Algorithms for Nonnegative Independent Component Analysis

Mark D. Plumbley

IEEE Transactions on Neural Networks, 4(3), 534-543, May 2003

Abstract

We consider the task of solving the independent component analysis (ICA) problem **x** = **As** given observations **x**, with a constraint of nonnegativity of the source random vector **s**. We refer to this as *nonnegative independent component analysis* and we consider methods for solving this task. For independent sources with nonzero probability density function (pdf) $p(s)$ down to $s = 0$ it is sufficient to find the orthonormal rotation **y** = **Wz** of prewhitened sources **z** = **Vx**, which minimizes the mean squared error of the reconstruction of **z** from the rectified version $\mathbf{y}^+$ of **y**. We suggest some algorithms which perform this, both based on a nonlinear principal component analysis (PCA) approach and on a geodesic search method driven by differential geometry considerations. We demonstrate the operation of these algorithms on an image separation problem, which shows in particular the fast convergence of the rotation and geodesic methods and apply the approach to a musical audio analysis task.

Index Terms

Geodesic, independent component analysis (ICA), nonnegativity, Stiefel manifold.

Algorithms for Nonnegative Independent Component Analysis

Mark D. Plumbley, *Member, IEEE*

***Abstract*—We consider the task of solving the independent component analysis (ICA) problem $\mathbf{x} = \mathbf{A}\mathbf{s}$ given observations $\mathbf{x}$, with a constraint of nonnegativity of the source random vector $\mathbf{s}$. We refer to this as *nonnegative independent component analysis* and we consider methods for solving this task. For independent sources with nonzero probability density function (pdf) $p(s)$ down to $s = 0$ it is sufficient to find the orthonormal rotation $\mathbf{y} = \mathbf{W}\mathbf{z}$ of prewhitened sources $\mathbf{z} = \mathbf{V}\mathbf{x}$, which minimizes the mean squared error of the reconstruction of $\mathbf{z}$ from the rectified version $\mathbf{y}^+$ of $\mathbf{y}$. We suggest some algorithms which perform this, both based on a nonlinear principal component analysis (PCA) approach and on a geodesic search method driven by differential geometry considerations. We demonstrate the operation of these algorithms on an image separation problem, which shows in particular the fast convergence of the rotation and geodesic methods and apply the approach to a musical audio analysis task.**

***Index Terms*—Geodesic, independent component analysis (ICA), nonnegativity, Stiefel manifold.**

I. Introduction

In many signal processing and data analysis applications, we want to estimate the sources of observations which, when mixed, give the data that we observe. In the simplest form of this problem, we may write the linear generative model

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

where $\mathbf{s} = (s_1, \ldots, s_n)$ is a source vector of real-valued random variables, $\mathbf{A}$ is a nonsingular $n \times n$ mixing matrix with a_{ij} representing the amount of source j that appears in observation i and $\mathbf{x} = (x_1, \ldots, x_n)$ is the random vector of mixtures from which our observations are generated. If we let $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_k)$ be a matrix where each column of $\mathbf{S}$ is one of k samples from the random vector $\mathbf{s}$, then we can write

$$\mathbf{X} = \mathbf{A}\mathbf{S} \tag{2}$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is the matrix whose columns are the corresponding samples from $\mathbf{x}$.

Our task is then to discover the source samples $\mathbf{S}$ and mixing matrix $\mathbf{A}$ given only the observations $\mathbf{X}$.

Manuscript received May 5, 2002; revised November 25, 2002. This work was supported by the U.K. Engineering and Physical Sciences Research Council under Grant GR/R54620. Part of this work was undertaken while the author was visiting the Neural Networks Research Centre at the Helsinki University of Technology, supported by a Leverhulme Trust Study Abroad Fellowship.

The author is with the Department of Electronic Engineering, Queen Mary University of London, London WC2R 2LS, U.K. (e-mail: mark.plumbley@elec.qmul.ac.uk.).

Digital Object Identifier 10.1109/TNN.2003.810616

Even for the case that $\mathbf{A}$ is square, as we assume here, this problem is underdetermined: given any pair $(\mathbf{A}^*, \mathbf{S}^*)$ which solve (2) for a given $\mathbf{X}$ we can generate a whole family of alternative solutions $(\mathbf{A}^*\mathbf{M}, \mathbf{M}^{-1}\mathbf{S}^*)$ where $\mathbf{M}$ is any invertible $n \times n$ matrix. Therefore, we need some constraints on $\mathbf{A}$ and/or $\mathbf{S}$ which will restrict the set of possible solutions. In standard independent component analysis (ICA), we impose the constraint that the sources $s_1, \ldots, s_n$ are assumed to be statistically independent [1]. This constraint, together with a requirement that the distribution of the components s_j must be non-Gaussian, is normally sufficient to find a unique estimate of $\mathbf{A}$ and $\mathbf{S}$, apart from a scaling and permutation ambiguity (see e.g., [2]).

However, in many real-world problems, we know that the sources s_j must be *nonnegative*, i.e., that the sources must be either zero or positive [3]. For example, the amount of pollutant emitted by a factory is nonnegative [4], the probability of a particular topic appearing in a linguistic document is nonnegative [5] and note volumes in musical audio are nonnegative [6]. Several authors have proposed methods to solve (2) with nonnegativity constraints. In environmental modeling, Paatero and Tapper [4] introduced positive matrix factorization (PMF) as a method of solving (2) with positivity constraints on both $\mathbf{S}$ and $\mathbf{A}$. Building on PMF, Lee and Seung [3] introduced a set of efficient algorithms for performing nonnegative matrix factorization (NMF), applying these to, for example, the discovery of parts-based representations of face images [3] and the analysis of medical image sequences [7]. Neural network models with nonlinearities that impose a nonnegativity constraint have also been suggested [8]–[10]. However, the nonnegativity constraint alone is insufficient for (2) to yield a unique solution [11, p. 68]. Several methods have been proposed to overcome these ambiguities using *a priori* knowledge [12] and a sparseness requirement for the recovered sources has also been suggested [13].

Therefore, we propose the use of both independence and nonnegativity constraints together, i.e., that the sources s_j are assumed to be independent from each other and also nonnegative. (We impose no constraint on the positivity or otherwise of the elements of the mixing matrix $\mathbf{A}$.) We shall refer to the combination of these constraints on the sources s_j as *nonnegative ICA*.

In an earlier paper, the present author proposed a neural network for nonnegative ICA, which used rectified outputs and anti-Hebbian lateral inhibitory connections between the output units [14]. In this paper, we will consider several alternative nonnegative ICA algorithms, which are all based on a two-stage process common to many other ICA algorithms. First, we prewhiten the observed data, to remove any second-order dependencies (correlations); second, we perform an orthogonal

rotation of the whitened data to find the directions of the sources. However, instead of using the usual nongausianity measures, such as kurtosis, in the second "rotation" stage, we shall instead use our nonnegativity constraint.

In the following sections, we shall confirm that the combination of decorrelation and a nonnegativity constraint is sufficient for us to perform nonnegative ICA, under certain conditions. We shall then use this process to develop algorithms to perform nonnegative ICA, including a version of the *nonnegative PCA* algorithm for ICA and new batch algorithms which can perform separation quickly by using the concepts of geodesic search on the manifold of orthonormal matrices.

