Riemannian adaptive stochastic gradient algorithms on matrix manifolds

Hiroyuki Kasai (The University of Electro-Communications, Japan), Pratik Jawanpuria (Microsoft, India) and Bamdev Mishra (Microsoft, India)

Problem of interest

- Consider the problem [1] of
  $$\min_{x\in\mathcal{M}} f(x). \quad \mathcal{M}: \text{Riemannian manifold}$$
  - $\mathcal{M}$ are represented as matrices of size $n \times r$.
  - Promising applications include, e.g., matrix/tensor completion, subspace tracking.

Contributions

- Propose a modeling adaptive weight matrices for row and column subspaces exploiting the geometry of manifold.
- Develop efficient Riemannian adaptive stochastic gradient algorithms (RASA).
- Achieve a rate of convergence order $O(\log(T)/\sqrt{T})$ for non-convex stochastic optimization under mild conditions.
- Show efficiency of RASA from numerical experiments on several applications.

Preliminaries

- Riemannian stochastic gradient update:
  $$\text{(RSGD)} \quad x_{t+1} = \underbrace{R_{x_t}}_{\text{retraction}}(-\alpha_t \underbrace{\mathrm{grad} f_t(x_t)}_{\substack{\text{Riemannian}\\ \text{stochastic gradient}}}),$$
  - $R_x(\zeta)$ maps $\zeta \in T_x\mathcal{M}$ (tangent space) onto $\mathcal{M}$.
  - When $\mathcal{M} = \mathbb{R}^d$ with standard Euclidean inner product, RSGD update results in
    $$\text{(SGD)} \quad x_{t+1} = x_t - \alpha_t \nabla f_t(x_t).$$
- Euclidean adaptive stochastic gradient updates:
  - Rescale the learning rate based on past gradients as
    $$x_{t+1} = x_t - \alpha_t \mathbf{V}_t^{-1/2} \nabla f_t(x_t).$$
  - $\mathbf{V}_t = \mathrm{Diag}(\mathbf{v}_t)$ is a diagonal matrix such as
    $$\text{(AgaGrad)} \quad \mathbf{v}_t = \sum_{k=1}^{t} \nabla f_k(x_k) \circ \nabla f_k(x_k),$$
    $$\text{(RMSProp)} \quad \mathbf{v}_t = \beta \mathbf{v}_{t-1} + (1-\beta)\nabla f_t(x_t) \circ \nabla f_t(x_t).$$

MATLAB source code

The code, which is compliant to Manopt (`https://www.manopt.org/`), is available at `https://github.com/hiroyuki-kasai/RSOpt/`.

RASA: Riemannian Adaptive Stochastic gradient Algorithms

Exploit matrix structure of Riemannian gradient $\mathbf{G}_t$ ($= \mathrm{grad} f_t(x_t) \in \mathbb{R}^{n\times r}$) by separating adaptive weight matrices corresponding to row subspace $\mathbf{L}_t$ and column subspaces $\mathbf{R}_t$.

c.f. [2] views $\mathbf{G}_t$ as a vector in $\mathbb{R}^{nr}$.

