Federated Learning on Riemannian Manifolds

Marco Sutti

Postdoctoral fellow at NCTS

NCTS Seminar on Scientific Computing

August 29th, 2022

Overview

Federated Learning on Riemannian Manifolds, Jiaxiang Li and Shiqian Ma, arXiv preprint, arXiv:2206.05668, June 12, 2022.

Contributions:

- Algorithms for Federated Learning (FL) with nonconvex constraints.
- New algorithm: RFedSVRG.
- Theoretical results on convergence.

This talk:

I. FL on Riemannian manifolds (RMs), federated kPCA and classical PCA.
II. Optimization on RMs, fundamental ideas and tools.
III. Algorithmic components of RFedSVRG.
IV. Numerical experiments on synthetic and real data.

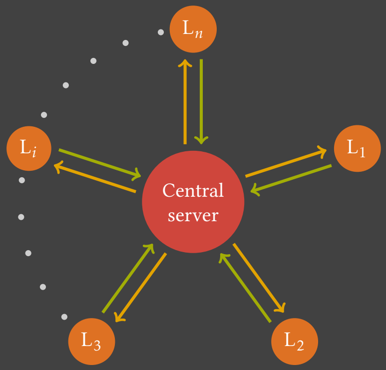
I. Introduction to Federated Learning

Federated learning (FL)

- Classical FL aims at solving the optimization problem

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

where each loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ is stored in a different local client/agent L_i that may have different physical locations and different hardware.
- A central server collects the information from the different agents and outputs a consensus that minimizes the sum of the loss functions $f_i(x)$ from all the clients.

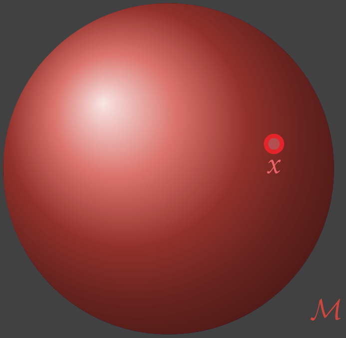


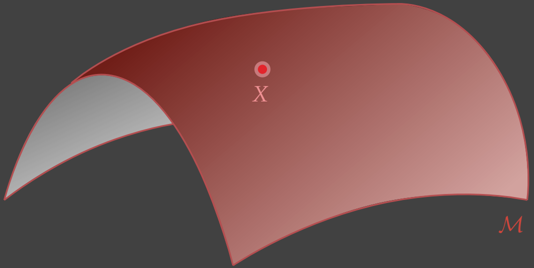
$\rightsquigarrow$ Aim of FL: use computational resources of different agents while maintaining the data privacy by not sharing data among all the local agents.

FL on Riemannian manifolds (RMs)

- FL problem over a Riemannian manifold

$$\min_{x\in\mathcal{M}} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad \text{where } f_i : \mathcal{M} \to \mathbb{R}.$$





Applications of FL on RMs

- Motivating application: federated kPCA problem, namely

$$\min_{X \in \mathrm{St}(d,r)} f(X) := \frac{1}{n}\sum_{i=1}^{n} f_i(X), \quad \text{where } f_i(X) = -\frac{1}{2}\,\mathrm{tr}(X^\top A_i X),$$

  where $\mathrm{St}(d,r) = \{X \in \mathbb{R}^{d\times r} | X^\top X = I_r\}$ is the Stiefel manifold, and $A_i = X_i X_i^\top$ is the covariance matrix of the data X_i stored in the ith local agent.

- When $r = 1$, we get the classical PCA, i.e.,

$$\min_{x \in \mathcal{S}^{d-1}} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad \text{where } f_i(x) = -\frac{1}{2}\,x^\top A_i x,$$

  where $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ is the unit $(d-1)$-sphere.

- Difficulty of existing algorithms: aggregating points over a nonconvex set.

Contributions of this paper

- Riemannian federated SVRG algorithm (RFedSVRG), with convergence rate $\mathcal{O}(1/\varepsilon^2)$ for obtaining an ε-stationary point.

  $\rightsquigarrow$ First algorithm for solving FL problems over RMs with convergence guarantees.

- Main novelty: consensus step on the tangent space to the manifold, instead of the widely used (so-called) “Karcher mean” approach (the Riemannian center of mass).

