ESTIMATING INDEPENDENT COMPONENTS BY MAPPING ONTO AN ORTHOGONAL MANIFOLD

SIMONE FIORI

Dipartimento di Elettronica, Intelligenza Artificiale e Telecomunicazioni (DEIT),
Università Politecnica delle Marche,
Via Brecce Bianche, I-60131 Ancona, Italy
fiori@deit.univpm.it

(Received 7 January 2008)

Abstract: Algorithms for independent component analysis (ICA) based on information-theoretic criteria optimization over differential manifolds have been devised over the last few years. The principles informing their design lead to various classes of learning rules, including the fixed-point and the geodesic-based ones. Such learning algorithms mainly differ by the way in which single learning steps are effected in the neural system's parameter space, *i.e.* by the action that a connection variable is moved by in the parameter space toward the optimal connection pattern. In the present paper, we introduce a new class of learning algorithms by recalling from the literature on differential geometry the concept of mapping onto manifolds, which provides a general way of acting upon a neural system's connection variable in order to optimize the learning criteria. The numerical behavior of the introduced learning algorithms is illustrated and compared with experiments carried out on mixtures of statistically-independent signals.

Keywords: independent component analysis, ICA, orthogonal group of matrices, mappings onto manifolds

1. Introduction

Any instance of independent component learning by neural networks involves the optimization of a non-linear function of the network's connection pattern over a suitable parameter space. In particular, pre-whitening the observations turns the optimization problem for linear independent component analysis (ICA) into a constrained optimization problem over a set of orthogonal matrices. In this case, the network parameter space is thus the set of multi-dimensional rotations/reflections.

Learning rules that insist on flat spaces relies on standard numerical techniques to be implemented, while learning algorithms that insist on curved parameter spaces necessarily involves theoretical concepts due to the differential geometry to be designed and effectively implemented. Several examples of geometry-based algorithm design in neural networks and in signal and image processing literature may be found in [1–4]. For instance, there is a macroscopic difference between the two classes of algorithms in that a learning step may be effected in a vector space via additive updating, giving rise to piece-wise straight paths over the parameter space, an option unavailable in a curved space.

Learning by optimizing a criterion function in a curved space amounts to effecting short steps in the parameter space until a reasonably optimal connection pattern is encountered. From a conceptual point of view, learning steps can be performed in a curved space in different ways, including methods that involve fixed-point and geodesic-type algorithms.

Fixed-point learning algorithms consist of double-step learning rules, which first effect a learning step toward a looking direction (*e.g.*, along the direction provided by the standard gradient of the learning criterion), followed by a back-projection to the parameter space [5–7]. Geodesic-based learning algorithms consist of single-step learning rules, which move the network connection pattern toward a looking direction directly in the parameter space along a suitably-defined geodesic arc on the parameter manifold [5, 8–12]. These algorithms may take advantage of special closed forms or numerical tricks to approximate the true geodesic paths [5, 13, 14].

Unlike other algorithms stemming from Newton-type optimization of the learning criteria [6, 15] or based on quadratic modeling of true criteria [16], which involve computing second-order derivatives (Hessians) of the criterion function, the learning algorithms introduced here rely on first-order derivatives only. At the same time, the related independent component analysis learning algorithms normally appear in the literature as single-unit learning rules [6, 7, 17], which allow extracting one component at a time; in the present paper, we deal with fully parallel (multi-unit) learning algorithms for independent component analysis.

The present contribution aims at discussing a new class of independent component learning algorithms based on criterion optimization over the orthogonal group of matrices. The learning algorithms are designed to allow the neural system to look for a separating connection pattern by moving within the curved parameter space by intrinsic actions, which are formally described by mappings on the tangent bundle of the parameter space. Three different mappings are recalled from the differential-geometry and geometrical-integration literature. The numerical behavior of these learning algorithms is illustrated and compared with numerical experiments carried out on synthetic and real-world mixtures of independent signals.

The present paper is organized as follows. Section 2 presents the relevant differential-geometrical concepts, instrumental in the development of mapping-based learning algorithms; the complementary concept of automatic learning stepsize selection is covered as well. Section 3 briefly reviews the concept of independent component analysis and customizes the general-purpose mapping-based learning algorithms devised in Section 2 to the separation of independent sources from linear mixtures by proper contrast-function optimization; equivariance of the considered independent-component learning algorithms is also discussed. Section 4 illustrates and discusses numerical results of analysis of speech/musical signals and images, aiming at elucidating the numerical behavior of the mapping-based learning algorithms devised in Section 3 in comparison to the FastICA algorithm [6]. Section 5 concludes the paper.

2. Differential geometrical concepts and algorithm design

The derivation of ICA algorithms in the present paper is based on notions from differential geometry and geometrical integration. The relevant notions are briefly surveyed below, their more detailed discussion to be found in textbooks or reports [1, 2, 9, 16].

The fundamental concept of interest here is that of a differential manifold. The notion of a differential manifold provides a generalization of curves and surfaces in high-dimensional spaces. A smooth curved manifold M of dimension m may be regarded as an object that is locally isomorphic to an Euclidean space, namely, any open neighborhood $U \subset M$ may be mapped onto $\mathscr{R}^m$ via so-termed coordinate charts. Computation on a manifold is essentially defined as computation on the coordinate sets in $\mathscr{R}^m$, while the calculus on manifolds is developed so that it is independent of the choice of coordinates.

Local linearizations of a manifold M at points $x \in M$ are provided by tangent spaces T_xM, whose union forms the tangent bundle TM. Any tangent space is isomorphic to $\mathscr{R}^m$. A Riemannian manifold is an (M,g) structure, where $g_x : T_xM \times T_xM \to \mathscr{R}$ is a scalar product that allows turning a Riemannian manifold into a metric space. The Riemannian gradient of a smooth function $f : M \to \mathscr{R}$ at a point $x \in M$ is an element of T_xM whose scalar product with every other element of T_xM does not depend on the particular choice of the Riemannian metric $g_x(\cdot,\cdot)$.

It will also be instrumental here to recall the notion of a Lie group [18]. A Lie group is an algebraic group that possesses the notable property of being a smooth manifold as well. The group and differential-geometrical structures of a Lie group must be compatible. A fundamental discovery in the theory of Lie groups has been that it is possible to describe (and perform computations on) any tangent space of a Lie group by simply describing in full detail the group's tangent space at the identity termed Lie algebra associated with the Lie group. Then, the group's left- or right-translation allows computation on the whole tangent bundle.