II. Prewhitening and Axis Rotations

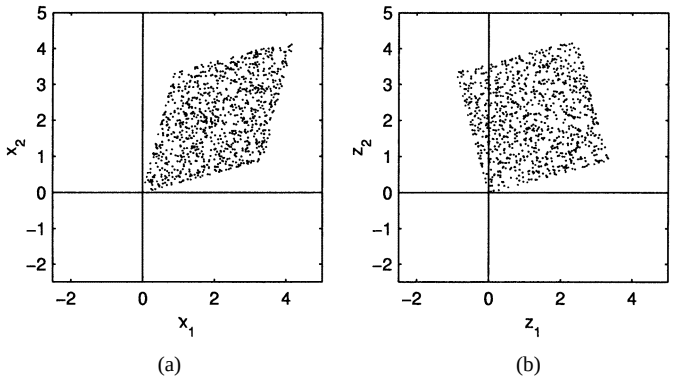
The first stage in our ICA process is to *whiten* the observed data $\mathbf{x}$ giving

$$\mathbf{z} = \mathbf{V}\mathbf{x} \tag{3}$$

where the $n \times n$ real whitening matrix $\mathbf{V}$ is chosen so that $\mathbf{C_z} = E\{(\mathbf{z} - \bar{\mathbf{z}})(\mathbf{z} - \bar{\mathbf{z}})^T\} = \mathbf{I}_n$ where $\bar{\mathbf{z}}$ is the mean of $\mathbf{z}$. If $\mathbf{E}$ is the orthogonal matrix of eigenvectors of $\mathbf{C_x} = E\{(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\}$ and $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$ is the diagonal matrix of correponding eigenvalues, so that $\mathbf{C_x} = \mathbf{E}\mathbf{D}\mathbf{E}^T$ and $\mathbf{E}^T\mathbf{E} = \mathbf{E}\mathbf{E}^T = \mathbf{I}_n$, then a suitable whitening matrix is $\mathbf{V} = \mathbf{C_x}^{-1/2} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T$ where $\mathbf{D}^{-1/2} = \mathrm{diag}(d_1^{-1/2}, \ldots, d_n^{-1/2})$ and $\mathbf{C_x}$ is normally estimated from the sample covariance [2]. Note that for nonnegative ICA, we do not remove the mean of the data in the whitening transform (3), since to do so would lose information about the nonnegativity of the sources [15].

Suppose that our sources s_j have unit variance, such that $\mathbf{C_s} = \mathbf{I}_n$. Then $\mathbf{I}_n = \mathbf{C_z} = \mathbf{V}\mathbf{A}\mathbf{C_s}(\mathbf{V}\mathbf{A})^T = \mathbf{V}\mathbf{A}(\mathbf{V}\mathbf{A})^T$ so the $\mathbf{s}$-to-$\mathbf{z}$ transform $\mathbf{V}\mathbf{A}$ is an orthonormal matrix. Therefore if $\mathbf{y} = \mathbf{W}\mathbf{z}$ where $\mathbf{W}$ is an orthonormal matrix, the $\mathbf{s}$-to-$\mathbf{y}$ transform $\mathbf{U} = \mathbf{W}\mathbf{V}\mathbf{A}$ must also be orthonormal. So, to find the original independent components, we need to find an orthonormal matrix $\mathbf{W}$ such that $\mathbf{y} = \mathbf{W}\mathbf{z} = \mathbf{U}\mathbf{s}$ is a permutation of the original sources $\mathbf{s}$, i.e., that the orthogonal matrix $\mathbf{U} = \mathbf{W}\mathbf{V}\mathbf{A}$ is a permutation matrix. Fig. 1 illustrates the process of whitening for nonnegative data in two dimensions. Whitening has succeeded in making the axes of the original sources orthogonal to each other [Fig. 1(b)], but there is a remaining orthonormal rotation to be performed. A typical ICA algorithm might search for a rotation that makes the resulting outputs as "non-Gaussian" as possible, for example by finding an extremum of kurtosis, since any sum of independent random variables will make the result "more Gaussian" [2].

However, Fig. 1 immediately suggests another approach: that we should search for a rotation where all of the data fits into the positive quadrant. As long as the distribution of each of the original sources is "tight" down to zero, then it is intuitively clear that this will be a unique solution, apart from a permutation and scaling of the axes. We shall call a source s *well-grounded*, if it has a nonzero pdf in the positive neighborhood of $s = 0$, i.e., for any $\delta > 0$ we have $\Pr(s < \delta) > 0$. Using this definition, we can state the following theorem which shows that our intuitive idea generalizes to n dimensions [15]:



(a) (b)

Fig. 1. Original data (a) is whitened (b) to remove second-order correlations.

Theorem 1: Let $\mathbf{s}$ be a random vector of real-valued, nonnegative and well-grounded independent sources, each with unit variance and let $\mathbf{y} = \mathbf{U}\mathbf{s}$ be an orthonormal rotation of $\mathbf{s}$ (i.e., $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}_n$). Then $\mathbf{U}$ is a permutation matrix if and only if $\mathbf{y}$ is nonnegative with probability 1.

In other words, if we search for some orthogonal rotation $\mathbf{W}$ such that $\mathbf{y} = \mathbf{W}\mathbf{z} = \mathbf{W}\mathbf{V}\mathbf{A}\mathbf{s}$ is nonnegative, then $\mathbf{y}$ must be a permutation of the original source vector $\mathbf{s}$, together with a scaling ambiguity if the original sources did not have unit variance.

A natural way to do this is to construct a cost function $J(\mathbf{W})$ for which $J = 0$ if and only if $\mathbf{W}$ is orthogonal and $\mathbf{y}$ is nonnegative with probability 1. Then finding the zero point of this cost function will also find the independent sources [15]. Specifically, we can choose the squared-reconstruction-error cost function

$$J = J_2(\mathbf{W}) = \frac{1}{2}E\left(\left|\mathbf{z} - \mathbf{W}^T\mathbf{y}^+\right|^2\right) \tag{4}$$

where $\mathbf{y}^+ = (y_1^+, \ldots, y_n^+)$ with $y_i^+ = \max(y_i, 0)$ is the rectified version of $\mathbf{y} = (y_1, \ldots, y_n)$. Thus we wish to construct an algorithm that searches for an orthonormal weight matrix $\mathbf{W}$ that minimizes the cost function J.

III. Nonnegative PCA

Minimization of the least mean squared reconstruction error (4) has been proposed as an objective principle for many neural-network principal component analysis (PCA) algorithms and PCA subspace algorithms [8]. In particular, this led to the *nonlinear PCA* algorithm [16], where $\mathbf{W}(t+1) = \mathbf{W}(t) + \Delta\mathbf{W}$ with

$$\Delta\mathbf{W} = \eta_t \mathbf{g}(\mathbf{y})\left[\mathbf{z}^T - \mathbf{g}(\mathbf{y}^T)\mathbf{W}\right] \tag{5}$$

where $\mathbf{g}(\mathbf{y}^T) = (g(y_1), \ldots, g(y_n))^T$ and $g(\cdot)$ is a nonlinear function. Algorithm (5) is a nonlinear version of the Oja and Karhunen PCA subspace algorithm [17], [18], [also introduced by Williams as his symmetric error correction (SEC) network [19]], which used this algorithm with $g(y) = y$ (and normally had a nonsquare $n \times m$ matrix $\mathbf{W}$ with $n < m$). The nonlinear PCA algorithm was shown to perform ICA on whitened data, if $g(\cdot)$ is an odd, twice-differentiable function [20].