- Numerical results show that RFedSVRG outperforms the Riemannian counterparts of two widely used FL algorithms: FedAvg and FedProx.

FSVRG algorithm: [Konečný et al. 2016]
Do not call it “Karcher mean”!: [Karcher 2014]

II. Optimization on Riemannian manifolds

Riemannian manifold

A manifold $\mathcal{M}$ endowed with a smoothly-varying inner product (called Riemannian metric g) is called Riemannian manifold.

$\rightsquigarrow$ A couple $(\mathcal{M}, g)$, i.e., a manifold with a Riemannian metric on it.

- Matrix manifold: any manifold that is constructed from $\mathbb{R}^{n \times p}$ by taking either embedded submanifolds or quotient manifolds.
  - Examples of embedded submanifolds: orthogonal Stiefel manifold, manifold of symplectic matrices, manifold of fixed-rank matrices, …
  - Example of quotient manifold: the Grassmann manifold.

Manifold optimization: [Edelman et al. 1998, Absil et al. 2008, Boumal 2022], …

The Stiefel manifold and tangent space

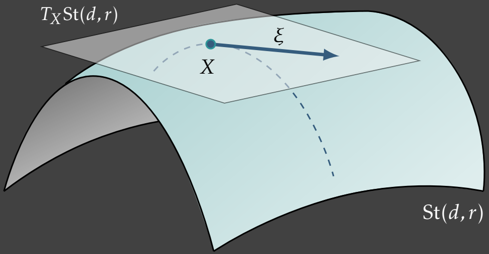
- Set of matrices with orthonormal columns:

$$\mathrm{St}(d, r) = \{X \in \mathbb{R}^{d\times r} : \; X^\top X = I_r\}.$$

- Tangent space to $\mathcal{M}$ at x: set of all tangent vectors to $\mathcal{M}$ at x, denoted $T_x\mathcal{M}$.

  $\rightsquigarrow$ For the Stiefel manifold $\mathrm{St}(d, r)$,

$$T_X\mathrm{St}(d, r) = \{\xi \in \mathbb{R}^{d\times r} : \; X^\top\xi + \xi^\top X = 0\}.$$



Stiefel manifold: [Stiefel, 1935]

Exponential and logarithm mapping

Given $x \in \mathcal{M}$ and $\xi \in T_x\mathcal{M}$, the exponential mapping $\mathrm{Exp}_x : T_x\mathcal{M} \to \mathcal{M}$ s.t. $\mathrm{Exp}_x(\xi) := \gamma(1)$, with γ being the geodesic with $\gamma(0) = x$, $\dot{\gamma}(0) = \xi$.

Corollary/Properties:

$$\mathrm{Exp}_x(t\xi) := \gamma(t), \quad t \in [0,1], \quad \text{and} \quad d(x, \mathrm{Exp}_x(\xi)) = \|\xi\|.$$

$\forall x,\ y \in \mathcal{M}$, the mapping $\mathrm{Exp}_x^{-1}(y) \in T_x\mathcal{M}$ is called the logarithm mapping.

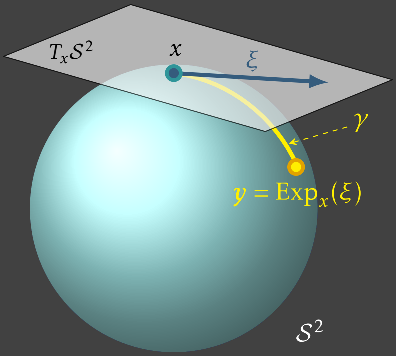
Example. Let $\mathcal{M} = \mathcal{S}^{n-1}$, then the exponential mapping at $x \in \mathcal{S}^{n-1}$ is

$$y = \mathrm{Exp}_x(\xi) = x\cos(\|\xi\|) + \frac{\xi}{\|\xi\|}\sin(\|\xi\|),$$

and the Riemannian logarithm is

$$\mathrm{Log}_x(y) = \xi = \arccos(x^\mathsf{T} y)\frac{\mathrm{P}_x y}{\|\mathrm{P}_x y\|},$$

where $y \equiv \gamma(1)$ and P_x is the projector onto $\big(\mathrm{span}(x)\big)^\perp$, i.e., $\mathrm{P}_x = I - xx^\mathsf{T}$.



