Federated Learning on Riemannian Manifolds with Differential Privacy

Speaker: Wen Huang

Xiamen University

Sept. 29. 2024

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

2024年数学优化青年学者研讨会
大连

Outline

Outline

Machine Learning (Empirical risk minimization):

$$\min_{x \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} f_i(x, z_i)$$

- Parameter x;
- Data $D = \{z_1, \ldots, z_n\}$;

Machine Learning (Empirical risk minimization):

$$\min_{x \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} f_i(x, z_i)$$

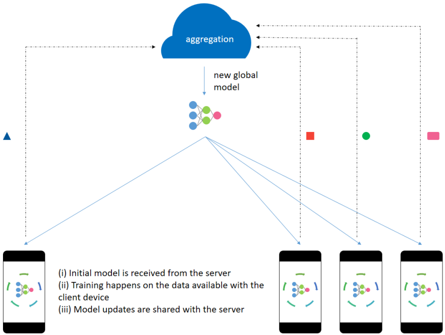
- Parameter x;
- Data $D = \{z_1, \ldots, z_n\}$;

Federated Learning:

- Privacy
- Computation ability

Applications:

- Healthcare
- Smart Home
- Financial Services
- Transportation



Federated Learning Optimization:

$$\min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^{N} p_i f_i(x) = \sum_{i=1}^{N} p_i \left(\frac{1}{N_i} \sum_{j=1}^{N_i} f_i(x, z_{i,j}) \right), \text{ with } p_i = \frac{N_i}{n}, \tag{1.1}$$

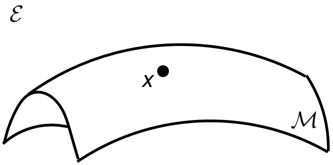
- N is the number of agents;
- $n = \sum_{i=1}^{N} N_i$;
- f_i is the local objective of agent i;
- $D_i = \{z_{i,1}, \ldots, z_{i,N_i}\}$ is the local data held by agent i with $|D_i| = N_i$.

Federated Learning Optimization:

$$\min_{x \in \mathcal{M}} f(x) = \sum_{i=1}^{N} p_i f_i(x) = \sum_{i=1}^{N} p_i \left(\frac{1}{N_i} \sum_{j=1}^{N_i} f_i(x, z_{i,j}) \right), \text{ with } p_i = \frac{N_i}{n}, \qquad (1.1)$$

- N is the number of agents;
- $n = \sum_{i=1}^{N} N_i$;
- f_i is the local objective of agent i;
- $D_i = \{z_{i,1}, \ldots, z_{i,N_i}\}$ is the local data held by agent i with $|D_i| = N_i$.

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



Federated Learning Optimization:

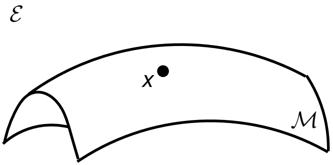
$$\min_{x \in \mathcal{M}} f(x) = \sum_{i=1}^{N} p_i f_i(x) = \sum_{i=1}^{N} p_i \left(\frac{1}{N_i} \sum_{j=1}^{N_i} f_i(x, z_{i,j}) \right), \text{ with } p_i = \frac{N_i}{n}, \qquad (1.1)$$

- N is the number of agents;
- $n = \sum_{i=1}^{N} N_i$;
- f_i is the local objective of agent i;
- $D_i = \{z_{i,1}, \ldots, z_{i,N_i}\}$ is the local data held by agent i with $|D_i| = N_i$.

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

Applications:

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



Euclidean version:

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

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

$$x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)};$$

9. **end for**

$$\min_{x_j \in \mathbb{R}^n} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$$

- 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 version:

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

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

$$x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)};$$

9. **end for**

$\min_{x_j \in \mathbb{R}^n} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$

- 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 [LM23]

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

$$x^{(t+1)} \leftarrow \mathrm{ave}(x_j^{(t+1)} \mid j \in \mathcal{S}_j);$$

9. **end for**

$\min_{x_j \in \mathcal{M}} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$

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

[LM23] J. Li and S. Ma. Federated learning on Riemannian manifolds. Applied Set-Valued Analysis and Optimization, 5(2), 2023. (Local training uses RSVRG)

Euclidean to Riemannian:

Algorithm: A Riemannian federated learning algorithm [LM23]

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

$$x^{(t+1)} \leftarrow \mathrm{ave}(x_j^{(t+1)} \mid j \in \mathcal{S}_j);$$

9. **end for**

$\min_{x_j \in \mathcal{M}} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$

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

[LM23] J. Li and S. Ma. Federated learning on Riemannian manifolds. Applied Set-Valued Analysis and Optimization, 5(2), 2023. (Local training uses RSVRG)

Euclidean to Riemannian:

Algorithm: A Riemannian federated learning algorithm [LM23]

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

$$x^{(t+1)} \leftarrow \mathrm{ave}(x_j^{(t+1)} \mid j \in \mathcal{S}_j);$$

9. **end for**

$\min_{x_j \in \mathcal{M}} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$

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

[LM23] J. Li and S. Ma. Federated learning on Riemannian manifolds. Applied Set-Valued Analysis and Optimization, 5(2), 2023. (Local training uses RSVRG)

Euclidean to Riemannian:

Algorithm: A Riemannian federated learning algorithm [LM23]

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

$$x^{(t+1)} \leftarrow \mathrm{ave}(x_j^{(t+1)} \mid j \in \mathcal{S}_j);$$

9. **end for**

$$\min_{x_j \in \mathcal{M}} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$$

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

[LM23] J. Li and S. Ma. Federated learning on Riemannian manifolds. Applied Set-Valued Analysis and Optimization, 5(2), 2023. (Local training uses RSVRG)

Euclidean to Riemannian:

Algorithm: A Riemannian federated learning algorithm [LM23]

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

$$x^{(t+1)} \leftarrow \mathrm{ave}(x_j^{(t+1)} \mid j \in \mathcal{S}_j);$$

9. **end for**

$\min_{x_j \in \mathcal{M}} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$

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

How to aggregates $\{x_j^{(t+1)}\}_{j \in \mathcal{S}_t}$ on a manifold?

[LM23] J. Li and S. Ma. Federated learning on Riemannian manifolds. Applied Set-Valued Analysis and Optimization, 5(2), 2023. (Local training uses RSVRG)

Euclidean to Riemannian (Aggregation):

- Naive generalization:
  $x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)} \not\Longrightarrow$ Riemannian setting

Euclidean to Riemannian (Aggregation):

- Naive generalization:
  $x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)} \not\Longrightarrow$ Riemannian setting
- An alternative approach:

$$x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)} \Longleftrightarrow x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \|x - x_j^{(t+1)}\|_F^2$$

$$\Longleftrightarrow x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \mathrm{dist}^2(x, x_j^{(t+1)}) \Longrightarrow \text{Riemannian setting;}$$

Euclidean to Riemannian (Aggregation):

- Naive generalization:

$$x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)} \not\Longrightarrow \text{Riemannian setting}$$

- An alternative approach:

$$x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)} \Longleftrightarrow x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \|x - x_j^{(t+1)}\|_F^2$$

$$\Longleftrightarrow x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \mathrm{dist}^2(x, x_j^{(t+1)}) \Longrightarrow \text{Riemannian setting;}$$

