SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2020, Vol. 63 152201:1–152201:11
https://doi.org/10.1007/s11432-019-9911-5

Application of gradient descent algorithms based on geodesic distances

Xiaomin DUAN[1], Huafei SUN[2*] & Linyu PENG[3]

[1]*School of Science, Dalian Jiaotong University, Dalian* 116028*, China;*
[2]*School of Mathematics and Statistics, Beijing Institute of Technology, Beijing* 100081*, China;*
[3]*Waseda Institute for Advanced Study, Waseda University, Tokyo* 169-8050*, Japan*

Received 20 April 2019/Revised 20 May 2019/Accepted 30 May 2019/Published online 26 March 2020

Abstract In this paper, the Riemannian gradient algorithm and the natural gradient algorithm are applied to solve descent direction problems on the manifold of positive definite Hermitian matrices, where the geodesic distance is considered as the objective function. The first proposed problem is the control for positive definite Hermitian matrix systems whose outputs only depend on their inputs. The geodesic distance is adopted as the difference of the output matrix and the target matrix. The controller to adjust the input is obtained such that the output matrix is as close as possible to the target matrix. We show the trajectory of the control input on the manifold using the Riemannian gradient algorithm. The second application is to compute the Karcher mean of a finite set of given Toeplitz positive definite Hermitian matrices, which is defined as the minimizer of the sum of geodesic distances. To obtain more efficient iterative algorithm than traditional ones, a natural gradient algorithm is proposed to compute the Karcher mean. Illustrative simulations are provided to show the computational behavior of the proposed algorithms.

Keywords system control, Riemannian gradient algorithm, natural gradient algorithm, Karcher mean, Toeplitz positive definite Hermitian matrix

Citation Duan X M, Sun H F, Peng L Y. Application of gradient descent algorithms based on geodesic distances. Sci China Inf Sci, 2020, 63(5): 152201, https://doi.org/10.1007/s11432-019-9911-5

1 Introduction

Gradient adaptation is often used to minimize an objective function by adjusting parameters. Although it is easy to carry out, the adaptive gradient algorithm has some shortcomings. For example, when the slope of the objective function varies greatly for a small change of the parameters, convergence rate of the algorithm can be slow. To overcome the weakness of slow convergence, the natural gradient algorithm based on the geometric structure of the statistical manifolds was developed by Amari et al. [1]. They gave the steepest descent direction in Riemannian spaces and also shown that the natural gradient is asymptotically Fisher-efficient about the maximum likelihood estimation. It means that its properties are exactly like the optimal batch estimation of the parameters. Up to now, the natural gradient algorithm has been widely applied into, e.g., neural networks and optimal control, offering a new way to solve such problems more effectively [2–6].

Although the natural gradient algorithm defines the steepest descent direction, the iterative trajectory of the parameters is not necessarily the shortest, let alone the difficulty of computing the inverse of the metric. These problems are solved by using the Riemannian gradient algorithm in particular for matrix manifolds, separately introduced by Barbaresco [7] and Lenglet et al. [8] with wide applications,

* Corresponding author (email: huafeisun@bit.edu.cn)

e.g., [9, 10]. It is realized that the iterative path of each parameter is along its geodesic. However, to satisfy the given accuracy, the descent speed of the algorithm is not the fastest in some cases and the scope of the algorithm is sometimes limited.

In this paper, the set of $n \times n$ positive definite Hermitian matrices is defined as a manifold $P(n)$, whose geodesic connecting two matrices was studied in Moakher [11]. Both the Riemannian gradient algorithm and the natural gradient algorithm are applied to solve the descent direction problems. Because the geodesic distance is the length of the shortest curve connecting two matrices on a manifold, we take the geodesic distance as an objective function for two algorithms mentioned above. The first problem is the control of positive definite Hermitian matrix systems on manifold $P(n)$ using different gradient algorithms. Supposing the output of the system depends only on the control input, we take the geodesic distance as the measure of the output matrix and the target matrix. Controller to adjust the control input is shown that the difference between the output matrix and the target matrix is as small as possible. Trajectory of the control input is also obtained. Second, both gradient algorithms are used to compute the Karcher mean of a finite set of given Toeplitz positive definite Hermitian matrices, taking the sum of geodesic distances as the objective function. The simulations illustrate that the natural gradient algorithm converges faster than the Riemannian gradient algorithm.

2 Riemannian metric and geodesic on manifold $P(n)$

Let $M(n, \mathbb{C})$ be the set of $n \times n$ complex matrices and $\mathrm{GL}(n, \mathbb{C})$ be its subset containing only non-singular matrices. It is well known that $\mathrm{GL}(n, \mathbb{C})$ is a Lie group, whose Lie algebra is denoted by $\mathrm{gL}(n, \mathbb{C})$. The Frobenius inner product on $M(n, \mathbb{C})$, which is the Euclidean inner product, is represented as

$$\langle A, B \rangle = \mathrm{tr}\left(A^{\mathrm{H}} B\right), \tag{1}$$

where tr denotes the trace and A^{H} stands for the conjugate transport of matrix A. The Frobenius norm is defined as

$$\|A\| = \langle A, A \rangle^{\frac{1}{2}}. \tag{2}$$

Moreover, $M(n, \mathbb{C})$ endowed the inner product defined above is a flat manifold.

It is a fact that the set $P(n)$ is an n^2-dimensional manifold. We use $H(n)$ to represent the space of all $n \times n$ Hermitian matrices, and then the exponential map from $H(n)$ to $P(n)$ is one-to-one and onto. Because $P(n)$ is an open subset of $H(n)$, we identify the set $T_A P(n)$ of tangent vectors to $H(n)$ at A, for each $A \in P(n)$.

Moreover, we have the Riemannian metric on $P(n)$ as follows:

$$\langle X, Y \rangle_A = \langle A^{-1} X, A^{-1} Y \rangle_I = \mathrm{tr}\left(A^{-1} X A^{-1} Y\right), \tag{3}$$

where $X, Y \in T_A P(n)$ and I represents the identity element of $P(n)$. The metric (3) is positive definite, which can be derived from the positive definiteness of the Frobenius inner product.