Riemannian gradient

⇝ For any embedded submanifold:

- Riemannian gradient: projection onto $T_X\mathcal{M}$ of the Euclidean gradient

$$\operatorname{grad} f(X) = \mathrm{P}_{T_X\mathcal{M}}(\nabla f(X)).$$

⇝ For the Stiefel manifold, the projection onto the tangent space is

$$\mathrm{P}_{T_X\mathrm{St}(d,r)}\xi = X\operatorname{skew}(X^\top\xi) + (I - XX^\top)\xi.$$

⇝ $\nabla f(X)$ is the Euclidean gradient of $f(X)$.

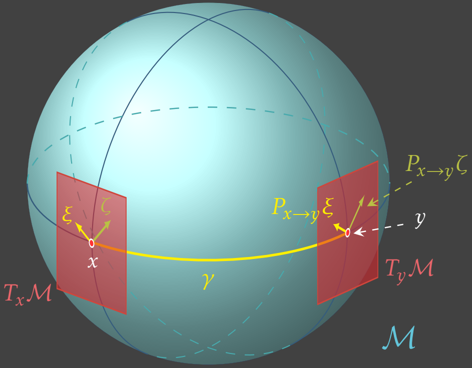
⇝ For example, if $f(X) = -\frac{1}{2}\operatorname{tr}(X^\top AX)$ (i.e., the local loss function in the kPCA problem), one has $\nabla f(X) = -AX$.

Symbolic matrix and vector calculus: The Matrix Cookbook, www.matrixcalculus.org, …

Parallel transport

- Parallel transport is used to define the Lipschitz condition for the Riemannian gradients and to prove convergence of the method.
- Given a RM $(\mathcal{M}, g)$ and two points $x, y \in \mathcal{M}$, the parallel transport $P_{x \to y} : T_x\mathcal{M} \to T_y\mathcal{M}$ is a linear operator that preserves the inner product:

$$\forall \xi, \zeta \in T_x\mathcal{M}, \qquad \langle P_{x\to y}\xi, P_{x\to y}\zeta \rangle_y = \langle \xi, \zeta \rangle_x.$$