- $x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \mathrm{dist}^2(x, x_j^{(t+1)})$: computationally expensive;

Euclidean to Riemannian (Aggregation):

- Naive generalization:
  $$x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)} \not\Longrightarrow \text{Riemannian setting}$$
- An alternative approach:

$$x^{(t+1)} \leftarrow \sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} x_j^{(t+1)} \Longleftrightarrow x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \|x - x_j^{(t+1)}\|_F^2$$

$$\Longleftrightarrow x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \mathrm{dist}^2(x, x_j^{(t+1)}) \Longrightarrow \text{Riemannian setting;}$$

- $x^{(t+1)} = \arg\min_x \sum_{j \in \mathcal{S}_j} \frac{N_j}{\sum_{j \in \mathcal{S}_t} N_j} \mathrm{dist}^2(x, x_j^{(t+1)})$: computationally expensive;
- One step of Riemannian gradient descent (called tangent mean) [LM23]:

$$x^{(t+1)} \leftarrow \mathrm{Exp}_{x^{(t)}}\left(\sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{i \in \mathcal{S}_t} N_i} \mathrm{Exp}_{x^{(t)}}^{-1}(x_j^{(t+1)}) \right);$$

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

Existing Riemannian Federated Learning:

- The reviewed Riemannian federated learning [LM23]
- Riemannian Federated Learning on Compact Submanifolds with Heterogeneous Data [Zha+24]
  - Use projection onto the manifold
  - Allow multiple agents and multiple local updates
- Riemannian Federated Learning via Averaging Gradient Stream [Hua+24]
  - Send tangent vectors rather than the local parameters
  - Allow abstract manifold
  - Allow multiple agents and multiple local updates

[LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds.
[Hua+24] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra. Riemannian Federated Learning via Averaging Gradient Stream. Applied Set-Valued Analysis and Optimization, 2023.
[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.

Existing Riemannian Federated Learning:

- The reviewed Riemannian federated learning [LM23]
- Riemannian Federated Learning on Compact Submanifolds with Heterogeneous Data [Zha+24]
  - Use projection onto the manifold
  - Allow multiple agents and multiple local updates
- Riemannian Federated Learning via Averaging Gradient Stream [Hua+24]
  - Send tangent vectors rather than the local parameters
  - Allow abstract manifold
  - Allow multiple agents and multiple local updates

Propose Riemannian federated learning with differential privacy.

[LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds.
[Hua+24] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra. Riemannian Federated Learning via Averaging Gradient Stream. Applied Set-Valued Analysis and Optimization, 2023.
[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.

Existing Riemannian Federated Learning:

- The reviewed Riemannian federated learning [LM23]
- Riemannian Federated Learning on Compact Submanifolds with Heterogeneous Data [Zha+24]
  - Use projection onto the manifold
  - Allow multiple agents and multiple local updates
- Riemannian Federated Learning via Averaging Gradient Stream [Hua+24]
  - Send tangent vectors rather than the local parameters
  - Allow abstract manifold
  - Allow multiple agents and multiple local updates

Propose Riemannian federated learning with differential privacy.

The later two works appear after the work in this talk.
No differential privacy is considered.

[LM23] J. Li and S. Ma. Federated Learning on Riemannian Manifolds.
[Hua+24] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra. Riemannian Federated Learning via Averaging Gradient Stream. Applied Set-Valued Analysis and Optimization, 2023.
[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.

Outline

Federated Learning:

- (Advantage) Data is stored in each agent.
- (Downside) It is possibly attacked by inference and thus agents' information is leaked.

Encryption method:

- k-anonymity
- Secure multiparty computing
- Homomorphic encryption
- Differential privacy

Federated Learning:

- (Advantage) Data is stored in each agent.
- (Downside) It is possibly attacked by inference and thus agents' information is leaked.

Encryption method:

- *k*-anonymity
- Secure multiparty computing
- Homomorphic encryption
- Differential privacy

Differential Privacy (DP): Differential Privacy offers a rigorous framework for addressing data privacy by precisely quantifying the deviation in the output distribution of a mechanism when a small number of input datasets are modified.

Differential Privacy (DP): Differential Privacy offers a rigorous framework for addressing data privacy by precisely quantifying the deviation in the output distribution of a mechanism when a small number of input datasets are modified.

Definition 1

Let $\mathcal{A}$ be a manifold-valued randomized mechanism, namely, $\mathcal{A} : \mathcal{Z}^n \to \mathcal{M}$. We say that $\mathcal{A}$ is (ϵ, δ)-differential privacy if for any two adjacent datasets D, D', which differ in at most one data point, and for any measurable sets $\mathcal{S} \subseteq \mathcal{M}$, it holds that

$$\mathrm{P}\{\mathcal{A}(D) \subseteq \mathcal{S}\} \leq \exp(\epsilon)\mathrm{P}\{\mathcal{A}(D') \subseteq \mathcal{S}\} + \delta.$$

Algorithm: A Riemannian federated learning algorithm [LM23]

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

$$x^{(t+1)} \leftarrow \mathrm{Exp}_{x^{(t)}}\left(\sum_{j \in \mathcal{S}_t} \frac{N_j}{\sum_{i \in \mathcal{S}_t} N_i} \mathrm{Exp}_{x^{(t)}}^{-1}(x_j^{(t)}) \right);$$

9. **end for**

$$\min_{x_j \in \mathcal{M}} \frac{1}{N_i} \sum_{i=1}^{N_i} f_i(x, z_{i,j})$$

- For t-th outer iteration on j-th agent: $x_j^{(t+1)}$ is an output of a mechanism on $D_j = \{z_{j,1}, \dots, z_{j,N_j}\}$;
- For outer iterations from 0 to t: x_{t+1} is an output of a mechanism on $D = \cup_{j=1,\dots,N} D_j = \{z_1, z_2, \dots, z_n\}$;
- Differential privacy: unlikely to infer any data from $x_j^{(t+1)}$ or $x^{(t)}$;

Algorithm: Riemannian federated learning with differential privacy (PriRFed)

for $t = 0, 1, \ldots, T-1$ **do**
  Sample $\mathcal{S}_t$ from $[N]$ without replacement of size $|\mathcal{S}_t| = s_t$;
  Broadcast the global parameter $x^{(t)}$ to selected agents;
  while $i \in \mathcal{S}_t$ **in parallel do**
    $\tilde{x}_i^{(t+1)} \leftarrow \mathrm{PrivateLocalTraining}(x^{(t)}, D_i, K, \sigma_i, \alpha_t)$;
    Send $\tilde{x}_i^{(t+1)}$ to the server;
  end while
  Aggregate the received local parameters to produce a global parameter $x^{(t+1)}$ using tangent mean;
end for
output option 1: $\tilde{x} = x^{(T)}$;
output option 2: $\tilde{x}$ is uniformly sampled from $\{x^{(t)}\}_{t=1}^{T}$ at random;

- Assumption: Local training satisfies (ϵ, δ)-DP;