- Exponentially weighted matrices:
  $$\mathbf{L}_t = \beta \mathbf{L}_{t-1} + (1-\beta)\mathbf{G}_t\mathbf{G}_t^\top / r, \quad (\in \mathbb{R}^{n\times n})$$
  $$\mathbf{R}_t = \beta \mathbf{R}_{t-1} + (1-\beta)\mathbf{G}_t^\top\mathbf{G}_t / n. \quad (\in \mathbb{R}^{r\times r})$$
  ($\beta \in (0,1)$: hyper-parameter)
- Adaptive Riemannian gradient $\tilde{\mathbf{G}}_t$:
  $$\tilde{\mathbf{G}}_t = \mathbf{L}_t^{-1/4}\mathbf{G}_t\mathbf{R}_t^{-1/4}.$$
- Full-matrix update:
  $$x_{t+1} = R_{x_t}(-\alpha_t \mathcal{P}_{x_t}(\tilde{\mathbf{G}}_t)).$$
  - $\mathcal{P}_x$, a linear operator, projects onto tangent space $T_x\mathcal{M}$.
- Diagonal modeling of $\{\mathbf{L}_t, \mathbf{R}_t\}$ as vectors $\{\mathbf{l}_t, \mathbf{r}_t\}$:
  $$\mathbf{l}_t = \beta \mathbf{l}_{t-1} + (1-\beta)\mathrm{diag}(\mathbf{G}_t\mathbf{G}_t^\top), \quad (\in \mathbb{R}^n)$$
  $$\mathbf{r}_t = \beta \mathbf{r}_{t-1} + (1-\beta)\mathrm{diag}(\mathbf{G}_t^\top\mathbf{G}_t). \quad (\in \mathbb{R}^r)$$
  - $\mathrm{diag}(\cdot)$ returns diagonal vector of a square matrix.
- Maximum operator for convergence:
  $$\hat{\mathbf{l}}_t = \max(\hat{\mathbf{l}}_{t-1}, \mathbf{l}_t), \quad \hat{\mathbf{r}}_t = \max(\hat{\mathbf{r}}_{t-1}, \mathbf{r}_t).$$

Alg.1: RASA

Require: Step size $\{\alpha_t\}_{t=1}^T$, hyper-parameter β.
1: Initialize $x_1 \in \mathcal{M}, \mathbf{l}_0 = \hat{\mathbf{l}}_0 = \mathbf{0}_n, \mathbf{r}_0 = \hat{\mathbf{r}}_0 = \mathbf{0}_r$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Compute Riemannian stochastic gradient $\mathbf{G}_t = \mathrm{grad} f_t(x_t)$.
4: Update $\mathbf{l}_t = \beta\mathbf{l}_{t-1} + (1-\beta)\mathrm{diag}(\mathbf{G}_t\mathbf{G}_t^T)/r$.
5: Calculate $\hat{\mathbf{l}}_t = \max(\hat{\mathbf{l}}_{t-1}, \mathbf{l}_t)$.
6: Update $\mathbf{r}_t = \beta\mathbf{r}_{t-1} + (1-\beta)\mathrm{diag}(\mathbf{G}_t^T\mathbf{G}_t)/n$.
7: Calculate $\hat{\mathbf{r}}_t = \max(\hat{\mathbf{r}}_{t-1}, \mathbf{r}_t)$.
8: $x_{t+1} = R_{x_t}(-\alpha_t\mathcal{P}_{x_t}(\mathrm{Diag}(\hat{\mathbf{l}}_t^{-1/4})\mathbf{G}_t\mathrm{Diag}(\hat{\mathbf{r}}_t^{-1/4})))$.
9: **end for**

- RASA variants:
  - RASA-L adapts only the row subspace.
  - RASA-R adapts only the column subspace.
  - RASA-LR adapts both the row and column subspaces.

Convergence rate analysis

Extend existing convergence analysis in Euclidean space, e.g., [3], into Riemannian setting. Additionally, need to take care of (i) upper bound of $\hat{\mathbf{v}}_t$ (Lem.4.3) for update, and (ii) projection $\mathcal{P}_x$ of weighted gradient onto $T_x\mathcal{M}$.