Let $\gamma : [0, 1] \to P(n)$ be a sufficiently smooth curve on manifold $P(n)$; then the length of $\gamma(t)$ is defined by

$$\ell(\gamma) := \int_0^1 \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}}\, \mathrm{d}t = \int_0^1 \sqrt{\mathrm{tr}\left(\gamma^{-1}(t)\dot{\gamma}(t)\right)^2}\, \mathrm{d}t. \tag{4}$$

Also, the minimal length of all curves connecting two matrices A and B on manifold $P(n)$ is called geodesic distance:

$$d(A, B) := \inf\left\{\ell(\gamma) \mid \gamma : [0, 1] \to P(n) \ \text{ with } \gamma(0) = A, \gamma(1) = B\right\}. \tag{5}$$

It can be seen that length-minimizing smooth curves are geodesics which minimize (5). The Hopf-Rinow theorem [12] shows that manifold $P(n)$ is geodesically complete. This means that the interval $[0, 1]$ can be extended to $(-\infty, +\infty)$. Thus, a geodesic $\gamma(t)$ can be found for any given pair A, B such that $\gamma(0) = A$ and $\gamma(1) = B$, where the initial velocity of $\gamma(t)$ is taken as $\dot{\gamma}(0) = A^{\frac{1}{2}} \log(A^{-\frac{1}{2}} B A^{-\frac{1}{2}}) A^{\frac{1}{2}}$.

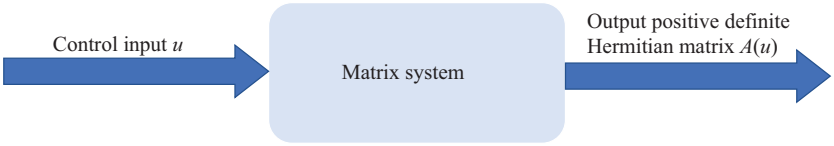

Figure 1 (Color online) Positive definite Hermitian matrix system.

Under congruent transformation $\gamma(t) \mapsto C\gamma(t)C^{\mathrm{H}}$ for any $C \in \mathrm{GL}(n, \mathbb{C})$, it can be shown that the length of $\gamma(t)$ is invariant. Moreover, as $\frac{\mathrm{d}}{\mathrm{d}t}\gamma^{-1}(t) = -\gamma^{-1}(t)\dot{\gamma}(t)\gamma^{-1}(t)$, we can show that this length is also invariant under inverse.

Let the geodesic $\gamma(t)$ be

$$\gamma(t) = A^{\frac{1}{2}}\left(A^{-\frac{1}{2}}BA^{-\frac{1}{2}}\right)^{t}A^{\frac{1}{2}} \in P(n) \tag{6}$$

with $\gamma(0) = A, \gamma(1) = B$ and $\gamma'(0) = \log(A^{-\frac{1}{2}}BA^{-\frac{1}{2}}) \in H(n)$. Let $A \circ B$ denote the midpoint of A and B by

$$A \circ B = A^{\frac{1}{2}}\left(A^{-\frac{1}{2}}BA^{-\frac{1}{2}}\right)^{\frac{1}{2}}A^{\frac{1}{2}}; \tag{7}$$

then the geodesic distance $d(A, B)$ is written explicitly as

$$d(A, B) = \left\|\log(A^{-\frac{1}{2}}BA^{-\frac{1}{2}})\right\| = \left(\sum_{i=1}^{n}\log(\lambda_i)^2\right)^{\frac{1}{2}}, \tag{8}$$

where λ_i are eigenvalues of $A^{-\frac{1}{2}}BA^{-\frac{1}{2}}$. Because λ_i are also eigenvalues of $A^{-1}B$, one can compute the distance $d(A, B)$, in practice, without invoking the matrix square root $A^{-\frac{1}{2}}$.

3 Control for positive definite Hermitian matrix systems

One of the purposes in control theory is to design the control input to make the output approximate the target. Many algorithms for specific approximation problems have been proposed [13, 14]. For example, a steepest descent algorithm was given based on the natural gradient, and the controller of an open-loop stochastic distribution control system was designed [5]. In biomedicine field, a controller law was developed, which can not only predict patients' diet but also give a probability description of their dietary behavior [15].

In this section, we take the geodesic distance on manifold $P(n)$ as the objective function to discuss control problems of positive definite Hermitian matrix systems. We assume that the output matrix in $P(n)$ depends only on the control input through a system, allowing us to define the output as a matrix $A(u) \in P(n)$ that is a function of the input $u = (u^1, u^2, \ldots, u^m)$. Our purpose is to design the control input u, such that the difference between $A(u)$ and another target positive definite Hermitian matrix B is as small as possible (see Figure 1).

The outline for designing an algorithm is as follows.

(1) Seek an appropriate objective function to represent the difference between the output matrix and the target.

(2) Compute the trajectory of input u to make the output system as close to the target as possible.

To make the output matrix $A(u)$ as close to the given target matrix B as possible, we use the geodesic distance (8) to measure the difference between the matrices $A(u)$ and B. Then we are going to design a controller and obtain u_* such that

$$u_* = \arg\min_{u} J(u), \tag{9}$$

where the objective function $J(u)$ is defined by

$$J(u) = d^2(A(u), B). \tag{10}$$

Let the system be defined such that

$$\mathcal{M} = \left\{ A(u) \mid u = (u^1, u^2, \ldots, u^m) \in \Theta \subset \mathbb{R}^m \right\} \tag{11}$$

is a submanifold of manifold $P(n)$, where the control input u plays the role of local coordinates.

In the following, we focus on this control problem using the Riemannian gradient descent algorithm and the natural gradient descent algorithm, respectively. Moreover, we will analyze the suitability of two algorithms in the illustrative examples. When the target matrix B lies on $\mathcal{M}$, both gradient descent algorithms are applicable. The main difference is that the Riemannian gradient descent algorithm realizes the optimization of the input trajectory while the natural gradient algorithm converges faster than the former. When the target matrix B does not lie on $\mathcal{M}$, however, only the natural gradient algorithm is applicable.