Thus an obvious suggestion for our nonnegative ICA problem is the nonlinear PCA algorithm (5) with the rectification nonlinearity $g(y) = y^+ = \max(y, 0)$, giving us

$$\Delta \mathbf{W} = \eta_t \mathbf{y}^+ \left[\mathbf{z}^T - \left(\mathbf{y}^+ \right)^T \mathbf{W} \right] \tag{6}$$

which we call the *nonnegative PCA* algorithm. Simulation results (see e.g., Section VI) indicate that (6) is effective in separating nonnegative sources. However, the rectification nonlinearity $g(y) = \max(y, 0)$ is neither an odd function, nor is it twice differentiable, so the standard convergence proof for nonlinear PCA algorithms does not apply. Its behavior will be considered in more detail elsewhere [21].

IV. Axis Pair Rotation

Recall from the previous section that our task is to find an $n \times n$ orthogonal matrix $\mathbf{W}$ with $\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^T = \mathbf{I}_n$ which minimizes the mean squared reconstruction error J in (4). While the nonnegative PCA algorithm (6) does find an orthogonal matrix $\mathbf{W}$, it does not *constrain* $\mathbf{W}$ to be orthonormal during the search. If we were able to force $\mathbf{W}$ to remain orthonormal, we may be able to construct faster search algorithms.

A. Rotation in Two Dimensions

Comon [1] proposed that such an orthonormal rotation matrix for ICA could be constructed from a product of simple two–dimensional (2-D) rotations, with each rotation taking the form

$$\begin{pmatrix} y_{i_1} \\ y_{i_2} \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} z_{i_1} \\ z_{i_2} \end{pmatrix}. \tag{7}$$

Therefore, let us consider building an update algorithm for $\mathbf{W}$ in the simplest nontrivial case, where $n = 2$.

Consider the system illustrated in Fig. 2. Here, we have $\mathbf{z} = (l\cos\psi, l\sin\psi)$ where $\psi = \phi + \theta$, a 2-D orthogonal rotation matrix

$$\mathbf{W} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \tag{8}$$

and hence $\mathbf{y} = (y_1, y_2) = (l\cos\theta, l\sin\theta)$. The matrix $\mathbf{W}$ is determined by the rotation angle ϕ and is constrained to be orthogonal. Now, the reconstruction error J will depend on the y axis quadrant in which the input data point $\mathbf{z}$ falls. Specifically

$$2J = \begin{cases} 0 & \text{if } y_1 \geq 0, y_2 \geq 0 \\ y_2^2 & \text{if } y_1 \geq 0, y_2 < 0 \\ y_1^2 & \text{if } y_1 < 0, y_2 \geq 0 \\ y_1^2 + y_2^2 = l^2 & \text{otherwise.} \end{cases} \tag{9}$$

Differentiating (9) with respect to θ and noticing that $dy_1/d\theta = -l\sin\theta = -y_2$ and $dy_2/d\theta = l\cos\theta = y_1$ we get

$$\frac{dJ}{d\theta} = \begin{cases} 0 & \text{if } y_1 \geq 0, y_2 \geq 0 \\ y_2 y_1 & \text{if } y_1 \geq 0, y_2 < 0 \\ -y_1 y_2 & \text{if } y_1 < 0, y_2 \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$= y_1^+ y_2^- - y_1^- y_2^+ \tag{11}$$

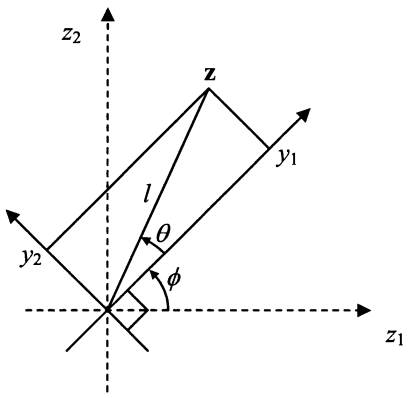


Fig. 2. Torque in the 2-D case.

where $y_j^+ = \max(y_j, 0)$ and $y_j^- = y_j - y_j^+$. Thus, we have

$$\frac{dJ}{d\phi} = -\frac{dJ}{d\theta} = -\left(y_1^+ y_2^- - y_1^- y_2^+ \right) \tag{12}$$

since $\phi = \psi - \theta$ where ψ is constant for a given input $\mathbf{z}$. We note that $dJ/d\phi$ is continuous, but that (12) is itself nondifferentiable at the quadrant boundaries, where one of y_1 or y_2 is zero.

It can sometimes be helpful to consider an interpretation of these algorithms in terms of a mechanical system, such as that by Fiori for PCA systems [22]. Here, this would lead to an interpretation of $\tau = -dJ/d\phi$ as a "torque" which, if allowed to rotate the y axis system within z space, would tend to reduce the "energy" J.

To construct the natural stochastic approximation gradient descent algorithm, we simply move ϕ "downwards" in proportion to $dJ/d\phi$, giving us the algorithm

$$\phi(t+1) = \phi(t) - \eta_\phi \cdot \frac{dJ}{d\phi} \tag{13}$$

$$= \phi(t) + \eta_\phi \left(y_1^+ y_2^- - y_1^- y_2^+ \right). \tag{14}$$

Now, a gradient descent algorithm such as this is typically limited to taking small steps in order to ensure good convergence to the solution. However, in contrast to the nonnegative PCA algorithm, we can take much larger steps if we wish, since the parameterization in terms of ϕ ensures that $\mathbf{W}$ always remains orthonormal. Therefore, we can use faster search algorithms, such as line search, to improve the speed of our algorithm.

B. Line Search Over Rotation Angle

While the analysis above does give us a natural stochastic approximation algorithm, we can also perform the minimization of J faster by performing a "line" search over the angle ϕ to minimize J, or alternatively to find a zero of the torque $-dJ/d\phi$. There are many standard methods available to find the zero of a function [23], with the Matlab `fzero` function, a convenient one that we have used.

For the 2-D nonnegative ICA problem, there is additional knowledge we can use to make this line search easier. Specifically, we know that if the sources are nonnegative and well-grounded (i.e., have nonzero pdf down to zero), then there must be a unique minimum where the outputs are all in the positive

quadrant [15], i.e., where $J = 0$ and $dJ/d\phi = 0$. Using a locally quadratic approximation of the error function J, from the point $\phi(t)$ with error $J(t)$ and derivative $dJ(t)/d\phi$ we can step to the locally quadratic estimate of the zero error at

$$\phi(t+1) = \phi(t) - \frac{2J(t)}{\left(\frac{dJ(t)}{d\phi}\right)}. \tag{15}$$

In test simulations, we have found that the update algorithm (15) is often sufficient on its own to find the zero of J in a few steps, without recourse to a full line search algorithm. Differentiation of (12) for the condition where only a single data point of a batch is just outside of the positive quadrant (i.e., $|y^+| \gg |y^-|$ for $y^+ > 0 > y^-$) confirms that the solution is locally quadratic, but the curvature increases if more data points "escape" outside of the positive quadrant.

C. Generalizing to More Than Two Dimensions

As we mentioned above, Comon [1] pointed out that a general n-dimensional orthonormal transform can be formed from a product of 2-D rotations, taking care that the order of application is important since such rotation matrices are not commutative in general. For a cumulant-based ICA algorithm, he proposed forming each major update to the orthogonal transform by sweeping through all $n(n-1)/2$ rotations [1, Algorithm 18]. As an alternative approach, we propose calculating the torque values for each axis pair before each line search and choosing to rotate the axis pair with the highest torque. In a Euclidean optimization algorithm, this would be equivalent to choosing to perform a line search along the axis with the steepest gradient.

