LEAST SQUARES APPROXIMATE JOINT DIAGONALIZATION ON THE ORTHOGONAL GROUP

Toshihisa Tanaka

Dept. of Electrical and Electronic Eng.,
Tokyo Univ. of Agriculture & Technology
184–8588 Tokyo, Japan.
tanakat@cc.tuat.ac.jp

Simone Fiori

DEIT, Facoltà di Ingegneria
Università Politecnica delle Marche
I-60131 Ancona, Italy.
fiori@deit.univpm.it

ABSTRACT

The theory and derivation of a novel method for approximate joint diagonalization (AJD) on the orthogonal group of matrices are presented. The proposed algorithms are fast and simple, hence, easy to implement. We introduce a least-squares-type cost function, which is to be minimized under the constraint that the matrix to be sought for is orthogonal. A gradient flow for optimizing such cost function is derived and its stability is analyzed within the framework of differential geometry. It is proposed to numerically approximate the gradient flow by using a geodesic-based and an Euler-like update algorithms. Numerical examples about blind source separation of speech signals are illustrated to support the analysis.

Index Terms— Adaptive learning, gradient flow, blind source separation, joint diagonalization, orthogonal group

1. INTRODUCTION

Approximate joint diagonalization (AJD) is an extensively used technique in blind source separation [1], which is the problem of recovering source signals only from their observed mixtures without any knowledge except the assumption of the source signals to be mutually statistically independent or uncorrelated. The underlying idea behind the AJD technique in BSS is that the assumption of independence implies that the correlation matrix and cross-correlation matrices with time delay become diagonal when the separation is successful [1]. Alternatively, if high-order statistical tensors of source signals, such as joint-cumulant matrices, are considered, they should diagonalize upon separation [1].

Consider a set $\mathcal{A}$ of K symmetric matrices of size $N \times N$, $\mathcal{A} = \{\boldsymbol{A}_i \in \mathbb{R}^{N\times N}\}_{i=1}^K$. In the framework of BSS, the $\boldsymbol{A}_i$'s can be fourth-order joint-cumulant matrices or (second-order) cross-correlation matrices. The AJD problem formulates as: seek a unitary or non-singular $N \times N$ matrix $\boldsymbol{U}$ that jointly diagonalizes all the matrices in $\mathcal{A}$. Several formulations to solve the AJD problem have been proposed. A pioneer work on this topic was conducted by Cardoso [2], and the cost function to be minimized is

$$J_{\mathrm{JADE}}[\boldsymbol{U}] = \sum_{i=1}^{K} \mathit{off}(\boldsymbol{U}\boldsymbol{A}_i\boldsymbol{U}^T) \tag{1}$$

where $\mathit{off}(\cdot)$ is the sum of the squares of the non-diagonal entries and $\boldsymbol{U}$ is assumed to be orthogonal. The minimizing matrix, $\boldsymbol{U}$, is parameterized by the so-called Jacobi angles [2], thus the optimal solution is sought for every single angle. The corresponding BSS algorithm is known as JADE. Yeredor [3] proposed another optimization criterion that exhibits superior performance in separation to JADE. In this method, the cost function is defined as

$$J_{\mathrm{LS}}[\boldsymbol{U}, \{\boldsymbol{\Lambda}_i\}_{i=1}^K\}] = \sum_{i=1}^{K} w_i \|\boldsymbol{A}_i - \boldsymbol{U}\boldsymbol{\Lambda}_i\boldsymbol{U}^H\|_F^2 \tag{2}$$

where $\boldsymbol{U}$ is not necessarily orthogonal but in $\mathbb{C}^{N\times N}$, the w_i's are positive weights (scalars), the $\boldsymbol{\Lambda}_i$'s are diagonal matrices in $\mathbb{C}^{N\times N}$, and $\|\cdot\|_F$ is the Frobenius norm. In this optimization problem, not only $\boldsymbol{U}$ but also $\boldsymbol{\Lambda}_i$ ($i = 1, \ldots, K$) should be found. It is worth pointing out that to accomplish the diagonalization, a large number of parameters, namely, $\boldsymbol{U}$ and $\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_K$ should be jointly optimized. Moreover, since matrix $\boldsymbol{U}$ is not bounded, the seeking space is not compact.