3.1 The Riemannian gradient descent algorithm

Now we consider how to solve the control problem proposed above based on the Riemannian gradient descent algorithm, under the condition that the target matrix B belongs to the output submanifold $\mathcal{M}$.

Because both the output $A(u)$ and the target matrix B lie on $\mathcal{M}$, the geodesic from $A(u)$ to B is adopted as the iterative trajectory and the negative gradient of the objective function $J(u)$ about $A(u)$ is taken as the direction of the algorithm.

Theorem 1. For the control input $u = (u^1, u^2, \ldots, u^m)$ of a given positive definite Hermitian matrix system, the iterative formula is given by

$$A(u_{k+1}) = A^{\frac{1}{2}}(u_k) \exp\left\{ -\eta_k \log\left(A^{-\frac{1}{2}}(u_k) B A^{-\frac{1}{2}}(u_k) \right) \right\} A^{\frac{1}{2}}(u_k), \tag{12}$$

where η_k is the learning rate at time k.

Proof. If the gradient of $J(u)$ about $A(u)$ is denoted by $\nabla_A J(u)$, then [7]

$$\nabla_A J(u_k) = A^{\frac{1}{2}}(u_k) \log\left(A^{-\frac{1}{2}}(u_k) B A^{-\frac{1}{2}}(u_k) \right) A^{\frac{1}{2}}(u_k). \tag{13}$$

Recall that the Riemannian exponential map $\exp_A$ on manifold $P(n)$ is defined by

$$\exp_A\{X\} = A^{\frac{1}{2}} \exp\left\{ A^{-\frac{1}{2}} X A^{-\frac{1}{2}} \right\} A^{\frac{1}{2}}, \tag{14}$$

where X is in the tangent space $T_A P(n)$. Then we obtain the iterative formula as

$$\begin{aligned} A(u_{k+1}) &= \exp_{A(u_k)}\{-\eta_k \nabla_A J(u_k)\} \\ &= A^{\frac{1}{2}}(u_k) \exp\left\{ -\eta_k A^{-\frac{1}{2}}(u_k) \nabla_A J(u_k) A^{-\frac{1}{2}}(u_k) \right\} A^{\frac{1}{2}}(u_k) \\ &= A^{\frac{1}{2}}(u_k) \exp\left\{ -\eta_k \log\left(A^{-\frac{1}{2}}(u_k) B A^{-\frac{1}{2}}(u_k) \right) \right\} A^{\frac{1}{2}}(u_k). \end{aligned} \tag{15}$$

This finishes the proof.

To solve the proposed control problem, the Riemannian gradient descent algorithm is given for positive definite Hermitian matrix systems (see Algorithm 1).

Algorithm 1 Riemannian gradient descent algorithm for positive definite Hermitian matrix systems

1. Set $u_0 = (u_0^1, u_0^2, \ldots, u_0^m)$ as an initial input. Choose a fixed learning rate η for simplicity and a desired tolerance $\varepsilon > 0$.
2. At time k, calculate $A(u_k)$ using Theorem 1 and $d(A(u_k), B)$.
3. If $d(A(u_k), B) < \varepsilon$ then stop. Otherwise, move to step 4.
4. Make k plus one, and then run step 2.

Remark 1. The initial output matrix $A(u_0)$ will converge to the final output matrix $A(u_*)$ along the geodesic connecting them. Hence this algorithm realizes the trajectory optimization of input u.

Remark 2. When the target matrix B does not belong to submanifold $\mathcal{M}$, there does not exist a geodesic on submanifold $\mathcal{M}$ to connect the output $A(u)$ and the target B. Thus, at this time, the Riemannian gradient algorithm needs to be improved.

3.2 Natural gradient descent algorithm

The ordinary gradient, which is often used in Euclidean spaces, does not give the steepest direction of an objective function on manifold, but the natural gradient does. Here, an important lemma about the natural gradient is first introduced, and then the natural gradient algorithm for the control problem is proposed.

Lemma 1. Let $L(\theta)$ be an objective function defined in a Riemannian manifold parameterized by $\theta \in \mathbb{R}^m$. The natural gradient algorithm for finding the minimum of $L(\theta)$ is written as [2]

$$\theta_{k+1} = \theta_k - \eta_k G^{-1} \nabla L(\theta_k), \tag{16}$$

where $G^{-1} = (g^{ij})$ represents the inverse of the Riemannian metric $G = (g_{ij})$ and

$$\nabla L(\theta) = \left(\frac{\partial}{\partial \theta^1} L(\theta), \frac{\partial}{\partial \theta^2} L(\theta), \ldots, \frac{\partial}{\partial \theta^m} L(\theta) \right). \tag{17}$$

In this subsection, the natural gradient descent algorithm for the considered system will be given from the viewpoint of information geometry. This algorithm can be applied whether the target matrix B is on the output submanifold $\mathcal{M}$ or not. The following lemma is useful for computing the gradient of objective function.

Lemma 2. A function-valued matrix is denoted by $X(t)$ with $t \in \mathbb{R}$, and two constant matrices are denoted by A and B. Then, if X is an invertible matrix with positive eigenvalues for any t in its definition domain, we have [16]

$$\frac{\mathrm{d}}{\mathrm{d}t} \operatorname{tr}\left(X^{\mathrm{T}}(t) X(t)\right) = 2 \operatorname{tr}\left(X^{\mathrm{T}}(t) \frac{\mathrm{d}}{\mathrm{d}t} X(t)\right), \tag{18}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \operatorname{tr}\left(\log X(t)\right) = \operatorname{tr}\left(X^{-1}(t) \frac{\mathrm{d}}{\mathrm{d}t} X(t)\right), \tag{19}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \operatorname{tr}\left(A X(t) B\right) = \operatorname{tr}\left(A \frac{\mathrm{d}}{\mathrm{d}t} X(t) B\right). \tag{20}$$

Let $u = (u^1, u^2, \ldots, u^m)$ belong to a parameter space on which an objective function $J(u)$ is defined; then we get the following theorem.

