Journal of Machine Learning Research 6 (2005) 743–781 Submitted 5/04; Revised 10/04; Published 5/05

Quasi-Geodesic Neural Learning Algorithms Over the Orthogonal Group: A Tutorial

Simone Fiori FIORI@UNIPG.IT
Facoltà di Ingegneria, Università di Perugia
Polo Didattico e Scientifico del Ternano
Località Pentima bassa, 21, I-05100 Terni, Italy

Editor: Yoshua Bengio

Abstract

The aim of this contribution is to present a tutorial on learning algorithms for a single neural layer whose connection matrix belongs to the orthogonal group. The algorithms exploit geodesics appropriately connected as piece-wise approximate integrals of the exact differential learning equation. The considered learning equations essentially arise from the Riemannian-gradient-based optimization theory with deterministic and diffusion-type gradient. The paper aims specifically at reviewing the relevant mathematics (and at presenting it in as much transparent way as possible in order to make it accessible to readers that do not possess a background in differential geometry), at bringing together modern optimization methods on manifolds and at comparing the different algorithms on a common machine learning problem. As a numerical case-study, we consider an application to non-negative independent component analysis, although it should be recognized that Riemannian gradient methods give rise to general-purpose algorithms, by no means limited to ICA-related applications.

Keywords: differential geometry, diffusion-type gradient, Lie groups, non-negative independent component analysis, Riemannian gradient

1. Introduction

From the scientific literature, it is known that a class of learning algorithms for artificial neural networks may be formulated in terms of matrix-type differential equations of network's learnable parameters, which give rise to learning flows on parameters' set. Often, such differential equations are defined over parameter spaces that may be endowed with a specific geometry, such as the general linear group, the compact Stiefel manifold, the orthogonal group, the Grassman manifold and the manifold of FIR filters[1] (Amari, 1998; Fiori, 2001, 2002; Liu et al., 2004; Zhang et al., 2002), that describes the constraints that the network parameters should fulfill and that is worth taking into account properly. From a practical viewpoint, the mentioned differential equations should be integrated (solved) properly through an appropriate numerical integration method that allows us to preserve the underlying structure (up to reasonable precision). This may be viewed as defining a suitable discretization method in the time domain that allows converting a differential learning equation into a discrete-time algorithm.

[1] Roughly speaking, the manifold of FIR filters may be regarded as the set of rectangular matrices whose entries are polynomials in a complex-valued variable.

With the present contribution, we aim at studying and illustrating learning algorithms for a single neural layer whose connection matrix belongs to the orthogonal group, that is the group of square orthogonal matrices. As an appropriate approximation of the exact learning flows, the algorithms exploit approximate geodesics suitably glued together, as formerly proposed by Fiori (2002) and Nishimori (1999).

As a case-study, we consider an application to geodesic-learning-based non-negative independent component analysis, as proposed by Plumbley (2003). We present three different learning algorithms that are based on gradient-type optimization of a non-negative independent component analysis criterion over the group of orthogonal matrices. The first two algorithms arise from the direct application of Riemannian gradient optimization without and with geodesic line search, as proposed by Plumbley (2003). The third algorithm relies on a randomized gradient optimization based on diffusion-type Riemannian gradient, as proposed by Liu et al. (2004).

The contribution of the present tutorial may be summarized via the following key points:

- It provides a clear and well-motivated introduction to the mathematics needed to present the geometry-based learning algorithms.
- It clearly states and illustrates the idea that, when we wish to implement a gradient-based algorithm on a computer, it is necessary to discretize the differential learning equations in some suitable way (the ‘gradient flow’ simply cannot be computed exactly in practice).
- In order to effect such discretization, we may not employ standard discretization methods (such as the ones based on Euler forward-backward discretization), that do not work as they stand on curved manifolds. We should therefore resort to more sophisticated integration techniques such as the one based on geodesics.
- In order to improve the numerical performances of the learning algorithm, we might tentatively try adding some stochasticity to the standard gradient (through annealed-MCMC method) and try a geodesic search. It is not guaranteed that the above-mentioned improvement works on a concrete application, therefore it is worth testing them on ICA$^+$ problem. The results on this sides are so far disappointing, because numerical simulations shown that standard Riemannian gradient with no geodesic search nor stochasticity added outperforms the other methods on the considered ICA$^+$ problem.

Although in the machine learning community the presented differential geometry-based learning algorithms have so far been primarily invoked in narrow contexts such as principal/independent component analysis (interested readers might want to consult, for example, Fiori (2001), Celledoni and Fiori (2004) and Plumbley (2003) for a wide review), it should be recognized that differential-geometrical methods provide a general-purpose way of designing learning algorithms, which is profitable in those cases where a learning problem may be formulated mathematically as an optimization problem over a smooth manifold. Some recent advances and applications of these methods are going to be described in the journal special issue whose content is summarized in the editorial by Fiori and Amari (2005).

The paper is organized as follows. The purpose of Section 2 is to briefly recall some concepts from algebra and differential geometry, which are instrumental in the development of the presented learning algorithms. In particular, the concepts of algebraic groups, differential manifolds and Lie groups are recalled, along with the concepts of right-translation, Riemannian gradient and geodesic curves. Then, these results are customized to the case of the orthogonal group of concern in the present paper. Geodesic-based approximations of gradient-type learning differential equations over the orthogonal group are also explained. The Section 2 also presents some notes on the stability of such learning equations as well as on the relationship between the presented learning theory and the well-known natural-gradient theory and information geometry theory. Section 3 presents two deterministic-gradient learning algorithms, one of which is based on the optimization of the learning stepsize via 'geodesic search'. Next, the concept of diffusion-type gradient on manifolds is recalled in details and a third learning algorithm based on it is presented. Such learning algorithm also takes advantage of simulated annealing optimization technique combined with Markov-Chain Monte-Carlo sampling method, which are also recalled in the Section 3, along with some of their salient features. Section 4 deals with non-negative independent component analysis. Its definition and main properties are recalled and the orthogonal-group Riemannian-gradient of the associated cost function is computed. Such computation allows customizing the three generic Riemannian-gradient-based geodesic algorithms to the non-negative independent component analysis case. Also, a fourth projection-based algorithm is presented for numerical comparison purpose. The details of algorithms implementation and the results of computer-based experiments performed on non-negative independent component analysis of gray-level image mixtures are also illustrated in the Section 4. Section 5 concludes the paper.

2. Learning Over the Orthogonal Group: Gradient-Based Differential Systems and Their Integration

The aims of the present section are to recall some basic concepts from differential geometry and to derive the general form of gradient-based learning differential equations over the orthogonal group. We also discuss the fundamental issue of solving numerically such learning differential equations in order to obtain a suitable learning algorithm.

2.1 Basic Differential Geometry Preliminaries

In order to better explain the subsequent issues, it would be beneficial to recall some basic concepts from differential geometry related to the orthogonal group $O(p)$.

An algebraic group (G,m,i,e) is a set G that is endowed with an internal operation $m: G\times G\rightarrow G$, usually referred to as group multiplication, an inverse operation $i: G\rightarrow G$, and an identity element e with respect to the group multiplication. These objects are related in the following way. For every elements $x,y,z\in G$, it holds that

$$m(x,i(x)) = m(i(x),x) = e,\ m(x,e) = m(e,x) = x$$
$$\text{and } m(x,m(y,z)) = m(m(x,y),z).$$

Note that, in general, the group multiplication is not commutative, that is, given two elements $x,y\in G$, it holds $m(x,y)\neq m(y,x)$.

Two examples of algebraic groups are $(\mathbb{Z},+,-,0)$ and $(Gl(p),\cdot,^{-1},\mathbf{I}_p)$. The first group is the set of all integer numbers endowed with the standard addition as group multiplication. In this case, the inverse is the subtraction operation and the identity is the null element. In the second example, we considered the set of non-singular matrices:

$$Gl(p)\overset{\text{def}}{=}\{\mathbf{X}\in\mathbb{R}^{p\times p}\,|\,\det(\mathbf{X})\neq 0\},\tag{1}$$

endowed with standard matrix multiplication '$\cdot$' as group multiplication operation. In this case, the inverse is the standard matrix inverse and the identity is the identity matrix $\mathbf{I}_p$. It is easy to show that both groups operations/identity satisfy the above general conditions. As a counterexample, the set of the non-negative integer numbers $\mathbb{Z}_0^+ (\equiv \mathbb{N})$ does not form a group under standard addition/subtraction. A remarkable difference between the two groups above is that the first one is a discrete group while the second one is a continuous group.

A useful concept for the economy of the paper is the one of differential manifold. The formal definition of a differential manifold is quite involved, because it requires precise definitions from mathematical topology theory and advanced calculus (Olver, 2003). More practically, a manifold may be essentially regarded as a generalization of curves and surfaces in high-dimensional space, that is endowed with the noticeable property of being locally similar to a flat (Euclidean) space. Let us consider a differential manifold $\mathcal{M}$ and a point ξ on it. From an abstract point of view, ξ is an element of a set $\mathcal{M}$ and does not necessarily possess any particular numerical feature. In order to be able to make computations on manifolds, it is convenient to 'coordinatize' it. To this aim, a neighborhood (open set) $U \subset \mathcal{M}$ is considered, which ξ belongs to, and a coordinate map $\psi : U \to \mathcal{E}$ is defined, where $\mathcal{E}$ denotes a Euclidean space (as for example, $\mathbb{R}^p$ – the set of p-dimensional real-valued vectors – or $\mathbb{R}^{p\times p}$ – the set of the $p \times p$ real-valued matrices –). The function ψ needs to be a one-to-one map (homeomorphism). In this way, we attach a coordinate $x = \psi(\xi)$ to the point ξ. As ψ is a homeomorphism, there is a one-to-one correspondence between a point on a manifold and its corresponding coordinate-point, therefore normally the two concepts may be confused and we may safely speak of a point $x \in \mathcal{M}$. About these concepts, two short notes are in order:

- Borrowing terms from maritime terminology, a triple (ψ, U, p) is termed *coordinate chart* associated to the manifold $\mathcal{M}$. Such notation evidences that the elements ψ and $U \subset \mathcal{M}$ are necessary to coordinatize a point on the manifold and that the coordinate space has dimension p. If the dimension is clear from the context, the indication of p may of course be dispensed of.

- A main concept of differential geometry is that *every geometrical property is independent of the choice of the coordinate system*. As a safety note, it is important to remark that, when we choose to express geometrical relationships in coordinates (as it is implicitly assumed by the above-mentioned 'confusion' between a point $\xi \in \mathcal{M}$ and its coordinate $x \in \mathcal{E}$) we are by no means abandoning this fundamental principle, but we are obeying to the practical need of algorithm implementation on a computer that requires – of necessity – some explicit representation of the quantities of interest.

In general, it is impossible to cover a whole manifold with a unique coordinate map. Therefore, the procedure for coordinatizing a manifold generally consists in covering it with a convenient number of neighborhoods U_k, each of which is endowed with a coordinate map $\psi_k : U_k \to \mathcal{E}_k$, with $\mathcal{E}_k$ being an Euclidean space of dimension p, which, by definition, denotes the dimension of the manifold itself. Technically, the set $\{U_k\}$ is termed a *basis* for the manifold and it does not need to be finite (but it is inherently countable). It is important to note that, in general, the neighborhoods U_k may be overlapping. In this case, the maps ψ_k need to satisfy some constraints termed 'compatibility conditions' which formalize the natural requirement that there should be a one-to-one smooth correspondence between any two different coordinate systems. Technically, if $U_k \cap U_h \neq \emptyset$ then the maps $\psi_k^{-1} \circ \psi_h$ and $\psi_h^{-1} \circ \psi_k$, which are termed 'transition functions' and represent coordinate changes, should be diffeomorphisms, that is, C^∞ functions endowed with C^∞ inverse.

A smooth manifold is by nature a continuous object. A simple example is the unit hyper-sphere $S^p \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}^{p+1} | \mathbf{x}^T \mathbf{x} = 1\}$. This is a smooth manifold of dimension p embedded in the Euclidean space $\mathbb{R}^{p+1}$, in fact with only p coordinates we can identify any point on the sphere. Olver (2003) shows how to coordinatize such manifold through for example, the stereographic projection, which requires two coordinate maps applied to two convenient neighborhoods on the sphere.

An interesting object we may think to on a differential manifold $\mathcal{M}$ is a smooth curve $\gamma : [a,b] \to \mathcal{M}$. In coordinates,[2] $x = \gamma(t)$ describes a curve on the manifold $\mathcal{M}$ delimited by the endpoints $\gamma(a)$ and $\gamma(b)$. Here, the manifold is supposed to be immersed in a suitable ambient Euclidean space $\mathcal{A}$ of suitable dimension (for instance, the sphere S^p may be though of as immersed in the ambient space $\mathcal{A} = \mathbb{R}^{p+1}$).

Let us now suppose $0 \in [a,b]$ and let us consider a curve γ passing by a given point $x \in \mathcal{M}$, namely $x = \gamma(0)$. The smooth curve admits a tangent vector $\mathbf{v}_x$ at the point x on the manifold, which is defined by

$$\mathbf{v}_x \stackrel{\text{def}}{=} \lim_{t \to 0} \frac{\gamma(t) - \gamma(0)}{t} \in \mathcal{A}.$$

Clearly, the vector $\mathbf{v}_x$ does not belong to the curved manifold $\mathcal{M}$ but is tangent to it in the point x. Let us imagine to consider every possible smooth curve on a manifold of dimension p passing through the point x and to compute the tangent vectors to these curves in the point x. The collection of these vectors span a linear space of dimension p, which is referred to as *tangent space* to the manifold $\mathcal{M}$ at the point x, and is denoted with $T_x\mathcal{M} \subseteq \mathcal{A}$.

As a further safety note, it might deserve to recall that, in differential geometry, the main way to regard for example, tangent spaces and vector fields is based on differential operators (Olver, 2003). This means, for instance, that a tangent vector $\mathbf{v} \in T_x\mathcal{M}$ of some smooth manifold $\mathcal{M}$ is defined in such a way that if $\mathcal{F}$ denotes a smooth functional space then for instance $\mathbf{v} : \mathcal{F} \to \mathbb{R}$, namely $\mathbf{v}(f)$ is a scalar for $f \in \mathcal{F}$. In this paper we chose not to invoke such notation. The reason is that we are interested in a special matrix-type Lie group (the orthogonal group), whose geometry may be conveniently expressed in terms of matrix-type quantities/operations. The theoretical bridge between the differential-operator-based representation and the matrix-based representation is given by the observation that every differential operator in $T_x\mathcal{M}$ may be written as a linear combination of elementary differential operators, that form a basis for the tangent space, through some coefficients. The structure of the tangent space is entirely revealed by the relationships among these coefficients. Therefore, we may choose to represent tangent vectors as algebraic vectors/matrices of coefficients, that is exactly what is implicitly done here.

It is now possible to give the definition of Riemannian manifold, which is a pair $(\mathcal{M}, g)$ formed by a differential manifold $\mathcal{M}$ and an inner product $g_x(\mathbf{v}_x, \mathbf{u}_x)$, locally defined in every point x of the manifold as a bilinear function from $T_x\mathcal{M} \times T_x\mathcal{M}$ to $\mathbb{R}$. It is important to remark that the inner product $g_x(\cdot,\cdot)$ acts on elements from the tangent space to the manifold at some given point, it therefore depends (smoothly) on the point x.

On a Riemannian manifold $(\mathcal{M}, g)$, we can measure the length of a vector $\mathbf{v} \in T_x\mathcal{M}$ as

$$\|\mathbf{v}\| \stackrel{\text{def}}{=} \sqrt{g_x(\mathbf{v}, \mathbf{v})}.$$

[2] It is worth remarking that a curve may interest different coordinate charts (ψ_k, U_k, p), therefore, it is generally necessary to split a curve in as many branches (or segments) as coordinate charts it bypasses.

Also, a remarkable property of Riemannian manifolds is that we can measure the length of a curve $\gamma : [a,b] \to \mathcal{M}$ on the manifold through the local metric on its tangent spaces. In fact, the length of the curve $\gamma(\cdot)$ is, by definition, $L_\gamma \overset{\text{def}}{=} \int_a^b ds$, where ds is the infinitesimal arc length. From geometry we know that $ds = \|\dot{\gamma}(t)\| dt$, therefore we have

$$L_\gamma = \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt. \tag{2}$$

The net result of this argument is that, through a definition of an inner product on the tangent spaces to a Riemannian manifold, we are able to measure the length of paths in the manifold itself, and this *turns the manifold into a metric space*.

A vector field $\mathbf{v}_x$ on manifold $\mathcal{M}$ specifies a vector belonging to the tangent space $T_x\mathcal{M}$ to the manifold at every point x.

With the notions of vector field and curve on a manifold, we may define the important concept of *geodesics*. A geodesic on a smooth manifold may be intuitively looked upon in at least three different ways:

- On a general manifold, the concept of geodesic extends the concept of straight line on a flat space to a curved space. An informal interpretation of this property is that a geodesic is a curve on a manifold that would resemble a straight line in an infinitesimal neighborhood of any of its points. The formal counterpart of this interpretation is rather involved because it requires the notion of covariant derivative of a vector field with respect to another vector field and leads to a second-order differential equation involving the Christoffel structural functions of the manifold (Amari, 1989).