This paper proposes to impose the orthogonal constraint on $\boldsymbol{U}$ and optimize it on the *manifold of orthogonal matrices*, that is, the *orthogonal group*. We apply a differential geometrical approach to optimizing $\boldsymbol{U}$. We first review the geometry of the orthogonal group and of gradient-based optimization on manifold in Section 2.1. The dynamics of matrix $\boldsymbol{U}$ is given as a gradient flow on the orthogonal group of matrices, and the related adaptive updating formula is derived. As it will be clarified in Section 2.2, such an approach implies the advantage that no optimization of matrices $\boldsymbol{\Lambda}_i$ is needed, namely, the only parameter to be updated is $\boldsymbol{U}$. The stability of the algorithm is then analyzed and numerical updating rules are given in Section 2.4. Numerical examples of speech mixtures are discussed in order to demonstrate the effectiveness of the present proposal in Section 3. The obtained AJD algorithm proves to be simple in implementation and faster than the well-known Jacobi algorithm.

2. DIFFERENTIAL GEOMETRICAL DERIVATION OF OPTIMIZATION ALGORITHMS

2.1. Riemannian Gradient Flow on the Orthogonal Group

We review in this section how to derive ordinary differential equations (ODEs) that generate gradient flows to solve optimization problems in the framework of differential geometry, which is the calculus of manifolds.

Let $T_\xi\mathcal{M}$ be the tangent space to a smooth Riemannian manifold $(\mathcal{M}, g^{\mathcal{M}})$ in point $\xi \in \mathcal{M}$, where $g_\xi^{\mathcal{M}} : T_\xi\mathcal{M} \times T_\xi\mathcal{M} \to \mathbb{R}$ denotes a bilinear scalar product that turns $\mathcal{M}$ into a metric space. In particular, the Euclidean scalar product denoted by $g^e : T_\xi\mathcal{M} \times T_\xi\mathcal{M} \to \mathbb{R}$ is necessary to define the gradient on a Riemannian manifold. The gradient, $\mathrm{grad}_\xi^{\mathcal{M}} f$, of a differentiable function, $f : \mathcal{M} \to \mathbb{R}$, in ξ on $(\mathcal{M}, g^{\mathcal{M}})$ is defined by the following two conditions [4]:

Toshihisa Tanaka is also with the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute.

1. $\text{grad}_\xi^{\mathcal{M}} f \in T_\xi \mathcal{M}$ (tangency condition)
2. $g_\xi^{\mathcal{M}}\left(\text{grad}_\xi^{\mathcal{M}} f, \boldsymbol{v}\right) = g^e\left(\dfrac{\partial f}{\partial \xi}, \boldsymbol{v}\right)$ for all $\boldsymbol{v} \in T_\xi \mathcal{M}$ (compatibility condition),

where $\dfrac{\partial f}{\partial \xi}$ is the standard gradient (or Jacobian) of f.

By gradient-based differential equation for the constrained optimization of function $f : \mathcal{M} \to \mathbb{R}$ on $\mathcal{M}$, where the smooth manifold, $\mathcal{M}$, fully describes the constraints, it is meant:

$$\dot{\xi}(t) = \pm \text{grad}_\xi^{\mathcal{M}} f(\xi(t)), \;\; \xi(0) = \xi_0 \in \mathcal{M}, \tag{3}$$

where the positive sign denotes maximization and the negative sign denotes minimization of f.

In the present paper, we make use of a special smooth manifold, which is the group of orthogonal matrices:

$$O(N) = \{\boldsymbol{U} \in \mathbb{R}^{N\times N} | \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_N\}. \tag{4}$$

This manifold is termed *orthogonal group*. The tangent spaces of the orthogonal group have structure

$$T_{\boldsymbol{U}} O(N) = \{\boldsymbol{U}\boldsymbol{H} \in \mathbb{R}^{N\times N} | \boldsymbol{H}^T = -\boldsymbol{H}\}. \tag{5}$$