- For analysis, we use additional notations as
  - $x_{t+1} = R_{x_t}(-\alpha_t\mathcal{P}_{x_t}(\mathbf{V}_t^{-1/2}\mathbf{g}_t(x_t)))$ for step 8 in **Alg.1**,
  - $\hat{\mathbf{V}}_t = \mathrm{Diag}(\hat{\mathbf{v}}_t)$, where $\hat{\mathbf{v}}_t = \hat{\mathbf{r}}_t^{1/2} \otimes \hat{\mathbf{l}}_t^{1/2}$, and
  - $\mathbf{g}_t(x)$ as the vectorized representation of $\mathrm{grad} f_t(x)$.
- Definition, assumptions, and lemma:

Def.4.1. (Upper-Hessian bounded) There exists a constant $L > 0$ such that $\frac{d^2 f(R_x(t\eta))}{dt^2} \leq L$, for $x \in \mathcal{U} \subset \mathcal{M}$ and $\eta \in T_x\mathcal{M}$ with $\|\eta\|_x = 1$, and all t such that $R_x(\tau\eta) \in \mathcal{U}$ for $\tau \in [0, t]$.

Asm.1.1. f is continuously differentiable and is lower bounded, *i.e.*, $f(x^*) > -\infty$.

Asm.1.2. f has H-bounded Riemannian stochastic gradient, i.e., $\|\mathrm{grad} f_i(x)\|_F \leq H$ or $\|\mathbf{g}_i(x)\|_2 \leq H$.

Asm.1.3. f is upper-Hessian bounded (Def.4.1).

Lem.4.2. Under Asm.1 and $L > 0$ in Def.4.1, we have $f(z) \leq f(x) + \langle \mathrm{grad} f(x), \xi \rangle_2 + \frac{1}{2}L\|\xi\|_2^2$, for $x \in \mathcal{M}$, where $\xi \in T_x\mathcal{M}$ and $R_x(\xi) = z$.

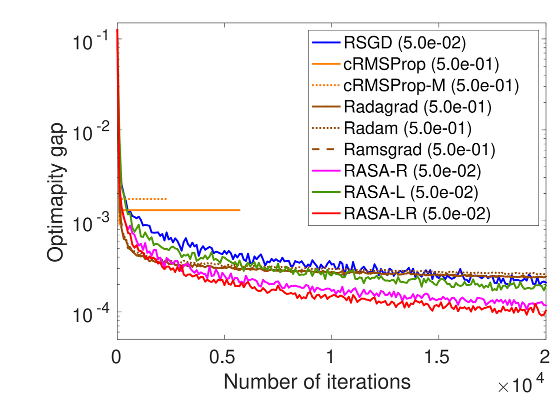
- Obtained results:

Thm.4.4. Let $\{x_t\}$ and $\{\hat{\mathbf{v}}_t\}$ be the sequences from **Alg.1**. Then, under Asm.1, we have
$$\mathbb{E}\left[\sum_{t=2}^{T}\alpha_{t-1}\left\langle \mathbf{g}(x_t), \frac{\mathbf{g}(x_t)}{\sqrt{\hat{\mathbf{v}}_{t-1}}}\right\rangle_2\right] \leq C +$$
$$\leq \mathbb{E}\left[\frac{L}{2}\sum_{t=1}^{T}\left\|\frac{\alpha_t \mathbf{g}_t(x_t)}{\sqrt{\hat{\mathbf{v}}_t}}\right\|_2^2 + H^2\sum_{t=2}^{T}\left\|\frac{\alpha_t}{\sqrt{\hat{\mathbf{v}}_t}} - \frac{\alpha_{t-1}}{\sqrt{\hat{\mathbf{v}}_{t-1}}}\right\|_1\right]$$
where C is a constant term independent of T.

