Riemannian Diffusion Adaptation for Distributed Optimization on Manifolds

Xiuheng Wang[†], Ricardo Borsoi[*], Cédric Richard[†], Ali Sayed[‡]

[*]Université de Lorraine, CNRS, CRAN, France
[†]Université Côte d'Azur, CNRS, OCA, France
[‡]École Polytechnique Fédérale de Lausanne, Switzerland

ICML 2025

Distributed optimization on manifolds

Multi-agent optimization problem seeking *consensus* on a Riemannian manifold:

$$\min_{w \in \mathcal{M}} \sum_{k=1}^{K} J_k(w), \quad J_k(w) = \mathbb{E}_{\boldsymbol{x}_k}\left\{Q(w; \boldsymbol{x}_k)\right\}. \tag{1}$$

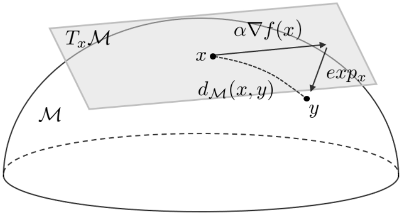
Riemannian manifold $\mathcal{M}$: curvature is induced by constraint, e.g., $\|\boldsymbol{w}\| = 1$ for the sphere, or metric, e.g., $\langle \boldsymbol{w}_1, \boldsymbol{w}_2 \rangle_{\boldsymbol{\Sigma}} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{w}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{w}_2)$ for the manifold of symmetric positive definite (SPD) matrices.

A wide range of applications can be written in the form of (1), including

- Principal component analysis (PCA);
- Gaussian mixture models (GMM);
- Low-rank matrix completion;
- Deep neural networks with orthogonal constraints.

This work focuses on fully intrinsic methods and thus can be applied to general manifolds.

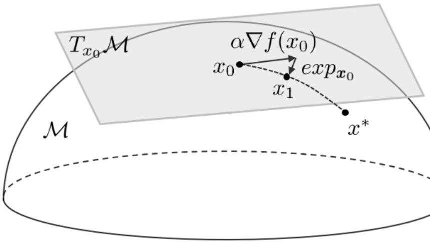
Riemannian optimization: main tools



A few important tools:

- Riemannian gradient: $\nabla f(x) \in T_x\mathcal{M}$;
- exponential mapping: $\exp_x : T_x\mathcal{M} \to \mathcal{M}$ (maps a vector in the tangent space back to the manifold);
- geodesic distance: $d_\mathcal{M}$ (length of the shortest path between two points on $\mathcal{M}$).

Riemannian optimization: R-SGD, basic structure



Considering a cost $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{M}$ we proceed as[1]:

- compute a stochastic approximation of $\nabla f(\boldsymbol{x})$ at $\boldsymbol{x}$;
- "take a step in the negative gradient direction" on $\mathcal{M}$ using the exponential mapping.

[1] Silvere Bonnabel. "Stochastic gradient descent on Riemannian manifolds". In: *IEEE Transactions on Automatic Control* 58.9 (2013), pp. 2217–2229.

Riemannian Diffusion adaptation

To encourage *consensus* ($\boldsymbol{w}_k = w, \forall k$) on manifolds, we consider the geodesic distance-based consensus problem[2], i.e., minimization of the penalty:

$$P(\boldsymbol{w}) \triangleq \sum_{k=1}^{K} P_k(\boldsymbol{w}_k), \quad \text{where} \quad P_k(\boldsymbol{w}_k) \triangleq \frac{1}{2} \sum_{\ell=1}^{K} c_{\ell k} d^2(\boldsymbol{w}_k, \boldsymbol{w}_\ell). \tag{2}$$

This results in the following optimization problem with a constraint:

$$\min_{\boldsymbol{w} \in \mathcal{M}^K} J(\boldsymbol{w}) \qquad s.t. \quad P(\boldsymbol{w}) = 0, \tag{3}$$

where $J(\boldsymbol{w}) \triangleq \frac{1}{K} \sum_{k=1}^{K} J_k(\boldsymbol{w}_k)$.

