

# Riemannian optimization on the simplex of positive definite matrices

**Bamdev Mishra**

*Microsoft, India*

**Hiroyuki Kasai**

*Waseda University, Japan*

**Pratik Jawanpuria**

*Microsoft, India*

BAMDEV.M@MICROSOFT.COM

HIROYUKI.KASAI@WASEDA.JP

PRATIK.JAWANPURIA@MICROSOFT.COM

## Abstract

In this work, we generalize the probability simplex constraint to matrices, i.e.,  $\mathbf{X}_1 + \mathbf{X}_2 + \dots + \mathbf{X}_K = \mathbf{I}$ , where  $\mathbf{X}_i \succeq 0$  is a symmetric positive semidefinite matrix of size  $n \times n$  for all  $i = \{1, \dots, K\}$ . By assuming positive definiteness of the matrices, we show that the constraint set arising from the matrix simplex has the structure of a smooth Riemannian submanifold. We discuss a novel Riemannian geometry for the matrix simplex manifold and show derivation of first- and second-order optimization related ingredients.

## 1. Introduction

Column (or row) stochastic matrices are those where each column (or row) has non-negative entries that sum to 1. Such matrices are shown to be useful in many machine learning applications [7, 9, 12, 15, 16]. The constraint of interest for those matrices is

$$x_1 + x_2 + \dots + x_K = 1, \text{ where } x_i \geq 0 \text{ for all } i = \{1, \dots, K\}, \quad (1)$$

which is also called the *probability simplex* constraint.

In this work, we propose to generalize the constraint (1) to constraints with matrices, i.e., the matrix simplex constraint

$$\mathbf{X}_1 + \mathbf{X}_2 + \dots + \mathbf{X}_K = \mathbf{I}, \quad (2)$$

where  $\mathbf{X}_i \succeq \mathbf{0}$  is a symmetric positive semidefinite matrix of size  $n \times n$  for all  $i = \{1, 2, \dots, K\}$ . Although the constraint (2) is a natural generalization of (1), its study is rather limited [10, 13]. An interesting property of this constraint is its ability to learn mutually orthogonal subspaces with applications in machine learning [6] and computer vision [8].

We discuss a novel Riemannian geometry for the set obtained from the constraint (2) with strict positive definiteness of the matrices. Strict positive definiteness of matrices is needed to obtain a differentiable manifold structure. The proposed Riemannian structure allows to handle potentially semidefinite, i.e., rank deficient, matrices gracefully by scaling those elements to the boundary of the manifold. The main aim of this work is to develop optimization-related ingredients that allow Riemannian optimization on this constraint set. The versatile framework of Riemannian optimization enables to perform first-order and second-order optimization of a smooth function. We show

the efficacy of the modeling on a toy problem of reconstructing orthogonal subspaces from their noisy estimates.

The manifold related code files are available with the manifold optimization toolbox Manopt [4].

## 2. The matrix simplex manifold

We define the matrix simplex manifold of interest as

$$\mathcal{M}_n^K := \{(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K) : \begin{array}{l} \mathbf{X}_1 + \mathbf{X}_2 + \dots + \mathbf{X}_K = \mathbf{I}, \\ \mathbf{X}_i \in \mathbb{R}^{n \times n}, \text{ and} \\ \mathbf{X}_i \succ \mathbf{0} \text{ for all } i \in \{1, 2, \dots, K\} \end{array}\}. \quad (3)$$

It should be noted that the positive semidefiniteness constraint  $\mathbf{X}_i \succeq \mathbf{0}$  is replaced with the positive definiteness constraint  $\mathbf{X}_i \succ \mathbf{0}$  to ensure that the set  $\mathcal{M}_n^K$  is differentiable. Below, we impose a Riemannian structure to the matrix simplex manifold (3) and discuss ingredients that allow to develop optimization algorithms systematically [1, 5].

### 2.1. Riemannian metric and tangent space projector

An element  $x$  of  $\mathcal{M}_n^K$  is numerically represented as the structure  $(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K)$  which is a collection of  $K$  symmetric positive definite matrices of size  $n \times n$ .