Another difference from traditional ICA algorithms is that we have a relatively simple second-order error to minimize, instead of a contrast function based on higher order statistics. Therefore, we obtain the following algorithm.

1) Start with white (but not zero-mean) data set $\mathbf{Z} = [\mathbf{z}_1, \ldots \mathbf{z}_p]$ and set $\mathbf{Z}(0) = \mathbf{Z}$ and $\mathbf{W}(0) = \mathbf{I}_n$ for step $t = 0$.
2) Calculate the output $\mathbf{Y} = \mathbf{Z}(t) = \mathbf{W}(t)\mathbf{Z}(0)$ and rectified parts $\mathbf{Y}_+$ with $y_{ik}^+ = \max(y_{ik}, 0)$ and $\mathbf{Y}_-$ with $y_{ik}^- = \min(y_{ik}, 0)$.
3) Calculate the torque values

$$g_{ij} = \sum_k y_{ik}^+ y_{jk}^- - y_{ik}^- y_{jk}^+ \tag{16}$$

for axis pairs $i < j$ (noting that $g_{ji} = -g_{ij}$).
4) If the maximum value of $|g_{ij}|$ is less than the tolerance required, stop.
5) Choose the axis pair i^*, j^* for which $|g_{ij}|$ is maximum and construct the $2 \times p$ matrix $\mathbf{Z}^*$ from selecting rows i^* and j^*, respectively, from $\mathbf{Z}(t)$.
6) Using this reduced data $\mathbf{Z}^*$, perform a 2-D rotation line search to minimize the error $J_{i^*j^*}$ using a suitable line search method and locate the rotation angle $\phi^*(t+1)$ corresponding to the minimum of the error term.
7) Form the rotation matrix $\mathbf{R}(t+1) = [r(t+1)_{ij}]$ from $\phi^*(t+1)$, where $r_{i^*i^*} = r_{j^*j^*} = \cos(\phi^*)$, $r_{i^*j^*} = \sin(\phi^*)$, $r_{j^*i^*} = -\sin(\phi^*)$, $r_{ii} = 1$ for all $i \neq i^*, j^*$ and all other entries of $\mathbf{R}$ are zero.
8) Form the updated weight matrix $\mathbf{W}(t+1) = \mathbf{R}(t+1)\mathbf{W}(t)$ and modified input data $\mathbf{Z}(t+1) = \mathbf{R}(t+1)\mathbf{Z}(t) = \mathbf{W}(t+1)\mathbf{Z}(0)$.
9) Increment the step count t and repeat from Step 2) until the maximum torque at Step 4) is sufficiently close to zero.

Note that at Step 6), it is only necessary to measure the components of J due to the projection onto the axis pair which is being rotated, since the components of J due to the other axes will remain unchanged during this rotation. At the end of this algorithm, $\mathbf{W}(t)$ should correspond to the required forward weight vector.

Simulations (see Section VI) indicate that this method can converge to a good solution in just a few iterations. In the example considered there, the quadratic step given in (15) appears to work well for the first few steps of a line search, but we would expect it to work less well as the search nears a minimum for which $J > 0$. Experiments on higher dimensional problems confirm that this is less reliable on more difficult problems and a full line search may need to be used.

As we mentioned above, we can consider this to be the equivalent of a line search in a more typical Euclidean search problem where we only move along each axis. In the next section, we shall see that by considering the geometry of the manifold over which we wish to search, we can construct another family of algorithms based on the concept of line search over more general geodesics.

V. Geodesic Search

Recently, several authors have studied PCA and other prewhitened ICA tasks as a problem of constrained optimization [22], [24]–[26]. Specifically, to find an extremum of a function $J(\mathbf{W})$ over some matrix $\mathbf{W}$ which is constrained to be orthonormal, we will be performing a search over a Grassman or Stiefel manifold [27]. The Grassman manifold arises under the homogeneity condition $J(\mathbf{W}) = J(\mathbf{QW})$ for any square orthonormal matrix $\mathbf{Q}$, which is the case for the PCA subspace network. The Stiefel manifold arises where this homogeneity condition does not hold, such as for true PCA or an ICA task, including our nonnegative ICA problem. Here, a rotation of the outputs will cause the error value (or the contrast function in the usual ICA formulation) to change.

Traditional "Euclidean" learning algorithms, such as the subspace algorithm, nonlinear PCA, or the nonnegative PCA algorithm we considered in Section III, find the local derivative of the error or contrast function with respect to the weight matrix $\mathbf{W}$ and update $\mathbf{W}$ according to that derivative. However, this update will be tangential to the (curved) Stiefel manifold itself, within which $\mathbf{W}$ is supposed to be constrained, so either the algorithm must correct any off-constraint movement itself, or a correction must be made to bring $\mathbf{W}$ back to the constraint surface of orthogonality. On the other hand, if the special geometry of the Stiefel manifold is taken into account, a modified form of update rule may be constructed where small updates will stay on (or closer to) the surface of the manifold, reducing the need for constraint applications as the algorithm proceeds. See [28] for a useful overview of these methods.

One particular concept that offers the potential for fast batch methods for nonnegative ICA is that of the *geodesic*: the shortest path between two points on a manifold [27]. In this paper, we restrict our consideration to the case $n = m$ of square matrices $\mathbf{W}$. In this case, the geodesic equation takes the simple form [27, eq. 2.14]

$$\mathbf{W}(\tau) = e^{\tau \mathbf{B}} \mathbf{W}(0) \tag{17}$$

where $\mathbf{B}$ is an $n \times n$ skew-symmetric matrix ($\mathbf{B}^T = -\mathbf{B}$) and τ is a scalar parameter determining the position along the geodesic. Thus the tangent space at $\mathbf{W}(0)$ has $n(n-1)/2$ degrees of freedom [27]. For $n = 2$, we can write

$$\mathbf{B} = \begin{pmatrix} 0 & b \\ -b & 0 \end{pmatrix} \tag{18}$$

giving

$$\mathbf{W}(\tau) = e^{\tau \mathbf{B}} \mathbf{W}(0) = \begin{pmatrix} \cos(\tau b) & \sin(\tau b) \\ -\sin(\tau b) & \cos(\tau b) \end{pmatrix} \mathbf{W}(0) \tag{19}$$

which is (8) with $\phi = \tau b$. Therefore, this geodesic approach is a generalization of the rotation method we described in Section IV.

Fiori [28] and Nishimori [26] have proposed the use of this concept to update $\mathbf{W}$ in ICA algorithms. Specifically, at a given time step t the update equation is

$$\mathbf{W}(t+1) = e^{-\eta \mathbf{B}(t)} \mathbf{W}(t) \tag{20}$$

where $\mathbf{B}(t)$ is skew-symmetric and η is a small positive constant. If $\mathbf{W}(t)$ is orthonormal, then so is $\mathbf{W}(t+1)$[28]. Such approaches have tended to work in the space of weight matrices $\mathbf{W}$, using the manifold structure to find a new $\mathbf{W}$ a short distance away [25], [26], [28]. However, Nishimori also suggests that other optimization methods, such as the Newton method, might be used [29], although the formulation in terms of $\mathbf{W}$ can appear complex. In this paper, we will instead work in the space of the coordinates of $\mathbf{B}$, converting to $\mathbf{W}$ as necessary to calculate $\mathbf{y} = \mathbf{W}\mathbf{z}$.