[2] Roberto Tron et al. "Riemannian consensus for manifolds with bounded curvature". In: *IEEE Transactions on Automatic Control* 58.4 (2012), pp. 921–934.

Riemannian Diffusion adaptation

We first apply an R-SGD to the risk $J(\boldsymbol{w})$ and subsequently descend along the penalty $P(\boldsymbol{w})$:

$$\phi_{k,t} = \exp_{\boldsymbol{w}_{k,t-1}}\left(-\mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,t-1})\right), \tag{4}$$

$$\boldsymbol{w}_{k,t} = \exp_{\phi_{k,t}}\left(-\alpha\nabla P_k(\phi_{k,t})\right) = \exp_{\phi_{k,t}}\left(\alpha\sum_{\ell=1}^{K} c_{\ell k}\exp^{-1}_{\phi_{k,t}}(\phi_{\ell,t})\right). \tag{5}$$

Adapt-then-combine scheme:

- An adaptation step: each agent k uses its own data $\boldsymbol{x}_{k,t-1}$ to update its solution $\phi_{k,t}$;
- A combination step: the intermediate estimates $\{\phi_{l,t}\}$ are combined, on the tangent space of $\phi_{k,t}$ to obtain the estimate $\boldsymbol{w}_{k,t}$.

Our algorithm reduces to the standard diffusion adaptation[3,4] when $\mathcal{M}$ is an Euclidean space.

[3] Jianshu Chen et al. "Diffusion adaptation strategies for distributed optimization and learning over networks". In: *IEEE Transactions on Signal Processing* 60.8 (2012), pp. 4289–4305.

[4] Ali H Sayed et al. "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior". In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 155–171.

Theoretical analysis

Rewrite (4) and (5) compactly as

$$\phi_t = \exp_{\boldsymbol{w}_{t-1}}\left(-\mu\widehat{\nabla J}(\boldsymbol{w}_{t-1})\right), \tag{6}$$

$$\boldsymbol{w}_t = \exp_{\phi_t}\left(-\alpha\nabla P(\phi_t)\right). \tag{7}$$

Step (7) can be regarded as a one-step Riemannian optimization to approximate a global minimum of $P(\phi)$, belonging to the *consensus submanifold* $\mathcal{A}$, defined as

$$\mathcal{A} \triangleq \{\phi \in \mathcal{M}^K \,|\, \phi_i = \phi_j, \forall i,j\}. \tag{8}$$

The local update in (4) is performed with a constant step size, which plays an important role in continuous learning and adaptation scenarios[5,6].

[5] Ali H Sayed et al. "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior". In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 155–171.

[6] Ali H Sayed. "Adaptive networks". In: *Proceedings of the IEEE* 102.4 (2014), pp. 460–497.

Evolution of the penalty:

Lemma

Under some mild assumptions including geodesic smoothness, suppose $\alpha \in (0, h_{max}^{-1}]$*. The sequence* $\{P(\phi_t)\}_{t\geq 0}$ *satisfies the following relation:*

$$\mathbb{E}\{P(\phi_{t+1}) - P(\phi_t)\} \leq -\frac{\alpha}{4}\mathbb{E}\|\nabla P(\phi_t)\|^2 + \frac{5\mu^2}{\alpha}G^2 + \frac{\mu^2}{\alpha}\sigma^2 . \tag{9}$$

Network Agreement

Approximately achieve consensus:

Theorem

Under some mild assumptions including geodesic convexity and smoothness, suppose $\alpha \in (0, h_{max}^{-1}]$. The sequence $\{P(\phi_t)\}_{t \geq 0}$ satisfies the following relation:

$$\mathbb{E}\{P(\phi_t)\} \leq \frac{11\mu^2}{2\alpha\tau} G^2 + \frac{3\mu^2}{\alpha\tau}\sigma^2 \,, \tag{10}$$

after sufficient iterations s_o, given by

$$s_o = \frac{2\log(\mu)}{\log(1-\tau)} + O(1) = O(\mu^{-1}) \,, \tag{11}$$

where $\tau = \min\{\frac{1}{2\zeta}, \alpha h_{min}\}$, the last equality holds for sufficiently small μ.