The tangent space of  $\mathcal{M}_n^K$  at an element  $x$  is the linearization of the manifold, i.e., the constraint (2). Accordingly, the tangent space characterization of  $\mathcal{M}_n^K$  at  $x$  is

$$T_x \mathcal{M}_n^K = \{(\xi_{\mathbf{X}_1}, \xi_{\mathbf{X}_2}, \dots, \xi_{\mathbf{X}_K}) : \begin{array}{l} \xi_{\mathbf{X}_1} + \xi_{\mathbf{X}_2} + \dots + \xi_{\mathbf{X}_K} = \mathbf{0} \\ \xi_{\mathbf{X}_i} \in \mathbb{R}^{n \times n}, \text{ and} \\ \xi_{\mathbf{X}_i}^\top = \xi_{\mathbf{X}_i} \text{ for all } i \in \{1, 2, \dots, K\} \end{array}\}. \quad (4)$$

It can be shown that  $\mathcal{M}_n^K$  is an embedding submanifold of the  $\mathcal{S}_n^K := \text{SPD}_n \times \text{SPD}_n \times \dots \times_K \text{SPD}_n$ , which is the Cartesian product of  $K$  manifolds of symmetric positive definite matrices of size  $n \times n$  [1, 5]. Here  $\text{SPD}_n$  denotes the manifold of  $n \times n$  symmetric positive definite matrices that has a well-known Riemannian geometry [2]. The dimension of the manifold  $\mathcal{M}_n^K$  is  $(K-1)n(n+1)/2$ .

We endow the manifold with a smooth metric  $g_x : T_x \mathcal{M}_n^K \times T_x \mathcal{M}_n^K \rightarrow \mathbb{R}$  (inner product) at every  $x \in \mathcal{M}_n^K$  [1]. A natural choice of the metric is based on the well-known bi-invariant metric of  $\text{SPD}_n$  [2], i.e.,

$$g_x(\xi_x, \eta_x) := \sum_i \text{trace}(\mathbf{X}_i^{-1} \xi_{\mathbf{X}_i} \mathbf{X}_i^{-1} \eta_{\mathbf{X}_i}). \quad (5)$$

Once the manifold  $\mathcal{M}_n^K$  is endowed with the metric (5), the manifold  $\mathcal{M}_n^K$  turns into a Riemannian submanifold of  $\mathcal{S}_n^K$ . Following [1, Chapters 3 and 4], the Riemannian submanifold structure allows the computation of the Riemannian gradient and Hessian of a function (on the manifold) in a straightforward manner from the partial derivatives of the function.

A critical ingredient in those computations is the computation of the linear projection operator of a vector in the ambient space  $\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \times \dots \times_K \mathbb{R}^{n \times n}$  onto the tangent space (4) at an element of  $\mathcal{M}_n^K$ . In particular, given  $z = (\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_K) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \times \dots \times_K \mathbb{R}^{n \times n}$  in the

ambient space, we compute the projection operator  $\Pi_x : \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \times \dots \times_K \mathbb{R}^{n \times n} \rightarrow T_x \mathcal{M}_n^K$ , orthogonal with respect to the metric (5), as [11]

$$\Pi_x(z) = \arg \min_{\xi_x \in T_x \mathcal{M}_n^K} -g_x(z, \xi_x) + \frac{1}{2}g_x(\xi_x, \xi_x),$$

which has the expression

$$\Pi_x(z) = (\mathbf{Z}_1 + \mathbf{X}_1 \mathbf{\Lambda} \mathbf{X}_1, \mathbf{Z}_2 + \mathbf{X}_2 \mathbf{\Lambda} \mathbf{X}_2, \dots, \mathbf{Z}_K + \mathbf{X}_K \mathbf{\Lambda} \mathbf{X}_K), \quad (6)$$

where  $\mathbf{\Lambda}$  is the symmetric matrix that is the solution to the linear system

$$\sum_i \mathbf{X}_i \mathbf{\Lambda} \mathbf{X}_i = - \sum_i \mathbf{Z}_i. \quad (7)$$

It is easy to verify that

- $\Pi_x(z)$  belongs to the tangent space  $T_x \mathcal{M}_n^K$  and
- $z - \Pi_x(z)$  and  $\Pi_x(z)$  complementary to each other with respect to the chosen metric (5) for all choices of  $z$ .