In $O(N)$, when the metric is selected to be

$$g_{\boldsymbol{U}}^{O(N)}(\boldsymbol{U}\boldsymbol{H}_1, \boldsymbol{U}\boldsymbol{H}_2) = \text{tr}[\boldsymbol{H}_1^T \boldsymbol{H}_2], \;\; \forall \boldsymbol{H}_1, \boldsymbol{H}_2 \in T_{\boldsymbol{U}} O(N), \tag{6}$$

the Riemannian gradient satisfying the tangency and compatibility conditions is described as

$$\text{grad}_{\boldsymbol{U}}^{O(N)} J = \frac{\partial J}{\partial \boldsymbol{U}} - \boldsymbol{U}\left(\frac{\partial J}{\partial \boldsymbol{U}}\right)^T \boldsymbol{U}. \tag{7}$$

In summary, if we have a differentiable function from $\mathcal{M}$ to $\mathbb{R}$, which is commonly a cost function in optimization, we can obtain the Riemannian gradient with respect to the pre-defined metric by calculating simply the Jacobian. Details of the development of learning algorithms on the orthogonal group can be found in [5], for example.

2.2. Derivation of AJD Flow

Following the theory reviewed in the previous subsection, we derive AJD algorithms in the following. Motivated by Yeredor [3], the cost function to be analyzed in this paper is given by

$$J[\boldsymbol{U}, \{\boldsymbol{\Lambda}_i\}_{i=1}^K] = \sum_{i=1}^K \|\boldsymbol{A}_i - \boldsymbol{U}\boldsymbol{\Lambda}_i \boldsymbol{U}^T\|_F^2, \tag{8}$$

where $\boldsymbol{U} \in O(N)$ and $\{\boldsymbol{\Lambda}_i\}_{i=1}^K$ is a set of diagonal matrices of size $N \times N$. Unlike in (2), we omit the weight coefficients; however, it is easy to generalize the following theory to the case of J being the weighted sum.

We could directly derive a "joint gradient flow" for both $\boldsymbol{U}$ and $\boldsymbol{\Lambda}_i$. Then, we would have $K + 1$ gradient flows, which would lead to a quite burdensome algorithm. As it will be seen later, however, we do not take this joint gradient approach. It is possible to derive a single gradient flow only for $\boldsymbol{U}$, while closed form expressions for the step-by-step optimal value of matrices $\boldsymbol{\Lambda}_i$ in terms of the current value of matrix $\boldsymbol{U}$ and of known matrices $\boldsymbol{A}_i$ may be derived. Under the assumption that $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, the cost function J rewrites

$$J[\boldsymbol{U}, \{\boldsymbol{\Lambda}_i\}_{i=1}^K] = \sum_{i=1}^K \text{tr}[\boldsymbol{A}_i^2 - 2\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U}\boldsymbol{\Lambda}_i + \boldsymbol{\Lambda}_i^2], \tag{9}$$

since $\boldsymbol{A}_i$ is assumed to be symmetric. The Jacobian with respect to matrix $\boldsymbol{U}$ is obtained as

$$\frac{\partial J}{\partial \boldsymbol{U}} \propto -\sum_{i=1}^K \boldsymbol{A}_i \boldsymbol{U}\boldsymbol{\Lambda}_i. \tag{10}$$

By plugging (10) into (7), we obtain the Riemannian gradient of the cost function J with respect to $\boldsymbol{U}$ as

$$\text{grad}_{\boldsymbol{U}}^{O(N)} J = \sum_{i=1}^K (\boldsymbol{U}\boldsymbol{\Lambda}_i \boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U} - \boldsymbol{A}_i \boldsymbol{U}\boldsymbol{\Lambda}_i). \tag{11}$$