Theorem 2. The iterative formula on manifold $P(n)$ is given by

$$u_{k+1} = u_k - \eta_k G^{-1} \nabla J(u_k), \tag{21}$$

where the component of $\nabla J(u_t)$ satisfies

$$\frac{\partial}{\partial u_k^i} J(u_k) = 2 \operatorname{tr}\left(B^{-\frac{1}{2}} \log\left(B^{-\frac{1}{2}} A(u_k) B^{-\frac{1}{2}} \right) B^{\frac{1}{2}} A^{-1}(u_k) \frac{\partial}{\partial u_k^i} A(u_k) \right), \quad i = 1, 2, \ldots, m. \tag{22}$$

Proof. From Lemma 1, the iterative formula can be obtained as

$$u_{k+1} = u_k - \eta_k G^{-1} \nabla J(u_k), \tag{23}$$

where the Riemannian metric G can be calculated by (3). If we set $X(u_k) = \log(A^{-\frac{1}{2}}(u_k) B A^{-\frac{1}{2}}(u_k))$, then $X(u_k)$ is symmetric. Using Lemma 2, we have the fact that

$$\begin{aligned} \frac{\partial}{\partial u_k^i} J(u_k) &= 2 \operatorname{tr}\left(\log\left(B^{-\frac{1}{2}} A(u_k) B^{-\frac{1}{2}} \right) \frac{\partial}{\partial u_k^i} \log\left(B^{-\frac{1}{2}} A(u_k) B^{-\frac{1}{2}} \right) \right) \\ &= 2 \operatorname{tr}\left(B^{-\frac{1}{2}} \log\left(B^{-\frac{1}{2}} A(u_k) B^{-\frac{1}{2}} \right) B^{\frac{1}{2}} A^{-1}(u_k) \frac{\partial}{\partial u_k^i} A(u_k) \right), \quad i = 1, 2, \ldots, m. \end{aligned} \tag{24}$$

This completes the proof of Theorem 2.

In summary, we develop the natural gradient algorithm for the considered system as Algorithm 2.

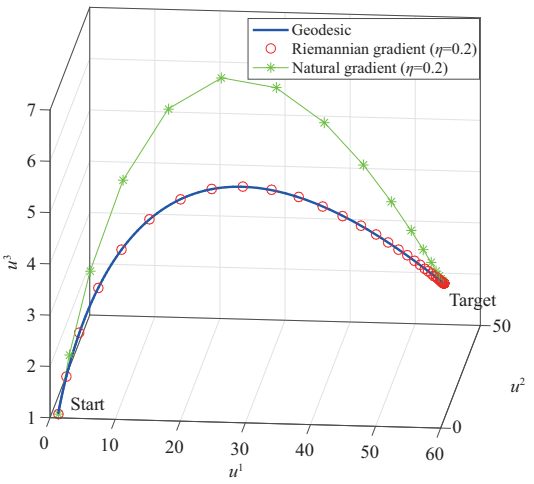

Figure 2 (Color online) Trajectory of u_k.

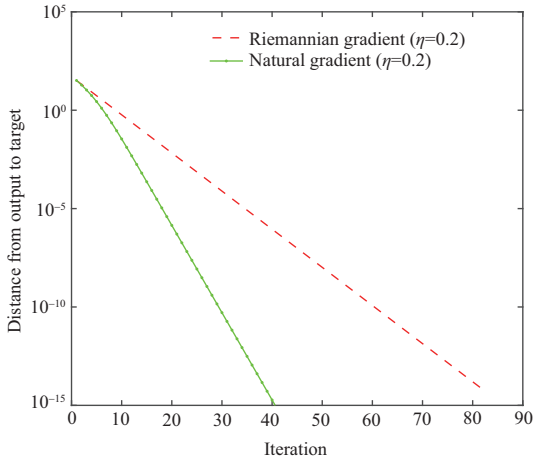

Figure 3 (Color online) Comparison of two algorithms.

Algorithm 2 Natural gradient algorithm for positive definite Hermitian matrix systems

1. Set $u_0 = (u_0^1, u_0^2, \ldots, u_0^m)$ as an initial input. Choose a fixed learning rate η and a desired tolerance $\varepsilon > 0$.
2. At time k, calculate u_k using Theorem 2 and $\nabla J(u_k)$.
3. If $\|\nabla J(u_k)\|_F < \varepsilon$, stop. Otherwise, move to step 4.
4. Make k plus one, and then run step 2.

3.3 Simulations

From the following examples, we will show the efficiency of the two proposed algorithms, where the tolerance is $\varepsilon = 10^{-15}$. In the first example, the target matrix B lies in submanifold $\mathcal{M}$, but in the second example it does not.

Example 1. We assume the target matrix B is a point of the output submanifold and hence both algorithms can be used. We choose a 3-dimensional input $u = (u^1, u^2, u^3)$ and define the matrix system as

$$A(u) = \begin{pmatrix} u^1 & \mathrm{i}u^3 \\ -\mathrm{i}u^3 & u^2 \end{pmatrix}, \quad u^1 > 0,\ u^1u^2 - (u^3)^2 > 0. \tag{25}$$

The output submanifold is then

$$\mathcal{M}_1 = \left\{ A(y;u) = \begin{pmatrix} u^1 & \mathrm{i}u^3 \\ -\mathrm{i}u^3 & u^2 \end{pmatrix} \;\middle|\; u^1 > 0, u^1u^2 - (u^3)^2 > 0 \right\}. \tag{26}$$

We take $u_0^1 = 1, u_0^2 = 2, u_0^3 = 1$ as the initial input u_0 and give the target matrix B by

$$B = \begin{pmatrix} 55 & 2\mathrm{i} \\ -2\mathrm{i} & 45 \end{pmatrix}, \tag{27}$$

and thus the coordinates of the target point B is $(55, 45, 2)$. Using Algorithms 1 and 2, the trajectories of the input u from the initial state $A(u_0)$ to the target matrix B are obtained efficiently (see Figure 2).