For nonnegative ICA we find that we cannot use the Newton method directly. Specifically, the derivative of the error function J is nondifferentiable at any point where $y_{jk} = 0$ for some j, k, i.e., when the mapping of any data point onto $\mathbf{y}$ crosses the positive octant boundary (see (12) for the 2-D case) so we cannot calculate a true Hessian. We will therefore instead look to construct a learning algorithm based on steepest-descent "line" search over these geodesics.

A. Finding the Steepest-Descent Geodesic

We wish to construct an algorithm which, at each step, searches over a geodesic for the minimum of J. We choose the geodesic in the direction of steepest descent, i.e., the one for which J is decreasing fastest for a given small distance moved. Consider (17), where we write $\mathbf{B} = \Phi - \Phi^T$ for an underlying upper-triangular parameter matrix $\Phi = [\phi_{ij}]$, constrained such that ϕ_{ij} for $i < j$ are the free parameters, with $\phi_{ij} = 0$ for $i \geq j$. For the steepest descent direction, we wish to choose Φ which maximizes the expression

$$-\lim_{\delta\tau \to 0} \frac{\frac{(\text{change in } J(\tau\Phi) \text{ due to } \delta\tau)}{\delta\tau}}{\frac{(\text{distance moved by } \tau\Phi \text{ due to } \delta\tau)}{\delta\tau}}. \tag{21}$$

For the numerator, we have

$$\lim_{\delta\tau \to 0} \frac{\delta J}{\delta \tau} = \frac{dJ}{d\tau} \tag{22}$$

while for the denominator, we measure "distance" r using the Euclidean distance between the elements $m_{ij}^{(1)}$ of $\mathbf{M}^{(1)} = \tau\Phi$ and $m_{ij}^{(2)}$ of $\mathbf{M}^{(2)} = (\tau + \delta\tau)\Phi$, i.e.,

$$r = ||(\tau + \delta\tau)\Phi - \tau\Phi||_F = |\delta\tau| \cdot ||\Phi||_F \tag{23}$$

where $||M||_F = \sqrt{\sum_{ij} m_{ij}^2}$ is the Frobenius norm of $\mathbf{M}$. Therefore, we have

$$\lim_{\delta\tau \to 0} \frac{\delta r}{\delta \tau} = \frac{\delta\tau ||\Phi||_F}{\delta\tau} = ||\Phi||_F \tag{24}$$

where we assume that $\delta\tau > 0$. We therefore wish to find $\arg\max_\Phi -(dJ/d\tau)/||\Phi||_F$. Now we have

$$J = \frac{1}{2}\text{trace}((\mathbf{Z} - \hat{\mathbf{Z}})^T(\mathbf{Z} - \hat{\mathbf{Z}})) \tag{25}$$

$$= \frac{1}{2}\text{trace}(\mathbf{Y}_-^T \mathbf{Y}_-) \tag{26}$$

$$= \frac{1}{2}\sum_{kl} (y_{\bar{kl}})^2 \tag{27}$$

so

$$\frac{dJ}{dw_{ij}} = \sum_{kl} y_{\bar{kl}} \frac{d}{dw_{ij}} y_{\bar{kl}}. \tag{28}$$

Now, since $y = y^+ + y^-$ and $y^+ y^- = 0$, to first order we have $y^-(dy_-/dw_{ij}) = y^-(dy/dw_{ij})$ and after some manipulation we get

$$\frac{dJ}{d\mathbf{W}} = \mathbf{Y}_- \mathbf{Z}^T. \tag{29}$$

We also have

$$\frac{d\mathbf{W}}{d\tau} = \frac{d}{d\tau}\left(e^{\tau \mathbf{B}} \mathbf{W}(0)\right) = \mathbf{B} e^{\tau \mathbf{B}} \mathbf{W}(0) = \mathbf{B}\mathbf{W} \tag{30}$$

leading to

$$\frac{dJ}{d\tau} = \left\langle \frac{dJ}{d\mathbf{W}}, \frac{d\mathbf{W}}{d\tau} \right\rangle \tag{31}$$

$$= \left\langle \mathbf{Y}_- \mathbf{Z}^T, \mathbf{B}\mathbf{W} \right\rangle \tag{32}$$

$$= \text{trace}\left(\mathbf{Y}_- \mathbf{Z}^T \mathbf{W}^T \mathbf{B}^T\right) \tag{33}$$

where $\langle \mathbf{C}, \mathbf{D} \rangle = \text{trace}(\mathbf{C}\mathbf{D}^T)$ is the inner product between matrices $\mathbf{C}$ and $\mathbf{D}$. Now $\mathbf{Y} = \mathbf{W}\mathbf{Z}$ and $\mathbf{B} = \Phi - \Phi^T$ so

$$\frac{dJ}{d\tau} = \text{trace}\left(\mathbf{Y}_- \mathbf{Y}^T \left(\Phi^T - \Phi\right)\right) \tag{34}$$

$$= \text{trace}\left(\left(\mathbf{Y}_- \mathbf{Y}^T - \mathbf{Y}\mathbf{Y}_-^T\right)\Phi^T\right) \tag{35}$$

$$= \text{trace}\left(\left(\mathbf{Y}_- \mathbf{Y}_+^T - \mathbf{Y}_+ \mathbf{Y}_-^T\right)\Phi^T\right) \tag{36}$$

$$= \left\langle \mathrm{UT}\left(\mathbf{Y}_{-}\mathbf{Y}_{+}^{T} - \mathbf{Y}_{+}\mathbf{Y}_{-}^{T}\right), \Phi \right\rangle \tag{37}$$

where the upper triangular operator $\mathrm{UT}(\cdot)$ has been introduced since Φ itself is zero on or below the diagonal. Thus, the steepest descent direction for Φ is that which maximizes

$$\left\langle \mathrm{UT}\left(\mathbf{Y}_{-}\mathbf{Y}_{+}^{T} - \mathbf{Y}_{+}\mathbf{Y}_{-}^{T}\right), \frac{\Phi}{||\Phi||_F} \right\rangle \tag{38}$$

which is maximized when Φ is in the direction of the first term in the inner product. Hence we have the steepest descent direction of J in Φ-space as

$$\Phi_{sd} \propto -\mathrm{UT}\left(\mathbf{Y}_{-}\mathbf{Y}_{+}^{T} - \mathbf{Y}_{+}\mathbf{Y}_{-}^{T}\right) \tag{39}$$

where $\mathrm{UT}(\cdot)$ together with the skew-symmetry of its contents ensures that only the valid parameters of Φ_{sd} are nonzero. To construct a steepest gradient descent algorithm, we would simply move Φ by a small amount in the direction of (39), producing a small multiplicative update step for $\mathbf{W}$ of

$$\mathbf{W}(t+1) = e^{-\eta(\mathbf{Y}_{-}\mathbf{Y}_{+}^{T} - \mathbf{Y}_{+}\mathbf{Y}_{-}^{T})}\mathbf{W}(t) \tag{40}$$

which is a direct application of the geodesic flow method [26], [28].

B. Line Search Along a Geodesic

Now that we have the geodesic in the direction of steepest descent, we can also perform a "line" search to find the minimum for J by taking larger jumps along this geodesic. A simple repeating line search algorithm is as follows.