If the structure of the tangent bundle of a manifold is easily handled, the tangent spaces may be conveniently used as coordinate spaces for the manifold. It is known in differential geometry that there exist particularly handy charts to map neighborhoods of the manifold to the tangent bundle and vice-versa. In the context of numerical implementation of learning algorithms, a useful instance of such charts are the so-termed mappings. It is required by definition that a restriction, R_x, of mapping $R : TM \to M$ to the tangent space T_xM at the point x of a manifold M satisfies the following conditions [16]:

1. the restriction, R_x, is defined in an open ball, $B(0,r_x)$, of radius r_x about $0 \in T_xM$;
2. the $R_x(v) = x$ equality holds if and only if $v = 0 \in T_xM$;
3. the $DR_x(0) = \mathrm{Id}_{T_xM}$ equality holds, where operator DR_x denotes the tangent map associated with R_x and Id_{T_xM} denotes the identity map in T_xM.

Given a Lie group G, an exponential map $\exp_x(v)$ is a standard chart from a neighborhood of $v \in T_xG$ to G: Exponential maps are instances of mapping and any other mapping may be regarded as a first-order approximation of an exponential map [16].

The numerical tool provided by mappings comes into effect in neural learning whenever an adapting procedure is formulated, *e.g.* in terms of gradient-based optimization of a network's performance criterion under constraints. The manifold structure accounts for the constraints and the gradient to be calculated is actually a Riemannian gradient.

2.1. Learning by mappings onto an orthogonal manifold

In the present contribution, we are interested in the orthogonal group of matrices, namely $O(p) \stackrel{\text{def}}{=} \{\mathbf{G} \in \mathscr{R}^{p\times p} | \mathbf{G}^T\mathbf{G} = \mathbf{I}_p\}$ (*i.e.* the group of p-dimensional rotations/reflections), which is a classical Lie group. The Lie algebra associated with the orthogonal group is denoted as $\mathbf{so}(p) \stackrel{\text{def}}{=} \{\mathbf{S} \in \mathscr{R}^{p\times p} | \mathbf{S}+\mathbf{S}^T = \mathbf{0}_p\}$, *i.e.* the set of p-dimensional skew-symmetric matrices. It holds that $T_{\mathbf{G}}O(p) = \mathbf{G}\cdot\mathbf{so}(p)$ for every $\mathbf{G} \in O(p)$. The identity of the orthogonal group is the p-dimensional identity matrix $\mathbf{I}_p$. In the present contribution, it is supposed that $p > 3$, a hypothesis leading to a quite general formulation of the mapping-based learning theory, as some properties enjoyed by group $O(2)$ (*e.g.* commutativity) and group $O(3)$ (oftentimes invoked, *e.g.* in robotics [14]) that would simplify treatment, have been ignored here.

The canonical scalar product in $TO(p)$ is $g_{\mathbf{G}}(\mathbf{GS}_1,\mathbf{GS}_2) \stackrel{\text{def}}{=} \text{tr}[\mathbf{S}_1^T\mathbf{S}_2]$, where $\text{tr}[\mathbf{Z}]$ denotes the trace of (square) matrix $\mathbf{Z}$. When $O(p)$ is endowed with the canonical metric, the Riemannian gradient of function $f: O(p) \to \mathscr{R}$ assumes the following form (see *e.g.* [9]):

$$\nabla_{\mathbf{G}} f = \frac{\partial}{\partial \mathbf{G}} f - \mathbf{G}\left(\frac{\partial}{\partial \mathbf{G}} f\right)^T \mathbf{G}, \tag{1}$$

while the exponential map from a point $\mathbf{GS} \in T_{\mathbf{G}}O(p)$ to $O(p)$ has the form of

$$\exp_{\mathbf{G}}(\mathbf{GS}) = \mathbf{G}\exp(\mathbf{S}). \tag{2}$$

In the above expressions, the $\frac{\partial}{\partial \mathbf{X}} f$ symbol denotes the Jacobian of function f with respect to matrix $\mathbf{G}$ entries, arranged again in the matrix form, and $\exp(\cdot)$ denotes matrix exponentiation. Thus, the exponential map (2) is a valid instance of mapping on $O(p)$.

Another kind of mapping invoked below is provided by the Cayley transform [19]:

$$\text{cay}_{\mathbf{G}}(\mathbf{GS}) \stackrel{\text{def}}{=} \mathbf{G}\left(\mathbf{I}_p + \frac{\mathbf{S}}{2}\right)\left(\mathbf{I}_p - \frac{\mathbf{S}}{2}\right)^{-1}. \tag{3}$$

As yet another kind of mapping, let us denote the set of full-rank $p\times p$ matrices as $Gl(p)$, the set of symmetric positive-definite $p\times p$ matrices – as $S^+(p)$, the map from $Gl(p)$ to $O(p)\times S^+(p)$ that associates a full-rank matrix with its polar decomposition – as 'pol' and the standard projection onto the first factor – as π_1. Then, a mapping for the orthogonal group is as follows [2]:

$$\text{pol}_{\mathbf{G}}(\mathbf{GS}) \stackrel{\text{def}}{=} (\pi_1 \circ \text{pol})(\mathbf{G}+\mathbf{GS}). \tag{4}$$

It is worth noting that the first factor of a polar decomposition may be given a closed-form expression. In fact, it can be easily proven that for any matrix $\mathbf{Z} \in Gl(p)$ the following will hold:

$$(\pi_1 \circ \text{pol})(\mathbf{Z}) = \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-\frac{1}{2}}. \tag{5}$$

Let us now consider an optimization problem that consists in finding the local extremes of a function $f:O(p)\to\mathscr{R}$. The solution $\mathbf{G}=\mathbf{G}(t)$ of the differential equation:

$$\frac{d}{dt}\mathbf{G}(t)=\nabla_{\mathbf{G}(t)}f, \quad \text{with } \mathbf{G}(0)=\mathbf{G}_0\in O(p), \tag{6}$$

will asymptotically tend to one of the local extremes of the $f(\cdot)$ function over $O(p)$, depending on the boundary value $\mathbf{G}_0$.

In practical terms, the optimization-type differential equation (6) may not be solved in closed form and therefore needs to be turned into a discrete-time algorithm via any suitable time-discretization scheme. Such operation may actually be effected with the help of mappings. Let us denote a generic discrete-time (learning step) index with n and the current step approximation of the solution to the differential equation (6) with $\mathbf{G}_n$. Now, a learning step may be effected as:

$$\mathbf{G}_{n+1}=R_{\mathbf{G}_n}(\mu_n\nabla_{\mathbf{G}_n}f), \quad n\geq 0, \tag{7}$$

where $R(\cdot)$ denotes any suitable mapping on the orthogonal group $O(p)$, while the schedule $\mu_n\in\mathscr{R}$ denotes an adjustment of the learning step size. The right-hand side of Equation (7) represents a learning step in the direction of the Riemannian gradient of the criterion $f(\cdot)$ to be optimized, starting from the previously learnt connection point.

