Linearly converge to a small neighborhood of the solution.

Theorem 8 (Nonconvex)

If we run RFedAGS with decaying step sizes $\alpha_{t,k} = \bar{\alpha}_t$, and not fixed but bounded batch sizes $B_{t,k} = \bar{B}_t$ for outer iterations satisfying (3.2) and (3.3) and $B_{\text{low}} \leq \bar{B}_t \leq B_{\text{up}}$ with B_{low} and B_{up} being positive integers, then the resulting sequence of iterates $\{\tilde{x}_t\}_{t=1}^T$ satisfies

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\bar{\alpha}_t}{\sum_{t=1}^{T}\bar{\alpha}_t}\|\operatorname{grad} F(\tilde{x}_t)\|^2\right] \leq \frac{2(F(\tilde{x}_1)-F(x^*))}{(K-1+\delta)\sum_{t=1}^{T}\bar{\alpha}_t} + \sum_{t=1}^{T}\frac{\bar{\alpha}_t^2K\sigma^2L}{(K-1+\delta)\bar{B}_t\sum_{t=1}^{T}\bar{\alpha}_t}H(\bar{\alpha}_t,K,S).$$

Further, if the step size $\bar{\alpha}_t$'s satisfy

$$\sum_{t=1}^{\infty} \bar{lpha}_t = \infty, \text{ and } \sum_{t=1}^{\infty} \bar{lpha}_t^2 < \infty,$$

then the following holds

$$\liminf_{t\to\infty} \mathbb{E}[\|\operatorname{grad} F(\tilde{x}_t)\|^2] = 0.$$

Theorem 8 (Nonconvex)

If we run RFedAGS with decaying step sizes $\alpha_{t,k} = \bar{\alpha}_t$, and not fixed but bounded batch sizes $B_{t,k} = \bar{B}_t$ for outer iterations satisfying (3.2) and (3.3) and $B_{\text{low}} \leq \bar{B}_t \leq B_{\text{up}}$ with B_{low} and B_{up} being positive integers, then the resulting sequence of iterates $\{\tilde{x}_t\}_{t=1}^T$ satisfies

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\bar{\alpha}_t}{\sum_{t=1}^{T}\bar{\alpha}_t}\|\operatorname{grad} F(\tilde{x}_t)\|^2\right] \leq \frac{2(F(\tilde{x}_1)-F(x^*))}{(K-1+\delta)\sum_{t=1}^{T}\bar{\alpha}_t} + \sum_{t=1}^{T}\frac{\bar{\alpha}_t^2K\sigma^2L}{(K-1+\delta)\bar{B}_t\sum_{t=1}^{T}\bar{\alpha}_t}H(\bar{\alpha}_t,K,S).$$

Further, if the step size $\bar{\alpha}_t$'s satisfy

$$\sum_{t=1}^{\infty} \bar{lpha}_t = \infty, ext{ and } \sum_{t=1}^{\infty} \bar{lpha}_t^2 < \infty,$$

then the following holds

 $\liminf_{t\to\infty} \mathbb{E}[\|\operatorname{grad} F(\tilde{x}_t)\|^2] = 0.$

The iterates has at least one accumulation point which is a critical point in the sense of expectation

Theorem 9 (RPL)

Under the same conditions as Theorem 7 except for that the step size sequence and the batch size sequence satisfy

$$\begin{aligned} \alpha_{t,k} &= \bar{\alpha}_t = \frac{\kappa}{\gamma + t} \text{ for some } \gamma > 0 \text{ and } \kappa > \frac{1}{\mu(K-1+\delta)} \text{ such that } \bar{\alpha}_1 \\ \text{ satisfies (3.2) and (3.3), and} \\ B_{t,k} &= \bar{B}_t \in [B_{\text{low}}, B_{\text{up}}]. \end{aligned}$$

Then for all $t \in \{1, 2, ..., T - 1\}$, the expected optimality gap is bounded by

$$\mathbb{E}[F(ilde{x}_t) - F(x^*)] \leq rac{
u}{\gamma+t},$$

where

$$\begin{split} \nu &= \max\left\{\frac{\kappa^2 K^2 \sigma^2 L}{SB_{\text{low}}(\kappa \mu (K-1+\delta)-1)}, \frac{\kappa^3 (2K-1)K(K-1)\sigma^2 L^2 M}{3\gamma B_{\text{low}}(\kappa \mu (K-1+\delta)-1)}, \right. \end{split}$$
$$(\gamma+1)(F(\tilde{x}_1)-F(x^*)) \bigg\}. \end{split}$$