III. The RFedSVRG algorithm

Illustration of the algorithm with 3 local agents

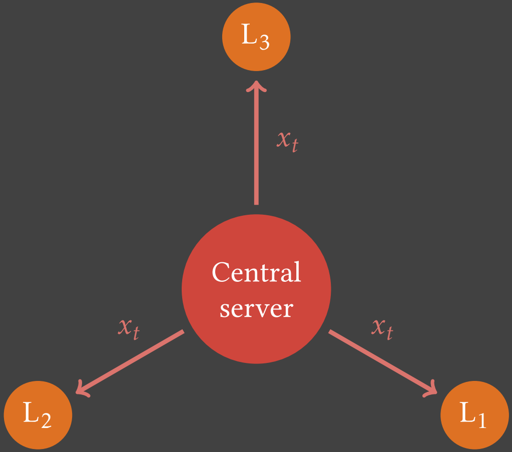


Illustration of the algorithm with 3 local agents

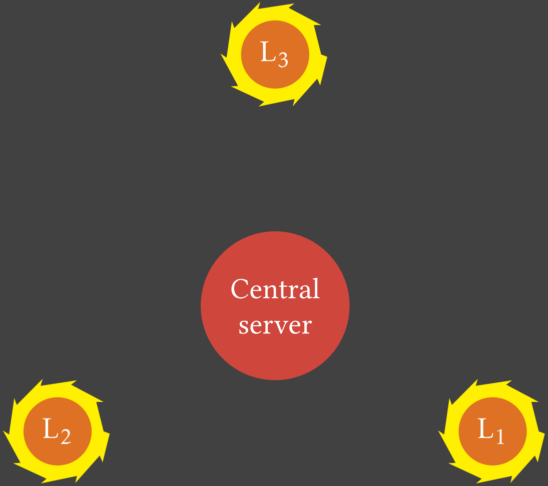


Illustration of the algorithm with 3 local agents

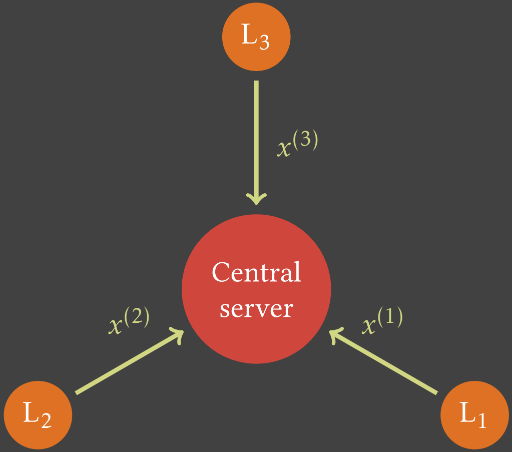


Illustration of the algorithm with 3 local agents

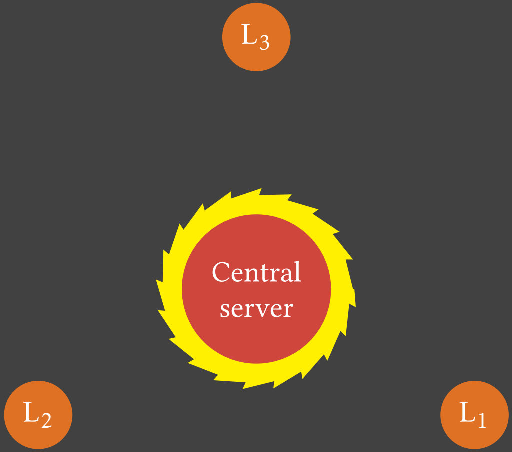
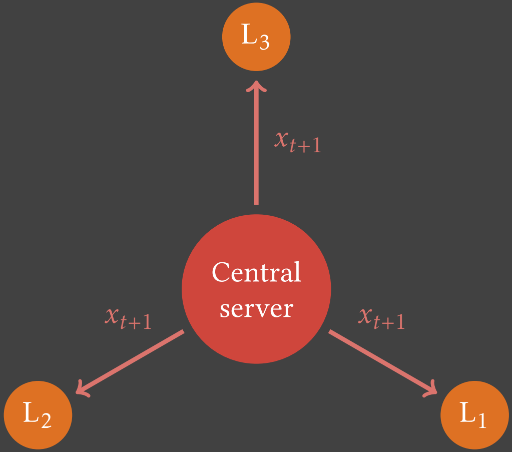


Illustration of the algorithm with 3 local agents



Aggregation on the central server/1

How to perform aggregation on the central server (: the consensus step)?

1. Riemannian center of mass of the points (the most common approach)

$$x_{t+1} \leftarrow \underset{x}{\operatorname{argmin}} \frac{1}{k} \sum_{i \in S_t} d^2(x, x^{(i)}).$$

   Here, $S_t \subset [n]$ is a subset of indices with cardinality $k = |S_t|$, $x^{(i)}$ is the data from each local server, $d(\cdot, \cdot)$ is the Riemannian distance, and x_{t+1} is the next iterate point on the central server.

2. Tangent space consensus step (the one used in this paper)

$$x_{t+1} \leftarrow \mathrm{Exp}_{x_t}\left(\frac{1}{k} \sum_{i \in S_t} \mathrm{Exp}_{x_t}^{-1}(x^{(i)})\right),$$

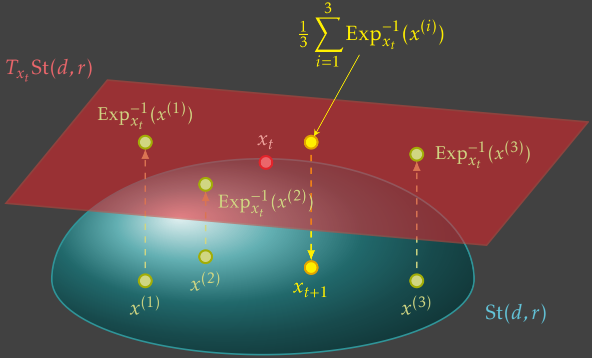
   where we "lift" each of the data points $x^{(i)}$ to the tangent space $T_{x_t}\mathcal{M}$, take their average on $T_{x_t}\mathcal{M}$, and finally map the average back to $\mathcal{M}$.