It is easy to see that the trajectory of the input u, which is given by the Riemannian gradient algorithm, is along the geodesic connecting the initial value u_0 and the target B so that the path is the shortest one. The natural gradient algorithm does not get the optimal trajectory of the input u, but the convergence rate is faster than that of the Riemannian gradient algorithm (see Figure 3).

Example 2. Now, we consider the situation that the target matrix B does not belong to the output submanifold. In this case, only Algorithm 2 is applied to the control problem. Setting the input u to be

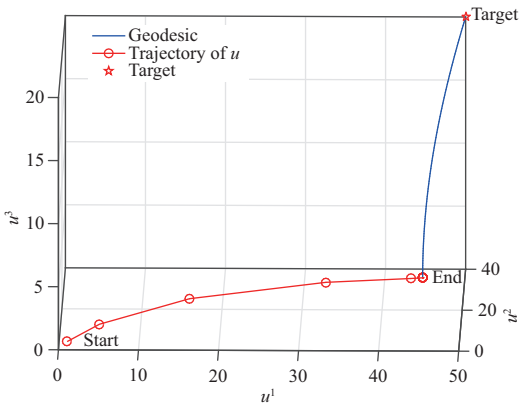

Figure 4 (Color online) Geodesic projection from target onto $\mathcal{M}_2$.

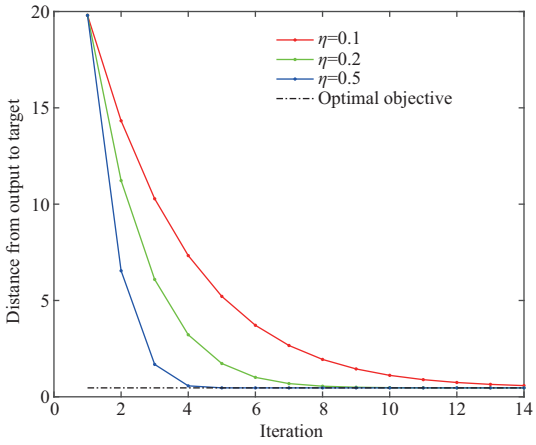

Figure 5 (Color online) Convergence of the natural gradient.

a 2-dimensional vector (u^1, u^2) and the output matrix to be that

$$A(u) = \begin{pmatrix} u^1 & 0 \\ 0 & u^2 \end{pmatrix}, \quad u^1, u^2 > 0, \tag{28}$$

then the output submanifold is

$$\mathcal{M}_2 = \left\{ A(u) = \begin{pmatrix} u^1 & 0 \\ 0 & u^2 \end{pmatrix} \;\middle|\; u^1, u^2 > 0 \right\}. \tag{29}$$

Let us take $u_0^1 = 1, u_0^2 = 4$ as the coordinates of the initial state $A(u_0)$ and give the target matrix B by

$$B = \begin{pmatrix} 50 & 20\mathrm{i} \\ -20\mathrm{i} & 40 \end{pmatrix}. \tag{30}$$

Then, using Algorithm 2 to simulate the control process, we obtain the trajectory of the input u from the initial state $A(u_0)$ to the approximate matrix $A(u_*)$ of the target matrix B efficiently. The coordinate of $A(u_*)$ is $u_* = (44.721, 35.777)$ which can be taken as the geodesic projection of the target B onto submanifold $\mathcal{M}_2$ (see Figure 4). Furthermore, when we set the learning rate $\eta = 0.1, 0.2, 0.5$, respectively, the efficiency and the convergence of Algorithm 2 are shown by Figure 5.

4 Karcher mean of Toeplitz positive definite Hermitian matrices

The mean of matrices plays an important role in many fields, such as signal processing, image denoising, and biomedical science [17–21]. Many algorithms have been developed to compute such means, e.g., [22–28]. From [7], the spatial or time complex data for the processing $Z_n = [z_1, z_2, \ldots, z_n]^{\mathrm{T}}$ is widely used in electromagnetics and other fields. Moreover, the covariance matrices $R_n = E[Z_n Z_n^{\mathrm{H}}]$ of Z_n are the Toeplitz positive definite Hermitian matrices:

$$R_n = \begin{pmatrix} r_0 & \overline{r}_1 & \cdots & \overline{r}_{n-1} \\ r_1 & r_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \overline{r}_1 \\ r_{n-1} & \cdots & r_1 & \overline{r}_0 \end{pmatrix}, \tag{31}$$

where $r_k = E[z_n \overline{z}_{n-k}]$ and $Z^{\mathrm{H}} R_n Z > 0$, for any $Z \in \mathbb{C}^n$.

In this section, we consider the complex circular multivariate Gaussian distribution of zero mean with probability density function

$$p(Z_n|R_n) = \frac{1}{\pi^n \det(R_n)} \exp\left\{-Z_n^{\mathrm{H}} R_n^{-1} Z_n\right\}. \tag{32}$$

Let $\mathrm{Sym}(n,\mathbb{C})$ be the set of all $n \times n$ Toeplitz positive definite Hermitian matrices. It is easy to show that $\mathrm{Sym}(n,\mathbb{C})$ is an (n^2-n+1)-dimensional submanifold of manifold $P(n)$. In the Riemannian sense, the mean $\overline{R}$ of N given positive definite Hermitian matrices $R^1, R^2, \ldots, R^N$ is defined as [29]

$$\overline{R} = \arg \min_{R \in \mathrm{Sym}(n,\mathbb{C})} \frac{1}{N} \sum_{i=1}^{N} d^2(R^i, R), \tag{33}$$

which is called the Karcher mean. For N distributions $p(\cdot|R^i), i = 1, 2, \ldots, N$, let R^k denote their covariance matrices and define the objective function as

$$L(R) = \frac{1}{N} \sum_{i=1}^{N} d^2(R^i, R). \tag{34}$$

Note that the local curvature of the distribution (32) is non-positive; hence the Karcher mean is unique [30]. In [31], it was shown that the Jacobi field of the Karcher mean vanishes. To estimate the Doppler ambiance, Barbaresco et al. [7] computed the Jacobi field and proposed the Riemannian gradient algorithm to compute the Karcher mean as

$$R_{k+1} = R_k^{\frac{1}{2}} \exp\left\{-\eta \sum_{i=1}^{N} \log\left(R_k^{-\frac{1}{2}} R^i R_k^{-\frac{1}{2}}\right)\right\} R_k^{\frac{1}{2}} \tag{35}$$

with η the learning rate.