Private Federated Learning

Algorithm: Riemannian federated learning with differential privacy (PriRFed)

```
for t = 0, 1, ..., T − 1 do
    Sample S_t from [N] without replacement of size |S_t| = s_t;
    Broadcast the global parameter x^(t) to selected agents;
    while i ∈ S_t in parallel do
        x̃_i^(t+1) ← PrivateLocalTraining(x^(t), D_i, K, σ_i, α_t);
        Send x̃_i^(t+1) to the server;
    end while
    Aggregate the received local parameters to produce a global
     parameter x^(t+1) using tangent mean;
end for
output option 1:  x̃ = x^(T);
output option 2:  x̃ is uniformly sampled from {x^(t)}_{t=1}^T at random;
```

- Assumption: Local training satisfies (ϵ, δ)-DP;
- Then the output $\tilde{x}$ satisfies $(\tilde{\epsilon}, \tilde{\delta})$-DP;

Algorithm: Riemannian federated learning with differential privacy (PriRFed)

```
for t = 0, 1, ..., T − 1 do
    Sample S_t from [N] without replacement of size |S_t| = s_t;
    Broadcast the global parameter x^(t) to selected agents;
    while i ∈ S_t in parallel do
        x̃_i^(t+1) ← PrivateLocalTraining(x^(t), D_i, K, σ_i, α_t);
        Send x̃_i^(t+1) to the server;
    end while
    Aggregate the received local parameters to produce a global
     parameter x^(t+1) using tangent mean;
end for
output option 1:  x̃ = x^(T);
output option 2:  x̃ is uniformly sampled from {x^(t)}_{t=1}^T at random;
```

- Assumption: Local training satisfies (ϵ, δ)-DP;
- Then the output $\tilde{x}$ satisfies $(\tilde{\epsilon}, \tilde{\delta})$-DP;

The derivation relies on the existing properties of differential privacy

Properties of differential privacy

Theorem 2 (Post-processing [DR+14, Proposition 2.1])

Suppose that $\mathcal{A} : \mathcal{Z}^n \to \mathcal{M}$ *is a randomized algorithm that is* (ϵ, δ)*-DP. Let* $P : \mathcal{M} \to \mathcal{M}'$ *be an arbitrary mapping. Then* $P \circ \mathcal{A} : \mathcal{Z}^n \to \mathcal{M}'$ *is* (ϵ, δ)*-DP.*

Theorem 3 (Sequential composition theorem [DR+14, Theorem 3.16])

Suppose that $\mathcal{A}_i : \mathcal{Z}^n \to \mathcal{M}_i$ *is an* (ϵ_i, δ_i)*-DP algorithm for* $i \in [k]$*. Then if* $\mathcal{A}_{[k]} : \mathcal{Z}^n \to \prod_{i=1}^k \mathcal{M}_i$ *is defined to be* $\mathcal{A}_{[k]} := (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k)$*, then* $\mathcal{A}_{[k]}$ *is* $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$*-DP.*

Theorem 4 (Advanced composition theorem [DRV10; KOV17; Whi+23])

Suppose that $\mathcal{A}_i$ *is an* (ϵ, δ)*-DP algorithm for* $i \in [k]$*. Then the adaptive composition* $\mathcal{A}_{[k]}$ *of* $\mathcal{A}_1, \ldots, \mathcal{A}_k$*, i.e.,* $\mathcal{A}_{[k]} = \mathcal{A}_k \circ \mathcal{A}_{k-1} \circ \cdots \circ \mathcal{A}_1$*, is* $(\epsilon', \delta' + k\delta)$*-DP with* $\epsilon' = \sqrt{2k \log(1/\delta')}\epsilon + k\epsilon(\exp(\epsilon) - 1)$ *and any* $\delta' > 0$.

Definition 5 (Subsample)

Given a dataset D of n points, namely $D = \{D_1, D_2, \ldots, D_n\}$ (where D_i may be a dataset), an operator $\mathcal{S}_\rho^{\mathrm{wor}}$, called subsample, randomly chooses a sample from the uniform distribution over all subsets of D of size m without replacement. The ratio $\rho := m/n$ is called the sampling rate of the operator $\mathcal{S}_\rho^{\mathrm{wor}}$. Formally, letting $\{i_1, i_2, \ldots, i_m\}$ be the sampled indices, $\mathcal{S}_\rho^{\mathrm{wor}}$ is defined by

$$\mathcal{S}_\rho^{\mathrm{wor}} : \mathcal{Z}^{N_1} \times \mathcal{Z}^{N_2} \times \ldots \times \mathcal{Z}^{N_n} \to \mathcal{Z}^{N_{i_1}} \times \mathcal{Z}^{N_{i_2}} \times \ldots \times \mathcal{Z}^{N_{i_m}} :$$
$$(D_1, D_2, \ldots, D_n) \mapsto (D_{i_1}, D_{i_2}, \ldots, D_{i_m}).$$

Lemma 6 (Subsampling lemma)

Let $\mathcal{S}_\rho^{\mathrm{wor}} : \mathcal{Z}^{N_1} \times \mathcal{Z}^{N_2} \times \ldots \times \mathcal{Z}^{N_n} \to \mathcal{Z}^{N_{i_1}} \times \mathcal{Z}^{N_{i_2}} \times \cdots \times \mathcal{Z}^{N_{i_m}}$ be a subsample operator (defined by Definition 5) with sampling rate $\rho = m/n$ and sampled indices $\{i_1, i_2, \ldots, i_m\}$. Suppose that $\mathcal{A} : \mathcal{Z}^{N_{i_1}} \times \mathcal{Z}^{N_{i_2}} \times \cdots \times \mathcal{Z}^{N_{i_m}} \to \mathcal{M}$ is a mechanism obeying (ϵ, δ)-DP. Then $\mathcal{A}' := \mathcal{A} \circ \mathcal{S}_\rho^{\mathrm{wor}} : \mathcal{Z}^{N_1} \times \mathcal{Z}^{N_2} \times \ldots \times \mathcal{Z}^{N_n} \to \mathcal{M}$ provides (ϵ', δ')-DP, where $\epsilon' = \log(1 + \rho(\exp(\epsilon) - 1))$ and $\delta' \le \rho\delta$.

Definition 7 (c-stable transformation [McS09, Definition 2])

Let $\bar{\mathcal{Z}} := \mathcal{Z}^{N_1} \times \mathcal{Z}^{N_2} \times \ldots \times \mathcal{Z}^{N_n}$. A transformation $T : \bar{\mathcal{Z}} \to \bar{\mathcal{Z}}$ is said c-stable if for any two data sets $D_1, D_2 \in \bar{\mathcal{Z}}$, the following holds

$$|T(D_1) \oplus T(D_2)| \leq c|D_1 \oplus D_2|.$$

where c is a constant.

Theorem 8 (Pre-processing)

Let $\bar{\mathcal{Z}} := \mathcal{Z}^{N_1} \times \mathcal{Z}^{N_2} \times \ldots \times \mathcal{Z}^{N_n}$. Suppose that $\mathcal{A} : \bar{\mathcal{Z}} \to \mathcal{M}$ be (ϵ, δ)-DP, and $T : \bar{\mathcal{Z}} \to \bar{\mathcal{Z}}$ be an arbitrary c-stable transformation. Then the composition $\mathcal{A} \circ T$ provides $(c\epsilon, c\exp((c-1)\epsilon)\delta)$-DP.