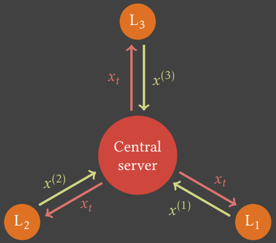
Aggregation on the central server/2

Recall the above formula for the tangent space consensus step:

$$x_{t+1} \leftarrow \operatorname{Exp}_{x_t}\left(\frac{1}{k}\sum_{i\in S_t}\operatorname{Exp}_{x_t}^{-1}(x^{(i)})\right).$$

Example with 3 local agents:





Local gradient update

Which calculations are performed on each client?

- Local gradient update

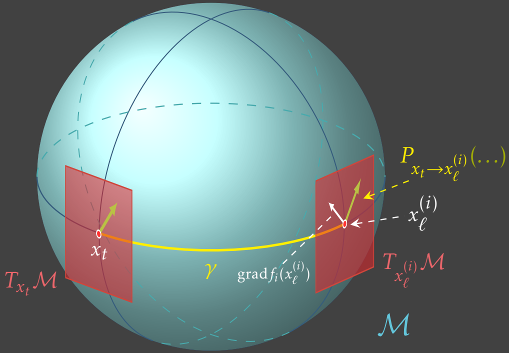
$$x_{\ell+1}^{(i)} \leftarrow \operatorname{Exp}_{x_\ell^{(i)}}\left[-\eta^{(i)}\left(\operatorname{grad} f_i(x_\ell^{(i)}) - P_{x_t \to x_\ell^{(i)}}(\operatorname{grad} f_i(x_t) - \operatorname{grad} f(x_t))\right)\right],$$

where $\eta^{(i)}$ is the stepsize.

- The parallel transport is used to bring the tangent vector

  $$(\operatorname{grad} f_i(x_t) - \operatorname{grad} f(x_t))$$

  on the same tangent space as that of $\operatorname{grad} f_i(x_\ell^{(i)})$, i.e., $T_{x_\ell^{(i)}}\mathcal{M}$, in order to perform addition and subtraction.



FSVRG algorithm: [Konečný et al. 2016]

RFedSVRG algorithm

RFedSVRG: manifold extension of the FSVRG algorithm.

Algorithm 1: Riemannian FedSVRG Algorithm (RFedSVRG)

input : n, k, T, $\{\eta^{(i)}\}$, $\{\tau_i\}$
output : **Option 1:** $\tilde{x} = x_T$; or **Option 2:** $\tilde{x}$ is uniformly sampled from $\{x_1, \ldots, x_T\}$

```
for t = 0, ..., T − 1 do
    Uniformly sample S_t ⊂ [n] with |S_t| = k;
    for each agent i in S_t do
        Receive x_0^(i) = x_t from the central server;
        for ℓ = 0, ..., τ_i − 1 do
            Take the local gradient step
        end
        Send x̂^(i) (obtained by one of the following options) to the central server
          • Option 1: x̂^(i) = x_{τ_i}^(i);
          • Option 2: x̂^(i) is uniformly sampled from {x_1^(i), ..., x_{τ_i}^(i)};
    end
    The central server aggregates the points by the tangent space mean
end
```

Local gradient step:

$$x_{\ell+1}^{(i)} \leftarrow \mathrm{Exp}_{x_\ell^{(i)}}\left[-\eta^{(i)}\left(\mathrm{grad}\, f_i(x_\ell^{(i)}) - P_{x_t \to x_\ell^{(i)}}(\mathrm{grad}\, f_i(x_t) - \mathrm{grad}\, f(x_t))\right)\right]$$

Tangent space mean:

$$x_{t+1} \leftarrow \mathrm{Exp}_{x_t}\left(\frac{1}{k}\sum_{i \in S_t} \mathrm{Exp}_{x_t}^{-1}(x^{(i)})\right)$$

Here, n is the total number of agents, k is the cardinality of S_t, T is the number of rounds, and τ_i in the inner loop denotes the number of local gradient steps.