1) Start with white (but not zero-mean) data set $\mathbf{Z}$ and set $\mathbf{Z}(0) = \mathbf{Z}$ and $\mathbf{W}(0) = \mathbf{I}$ for step $t = 0$.
2) Calculate the output $\mathbf{Y} = \mathbf{Z}(t) = \mathbf{W}(t)\mathbf{Z}(0)$ with rectified parts $\mathbf{Y}_{+}$ and $\mathbf{Y}_{-}$.
3) Calculate the Φ-gradient $\mathbf{G}(t) = \mathrm{UT}(\mathbf{Y}_{-}\mathbf{Y}_{+}^{T} - \mathbf{Y}_{+}\mathbf{Y}_{-}^{T})$ and $\mathbf{B}$-space movement direction $\mathbf{H}(t) = -(\mathbf{G}(t) - \mathbf{G}(t)^{T}) = \mathbf{Y}_{+}\mathbf{Y}_{-}^{T} - \mathbf{Y}_{-}\mathbf{Y}_{+}^{T}$.
4) Stop if $\|\mathbf{G}(t)\|$ is below a tolerance threshold
5) Perform a line search for τ^{*} which minimizes

$$J(\tau) = \frac{1}{2}\mathrm{trace}\left(\mathbf{Y}_{-}^{T}(\tau)\mathbf{Y}_{-}(\tau)\right) \tag{41}$$

where $\mathbf{Y}_{-}$ is the negative part of $\mathbf{Y}(\tau) = \mathbf{R}(\tau)\mathbf{Z}(t)$ and $\mathbf{R}(\tau) = e^{-\tau\mathbf{H}}$.
6) Form the updated weight matrix $\mathbf{W}(t+1) = \mathbf{R}(\tau^{*})\mathbf{W}(t)$ and modified input data $\mathbf{Z}(t+1) = \mathbf{R}(\tau^{*})\mathbf{Z}(t) = \mathbf{W}(t+1)\mathbf{Z}(0)$.
7) Increment the step count t and repeat from Step 2) until the norm of the gradient at Step 4) is close enough to zero.

In the line search of Step 5), we have

$$\frac{dJ}{d\tau} = \left\langle \mathrm{UT}(\mathbf{Y}(\tau)_{-}\mathbf{Y}(\tau)_{+}^{T} - \mathbf{Y}(\tau)_{+}\mathbf{Y}(\tau)_{-}^{T}), \mathbf{G}(t) \right\rangle \tag{42}$$

$$= \frac{1}{2}\left\langle \mathbf{Y}(\tau)_{+}\mathbf{Y}(\tau)_{-}^{T} - \mathbf{Y}(\tau)_{-}\mathbf{Y}(\tau)_{+}^{T}, \mathbf{H}(t) \right\rangle \tag{43}$$

so we can search for a zero of this expression. In the first few steps of our line search, we can take advantage of the known zero-minimum of J, before we pass on to standard methods such as the Matlab `fzero` function. Matlab also uses efficient algorithms to calculate the required matrix exponential.

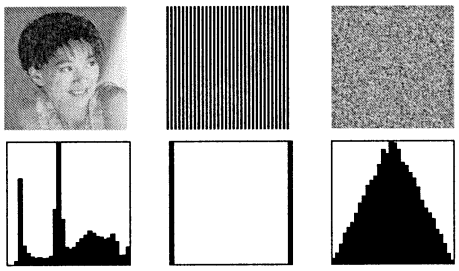


Fig. 3. Source images and histograms used for the nonnegative ICA algorithms. (The images were kindly supplied by W. Kasprzak and A. Cichocki.)

C. Single Step Algorithm

As a simple alternative to the line search, we may assume that we are in a quadratic bowl pointing in the direction of the minimum with value $J = 0$ and make a step equivalent to (15). When there is a single negative data point remaining, this will be an exact solution: with more data points remaining, we will expect this step not to be far enough since the curvature increases as more data points emerge from the positive quadrant.

This seems to perform quite well on a simple three-source separation problem (see simulations in Section VI), but on more complex problems might get caught out by "valleys" in the search space. A more robust algorithm may combine this method with a line search algorithm which is used when the quadratic step leads to an increase in the error, indicating that the minimum has been bracketed.

VI. Simulation Results

We illustrate the operation of some of these algorithms using a blind image separation problem (see, e.g., [30]). This is suitable for nonnegative ICA, since the source images have nonnegative pixel values. The original images used in this section are shown in Fig. 3. They are square 128×128 images (downsampled by a factor of four from the original 512×512 images) with integer pixel intensities between 0 and 255 inclusive, which were then scaled to unit variance. Each source sequence s_k is considered to be the sequence of pixel values obtained as we scan across the image from top left ($k = 0$) to bottom right ($k = 128^2$).

We found that face images tended to have significant correlation with other face images, breaking the independence assumption of ICA methods. Consequently, we used one face and two artificial images as the sources for these demonstrations. Note that the histograms indicate that the face image has a nonzero minimum value, which does violate our assumption that the sources are *well-grounded* [15]. Nevertheless, we will see that we will get reasonable (although not perfect) separation performance from these nonnegative ICA algorithms.

To measure the separation performance of the algorithm, we use two performance measures: a nonnegative reconstruction error

$$e_{NNR} = \frac{1}{np}\left\|\mathbf{Z} - \mathbf{W}^{T}\mathbf{Y}_{+}\right\|_F^2 \tag{44}$$

which is a scaled version of J and a cross-talk error

$$e_{XT} = \frac{1}{n^2}\left\|\mathrm{abs}(\mathbf{WVA})^{T}\mathrm{abs}(\mathbf{WVA}) - \mathbf{I}_n\right\|_{\mathrm{F}}^2 \tag{45}$$

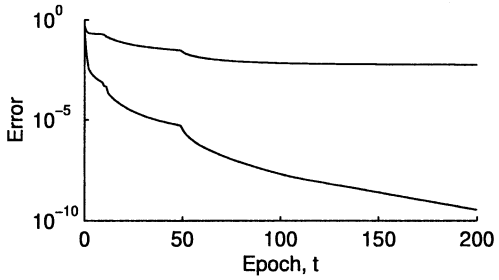


Fig. 4. Learning curve for the nonnegative PCA algorithm, showing nonnegative reconstruction error (lower curve) and crosstalk error (upper curve).

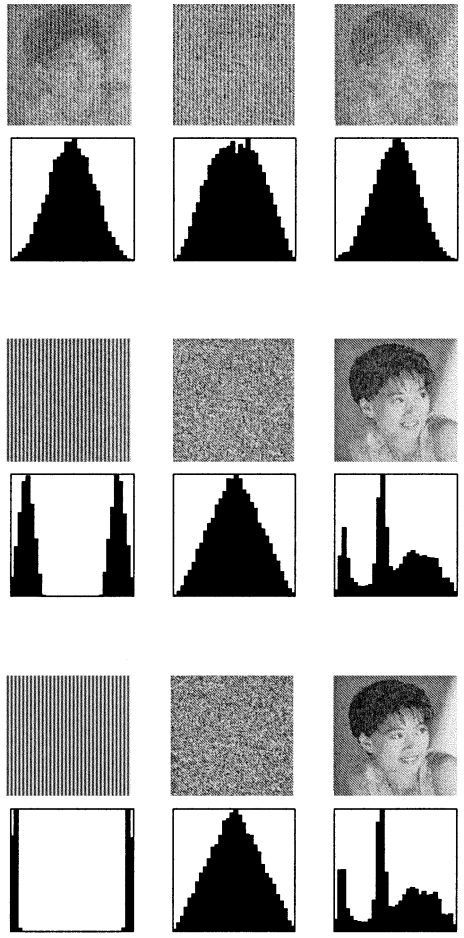


Fig. 5. Image separation process for the nonnegative PCA algorithm, showing (a) the initial state and progress after (b) 50 epochs and (c) 200 epochs.

where abs$(\mathbf{M})$ is the matrix of absolute values of the elements of $\mathbf{M}$, which is zero only if $\mathbf{y} = \mathbf{WVAs}$ is a permutation of the sources, i.e., only if the sources have been successfully separated.