- On a Riemannian manifold, a geodesic among two points is locally defined as the *shortest curve* on the manifold connecting these endpoints. Therefore, once a metric $g(\cdot,\cdot)$ is specified, the equation of the geodesic arises from the minimization of the functional (2) with respect to γ. In general, the obtained equation is difficult to solve in closed form.

- Another intuitive interpretation is based on the observation that a geodesic emanating from a point x on the manifold coincides to the path followed by a particle sliding on the manifold itself with constant scalar speed specified by the norm of the vector $\mathbf{v}_x$. For a manifold embedded in a Euclidean space, this is equivalent to require that the acceleration of the particle is either zero or perpendicular to the tangent space to the manifold in every point.

The concept of geodesic and geodesic equation are recalled here only informally. Appendix A provides a detailed account of these and related concepts just touched here, such as the Christoffel functions (or affine-connection coefficients).

An important vector field often considered in the literature of function optimization over manifolds is the *gradient vector field*. If we consider a smooth function $f : \mathcal{M} \to \mathbb{R}$ and define its gradient $\text{grad}_x^{\mathcal{M}} f$, then a oft-considered differential equation is:

$$\frac{dx}{dt} = \pm \, \text{grad}_x^{\mathcal{M}} f, \tag{3}$$

where the signs $+$ or $-$ denote maximization or minimization of the function f over the manifold. The solution of the above differential equation is referred to as *gradient flow* of f on $\mathcal{M}$.

Formally, the concept of *gradient* on a Riemannian manifold may be defined as follows. Let us consider a Riemannian manifold $(\mathcal{M}, g)$ and, for every point x, the tangent space $T_x\mathcal{M}$. Let us also consider a smooth function $f : \mathcal{M} \to \mathbb{R}$, the standard Euclidean inner product $g^{\mathcal{E}}$ in $T_x\mathcal{M}$ and the Jacobian $\mathrm{grad}_x^{\mathcal{E}} f = \frac{\partial f}{\partial x}$ of the function f with respect to x. The Riemannian gradient $\mathrm{grad}_x^{\mathcal{M}} f$ of the function f over the manifold $\mathcal{M}$ in the point x is uniquely defined by the following two conditions:

- Tangency condition. For every $x \in \mathcal{M}$, $\mathrm{grad}_x^{\mathcal{M}} f \in T_x\mathcal{M}$.
- Compatibility condition. For every $x \in \mathcal{M}$ and every $\mathbf{v} \in T_x\mathcal{M}$, $g_x(\mathrm{grad}_x^{\mathcal{M}} f, \mathbf{v}) = g^{\mathcal{E}}(\mathrm{grad}_x^{\mathcal{E}} f, \mathbf{v})$.

The tangency condition expresses the fact that a gradient vector is always tangent to the base-manifold, while the compatibility condition states that the inner product, under a metric on a manifold, of a gradient vector with any other tangent vector is invariant with the chosen metric. However, note that the gradient *does* depend on the metric. The ‘reference’ inner product is assumed as the Euclidean inner product that a flat space may be endowed with. For instance, if the base manifold $\mathcal{M}$ has dimension p, then it may be assumed $\mathcal{E} = T_x\mathcal{E} = \mathbb{R}^p$ in every point x and $g^{\mathcal{E}}(\mathbf{u}, \mathbf{v}) = \mathbf{v}^T\mathbf{u}$. It is worth noting that such special metric is *uniform*, in that it does not actually depend on the point x.

In order to facilitate the use of the compatibility condition for gradient computation, it is sometimes useful to introduce the concept of *normal space* of a Riemannian manifold in a given point under a chosen metric $g^{\mathcal{A}}$:

$$N_x\mathcal{M} \overset{\text{def}}{=} \{\mathbf{n} \in \mathcal{A} | g_x^{\mathcal{A}}(\mathbf{n}, \mathbf{v}) = 0 \ , \ \forall \mathbf{v} \in T_x\mathcal{M}\}.$$

It represents the orthogonal complement of the tangent space with respect to an Euclidean ambient space $\mathcal{A}$ that the manifold $\mathcal{M}$ is embedded within.

With the notion of algebraic group and smooth manifold, we may now define a well-known object of differential geometry, that is the *Lie group*. A Lie group conjugates the properties of an algebraic group and of a smooth manifold, as it is a set endowed with both group properties and manifold structure. An example of a Lie group that we are interested in within the paper is the *orthogonal group*:

$$O(p) \overset{\text{def}}{=} \{\mathbf{X} \in \mathbb{R}^{p \times p} | \mathbf{X}^T\mathbf{X} = \mathbf{I}_p\}. \tag{4}$$

It is easy to verify that it is a group (under standard matrix multiplication and inversion) and it is also endowed with the structure of a smooth manifold.

Consequently, we may for instance consider the tangent space T_xG of a Lie group G at the point x. A particular tangent space is T_eG, namely the tangent at identity, which, properly endowed with a binary operator termed *Lie bracket*, has the structure of a *Lie algebra* and is denoted with $\mathfrak{g}$.

An essential peculiarity of the Lie groups (G, m, i, e) is that the whole group may be always brought back to a convenient neighborhood of the identity e and the same holds for every tangent space T_xG, $\forall x \in G$, that may be brought back to the algebra $\mathfrak{g}$. Let us consider, for instance, a curve $\gamma(t) \in G$ passing through the point x, with $t \in [a, b]$ such that $0 \in [a, b]$ and $x = \gamma(0)$. We may define the new curve $\tilde{\gamma}(t) \overset{\text{def}}{=} m(\gamma(t), i(x))$ that enjoys the property $\tilde{\gamma}(0) = e$; conversely, $\gamma(t) = m(\tilde{\gamma}(t), x)$. This operation closely resembles a translation of a curve into a convenient neighborhood of the group identity, so that we can define a special operator referred to as *right translation* as

$$R_x : G \to G \ , R_x(\gamma) \overset{\text{def}}{=} m(\gamma, i(x)).$$

It is clear that every tangent vector $\mathbf{v}_x$ to the curve γ at x is also translated to a tangent vector $\tilde{\mathbf{v}}$ of the curve $\tilde{\gamma}(t)$ by a conveniently defined operator:

$$dR_x : T_x G \to T_e G\ , \tilde{\mathbf{v}} = dR_x(\mathbf{v}),$$

which is commonly referred to as *tangent map* associated to the (right) translation R_x. Such map is invertible and allows us to translate a vector belonging to a tangent space of a group to a vector belonging to its algebra (and vice-versa).[3]

From the above discussion, it is straightforward to see that, if the structure of $\mathfrak{g}$ is known for a group G, it might be convenient to coordinatize a neighborhood of the identity of G through elements of the associated algebra with the help of a conveniently-selected homeomorphism. Such homeomorphism is known in the literature as *exponential map* and is denoted with $\exp : \mathfrak{g} \to G$. It is important to recall that 'exp' is only a symbol and, even for matrix-type Lie groups, *does not necessarily denote matrix exponentiation*.

2.2 Gradient Flows on the Orthogonal Group

As mentioned, the orthogonal group $O(p)$ is a Lie group, therefore it is endowed with a manifold structure. Consequently, we may use the above-recalled instruments in order to define gradient-based learning equations of the kind (3) over $O(p)$ and to approximately solve them.

Some useful facts about the geometrical structure of the orthogonal group $O(p)$ are:

- The standard group multiplication on $O(p)$ is non-commutative (for $p \geq 3$).
- The group $O(p)$ manifold structure has dimension $\frac{p(p-1)}{2}$. In fact, every matrix in $O(p)$ possesses p^2 entries which are constrained by $\frac{p(p+1)}{2}$ orthogonality/normality restrictions.
- The inverse operation $i(\mathbf{X}) = \mathbf{X}^{-1}$ coincides with the transposition, namely $i(\mathbf{X}) = \mathbf{X}^T$.
- The tangent space of the Lie group $O(p)$ has the structure $T_{\mathbf{X}}O(p) = \{\mathbf{V} \in \mathbb{R}^{p \times p} | \mathbf{V}^T\mathbf{X} + \mathbf{X}^T\mathbf{V} = 0_p\}$. This may be proven by differentiating a generic curve $\gamma(t) \in O(p)$ passing by $\mathbf{X}$ for $t = 0$. Every such curve satisfies the orthogonal-group characteristic equation (4), namely $\gamma^T(t)\gamma(t) = \mathbf{I}_p$, therefore, after differentiation, we get $\dot{\gamma}^T(0)\gamma(0) + \gamma^T(0)\dot{\gamma}(0) = \mathbf{0}_p$. By recalling that the tangent space is formed by the velocity vectors $\dot{\gamma}(0)$, the above-mentioned result is readily achieved.
- The Lie algebra associated to the orthogonal group is the set of skew-symmetric matrices $\mathfrak{so}(p) \overset{\text{def}}{=} \{\tilde{\mathbf{V}} \in \mathbb{R}^{p \times p} | \tilde{\mathbf{V}} + \tilde{\mathbf{V}}^T = \mathbf{0}_p\}$. In fact, at the identity ($\mathbf{X} = \mathbf{I}_p$), we have $T_{\mathbf{I}_p}O(p) = \mathfrak{so}(p)$.
- The Lie algebra $\mathfrak{so}(p)$ is a vector space of dimension $\frac{p(p-1)}{2}$.

First, it is necessary to compute the gradient of a function $f : O(p) \to \mathbb{R}$ over the group $O(p)$ in view of computing the geodesic that emanates from a point $\mathbf{X} \in O(p)$ with velocity proportional to $\text{grad}_{\mathbf{X}}^{O(p)} f$. In this derivation, we essentially follow the definition of Riemannian gradient given in Section 2.1.

[3] This is the reason for which the Lie algebra of a Lie group is sometimes termed the 'generator' of the group.

Let the manifold $O(p)$ be equipped with the canonical induced metric $g^{O(p)}$, that is $g^{O(p)}_{\mathbf{X}}(\mathbf{U},\mathbf{V})\overset{\text{def}}{=}\text{tr}[\mathbf{U}^T\mathbf{V}]$, for every $\mathbf{X}\in O(p)$ and every $\mathbf{U},\mathbf{V}\in T_{\mathbf{X}}O(p)$. This metric coincides with the standard Euclidean metric $g^{\mathbb{R}^{p\times p}}$ in $\mathbb{R}^{p\times p}$. Having endowed the manifold $O(p)$ with a metric, it is possible to describe completely its normal space, provided the ambient space $\mathcal{A}$ is endowed with the canonical Euclidean metric. In fact, we have

$$N_{\mathbf{X}}O(p)=\{\mathbf{N}=\mathbf{X}\mathbf{S}\in\mathbb{R}^{p\times p}|\text{tr}[\mathbf{N}^T\mathbf{V}]=0\ ,\ \forall\mathbf{V}\in T_{\mathbf{X}}O(p)\}.$$

The matrix $\mathbf{S}$ should have a particular structure. In fact, the normality condition, in this case, writes $0=\text{tr}[\mathbf{V}^T(\mathbf{X}\mathbf{S})]=\text{tr}[\mathbf{S}\mathbf{V}^T\mathbf{X}]=\text{tr}[(\mathbf{X}^T\mathbf{V})\mathbf{S}^T]$. The latter expression, thanks to the structure of tangent vectors, is equivalent to $-\text{tr}[(\mathbf{V}^T\mathbf{X})\mathbf{S}^T]$, therefore the normality condition may be equivalently rewritten as $\text{tr}[(\mathbf{V}^T\mathbf{X})(\mathbf{S}-\mathbf{S}^T)]=0$. In order for this to be true, in the general case, it is necessary and sufficient that $\mathbf{S}=\mathbf{S}^T$. Thus:

$$N_{\mathbf{X}}O(p)=\{\mathbf{X}\mathbf{S}|\mathbf{S}^T=\mathbf{S}\in\mathbb{R}^{p\times p}\}.$$

Let $\text{grad}^{O(p)}_{\mathbf{X}}f$ be the gradient vector of f at $\mathbf{X}\in O(p)$ derived from the metric $g^{O(p)}$. According to the compatibility condition for the Riemannian gradient:

$$g^{\mathbb{R}^{p\times p}}_{\mathbf{X}}(\mathbf{V},\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f)=g^{O(p)}_{\mathbf{X}}(\mathbf{V},\text{grad}^{O(p)}_{\mathbf{X}}f),$$

for every tangent vector $\mathbf{V}\in T_{\mathbf{X}}O(p)$, therefore:

$$g^{\mathbb{R}^{p\times p}}_{\mathbf{X}}(\mathbf{V},\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f-\text{grad}^{O(p)}_{\mathbf{X}}f)=0,$$

for all $\mathbf{V}\in T_{\mathbf{X}}O(p)$. This implies that the quantity $\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f-\text{grad}^{O(p)}_{\mathbf{X}}f$ belongs to $N_{\mathbf{X}}O(p)$. Explicitly:

$$\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f=\text{grad}^{O(p)}_{\mathbf{X}}f+\mathbf{X}\mathbf{S}. \tag{5}$$

In order to determine the symmetric matrix $\mathbf{S}$, we may exploit the tangency condition on the Riemannian gradient, namely $(\text{grad}^{O(p)}_{\mathbf{X}}f)^T\mathbf{X}+\mathbf{X}^T(\text{grad}^{O(p)}_{\mathbf{X}}f)=\mathbf{0}_p$. Let us first pre-multiply both sides of the equation (5) by $\mathbf{X}^T$, which gives

$$\mathbf{X}^T\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f=\mathbf{X}^T\text{grad}^{O(p)}_{\mathbf{X}}f+\mathbf{S}.$$

The above equation, transposed hand-by-hand, becomes

$$(\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f)^T\mathbf{X}=(\text{grad}^{O(p)}_{\mathbf{X}}f)^T\mathbf{X}+\mathbf{S}.$$

Hand-by-hand summation of the last two equations gives

$$(\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f)^T\mathbf{X}+\mathbf{X}^T(\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f)=2\mathbf{S},$$

that is:

$$\mathbf{S}=\frac{(\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f)^T\mathbf{X}+\mathbf{X}^T(\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f)}{2}. \tag{6}$$

By plugging the expression (6) into expression (5), we get the form of the Riemannian gradient in the orthogonal group, which is:

$$\text{grad}^{O(p)}_{\mathbf{X}}f=\frac{\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f-\mathbf{X}(\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{X}}f)^T\mathbf{X}}{2}.$$

About the expression of the geodesic, as mentioned, in general it is not easy to obtain in closed form. In the present case, with the assumptions considered, the geodesic on $O(p)$ departing from the identity with velocity $\tilde{\mathbf{V}} \in \mathfrak{so}(p)$ has expression $\tilde{\gamma}(t) = \exp(t\tilde{\mathbf{V}})$. (It is immediate to verify that $\tilde{\gamma}(0) = \mathbf{I}_p$ and $\left.\frac{d\tilde{\gamma}(t)}{dt}\right|_{t=0} = \tilde{\mathbf{V}}$.) It might be useful to verify such essential result by the help of the following arguments.

As already recalled in Section 2.1, when a manifold is embedded in a Euclidean space, the second derivative of the geodesic with resepct to the parameter is either zero or perpendicular to the tangent space to the manifold in every point (see Appendix A). Therefore, a geodesic $\tilde{\gamma}(t)$ on the Riemannian manifold $(O(p), g^{O(p)})$ embedded in the Euclidean ambient space $(\mathbb{R}^{p\times p}, g^{\mathbb{R}^{p\times p}})$, departing from the identity $\mathbf{I}_p$, should be such that $\ddot{\tilde{\gamma}}(t) \in N_{\mathbf{I}_p}O(p)$, therefore it should hold:

$$\ddot{\tilde{\gamma}}(t) = \tilde{\gamma}(t)\mathbf{S}(t)\ ,\ \text{with } \mathbf{S}^T(t) = \mathbf{S}(t). \tag{7}$$

Also, we known that any geodesic branch belongs entirely to the base manifold, therefore $\tilde{\gamma}^T(t)\tilde{\gamma}(t) = \mathbf{I}_p$. By differentiating two times such expression with respect to the parameter t it is easily gotten:

$$\ddot{\tilde{\gamma}}^T(t)\tilde{\gamma}(t) + 2\dot{\tilde{\gamma}}^T(t)\dot{\tilde{\gamma}}(t) + \tilde{\gamma}^T(t)\ddot{\tilde{\gamma}}(t) = \mathbf{0}_p. \tag{8}$$

By plugging equation (7) into equation (8), we find $\mathbf{S}(t) = -\dot{\tilde{\gamma}}^T(t)\dot{\tilde{\gamma}}(t)$, which leads to the second-order differential equation on the orthogonal group:

$$\ddot{\tilde{\gamma}}(t) = -\tilde{\gamma}(t)(\dot{\tilde{\gamma}}^T(t)\dot{\tilde{\gamma}}(t)),$$

to be solved with initial conditions $\tilde{\gamma}(0) = \mathbf{I}_p$ and $\dot{\tilde{\gamma}}(0) = \tilde{\mathbf{V}}$. It is a straightforward task to verify that the solution to this second-order differential equation is given by the one-parameter curve $\tilde{\gamma}(t) = \exp(t\tilde{\mathbf{V}})$, where $\exp(\cdot)$ denotes matrix exponentiation.