## 2.2. Retraction operator

Given a vector in the tangent space, the retraction operator maps it to an element of the manifold [1, Chapter 4]. Overall, the notion of retraction operation allows to move on the manifold, which is required by any optimization algorithm.

A natural choice of the retraction operator on the manifold  $\mathcal{M}_n^K$  is inspired from the well-known exponential mapping operation on  $\mathbb{S}\mathbb{P}\mathbb{D}_n$ , the manifold of positive definite matrices [2]. However, this only ensures positive definiteness of the output matrices. To maintain the summation equal to  $\mathbf{I}$  constraint, we additionally *normalize* in a particular fashion. Overall, given a tangent vector  $\xi_x \in T_x \mathcal{M}_n^K$ , the expression for the retraction operator  $R_x : T_x \mathcal{M}_n^K \rightarrow \mathcal{M}_n^K$  is

$$R_x(\xi_x) := (\mathbf{Y}_{\text{sum}}^{-1/2} \mathbf{Y}_1 \mathbf{Y}_{\text{sum}}^{-1/2}, \mathbf{Y}_{\text{sum}}^{-1/2} \mathbf{Y}_2 \mathbf{Y}_{\text{sum}}^{-1/2}, \dots, \mathbf{Y}_{\text{sum}}^{-1/2} \mathbf{Y}_K \mathbf{Y}_{\text{sum}}^{-1/2}), \quad (8)$$

where  $\xi_x = (\xi_{\mathbf{X}_1}, \xi_{\mathbf{X}_2}, \dots, \xi_{\mathbf{X}_K})$ ,  $\mathbf{Y}_i = \mathbf{X}_i(\exp(\mathbf{X}_i^{-1} \xi_{\mathbf{X}_i}))$ ,  $\mathbf{Y}_{\text{sum}} = \sum_i \mathbf{Y}_i$ , and  $\exp(\cdot)$  is the matrix exponential operator.

To show that the operator (8) is a retraction operator, we need to verify the conditions [1, Chapter 4]:

- the centering condition, i.e.,  $R_x(0_x) = x$  and
- the local rigidity condition, i.e.,  $DR_x(0_x) = \text{id}_{T_x \mathcal{M}_n^K}$ , where  $\text{id}_{T_x \mathcal{M}_n^K}$  denotes the identity mapping on  $T_x \mathcal{M}_n^K$ .

The centering condition for (8) is straightforward to verify by setting  $\xi_x = 0$ . To verify the local rigidity condition, we analyze the differential of the retraction operator locally, which is the composition of two steps: the first one is through the matrix exponential and the second is through the normalization by pre and post multiplying with  $\mathbf{Y}_{\text{sum}}^{-1/2}$ . The matrix exponential is locally rigid due to the fact that it defines the well-known exponential mapping on the  $\mathbb{S}\mathbb{P}\mathbb{D}_n$  manifold [1, 2]. The normalization step (with pre and post multiplying by  $\mathbf{Y}_{\text{sum}}^{-1/2}$ ) does not change local rigidity. Hence, the overall composition (8) satisfies both the centering and local rigidity conditions needed to be a retraction operation.

### 2.3. Riemannian gradient and Hessian computations

As mentioned earlier, a benefit of the Riemannian submanifold structure is that it allows to compute the Riemannian gradient and Hessian of a function in a systematic manner. To that end, we consider a smooth function  $f : \mathcal{M}_n^K \rightarrow \mathbb{R}$  on the manifold. We also assume that it is well-defined on  $\mathcal{S}_n^K$ .