As noted above, any tangent direction in $T_{\mathbf{G}_n}O(p)$ is expressible as product $\mathbf{G}_n\mathbf{S}_n$, where the skew-symmetric $p\times p$ matrix $\mathbf{S}_n$ is required to compute explicitly, *e.g.* when using the exponential map and the Cayley-type mappings. It is thus necessary to compute the $\mathbf{S}_n=\mathbf{G}_n^T\nabla_{\mathbf{G}_n}f$ quantity beforehand. It is apparent from expression (1) that:

$$\mathbf{S}_n=\mathbf{G}_n^T\left(\frac{\partial}{\partial\mathbf{G}_n}f\right)-\left(\frac{\partial}{\partial\mathbf{G}_n}f\right)^T\mathbf{G}_n. \tag{8}$$

In the section devoted to numerical experiments, the three above-mentioned mappings will be applied to learn orthogonal neural connection patterns. Their differences in terms of separation performance and computational complexity will be evaluated.

2.2. Selection of a learning stepsize schedule

On the basis of the general structure of the mapping-based learning algorithm (7) and the geometry of orthogonal group $O(p)$, it is possible to find a schedule for the learning stepsize, μ_n, to be employed during learning. Below, we consider a linear artificial neural network of connection pattern $\mathbf{G}\in O(p)$ described by the input-output relationship $\mathbf{y}_n=\mathbf{G}_n^T\mathbf{x}$, where $\mathbf{x}\in\mathscr{R}^p$ denotes the input stream and $\mathbf{y}_n\in\mathscr{R}^p$ denotes the output stream at learning iteration n. In the present paper, a vector always denotes a *column-type array*. According to the notation introduced in Section 2.1, the network learning criterion is denoted here as $f:O(p)\to\mathscr{R}$ and is supposed to be of the following form:

$$f(\mathbf{G})\overset{\text{def}}{=}\sum_{i=1}^{p}\mathbb{E}[F(\mathbf{g}_i^T\mathbf{x})], \tag{9}$$

where $\mathbf{g}_i$ denotes the i^{th} column of connection matrix $\mathbf{G}$, while $\mathbb{E}[F(z)]$ denotes the statistical expectation of function $F:\mathscr{R}\to\mathscr{R}$ over the distribution of random variable $z \in \mathscr{R}$. The scalar-to-scalar function $F(\cdot)$ is required here to be convex and belong to the $C^2(\mathscr{R})$ class. In a slight notation abuse, we shall also evaluate the $F(\mathbf{y})$ quantity, where the scalar-to-scalar function $F(\cdot)$ is supposed to act component-wise on the vector argument and returns a vector of the same size (*viz.* of size p in the present context).

We begin our analysis by observing that, because of the Lie-group structure of the base manifold, $O(p)$, learning scheme (7) may be rewritten via a left-translation of the Lie algebra $\mathbf{so}(p)$ as follows:

$$\mathbf{G}_{n+1} = \mathbf{G}_n R_{\mathbf{I}_p}(\mu_n \mathbf{S}_n). \tag{10}$$

Consequently, the neural network output stream at learning iteration $n+1$ may be written as:

$$\mathbf{y}_{n+1} = R^T_{\mathbf{I}_p}(\mu_n \mathbf{S}_n)\mathbf{y}_n. \tag{11}$$

At the same time, the $f_n \overset{\text{def}}{=} f(\mathbf{G}_n)$ learning criterion at iterations n and $n+1$ may be concisely rewritten from Equation (9) as:

$$f_n = \mathbf{1}^T \mathbb{E}[F(\mathbf{y}_n)], \quad f_{n+1} = \mathbf{1}^T \mathbb{E}[F(R^T_{\mathbf{I}_p}(\mu_n \mathbf{S}_n)\mathbf{y}_n)], \tag{12}$$

with $\mathbf{1} \overset{\text{def}}{=} [1\ 1\ 1\ \cdots\ 1] \in \mathscr{R}^p$.

For the convex function, $F(\cdot)$, the following inequality holds [7]:

$$F(\mathbf{z}_2) - F(\mathbf{z}_1) \geq \left(\frac{\partial}{\partial \mathbf{z}} F \bigg|_{\mathbf{z}_1} \right)^T (\mathbf{z}_2 - \mathbf{z}_1), \tag{13}$$

for all $\mathbf{z}_1$ and $\mathbf{z}_2$ in $\mathscr{R}^p$.

We may also note that mappings (2), (3) and (4) are described by analytic functions in suitable subsets of $\mathbf{so}(p)$, so that they may be expanded in power series about $\mathbf{0}_p \in \mathbf{so}(p)$ as:

$$R_{\mathbf{I}_p}(\mathbf{S}) = \sum_{k=0}^{\infty} \rho_k \mathbf{S}^k, \quad \rho_k \overset{\text{def}}{=} \frac{1}{k!} \frac{d}{dz} R_{\mathbf{I}_p}(z) \bigg|_{z=0 \in \mathscr{R}}. \tag{14}$$

In particular, we have:

$$\exp_{\mathbf{I}_p}(\mathbf{S}) = \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{S}^k, \quad \mathbf{S} \in \mathbf{so}(p), \tag{15}$$

$$\begin{aligned} \mathrm{cay}_{\mathbf{I}_p}(\mathbf{S}) &= \left(\mathbf{I}_p + \frac{\mathbf{S}}{2} \right) \sum_{k=0}^{\infty} \left(\frac{\mathbf{S}}{2} \right)^k \\ &= \mathbf{I}_p + 2 \sum_{k=1}^{\infty} \left(\frac{\mathbf{S}}{2} \right)^k, \quad \mathbf{S} \in \mathbf{so}(p) \cap B(\mathbf{0}_p, 2), \end{aligned} \tag{16}$$

$$\begin{aligned} \mathrm{pol}_{\mathbf{I}_p}(\mathbf{S}) &= (\mathbf{I}_p + \mathbf{S})[(\mathbf{I}_p + \mathbf{S})^T(\mathbf{I}_p + \mathbf{S})]^{-\frac{1}{2}} \\ &= (\mathbf{I}_p + \mathbf{S})(\mathbf{I}_p - \mathbf{S}^2)^{-\frac{1}{2}} \\ &= (\mathbf{I}_p + \mathbf{S}) \left(\mathbf{I}_p + \frac{1}{2}\mathbf{S}^2 + \cdots \right) \\ &= \mathbf{I}_p + \mathbf{S} + \frac{1}{2}\mathbf{S}^2 + \frac{1}{2}\mathbf{S}^3 + \cdots, \quad \mathbf{S} \in \mathbf{so}(p) \cap B(\mathbf{0}_p, 1). \end{aligned} \tag{17}$$

The analytic expression of the polar-decomposition mapping apparently comes from the closed form expression (5). Interestingly, all the considered mappings share the same low-order terms, as already mentioned in the introduction to the present section. It will be particularly useful to note that identities $\rho_0 = \rho_1 = 1$ and $\rho_2 = \frac{1}{2}$ hold for the three considered mappings. Therefore, a common representation for mappings (2), (3) and (4) is:

$$R_{\mathbf{I}_p}(\mathbf{S}) = \mathbf{I}_p + \mathbf{S} + \frac{1}{2}\mathbf{S}^2\Phi(\mathbf{S}), \tag{18}$$

where map $\Phi : \mathbf{so}(p) \to \mathscr{R}^{p\times p}$ denotes a residual factor.