Convergence of RFedSVRG

Use standard assumptions for optimization on manifolds:

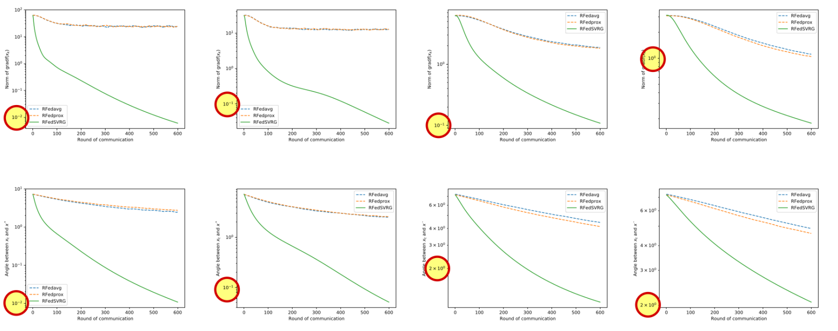
1. Lipschitz smoothness on manifolds: $f : \mathcal{M} \to \mathbb{R}$ is Lipschitz smooth on $\mathcal{M}$ if $\exists L \geqslant 0$ s.t.
$$\|\operatorname{grad} f(y) - P_{y \to x} \operatorname{grad} f(x)\| \leqslant L\, d(x, y).$$
2. The manifold is complete, and there exists a compact set $\mathcal{D} \subset \mathcal{M}$ such that all the iterates generated by the RFedSVRG algorithm are contained in $\mathcal{D}$.
3. The sectional curvature is bounded.
4. The objective function is geodesically convex.

$\rightsquigarrow$ Convergence rate results for $\tau_i = 1$ (Theorem 7), $\tau_i > 1$ (Theorem 8), and for a geodesically convex objective function (Theorem 9).

IV. Numerical experiments

Numerical experiments with synthetic data/1

- Compare RFedSVRG to the natural manifold extensions of two existing algorithms (FedProx and FedAvg). Results for kPCA.
- Operations on RMs: Manopt and PyManopt.
- Data: data matrix X_i, covariance matrix $A_i := X_i X_i^\top$. Test the algorithms with different number of agents $n = \{50, 100, 500, 1000\}$, $k = n/10$, and $(d, r) = (200, 5)$.
- Monitored quantities: $\|\text{grad}\, f(x_t)\|$ and the principal angle between x_t and x^*.

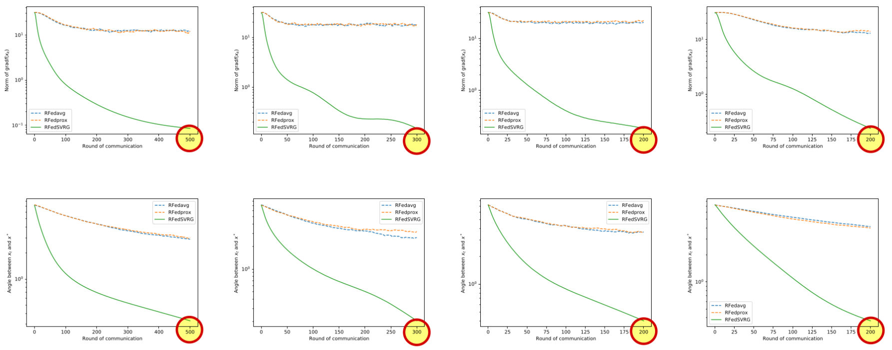


(a) $(n, k) = (50, 5)$ (b) $(n, k) = (100, 10)$ (c) $(n, k) = (500, 50)$ (d) $(n, k) = (1000, 100)$

FedAvg: [McMahan et al. 2017], FedProx algorithm: [Li et al. 2020], Manopt: [Boumal et al. 2014]

Numerical experiments with synthetic data/2

Experiments to test the effect of the number of local gradient steps τ.
Here, $n = 100$, $k = 10$, $(d, r) = (200, 5)$, and $\tau = \{1, 10, 50, 100\}$.



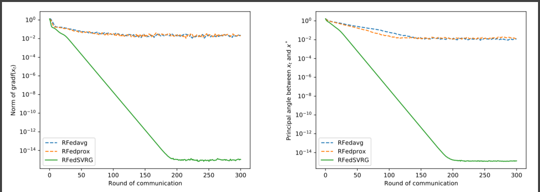
(a) $\tau = 1$ (b) $\tau = 10$ (c) $\tau = 50$ (d) $\tau = 100$

(My) observation. I am really surprised by such low accuracy (in absolute terms).