The expression of the geodesic in the position of interest may be made explicit by taking advantage of the Lie-group structure of the orthogonal group endowed with the canonical metric. In fact, let us consider the pair $\mathbf{X} \in O(p)$ and $\mathrm{grad}^{O(p)}_{\mathbf{X}} f \in T_{\mathbf{X}}O(p)$ as well as the geodesic $\gamma(t)$ that emanates from $\mathbf{X}$ with velocity $\mathbf{V}$ proportional to $\mathrm{grad}^{O(p)}_{\mathbf{X}} f$, and let us suppose for simplicity that $\gamma(0) = \mathbf{X}$. Let us now consider the right-translated curve $\tilde{\gamma}(t) = \gamma(t)\mathbf{X}^T$. The new curve enjoys the following properties:

1. It is such that $\tilde{\gamma}(0) = \mathbf{I}_p$, therefore it passes through the identity of the group $O(p)$.

2. The tangent vector $\mathbf{V}$ to the curve at $\mathbf{X}$ is ‘transported’ into the tangent vector:

   $$\tilde{\mathbf{V}} = \mathbf{V}\mathbf{X}^T, \tag{9}$$

   at the identity, so $\tilde{\mathbf{V}} \in \mathfrak{so}(p)$.

3. As the right-translation is an isometry, the curve $\tilde{\gamma}(t)$ is still a geodesic departing from the identity matrix with velocity proportional to $\tilde{\mathbf{V}} = (\mathrm{grad}^{O(p)}_{\mathbf{X}} f)\mathbf{X}^T$.

From these observations, we readily obtain the geodesic in the position of interest $\mathbf{X} \in O(p)$, namely $\gamma(t) = \exp(t\tilde{\mathbf{V}})\mathbf{X}$.

As this is an issue of prime importance, we deem it appropriate to verify that the curve $\gamma(t)$ just defined belongs to the orthogonal group at any time. This may be proven by computing the quantity $\gamma^T(t)\gamma(t)$ and taking into account the identity $\exp^T(t\tilde{\mathbf{V}}) = \exp(-t\tilde{\mathbf{V}})$. Then we have

$$\gamma^T(t)\gamma(t) = \mathbf{X}^T \exp^T(t\tilde{\mathbf{V}})\exp(t\tilde{\mathbf{V}})\mathbf{X} = \mathbf{X}^T\exp(-t\tilde{\mathbf{V}})\exp(t\tilde{\mathbf{V}})\mathbf{X} = \mathbf{X}^T\mathbf{X} = \mathbf{I}_p.$$

2.3 Comments on Stability and the Relationship with Natural Gradient Theory

Some comments on the questions of the stability of gradient-based learning algorithms on the orthogonal group and on the relationship of Riemannian gradient-based learning algorithms on the orthogonal group with the well-known ‘natural’ gradient-based optimization theory are in order.

When applied to the manifold $\mathcal{M} = O(p)$, the general gradient-based learning equation (3) has the inherent property of keeping the connection matrix $\mathbf{X}$ within the group $O(p)$ at any time. It is very important to note that the discrete-time version of this learning equation, described in Section 3, also enjoys this noticeable property. When for example, learning algorithms based on the manifold $Gl(p)$, defined in equation (1), are dealt with, one of the theoretical efforts required to prove their stability consists in showing that there exists a compact sub-manifold that is an attractor for the learning system. The above observations reveal that the problem of the existence of an orthogonal-group-attractor for discrete-time learning systems based on the orthogonal Lie group does not arise when a proper integration algorithm is exploited. Moreover, as opposed to the Euclidean space $\mathbb{R}^p$ and the general-linear group $Gl(p)$, the orthogonal group $O(p)$ is a compact space. This means that no diverging trajectories exist for the learning system (3) or its discrete-time counterpart. Such effect may be easily recognized in the two-dimensional ($p = 2$) case, through the parameterization $\psi^{-1} : [-\pi, \pi[\to SO(2)$:

$$\mathbf{X} = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}. \tag{10}$$

It is worth noting that $\det(\mathbf{X}) = 1$, while, in general, the determinant of an orthonormal matrix may be either -1 or $+1$, in fact $1 = \det(\mathbf{X}^T\mathbf{X}) = \det^2(\mathbf{X})$ for $\mathbf{X} \in O(p)$. This means that the above parameterization spans one of the two components of the orthogonal group termed *special orthogonal* group and denoted by $SO(p)$. (In the above notation, we easily recognize a coordinate chart $(\psi, SO(2), 1)$ associated to $O(2)$.) Now, by singling out the columns of the matrix $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2]$, we easily see that $\|\mathbf{x}_1\| = \|\mathbf{x}_2\| = 1$, which proves the space $SO(2)$ is compact. The same reasoning may be repeated for the remaining component of $O(2)$.

In its general formulation, the widely-known ‘natural gradient’ theory for learning may be summarized as follows. The base-manifold for learning is the group of non-singular matrices $Gl(p)$ that is endowed with a metrics based on the Fisher metric tensor which, in turn, derives from a truncated expansion of the Kullback-Leibler informational divergence (KLD) (Amari, 1998). The latter choice derives from the possibility – offered by the KLD – to induce a metrics in the abstract space of neural networks having same topology but different connection parameters, which is referred to as *neural manifold*.

In the independent component analysis case, a special structure was envisaged by Yang and Amari (1997) for the natural gradient by imposing a Riemannian structure on the Lie group of non-singular matrices $Gl(p)$. We believe it could be useful to briefly recall this intuition here by using the language of Lie groups recalled in Section 2.1. First, the tangent space at identity $\mathbf{I}_p$ to $Gl(p)$ is denoted by $\mathfrak{gl}(p)$, as usual. Such Lie algebra may be endowed with a scalar product $g^{Gl(p)}_{\mathbf{I}_p}(\cdot,\cdot) : \mathfrak{gl}(p) \times \mathfrak{gl}(p) \to \mathbb{R}$. As there is no reason to weight in a different way the components of the matrices in $\mathfrak{gl}(p)$, it is assumed $g^{Gl(p)}_{\mathbf{I}_p}(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}) \stackrel{\text{def}}{=} \text{tr}[\tilde{\mathbf{U}}^T\tilde{\mathbf{V}}]$. The question is now how to define the scalar product in a generic tangent space $T_{\mathbf{X}}Gl(p)$, with $\mathbf{X} \in Gl(p)$. Let us consider, to this purpose, a curve $\gamma(t) \in Gl(p)$ passing by the point $\mathbf{X}$ at $t = 0$, namely $\gamma(0) = \mathbf{X}$. This curve may always be translated into a neighborhood of the identity of the group by the left-translation $\tilde{\gamma}(t) \stackrel{\text{def}}{=} \mathbf{X}^{-1}\gamma(t)$, in fact, the inverse $\mathbf{X}^{-1}$ surely exists because $Gl(p)$ is the set of all invertible $p \times p$ matrices by definition and now $\tilde{\gamma}(0) = \mathbf{I}_p$. Therefore, if $\mathbf{V} \in T_{\mathbf{X}}Gl(p)$ denotes the tangent vector to the curve $\gamma(t)$ at $t = 0$ and $\tilde{\mathbf{V}} \in \mathfrak{gl}(p)$ denotes the tangent vector to the curve $\tilde{\gamma}(t)$ at $t = 0$, they are related by the corresponding tangent map $\mathbf{V} \to \tilde{\mathbf{V}} = \mathbf{X}^{-1}\mathbf{V}$. This observation may be exploited to define an inner product on the tangent spaces of $Gl(p)$ by *imposing* the Riemannian-structure invariance property:

$$g_{\mathbf{X}}^{Gl(p)}(\mathbf{U},\mathbf{V}) \overset{\text{def}}{=} g_{\mathbf{I}_p}^{Gl(p)}(\mathbf{X}^{-1}\mathbf{U},\mathbf{X}^{-1}\mathbf{V}) = \text{tr}[\mathbf{U}^T(\mathbf{X}^T)^{-1}\mathbf{X}^{-1}\mathbf{V}].$$

Having defined a general (non-uniform) metric in the tangent spaces to $Gl(p)$, we may now compute the Riemannian (natural) gradient on it, by invoking the tangency and compatibility conditions as stated in Section 2.1. Actually, the tangency condition does not provide any constraint in the present case, because every $T_{\mathbf{X}}Gl(p)$ is ultimately isomorphic to $\mathbb{R}^{p\times p}$. The compatibility condition, instead, writes, for a smooth function $f : Gl(p) \to \mathbb{R}$:

$$g_{\mathbf{X}}^{Gl(p)}(\text{grad}_{\mathbf{X}}^{Gl(p)} f,\mathbf{V}) = \text{tr}\left[\left(\frac{\partial f}{\partial \mathbf{X}}\right)^T \mathbf{V}\right] \ , \ \forall \mathbf{V} \in T_{\mathbf{X}}Gl(p).$$

This condition implies:

$$\begin{aligned}
& \text{tr}\left[\left\{(\text{grad}_{\mathbf{X}}^{Gl(p)} f)^T(\mathbf{X}^T)^{-1}\mathbf{X}^{-1} - \left(\frac{\partial f}{\partial \mathbf{X}}\right)^T\right\}\mathbf{V}\right] = 0 \ , \ \forall \mathbf{V} \in T_{\mathbf{X}}Gl(p) \\
\Rightarrow \quad & (\text{grad}_{\mathbf{X}}^{Gl(p)} f)^T(\mathbf{X}^T)^{-1}\mathbf{X}^{-1} = \left(\frac{\partial f}{\partial \mathbf{X}}\right)^T \\
\Rightarrow \quad & \text{grad}_{\mathbf{X}}^{Gl(p)} f = (\mathbf{X}\mathbf{X}^T)\frac{\partial f}{\partial \mathbf{X}}.
\end{aligned}$$

Of course, a different form for the natural gradient may be obtained by choosing the right-translation $\tilde{\gamma}(t) \overset{\text{def}}{=} \gamma(t)\mathbf{X}^{-1}$ as a basis for invariance, as done for example, by Yang and Amari (1997). The 'natural gradient' theory for $Gl(p)$ and the Riemannian-gradient-theory for the group $O(p)$ are thus somewhat unrelated, even if ultimately the 'natural gradient' is a Riemannian gradient on the group $Gl(p)$ arising from a specific metric. Some further details on the optimization problem over the general linear group (about for example, using the exponential map on $Gl(p)$) have been presented by Akuzawa (2001).

Another interesting comparison is with the information-geometry theory for learning.[4] In the spirit of information geometry, the natural gradient works on a manifold of parameterized likelihood. Now, in two dimensions, the Riemannian geometry of the orthogonal group, defined by the parameterization (10) above, may be clearly related to the information geometry of the binomial distribution defined by the variables r, q such that $r + q = 1$, via the transform $r = \cos^2(\beta)$, $q = \sin^2(\beta)$. Whether such link exists in any dimension ($p \geq 3$) is not known to the author and would be worth investigating in future works. The same holds for the relationship with second order (Newton) method, which is known for the natural gradient (see, for example, Park et al. (2000) and references therein) but whose relationship with general Riemannian gradient theory is to be elucidated.

[4] This interesting connection was suggested by a reviewer.

3. Learning Over the Orthogonal Group: Three Algorithms

In order to numerically integrate a continuous learning differential equation on a manifold, a proper discretization method should be exploited. On a flat space, a possible discretization method is line approximation based on Euler's or trapezoidal technique (or some more sophisticated techniques such as the Runge-Kutta method). However, if applied to differential equations based on curved manifolds, such ordinary discretization methods produce updating rules that do not satisfy the manifold constraints. Following the general differential-geometric knowledge, two possible ways to tackle the problem are:

- The projection method. It consists in projecting the updated value to the manifold after each iteration step. More formally, this method consists in embedding the manifold $\mathcal{M}$ of interest into a Euclidean space of proper dimension $\mathcal{A}$ and to discretize the differential equation whose variable is regarded as belonging to $\mathcal{A}$ through any suitable ordinary method. Then, in each iteration, the newly found approximated solution is projected back to the manifold through a suitable *projector* $\Pi : \mathcal{A} \to \mathcal{M}$. The next iteration starts from the projected putative solution.

- The geodesic method. The principle behind the geodesic method is to replace the line approximation to the original differential equation by the geodesic approximation in the manifold. From a geometrical point of view, this seems a natural approximation because a geodesic on a manifold is a counterpart of a line in the Euclidean space. Furthermore, a geodesic on a Riemannian manifold is a length-minimizing curve between two points, which looks quite appealing if we regard an optimization process as connecting an initial solution to a stationary point of a criterion function through the shortest path.

The viewpoint adopted in the present contribution is that the geodesic-based approach is the most natural one from a geometric perspective and the most capable of future extensions to different base-manifolds. The projection method will also be considered, for comparison purposes only, in the section devoted to simulation results.

In particular, we suppose to approximate the flow of the differential learning equation (3) through geodesic arcs properly connected, so as to obtain a piece-wise geodesic-type approximation of the exact gradient flow. If we denote by $\mathbf{W} \in O(p)$ the pattern to be learnt (for instance the connection matrix of a one-layer neural network), the considered geodesic-based learning algorithm corresponding to the exact Riemannian gradient flow is implemented by considering learning steps of the form:

$$\mathbf{W}_{n+1} = \exp(\eta_n((\mathrm{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{W}_n} f)\mathbf{W}_n^T - \mathbf{W}_n(\mathrm{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{W}_n} f)))\mathbf{W}_n, \tag{11}$$

where the index $n \in \mathbb{N}$ denotes a learning step counter and η_n denotes an integration or learning stepsize (the factor $\frac{1}{2}$ may be safely absorbed in η_n) usually termed *(learning) schedule* or step-size. It deserves underlining that the integration step-size may change across iterations because it may be beneficial to vary the step-size according to the progress of learning. The initial solution $\mathbf{W}_0$ should be selected in $O(p)$. It should be noted that the matrix $\mathbf{W}$ plays now the role of the general matrix $\mathbf{X}$ used in the previous section.

The aim of the present section is to consider three Riemannian gradient algorithms over the Lie group of orthogonal matrices. All three algorithms ensure that the current network-state matrix remain within the orthogonal group:

- Algorithm 1 uses a fixed step-size in the general geodesic-based learning equation (11).

- Algorithm 2 uses a geodesic line search for optimizing the step-size in the general geodesic-based learning equation (11).
- Algorithm 3 introduces stochasticity in the Algorithm 1, using a Markov-Chain Monte-Carlo method, jointly with an annealing procedure.

3.1 Deterministic Algorithms

A learning algorithm based on the findings of Section 2 may be stated as follows, where it is supposed that a constant learning step-size is employed.

$\diamond$ **Learning algorithm 1**:

1. Set $n = 0$, generate an initial solution $\mathbf{W}_0$ and set $f_0 = f(\mathbf{W}_0)$ and define a constant step-size η.
2. Compute a candidate solution $\mathbf{W}_{n+1}$ through the equation (11), increment n and return to 2, unless n exceeds the maximum number of iteration permitted: In this case, exit.

Formally, as mentioned in Section 2.1, the concept of geodesic is essentially local, therefore the discrete steps (11) on the orthogonal group should be extended for small values of η_n. Instead of keeping η_n constant or letting it progressively decreases through some 'cooling scheme', as it is customary in classical learning algorithms, it could allegedly be convenient to optimize it during learning. It is worth underlining at this point that the numerical evaluation of the geodesic curve through the exponential map, as well as the effective movement along a geodesic, are computationally expensive operations.

Step-size adaptation may be accomplished through a proper 'line search', as explained in what follows. Let us first define the following quantities for the sake of notation conciseness:

$$\tilde{\mathbf{V}}_n \stackrel{\text{def}}{=} (\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{W}_n} f)\mathbf{W}_n^T - \mathbf{W}_n(\text{grad}^{\mathbb{R}^{p\times p}}_{\mathbf{W}_n} f)^T,\ \mathbf{E}_n(t) \stackrel{\text{def}}{=} \exp(t\tilde{\mathbf{V}}_n). \tag{12}$$

Starting from a point $\mathbf{W}_n$ at iteration step n, according to equation (11), the next point would be $\mathbf{E}_n(t)\mathbf{W}_n$, therefore the learning criterion function would descend from $f(\mathbf{W}_n)$ to $f_n(t) \stackrel{\text{def}}{=} f(\mathbf{E}_n(t)\mathbf{W}_n)$. From the definition of f, which is continuous and defined on a compact manifold, it follows that the function $f_n(t)$ admits a point of minimum for $t \in \mathbb{T} \subset \mathbb{R}^-$, that may be denoted as $t_\star$. If we are able to find $t_\star$ in a computationally convenient way, we may then select $\eta_n = t_\star$. The operation of searching for a convenient value as close as possible to $t_\star$ is termed *geodesic search* as it closely resembles the familiar concept of 'line search'.

Basically, we may perform a geodesic search in two different ways:

- By sampling the interval $\mathbb{T}$ through a sequence of discrete indices t_k, computing the value of $f_n(t_k)$ and selecting the value that grants the smallest cost.
- By computing the derivative $\frac{d f_n(t)}{dt}$ and looking for the value of the index t for which it is equal (or sufficiently close) to zero. This approach would look advantageous if the expression of such equation could be handled analytically in an straightforward way. We found it is not the case and that this approach looks excessively cumbersome from a computational viewpoint, therefore it will not be adopted in this paper.

A second learning algorithm based on the above considerations may be stated as follows.