The aim of the present analysis is to find a learning stepsize schedule, μ_n, that ensures almost monotonic dynamics of the criterion function, f_n, during learning. To this aim, let us expand mapping $R_{\mathbf{I}_p}(\mu_n\mathbf{S}_n)$ with respect to its argument. This amounts to:

$$R_{\mathbf{I}_p}(\mu_n\mathbf{S}_n) = \mathbf{I}_p + \mu_n\mathbf{S}_n + \mathbf{A}(\mu_n\mathbf{S}_n), \tag{19}$$

where the term $\mathbf{A}$ represents a residual, whose evaluation is instrumental in defining the appropriate stepsize schedule. From Equation (18), residual $\mathbf{A}$ is to be written as $\mathbf{A}(\mu_n\mathbf{S}_n) \overset{\text{def}}{=} \frac{\mu_n^2\mathbf{S}_n^2}{2}\Phi(\mu_n\mathbf{S}_n)$.

Considering expansion (19), the network output stream at learning step $n+1$ expressed by Equation (11) is as follows:

$$R^T_{\mathbf{I}_p}(\mu_n\mathbf{S}_n)\mathbf{y}_n = \mathbf{y}_n - \mu_n\mathbf{S}_n\mathbf{y}_n + \mathbf{A}^T(\mu_n\mathbf{S}_n)\mathbf{y}_n. \tag{20}$$

Based on the convexity inequality (13), it thus holds that:

$$\begin{aligned}\mathbf{1}^T\mathbb{E}[F(R^T_{\mathbf{I}_p}(\mu_n\mathbf{S}_n)\mathbf{y}_n)] &= \mathbf{1}^T\mathbb{E}[F(\mathbf{y}_n - \mu_n\mathbf{S}_n\mathbf{y}_n + \mathbf{A}^T\mathbf{y}_n)]\\ &\geq \mathbf{1}^T\mathbb{E}[F(\mathbf{y}_n)] + \mathbb{E}[F'(\mathbf{y}_n^T)(-\mu_n\mathbf{S}_n\mathbf{y}_n + \mathbf{A}^T\mathbf{y}_n)].\end{aligned} \tag{21}$$

Consequently, the following holds as well:

$$f_{n+1} - f_n \geq -\mu_n\mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{S}_n\mathbf{y}_n] + \mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{A}^T\mathbf{y}_n]. \tag{22}$$

An upper bound for learning schedule μ_n, that ensures $f_{n+1} - f_n \geq 0$ at any learning step, is thus as follows:

$$\mu_n \leq \frac{\mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{A}^T(\mu_n\mathbf{S}_n)\mathbf{y}_n]}{\mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{S}_n\mathbf{y}_n]}. \tag{23}$$

It is worth noting that $\mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{A}^T\mathbf{y}_n] = \operatorname{tr}\left[\mathbf{A}^T\mathbb{E}[\mathbf{y}_nF'(\mathbf{y}_n^T)]\right]$. Therefore, the following inequality holds: $\mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{A}^T\mathbf{y}_n] \leq \|\mathbf{A}(\mu_n\mathbf{S}_n)\|_{\text{F}} \cdot \|\mathbb{E}[\mathbf{y}_nF'(\mathbf{y}_n^T)]\|_{\text{F}}$. By definition of the $\mathbf{A}(\mu_n\mathbf{S}_n)$ term, it apparently holds that

$$\|\mathbf{A}(\mu_n\mathbf{S}_n)\|_{\text{F}} \approx \frac{\mu_n^2\|\mathbf{S}_n^2\|_{\text{F}}}{2}\text{bnd}\|\Phi(\mu_n\mathbf{S}_n)\|_{\text{F}}, \tag{24}$$

where the $\text{bnd}\|\Phi(\mu_n\mathbf{S}_n)\|_{\text{F}}$ term denotes a reasonable approximation or bound of the $\|\Phi(\mu_n\mathbf{S}_n)\|_{\text{F}}$ quantity no longer dependent on μ_n. In practice, we may select the learning stepsize by meeting the equality signs in relationships (23) and (24). This ultimately yields the expression of choice for the learning stepsize schedule:

$$\mu_n = \frac{2\mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{S}_n\mathbf{y}_n]}{\|\mathbf{S}_n^2\|_{\text{F}}\|\mathbb{E}[\mathbf{y}_nF'(\mathbf{y}_n^T)]\|_{\text{F}}\text{bnd}\|\Phi(\mu_n\mathbf{S}_n)\|_{\text{F}}}. \tag{25}$$

It is now necessary to customize formula (25) to the three mappings (exponential (2), Cayley-type (3) and polar-type (4)). We reckon that, as long as the norm of $\mu_n\mathbf{S}$ is sufficiently less than 1 during learning, the residual $\Phi(\mu_n\mathbf{S}_n)$ may be safely approximated as the identity matrix $\mathbf{I}_p$, so that we may ultimately assume $\mathrm{bnd}\|\Phi(\mu_n\mathbf{S}_n)\|_{\mathrm{F}} = \sqrt{p}$.

In conclusion, we accept the following learning stepsize schedule for exponential, Cayley-type and polar-decomposition mappings:

$$\mu_n = \frac{2\mathbb{E}[F'(\mathbf{y}_n^T)\mathbf{S}_n\mathbf{y}_n]}{\sqrt{p}\|\mathbf{S}_n^2\|_{\mathrm{F}}\|\mathbb{E}[\mathbf{y}_nF'(\mathbf{y}_n^T)]\|_{\mathrm{F}}}. \tag{26}$$

We deem it appropriate to extend the use of the above learning stepsize schedule to all the three mapping-based neural learning algorithms.

3. ICA algorithms by mapping onto an orthogonal manifold

In the present section, we briefly recall the concept of ICA and present ICA-type learning algorithms based on mappings on the orthogonal group of matrices.

3.1. Brief overview of ICA

The ICA technique aims at recovering statistically independent source signals from their observable mixtures [6, 17]. In the present paper, we consider linear, instantaneous and noiseless mixtures of the following structure:

$$\mathbf{x} = \mathbf{M}\mathbf{s} + \mathbf{q}, \tag{27}$$

where $\mathbf{s} \in \mathscr{R}^p$ is a stream of statistically independent source signals, $\mathbf{x} \in \mathscr{R}^p$ is the observation stream, $\mathbf{M} \in \mathscr{R}^{p\times p}$ is a full-rank constant mixing matrix, and $\mathbf{q} \in \mathscr{R}^p$ denotes an observation disturbance supposed to be so weak to neglect it.