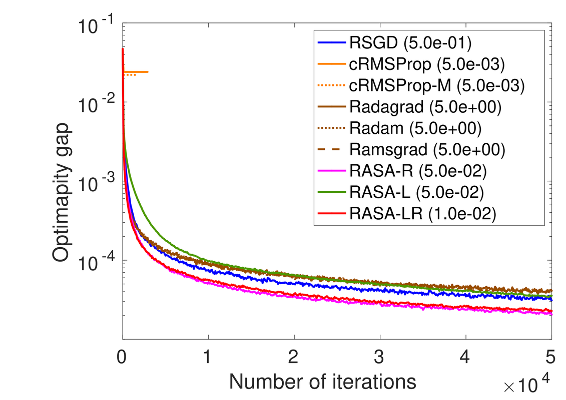
Cor.4.5. Let $\alpha_t = 1/\sqrt{t}$ and $\min_{j\in[d]}\sqrt{(\hat{\mathbf{v}}_1)_j}$ is lower-bounded by a constant $c > 0$, where d is the dimension of $\mathcal{M}$. Then, under Asm.1, the output of x_t of **Alg.1** satisfies
$$\min_{t\in[2,\ldots,T]} \mathbb{E}\|\mathrm{grad} f(x_t)\|_F^2 \leq \frac{1}{\sqrt{T-1}}(Q_1 + Q_2\log(T)),$$
where $Q_2 = LH^3/2c^2$ and
$$Q_1 = Q_2 + \frac{2dH^3}{c} + H\mathbb{E}[f(x_1) - f(x^*)].$$

Numerical evaluations

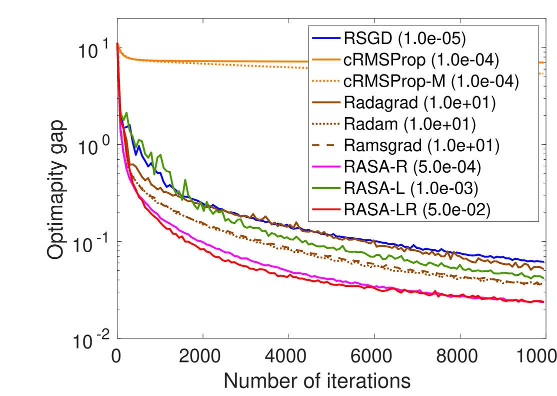
- PCA problem



(a) Case P1: Synthetic dataset.

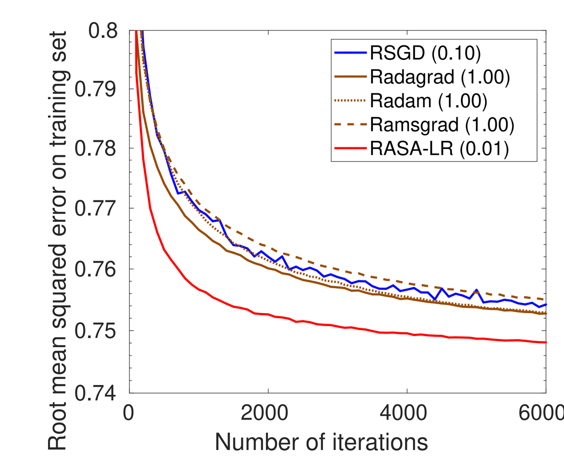


(b) Case P2: MNIST dataset.

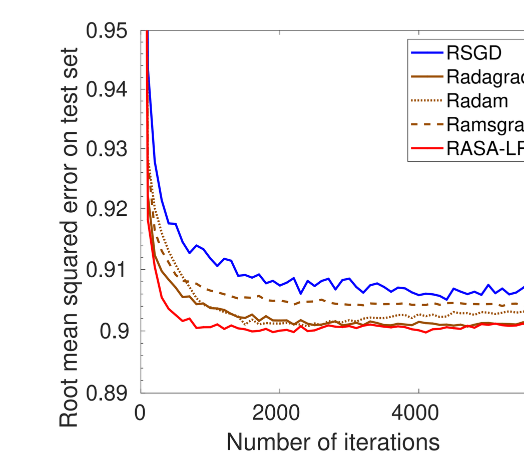


(c) Case P3: COIL100 dataset.