◇ **Learning algorithm 2**:

1. Set $n = 0$, generate an initial solution $\mathbf{W}_0$ and set $f_0 = f(\mathbf{W}_0)$.
2. Compute the quantity $\tilde{\mathbf{V}}_n$ in the equations (12).
3. Perform a geodesic-search for the optimal step-size η_n.
4. Compute a candidate solution $\mathbf{W}_{n+1}$ through the equation (11) and evaluate $f_{n+1} = f(\mathbf{W}_{n+1})$.
5. If $f_{n+1} < f_n$ then accept the candidate solution, increment n and return to 2, unless n exceeds the maximum number of iteration permitted: In this case, exit. If $f_{n+1} \geq f_n$, then proceed to 6.
6. Generate a small random step-size η_n.
7. Compute the candidate solution $\mathbf{W}_{n+1}$ through the equation (11), evaluate $f_{n+1} = f(\mathbf{W}_{n+1})$, increment n and return to 2.

The steps 6 and 7 in the above algorithm have been introduced in order to tackle the case in which the geodesic search gives rise to a candidate solution that causes the network's connection pattern to ascend the cost function f instead of making it descend. In this case, moving along the geodesic of a small random quantity does not ensure monotonic decreasing of the cost function, but it might help moving to another zone of the parameter space in which the geodesic learning might be effective.

3.2 Diffusion-Type Gradient Algorithm

In order to mitigate the known numerical convergence difficulties associated to the plain gradient-based optimization algorithms, it might be beneficial to perturb the standard Riemannian gradient to obtain a randomized gradient. In particular, following Liu et al. (2004), we may replace the gradient-based optimization steps with joint *simulated annealing* and *Markov-Chain Monte-Carlo* (MCMC) optimization technique, which gives rise to a so-termed *diffusion-type optimization process*. The Markov-Chain Monte-Carlo method was proposed and developed in the classical papers by Hastings (1970) and Metropolis et al. (1953).

It is worth recalling that, in classical algorithms, perturbations are easily introduced by sampling each network input one by one and by exploiting only such instantaneous information at a time. When used in conjunction with gradient-based learning algorithms, this inherently produces a stochastic gradient optimization based on a random walk on the parameters space. The two main reasons for which such choice is not adopted here are:

- When statistical expectations are replaced by one-sample mean, as it is customarily done, for example, in on-line signal processing, part of the information content pertaining to past samples is discarded from the learning system, and this might be a serious side effect on learning capability.
- The annealed MCMC method offers the possibility of actually *controlling* the amount of stochasticity introduced in the learning system by properly setting the method's free parameters such as the annealing temperature. Classical random-walk learning algorithms – as the one based on sampling each network input one by one – do not seem to offer such possibility.

A general discussion on the possible benefits owing to the introduction of stochasticity in gradient-based learning systems has been presented by Wilson and Martinez (2003).

It is understood that in a learning process having a Euclidean space as base-manifold, each step is simply proportional to the gradient computed in the departing point, therefore the learning steps may be directly perturbed in order to exploit randomized parameter-space search. In the present context, however, the base manifold $O(p)$ is curved, therefore it is sensible to perturb the gradient in the Lie algebra and then apply the formulas explained in the Section 2.2 to compute the associated step in the base-group.

In short, simulated annealing consists in adding to the deterministic gradient a random component whose amplitude is proportional to a parameter referred to as *temperature*. This mechanism may help the optimization algorithm to escape local solutions, but it has the drawback of occasionally leading to changes of the variable of interest toward the wrong direction (that is, it may lead to intermediate solutions with higher values of the criterion function when its minimum is sought for or vice-versa). Such drawback may be gotten rid of by adopting a MCMC-type simulated annealing optimization strategy where the diffusion-type gradient is exploited to generate a possible candidate for the next intermediate solution which is accepted/rejected on the basis of an appropriate probability distribution.

According to Liu et al. (2004), the diffusion-type gradient on the algebra $\mathfrak{so}(p)$ may be assumed as

$$\tilde{\mathbf{V}}_{\mathrm{diff}}(t) = \tilde{\mathbf{V}}(t) + \sqrt{2\Theta} \sum_{k=1}^{p(p-1)/2} \mathbf{L}_k \frac{d\mathcal{W}_k}{dt}, \tag{13}$$

where $\tilde{\mathbf{V}}(t)$ is the gradient (9), $\{\mathbf{L}_k\}$ is a basis of the Lie algebra $\mathfrak{so}(p)$, orthogonal with respect to the metric $g^{O(p)}_{\mathbf{I}_p}$, the $\mathcal{W}_k(t)$ are real-valued, independent standard Wiener processes and the parameter $\Theta > 0$ denotes the aforementioned temperature, which proves useful for simulating annealing during learning. It is worth recalling that a Wiener process is a continuous-time stochastic process $\mathcal{W}(t)$ for $t \geq 0$, that satisfies the following conditions (Higham, 2001):

- $\mathcal{W}(0) = 0$ with probability 1.

- For $0 \leq \tau < t$ the random variable given by the increment $\mathcal{W}(t) - \mathcal{W}(\tau)$ is normally distributed with mean zero and variance $t - \tau$. Equivalently, $\mathcal{W}(t) - \mathcal{W}(\tau) \sim \sqrt{t-\tau}\mathcal{N}(0,1)$, where $\mathcal{N}(0,1)$ denotes a normally distributed random variable with zero mean and unit variance.

- For $0 \leq \tau < t < u < v$, the increments $\mathcal{W}(t) - \mathcal{W}(\tau)$ and $\mathcal{W}(v) - \mathcal{W}(u)$ are statistically independent.

The learning differential equation on the orthogonal group associated to the gradient (13) reads

$$\frac{d\mathbf{W}}{dt} = -\tilde{\mathbf{V}}_{\mathrm{diff}}(t)\mathbf{W}(t), \tag{14}$$

is a *Langevin-type stochastic differential equation* (LDE).

By analogy with physical phenomena described by this equation, such as the Brownian motion of particles, the solution to the LDE is termed a *diffusion process*. Under certain conditions on the criterion function f, the solution of equation (14) is a Markov process endowed with a *unique stationary* probability density function (Srivastava et al., 2002), described by

$$\pi_{\mathrm{LDE}}(\mathbf{W}) \overset{\mathrm{def}}{=} \frac{1}{Z(\Theta)} \exp(-f(\mathbf{W})/\Theta), \tag{15}$$

where $Z(\Theta)$ denotes the density-function normalizer (partition function).[5] In other terms, the LDE 'samples' from the distribution $\pi_{\mathrm{LDE}}(\mathbf{W})$: This is a main concept in the method of using the LDE to generate random samples according to a given energy/cost function.

The choice of assuming the probability π_{LDE} inversely proportional to the value of $f(\mathbf{W})$ serves at discouraging network states corresponding to high values of the learning cost function. Also, it deserves to note that care should be taken of the problem related to the consistency of the above definition: The problem of the existence of π_{LDE}, that is connected to the existence of the partition function $Z(\Theta)$, must be dealt with. To this aim, it is worth noting that $f(\mathbf{W})$ is a continuous function of the argument which belongs to a compact space, we may therefore argue that $f(\mathbf{W})$ is bounded from above and from below. Thus, the function $\exp(-f(\mathbf{W}))$ is bounded and its integral over the whole orthogonal group through a coordinate-invariant measure of volume, such as the Haar measure (Srivastava et al., 2002), is surely existent.

In order to practically perform statistical sampling via the LDE, we can distinguish between *rejection* and *MCMC* methods:

1. The rejection algorithm is designed to give an exact sample from the distribution. Let us denote by $\pi(x)$ a density to sample from a set $\mathcal{X}$: We can sample from another distribution $\mu(x)$ (instrumental distribution) such that sampling from it is practically easier than actually sampling from $\pi(x)$. Then, it is possible to generate x^* from $\mu(x)$ and accept it with probability

$$\alpha \overset{\mathrm{def}}{=} \frac{\pi(x^*)}{\mu(x^*)M},$$

   where M is a constant such that $\pi(x)/\mu(x) \leq M$ for all $x \in \mathcal{X}$. If the generated sample is not accepted, rejection is performed until acceptance. When accepted, it is considered to be an exact sample from $\pi(x)$. A consequence of adopting this method is that the number of necessary samplings from $\mu(x)$ is unpredictable.

2. In MCMC, a Markov chain is formed by sampling from a conditional distribution $\mu(x|y)$: The algorithm starts from x_0 and proceeds iteratively as follows: At step n, sample x^* from $\mu(x|x_n)$ and compute the acceptation (Metropolis-Hastings) probability as

$$\alpha_n \overset{\mathrm{def}}{=} \min\left\{1, \frac{\pi(x^*)\mu(x_n|x^*)}{\pi(x_n)\mu(x^*|x_n)}\right\}, \tag{16}$$

   then accept x^* with probability α_n. This means letting $x_{n+1} = x^*$ with probability α_n, otherwise $x_{n+1} = x_n$. This is the main difference with rejection method: If the candidate sample is not accepted, then the previous value is retained.

[5] The theory presented by Srivastava et al. (2002) deals with the special case in which the base-manifold is $O(3)$. This result is not related to the dimension of the orthogonal group of interest, indeed, therefore it may be extended without difficulty to the general case $O(p)$ of concern in the present paper.

In the MCMC method, the quantity $\mu(x|y)$ denotes a *transition probability* as it describes the probability of 'jumping' from state y to state x. The total probability of transition from state x_n to state x_{n+1} is given by the combination of the instrumental distribution $\mu(x|y)$ and the Metropolis-Hastings acceptation probability: The transition kernel $K(x_{n+1}|x_n)$ is, in fact:

$$K(x_{n+1}|x_n) \overset{\text{def}}{=} \alpha_n \mu(x_{n+1}|x_n) + (1-\alpha_n)\delta(x_{n+1}-x_n).$$

In order to gain a physical interpretation of the instrumental probability $\mu(x|y)$, it pays to take for example a symmetric instrumental $\mu(x|y)$. Under this hypothesis, the ratio in the definition (16) would become $\pi(x^*)/\pi(x_n)$: The chain jumps to the state x^* if it is more plausible ($\alpha_n = 1$) than the previous state x_n, otherwise (case $\alpha_n < 1$), the chain jumps to the generated state according to the probability α_n. As an example of symmetric instrumental conditional probability, $\mu(x|y)$ may be assumed as Gaussian in x with mean y.

If the Markov chain $\{x_n\}_{n=1,\ldots,N}$ converges to the true probability $\pi(x)$, then x_n is asymptotically drawn from $\pi(x)$, so x_n is not an exact sample as in the rejection method. However, there is a powerful mathematical result that warrants that the empirical average (ergodic sum) $\sum_n \ell(x_n)/N$, for a regular function $\ell : \mathcal{X} \to \mathbb{R}$, converges to $\mathbb{E}[\ell(x)]$ if the chain converges asymptotically to the true distribution. For example, if x is a zero-mean scalar random variable and $\mathcal{X} = \mathbb{R}$, then $\ell(x) = x^2$ for the variance and $\ell(x) = x^4$ for the kurtosis of the variable. For this reason, MCMC methods are considered to be preferable over rejection method because in this latter only one exact sample is obtained, while with the former we obtain a chain and are thus able to approximate expectations. In order to perform MCMC, there is a great flexibility in choosing the instrumental probability density $\mu(x|y)$.

For a recent review of the MCMC method, interested readers may consult for instance the surveys by Kass et al. (1998) and Warnes (2001).

In order to numerically integrate the learning LDE, it is necessary to discretize the Wiener random process. Let us denote again by η the chosen (constant) step-size: A time-discretization of the stochastic gradient (13) may be written as

$$\tilde{\mathbf{V}}_{\text{diff},n} = \tilde{\mathbf{V}}_n + \sqrt{\frac{2\Theta}{\eta}} \sum_{k=1}^{p(p-1)/2} \mathbf{L}_k \nu_k, \tag{17}$$

where each ν_k is a independent, identically distributed normal random variable (Higham, 2001) and the gradient $\tilde{\mathbf{V}}_n$ is given in equation (12).

Having defined the new diffusion-type gradient (and its time-discretized version), the associated stochastic flow may be locally approximated through the geodesic learning algorithm explained in Section 2.2. Also, at every learning step n, the temperature Θ_n may be decreased in order to make the diffusive disturbance term peter out after the early stages of learning. This gives rise to the following simulated-annealing/MCMC learning scheme.

◇ **Learning algorithm 3**:

1. Set $n = 0$, generate an initial solution $\mathbf{W}_0$ and set $f_0 = f(\mathbf{W}_0)$, select a constant learning step-size η, select a temperature value Θ_0 and select a $g^{O(p)}$-orthonormal base $\mathbf{L}_k$ of the Lie algebra $\mathfrak{so}(p)$.

2. Generate a set of identically-distributed, independent standard Gaussian random variables ν_k.

3. Compute the diffusive gradient (17), compute a candidate solution $\mathbf{W}_{n+1}$ through the equation (11), where the deterministic gradient is replaced by the diffusive gradient, and evaluate $f_{n+1} = f(\mathbf{W}_{n+1})$.

4. Compute the MCMC probability $\pi_{MCMC} \overset{\text{def}}{=} \min\{1, \exp(-(f_{n+1} - f_n)/\Theta_n)\}$.

5. Accept the candidate solution with probability π_{MCMC} (or reject the candidate solution with probability $1 - \pi_{MCMC}$). Rejection corresponds to assuming $\mathbf{W}_{n+1} = \mathbf{W}_n$.

6. Decrease the temperature Θ_n to Θ_{n+1} following a pre-defined cooling scheme.

7. Increment n and return to 2, unless n exceeds the maximum number of iteration permitted: In this case, exit.

4. Application to Non-Negative Independent Component Analysis: Algorithms Implementation and Numerical Experiments

The aims of the present section are to recall the concept of non-negative independent component analysis (ICA$^+$) and the basic related results, to customize the general learning algorithms on the orthogonal group to the case of ICA$^+$, and to present and discuss some numerical cases related to non-negative ICA applied to the separation of gray-level images.

4.1 Non-Negative Independent Component Analysis

Independent component analysis (ICA) is a signal/data processing technique that allows to recover independent random processes from their unknown combinations (Cichocki and Amari, 2002; Hyvärinen et al., 2001). In particular, standard ICA allows the decomposition of a random process $\mathbf{x}(t) \in \mathbb{R}^p$ into the affine instantaneous model:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \tag{18}$$

where $\mathbf{A} \in \mathbb{R}^{p \times p}$ is the *mixing* operator, $\mathbf{s}(t) \in \mathbb{R}^p$ is the *source stream* and $\mathbf{n}(t) \in \mathbb{R}^p$ denotes the disturbance affecting the measurement of $\mathbf{x}(t)$ or some nuisance parameters that are not taken into account by the linear part of the model.

The classical hypotheses on the involved quantities are that the mixing operator is full-rank, that at most one among the source signals exhibit Gaussian statistics, and that the source signals are statistically independent at any time. The latter condition may be formally stated through the complete factorization principle, which ensures that the joint probability density function of statistically independent random variables factorizes into the product of their marginal probability density functions. We also add the technical hypothesis that the sources do not have degenerate (that is, point-mass-like) joint probability density function. This implies that for example, the probability that the sources are simultaneously exactly zero is null. Under these hypotheses, it is possible to recover the sources up to (usually unessential) re-ordering and scaling, as well as the mixing operator.

Neural ICA consists in training an artificial neural network described by $\mathbf{y}(t) = \mathbf{W}(t)\mathbf{x}(t)$, with $\mathbf{y}(t) \in \mathbb{R}^p$ and $\mathbf{W}(t) \in \mathbb{R}^{p \times p}$, so that the network output signals become as statistically independent as possible.

Due to the difficulty of measuring the statistical independence of the network's output signals, several different techniques have been developed in order to perform ICA. The most common approaches to ICA are those based on working out the fourth-order statistics of the network outputs and to the minimization of the (approximate) mutual information among the network's outputs. The existing approaches invoke some approximations or assumptions in some stage of ICA-algorithm development, most of which concern the (unavailable) structure of the source's probability distribution.

As it is well-known, a linear, full-rank, *noiseless* and instantaneous model may be always replaced by an orthogonal model, in which the mixing matrix $\mathbf{A}$ is supposed to belong to $O(p)$. This result may be obtained by pre-whitening the observed signal $\mathbf{x}$, which essentially consists in removing second-order statistical information from the observed signals. When the mixture is orthogonal, the separating network's connection matrix must also be orthogonal, so we may restrict the learning process to searching the proper connection matrix within $O(p)$.

An interesting variant of standard ICA may be invoked when the additional knowledge on the non-negativity of the source signals is considered. In some signal processing situations, in fact, it is *a priori* known that the sources to be recovered have non-negative values (Plumbley, 2002, 2003). This is the case, for instance, in image processing, where the values of the luminance or the intensity of the color in the proper channel are normally expressed through non-negative integer values. Another interesting potential application is spectral unmixing in remote sensing (Keshava and Mustard, 2002). The evolution of passive remote sensing has witnessed the collection of measurements with great spectral resolution, with the aim of extracting increasingly detailed information from pixels in a scene for both civilian and military applications. Pixels of interest are frequently a combination of diverse components: In hyper-spectral imagery, pixels are a mixture of more than one distinct substance. In fact, this may happen if the spatial resolution of a sensor is so low that diverse materials can occupy a single pixel, as well as when distinct materials are combined into a homogeneous mixture. Spectral demixing is the procedure with which the measured spectrum is decomposed into a set of component spectra and a set of corresponding abundances, that indicate the proportion of each component present in the pixels. The theoretical foundations of the *non-negative independent component analysis* (ICA$^+$) have been given by Plumbley (2002), and then Plumbley (2003) proposed an optimization algorithm for non-negative ICA based on geodesic learning and applied it to the blind separation of three gray-level images. Further recent news on this topic have been published by Plumbley (2004). In our opinion, non-negative ICA as proposed by Plumbley (2003) is an interesting task and, noticeably, it also gives rise to statistical-approximation-free and parameter-free learning algorithms.