For the ICA neural network, we invoke the familiar input-output description, $\mathbf{y} = \mathbf{G}^T\mathbf{x}$, where $\mathbf{y} \in \mathscr{R}^m$ denotes the network's response vector stream and $\mathbf{G} \in \mathscr{R}^{p\times p}$ denotes its connection pattern formed by connection weights/strengths.

As every full-rank mixture may be reduced to an orthogonal one by whitening the observations, we may restrict our analysis to orthogonal mixtures without loss of generality. We shall assume $\mathbf{M} \in O(p)$, in this case the separating network may be described by an orthogonal connection pattern, $\mathbf{G}$ in $O(p)$.

A class of ICA algorithms stems from the following optimization principle: Under constraint $\mathbf{G} \in O(p)$, optimize criterion $\Psi(\mathbf{G}) : O(p) \to \mathscr{R}$.

It will be instrumental to recall here two suitable criteria devised under the basic hypothesis that all source streams in a mixture possess same-sign kurtosis. One criterion is as follows:

$$\Psi_+(\mathbf{G}) \overset{\mathrm{def}}{=} \frac{1}{4}\sum_{i=1}^{p}\mathbb{E}[(\mathbf{g}_i^T\mathbf{x})^4], \tag{28}$$

which allows separating source signals of positive kurtosis and the criterion should be maximized under the constraint $\mathbf{G} \in O(p)$. Another criterion is:

$$\Psi_-(\mathbf{G}) \overset{\mathrm{def}}{=} \frac{1}{\lambda}\sum_{i=1}^{p}\mathbb{E}[\log\cosh(\lambda\mathbf{g}_i^T\mathbf{x})], \tag{29}$$

which allows separating source signals of negative kurtosis and the criterion should be minimized under constraint $\mathbf{G} \in O(p)$. In expression (29), the λ constant should be selected in the $1 \le \lambda \le 2$ range.

In the following section, we introduce neural learning algorithms enabling a linear neural network to analyze simultaneously the independent components forming a mixture. The derivation of such algorithms is based on the geometric properties of the orthogonal group of matrices and the concept of mapping as introduced in Section 2.

It should be noted that criteria (28) and (29) are of the (9) type, which was taken as a basic assumption in Section 2 in order to develop the mapping-based learning theory.

3.2. Algorithm description

Much of the theory instrumental in developing mapping-based ICA-type learning algorithms has been presented in Section 2 above. Let us summarize here the fundamental ideas introduced so far:

- As seen in Section 3.1, an instance of ICA may be formulated as an optimization problem based on the criterion function $\Psi(\mathbf{G})$, to be optimized under the orthogonality constraint of the connection pattern $\mathbf{G}$.
- The orthogonality constraint may be conveniently handled by recognizing that the optimization process should be effected over the group/manifold of orthogonal matrices $O(p)$, whose geometry has been discussed in Section 2.
- Optimization may be effected via a Riemannian-gradient-type ascent rule over the group/manifold of orthogonal matrices (6). Thus, we have recalled the structure of a Riemannian gradient on manifold $O(p)$.
- The Riemannian-gradient-type ascent rule, which appears as a differential equation over the $O(p)$ base-manifold, may be turned into a learning algorithm via the mapping-based numerical approximation scheme (7).
- The three available mappings (exponential (2), Cayley-type (3) and polar-type (4)) may be effectively used to perform learning.
- The learning stepsize schedule expressed by Equation (26) may be used to let the learning algorithm self-control the learning speed according to the progress of learning.

In the following, we give explicit expressions for the quantities needed to implement a generic learning step n of the learning algorithms. All the variables of interest that change during learning will be n-footed for the sake of clarity.

The Jacobian of the criterion function $\Psi(\mathbf{G}_n)$ is as follows:

$$\frac{\partial}{\partial \mathbf{G}_n}\Psi = \mathbb{E}[\mathbf{x}F(\mathbf{y}_n^T)]. \tag{30}$$

where $\mathbf{y}_n \overset{\text{def}}{=} \mathbf{G}_n^T\mathbf{x}$ and the scalar function $F(\cdot)$ is allowed to operate component-wise. According to Equations (1) and (8), the further two quantities of interest are:

$$\nabla_{\mathbf{G}_n}\Psi = \mathbb{E}[\mathbf{x}F(\mathbf{y}_n^T)] - \mathbf{G}_n\mathbb{E}[F(\mathbf{y}_n)\mathbf{y}_n^T], \tag{31}$$

$$\mathbf{S}_n = \mathbb{E}[\mathbf{y}_nF(\mathbf{y}_n^T) - F(\mathbf{y}_n)\mathbf{y}_n^T]. \tag{32}$$

According to the three mappings described in Section 2, we obtain the following learning algorithms:

$$\text{EXPRET} \quad \mathbf{G}_{n+1} = \mathbf{G}_n\exp(\mu_n\mathbf{S}_n), \tag{33}$$

$$\text{CAYRET}\quad \mathbf{G}_{n+1}=\mathbf{G}_n\left(\mathbf{I}_p+\frac{\mu_n\mathbf{S}_n}{2}\right)\left(\mathbf{I}_p-\frac{\mu_n\mathbf{S}_n}{2}\right)^{-1}, \tag{34}$$

$$\text{POLRET}\quad \mathbf{G}_{n+1}=(\pi_1\circ\mathrm{pol})(\mathbf{G}_n+\mu_n\nabla_{\mathbf{G}_n}\Psi). \tag{35}$$

The simplest choice for the initial state is $\mathbf{G}_0=\mathbf{I}_p$, while the learning stepsize schedule, μ_n, may be selected as in expression (26). In order to develop a computer-based implementation of the above learning equations, all the ensemble averages denoted by the statistical expectation operator should be replaced with sample means.

3.3. Algorithms' equivariance

The concept of equivariance of an estimation algorithm in ICA was introduced in [20]. In short, an equivariant ICA algorithm and, hence, its separation performance are not explicitly dependent either on the mixing matrix $\mathbf{M}$ or the separation pattern $\mathbf{G}_n$; they rather depend on the separation product $\mathbf{P}_n\overset{\text{def}}{=}\mathbf{G}_n^T\mathbf{M}$ as a whole. The equivariance of an algorithm is warranted if it is possible to write the $\mathbf{P}_{n+1}$ matrix in terms of $\mathbf{P}_n$ only using the equation(s) that define the learning algorithm's steps.

The three learning algorithms presented in Section 3.2 are equivariant. In order to prove this for the EXPRET (33) and CAYRET (34) algorithms, it is sufficient to note that:

- the network output signal at any iteration step may be written as $\mathbf{y}_n=\mathbf{G}_n^T\mathbf{x}=\mathbf{G}_n^T\mathbf{M}\mathbf{s}=\mathbf{P}_n\mathbf{s}$;
- the matrix $\mathbf{S}_n$, computed from Equation (32), depends on the vector stream $\mathbf{y}_n$ only; thus, by virtue of the above observation, it depends on the product matrix $\mathbf{P}_n$ only;
- having transposed and post-multiplied both sides of the EXPRET (33) and CAYRET (34) learning equations by matrix $\mathbf{M}$, it is apparent that $\mathbf{P}_{n+1}$ may be written in terms of $\mathbf{P}_n$ only. This proves the equivariance of the EXPRET (33) and CAYRET (34) algorithms.