Prior to obtaining the gradient flow for $\boldsymbol{U}$, we show that for a fixed $\boldsymbol{U}$, the matrices, $\boldsymbol{\Lambda}_i$, that minimize J can be obtained in closed form. In fact, the cost function (9) may be advantageously recast as

$$\begin{aligned} J[\boldsymbol{U}, \{\boldsymbol{\Lambda}_i\}_{i=1}^K] &= \sum_{i=1}^K \left(\text{tr}[\boldsymbol{A}_i^2] - \sum_{j=1}^N \left\{ 2(\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U})_{jj} \lambda_{i,j} + \lambda_{i,j}^2 \right\} \right) \\ &= \sum_{i=1}^K \sum_{j=1}^N \left(\lambda_{i,j} - (\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U})_{jj} \right)^2 + J_0, \end{aligned} \tag{12}$$

where $\lambda_{i,j}$ is the j^{th} diagonal element of matrix $\boldsymbol{\Lambda}_i$, symbol $(\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U})_{jj}$ denotes the $(j, j)^{\text{th}}$ entry of matrix $\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U}$, and J_0 is the residual part, independent of matrices $\boldsymbol{\Lambda}_i$. This implies that for a fixed $\boldsymbol{U}$, the cost function J is minimized when

$$\boldsymbol{\Lambda}_i = \text{diag}(\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U}), \tag{13}$$

where $\text{diag}(\cdot)$ is the diagonalizing operator that keeps all diagonal elements and sets the off-diagonal entries to zero. It therefore follows from (3), (11), and (13) that the dynamics of $\boldsymbol{U}$ is obtained as follows:

$$\dot{\boldsymbol{U}} = -\sum_{i=1}^K (\boldsymbol{U}\text{diag}(\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U})\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U} - \boldsymbol{A}_i \boldsymbol{U}\text{diag}(\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U})). \tag{14}$$

2.3. Stability Analysis

We first show that the matrix product $\boldsymbol{B} = \boldsymbol{U}^T\boldsymbol{U}$ is an invariant of the dynamical system (14). in the dynamics.

Theorem 1 *$\boldsymbol{B}(t) = \boldsymbol{U}^T(t)\boldsymbol{U}(t)$ is an invariant of the dynamics generated by the differential equation (14) if the initial point $\boldsymbol{U}(0)$ is on $O(N)$.*

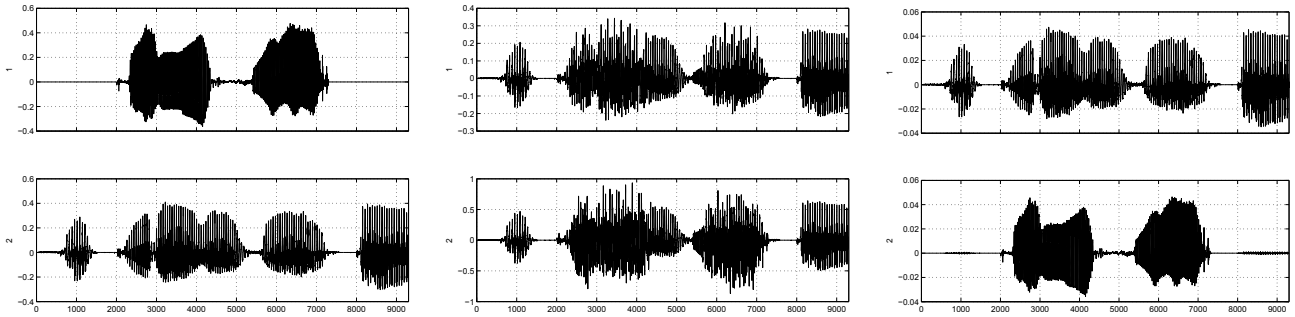
Proof. By the direct differentiation, we obtain

$$\begin{aligned} \dot{\boldsymbol{B}} = \dot{\boldsymbol{U}}^T\boldsymbol{U} + \boldsymbol{U}^T\dot{\boldsymbol{U}} = &\sum_{i=1}^K \boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U}\text{diag}(\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U})(\boldsymbol{I} - \boldsymbol{U}^T\boldsymbol{U}) \\ &+ (\boldsymbol{I} - \boldsymbol{U}^T\boldsymbol{U}) \sum_{i=1}^K \text{diag}(\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U})\boldsymbol{U}^T \boldsymbol{A}_i \boldsymbol{U}, \end{aligned} \tag{15}$$

which implies that if $\boldsymbol{B}(0) = \boldsymbol{I}_N$, then $\boldsymbol{B}(t) = \boldsymbol{I}_N$ always.