Under the hypotheses motivated by Plumbley (2002), a way to perform non-negative independent component analysis is to construct a cost function $f(\mathbf{W})$ of the network connection matrix that is identically zero if and only if the entries of network's output signal $\mathbf{y}$ are non-negative with probability 1. The criterion function chosen by Plumbley (2003) is $f : O(p) \to \mathbb{R}_0^+$ defined by

$$f(\mathbf{W}) \overset{\text{def}}{=} \frac{1}{2}\mathbb{E}_{\mathbf{x}}[\|\mathbf{x} - \mathbf{W}^T \rho(\mathbf{W}\mathbf{x})\|_2], \tag{19}$$

where $\mathbb{E}_{\mathbf{x}}[\cdot]$ denotes statistical expectation with respect to the statistics of $\mathbf{x}$, $\|\cdot\|_2$ denotes the standard L_2 vector norm and the function $\rho(\cdot)$ denotes the 'rectifier':

$$\rho(u) \overset{\text{def}}{=} \begin{cases} u \text{ , if } u \geq 0 \text{ ,} \\ 0 \text{ , otherwise.} \end{cases}$$

In the definition (19), the rectifier acts component-wise on vectors. From the definition (19), it is clear that when all the network output signals have positive values, it results $f = 0$, otherwise $f \neq 0$. The described cost function closely resembles a non-linear principal component analysis criterion designed on the basis of the minimum reconstruction error principle (Hyvärinen et al., 2001). This observation would be beneficial for future extensions to complex-weighted neural networks, as suggested by Fiori (2004).

In this case, learning a ICA$^+$ network may thus be accomplished by minimizing the criterion function f.

In order to design a gradient-based learning algorithm over the orthogonal group according to the general theory developed in the Section 2.2, it is necessary to compute the Euclidean gradient of the function (19) with respect to the connection matrix $\mathbf{W}$. After rewriting the learning criterion function as

$$2f(\mathbf{W}) = \mathbb{E}_{\mathbf{x}}[\|\mathbf{x}\|_2 + \|\rho(\mathbf{y})\|_2 - 2\mathbf{y}^T\rho(\mathbf{y})],$$

some lengthy but straightforward computations lead to the expression:

$$\mathrm{grad}_{\mathbf{W}}^{\mathbb{R}^{p\times p}} f = \mathbb{E}_{\mathbf{x}}[((\rho(\mathbf{y}) - \mathbf{y}) \diamond \rho'(\mathbf{y}))\mathbf{x}^T - \rho(\mathbf{y})\mathbf{x}^T],$$

where the symbol $\diamond$ denotes component-wise (Hadamard) product of two vectors and $\rho'(\cdot)$ denotes the derivative of the rectifier, that is, the unit-step function. This is undefined in the origin. From a practical point of view, this is a minor difficulty: In fact, thanks to the hypothesis of non-degeneracy of the joint probability density function of the source, the probability that the components of the networks output vector vanish to zero simultaneously is equal to zero. It is now easy to recognize that the vector $(\rho(\mathbf{y}) - \mathbf{y}) \diamond \rho'(\mathbf{y})$ is identically zero (where it is defined), therefore the above gradient reduces to the simple expression:

$$\mathrm{grad}_{\mathbf{W}}^{\mathbb{R}^{p\times p}} f = -\mathbb{E}_{\mathbf{x}}[\rho(\mathbf{y})\mathbf{x}^T].$$

Following the notation introduced by Plumbley (2003), we find it convenient to define the rectified network output:

$$\mathbf{y}_n^+ \stackrel{\mathrm{def}}{=} \rho(\mathbf{y}_n) \text{ , where } \mathbf{y}_n \stackrel{\mathrm{def}}{=} \mathbf{W}_n\mathbf{x}. \tag{20}$$

With this convention, the Riemannian gradient and the associate learning algorithm (valid for example, for the versions of Algorithms 1 and 2) write, respectively:

$$\begin{aligned} &2\,\mathrm{grad}_{\mathbf{W}_n}^{O(p)} f = \mathbb{E}_{\mathbf{x}}[\mathbf{y}_n(\mathbf{y}_n^+)^T\mathbf{W}_n] - \mathbb{E}_{\mathbf{x}}[(\mathbf{y}_n^+)\mathbf{x}^T], \\ &\mathbf{W}_{n+1} = \exp(\eta_n(\mathbb{E}_{\mathbf{x}}[\mathbf{y}_n(\mathbf{y}_n^+)^T] - \mathbb{E}_{\mathbf{x}}[\mathbf{y}_n^+\mathbf{y}_n^T]))\mathbf{W}_n \text{ ,} \\ &n = 1,\ 2,\ 3,\ \ldots \end{aligned}$$

The initial connection matrix $\mathbf{W}_0$ may be randomly picked in $O(p)$. Another practical choice is $\mathbf{W}_0 = \mathbf{I}_p$.

4.2 Details on the Used Data and on Algorithms Implementation

The gray-level images used in the experiments are illustrated in the Figure 1. It is important to note that, in general, real-world images are not completely statistically independent. For instance,

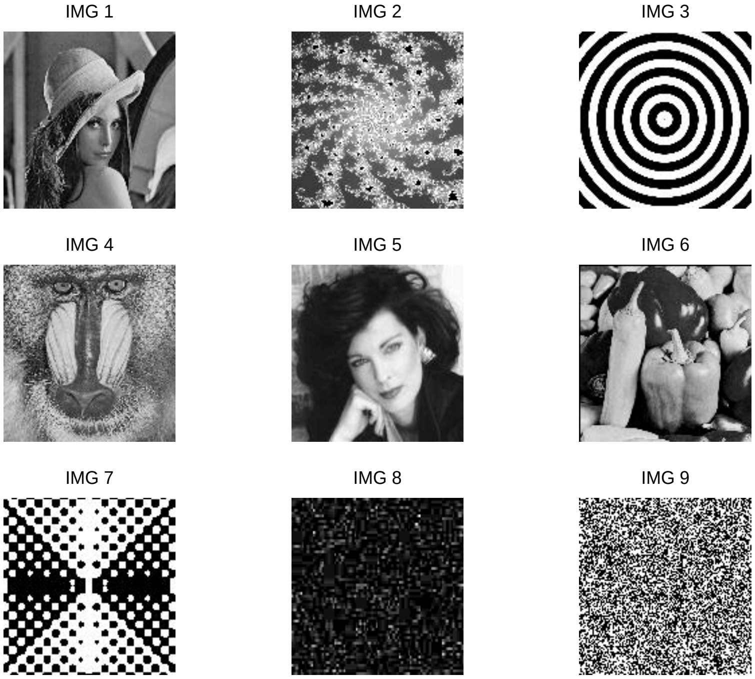

Figure 1: The nine gray-level images used in the experiments.

the images used in the present experiments are slightly statistically correlated, as can be seen by computing their 9×9 covariance matrix (approximated to two decimal digits) $\mathbf{C}_{\mathbf{s}} =$

$$10^3 \times \begin{bmatrix} 2.81 & 0.07 & 0.1 & -0.05 & -0.33 & -0.55 & 0.29 & -0.04 & -0.12 \\ 0.07 & 4.52 & 0 & -0.04 & 0.61 & 0.49 & -0.16 & 0.01 & -0.02 \\ 0.1 & 0 & 15.05 & 0.33 & 0.14 & 0.06 & -0.37 & -0.09 & 0.01 \\ -0.05 & -0.04 & 0.33 & 2.32 & 0.17 & 0.38 & -0.43 & 0 & -0.09 \\ -0.33 & 0.61 & 0.14 & 0.17 & 5.49 & 0.67 & 0.8 & 0.01 & 0.02 \\ -0.55 & 0.49 & 0.06 & 0.38 & 0.67 & 5.69 & -0.63 & -0.04 & -0.04 \\ 0.29 & -0.16 & -0.37 & -0.43 & 0.8 & -0.63 & 15.3 & -0.01 & 0.12 \\ -0.04 & 0.01 & -0.09 & 0 & 0.01 & -0.04 & -0.01 & 0.89 & -0.01 \\ -0.12 & -0.02 & 0.01 & -0.09 & 0.02 & -0.04 & 0.12 & -0.01 & 15.33 \end{bmatrix},$$

which is not diagonal, but diagonal-dominated.

It is now necessary to explain in details the pre-whitening algorithm. We distinguish between the noiseless and noisy case.

- In the noiseless case (namely, $\mathbf{n}(t) \equiv 0$), the pre-whitening stage is based on the observation that in the model (18) the square matrix $\mathbf{A}$ may be written through the singular value decomposition (SVD) as $\mathbf{F}_1\mathbf{D}\mathbf{F}_2^T$, where $\mathbf{F}_1, \mathbf{F}_2 \in O(p)$ and $\mathbf{D} \in \mathbb{R}^{p\times p}$ is diagonal invertible. Then, it is readily verified that $\mathbf{C}_{\mathbf{x}} \overset{\text{def}}{=} \mathbb{E}_{\mathbf{x}}[\overline{\mathbf{x}}\overline{\mathbf{x}}^T] = \mathbf{A}\mathbb{E}_{\mathbf{s}}[\overline{\mathbf{s}}\overline{\mathbf{s}}^T]\mathbf{A}^T$, where the overline denotes centered signals (for example, $\overline{\mathbf{x}} \overset{\text{def}}{=} \mathbf{x} - \mathbb{E}_{\mathbf{x}}[\mathbf{x}]$.) In the (non-restrictive) hypothesis that $\mathbb{E}_{\mathbf{s}}[\overline{\mathbf{s}}\overline{\mathbf{s}}^T] = \mathbf{I}_p$, we thus have $\mathbf{C}_{\mathbf{x}} = \mathbf{A}\mathbf{A}^T = \mathbf{F}_1\mathbf{D}^2\mathbf{F}_1^T$. The factors $\mathbf{F}_1$ and $\mathbf{D}$ may thus be computed through the standard eigenvalue decomposition of the covariance $\mathbf{C}_{\mathbf{x}}$. The whitened observation signal is then

$$\hat{\mathbf{x}} \overset{\text{def}}{=} \mathbf{D}^{-1}\mathbf{F}_1^T\mathbf{x} = \mathbf{F}_2^T\mathbf{s}.$$

It is now clear that the last rotation $\mathbf{F}_2^T$ of the source signals cannot be removed by second-order statistics, while orthogonal non-negative ICA may be effective to separate out the independent/non-negative components.

- In the noisy case, when the model of the observed signal is given by (18), the noise component cannot be filtered out by using pre-whitening nor independent component analysis itself. However, pre-whitening still makes it possible to use orthogonal ICA$^+$, provided *the additive noise affecting the observations is not too strong*. In fact, by hypothesizing the noise component $\mathbf{n}(t)$ is a zero-mean multivariate random sequence with covariance matrix $\sigma^2\mathbf{I}_p$, termed 'spherical' noise, the covariance of the observations writes $\mathbf{C}_{\mathbf{x}} = \mathbf{A}\mathbf{A}^T + \sigma^2\mathbf{I}_p$. In case of strong disturbance, it is therefore clear that, in general, pre-whitening cannot rely on eigenvalue decomposition of $\mathbf{C}_{\mathbf{x}}$. In any case, the difficulty due to the presence of strong additive noise is theoretically unavoidable, even if pre-whitening is dispensed of and ICA algorithms that search in $Gl(p)$ are employed.

In order to compute a separation performance index, we consider that, at convergence, the separation product $\mathbf{P}_n \overset{\text{def}}{=} \mathbf{W}_n\mathbf{D}^{-1}\mathbf{F}_1^T\mathbf{A} \in \mathbb{R}^{p\times p}$ should ideally exhibit only one entry per row (or column) different from zero, while the magnitude of non-zero values does not care. In a real-word situation, of course some residual interference should be tolerated. Therefore, a valid separation index is

$$Q_n \overset{\text{def}}{=} \frac{1}{p}\|\mathbf{P}_n\mathbf{P}_n^T - \text{diag}(\mathbf{P}_n\mathbf{P}_n^T)\|_{\text{F}}, \tag{21}$$

where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm. The index above is based on the fact that ideally the matrix $\mathbf{P}_n\mathbf{P}_n^T$ should be diagonal, therefore Q_n measures the total off-diagonality averaged over the total number of network's outputs. (As normally the index Q_n assumes very low values, it is worth normalizing it to its initial value, namely by Q_n/Q_0.)

Another valid network-performance index is the criterion function (19) itself. For easy computation of the index, we note that by defining $\mathbf{y}_n^- \overset{\text{def}}{=} \mathbf{W}_n\mathbf{x} - \rho(\mathbf{W}_n\mathbf{x})$, the value of the cost function at the n-th learning step computes as $f_n = \frac{1}{2}\mathbb{E}_{\mathbf{x}}[\|\mathbf{y}_n^-\|_2]$. (The learning algorithm seeks for a neural transformation that minimizes the negativity of its outputs, in fact.)

With regard to the computational complexity analysis of the described algorithms, we consider the number of floating-point operations (flops) per iteration and the average run-time per iteration. The codes were implemented in MATLAB on a 600 MHz, 128 MB platform.

With regard to the selection of the schedule η_n, in the experiments we found it convenient to write first the learning step-size η_n as $\tilde{\eta}_n/\|\tilde{\mathbf{V}}_n\|_{\text{F}}$, where $\tilde{\mathbf{V}}_n$ denotes again the gradient on the Lie algebra of $O(p)$ defined in the equations (12) and then to optimize the normalized step-size $\tilde{\eta}_n$. This convention keeps valid throughout the remaining part of the paper, so we can continue to use the notation η_n even for the normalized step-size without confusion.

In order to establish a numerically efficient geodesic search for the Algorithm 2, we seek for the optimal η_n in a suitable interval by sampling this interval at sub-intervals of proper size. The details on these quantities are given in the section dedicated to the numerical experiments for each category of experiment.

About the cooling scheme for the simulated-annealing/MCMC algorithm, according to Liu et al. (2004), we adopted the schedule $\Theta_{n+1} = \Theta_n/1.025$.

As a general note, the ensemble average denoted by the statistical expectation operator $\mathbb{E}[\cdot]$ is replaced everywhere by sample (empirical) mean.

4.3 Results of Numerical Experiments

The present part of the paper aims at presenting some numerical results obtained with the above-described learning algorithms applied to non-negative independent component analysis. The numerical analysis begins with the illustration of some toy experiments that aim at showing the consistency of the adopted 'non-negativity' optimization principle. Then, the analysis continues with an investigation and a comparison of the behavior of the three algorithms described in the previous sections.

4.3.1 Preliminary Experiments

As a case study, we consider the mixing of two images with a randomly generated mixing matrix $\mathbf{A} \in \mathbb{R}^{2\times 2}$. As the orthogonal separation matrix $\mathbf{W}$ is of size 2×2, it may be easily parameterized, as in equation (10), by

$$\mathbf{W}(\beta) = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix},$$

with $\beta \in [-\pi, \pi[$ being the separation angle. As already underlined in Section 2.3, this parameterization does not cover the whole group $O(2)$, but this problem is unessential for ICA purpose. By properly sampling the interval $[-\pi, \pi[$, it is possible to give a graphical representation of the behavior of the non-negative independent component analysis criterion $f(\mathbf{W}(\beta))$ defined in equation (19) and of the separation index $Q(\beta)$ defined by equation (21) (which depends on variable β through the separation product $\mathbf{P}$).

The results of this analysis for a randomly generated mixing matrix, with source images number 1 and 2 of Figure 1, are shown in the Figure 2. The Figure 2 shows the two-image mixtures, the behavior of the cost function f and of the separation index Q as well as the separated images obtained with the optimal separation angle, which is defined as the angle corresponding to the *minimal criterion function value*. As it clearly emerges from the above figure, the cost function has a only minimum, which coincides with one of the minima of the separation index. The minimum of the cost function corresponds to a pair of well-separated network outputs.

The result of the analysis with source images number 3 and 4 of Figure 1 are shown in the Figure 3. The Figure 3 shows the mixtures, the behavior of the cost function and of the separation index as well as the separated images. Again, the cost function exhibits a only minimum that coincides with one of the minima of the separation index, which, in turn, corresponds to a pair of well-separated non-negative independent components. This second result, compared with the previous one, illustrates the dependency of the shape of the cost function on the mixing matrix as well as on the mixed components.