In order to prove that equivariance holds for the POLRET algorithm (35), it is instrumental to additionally note that:

- the Riemannian gradient (31) may be written as $\nabla_{\mathbf{G}_n}\Psi=\mathbf{G}_n\mathbf{S}_n$;
- the polar $\mathrm{pol}(\cdot)$ decomposition in Equation (35) is unaffected by a pre-multiplication of the argument by an $O(p)$-matrix. In particular, the first projection is unaffected;
- the terms on the right-hand side of the POLRET learning equations (35) may thus be written as $\mathbf{G}_n(\pi_1\circ\mathrm{pol})(\mathbf{I}_p+\mu_n\mathbf{S}_n)$;
- having transposed and post-multiplied both sides of the POLRET learning equation (35) by matrix $\mathbf{M}$, it is apparent that $\mathbf{P}_{n+1}$ may be written in terms of $\mathbf{P}_n$ only. This proves the equivariance of the POLRET algorithm (35).

Equivariance is an important property in blind signal processing: it implies that the component extraction ability of an equivariant algorithm is independent of the conditioning of the mixing matrix.

4. Numerical results

Results of numerical experiments are discussed below in order to evaluate and compare the numerical behavior of the algorithms described in Section 3.2. The presented experiments were carried out over synthetic mixtures of speech/sounds and gray-scale images. In these experiments, the mixing matrix $\mathbf{M}$ is randomly generated in the signal model (27), each entry distribution being uniform. Interchannel interference (ICI) was selected as the component extraction performance index to measure the separation ability of the considered algorithms, defined as:

$$\mathrm{ICI} \stackrel{\mathrm{def}}{=} \frac{1}{p} \frac{\sum_{ij} P_{ij}^2 - \sum_i \max_k \{P_{ik}^2\}}{\sum_i \max_k \{P_{ik}^2\}}, \tag{36}$$

where the separation product $\mathbf{P}$ is again defined by $\mathbf{y} = \mathbf{P}\mathbf{s}$.

Pre-whitening of the observed signals was effected through standard eigenvalue decomposition of the covariance matrix by computing the covariance matrix $\boldsymbol{\Sigma}_x \stackrel{\mathrm{def}}{=} \mathbb{E}[\mathbf{x}\mathbf{x}^T]$, its eigen-pair $(\mathbf{E}, \boldsymbol{\Lambda})$ such that $\boldsymbol{\Sigma}_x = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T$ and then the projection $\sqrt{\boldsymbol{\Lambda}}\mathbf{E}^T\mathbf{x}$.

As general-purpose index we used to evaluate the numerical behavior of the independent component analysis algorithms, an orthonormality measure was defined in order to check the adherence of the network connection matrix to the orthogonal group:

$$\delta(\mathbf{G}) \stackrel{\mathrm{def}}{=} \|\mathbf{G}^T\mathbf{G} - \mathbf{I}_p\|_{\mathrm{F}}, \tag{37}$$

where $\|\cdot\|_{\mathrm{F}}$ again denotes the Frobenius norm of the matrix argument. We measured the total flops per learning iteration and the total time required by the algorithms to run on a 1.86 GHz – 512 MB platform as indices of computational complexity.

For further comparison, the numerical behavior of the EXPRET (33), CAYRET (34) and POLRET (35) algorithms was compared with the numerical behavior of the FastICA algorithm [6].

4.1. Experiment with sampled speech/musical signals

In our first experiment, we separated five speech/musical signals. A criterion function (29) was selected to analyze the components forming the five synthetic mixtures, with constant λ set at 1.

The values of the ICI residual and the criterion function (29) during learning is shown in Figure 1 for the EXPRET (33), CAYRET (34), POLRET (35) and FastICA algorithms (with a 'tanh' non-linearity). The proposed algorithms behaved satisfactorily in this experiment: in particular, they nearly behaved almost identically, with only slight differences noticeable in the curves. The numerical features of the EXPRET (33), CAYRET (34) and POLRET (35) algorithms are also comparable to the performance of the FastICA algorithm [6], both in terms of separation ability and convergence speed.

All the considered algorithms exhibit excellent numerical precision, $-310 \leq \log_{10}\delta \leq -280$. The three mapping-based algorithms exhibit very good numerical precision performance in this analysis, the best of them being the POLRET algorithm (35). The reason of this behavior is directly recognizable in the structure of Equations (33), (34) and (35): in the EXPRET and CAYRET equations, the connection/weight matrix $\mathbf{G}_n$ is subjected to serial updating, which inevitably causes accumulation of numerical errors; in contrast, the structure of the POLRET equation suggests that the connection/weight matrix is renewed at each iteration and it adheres to the orthogonal group of matrices up to machine precision at each iteration, hence avoiding the accumulation of numerical errors.

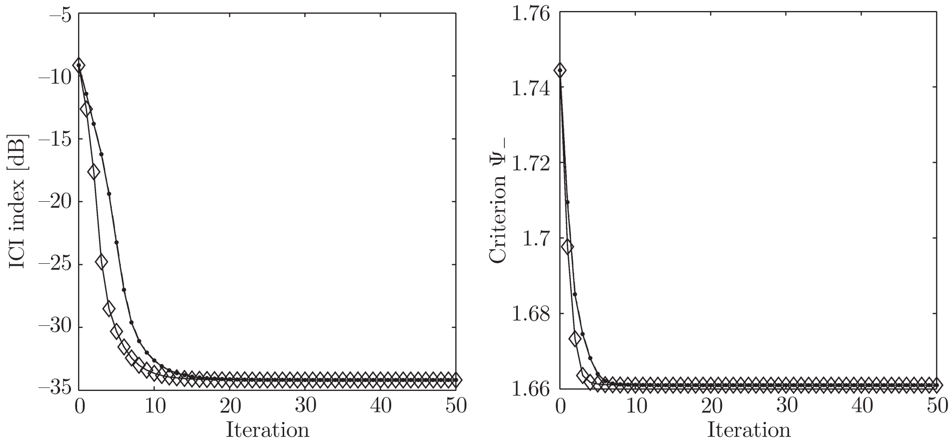

Figure 1. Values of ICI indices (36) and the criterion function (29) during learning for EXPRET (solid line), CAYRET (dotted line), POLRET (dot-dashed line) and FastICA (diamond-dashed line); results of the experiment with sampled speech/musical signals

The computational burden of the four considered algorithms, in terms of flops per iteration and run-time count, is illustrated in Figure 2, which also includes a comparison of the four algorithms' ultimate performance. A conclusion to be drawn from the obtained results is that the differences among the considered algorithms are far from apparent. However, the CAYRET algorithm offers a slightly better trade-off of separation performance, numerical precision and computational burden than the other considered algorithms.