Theorem 9 (RPL)

Under the same conditions as Theorem 7 except for that the step size sequence and the batch size sequence satisfy

$$\begin{aligned} \alpha_{t,k} &= \bar{\alpha}_t = \frac{\kappa}{\gamma + t} \text{ for some } \gamma > 0 \text{ and } \kappa > \frac{1}{\mu(K-1+\delta)} \text{ such that } \bar{\alpha}_1 \\ \text{ satisfies (3.2) and (3.3), and} \\ B_{t,k} &= \bar{B}_t \in [B_{\text{low}}, B_{\text{up}}]. \end{aligned}$$

Then for all $t \in \{1, 2, ..., T - 1\}$, the expected optimality gap is bounded by

$$\mathbb{E}[F(ilde{x}_t) - F(x^*)] \leq rac{
u}{\gamma+t},$$

where

$$\nu = \max\left\{\frac{\kappa^2 K^2 \sigma^2 L}{SB_{\text{low}}(\kappa \mu (K-1+\delta)-1)}, \frac{\kappa^3 (2K-1)K(K-1)\sigma^2 L^2 M}{3\gamma B_{\text{low}}(\kappa \mu (K-1+\delta)-1)}, (\gamma+1)(F(\tilde{x}_1)-F(x^*))\right\}.$$

The expected optimal gaps vanish sublinearly

Riemannian Federated Learning via Averaging Gradient Stream

- Consider "nonconvex" and "RPL" scenarios;
- Fixed step size and decaying step size;
- Techniques inspired from [ZC18] (nonconvex) and [BCN18] (RPL);
- Convergence guaranteed for multiple agents S > 1 and multiple local updates K > 1;

[ZC18] F. Zhou and G. Cong. On the convergence properties of a K-step averaging stochastic gradient descent algorithm for nonconvex optimization. International Joint Conference on Artificial Intelligence, 2018. [BCN18] L. Bottou, F. E. Curtis, and J. Nocedal, Optimization Methods for Large-Scale Machine Learning, SIAM Review, 2018. An important question of RFedAGS is whether multiple inner iterations, i.e., K > 1, bring benefits.

In other words, is the optimal choice of K, denoted by K^* , greater than 1?

An important question of RFedAGS is whether multiple inner iterations, i.e., K > 1, bring benefits.

In other words, is the optimal choice of K, denoted by K^* , greater than 1?

Yes!

Theorem 10 (Fixed step size)

We run RFedAGS with a fixed batch size $B_{t,k} = \overline{B}$ and a fixed step size $\alpha_{t,k} = \overline{\alpha}$ satisfying Conditions (3.2) and (3.3). Under the same conditions as Theorem 5, if the number of outer iterations T satisfies

$$(F(\tilde{x}_1) - F(x^*)) > \frac{(3\delta - 1)\bar{\alpha}^2 T L \sigma^2}{2S\bar{B}} + \frac{\delta \bar{\alpha}^3 \sigma^2 L^2 T M}{\bar{B}},$$

then the optimal choice of K, the number of inner iterations, is greater than 1.

Theorem 11 (decaying step size)

We run RFedAGS with batch sizes $B_{t,k} = \overline{B}_t$ and decaying step sizes $\alpha_{t,k} = \overline{\alpha}_t$ such that $\overline{\alpha}_1$ satisfying Conditions (3.2) and (3.3). Under the same conditions as Theorem 8, if the number of outer iterations T satisfies

$$(F(\tilde{x}_1) - F(x^*)) > \delta\sigma^2 L \sum_{t=1}^T \frac{\bar{\alpha}_t^2}{\bar{B}_t} \left(\bar{\alpha}_t M L + \frac{2}{S} \right),$$

then the optimal choice K is greater than 1.