In the following, the natural gradient algorithm is proposed to calculate the Karcher mean of N Toeplitz positive definite Hermitian matrices R^i $(i = 1, 2, \ldots, N)$.

4.1 Natural gradient descent algorithm

Let $L(\theta)$ be an objective function with the parameter $\theta = (\theta^1, \theta^2, \ldots, \theta^m)$. Analogous to the proof of Theorem 2, the following theorem is proposed.

Theorem 3. The iterative formula on manifold $\mathrm{Sym}(n,\mathbb{C})$ is obtained as follows:

$$\theta_{k+1} = \theta_k - \eta G^{-1} \nabla|_{\theta=\theta_k} L(\theta), \tag{36}$$

where

$$\nabla L(\theta) = \left(\frac{\partial}{\partial \theta^1} L(\theta), \frac{\partial}{\partial \theta^2} L(\theta), \ldots, \frac{\partial}{\partial \theta^m} L(\theta)\right) \tag{37}$$

and the component of gradient $\nabla L(\theta)$ satisfies

$$\frac{\partial}{\partial \theta^j} L(\theta) = \frac{2}{N} \mathrm{tr}\left(\sum_{i=1}^{N} R_i^{-\frac{1}{2}} \log\left(R_i^{-\frac{1}{2}} R R_i^{-\frac{1}{2}}\right) R_i^{\frac{1}{2}} R^{-1} \frac{\partial}{\partial \theta^j} R\right), \quad j = 1, 2, \ldots, m. \tag{38}$$

Now we are ready to formulate the natural gradient algorithm (see Algorithm 3).

Algorithm 3 Natural gradient algorithm for the Karcher mean of N matrices on manifold $\mathrm{Sym}(n,\mathbb{C})$

1. Take the arithmetic mean $\frac{1}{N}\sum_{i=1}^{N} R^i$ as the initial point θ_0. Choose a learning rate η and a desired tolerance $\varepsilon > 0$.
2. At time k, calculate θ_k using Theorem 3 and $\nabla L(\theta_k)$.
3. If $\|\nabla L(\theta_k)\| < \varepsilon$, stop. Otherwise, move to step 4.
4. Make k plus one, and then run step 2.

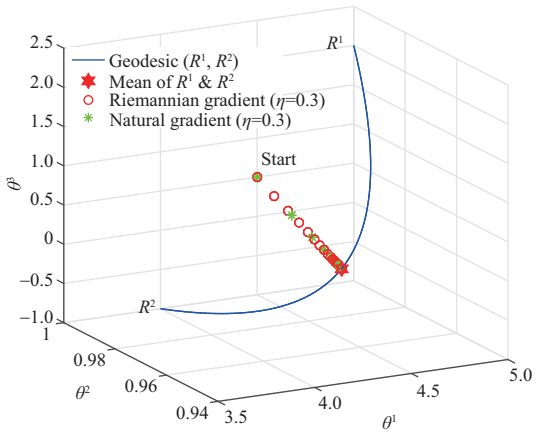

Figure 6 (Color online) Iterative process.

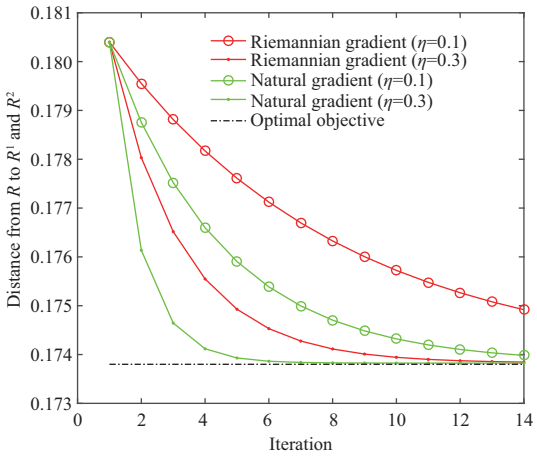

Figure 7 (Color online) Comparison of two algorithms.

4.2 Simulations

In this subsection, we use two gradient descent algorithms mentioned above to compute Karcher mean of N given Toeplitz positive definite Hermitian matrices with $\varepsilon = 10^{-15}$. From the following examples, it is shown that Algorithm 3 is more efficient than the algorithm (35).

For simplicity, we choose a 2-dimensional spatial or time complex data for the processing: $Z_2 = [z_1, z_2]^{\mathrm{T}}$. Then, the covariance matrices of Z_2 can be written as

$$\begin{pmatrix} \theta^1 & \theta^2 + \mathrm{i}\,\theta^3 \\ \theta^2 - \mathrm{i}\,\theta^3 & \theta^1 \end{pmatrix}. \tag{39}$$

Example 3. We compute the Karcher mean of R^1, R^2 on manifold $\mathrm{Sym}(2, \mathbb{C})$ as the first example, where

$$R^1 = \begin{pmatrix} 5 & 1+2\,\mathrm{i} \\ 1-2\,\mathrm{i} & 5 \end{pmatrix}, \quad R^2 = \begin{pmatrix} 4 & 1-\mathrm{i} \\ 1+\mathrm{i} & 4 \end{pmatrix}. \tag{40}$$

It is easy to get their Karcher mean $R^1 \circ R^2$ from (7). To show the algorithm's efficiency, Algorithm 3 and the algorithm (35) are used, respectively. Choose the arithmetic mean $\frac{1}{2}\sum_{i=1}^{2} R^i$ as the initial point θ_0. On manifold $\mathrm{Sym}(2, \mathbb{C})$, the adjustment of the coordinate vector θ is given by (35) and (36). From (6), the geodesic between R^1 and R^2 is computed (see Figure 6). We mark the Karcher mean by a red pentacle. It is shown that both algorithms converge to the red pentacle. It also demonstrates that Algorithm 3 converges faster than the algorithm (35) (see Figure 7).

