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:

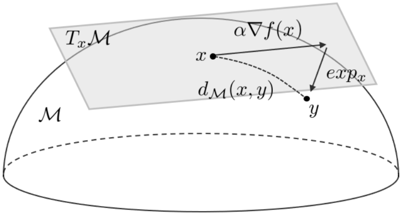
$$\min_{w \in \mathcal{M}} \sum_{k=1}^{K} J_k(w), \quad J_k(w) = \mathbb{E}_{\boldsymbol{x}_k}\left\{Q(w; \boldsymbol{x}_k)\right\}. \tag{1}$$

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.

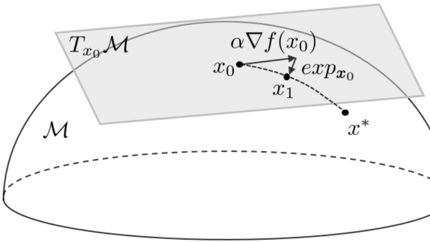
Riemannian optimization: main tools



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

One can encourage *consensus* ($\boldsymbol{w}_k = w, \forall k$) on manifolds with a penalty:

$$\min_{\boldsymbol{w} \in \mathcal{M}^K} J(\boldsymbol{w}) \qquad s.t. \quad P(\boldsymbol{w}) = 0, \tag{2}$$

where $J(\boldsymbol{w}) \triangleq \frac{1}{K}\sum_{k=1}^{K} J_k(\boldsymbol{w}_k)$ and $P(\boldsymbol{w}) \triangleq \frac{1}{2}\sum_{k=1}^{K}\sum_{\ell=1}^{K} c_{\ell k} d^2(\boldsymbol{w}_k, \boldsymbol{w}_\ell)$.

The proposed algorithm:

$$\phi_{k,t} = \exp_{\boldsymbol{w}_{k,t-1}}\left(-\mu \widehat{\nabla J}_k(\boldsymbol{w}_{k,t-1})\right), \tag{3}$$

$$\boldsymbol{w}_{k,t} = \exp_{\phi_{k,t}}\left(\alpha \sum_{\ell=1}^{K} c_{\ell k} \exp^{-1}_{\phi_{k,t}}(\phi_{\ell,t})\right). \tag{4}$$

A special case: our algorithm reduces to the standard diffusion adaptation[2,3] when $\mathcal{M} = \mathbb{R}^n$.

[2] 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.

[3] 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.

Network Agreement

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:*

$$\mathbb{E}\{P(\phi_t)\} \leq \frac{11\mu^2}{2\alpha\tau} G^2 + \frac{3\mu^2}{\alpha\tau}\sigma^2 \,, \tag{5}$$

after sufficient iterations s_o*, given by*

$$s_o = \frac{2\log(\mu)}{\log(1-\tau)} + O(1) = O(\mu^{-1}) \,, \tag{6}$$

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

Non-asymptotic convergence

To handle the manifold curvature, we design a Lyapunov function[4] of $\boldsymbol{w}_t$ 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 $s_o + 1 < s \leq t-2$
- $\boldsymbol{w}'_t = \boldsymbol{w}'_{t-1} + \frac{2\zeta}{2\zeta+t-s_o-1}(\boldsymbol{w}'_{t-1} - \boldsymbol{w}_t)$

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

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:*

$$\mathbb{E}\Delta'_t \leq \frac{\zeta L D^2 + (t - s_o)\left(\frac{231\zeta\alpha\mu}{2\tau}G^2 + \frac{63\zeta\alpha\mu}{\tau}\sigma^2\right)}{2\zeta + t - s_o - 1}, \quad (7)$$

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[5];
- Distributed GMM inference: ECGMM[6,7].

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

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

[7] 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).

We selected $K = 20$ agents, the weights in matrix $\boldsymbol{A}$ with Metropolis rule[8].

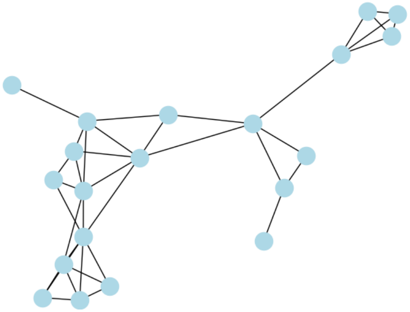


Figure: Graph topology.

[8] 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

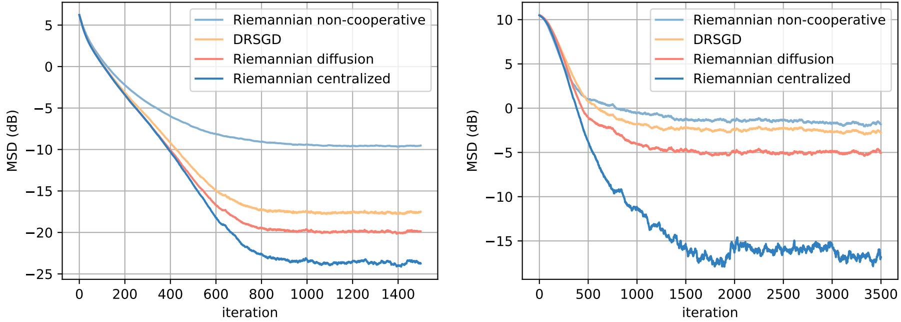


Figure: Illustration of MSD performance of the algorithms for distributed PCA on synthetic (left) and real (right) data.

Distributed GMM inference

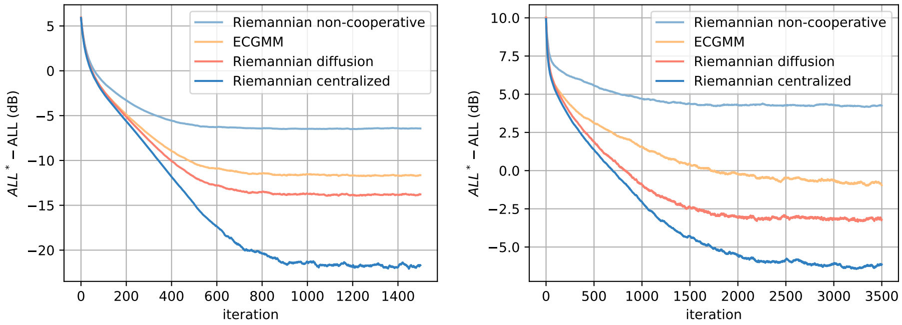


Figure: Illustration of ALL differences of the algorithms for distributed GMM inference on synthetic (left) and real (right) data.

Thanks for watching!