To end the series of preliminary experiments, we consider here again the mixing of images 1 and 2 with a randomly generated mixing matrix $\mathbf{A}$ in the *noisy mixture case*. In particular, as anticipated in the Section 4.1, 'spherical' additive white Gaussian noise is supposed to contaminate the observations as in the original ICA model (18). The quantity that describes the relative weight of the noise in the mixture is the signal-to-noise ratio (SNR), which, in this particular case, may be

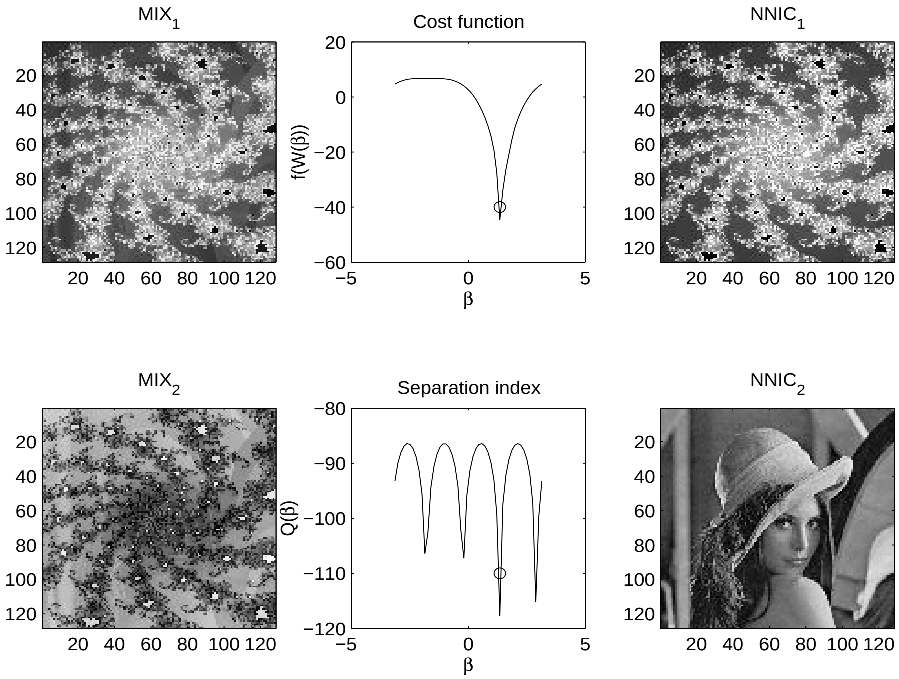

Figure 2: Images 1 and 2 mixtures (MIX_1 and MIX_2), behavior of cost function f and separation index Q (shown in dB scales) and separated images (NNIC_1 and NNIC_2) obtained with the optimal separation angle. The open circle denotes the value of the the parameter β corresponding to the minimum criterion $f(\mathbf{W}(\beta))$ value.

compactly defined as

$$SNR \overset{\text{def}}{=} 10\log_{10}\sqrt{\exp(\text{trace}\{\log[(\text{diag}(\mathbf{C_m})\text{diag}(\mathbf{C_n})^{-1})]\})},$$

where $\text{diag}(\mathbf{C_m})$ denotes the diagonal part of the 2×2 covariance matrix of the noiseless observation (term $\mathbf{As}(t)$) while $\text{diag}(\mathbf{C_n})$ denotes the diagonal part of the covariance matrix of the noise term $\mathbf{n}(t)$, referred to the ICA model (18).

The results of this analysis are shown in the Figures 4 and 5, which illustrate the behavior of the cost function f and of the separation index Q as well as the separated images obtained with the optimal separation angle, for two different noisy mixtures. In the experiment illustrated in the Figure 4, the value of the signal-to-noise ratio was $SNR = 11.64$ dB. The Figure shows that the cost function exhibits a only minimum that is quite close to one of the minima of the separation index, which, in turn, corresponds to a pair of well-separated non-negative independent components. Of course, the mixtures *as well as the recovered components* look a little noisy. In the experiment illustrated in the Figure 5, the value of the signal-to-noise ratio was $SNR = 4.18$ dB. In this experiment, the power of the disturbance is close to the power of the source-images, therefore the mixture may be considered as rather noisy. The Figure 5 shows that the cost function exhibits a only minimum that is quite far from the minima of the separation index. The neural network outputs look very noisy and do not resemble the original independent components. This result confirms the observations of Section 4.1

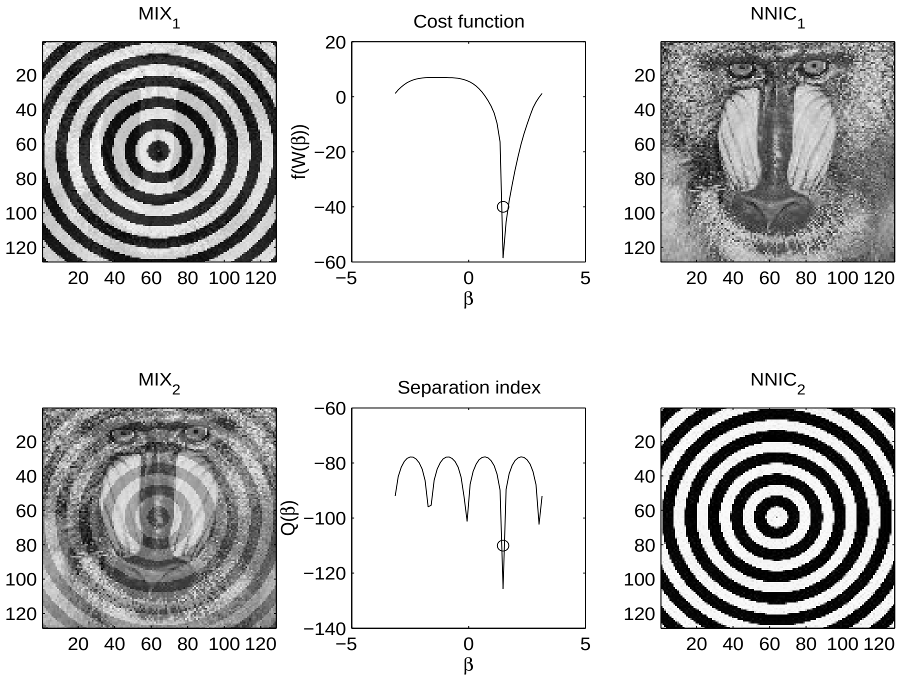

Figure 3: Images 3 and 4 mixtures (MIX$_1$ and MIX$_2$), behavior of cost function f and separation index Q (shown in dB scales) and separated images (NNIC$_1$ and NNIC$_2$) obtained with the optimal separation angle. The open circle denotes the value of the the parameter β corresponding to the minimum criterion $f(\mathbf{W}(\beta))$ value.

about the unavoidability of the problems related to the presence of strong noise in the mixture by plain ICA.

In the next sections, we shall therefore take into account noiseless mixtures, which also illustrate the behavior of the algorithm in presence of *weak* disturbances. It is in fact to be recognized that the pre-whitening/sphering issue is a different problem from optimization on $O(p)$: Noisy mixtures cannot be pre-whitened, but if the noise is weak, its presence has negligible effects on the separation performances.

4.3.2 A Further ‘Conventional’ Algorithm for Numerical Comparison Purposes

In order to gain incremental knowledge on the advantages offered by Lie-group methods via numerical comparisons, it would be beneficial to consider a ‘conventional’ learning algorithm in which the ordinary gradient and explicit orthogonalization are employed.[6] To this aim, we defined the following non-Lie-group algorithm:

$$\tilde{\mathbf{W}}_{n+1} = \mathbf{W}_n - \eta \mathbb{E}[\mathbf{y}_n^+ \mathbf{x}^T], \tag{22}$$

$$\mathbf{W}_{n+1} = (\tilde{\mathbf{W}}_{n+1} \tilde{\mathbf{W}}_{n+1}^T)^{-\frac{1}{2}} \tilde{\mathbf{W}}_{n+1}, \tag{23}$$

[6] This comparison was suggested by a reviewer.

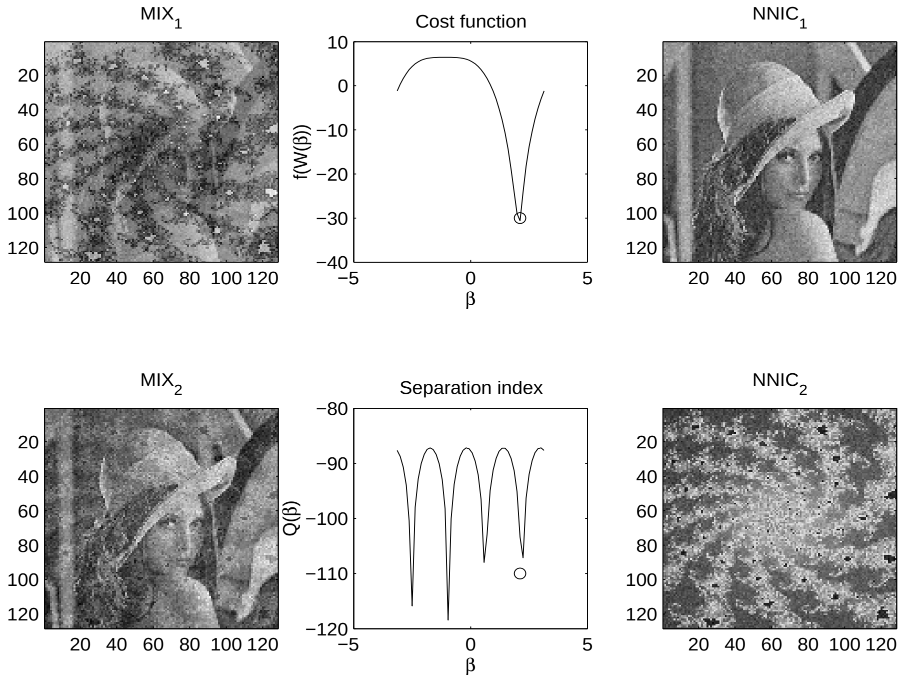

Figure 4: Images 1 and 2 weakly-noisy mixtures (MIX_1 and MIX_2), behavior of cost function f and separation index Q (shown in dB scales) and separated images (NNIC_1 and NNIC_2) obtained with the optimal separation angle. The open circle denotes the value of the the parameter β corresponding to the minimum criterion $f(\mathbf{W}(\beta))$ value.

where the rectified network output is defined as in equation (20) and with the initial connection pattern $\mathbf{W}_0 \in O(p)$ and the learning step-size $\eta < 0$ being chosen according to the same rules used with the Algorithms 1, 2 and 3. It is worth remarking that we again consider the normalization $\eta = \tilde{\eta}/\|\mathbb{E}[\mathbf{y}_n^+\mathbf{x}^T]\|$, so the actual step-size to be selected is $\tilde{\eta}$, as previously assumed for the Algorithms 1, 2 and 3.

The first line of the above algorithm moves the connection pattern at step n from the matrix $\mathbf{W}_n$ over the orthogonal group toward the direction of the Euclidean gradient of the ICA$^+$ cost function to the new point $\tilde{\mathbf{W}}_{n+1}$. However, the matrix $\tilde{\mathbf{W}}_{n+1}$ does not belong to the orthogonal group so it is necessary to project it back to the group with the help of a suitable projector (according to what granted in Section 2.1). In this case, it is assumed $\Pi : \mathbb{R}^{p\times p} \to O(p)$ as

$$\Pi(\mathbf{X}) \overset{\mathrm{def}}{=} (\mathbf{X}\mathbf{X}^T)^{-\frac{1}{2}}\mathbf{X}. \tag{24}$$

(It is straightforward to verify that $\Pi^T(\mathbf{X})\Pi(\mathbf{X}) = \mathbf{I}_p$ for all $\mathbf{X} \in Gl(p)$.) In the case of the orthogonal-group projector, the ambient space was assumed as $\mathcal{A} = \mathbb{R}^{p\times p}$. It is worth underlining that, from a theoretical point of view, there is no guarantee that the partially updated matrix $\tilde{\mathbf{W}}_{n+1}$ belongs to $Gl(p) \subset \mathbb{R}^{p\times p}$ and, therefore, there is no guarantee that the projector Π may be computed at every iteration.

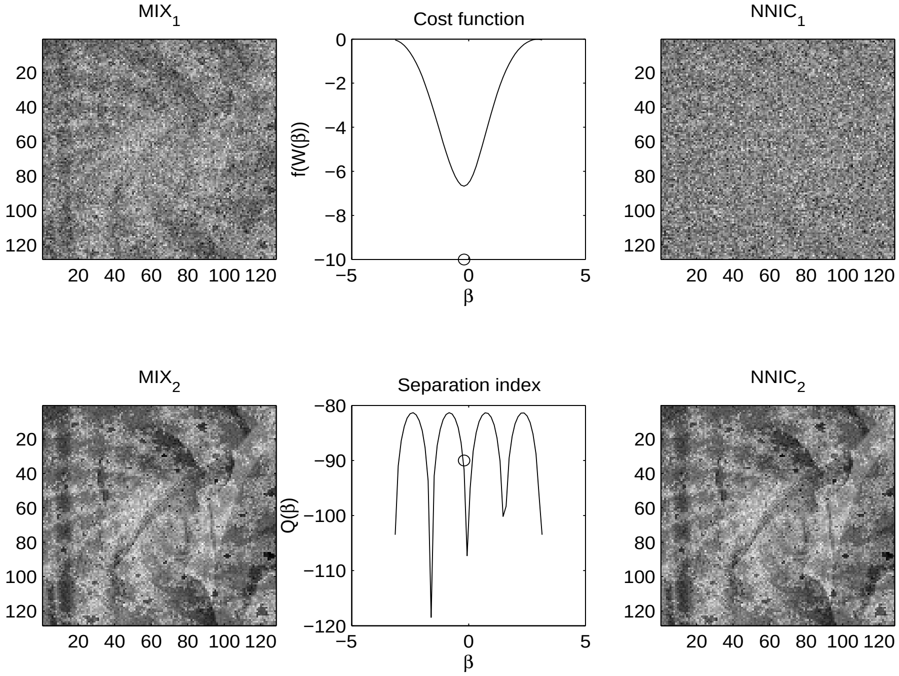

Figure 5: Images 1 and 2 strongly-noisy mixtures (MIX$_1$ and MIX$_2$), behavior of cost function f and separation index Q (shown in dB scales) and separated images (NNIC$_1$ and NNIC$_2$) obtained with the optimal separation angle. The open circle denotes the value of the the parameter β corresponding to the minimum criterion $f(\mathbf{W}(\beta))$ value.

4.3.3 Numerical Analysis and Comparison of the ICA$^+$ Algorithms

The first experiment of this section aims at investigating a 4×4 ICA$^+$ case tackled with the help of the deterministic-gradient-based algorithm endowed with geodesic search (Algorithm 2). In particular, in this case the optimal step-size is searched for within the interval $[-1, -0.1]$ partitioned into 10 bins and the random step-size generated in case of non-acceptance is a small random number uniformly picked in $[-0.1, 0[$. The maximum number of iterations has been fixed to 100 and the used images are number 1, 2, 3 and 4 of Figure 1.

The results of this experiment are shown in the Figures 6, 7 and 8.

In particular, the Figure 6 shows the behavior of the (normalized) separation index Q_n/Q_0 and of the cost function f_n versus the iteration index n. As these panels show, the separation index as well as the cost function values decrease from initial values to lower values, confirming that the separation behavior is good, in this experiment. The same Figure also shows the Frobenius norm of the Riemannian gradient $\tilde{\mathbf{V}}_n$ defined in equation (12), which decreases to low values during iteration, as well as the value of the ‘optimal’ learning step-size η_n selected at each iteration.

The Figure 7 shows a picture of the cost function as seen by the ‘geodesic search’ procedure: It shows, at each iteration, the shape of the cost function $f_n(\eta)$ as a function of the step-size η and shows the numerical minimal value to be selected as ‘optimal’ learning step-size. As explained in the description of the Algorithm 2, such value is actually selected only if the corresponding value

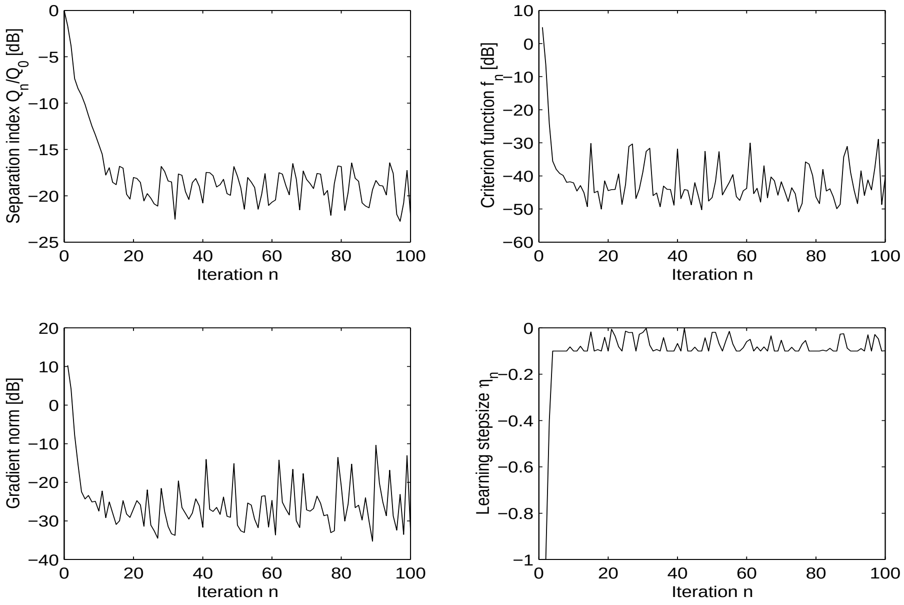

Figure 6: Four-source problem. Top-left: Normalized separation index versus the iteration index n. Top-right: Cost function f_n versus the iteration index n. Bottom-left: Norm of the Riemannian gradient of the ICA$^+$ cost function versus the iteration index n. Bottom-right: 'Optimal' learning step-size η_n selected at each iteration.

of the cost function is smaller than the value of the cost function achieved in the previous iteration, otherwise the result of the geodesic search is ignored and a small random step-size is selected. From the picture, it clearly emerges that the function $f_n(\eta)$ exhibits a only minimum in the interval of interest for η. Also, as the learning procedure progresses, the minimal value is almost always located at relatively low values of η because of the sharpness of the cost function around the optimal separating solution evidenced by the Figures 2 and 3.