- 1. Federated Learning Review
- 2. Riemannian Federated Learning Averaging Gradient Stream
- 3. Convergence Analysis
- 4. Numerical Experiments
- 5. Summary

Numerical Experiments

- The experiments conducted on the empirical risk minimization
- I.I.D. data and full agent participation

$$\min_{x \in \mathcal{M}} F(x) := \frac{1}{S} \sum_{i=1}^{S} f(x; \mathcal{D}_i) = \frac{1}{S} \sum_{i=1}^{S} \sum_{j=1}^{N} \frac{1}{N} f(x; z_{i,j}),$$

where *S* is the number of agents, $D_i = \{z_{i,1}, \ldots, z_{i,N}\}$ is the local dataset with size of *N* held by agent *i*.

For decaying step sizes cases, the step sizes are computed by the following formulation:

$$\bar{\alpha}_t = \begin{cases} \alpha_0 & \text{if } t = 0\\ \alpha_0/(\beta + c_t) & \text{if } t > 0, \end{cases} \text{ with } c_t = \begin{cases} 0 & \text{if } t = 0, \\ c_{t-1} + 1 & \text{if } \operatorname{mod}(t, \operatorname{dec}) = 0, \\ c_{t-1} & \text{otherwise}, \end{cases}$$

where α_0 is the initial step size, β is the decaying parameter, and dec is the decaying gap.

Three simulation experiments

Computing principal eigenvector over sphere manifolds (CPESph)

•
$$\min_{x \in S^d} F(x) := -\frac{1}{S} \sum_{i=1}^{S} \frac{1}{N} \sum_{j=1}^{N} x^T(z_{i,j} z_{i,j}^T) x$$
 with
 $S^d = \{x \in \mathbb{R}^{d+1} : ||x||_2 = 1\}$

- The objective locally satisfies RPL condition
- Synthetize the samples $\mathcal{D}_i = \{z_{i,1}, \ldots, z_{i,N}\}$ for all $i = 1, \ldots, S$:
 - Diagonal matrix Σ_i = diag{1, 1 − 1.1ν, ..., 1 − 1.4ν, |y₁|/(d+1), |y₂|/(d+1), ...} of size (d + 1) × (d + 1) with ν being the eigengap and y_i ∈ ℝ being sampled from the standard Gaussian distribution
 - Set $Z_i = U_i \Sigma_i V_i$ with $U_i \in \mathbb{R}^{N \times (d+1)}$ and $V \in \mathbb{R}^{(d+1) \times (d+1)}$ being two orthonormal matrix

Three simulation experiments

Computing Fréchet mean over SPD manifolds (CFMSPD)

•
$$\min_{X \in \mathbb{S}_{++}^d} F(X) := \frac{1}{S} \sum_{i=1}^S \frac{1}{N} \sum_{j=1}^N \|\log(X^{-1/2} Z_{i,j} X^{-1/2})\|_F^2$$
 with
 $\mathbb{S}_{++}^d = \{X \in \mathbb{R}^{d \times d} : X \succ 0\}$

- The objective locally satisfies the RPL condition
- Synthetize the samples $\mathcal{D}_i = \{Z_{i,1}, \dots, Z_{i,N}\} \subset \mathbb{S}^d_{++}$:
 - Each data point is sampled from the Wishart distribution $W(I_d/d, d)$ with a diameter D_{W}

Minimization of the Brockett cost function over Stiefel manifolds (MBCFSti)

• $\min_{X \in \operatorname{St}(p,d)} F(X) = \frac{1}{S} \sum_{i=1}^{S} \sum_{j=1}^{N} \operatorname{trace}(X^T A_{i,j} X H)$ with $\operatorname{St}(p,d) = \{X \in \mathbb{R}^{d \times p} : X^T X = I_p\}$

•
$$H = \operatorname{diag}\{p, p-1, \ldots, 1\}$$

- The objective locally satisfies the RPL condition
- Synthetize the samples $\mathcal{D}_i = \{A_{i,1}, \dots, A_{i,N}\} \subset \mathbb{S}_{++}^d$:
 - Set $A_{i,j} = B + B^T$ with B being drawn from the standard normal distribution

Table 1: The parameters of the three problems and RFedAGS. Notation $a.b_k$ denotes a number $a.b \times 10^k$ and the dash "-" means that the parameter does not exist in the problem.