Theorem 9 (Privacy guarantee of PriRFed)

If the privately local training satisfies (ϵ, δ)-DP for agent $i \in [N]$, then PriRFed obeys (ϵ', δ')-DP with $\epsilon' = \min(T\tilde{\epsilon}, \sqrt{2T \ln(1/\hat{\delta})}\tilde{\epsilon} + T\tilde{\epsilon}(\exp(\tilde{\epsilon}) - 1))$ *and* $\delta' = \hat{\delta} + T\tilde{\delta}$ *for any* $\hat{\delta} \in (0, 1)$*, where* $\tilde{\epsilon} = \log(1 + \rho(\exp(s\epsilon) - 1))$*,* $\tilde{\delta} = \rho s \delta$*,* $\mathcal{S}_t \subseteq [N]$ *with* $s_t = s \leq N$*, and* $\rho = s/N$ *is the sampling rate.*

How to make the local training algorithm satisfy (ϵ, δ)-DP?

Private Federated Learning

How to make the local training algorithm satisfy (ϵ, δ)-DP?

Add noise to the local training algorithm.

How to make the local training algorithm satisfy (ϵ, δ)-DP?

Add noise to the local training algorithm.

Gaussian Mechanism:

- In Euclidean setting:
  - Given a function $H : \mathcal{Z}^n \to \mathbb{R}^m$, the Gaussian mechanism associated with H is given by $\mathcal{H}(D) = H(D) + \varepsilon$, where ε is a Gaussian noise.
  - Letting $\Delta H = \sup_{D \sim D'} \|H(D) - H(D')\|_2$ be the sensitivity of H, if $\sigma^2 \geq 2 \log(1.25/\delta)\Delta_H^2/\epsilon^2$, then $\mathcal{H}$ obeys (ϵ, δ)-DP.

How to make the local training algorithm satisfy (ϵ, δ)-DP?

Add noise to the local training algorithm.

Gaussian Mechanism:

- In Euclidean setting:
  - Given a function $H : \mathcal{Z}^n \to \mathbb{R}^m$, the Gaussian mechanism associated with H is given by $\mathcal{H}(D) = H(D) + \varepsilon$, where ε is a Gaussian noise.
  - Letting $\Delta H = \sup_{D \sim D'} \|H(D) - H(D')\|_2$ be the sensitivity of H, if $\sigma^2 \geq 2\log(1.25/\delta)\Delta_H^2/\epsilon^2$, then $\mathcal{H}$ obeys (ϵ, δ)-DP.
- In Riemannian setting:
  - Given any $x \in \mathcal{M}$, for a function $H : \mathcal{Z}^n \to \mathrm{T}_x\mathcal{M}$, the Gaussian mechanism associated with f is given by $\mathcal{H}(D) = H(D) + \varepsilon$, where ε is a tangent Gaussian noise;
  - Letting $\Delta H = \sup_{D \sim D'} \|H(D) - H(D')\|_x$ be the sensitivity of H, if $\sigma^2 \geq 2\log(1.25/\delta)\Delta_H^2/\epsilon^2$, then $\mathcal{H}$ obeys (ϵ, δ)-DP.

Algorithm: A Riemannian SGD Algorithm with DP (DP-RSGD) [Han+24]

01. Set $x_{i,0}^{(t)} \leftarrow x^{(t)}$;
02. **for** $k = 0, 1, \ldots, K-1$ **do**
03. Select $\mathcal{B}_k \subseteq [N_i]$ of size b_i, where the samples are randomly selected uniformly without replacement;
04. Set $\eta_k^{(t)} \leftarrow \left(\frac{1}{b_i} \sum_{p \in \mathcal{B}_k} \mathrm{clip}_\tau(\mathrm{grad} f_{i,p}(x_{i,k}^{(t)})) \right) + \varepsilon_{i,k}^{(t)}$, where $\mathrm{clip}_\tau(\cdot) : \mathrm{T}_x\mathcal{M} \to \mathrm{T}_x\mathcal{M} : v \mapsto \min\{\tau / \|v\|_x, 1\}$ and $\varepsilon_{i,k}^{(t)} \sim \mathcal{N}_{x_{i,k}^{(t)}}(0, \sigma_i^2)$;
05. Update $x_{i,k+1}^{(t)} \leftarrow \mathrm{Exp}_{x_{i,k}^{(t)}}(-\alpha \eta_k^{(t)})$;
06. **end for**
07. **output option 1:** $\tilde{x}_i^{(t+1)} \leftarrow x_{i,K}^{(t)}$ if **Option 1** is chosen in PriRFed;
08. **output option 2:** $\tilde{x}_i^{(t+1)}$ as uniformly selected from $\{x_{i,k}^{(t)}\}_{k=0}^{K}$ otherwise;

- Add tangent Gaussian noise $\varepsilon_{i,k}^{(t)}$ to the stochastic gradient direction;
- If $\sigma_i^2 = o_i \frac{K \log(1/\delta) \tau^2}{N_i^2 \epsilon^2}$, then DP-RSGD satisfies (ϵ, δ)-DP.

[Han+24] A. Han, B. Mishra, P. Jawanpuria, J, Gao. Differentially private Riemannian optimization. Machine Learning, 113, P.1133–1161, 2024.

Algorithm: A Riemannian SVRG Algorithm with DP (DP-RSVRG) [Upt+23]

01. Set $\tilde{x}_{i,0}^{(t)} \leftarrow x^{(t)}$
02. **for** $k = 0, 1, \ldots, K-1$ **do**
03. $\quad x_{k+1,0}^{(t)} \leftarrow \tilde{x}_{i,k}^{(t)}$ and $g_{k+1}^{(t)} \leftarrow \frac{1}{N_i} \sum_{j=1}^{N_i} \mathrm{clip}_\tau(\mathrm{grad} f_{i,j}(\tilde{x}_{i,k}^{(t)}))$;
04. $\quad$ **for** $j = 0, 1, \ldots, m-1$ **do**
05. $\quad\quad$ Randomly pick $l_j \in [N_i]$ and $\varepsilon_{k+1,j}^{(t)} \sim \mathcal{N}_{x_{k+1,j}^{(t)}}(0, \sigma_i^2)$;
06. $\quad\quad \eta_{k+1,j}^{(t)} \leftarrow \mathrm{clip}_\tau(\mathrm{grad} f_{i,l_j}(x_{k+1,j}^{(t)})) - \Gamma_{\tilde{x}_{i,k}^{(t)}}^{x_{k+1,j}^{(t)}}(\mathrm{clip}_\tau(\mathrm{grad} f_{i,l_j}(\tilde{x}_{i,k}^{(t)}) - g_{k+1}^{(t)})) + \varepsilon_{k+1,j}^{(t)}$;
07. $\quad\quad x_{k+1,j+1}^{(t)} \leftarrow \mathrm{Exp}_{x_{k+1,j}^{(t)}}(-\alpha_t \eta_{k+1,j}^{(t)})$;
08. $\quad$ **end for**
09. $\quad \tilde{x}_{i,k+1}^{(t)} \leftarrow x_{k+1,m}^{(t)}$;
10. **end for**
11. **output option 1**: $\tilde{x}_i^{(t+1)} \leftarrow x_{K,m}^{(t)}$;
12. **output option 2**: $\tilde{x}_i^{(t+1)}$ is uniformly randomly selected from $\{\{x_{k+1,j}^{(t)}\}_{j=0}^{m-1}\}_{k=0}^{K-1}$;

- Add tangent Gaussian noise $\varepsilon_{i,k}^{(t)}$ to the stochastic gradient direction;
- If $\sigma_i^2 = \tilde{o}_i \frac{mK \log(1/\delta)\tau^2}{N_i^2 \epsilon^2}$, then DP-RSVRG satisfies (ϵ, δ)-DP.