If  $\nabla_x f$  is the Euclidean gradient of  $f$  at  $x \in \mathcal{M}_n^K$ , then the Riemannian gradient  $\text{grad}_x f$  has the expression

$$\begin{aligned} \text{grad}_x f &= \Pi_x(\text{gradient on } \mathcal{S}_n^K) \\ &= \Pi_x(\mathbf{X}_1 \text{symm}(\nabla_{\mathbf{X}_1} f) \mathbf{X}_1, \mathbf{X}_2 \text{symm}(\nabla_{\mathbf{X}_2} f) \mathbf{X}_2, \dots, \mathbf{X}_K \text{symm}(\nabla_{\mathbf{X}_K} f) \mathbf{X}_K), \end{aligned}$$

where  $\nabla_{\mathbf{X}_i} f$  is the partial derivative of  $f$  at  $x$  with respect to  $\mathbf{X}_i$  and  $\Pi_x$  is the tangent space projection operator defined in (6). Here,  $\text{symm}(\cdot)$  extracts the symmetric part of a matrix, i.e.,  $\text{symm}(\mathbf{\Delta}) = (\mathbf{\Delta} + \mathbf{\Delta}^\top)/2$ .

The computation of the Riemannian Hessian on the manifold  $\mathcal{M}_n^K$  involves the notion of Riemannian connection [1, Section 5.5]. The Riemannian connection, denoted as  $\nabla_{\xi_x} \eta_x$ , at  $x \in \mathcal{M}_n^K$  generalizes the *covariant-derivative* of the tangent vector  $\eta_x \in T_x \mathcal{M}_n^K$  along the direction of the tangent vector  $\xi_x \in T_x \mathcal{M}_n^K$  on the manifold  $\mathcal{M}_n^K$ . Since  $\mathcal{M}_n^K$  is a Riemannian submanifold of the manifold  $\mathcal{S}_n^K$ , the computation of the Riemannian connection enjoys a simple expression in terms of the computations on the symmetric positive definite manifold  $\mathbb{SPD}_n$  [2]. In particular, the Riemannian connection on  $\mathcal{M}_n^K$  is obtained by restricting the connection on  $\mathcal{S}_n^K$  to the tangent space  $T_x \mathcal{M}_n^K$ . The connection on  $\mathcal{S}_n^K$  is easy to derive thanks to the well-known Riemannian geometry of  $\mathbb{SPD}_n$ . Overall, the Riemannian connection expression for  $\mathcal{M}_n^K$  is

$$\begin{aligned} \nabla_{\xi_x} \eta_x &= \Pi_x(\text{connection on } \mathcal{S}_n^K) \\ &= \Pi_x(D\eta_x[\xi_x] - (\text{symm}(\xi_{\mathbf{X}_1} \mathbf{X}_1^{-1} \eta_{\mathbf{X}_1}), \dots, \text{symm}(\xi_{\mathbf{X}_K} \mathbf{X}_K^{-1} \eta_{\mathbf{X}_K}))), \end{aligned} \quad (9)$$

where  $D\eta_x[\xi_x]$  denotes the directional derivative of  $\eta_x$  along  $\xi_x$ . Based on the expression (9), the Riemannian Hessian operation  $\text{Hess}_x f[\xi_x]$  along a tangent vector  $\xi_x \in T_x \mathcal{M}_n^K$  has the expression

$$\text{Hess}_x f[\xi_x] = \nabla_{\xi_x} \text{grad}_x f.$$

### 2.4. Computational cost of optimization ingredients

The expressions shown earlier involve matrix operations that cost  $O(n^3 K)$ . The solution to the system (7) can be obtained iteratively using standard linear equation solvers. The overall cost for the computations is linear in  $K$ .

## 3. Extension to other cases

### The Hermitian case

The earlier developments easily extend to Hermitian positive definite matrices satisfying the constraint (2). The matrix transpose operation is replaced with the conjugate transpose operation [14]. The expressions in Section 2 are modified accordingly.