A further analysis regards the asymptotic stability of the base manifold. Such result ensures that, under suitable conditions, the base manifold $O(N)$ is an attractor for the differential equation (14) under small perturbations. Such result is relevant in the case that the differential equation (14) is integrated by a inexact Euler method.

Theorem 2 *If matrices $\boldsymbol{A}_i$ are positive definite, then the manifold $O(N)$ is asymptotically stable for the dynamical system (14).*



(a) Original (b) Mixture (c) Separated

Fig. 1: Joint diagonalization of speech mixtures

Brief proof. To linearize (15) around a point $\boldsymbol{U}_0 \in O(N)$, let $\boldsymbol{U} = \boldsymbol{U}_0 + \boldsymbol{E}$, where $\boldsymbol{E}$ is a small perturbation. Then, we obtain

$$\boldsymbol{B} \approx \boldsymbol{I} + \boldsymbol{F} \tag{16}$$

where $\boldsymbol{F} = \boldsymbol{U}_0^T \boldsymbol{E} + \boldsymbol{E}^T \boldsymbol{U}_0$. On the other hand, the right hand side of (15) becomes

$$\dot{\boldsymbol{U}}^T \boldsymbol{U} + \boldsymbol{U}^T \dot{\boldsymbol{U}} = -\sum_{i=1}^{K} (\boldsymbol{H}_i \boldsymbol{\Lambda}_i \boldsymbol{F} + \boldsymbol{F} \boldsymbol{\Lambda}_i \boldsymbol{H}_i) \tag{17}$$

where $\boldsymbol{H}_i = \boldsymbol{U}_0^T \boldsymbol{A}_i \boldsymbol{U}_0$. Therefore, we have a linear ordinary differential equation with respect to $\boldsymbol{F}$:

$$\dot{\boldsymbol{F}} = -\sum_{i=1}^{K} (\boldsymbol{H}_i \boldsymbol{\Lambda}_i \boldsymbol{F} + \boldsymbol{F} \boldsymbol{\Lambda}_i \boldsymbol{H}_i). \tag{18}$$

Considering $\mathrm{tr}[\boldsymbol{F}\boldsymbol{F}^T]$, which is a Lyapunov function, we obtain

$$\frac{d}{dt}\mathrm{tr}[\boldsymbol{F}\boldsymbol{F}^T] = -4 \sum_{i=1}^{K} \mathrm{tr}\left[\boldsymbol{F} \boldsymbol{\Lambda}_i^{1/2} \boldsymbol{H}_i \boldsymbol{\Lambda}_i^{1/2} \boldsymbol{F}\right] < 0, \tag{19}$$

since the positive definiteness of $\boldsymbol{\Lambda}_i^{1/2} \boldsymbol{H}_i \boldsymbol{\Lambda}_i^{1/2}$. Therefore, it holds that $\mathrm{tr}[\boldsymbol{F}\boldsymbol{F}^T] \to 0$ as $t \to \infty$, which implies that $\boldsymbol{E} \to \boldsymbol{0}$.

2.4. Numerical Integration of the ODE

In order to design an effective adapting algorithm, it is necessary to develop a suitable numerical integration method for numerically solving the differential equation (14). A suitable numerical integration is a method based on the concept of *geodesics*, A geodesic is a counterpart of a straight line of flat surface on a curved space. See details of geodesics in [5, 6], for example. Since a geodesic is a curve on the manifold, a stable numerical integration can be performed [5]. In the case of the orthogonal group endowed with the standard bi-invariant canonical metric (6), a geodesic $\gamma(t) : [0, 1] \to O(N)$ with the initial conditions, $\gamma(0) = \boldsymbol{U}$ and $\dot{\gamma}(0) = -\mathrm{grad}_{\boldsymbol{U}}^{O(N)} J$, can be obtained in the closed form as

$$\gamma_{\boldsymbol{U}, -\mathrm{grad}_{\boldsymbol{U}}^{O(N)} J}^{O(N)}(t) = \left(\exp\left[-t\left(\left(\frac{\partial J}{\partial \boldsymbol{U}}\right)\boldsymbol{U}^T - \boldsymbol{U}\left(\frac{\partial J}{\partial \boldsymbol{U}}\right)^T\right)\right]\right)\boldsymbol{U}, \tag{20}$$

where $\exp(\cdot)$ denotes the matrix exponential.