The Figure 8 shows the result of this analysis for a randomly generated 4×4 mixing matrix with four source images. The de-mixing matrix is the optimal one as obtained by the learning procedure. The visual appearance of the obtained components confirms the quality of the blind recovering procedure.

The second experiment of this section aims at investigating a 9×9 ICA$^+$ case tackled with the help of the deterministic-gradient-based algorithm endowed with geodesic search (Algorithm 2). In particular, in this case the optimal step-size is searched for within the interval $[-2, -0.1]$ partitioned into 10 bins and the random step-size generated in case of non-acceptance is a small random number uniformly picked in $[-0.1, 0[$. The maximum number of iterations has been fixed to 200. The results of this experiment are shown in the Figures 9 and 10. In this experiment, the separated images have been recovered sufficiently faithfully.

The same separation problem was also tackled through the deterministic-gradient-based algorithm without geodesic search (Algorithm 1). From the previous experiment, it emerges that the 'optimal' value of the step-size is almost always selected within the interval $[-0.1, 0[$. Therefore, in this experiment, the learning step-size was fixed to -0.05 and the number of iterations was set

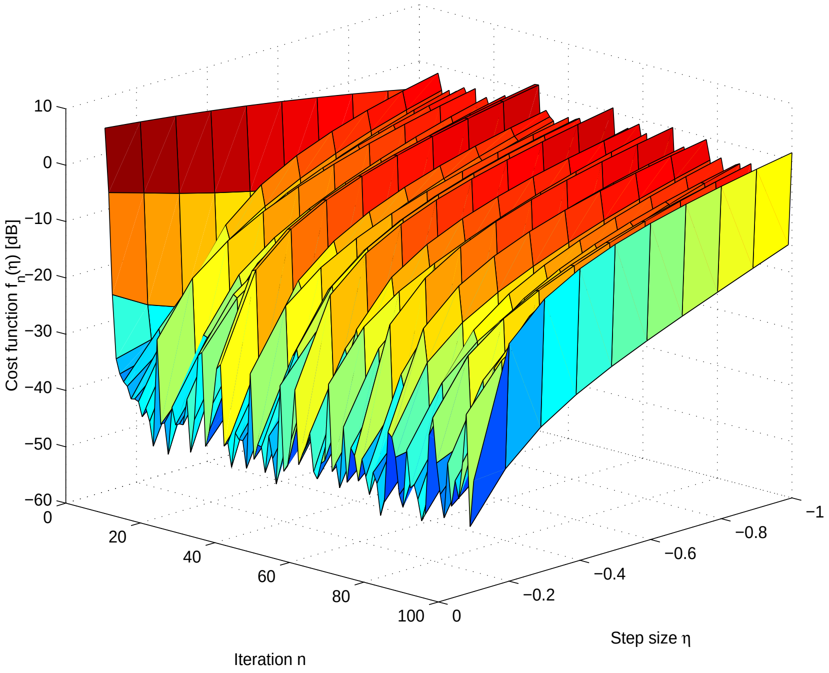

Figure 7: Four-source problem. Shape of the ICA$^+$ cost function as seen by the 'geodesic search' procedure.

to 400. It is worth noting that, in this case, not only the learning step-size was set to a constant value, but every move in the parameter manifold is accepted without checking if it actually leads to a decrease of the value of the learning criterion. The objective results of this experiment are shown in the Figure 11, while the resulting recovered components are not shown because they are similar to those illustrated in the Figure 10.

The nine-source separation problem was also tackled through the diffusion-type gradient-based algorithm (Algorithm 3). In this case, the learning step-size was set to -0.1, the initial temperature was set to $\Theta_0 = 0.5$ and the number of iterations was set to 400. The objective results of this experiment are shown in the Figure 12, while the resulting components are not shown because they are similar to those illustrated in the Figure 10.

As mentioned in Section 4.3.2, the behavior of Algorithms 1, 2 and 3 may be compared to the behavior of a non-Lie-group algorithm based on explicit orthogonalization via projection. Therefore, the nine-source separation problem was also tackled through the projection-based learning algorithm. In this case, the number of iterations was set to 400. The obtained results are not comforting about the suitability of this algorithm to non-negative independent component analysis. In spite that several values of the learning step-size were tried (ranging from -0.5 to -0.005), no good results were obtained, in this case. Two possible explanations of the observed behavior are that:

- The projection operation wastes the most part of the time in canceling out the component of the Euclidean gradient that is normal to the manifold instead of advancing the solution toward the most appropriate direction.

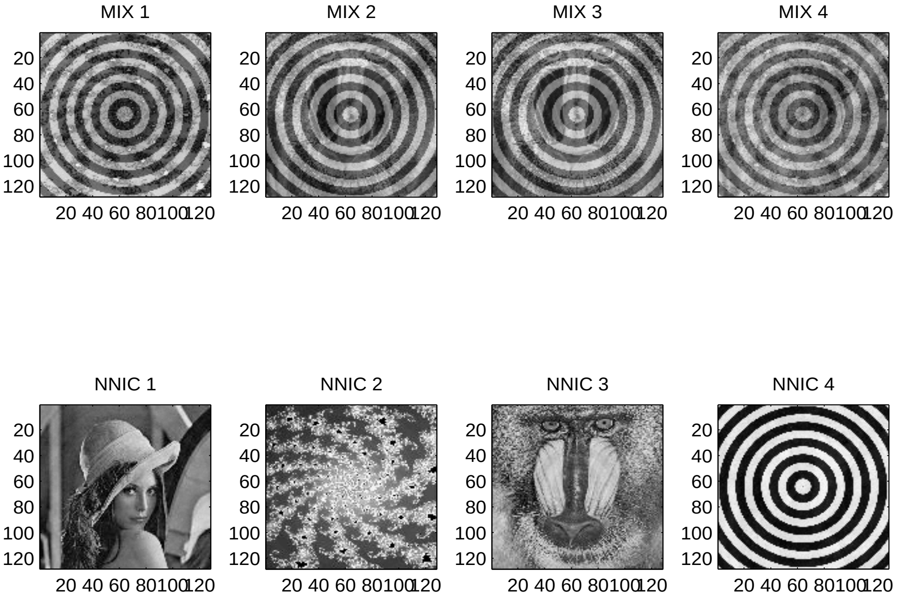

Figure 8: Four-source problem. Mixtures and separated images.

ALGORITHM	AVERAGE RUN-TIME (SEC.S)	FLOPS PER ITERATION
Algorithm 1	0.27	5.46×10^6
Algorithm 2	1.83	3.69×10^7
Algorithm 3	0.27	4.76×10^6
Projection	0.29	5.48×10^6

Table 1: Nine-source problem. Computational complexity comparison of Algorithms 1, 2, 3 and the projection-based learning algorithm (in terms of flops and run-time per iteration).

- The algorithm described by equations (22) and (23) looks essentially as fixed-point algorithm: Such kind of algorithms may easily get trapped in non-converging or very-slowly-converging cycles if the operator that describes the fixed-point iteration is not contractive. However, proving (or forcing) the convergence of such algorithms is far from being an easy task. A short discussion on this topic has been recently presented by Fiori (2002).

With regard to the computational complexity comparison of the algorithms on the nine-source separation problem, the number of flops per iteration and the average run-times per iteration are reported in the Table 1. It is worth underlining that both run-times and flop-counts depend on the platform and on the specific implementation of the algorithms, therefore only differences than span one or more magnitude orders should be retained as meaningful.

The conclusion of the above numerical analysis pertaining to the nine-source problem is quite straightforward: In the present problem, the adoption of the diffusion-type gradient is not beneficial as the initial ‘burn-in’ stage due to MCMC is quite long and the final achieved result is completely comparable to those exhibited by the other two algorithms. Among Algorithms 1 and 2, they achieve

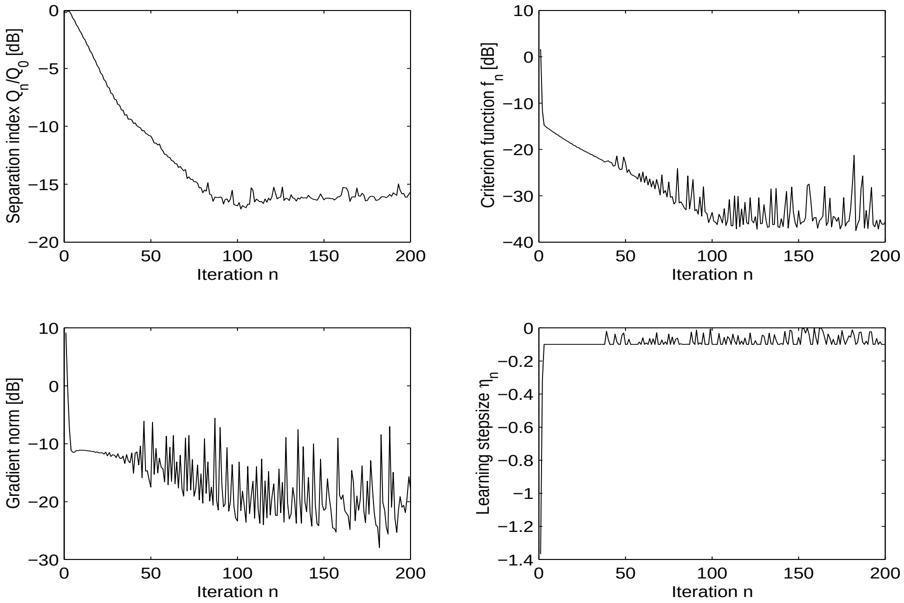

Figure 9: Nine-source problem, Algorithm 2. Top-left: Normalized separation index versus the iteration index n. Top-right: Cost function f_n versus the iteration index n. Bottom-left: Norm of the Riemannian gradient of the ICA$^+$ cost function versus the iteration index n. Bottom-right: 'Optimal' learning step-size η_n selected at each iteration.

comparable separation results, but the Algorithm 1 is definitely lighter, in terms of computational complexity, than the Algorithm 2. The computational complexity pertaining to the projection-based algorithm is comparable to the complexity exhibited by Algorithms 1 and 3.

5. Conclusion

The aim of the present tutorial was to illustrate learning algorithms based on Riemannian-gradient-based criterion optimization on the Lie group of orthogonal matrices. Although the presented differential-geometry-based learning algorithms have so far been mainly exploited in narrow contexts they may aid the design of general-purpose learning algorithms in those cases where a learning task may be formulated as an optimization one over a smooth manifold. The considered algorithms have been applied to non-negative independent component analysis both in the standard version equipped with geodesic-line search and in the diffusion-type gradient version.

The analytical developments evidenced the following advantages and similarities of the $O(p)$-type learning algorithm with respect to the existing $Gl(p)$-type algorithms:

- In the general case, the search for a connection pattern should be performed in the Lie group $Gl(p)$, while in the second case the search is performed in the orthogonal Lie group $O(p)$. The group $O(p)$ is compact (that is, closed and limited) therefore the stability of a $O(p)$-type learning algorithm is inherently ensured (up to machine precision), while this is not true for the $Gl(p)$-type learning algorithms.

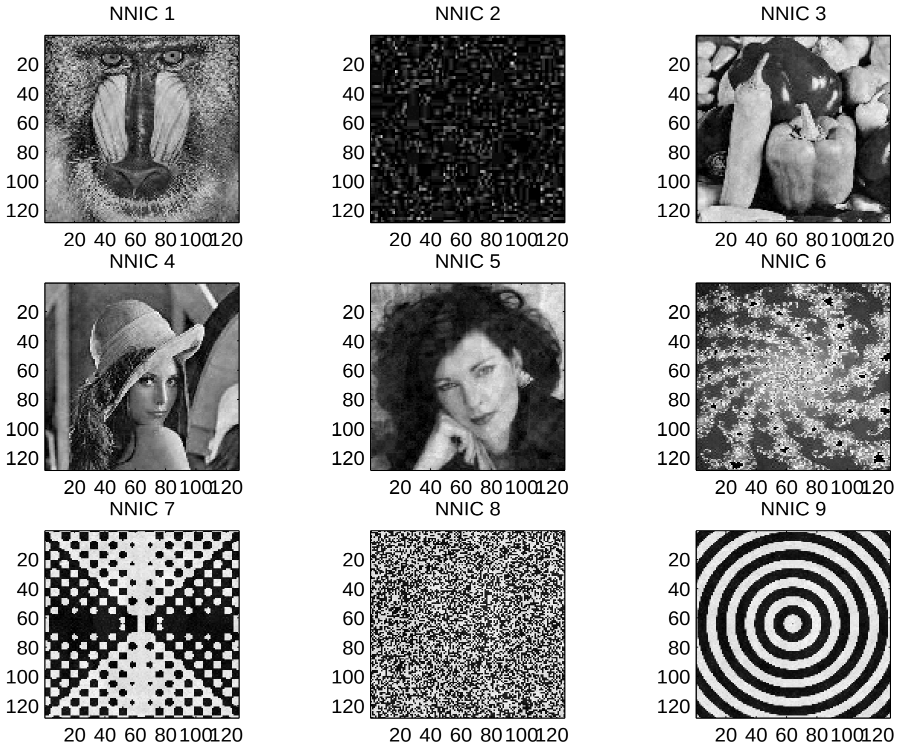

Figure 10: Nine-source problem, Algorithm 2. Separated images.

- In general, the $Gl(p)$-type learning algorithms cannot avoid quasi-degeneracy of the neural network, that is the case in which more than one neuron nearly happen to encode the same feature. In the context of $O(p)$-type learning algorithms, this case is inherently impossible.

- The possible amplification of the additive noise in the noisy ICA case is not avoided by the $O(p)$-type learning algorithms, even if care should be taken in this context of properly computing the pre-whitening operator. Even the $Gl(p)$-type learning algorithms, that do not require pre-whitening, cannot avoid the amplification of the disturbance on the observations.

The conclusions of the comparative analysis pertaining to the nine-source ICA problem are quite straightforward: The simple gradient adaptation, with a properly chosen learning step-size, is sufficient to achieve good separation performance at low computational burden. It deserves to remark, however, that the 'geodesic search' procedure automatically provides a suitable value of the learning step-size, which should be manually selected in absence of any tuning procedure.

It is worth underlining that the Algorithm 1, which appears to be the solution of choice in the context of ICA problem, as well as Algorithms 2 and 3, has been derived in a framework that is more general than ICA, but has only been applied it to ICA in the present manuscript. In the ICA$^+$ context, and with the chosen metric for the orthogonal group, the Algorithms 1 and 2 essentially coincide to the algorithms presented by Plumbley (2003). With respect to the work of Plumbley (2003), the conclusion we draw from the presented numerical analysis on ICA$^+$ problems is that, for general high-dimensional ICA$^+$ problems, the introduction of geodesic-search is not beneficial.

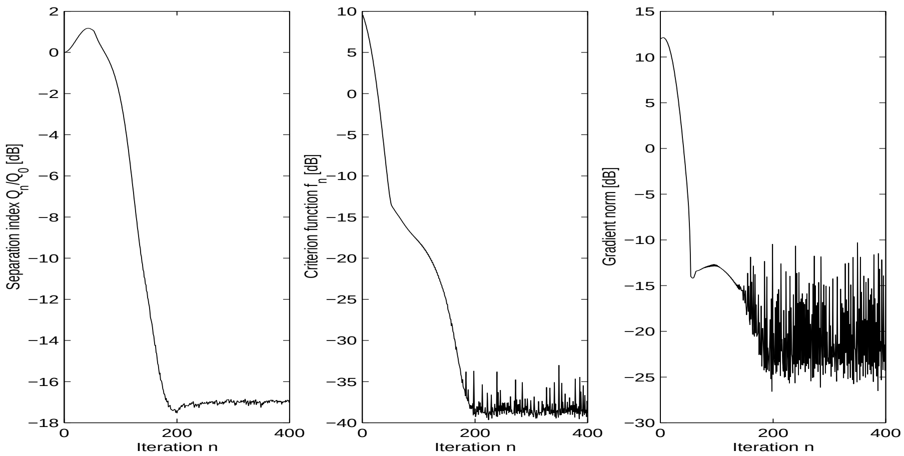

Figure 11: Nine-source problem, Algorithm 1. Left panel: Normalized separation index versus the iteration index n. Middle panel: Cost function f_n versus the iteration index n. Right: Norm of the Riemannian gradient of the ICA$^+$ cost function versus the iteration index n.

The same holds for the introduction of stochasticity under the form of annealed MCMC, that does not helped speeding up network learning convergence in the considered analysis.