4.2. Experiment with digital images

As our second numerical experiment, we separated the eight 256-gray-level, 128×128, real-world images shown in Figure 5. The source images were selected so that all of them exhibit negative kurtosis. In order to analyze the components forming the eight synthetic mixtures, the criterion function (29) and the learning stepsize schedule (26) were put into effect, as we have confirmed in the previous experiments that the same learning stepsize schedule may be adopted for the EXPRET (33), CAYRET (34) and POLRET (35) algorithms.

The values of the ICI index and the criterion function (29) during learning for the EXPRET (33), CAYRET (34), POLRET (35) and FastICA algorithms (with a 'cube' non-linearity) are shown in Figure 3. In this experiment all the algorithms behaved similarly. The behavior of the considered mapping-based algorithms was compared with that of the FastICA algorithm: in this case, convergence was achieved much faster by the FastICA algorithm, while the three mapping-based algorithms achieved comparable separation results after learning completion. This observation suggests that while the selected learning stepsize schedule may be too low and its structure could be improved with further investigation, it does ensure monotonic convergence and is therefore useful.

The values of the stepsize schedule (26) and the $\|\mu \mathbf{S}\|_{\mathrm{F}}$ quantity during learning for the EXPRET (33), CAYRET (34) and POLRET (35) algorithms is shown in Figure 4. We recall that the FastICA algorithm has no tunable parameters to be adjusted. As

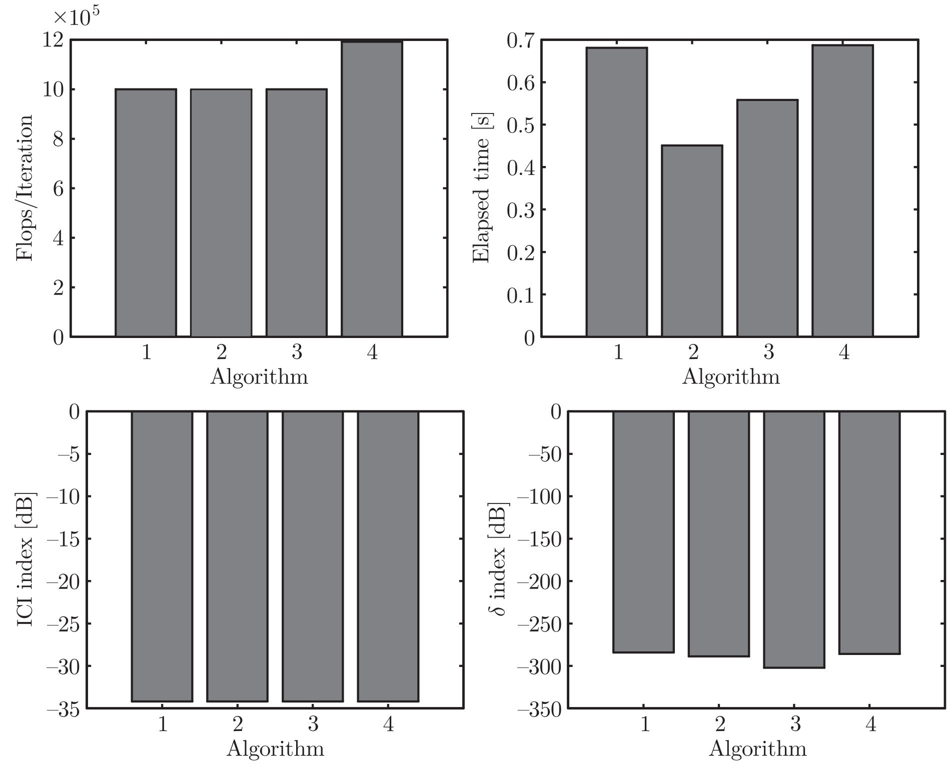

Figure 2. Flops per iteration and time-count and the ICI (36) and δ (37) indices values after learning (algorithms: 1 – EXPRET, 2 – CAYRET, 3 – POLRET and 4 – FastICA); results of the experiment with sampled speech/musical signals

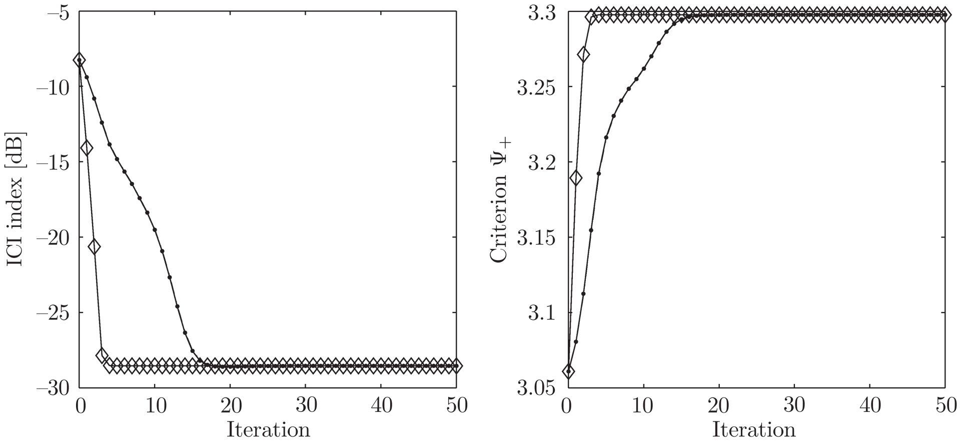

Figure 3. Values of the ICI index (36) and the criterion function (28) during learning for EXPRET (solid line), CAYRET (dotted line), POLRET (dot-dashed) and FastICA (diamond-dashed); results of the experiment with digital images

mentioned in Section 2.2, the $\|\mu_n\mathbf{S}_n\|_\mathrm{F}$ norm is small enough for the approximation of the bound selected for the scalar quantity $\|\Phi(\mu_n\mathbf{S}_n)\|_\mathrm{F}$ be be acceptable. This quantity tends to zero when learning is about to be accomplished.