A. Nonlinear PCA Algorithm

Fig. 4 gives an example learning curve for the nonnegative PCA algorithm of Section III.

The learning rate was manually adjusted to improve the convergence time, using $\eta = 10^4$ initially, $\eta = 10^3$ for $10 \leq t < 50$ and $\eta = 10^2$ for $t \geq 50$. As the algorithm progresses, the images are successfully separated and the histograms become less Gaussian and closer to those of the original sources (Fig. 5). As a nonnegative algorithm, we never see inverted sources recovered, such as we might expect from traditional ICA algorithms.

From the curves, we see that the nonnegative reconstruction error is decreasing steadily after the initial stage. However, the crosstalk error, measuring the distance away from separation, reaches a minimum of 5.73×10^{-3} after 200 epochs.

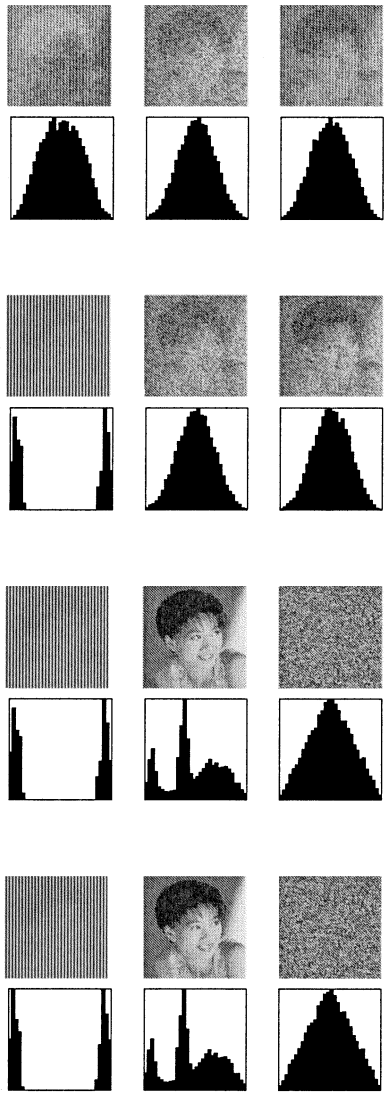


Fig. 6. Image separation after five iterations of 2-D rotation algorithm. The effect of successive rotations can clearly be seen.

B. Torque-Based Rotation Algorithm

For separation of three images using successive rotations (Section IV-C) we used the quadratic step update within the line search and the rotation directions were permitted to change every five epochs. Here the algorithm first selected axes 1–3 (Epochs 1 to 5), then 2–3 (Epochs 6 to 10 and 11 to 15), then axes 1–3 (Epochs 16 to 20 and 21). We can clearly see this sequence in the output images (Fig. 6) with the restarts at 5, 15, and 20 visible on the learning curve (Fig. 7). The final crosstalk error is 7.52×10^{-3}.

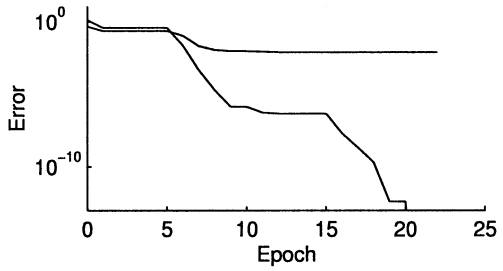


Fig. 7. Learning curves for the successive rotation algorithm, showing nonnegative reconstruction error (lower curve after five epochs) and crosstalk error (upper curve after five epochs).

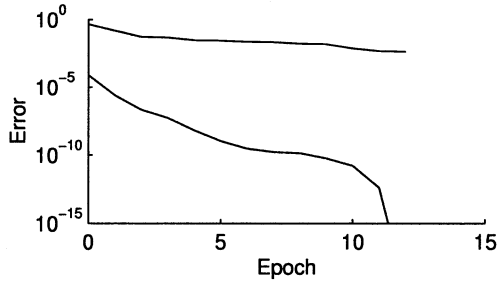


Fig. 8. Error curves for the geodesic step algorithm.

C. Geodesic Step Algorithm

The geodesic step also finds a good reconstruction in a few iterations (Fig. 8). The final crosstalk error for this algorithm was 4.32×10^{-3}, which was found in 12 passes through the data set. The images and histograms at the solution are qualitatively similar to Fig. 6(d) and are not reproduced here.

VII. Application to Analysis of Musical Audio

We used our nonnegative ICA approach to analyze a short extract from a polyphonic musical audio signal, i.e., an audio signal where more than one note may be played at once [6]. We consider the note volumes to be our nonnegative source signals and we assume that notes are played approximately independently from each other. Since the short-term power spectrum of the observed audio signal is approximately the sum of the power spectra due to each note, the linear ICA approach can be applied to this problem.

We used a short segment of a Liszt "Etude," played on a MIDI synthesized piano [Fig. 9(a)], with the resulting audio sampled at 16 kHz with 16 bits resolution. The audio was transformed to a power spectrogram using 512-sample frames with a Hanning window and 50% overlap between frames. The redundant negative frequency bins were discarded, yeilding a power spectrum with 467 frames of 257 frequency bins each [Fig. 9(b)]. Note that the frequency components of a note often remain beyond the end of the corresponding MIDI notes, as the note gradually decays away. The observation vectors were reduced from 257 to ten dimensions using PCA, being careful to retain the mean of the data in the resulting ten-dimensional vectors. This resulted in an observation matrix $\mathbf{X}$ of size $n \times p = 10 \times 467$, capturing 99.64% of the variance in the original power spectrum. This was then prewhitened to give a whitened observation matrix $\mathbf{Z}$ according to (3).

We used the geodesic flow algorithm (40) with update factor $\eta = 0.01$, from a starting point of $\mathbf{W} = \mathbf{I}_n$ until covergence

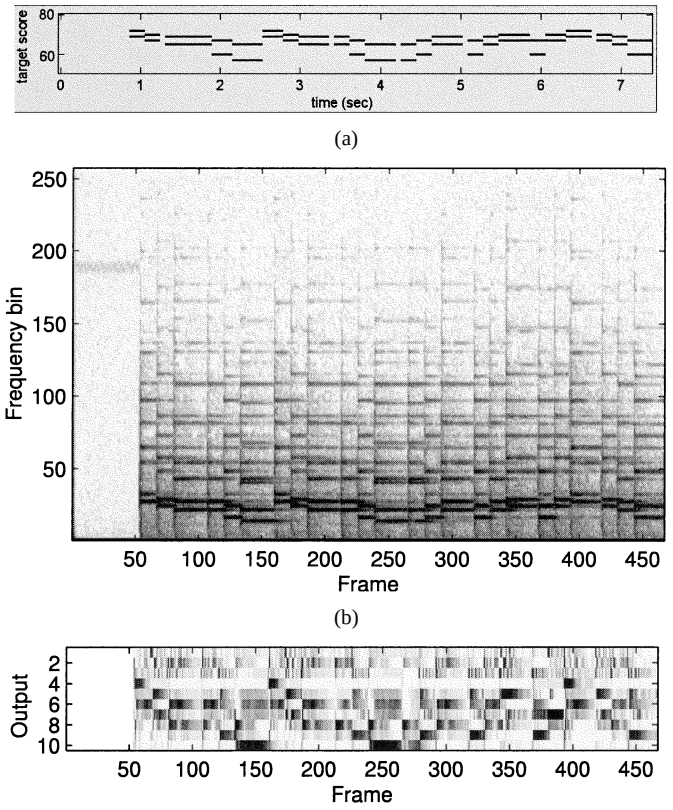


Fig. 9. Results of the music analysis task, showing (a) the original MIDI notes, (b) the power spectrogram of the audio, and (c) the output from the nonnegative ICA network.