Approximately achieve consensus:

This result establishes that after sufficient iterations $s_o = O(\mu^{-1})$, we have:

$$\mathbb{E}\{P(\phi_t)\} \leq O(\mu^2)\,, \tag{12}$$

or, from Markov's inequality:

$$\Pr\{P(\phi_t) \geq \mu\} \leq O(\mu)\,, \tag{13}$$

which means the local estimates in ϕ_t coalesce around $\phi_t^* \in \mathcal{A}$ (where $P(\phi_t^*) = 0$) with high probability.

Evolution of the cost:

Lemma

Under some mild assumptions including geodesic smoothness, suppose $\mu \in (0, L^{-1}]$. *The sequence* $\{J(\boldsymbol{w}_t)\}_{t \geq 0}$ *satisfies the following relation:*

$$\mathbb{E}\{J(\boldsymbol{w}_{t+1}) - J(\boldsymbol{w}_t)\} \leq -\frac{\mu}{4}\mathbb{E}\|\widehat{\nabla J}(\boldsymbol{w}_t)\|^2 + \frac{5\alpha^2}{\mu}\mathbb{E}\|\nabla P(\phi_{t+1})\|^2 . \quad (14)$$

Design of a Lyapunov function:

To handle the manifold curvature, we design a Lyapunov function[7] as $\Delta'_t \triangleq J(\boldsymbol{w}'_t) - J(\boldsymbol{w}^*)$, we study the convergence of $\{\boldsymbol{w}_{s_o+1}, \cdots, \boldsymbol{w}_t\}$ by auxiliary variables $\{\boldsymbol{w}'_{s_o+1}, \cdots, \boldsymbol{w}'_t\}$:

- $\boldsymbol{w}'_{s_o+1} = \boldsymbol{w}_{s_o+1}$
- $\boldsymbol{w}'_{s+1} = \exp_{\boldsymbol{w}'_s}\left(\frac{1}{s-s_o+1}\exp^{-1}_{\boldsymbol{w}'_s}(\boldsymbol{w}_{s+1})\right)$ for $s_o + 1 \leq s \leq t-2$
- $\boldsymbol{w}'_t = \exp_{\boldsymbol{w}'_{t-1}}\left(\frac{2\zeta}{2\zeta+t-s_o-1}\exp^{-1}_{\boldsymbol{w}'_{t-1}}(\boldsymbol{w}_t)\right)$

For example, when $\mathcal{M} = \mathbb{R}^n$, the streaming average reduces to

- $\boldsymbol{w}'_{s_o+1} = \boldsymbol{w}_{s_o+1}$
- $\boldsymbol{w}'_{s+1} = \boldsymbol{w}'_s + \frac{1}{s-s_o-1}(\boldsymbol{w}'_s - \boldsymbol{w}_{s+1})$ for $s_o + 1 < s \leq t-2$
- $\boldsymbol{w}'_t = \boldsymbol{w}'_{t-1} + \frac{2\zeta}{2\zeta+t-s_o-1}(\boldsymbol{w}'_{t-1} - \boldsymbol{w}_t)$

[7] Hongyi Zhang et al. "First-order methods for geodesically convex optimization". In: *Conference on Learning Theory*. 2016, pp. 1617–1638.

Convergence

Non-asymptotic convergence:

Theorem

Under some mild assumptions including geodesic convexity and smoothness, suppose $\alpha \in (0, h_{max}^{-1}]$ *and* $\mu \in (0, L^{-1}]$. *The sequence* $\{J(\mathbf{w}'_t)\}_{t \geq s_o+1}$ *satisfies the following relation:*

$$\mathbb{E}\Delta'_t \leq \frac{\zeta L D^2 + (t - s_o)\left(\frac{231\zeta\alpha\mu}{2\tau} G^2 + \frac{63\zeta\alpha\mu}{\tau}\sigma^2\right)}{2\zeta + t - s_o - 1}, \tag{15}$$

where $\Delta'_t = J(\mathbf{w}'_t) - J(\mathbf{w}^*)$.