Parameters	Problem-related				Algorithm-related						
Problems	d	р	ν	$D_{\mathcal{W}}$	S	Ν	$\bar{\alpha}$	$lpha_0$	β	dec	Ē
CPESph	2.5 ₁	-	1_3	-	1.0 ₁	8.0 ₁	1	1	1.0_1	5.0 ₁	6.4 ₁
CFMSPD	2	-	-	1	1.0 ₁	6.0 ₁	3.0 ₋₃	8.0_3	1.0_{-1}	2.0 ₁	3.0 ₁
MBCFSti	2.5 ₁	2	-	-	2.0 ₁	5.0 ₁	3.0_{-3}	2.0_{-2}	1.0_{-1}	5.0 ₁	2.5 ₁

Three simulation experiments



Figure 2: The influence of the different number, K, of local updates on synthetic data. Fixed step size cases (first row) and decaying step size cases (second row).
Three simulation experiments



Figure 2: The influence of the different number, K, of local updates on synthetic data. Fixed step size cases (first row) and decaying step size cases (second row).

- Linear convergence if fixed step sizes
- More accurate if decaying step size
- Optimal K greater than 1

Low-dimensional multitask feature learning

- Let $\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_S\}$ denote the *S* group task
- Each group task T_i is consisted of N tasks, i.e., $T_i = \{T_{i,1}, \dots, T_{i,N}\}$
- All tasks share a latent low-dimensional feature representation
- For each task $\mathcal{T}_{i,j}$:
 - (X_{ij}, y_{ij}) with $X_{ij} \in \mathbb{R}^{d_{ij} \times m}$ and $y_{ij} \in \mathbb{R}^{d_{ij}}$ being the training instances and the corresponding labels

The low-dimensional multitask feature learning problem can be formally formulated as follows:

$$\min_{\mathcal{U} \in Gr(r,m)} \frac{1}{S} \sum_{i=1}^{S} \left(\frac{1}{N} \sum_{j=1}^{N} 0.5 \|X_{ij} U w_{ijU} - y_{ij}\|_{F}^{2} \right)$$

- The Grassmann manifold Gr(r, m) is equipped with the quotient manifold structure Gr(r, m) = St(r, m)/Op
- For a given U, w_{ijU} is the least-squares solution to min_{w_{ij}∈ℝ^r} 0.5||X_{ij}Uw_{ij} − y_{ij}||²_F + λ||w_{ij}||²_F, which has a closed form for λ ≥ 0

Synthetic case for each task $T_{i,j}$

- the number of instances *d_{ij}* is randomly chosen between 10 and 50
- the training instances $X_{ij} \in \mathbb{R}^{d_{ij} \times m}$ with m = 100 are given from the standard Gaussian distribution
- the subspace U^* for the problem is a random generated point in St(5, 100) with the dimension r = 5
- the labels y_{ij} for training instances for $T_{i,j}$ are generated by $y_{ij} = X_{ij}U^*(U^*)^T w_{ij}$ with w_{ij} being generated by the standard Gaussian distribution
- then the labels are perturbed by a random Gaussian noise with zero mean and 10⁻⁶ standard deviation

A real-world application

Synthetic case $S = 20, N = 50, \lambda = 0$, the fixed step size $\bar{\alpha} = 0.003$ and the fixed batch size $\bar{B} = 25$



Figure 3: The influence of the different number, K, of local updates on synthetic data. Excess risk (left) and distance to the optimal subspaces, i.e., dist(U^k , U^*), (right).

A real-world application

Synthetic case $S = 20, N = 50, \lambda = 0$, the fixed step size $\bar{\alpha} = 0.003$ and the fixed batch size $\bar{B} = 25$



Figure 3: The influence of the different number, K, of local updates on synthetic data. Excess risk (left) and distance to the optimal subspaces, i.e., dist(U^k , U^*), (right).

- Linear convergence
- Optimal K > 1 exists.