Example 4. Here, we consider three 2×2 Toeplitz positive definite Hermitian matrices

$$R^1 = \begin{pmatrix} 3 & 1.5+2\,\mathrm{i} \\ 1.5-2\,\mathrm{i} & 3 \end{pmatrix}, \quad R^2 = \begin{pmatrix} 2 & 1-\mathrm{i} \\ 1+\mathrm{i} & 2 \end{pmatrix}, \quad R^3 = \begin{pmatrix} 4 & 1+2\,\mathrm{i} \\ 1-2\,\mathrm{i} & 4 \end{pmatrix}. \tag{41}$$

Using (6), we can get the geodesics between each two of the three points R^1, R^2 and R^3 on $\mathrm{Sym}(2, \mathbb{C})$, which form a geodesic triangle (see Figure 8). The midpoint of each geodesic is obtained using (7). Thus, each median connects a vertex with the midpoint of its opposing side. It can be seen that these centerlines always meet in a single point in Euclidean spaces, namely the Karcher mean. But this phenomenon usually does not occur in curved spaces, as Figure 8 shows. In fact, when we change the perspective of Figure 8, it is shown that the three midlines are most likely non-coplanar (see Figure 9).

Now we compute the Karcher mean $p(\cdot|\overline{R})$ of the distributions $p(\cdot|R^i), i = 1, 2, 3$ using these two algorithms. Similar to Example 3, first we still take the arithmetic mean $\frac{1}{3}\sum_{i=1}^{3} R^i$ as the initial point

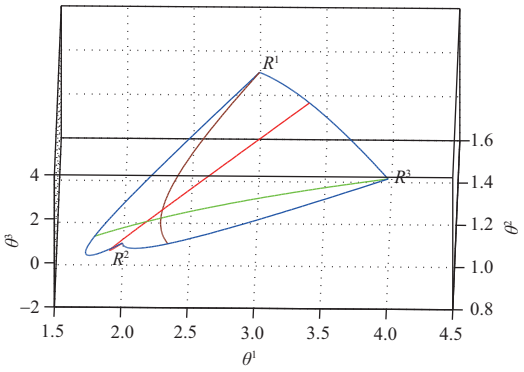

Figure 8 (Color online) Geodesic triangle (position 1).

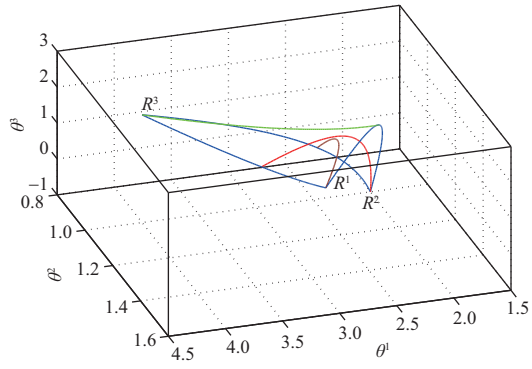

Figure 9 (Color online) Geodesic triangle (position 2).

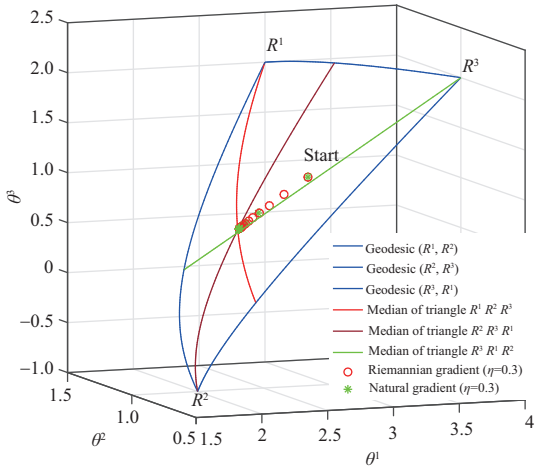

Figure 10 (Color online) Convergence to the Karcher mean.

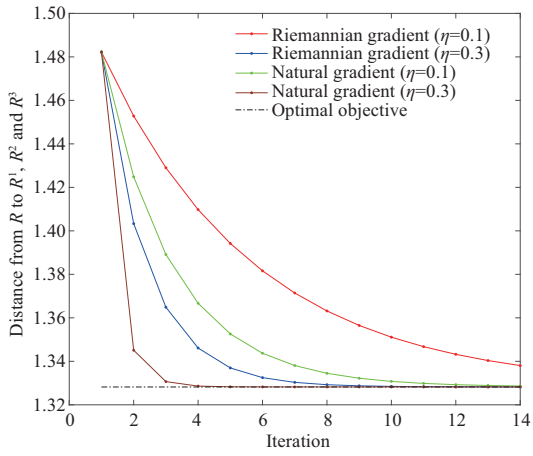

Figure 11 (Color online) Comparison of two algorithms.

θ_0. On manifold $\mathrm{Sym}(2,\mathbb{C})$, the adjustment of the coordinate vector θ is given by (35) and (36). Finally, both algorithms converge to the Karcher mean (see Figure 10):

$$\overline{R} = \arg \min_{R\in \mathrm{Sym}(2,\mathbb{C})} \frac{1}{3}\sum_{i=1}^{3} d^2(R^i, R) = \begin{pmatrix} 2.295 & 0.980 + 0.617\,\mathrm{i} \\ 0.980 - 0.617\,\mathrm{i} & 2.295 \end{pmatrix}. \tag{42}$$

From Figure 11 we can find that Algorithm 3 is faster than the algorithm (35).

5 Conclusion

In this paper, we applied the Riemannian gradient descent algorithm and the natural gradient descent algorithm to the control of positive Hermitian matrix systems as well as the computation of the Karcher mean of Toeplitz positive definite Hermitian matrices. Their behaviors were also compared.