About further and future efforts, we believe the following notes are worth mentioning:

- As a general remark on the computational complexity of the discussed algorithms, it is worth noting that the most burdensome operation is the computation of the exponential map in the updating rule (11). In the present paper we employed MATLAB's 'expm' primitive but, of course, several ways are known in the scientific literature to compute exponential maps. Two examples are the Cayley transform and the canonical coordinates of the first kind (interested readers might consult, for example, Celledoni and Fiori (2004) and references therein). A promising alternative solution would be to exploit the latest advancements in the field of numerical calculus on manifold for exponential maps computation, which should allegedly lead to a considerable saving of computational effort without detriment of separation effectiveness.

- As mentioned in the Section 2.1, learning algorithms based on the ordinary gradient and explicit orthogonalization (projection) are known in the scientific literature. The issue whether Lie-group methods are more advantageous, compared to methods based on the projection to the feasible set by orthogonalization, is currently being investigated.

- As it also emerges from Section 2.1, all the learning equations/algorithms developed in this manuscript are based on a particular choice of the metric that turns the Lie-algebra associated to the Lie group of orthogonal matrices into a metric space. Although, in principle, the choice of the metric may be shown not to affect the final result of learning, nor should it affect the learning path over the base-manifold, preliminary experiments suggest that the choice of metric indeed affects the behavior of discrete-time algorithms when implemented on a computer due to accumulation of numerical errors (Fiori, 2005).

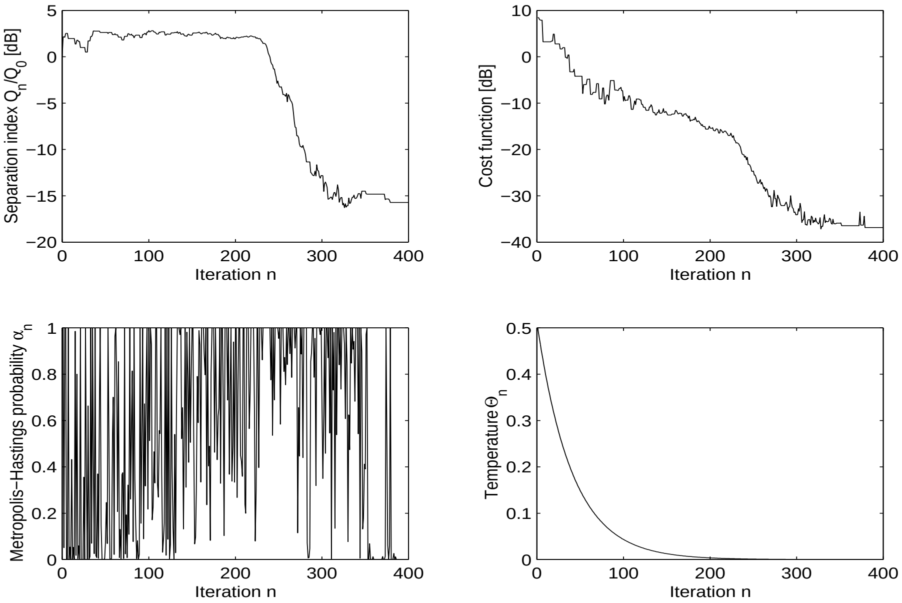

Figure 12: Nine-source problem, Algorithm 3. Top-left: Normalized separation index versus the iteration index n. Top-right: Cost function f_n versus the iteration index n. Bottom-left: Metropolis-Hastings probability α_n versus the iteration index n. Bottom-right: Simulated-annealing temperature Θ_n versus the iteration index n.

Acknowledgments

The author wishes to gratefully thank Elena Celledoni and Yasunori Nishimori for many insightful discussions on the differential geometry of the orthogonal group, Andrzej Cichocki for an insightful discussion on the noisy-mixture ICA case, Toshihisa Tanaka for kindly making it available the images used in the presented experiments, Mark Plumbley for kindly sharing viewpoints and codes on ICA$^+$ and Hichem Snoussi for the insightful discussions and useful suggestions about the theory and implementation of the MCMC sampling method.

The author also wishes to gratefully thank the JMLR Action Editor who handled the submission of this paper, Yoshua Bengio, and the anonymous reviewers, for providing careful and detailed comments that helped improving the clarity and thoroughness of the presented scientific material.

The penultimate version of this manuscript was prepared while the author was a visiting researcher at the Faculty of Information Technology, Mathematics and Electrical Engineering of Norwegian University of Science and Technology (Trondheim, Norway) during January-February 2005. The author wishes to gratefully thank Elena Celledoni for making this fruitful visit be possible.

The last version of this manuscript was prepared while the author was a visiting professor at the Laboratory for Signal and Image Processing of the Tokyo University of Agriculture and Technology (Tokyo, Japan) during March-April 2005. The author wishes to gratefully thank Toshihisa Tanaka for making this fruitful visit be possible.

Appendix A. Geodesic Equation and Relevant Properties

In the present appendix, we consider the problem of constructing a geodesic curve on a Riemannian manifold $(\mathcal{M},g)$ and illustrate some relevant properties of geodesics on Riemannian manifolds embedded in a Euclidean ambient space $\mathbb{R}^p$. The result of the following calculation will be a second-order differential equation in the components x_k of x ($k=1,\ 2,\ \cdots,\ p$).[7]

Before considering the problem of geodesic calculation, it is instrumental to consider the general variational problem of minimizing the functional:

$$A \overset{\text{def}}{=} \int_{t_0}^{t_1} H(x,\dot{x})dt, \tag{25}$$

where $H : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is a potential function, $x=x(t)$ is a curve on $\mathcal{M}$ with parameter $t \in [t_0\ ,\ t_1]$ and A is an integral functional of $x(t)$ (sometimes termed *action*). In the above equation and thereafter, overdots denote derivation with respect to the parameter t.

It is know that, under proper conditions, the solution of the above variational problem is given by the solution of the Euler-Lagrange equation:

$$\frac{\partial H}{\partial x_k} - \frac{d}{dt}\frac{\partial H}{\partial \dot{x}_k} = 0\ ,\ k=1,\ 2,\ \ldots,\ p.$$

By comparing the equation (25) and the curve-length equation (2), it is readily seen that, in order to set a curve-length minimization problem into an action minimization problem, it suffices to set $H(x,\dot{x}) = \sqrt{g_x(\dot{x},\dot{x})}$ in the above setting. To this purpose, it is worth noting that, thanks to the bi-linearity of the scalar product and according to the decomposition $\dot{x} = \sum_i \dot{x}_i e_i$, where $\{e_i\}$ denotes whatever basis of $\mathbb{R}^p$, it holds $g_x(\dot{x},\dot{x}) = \sum_i \sum_j g_{ij}\dot{x}_i\dot{x}_j$, where the functions $g_{ij} \overset{\text{def}}{=} g_x(e_i,e_j)$ denote the components of the so-termed *metric tensor* and specify completely the metric properties of the manifold $\mathcal{M}$. The components of the metric tensor are functions of the coordinates $x_1,\ \cdots,\ x_p$. The metric tensor is symmetric, that is, $g_{ij}=g_{ji}$ for every $i,j \in \{1,\ 2,\ \ldots,\ p\}$ and non-singular, that is its inverse exists everywhere.

By replacing the above expression of the potential into the Euler-Lagrange equation and calculating the required derivatives, we get

$$\sum_i \sum_j \frac{\partial g_{ij}}{\partial x_k}\dot{x}_i\dot{x}_j - 2\sum_i \frac{dg_{ik}}{dt}\dot{x}_i - 2\sum_i g_{ik}\ddot{x}_i = 0.$$

Now, the following identities are of use:

$$\sum_i \frac{dg_{ik}}{dt}\dot{x}_i = \sum_i \sum_\ell \frac{\partial g_{ik}}{\partial x_\ell}\dot{x}_i\dot{x}_\ell = \sum_i \sum_\ell \frac{\partial g_{\ell k}}{\partial x_i}\dot{x}_i\dot{x}_\ell,$$

because the indices i and ℓ may be swapped in the second-last expression. Then the equation of minimizing curve becomes:

$$\sum_i g_{ik}\ddot{x}_i + \frac{1}{2}\sum_i \sum_j \frac{\partial g_{ik}}{\partial x_j}\dot{x}_i\dot{x}_j + \frac{1}{2}\sum_i \sum_j \frac{\partial g_{jk}}{\partial x_i}\dot{x}_i\dot{x}_j - \frac{1}{2}\sum_i \sum_j \frac{\partial g_{ij}}{\partial x_k}\dot{x}_i\dot{x}_j = 0.$$

[7] In the present paper, we do not make use of the standard covariant/contra-variant notation for tensor indices nor of the Einstein convention for summations.

It is now worth introducing the inverse of the metric tensor, whose elements are denoted by g^{ab}, defined by the equations $\sum_b g^{ab} g_{bc} = \delta^a_c$, where δ^a_c denotes the fundamental tensor (and may be regarded as a Kronecker 'delta'). By multiplying both sides of the above equation by $g^{\ell k}$ and summing with respect to k, the result is

$$\sum_k \sum_i g_{ik} g^{k\ell} \ddot{x}_i + \frac{1}{2} \sum_k \sum_i \sum_j g^{k\ell} \left(\frac{\partial g_{ik}}{\partial x_j} + \frac{\partial g_{jk}}{\partial x_i} - \frac{\partial g_{ij}}{\partial x_k} \right) \dot{x}_i \dot{x}_j = 0.$$

Let us further define the Christoffel (or affine connection) coefficients as

$$\Gamma^k_{ij} \overset{\text{def}}{=} \frac{1}{2} \sum_\ell g^{k\ell} \left(\frac{\partial g_{i\ell}}{\partial x_j} + \frac{\partial g_{j\ell}}{\partial x_i} - \frac{\partial g_{ij}}{\partial x_\ell} \right),$$

through which the geodesic equation assumes the classical expression:

$$\ddot{x}_k + \sum_i \sum_j \Gamma^k_{ij} \dot{x}_i \dot{x}_j = 0 \ , \ k = 1 \ , 2 \ , \ \cdots \ , \ p. \tag{26}$$

As anticipated, it appears under the form of a set of second-order differential equations in the coordinates x_k and needs therefore two boundary conditions. These may specify the geodesic endpoints: $x(t_0) = x_0 \in \mathcal{M}$ and $x(t_1) = x_1 \in \mathcal{M}$, or the initial position and initial velocity: $x(t_0) = x_0 \in \mathcal{M}$ and $\dot{x}(t_0) = \mathbf{v}_0 \in T_{x_0}\mathcal{M}$.

A result we make use of in the paper is that, when a Riemannian manifold is embedded into an Euclidean space, the second derivative of the geodesic ($\ddot{x}$) belongs to the normal space to the embedded manifold at x. Let us begin the proof of this important property by proving that, along a geodesic, the quantity $g_x(\dot{x}, \dot{x})$ is constant with respect to the parameter t or, equivalently, that $\frac{d}{dt} g_x(\dot{x}, \dot{x}) = 0$. We have

$$\begin{aligned}
\frac{d}{dt} g_x(\dot{x}, \dot{x}) &= \frac{d}{dt} \sum_a \sum_b g_{ab} \dot{x}_a \dot{x}_b \\
&= \sum_a \sum_b \left(g_{ab} \ddot{x}_a \dot{x}_b + g_{ab} \dot{x}_a \ddot{x}_b + \frac{dg_{ab}}{dt} \dot{x}_a \dot{x}_b \right) \\
&= 2 \sum_a \sum_b g_{ab} \ddot{x}_a \dot{x}_b + \sum_a \sum_b \frac{dg_{ab}}{dt} \dot{x}_a \dot{x}_b.
\end{aligned}$$

By replacing the expression of $\ddot{x}_a$ from the geodesic equation (26) into the last expression, we get

$$\frac{d}{dt} g_x(\dot{x}, \dot{x}) = -2 \sum_a \sum_b \sum_i \sum_j g_{ab} \Gamma^a_{ij} \dot{x}_b \dot{x}_i \dot{x}_j + \sum_a \sum_b \frac{dg_{ab}}{dt} \dot{x}_a \dot{x}_b.$$

Now, the following identity holds:

$$\sum_b g_{ab} \Gamma^b_{ij} = \frac{1}{2} \left(\frac{\partial g_{ia}}{\partial x_j} + \frac{\partial g_{ja}}{\partial x_i} - \frac{\partial g_{ij}}{\partial x_a} \right),$$

thus it may be further written:

$$\begin{aligned}
\frac{d}{dt} g_x(\dot{x}, \dot{x}) &= -\sum_a \sum_i \sum_j \frac{\partial g_{ia}}{\partial x_j} \dot{x}_j \dot{x}_i \dot{x}_a - \sum_a \sum_i \sum_j \frac{\partial g_{ja}}{\partial x_i} \dot{x}_i \dot{x}_j \dot{x}_a \\
&+ \sum_a \sum_i \sum_j \frac{\partial g_{ij}}{\partial x_a} \dot{x}_a \dot{x}_i \dot{x}_j + \sum_a \sum_b \frac{dg_{ab}}{dt} \dot{x}_a \dot{x}_b.
\end{aligned}$$

It is readily recognized that, e.g., $\sum_j \frac{\partial g_{ia}}{\partial x_j}\dot{x}_j = \frac{dg_{ia}}{dt}$, therefore, all the sums in the above expression are equal up to proper index re-ordering/renaming. As a consequence, the four terms sum up to zero.

The last step consists in recalling that the manifold has been supposed to be embedded in a Euclidean ambient space and we assume $g_x(\dot{x},\dot{x}) \overset{\text{def}}{=} \dot{x}^T\dot{x}$. Its derivative is thus $\frac{d}{dt}g_x(\dot{x},\dot{x}) = 2\ddot{x}^T\dot{x} = 0$, which proves that, under the specified conditions, the second derivative $\ddot{x}$ is orthogonal to the first derivative $\dot{x}$ in any point of the embedded geodesic.

References

S.-i. Amari. *Differential-Geometrical Methods in Statistics*. Lecture Notes in Statistics 28, Springer-Verlag, 1989.

S.-i. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

T. Akuzawa. New fast factorization method for multivariate optimization and its realization as ICA algorithm. In *Proceedings of the 3$^{\text{rd}}$ International Conference on Independent Component Analysis and Blind Signal Separation* pages 114–119, San Diego, California, USA, 2001

E. Celledoni and S. Fiori. Neural learning by geometric integration of reduced ‘rigid-body’ equations. *Journal of Computational and Applied Mathematics*, 172(2):247–269, 2004.

A. Cichocki and S.-i. Amari. *Adaptive Blind Signal and Image Processing*, J. Wiley & Sons, 2002

S. Fiori. A theory for learning by weight flow on Stiefel-Grassman manifold. *Neural Computation*, 13(7):1625–1647, 2001.

S. Fiori. A theory for learning based on rigid bodies dynamics. *IEEE Trans. on Neural Networks*, 13(3):521–531, 2002.

S. Fiori. A fast fixed-point neural blind deconvolution algorithm. *IEEE Trans. on Neural Networks*, 15(2):455–459, 2004.

S. Fiori. Non-linear complex-valued extensions of Hebbian learning: An essay. *Neural Computation*, 17(4):779–838, 2005.

S. Fiori. Formulation and integration of learning differential equations on the Stiefel manifold. *IEEE Trans. on Neural Networks*, forthcoming.

S. Fiori and S.-i. Amari. Editorial: Special issue on “Geometrical Methods in Neural Networks and Learning”, *Neurocomputing*, forthcoming.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

A. Hyvärinen, J. Karhunen and E. Oja. *Independent Component Analysis*, John Wiley & Sons, 2001.

D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001.

R. E. Kass, B. P. Carlin, A. Gelman and R. M. Neal. Markov Chain Monte Carlo in practice: A roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57, 2002.

X. Liu, A. Srivastava and K. Gallivan. Optimal linear representation of images for object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5):662–666, 2004.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

Y. Nishimori. Learning algorithm for ICA by geodesic flows on orthogonal group. In *Proc. of the International Joint Conference on Neural Networks* pages 1625–1647, 1999.

P. J. Olver. Applications of Lie groups to differential equations. *Graduate Texts in Mathematics 107*, Second Edition, Springer, 2003.

H. Park, S.-i. Amari and K. Fukumizu. Adaptive Natural Gradient Learning Algorithms for Various Stochastic Models. *Neural Networks*, 13:755–764, 2000.

M. D. Plumbley. Conditions for non-negative independent component analysis. *IEEE Signal processing Letters*, 9(6):177–180, 2002.

M. D. Plumbley. Algorithms for nonnegative independent component analysis. *IEEE Trans. on Neural Networks*, 14(3):534–543, 2003.

M. D. Plumbley. Lie group methods for optimization with orthogonality constraints. In *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, pages 1245–1252, Granada, Spain, 2004.

A. Srivastava, U. Grenander, G. R. Jensen and M. I. Miller. Jump-diffusion Markov processes on orthogonal groups for object recognition. *Journal of Statistical Planning and Inference*, 103(1/2):15–37, 2002.

H. H. Yang and S.-i. Amari. Adaptive online learning algorithms for blind separation: Maximum entropy and minimum mutual information. *Neural Computation*, 9:1457–1482, 1997.

G. R. Warnes. The normal kernel coupler: An adaptive MCMC method for efficiently sampling from multi-modal distributions. Technical Report 39, Dept. of Statistics, University of Washington, 2001.

D. R. Wilson and T. R. Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10):1429–1451, 2003.

L.-Q. Zhang, A. Cichocki and S.-i. Amari. Geometrical structures of FIR manifold and multichannel blind deconvolution. *Journal of VLSI for Signal Processing Systems*, 31:31–44, 2002.