[Upt+23] A. Utpala, A. Han, P. Jawanpuria, B. Mishra. Improved Differentially Private Riemannian Optimization: Fast Sampling and Variance Reduction. Transactions on Machine Learning Research, ISSN:2835–8856, 2023.

Outline

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth, i.e., Lipschitz continuous Riemannian gradient for any i, j;

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth, i.e., Lipschitz continuous Riemannian gradient for any i, j;

Definition 10 (Geodesic Lipschitz continuity)

A function $f : \mathcal{M} \to \mathbb{R}$ is called geodesically L_f-Lipschitz continuous if for any $x, y \in \mathcal{M}$, there exists $L_f \geq 0$ such that $|f(x) - f(y)| \leq L_f \mathrm{dist}(x, y)$.

Definition 11 (Geodesic smoothness [ZS16])

A differentiable function $f : \mathcal{M} \to \mathbb{R}$ is call geodesically L_g-smooth if its gradient is L_g-Lipschitz continuous, that is, $\|\mathrm{grad} f(x) - \Gamma_y^x(\mathrm{grad} f(y))\|_x \leq L_g \mathrm{dist}(x, y)$.

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth, i.e., Lipschitz continuous Riemannian gradient for any i, j;

Theorem 10 (Nonconvex, $K = 1$)

*Consider PriRFed-DP-RSGD with **Option 1**. Set $s_t = N$, $K = 1$, $b_i = N_i$, and $\alpha_t = \alpha \leq 1/L_g$. It holds that*

$$\min_{0 \leq t \leq T} \mathbb{E}[\|\mathrm{grad} f(x^{(t)})\|^2] \leq \frac{2}{T\alpha}(f(x^{(0)}) - f(x^*)) + \frac{d L_g \alpha}{(\sum_{i=1}^{N} N_i)^2} \sum_{i=1}^{N} N_i^2 \sigma_i^2,$$

where the expectation is taken over $\varepsilon_{i,0}^{(t)}$, and $x^* = \arg\min_{x \in \mathcal{M}} f(x)$.

Theorem 11 (Nonconvex, $K = 1$)

Same parameters except that the step sizes $\{\alpha_t\}_{t=0}^{T-1}$ satisfies

$$\lim_{T\to\infty} \sum_{t=0}^{T-1} \alpha_t = \infty, \ \lim_{T\to\infty} \sum_{t=0}^{T-1} \alpha_t^2 < \infty, \text{ and } L_g \alpha_t \leq 2\delta,$$

for a constant $\delta \in (0, 1)$. It holds that $\liminf_{T\to\infty} \mathbb{E}\left[\|\mathrm{grad} f(x^{(t)})\|^2\right] = 0$.

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth, i.e., Lipschitz continuous Riemannian gradient for any i, j;

Theorem 10 (Nonconvex, $K = 1$)

*Consider PriRFed-DP-RSGD with **Option 1**. Set $s_t = N$, $K = 1$, $b_i = N_i$, and $\alpha_t = \alpha \leq 1/L_g$. It holds that*

$$\min_{0 \leq t \leq T} \mathbb{E}[\|\mathrm{grad} f(x^{(t)})\|^2] \leq \frac{2}{T\alpha}(f(x^{(0)}) - f(x^*)) + \frac{d L_g \alpha}{(\sum_{i=1}^N N_i)^2} \sum_{i=1}^N N_i^2 \sigma_i^2,$$

where the expectation is taken over $\varepsilon_{i,0}^{(t)}$, and $x^* = \arg\min_{x \in \mathcal{M}} f(x)$.

Theorem 11 (Nonconvex, $K = 1$)

Same parameters except that the step sizes $\{\alpha_t\}_{t=0}^{T-1}$ satisfies

$$\lim_{T \to \infty} \sum_{t=0}^{T-1} \alpha_t = \infty, \ \lim_{T \to \infty} \sum_{t=0}^{T-1} \alpha_t^2 < \infty, \text{ and } L_g \alpha_t \leq 2\delta,$$

for a constant $\delta \in (0, 1)$. It holds that $\liminf_{T \to \infty} \mathbb{E}\left[\|\mathrm{grad} f(x^{(t)})\|^2\right] = 0$.

The noise does not influence the convergences for decaying step sizes.

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- Manifold is complete;
- All iterates stay in a compact subset $\mathcal{W} \subset \mathcal{M}$ and the sectional curvature in $\mathcal{W}$ is bounded in $[\kappa_{\min}, \kappa_{\max}]$;
- The function f_i is geodesically convex in $\mathcal{W}$ for any i;

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- Manifold is complete;
- All iterates stay in a compact subset $\mathcal{W} \subset \mathcal{M}$ and the sectional curvature in $\mathcal{W}$ is bounded in $[\kappa_{\min}, \kappa_{\max}]$;
- The function f_i is geodesically convex in $\mathcal{W}$ for any i;

Definition 12 (Geodesic convexity [ZS16])

A function $\mathcal{M} \to \mathbb{R}$ is called geodesically convex on $\mathcal{W}$ if for any $x, y \in \mathcal{W}$, a geodesic $\gamma : [0, 1] \to \mathcal{M}$ satisfying $\gamma(0) = x$, $\gamma(1) = y$, and $\gamma(t) \in \mathcal{W}, \forall t \in [0, 1]$, it holds that $f(\gamma(t)) \leq (1 - t)f(x) + tf(y)$.

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- Manifold is complete;
- All iterates stay in a compact subset $\mathcal{W} \subset \mathcal{M}$ and the sectional curvature in $\mathcal{W}$ is bounded in $[\kappa_{\min}, \kappa_{\max}]$;
- The function f_i is geodesically convex in $\mathcal{W}$ for any i;

Theorem 12 (Convex, $K = 1$)

Consider PriRFed-DP-RSGD with ***Option 1****. Set $s_t = N$, $K = 1$, $b_i = N_i$, and $\alpha_t = \alpha \leq 1/(2L_g)$. It holds that*

$$\mathbb{E}[f(x^{(T)}) - f(x^*)] \leq \frac{\zeta \mathrm{dist}^2(x^{(0)}, x^*)}{2\alpha(\zeta + T - 1)} + \frac{T(6\zeta + T - 1)d}{16L_g(\zeta + T - 1)(\sum_{i=1}^N N_i)^2} \sum_{i=1}^N N_i^2 \sigma_i^2,$$

where the expectation is taken over $\varepsilon_{i,0}^{(t)}$, $x^* = \arg\min_{x \in \mathcal{M}} f(x)$, and ζ depends on $\kappa_{\min}$.

- An optimal value of T;
- Consistent with [LM23, Theorem 9] when non-DP;

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- The function f_i is geodesically convex in $\mathcal{W}$ for any i;
- f satisfies the Riemannian Polyak-Lojasiewicz (RPL) condition with a positive constant μ;
- For any i and $x \in \mathcal{M}$, $\|\mathrm{grad} f_i(x) - \mathrm{grad} f(x)\|^2 \leq v_i$ and $\mathbb{E}[v_i] = v$.