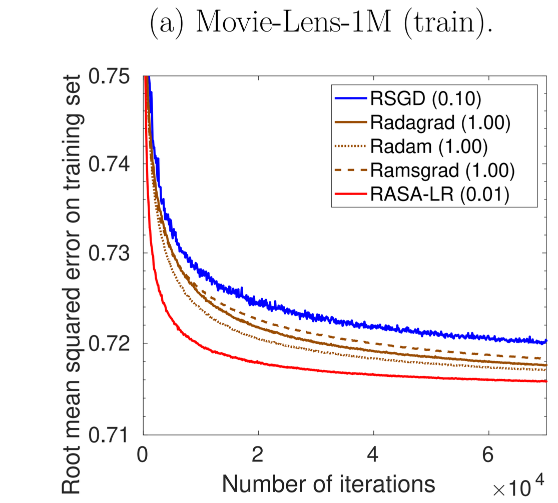
- Matrix completion problem



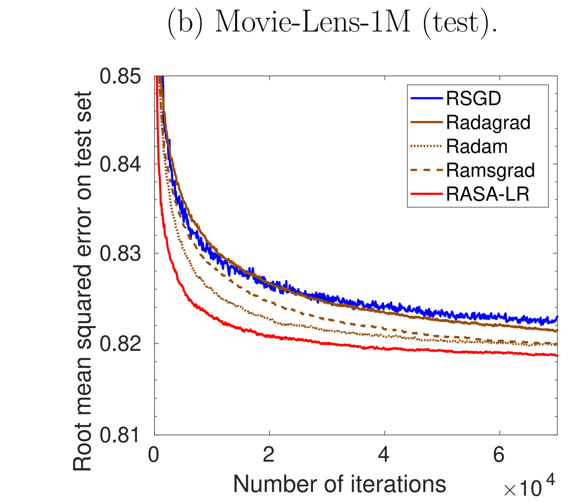
(a) Movie-Lens-1M (train).



(b) Movie-Lens-1M (test).

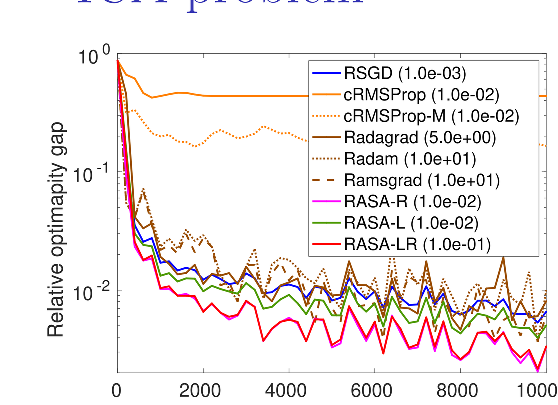


(c) Movie-Lens-10M (train).

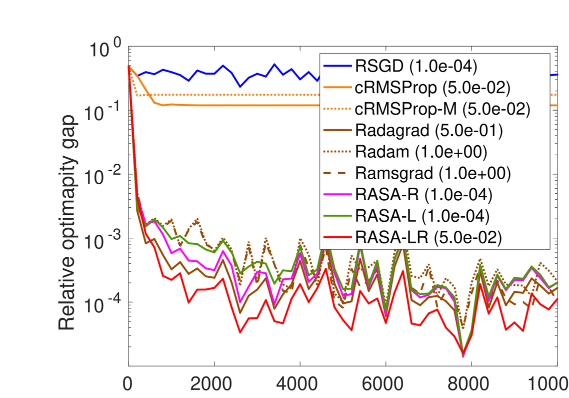


(d) Movie-Lens-10M (test).

- ICA problem



(a) Case I1: YaleB dataset.



(b) Case I2: COIL100 dataset.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, Optimization Algorithms on Matrix Manifolds. Princeton University Press, 2008.
[2] S.K. Roy, Z. Mhammedi, and M. Harandi, Geometry aware constrained optimization techniques for deep learning, CVPR, 2018.
[3] X. Chen, S. Liu, R. Sun, and M. Hong, On the convergence of a class of Adam-type algorithms for non-convex optimization, ICLR, 2019.