Together with the geodesic formula, equations (11) and (13), we obtain the **geodesic-based AJD algorithm**:

$$\boldsymbol{U}(n+1) = \exp\left(\beta \sum_{i=1}^{K} (\boldsymbol{A}_i \boldsymbol{U}(n) \boldsymbol{\Lambda}_i(n) \boldsymbol{U}^T(n) - \boldsymbol{U}(n) \boldsymbol{\Lambda}_i(n) \boldsymbol{U}^T(n)) \boldsymbol{A}_i\right) \boldsymbol{U}(n), \tag{21}$$

where $\boldsymbol{\Lambda}_i(n) = \mathrm{diag}(\boldsymbol{U}^T(n) \boldsymbol{A}_i \boldsymbol{U}(n))$ and β is an appropriate stepsize. Also, by employing the Euler update formula, we obtain the **Euler-like AJD algorithm**:

$$\boldsymbol{U}(n+1) = \boldsymbol{U}(n) - \beta \sum_{k=1}^{K} [\boldsymbol{U}(n) \boldsymbol{\Lambda}_i(n) \boldsymbol{U}^T(n) \boldsymbol{A}_i \boldsymbol{U}(n) - \boldsymbol{A}_i \boldsymbol{U}(n) \boldsymbol{\Lambda}_i(n)]. \tag{22}$$

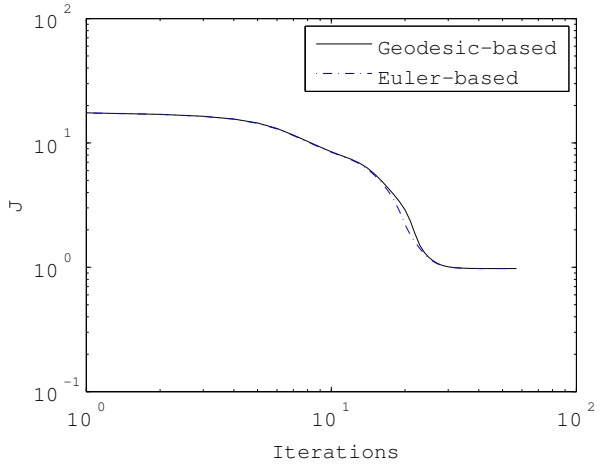
3. NUMERICAL EXAMPLES

A comparison study is conducted to understand the behavior of the proposed algorithms and confirm the advantage over algorithm [2]. In all the discussed tests, observed mixtures are pre-whitened, and matrices $\boldsymbol{A}_i$ are computed as second-order correlation matrices with time delay. The algorithms were implemented in MATLAB and run on an Intel Core Duo processor.

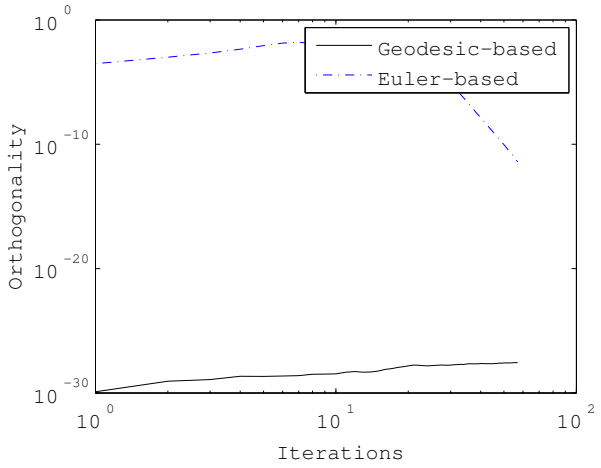
Figure 1 illustrates the results of separation obtained by using the proposed geodesic-based algorithm. In the experiment the figure refers to, two channel speech signals were mixed by a randomly generated mixing matrix. As seen in the figure, the separation ability of geodesic-based AJD is very good.