Real-world case

- benchmark dataset [Gol91; Evg+05]: School, which is consisted of 15362 students' information from 139 secondary.
- There are 139 tasks each of which is to predict student performance in each school.
- We randomly sample 80% students from each school to form the training set and set the remainder as the testing set
- In terms of FL setting, we consider S = 6 and N = 23 (implying that one of the schools is ignored)
- measure the performance of those methods using the normalized mean square error (NMSE)

NMSE =
$$\frac{\text{MSE}(\hat{y}, y)}{\text{var}(y)}$$
 with MSE $(\hat{y}, y) = \frac{\sum_{i=1}^{S} \sum_{j=1}^{N} \sum_{k=1}^{d_{ij}} (\hat{y}_{ij}(k) - y_{ij}(k))^2}{\sum_{i=1}^{S} \sum_{j=1}^{N} d_{ij}}$,

A real-world application

Real-world case $S = 6, N = 23 \ \bar{\alpha} = 1.0 \times 10^{-6}, \bar{B} = 18, \lambda = 1.0 \times 10^{-3}$



Figure 4: Costs (first row) and NMSE scores (second row) against iterations for RFedAGS with $K \in \{1, 4, 8, 10\}$, RFedAvg, RSD, RCG and RLBFGS. Here the *x*-axis "iteration" in FL setting means the number of outer iterations

Real-world case $S = 6, N = 23 \ \bar{\alpha} = 1.0 \times 10^{-6}, \bar{B} = 18, \lambda = 1.0 \times 10^{-3}$

Table 2: the best NMSE scores (lower is better) on testing set for different subspace dimension *r*. Here a number $a.b_k$ means $a.b \times 10^k$. The numbers in parentheses represent their corresponding *x*-axis coordinates in Figure 4.

Dimension	RFedAGS				RFedAvg	RSD	RCG	RLBFGS
Dimonolon	<i>K</i> = 1	<i>K</i> = 4	K = 8	<i>K</i> = 10				
<i>r</i> = 3	5.09_1(100)	4.78_1(100)	4.72_1(100)	4.70_1(100)	5.09_1(100)	4.65_1(62)	4.60_1(62)	4.60_1(99)
<i>r</i> = 4	4.38_1(100)	4.37_1(30)	$4.37_{-1}(15)$	$4.37_{-1}(12)$	4.38_1(100)	4.32_1(100)	$4.39_{-1}(5)$	4.29_1(23)
<i>r</i> = 5	4.07_1(100)	$4.05_{-1}(51)$	4.05_1(23)	4.05_1(18)	4.07_1(100)	4.03_1(60)	$3.96_{-1}(44)$	3.98_1(18)

- Introduced the federated learning;
- Proposed a new server aggregation;
- Extensive convergence analysis;
- Numerical experiments;

Thank you for your attention!

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. "Optimization Methods for Large-Scale Machine Learning". In: *SIAM Review* 60.2 (2018), pp. 223–311.

S. Bonnabel. "Stochastic Gradient Descent on Riemannian Manifolds". In: *IEEE Transactions on Automatic Control* 58.9 (2013), pp. 2217–2229. DOI: 10.1109/TAC.2013.2254619.

Theodoros Evgeniou et al. "Learning multiple tasks with kernel methods.". In: *Journal of Machine Learning Research* 6.4 (2005).

Harvey Goldstein. "Multilevel modelling of survey data". In: Journal of the Royal Statistical Society. Series D (The Statistician) 40.2 (1991), pp. 235–244.

Zhenwei Huang et al. Federated Learning on Riemannian Manifolds with Differential Privacy. 2024. arXiv: 2404.10029 [cs.LG].

References II

Sai Praneeth Karimireddy et al. "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR, 2020.

Jiaxiang Li and Shiqian Ma. "Federated Learning on Riemannian Manifolds". In: *Applied Set-Valued Analysis and Optimization* 5.2 (2023).

Brendan McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 20–22 Apr 2017, pp. 1273–1282. URL: https:

//proceedings.mlr.press/v54/mcmahan17a.html.

Sashank J. Reddi et al. "Adaptive Federated Optimization". In: International Conference on Learning Representations. 2021. Fan Zhou and Guojing Cong. "On the convergence properties of a K-step averaging stochastic gradient descent algorithm for nonconvex optimization". In: *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence. 2018.

Jiaojiao Zhang et al. Nonconvex Federated Learning on Compact Smooth Submanifolds With Heterogeneous Data. 2024. arXiv: 2406.08465 [cs.LG]. URL: https://arxiv.org/abs/2406.08465.