### Decomposition of a symmetric positive definite matrix

The constraint (2) can be generalized to decomposition of a general symmetric positive definite matrix  $\mathbf{M} \succ \mathbf{0}$ , i.e.,

$$\mathbf{X}_1 + \mathbf{X}_2 + \dots + \mathbf{X}_K = \mathbf{M}. \quad (10)$$

The constraint (10) can be translated to the simplex constraint (2) with a variable change by pre- and post-multiplying with  $\mathbf{M}^{-1/2}$ . Consequently, the optimization ingredients developed in Section 2 can be used to perform optimization over the constraint set (10).

### Dealing with large-scale scenarios

As discussed in Section 2.4, the manifold related operations cost  $O(n^3)$ . For large values of  $n$ , the computational cost is prohibitive. A way forward is to learn the matrices  $\mathbf{X}_i$ s in a restricted subspace. This is achieved by modeling  $\mathbf{X}_i = \mathbf{U}\mathbf{B}_i\mathbf{U}^\top$  as a rank- $r$  matrix, where the subspace is captured by  $\mathbf{U} \in \mathbb{R}^{n \times r}$  and  $\mathbf{B}_i \succ \mathbf{0}$  is a  $r$ -by- $r$  symmetric positive definite matrix [3]. It should be noted that  $\mathbf{U}$  is common across all  $\mathbf{X}_i$ s. Learning of  $\mathbf{U}$  and  $\mathbf{B}_i$ s is then equivalent to optimizing with the constraints  $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$  and  $\mathbf{B}_1 + \mathbf{B}_2 + \dots + \mathbf{B}_K = \mathbf{I}$ , which is viewed as a Cartesian product of Stiefel and matrix simplex manifolds. The computations of optimization ingredients in this case cost  $O(nr^2 + r^3K)$  which is much less than  $O(n^3K)$  for  $r \ll n$ .

## 4. Toy example: learning mutually orthogonal subspaces

The constraint (2) allows to learn mutually orthogonal subspaces, i.e.,  $\mathbf{X}_1, \mathbf{X}_2, \dots$ , and  $\mathbf{X}_K$  are mutually orthogonal. To this end, we consider three matrices  $\mathbf{A}_1, \mathbf{A}_2$ , and  $\mathbf{A}_3$  of size  $100 \times 100$  such that they share mutually orthogonal eigenvectors and have all eigenvalues as either 1 or 0. The eigenvalues are added with uniformly random numbers between 0 and 1 to generate noisy matrices  $\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2$ , and  $\hat{\mathbf{A}}_3$ . We solve the problem:

$$\min_{(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K) \in \mathcal{M}_n^K} - \sum_k \text{trace}(\mathbf{X}_k \hat{\mathbf{A}}_k). \quad (11)$$

Here,  $n = 100$  and  $K = 3$ . To measure effectiveness of learning mutually orthogonal matrices, we define the mutual orthogonality error for  $\mathbf{X}_i$  as  $\frac{\text{trace}(\mathbf{X}_i(\mathbf{I} - \mathbf{X}_i))}{\|\mathbf{X}_i\|_{\text{Fro}}\|\mathbf{I} - \mathbf{X}_i\|_{\text{Fro}}}$ . A zero mutual orthogonality error for  $\mathbf{X}_i$  ensures that  $\mathbf{X}_i$  is orthogonal to other matrices.

We implement a trust-region algorithm for solving (11). Figure 1 shows the performance of the algorithm on the problem instance. We observe that our algorithm is able to achieve a high degree of mutual orthogonality between the learned matrices as the error falls below  $10^{-6}$ .

In this case, the mutual orthogonality error for  $\hat{\mathbf{A}}_1$ , which is  $\frac{\text{trace}(\hat{\mathbf{A}}_1(\hat{\mathbf{A}}_2 + \hat{\mathbf{A}}_3))}{\|\hat{\mathbf{A}}_1\|_{\text{Fro}}\|\hat{\mathbf{A}}_2 + \hat{\mathbf{A}}_3\|_{\text{Fro}}}$  is 0.68. Similarly, the mutual orthogonality errors for  $\hat{\mathbf{A}}_2$  and  $\hat{\mathbf{A}}_3$  are 0.66 and 0.65, respectively.