Figures 2a and 2b compare the behavior of two proposed algorithms. It can be observed in Fig. 2a that both algorithms decrease the value of the cost function $J(n)$ similarly. Also, they converge with a small number of iterations. Figure 2b shows the result of evaluation of the orthogonality of matrix $\boldsymbol{U}$ during learning. The orthogonality index is defined as $\eta(n) = \|\boldsymbol{U}^T(n)\boldsymbol{U}(n) - \boldsymbol{I}_N\|_F^2$. From the figure, it can be seen that the geodesic-based algorithm strictly preserves orthogonality. However, the Euler-like integration preserves the orthogonality in a satisfactory way as well.

Table 1 lists results of blind source separation of speech mixtures. Twenty speech signals, each of 3,500 samples, were mixed by a randomly generated mixing matrix. We assume that the number of observations is the same as that of the original signals. Separation by three algorithms (the proposed geodesic-based, the proposed Euler-like and the one based on Jacobi angles parameterization) was performed 1,000 times and performance index (PI) [1] and processing time were averaged over all the independent trials. In the pro-



(a) Criterion $J(n)$



(b) Orthogonality $\eta(n)$

Fig. 2: Behavior of the adaptive algorithms

posed algorithm, β was chosen to be 0.15, and the stopping criterion was tuned in such a way that three algorithms provides similar PIs. The same simulation was conducted for the noiseless case as well as in the noisy case with three different signal-to-noise ratio (SNR) values. From the table, it emerges that the proposed AJD algorithms are faster than the Jacobi iteration algorithm in all the cases. An interesting observation is about computational times of the Euler-like and the geodesic-based integration, which are very similar. In all the cases, the performance of the Jacobi iteration algorithm is inferior to the others in terms of computation time.

4. CONCLUSION

A novel algorithm for approximate joint diagonalization (AJD), which is derived from a gradient flow on the orthogonal group, has been proposed. The updating rules exhibit a simple structure and are easy to implement. It has theoretically been shown how to approximate the gradient flow by piece-wise geodesic arcs as well as that simpler Euler-like integration is feasible, provided the stepsize is sufficiently

Table 1: Blind source separation results of speech mixtures

(a) Noiseless

Algorithm	Averaged PI	Averaged Time [s]
Geodesic	0.011602	0.506958
Euler	0.011623	0.535631
Cardoso [2]	0.011603	0.844707

(b) SNR 20dB

Algorithm	Averaged PI	Averaged Time [s]
Geodesic	0.019701	0.647791
Euler	0.019711	0.684150
Cardoso [2]	0.019784	0.974539

(c) SNR 10dB

Algorithm	Averaged PI	Averaged Time [s]
Geodesic	0.031210	0.881047
Euler	0.031199	0.940432
Cardoso [2]	0.031550	1.202101

(d) SNR 5dB

Algorithm	Averaged PI	Averaged Time [s]
Geodesic	0.040525	1.177167
Euler	0.040532	1.270140
Cardoso [2]	0.041033	1.429610

small, as the orthogonal group is an attractor of the optimization dynamics. Experiments on BSS were illustrated: The proposed algorithms were found to generate faster convergence and better performances in separation than the Jacobi angle iteration. Although the proposed algorithm is efficient in the application of BSS, there would be some open problems. First, the update parameter β has to be handcrafted: Finding the optimal parameter would lead to a faster convergence. Second, it has been pointed out in [7] that other maps from $TO(N)$ to $O(N)$, such as the Cayley transform, are known. We should investigate the behavior of the algorithm where these maps are used instead of the geodesic-based one.

5. REFERENCES

[1] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithmsand Applications*. England: John Wiley & Sons, 2002.

[2] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 17, pp. 161–164, Jan. 1996.

[3] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. Signal Processing*, vol. 50, pp. 1545–1553, July 2002.

[4] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. London, UK: Springer-Verlag, 1994.

[5] S. Fiori, "Quasi-geodesic neural learning algorithms over the orthogonal group: A tutorial," *Journal of Machine Learning Research*, vol. 6, pp. 743–781, 2005.

[6] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. and App.*, vol. 20, no. 2, pp. 303–353, 1998.

[7] S. Fiori, "Neural learning by retractions on manifolds," in *Proc. of 2007 IEEE Int. Symp. Circuits and Systems (ISCAS 2007)*, (New Orleans, USA), May 2007. accepted.