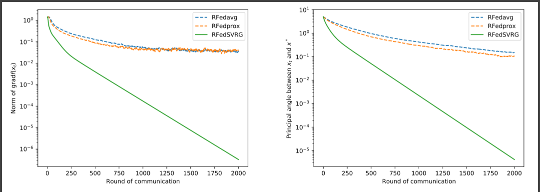
Numerical experiments with real data/1

- kPCA with the Iris and wine datasets. Randomly partition the datasets into $n = 10$ agents, and at each iteration take $k = 5$ agents.
- Numerical iterates are compared to the ground truth, given by the first r principal directions and the exact optimal loss value $f(x^*)$ computed directly.

Iris dataset



wine dataset

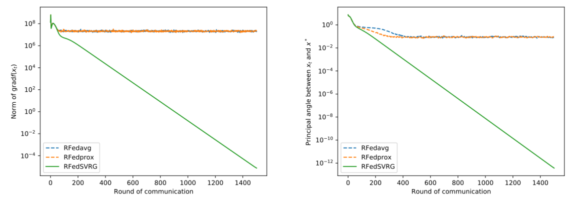


Iris and wine datasets: [Forina et al. 1998]

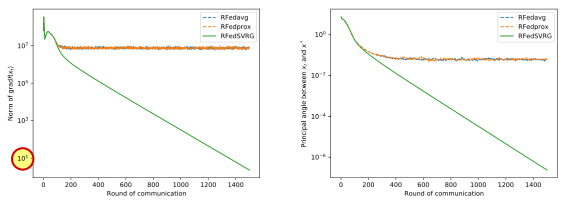
Numerical experiments with real data/2

- kPCA with the MNIST dataset.
- The (training) dataset contains 60 000 handwritten images of size 28×28, i.e., $d = 784$. Test RedFSVRG with $n = \{100, 200\}$.

$n = 100$



$n = 200$



MNIST dataset: [LeCun et al. 1998]

Conclusions

Contributions:

- A new effective algorithm for FL on RMs.
- Theoretical results on convergence.
- Numerical experiments on some common datasets.

Future research directions:

- Lower communication cost.
- Better scalability of the algorithm.
- Sparse solutions.

謝謝！

⇝ Download slides: https://www.marcosutti.net/research.html

V. Bonus material

Hopf–Rinow Theorem

Theorem ([Hopf/Rinow]) Let $(\mathcal{M}, g)$ be a (connected) Riemannian manifold. Then the following conditions are equivalent:

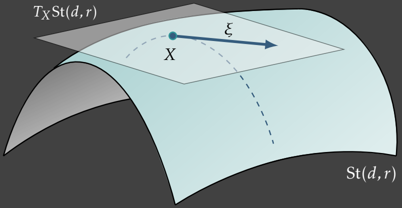
1. Closed and bounded subsets of $\mathcal{M}$ are compact;
2. $(\mathcal{M}, g)$ is a complete metric space;
3. $(\mathcal{M}, g)$ is geodesically complete, i.e., for any $x \in \mathcal{M}$, the exponential map Exp_x is defined on the entire tangent space $T_x\mathcal{M}$.

Any of the above implies that given any two points $x, y \in \mathcal{M}$, there exists a length-minimizing geodesic connecting these two points.

Stiefel manifold is compact/complete/geodesically complete $\rightsquigarrow$ length-minimizing geodesics exist.

Riemannian Geometry, Sakai '92

The Stiefel manifold/2



- Alternative characterization:

$$T_X \mathrm{St}(n,p) = \{X\Omega + X_\perp K : \Omega = -\Omega^\top,\ K \in \mathbb{R}^{(n-p)\times p}\}.$$

- Dimension: since $\dim\Big(\mathrm{St}(n,p)\Big) = \dim\Big(T_X \mathrm{St}(n,p)\Big)$, the dimension of the Stiefel manifold is

$$\dim(\mathrm{St}(n,p)) = \dim(\mathcal{S}_{\mathrm{skew}}) + \dim(\mathbb{R}^{(n-p)\times p}) = np - \frac{1}{2}p(p+1).$$