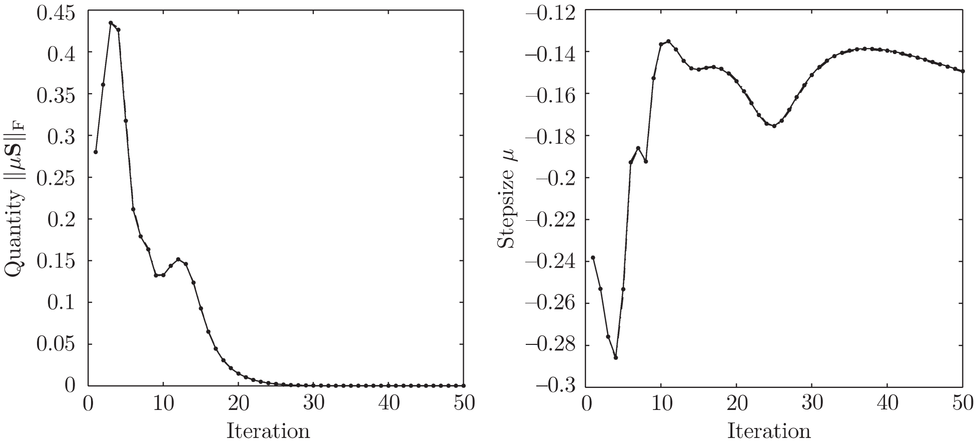

Figure 4. Values of the learning stepsize schedule (26) and the $\|\mu\mathbf{S}\|_{\mathrm{F}}$ product during learning for EXPRET (solid line), CAYRET (dotted line) and POLRET (dot-dashed); results of the experiment with digital images

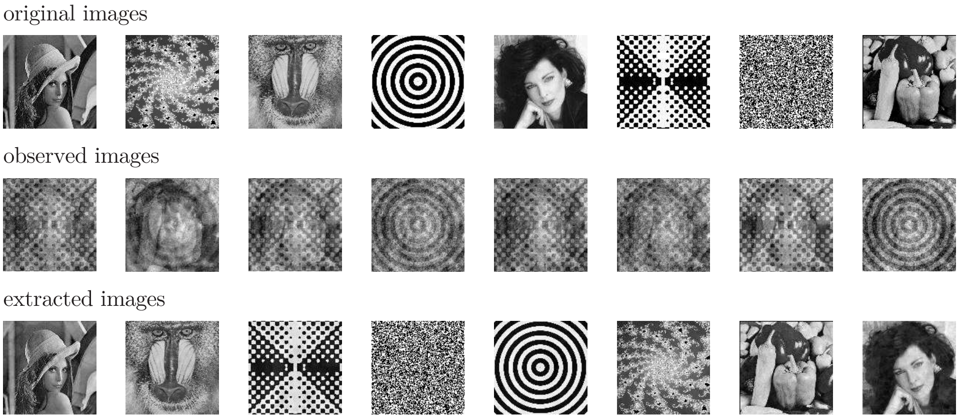

Figure 5. Source images, mixtures and estimated components; results of the experiment with digital images

The source signals, mixtures and estimated components as obtained by running the POLRET algorithm are shown in Figure 5. The extracted images are well discernible.

5. Conclusion

The aim of the present research has been to illustrate the unifying concept of learning by mappings in the special case of function optimization on a manifold of orthogonal matrices for ICA applications. In summary, the theoretical work underlying the present paper consisted in:

- Formulating a principle for learning criteria optimization over Riemannian manifolds based on mappings that would generalize the notion of geodesic-based learning. Such general framework is interesting from a theoretical point of view, as it helps dealing with the fundamental question whether a coordinate chart leading to good learning ability corresponds to any geodesic curve under some metric to be determined.

- Finding an appropriate learning stepsize schedule for the algorithms in question that would guarantee almost monotonic convergence.
- Obtaining equivariance of the ICA-type learning algorithms in question.

The developed algorithms were tested on signals mixtures; the obtained numerical results can be summarized as follows:

- The learning stepsize schedule devised in Section 2.2 is appropriate for the three mapping-based learning algorithms and guarantees monotonic convergence.
- The behavior of the considered mapping-based learning algorithms in terms of separation ability and convergence speed is very similar.
- The behavior of the considered mapping-based learning algorithms in terms of computational complexity is also very similar, though this issue depends on how the mappings are effectively computed in practice. For example, the Matlab developing environment offers several possible algorithms to compute the matrix exponential. The orthogonal factor of the polar decomposition was computed by invoking the singular-value decomposition of the argument.

While computationally more expensive than first-order methods, second-order methods of computing flows generated by differential equations on manifolds (*e.g.* the Newton method) exhibit better optimization performance. Second-order methods rely on maps from tangent spaces to the base manifold as well as first-order methods. Therefore, application of general mapping-based tools to second-order methods for learning on curved manifolds is an avenue worth investigating.

Acknowledgements

The author wishes to thank Prof. P.-A. Absil for kindly sharing his thoughts on polar-type mapping.

References

[1] Chefd'hotel C, Tschumperlé D, Deriche R and Faugeras O D 2004 *J. Math. Imaging and Vision (JMIV)* **20** (1-2) 147

[2] Dehaene J 1995 *Continuous-type Matrix Algorithms, Systolic Algorithms and Adaptive Neural Networks*, PhD Thesis of the Department Elektrotechniek-ESAT, Faculteit der Toegepaste Wetenschappen, Katholieke Universiteit Leuven

[3] Smith S T 2005 *IEEE Trans. on Signal Processing* **53** (5) 1610

[4] Tanaka T and Fiori S 2006 *Int. Conf. Acoustics, Speech and Signal Processing (IEEE-ICASSP)*, Toulouse, France, **III**, pp. 548–551

[5] Fiori S 2002 *IEEE Trans. on Neural Networks* **13** (3) 521

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

[7] Regalia P A and Kofidis E 2003 *IEEE Trans. on Neural Networks* **14** (4) 943

[8] Fiori S 2001 *Neural Computation* **13** (7) 1625

[9] Fiori S 2005 *J. Machine Learning Research* **6** 743

[10] Liu X, Srivastava A and Gallivan K 2004 *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26** (5) 662

[11] Nishimori Y and Akaho S 2005 *Neurocomputing* special issue on *Geometrical Methods in Neural Networks and Learning*, (Fiori S and Amari S-i, Eds), **67** 106

[12] Plumbley M D 2003 *IEEE Trans. on Neural Networks* **14** (3) 534

[13] Celledoni E and Fiori S 2004 *J. Comput. Appl. Math. (JCAM)* **172** (2) 247

[14] Taylor C J and Kriegman D J 1994 *Technical Report No. 9405*, Yale University

[15] Akuzawa T 2000 *Proc. 2nd Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA'2000)*, Helsinki, Finland, pp. 521–525

[16] Absil P-A, Baker C G and Gallivan K A 2004 *Technical Report FSU-CSIT-04-13*, School of Computational Science at Florida State University

[17] Chicocki A and Amari S-i 2002 *Adaptive Blind Signal and Image Processing*, J. Wiley & Sons

[18] Hall B C 2004 *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, Graduate Texts in Mathematics, Springer-Verlag, New York

[19] Diele F, Lopez L and Politi T 1998 *J. Comput. Appl. Math.* **89** 219

[20] Cardoso J F and Laheld B 1996 *IEEE Trans. on Signal Processing* **44** (12) 3017