## 5. Conclusion

We discussed the matrix simplex manifold as a generalization of the popular probability simplex constraint to symmetric positive semidefinite matrices. The main aim of the work was to understand the geometry of the manifold from an optimization perspective. To this end, the expressions of optimization-related tools were developed.

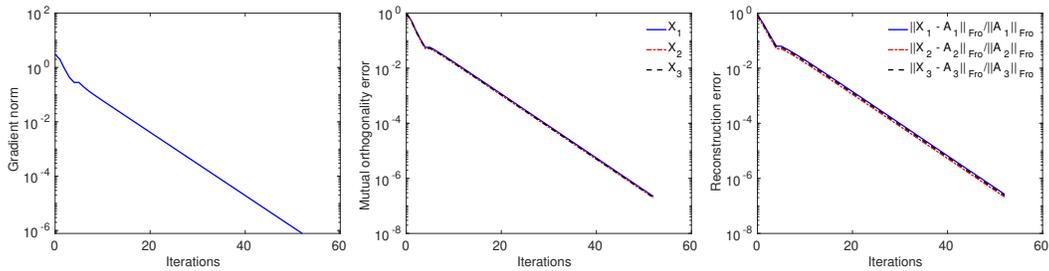


Figure 1: The matrix simplex constraint allows learning of mutually orthogonal subspaces. Solving (11) leads to identifying mutually orthogonal matrices while ensuring a low reconstruction error.

## Acknowledgment

H. Kasai was partially supported by JSPS KAKENHI Grant Numbers JP16K00031 and JP17H01732, and by Support Center for Advanced Telecommunications Technology Research (SCAT).

## References

- [1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.
- [2] R. Bhatia. *Positive definite matrices*. Princeton university press, 2009.
- [3] S. Bonnabel and R. Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, 2010.
- [4] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(Apr):1455–1459, 2014.
- [5] Nicolas Boumal. An introduction to optimization on smooth manifolds. Available online, Aug 2020. URL <http://www.nicolasboumal.net/book>.
- [6] S. Giguere, F. Garcia, and S. Mahadevan. A manifold approach to learning mutually orthogonal subspaces. Technical report, arXiv preprint arXiv:1703.02992, 2017.
- [7] Ryo Inokuchi and Sadaaki Miyamoto. C-means clustering on the multinomial manifold. In *International Conference on Modeling Decisions for Artificial Intelligence (MDAI)*, 2007.
- [8] T.-K. Kim, J. Kittler, and R. Cipolla. On-line learning of mutually orthogonal subspaces for face recognition by image sets. *IEEE Transactions on Image Processing*, 19(4):1067–1074, 2009.
- [9] G. Lebanon and J. Lafferty. Hyperplane margin classifiers on the multinomial manifold. In *International conference on Machine learning (ICML)*, 2004.

- [10] K. L. Lee, J. Shang, W. K. Chua, S. Y. Looi, and B.-G. Englert. SOMIM: An open-source program code for the numerical search for optimal measurements by an iterative method. Technical report, arXiv preprint arXiv:0805.2847, 2008.
- [11] B. Mishra and R. Sepulchre. Riemannian preconditioning. *SIAM Journal on Optimization*, 26(1):635–660, 2016.
- [12] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9(Nov):2491–2521, 2008.
- [13] J. Řeháček, B.-G. Englert, and D. Kaszlikowski. Iterative procedure for computing accessible information in quantum communication. *Physical Review A*, 71(5):054303, 2005.
- [14] S. Sra and R. Hosseini. Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization*, 25(1):713–739, 2015.
- [15] Y. Sun, J. Gao, X. Hong, B. Mishra, and B. Yin. Heterogeneous tensor decomposition for clustering via manifold optimization. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):476–489, 2015.
- [16] D. Zhang, X. Chen, and W. S. Lee. Text classification with kernels on the multinomial manifold. In *International ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, 2005.