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- The function f_i is geodesically convex in $\mathcal{W}$ for any i;
- f satisfies the Riemannian Polyak-Lojasiewicz (RPL) condition with a positive constant μ;
- For any i and $x \in \mathcal{M}$, $\|\mathrm{grad} f_i(x) - \mathrm{grad} f(x)\|^2 \leq v_i$ and $\mathbb{E}[v_i] = v$.

Definition 13 (Riemannian Polyak-Lojasiewicz condition)

A function $\mathcal{M} \to \mathbb{R}$ satisfies the Riemannian Polyak-Lojasiewicz (RPL) condition with a positive constant μ if $f(x) - f(x^*) \leq \frac{1}{2\mu}\|\mathrm{grad} f(x)\|^2$, where x^* is a global minimizer of f.

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- The function f_i is geodesically convex in $\mathcal{W}$ for any i;
- f satisfies the Riemannian Polyak-Lojasiewicz (RPL) condition with a positive constant μ;
- For any i and $x \in \mathcal{M}$, $\|\mathrm{grad} f_i(x) - \mathrm{grad} f(x)\|^2 \leq v_i$ and $\mathbb{E}[v_i] = v$.

Theorem 13 (Convex and RPL, $K = 1$)

Consider PriRFed-DP-RSGD with ***Option 1****. Set $N_i = N_j$ ($\forall\, i, j \in [N]$), $s_t = S \leq N$, $K = 1$, and $b_i = N_i$. Then it holds that*

$$\mathbb{E}[f(x^{(T)})] - f(x^*) \leq P^T(f(x^{(0)}) - f(x^*)) + (1 - P^T)\left(\kappa_0 \frac{\sum_{i \in S} \sigma_i^2}{S^2} + \kappa_1 \frac{N-S}{S(N-1)}\right),$$

where $P = (1 - 2\mu\alpha + \mu\alpha^2 L_g)$, $\kappa_0 = \frac{L_g \alpha d}{2\mu(2 - L_g\alpha)}$, *and* $\kappa_1 = v\kappa_0 = \frac{L_g \alpha d v}{2\mu(2 - L_g\alpha)}$.

- An optimal value of T;
- The second term $\neq 0$ even if $\sigma_i = 0$ (non-DP);
- Consistent with the Euclidean FL in [Wei+22];

PriRFed with DP-RSGD

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- Manifold is complete;
- All iterates stay in a compact subset $\mathcal{W} \subset \mathcal{M}$ and the sectional curvature in $\mathcal{W}$ is bounded in $[\kappa_{\min}, \kappa_{\max}]$;

Theorem 14 (Nonconvex, $K > 1$)

*Consider PriRFed-DP-RSGD and **Option 2**. Set $s_t = 1$, $K > 1$, $\alpha_t = \alpha < (\beta^{\frac{1}{K-1}} - 1)/\beta$ with $\beta > 1$ being a free constant. It holds that*

$$\mathbb{E}[\|\mathrm{grad} f(\tilde{x})\|^2] \leq \frac{f(x^{(0)}) - f(x^*)}{KT\delta_{\min}} + q_0 q,$$

where $\delta_{\min} = \alpha(1 - \frac{(1+\alpha\beta)^{K-1}}{\beta})$, $q_0 = \frac{\alpha\beta(L_g + 2(1+\alpha\beta)^{K-1}\zeta)}{2(\beta-(1+\alpha\beta)^{K-1})}$, $q = L_f^2 + d\sigma^2$, and $\sigma = \max_{i=1,2,\ldots,N} \sigma_i$.

Allow multiple inner iteration, i.e., $K > 1$, but only one agent is used, i.e., $s_t = 1$;

PriRFed with DP-RSVRG

Assumptions:

- The function f_{ij} is geodesic L_f-Lipschitz continuous for any i, j;
- The function f_{ij} is geodesically L_g-smooth;
- Manifold is complete;
- All iterates stay in a compact subset $\mathcal{W} \subset \mathcal{M}$ and the sectional curvature in $\mathcal{W}$ is bounded in $[\kappa_{\min}, \kappa_{\max}]$;

Theorem 15 (Nonconvex, $K > 1$, $m > 1$)

Consider PriRFed-DP-RSVRG with ***Option 2****. Set $s_t = 1$, $K > 1$, $m = \lfloor 10N/(3\zeta^{1/2}) \rfloor > 1$, and $\alpha_t = \alpha \leq 1/(10 L_g N^{2/3} \zeta^{1/2})$. It holds that*

$$\mathbb{E}[\|\mathrm{grad} f(\tilde{x})\|^2] \leq c \frac{L_g \zeta^{1/2}}{N^{1/3} K T} [f(x^{(0)}) - f(x^*)] + \frac{d\sigma^2}{100\zeta^{1/2}} \left(\frac{1}{N^{2/3}} + \frac{1}{N} \right),$$

where $c > 10N/(3m)$ is constant, $\sigma = \max_{i=1,2,\ldots,N} \sigma_i$, and ζ depends on $\kappa_{\min}$.

Convergence Analysis

- Consider “nonconvex”, “convex”, “RPL” scenarios;
- Some consistent with existing Riemannian FL in [LM23];
- Some consistent with Euclidean FL-DP in [Wei+22];
- Not guarantee convergence for all parameter settings;
- Importantly, $s_t > 1$ and $K > 1$ not considered;

Outline

Principal Eigenvector over Sphere Manifold

$$\min_{x \in \mathbb{S}^d} f(x) := -\sum_{i=1}^{N} p_i \left(\frac{1}{N_i} \sum_{j=1}^{N_i} x^T (z_{i,j} z_{i,j}^T) x \right), \tag{4.1}$$

where $\mathbb{S}^d = \{x \in \mathbb{R}^{d+1} : \|x\|_2 = 1\}$, $p_i = \frac{N_i}{\sum_{i=1}^{N} N_i}$, and $D_i = \{z_{i,1}, \ldots, z_{i,N_i}\}$ with $z_{i,j} \in \mathbb{R}^{d+1}$.

Principal Eigenvector over Sphere Manifold

$$\min_{x \in \mathbb{S}^d} f(x) := -\sum_{i=1}^{N} p_i \left(\frac{1}{N_i} \sum_{j=1}^{N_i} x^T (z_{i,j} z_{i,j}^T) x \right)$$

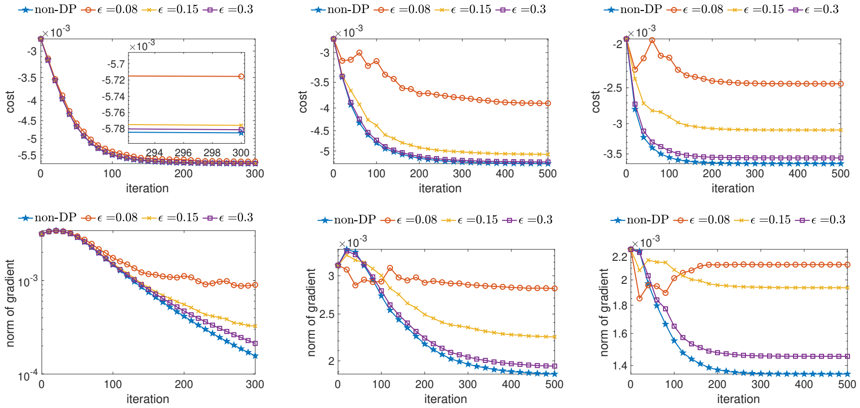
Synthetic data:

- construct a $(d+1) \times (d+1)$ diagonal matrix $\Sigma_i = \mathrm{diag}\{1, 1 - 1.1\nu, \ldots, 1 - 1.4\nu, |y_1|/(d+1), |y_2|/(d+1), \ldots\}$ where ν is the eigengap and $y_1, y_2, \ldots$ are drawn from the standard Gaussian distribution;
- construct $Z_i = U_i \Sigma_i V_i$ where U_i, V_i of size $N_i \times (d+1)$ and $(d+1) \times (d+1)$ are random column orthonormal matrices

MNIST:

- 60,000 hand-written images of size 28×28 (in the case, $d + 1 = 28^2 = 784$);

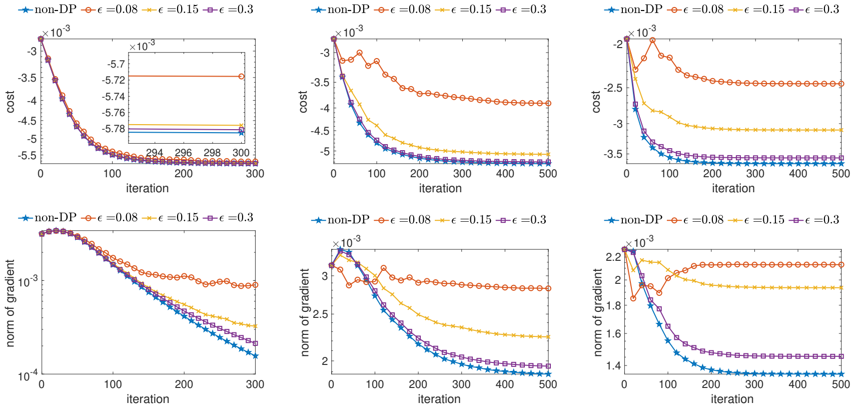
Synthetic data: An average result of 10 random runs ($d = 24$)



(a) $N = 16$, $N_i = 70$, DP-RSGD (b) $N = 16$, $N_i = 70$, DP-RSGD (c) $N = 16$, $N_i = 100$, DP-RSVRG

- The smaller ϵ is, the larger the σ_i is;
- Left column: $s_t = N$, $K = 1$, , $b_i = N_i$ and $\alpha_t = 1/(2L_g)$;
- Middle column: $s_t = 1$, $K = 5$, $b_i = N_i/2$, and $\alpha_t = 1.0$;
- Right column: $s_t = 1$, $K = 2$, $m = \lfloor 10N/3 \rfloor$, and $\alpha_t = 1/(10N^{2/3}L_g)$;

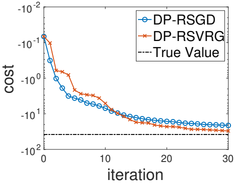
Synthetic data: An average result of 10 random runs ($d = 24$)



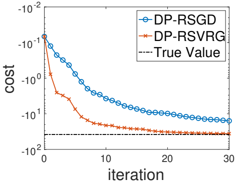
(a) $N = 16$, $N_i = 70$, DP-RSGD (b) $N = 16$, $N_i = 70$, DP-RSGD (c) $N = 16$, $N_i = 100$, DP-RSVRG

Large noise slows down the convergence speed and increases the noise floor.

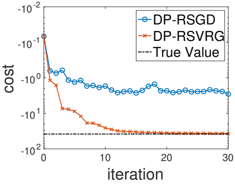
MNIST: An average result of 10 random runs ($d = 783$)



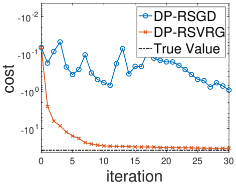
(d) $N = 50, N_i = 1200$



(e) $N = 60, N_i = 1000$



(f) $N = 80, N_i = 750$



(g) $N = 100, N_i = 600$

- $\epsilon = 0.15$, $\delta = 10^{-4}$, $s_t = 1$, $K = 3$, $b_i = N_i/2$, and $\alpha_t = 0.1$;
- DP-SVRG: $m = \lfloor 10N/3 \rfloor$;

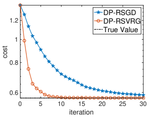
$$\arg\min_{X \in \mathbb{S}^d_{++}} f(X) = \sum_{i=1}^{N} \frac{p_i}{N_i} \sum_{j=1}^{N_i} \|\mathrm{logm}(X^{-1/2} Z_{i,j} X^{-1/2})\|_F^2, \tag{4.2}$$

where $p_i = N_i / \sum_{i=1}^{N} N_i$, $D_i = \{Z_{i,1}, \dots, Z_{i,N_i}\}$, $\mathrm{logm}(\cdot)$ is the principal matrix logarithm, $f_{i,j}(X) = \|\mathrm{logm}(X^{-1/2} Z_{i,j} X^{-1/2})\|_F^2$, and $f_i(X) = \frac{1}{N_i} \sum_{j=1}^{N_i} f_{i,j}(X)$.

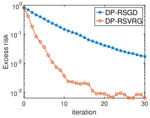
- Riemannian metric: the affine-invariant metric $\langle U, V \rangle_X = \mathrm{trace}(UX^{-1}VX^{-1})$;
- The exponential map $\mathrm{Exp}_X(U) = X^{1/2} \mathrm{expm}(X^{-1/2} U X^{-1/2}) X^{1/2}$ with $\mathrm{expm}(\cdot)$ the principal matrix exponential.

Synthetic data

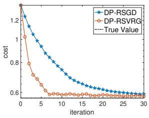
- SPD $\sim$ Wishart distribution $W(I_d/d, d)$ with a diameter bound $D_{\mathcal{W}} = 1$ and $d = 2$;
- $\epsilon = 0.15$, $\delta = 10^{-4}$, $s_t = 1$, $K = 3$, $m = 10$, $\alpha_t = 0.05$, and $\tau = 1.0$;



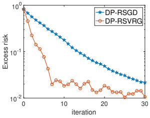
(h) $N = 20$, $N_i = 250$



(i) $N = 20$, $N_i = 250$



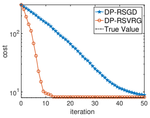
(j) $N = 50$, $N_i = 100$



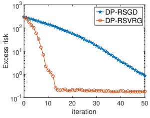
(k) $N = 50$, $N_i = 100$

PATHMNIST

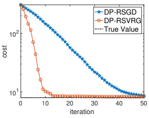
- 20000 chosen from 89996 RGB images
- Descriptor of each image: 9×9 SPD matrix [TPM06];
- $\epsilon = 0.15$, $\delta = 10^{-4}$, $s_t = 1$, $K = 3$, $m = 10$, $\alpha_t = 0.3$, and $\tau = 1.0$;