For the control system, when the target matrix belongs to the output submanifold, the Riemannian gradient descent algorithm and the natural gradient descent algorithm are both applicable. It was shown that the former realizes the optimization of input trajectory while the latter has preferable convergence. We also proposed the natural gradient descent algorithm for the Karcher mean of matrices on the submanifold $\mathrm{Sym}(n,\mathbb{C})$, taking the sum of geodesic distances as the objective function. The simulations showed that the convergence rate of the natural gradient descent algorithm is faster than that of the Riemannian gradient descent algorithm.

Acknowledgements Xiaomin DUAN was supported by National Science and Technology Major Project of China (Grant No. 2016YFF02030012), National Natural Science Foundation of China (Grant No. 61401058) and Natural Science Foundation of Liaoning Province (Grant No. 20180550112). Huafei SUN was partially supported by National Natural Science Foundation of China (Grant No. 61179031). Linyu PENG was supported by JSPS Grant-in-Aid for Scientific Research (Grant No. 16KT0024), the MEXT "Top Global University Project", Waseda University Grant for Special Research Projects (Grant Nos. 2019C–179, 2019E–036), and Waseda University Grant Program for Promotion of International Joint Research.

References

1 Amari S, Douglas S C. Why natural gradient? In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, 1998. 2: 1213–1216
2 Amari S. Natural gradient works efficiently in learning. Neural Comput, 1998, 10: 251–276
3 Tang Y, Li J. Normalized natural gradient in independent component analysis. Signal Process, 2010, 90: 2773–2777
4 Li C H, Zhang E C, Jiu L, et al. Optimal control on special Euclidean group via natural gradient algorithm. Sci China Inf Sci, 2016, 59: 112203
5 Zhang Z, Sun H, Peng L, et al. A natural gradient algorithm for stochastic distribution systems. Entropy, 2014, 16: 4338–4352
6 Zhao J, Yu X. Adaptive natural gradient learning algorithms for Mackey-Glass chaotic time prediction. Neurocomputing, 2015, 157: 41–45
7 Barbaresco F, Roussigny H. Innovative tools for Radar signal processing based on Cartan's geometry of SPD matrices and information geometry. In: Proceedings of IEEE International Radar Conference, Rome, 2008. 1–6
8 Lenglet C, Rousson M, Deriche R, et al. Statistics on the manifold of multivariate normal distributions: theory and application to diffusion tensor MRI processing. J Math Imag Vis, 2006, 25: 423–444
9 Duan X M, Sun H F, Zhao X Y. Riemannian gradient algorithm for the numerical solution of linear matrix equations. J Appl Math, 2014, 2014: 1–7
10 Moakher M. On the averaging of symmetric positive-definite tensors. J Elasticity, 2006, 82: 273–296
11 Moakher M. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. SIAM J Matrix Anal Appl, 2005, 26: 735–747
12 Jost J. Riemannian Geometry and Geometric Analysis. 3rd ed. Berlin: Springer, 2002
13 Gozde H, Taplamacioglu M C, Kocaarslan İ. Comparative performance analysis of artificial bee colony algorithm in automatic generation control for interconnected reheat thermal power system. Int J Electrical Power Energy Syst, 2012, 42: 167–178
14 Kim B G, Lee J W. Stochastic utility-based flow control algorithm for services with time-varying rate requirements. Comput Netw, 2012, 56: 1329–1342
15 Hughes C S, Patek S D, Breton M, et al. Anticipating the next meal using meal behavioral profiles: a hybrid model-based stochastic predictive control algorithm for T1DM. Comput Methods Programs Biomed, 2011, 102: 138–148
16 Zhang X D. Matrix Analysis and Application. Beijing: Tsinghua University Press, 2004
17 Adamczak R, Litvak A E, Pajor A, et al. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. J Amer Math Soc, 2010, 23: 535–561
18 Bhatia R, Jain T, Lim Y. Inequalities for the Wasserstein mean of positive definite matrices. Linear Algebra Its Appl, 2019, 576: 108–123
19 Liu J K, Wang X S, Wang T, et al. Application of information geometry to target detection for pulsed-Doppler radar. J Natl Univ Defense Tech, 2011, 33: 77–80
20 Takahashi R, Yoshida N, Takada M, et al. Simulations of baryon acoustic oscillations. II. Covariance matrix of the matter power Spectrum. Astrophys J, 2009, 700: 479–490
21 Fiori S. Learning the Fréchet mean over the manifold of symmetric positive-definite matrices. Cogn Comput, 2009, 1: 279–291
22 Chebbi Z, Moakher M. Means of Hermitian positive-definite matrices based on the log-determinant α-divergence function. Linear Algebra Its Appl, 2012, 436: 1872–1889
23 Guven A. Approximation of continuous functions by matrix means of hexagonal Fourier series. Results Math, 2018, 73: 18
24 Nobari E, Kakavandi B A. A geometric mean for Toeplitz and Toeplitz-block block-Toeplitz matrices. Linear Algebra Its Appl, 2018, 548: 189–202
25 Bini D A, Iannazzo B. Computing the Karcher mean of symmetric positive definite matrices. Linear Algebra Its Appl, 2013, 438: 1700–1710
26 Iannazzo B, Porcelli M. The Riemannian Barzilai-Borwein method with nonmonotone line search and the matrix geometric mean computation. IMA J Numer Anal, 2018, 38: 495–517
27 Fiori S, Tanaka T. An algorithm to compute averages on matrix Lie groups. IEEE Trans Signal Process, 2009, 57: 4734–4743
28 Kaneko T, Fiori S, Tanaka T. Empirical arithmetic averaging over the compact stiefel manifold. IEEE Trans Signal Process, 2013, 61: 883–894
29 Grove K, Karcher H, Ruh E A. Jacobi fields and Finsler metrics on compact Lie groups with an application to differentiable pinching problems. Math Ann, 1974, 211: 7–21
30 Karcher H. Riemannian center of mass and mollifier smoothing. Commun Pure Appl Math, 1977, 30: 509–541
31 Arnaudon M, Li X M. Barycenters of measures transported by stochastic flows. Ann Probab, 2005, 33: 1509–1543