Applications and experiment setups

We apply our strategy to two manifolds as examples:

- PCA: the Grassmann manifold $\mathcal{G}_n^p$;
- GMM inference: the manifold of SPD matrices $\mathcal{S}_n^{++}$.

Baselines:

- Distributed PCA: DRSGD[8];
- Distributed GMM inference: ECGMM[9,10].

[8] Shixiang Chen et al. "Decentralized Riemannian gradient descent on the Stiefel manifold". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 1594–1605.

[9] Angelia Nedic et al. "Constrained consensus and optimization in multi-agent networks". In: *IEEE Transactions on Automatic Control* 55.4 (2010), pp. 922–938.

[10] Xiangru Lian et al. "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent". In: *Advances in Neural Information Processing Systems* 30 (2017).

Multi-agent system

We selected $K = 20$ agents, the weights in matrix $\boldsymbol{C}$ with Metropolis rule[11].

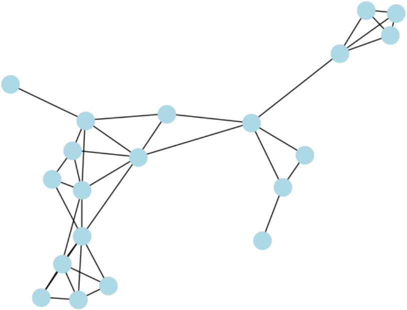


Figure: Graph topology.

[11] Lin Xiao et al. "A space-time diffusion scheme for peer-to-peer least-squares estimation". In: *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*. 2006, pp. 168–176.

Distributed PCA

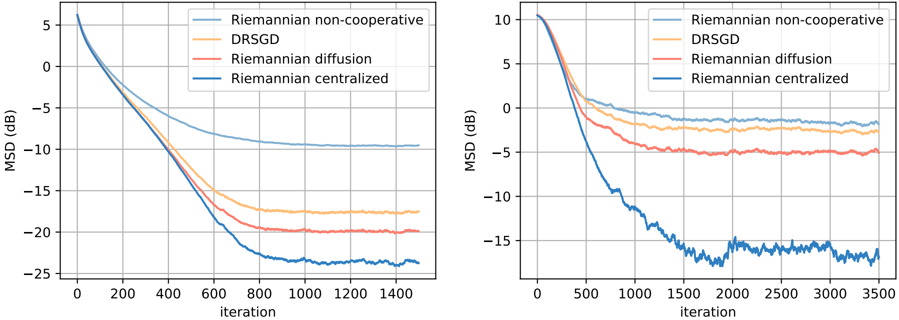


Figure: Illustration of MSD performance of the algorithms for distributed PCA on synthetic (left) and real (right) data.

Distributed GMM inference

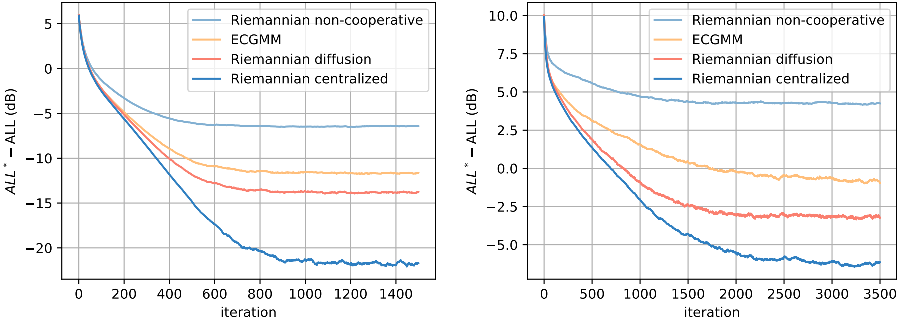


Figure: Illustration of ALL differences of the algorithms for distributed GMM inference on synthetic (left) and real (right) data.

Thanks for your attention!