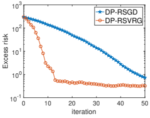
(l) $N = 50$, $N_i = 400$



(m) $N = 50$, $N_i = 400$



(n) $N = 100$, $N_i = 200$



(o) $N = 100$, $N_i = 200$

$$h(x) := \arg\min_{x \in \mathcal{H}^d} f(x) = \sum_{i=1}^{N} p_i \sum_{j=1}^{N_i} \frac{1}{N_i} \alpha_{i,j}(w) \mathrm{dist}^2(x, y_{i,j}), \tag{4.3}$$

where features $w = \{w_{i,j}\}$ with $w_{i,j} \in \mathbb{R}^r$ and hyperbolic embeddings $y_{i,j} \in \mathcal{H}^d$ are given,

- $\alpha_i(w) = (\alpha_{i,1}(w), \ldots, \alpha_{i,N_i}(w))^T \in \mathbb{R}^{N_i}$;
- $\alpha_i(w) = (K_i + \gamma I)^{-1} K_{i,w}$ with a given $\gamma > 0$;
- $K_i \in \mathbb{R}^{N_i \times N_i}$ and $(K_i)_{l,j} = k(w_{i,l}, w_{i,j})$;
- $K_{i,w} \in \mathbb{R}^{N_i}$ and $(K_{i,w})_j = k(w_{i,j}, w)$;
- The kernel function $k(w, w') = \exp(-\|w - w'\|_2^2 / (2\bar{v})^2)$.

$$h(x) := \arg\min_{x \in \mathcal{H}^d} f(x) = \sum_{i=1}^{N} p_i \sum_{j=1}^{N_i} \frac{1}{N_i} \alpha_{i,j}(w) \mathrm{dist}^2(x, y_{i,j}), \tag{4.3}$$

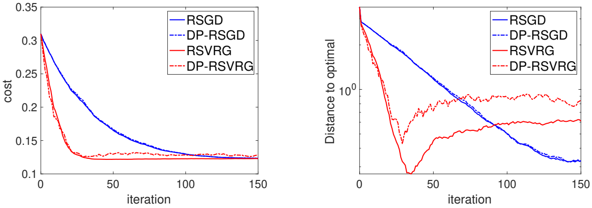
where features $w = \{w_{i,j}\}$ with $w_{i,j} \in \mathbb{R}^r$ and hyperbolic embeddings $y_{i,j} \in \mathcal{H}^d$ are given,

- $\alpha_i(w) = (\alpha_{i,1}(w), \ldots, \alpha_{i,N_i}(w))^T \in \mathbb{R}^{N_i}$;
- $\alpha_i(w) = (K_i + \gamma I)^{-1} K_{i,w}$ with a given $\gamma > 0$;
- $K_i \in \mathbb{R}^{N_i \times N_i}$ and $(K_i)_{l,j} = k(w_{i,l}, w_{i,j})$;
- $K_{i,w} \in \mathbb{R}^{N_i}$ and $(K_{i,w})_j = k(w_{i,j}, w)$;
- The kernel function $k(w, w') = \exp(-\|w - w'\|_2^2 / (2\bar{v})^2)$.

- Hyperbolic manifold $\mathcal{H}^d := \{x \in \mathbb{R}^{d+1} : \langle x, x \rangle_{\mathcal{L}} = -1\}$ with $\langle x, y \rangle_{\mathcal{L}} = x^T y - 2x_1 y_1$ (Riemannian metric);
- The exponential map: $\mathrm{Exp}_x(v) = \cosh(\|v\|_{\mathcal{L}}) x + v \frac{\sinh(\|v\|_{\mathcal{L}})}{\|v\|_{\mathcal{L}}}$;
- The logarithm map $\mathrm{Exp}_x^{-1}(y) = \frac{\cosh^{-1}(-\langle x, y \rangle_{\mathcal{L}})}{\sinh(\cosh^{-1}(-\langle x, y \rangle_{\mathcal{L}}))} (y + \langle x, y \rangle_{\mathcal{L}} x)$;
- The distance function $\mathrm{dist}(x, y) = \cosh^{-1}(-\langle x, y \rangle_{\mathcal{L}})$.

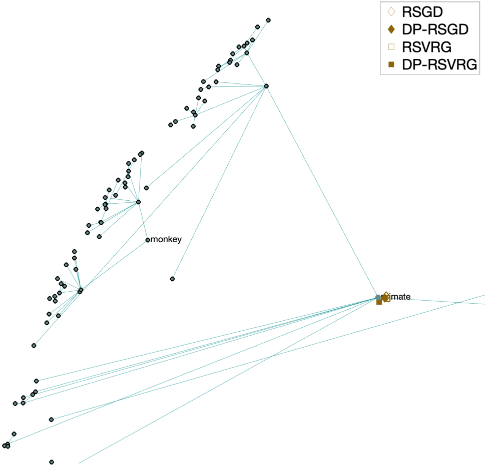
Mammals Substree of WordNet

- Mammals Substree are embedded on $\mathcal{H}^2$;
- Transitive closure containing $n = 1180$ nodes and 6540 edges;
- The feature are stemmed from Laplacian eigenmap to dimension $r = 3$;
- The word "primate" as the test sample;



(p) (q)

(r) Full embeddings



(s) Partial embedding

- Introduced the federated learning;
- Introduced the differential privacy;
- Proposed a Riemannian federated learning framework with differential privacy guaranteed (PriRFed);
- Convergence analysis for two instances of PriRFed, i.e., with DP-RSGD and DP-RSVRG;
- Numerical experiments verify the performance;

For more details, see

Zhenwei Huang, Wen Huang, Pratik Jawanpuria, and Bamdev Mishra. Federated learning on Riemannian manifolds with differential privacy. arXiv:2404.10029, 2024.

Thank you

Thank you for your attention!

References I

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.

Cynthia Dwork, Aaron Roth, et al. "The algorithmic foundations of differential privacy". In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. "Boosting and differential privacy". In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE. 2010, pp. 51–60.

Andi Han et al. "Differentially private Riemannian optimization". In: *Machine Learning* 113.3 (2024), pp. 1133–1161.

Zhenwei Huang et al. *Riemannian Federated Learning via Averaging Gradient Stream*. 2024. arXiv: 2409.07223 [cs.LG]. URL: https://arxiv.org/abs/2409.07223.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. "The Composition Theorem for Differential Privacy". In: *IEEE Transactions on Information Theory* 63.6 (2017), pp. 4037–4049.

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.

Frank D McSherry. “Privacy integrated queries: an extensible platform for privacy-preserving data analysis”. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 2009, pp. 19–30.

Oncel Tuzel, Fatih Porikli, and Peter Meer. “Region covariance: A fast descriptor for detection and classification”. In: *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9*. Springer. 2006, pp. 589–600.

Saiteja Uptake et al. "Improved Differentially Private Riemannian Optimization: Fast Sampling and Variance Reduction". In: *Transactions on Machine Learning Research* (2023). ISSN: 2835-8856.

Kang Wei et al. "User-Level Privacy-Preserving Federated Learning: Analysis and Performance Optimization". In: *IEEE Transactions on Mobile Computing* 21.9 (2022), pp. 3388–3401.

Justin Whitehouse et al. "Fully-Adaptive Composition in Differential Privacy". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 36990–37007.

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.

Hongyi Zhang and Suvrit Sra. "First-order Methods for Geodesically Convex Optimization". In: *29th Annual Conference on Learning Theory*. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 1617–1638.