was observed after 350 iterations. The resulting rectified output $\mathbf{Y}_+$ is shown in Fig. 9(c): the outputs have been permuted manually and contrast-enhanced using a grey scale of $\tanh(0.4y_+)$ to allow a visual comparison with the original MIDI note tracks [Fig. 9(a)]. From the figure, we can see that of the ten extracted "independent" components, the lower seven broadly represent the seven notes present in this extract. The remaining three outputs appear to respond to transitions such as note onsets, corresponding to frames which would contain a wider range of frequency components than those found in the main part of a note.

These results are quite encouraging, given that there are significant objections which can be made to a number of our assumptions on this task. First, the notes are not truly independent: in this piece they tend to be played in pairs, which may be one reason why, e.g., output eight appears to respond only near the onset of notes. Second, power spectra only add approximately: the true sum depends on the relative phases of the frequency components, which may lead to a larger effective noise than would otherwise be expected. Third, the spectra of musical notes rarely take the form of a simple power spectrum vector multiplied by a vector that varies over time, as assumed in the ICA model: rather, there is more high-frequency ("bright") spectral content at the start of a note, which decay faster than the low- frequency harmonics, to give a "darker" sound as the note continues to sound. Nevertheless, despite the clear improvements that could be made to the model, these initial results indicate that the combination of nonnegativity and independence is enough to extract some underlying information about the notes present in the audio signal.

VIII. Discussion

The nonnegative ICA approach outlined in this paper relies heavily on the preservation of the zero values of the data and source signals. If a bias has been added during the data generation process, or the mean has been subtracted during the observation process, this could lead to failure of the algorithm, with either "slack" in the solution (a set of possible rotations all with zero nonnegative reconstruction error), or no rotation able to produce zero reconstruction error. One possible approach to overcome this problem might be to include an adaptive offset in the nonnegative ICA analysis, modifying the algorithm to optimize a weighted sum of the original error J and the norm of the offset required, so as to produce a small error J with the small offset. However, while in theory this would work on noise-free data, analysis of noisy data would require adjustment of the relative weighing between the error and offset and this approach has not yet been tried in practical simulations.

An approach related to the geodesic method used here was suggested by Akuzawa [31] for optimization of kurtosis. He pointed out a duality between this formulation and the dynamics of a quantum lattice, where it is common to ignore N-body interactions for $N > 2$. This led him to suggest a simplified "nested Newton" method which performs a Newton-like search on the separate components of Φ, in effect, using a separate Newton method along each axis of the search space [32].

We note that the update algorithms based on rotation and geodesic methods can be expressed in forms which depend only on the output $\mathbf{Y}$ and consequently are *equivariant* [33]. Also, it is interesting to note that the use of the Frobenius norm of Φ in (23) to measure distance on the manifold, together with the equalities $\delta\mathbf{W} = (\delta\mathbf{B})\mathbf{W}$ and $||\delta\mathbf{B}||_F^2 = 2||\delta\Phi||_F^2$ implies the same squared distance measure expressed in terms of weight $\mathbf{W}$ is

$$\langle \delta\mathbf{W}, \delta\mathbf{W} \rangle_{\mathbf{W}} = 2||\delta\Phi||_F^2 \tag{46}$$

which (apart from a factor of 2) is the form used in the construction of the *natural gradient* [34]. See also [28] for a discussion of the relations between natural gradient, relative gradient and Stiefel manifold learning.

The rotation and geodesic search algorithms are batch algorithms, making one update after each pass through the whole data set. For a large data set (our image separation problem was downsampled from images with $p = 512^2$) this can be considerable. Faster convergence to a nearly optimal solution may be obtained by subsampling the data initially, going over to the final data set for final convergence to an accurate solution.

Our introduction of a "line search" algorithm, or perhaps strictly a "geodesic search" algorithm, leads naturally to the consideration of conjugate gradient methods on this manifold. While this is beyond the scope of the current paper, Edelman *et al.* [27], [35], discuss conjugate gradient algorithms on this constrained space and Martin-Clemente *et al.* [36] have recently used the conjugate gradient approach for ICA.

In this paper, we have developed these rotation and geodesic algorithms specifically for our nonnegative ICA problem. However, there is no reason why they should not also be applied to other ICA methods which use prewhitened data and a differentiable contrast function, provided we can calculate the required steepest gradient descent direction.

IX. Conclusion

We have considered the task of nonnegative ICA, that is, the task of solving the independent component analysis problem $\mathbf{x} = \mathbf{A}\mathbf{s}$ given a sequence $\mathbf{x}$ of observations, with constraints of 1) independence of the sources $\mathbf{s} = (s_1, \ldots, s_n)$ and 2) nonnegativity of the sources $\mathbf{s}_j$. Nonnegativity of $\mathbf{A}$ is not required here.

For sources with nonzero pdf $p(s)$ down to $s = 0$, it is sufficient to find an orthonormal rotation $\mathbf{y} = \mathbf{W}\mathbf{z}$ of the prewhitened observations $\mathbf{z} = \mathbf{V}\mathbf{x}$ which minimizes the mean squared reconstruction error from the rectified version $\mathbf{y}^+$ of $\mathbf{y}$.

We suggested some algorithms which perform this, both based on a simple nonlinear PCA approach and on a geodesic search method driven by differential geometry considerations. We illustrated the operation of these algorithms on an image separation problem and in particular the fast convergence of the rotation and geodesic methods. Finally, we demonstrated the nonnegative ICA approach on the analysis of a musical audio signal, showing that some information about the underlying notes is extracted.

Acknowledgment

The author would like to thank L. Chan and M. Davies for many discussions and suggestions on this work and the comments of three anonymous referees which helped to significantly improve this article. A. Cichocki and W. Kasprzak kindly supplied the images used in Section VI and the music sequence in Fig. 9 is used by permission from the Classical Piano Midi Page http://www.piano-midi.de, copyright Bernd Krueger.

References

[1] P. Comon, "Independent component analysis—A new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[2] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[3] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[4] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, pp. 111–126, 1994.

[5] M. Novak and R. Mammone, "Use of nonnegative matrix factorization for language model adaptation in a lecture transcription task," in *Proc. 2001 IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, Salt Lake City, UT, May 7–11, 2001, pp. 541–544.

[6] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler, "Automatic music transcription and audio source separation," *Cybern. Syst.*, vol. 33, pp. 603–627, Sept. 2002.

[7] J. S. Lee, D. D. Lee, S. Choi, and D. S. Lee, "Application of nonnegative matrix factorization to dynamic positron emission tomography," in *Proc. Int. Conf. Independent Component Anal. Signal Separation*, T.-W. Lee, T.-P Jung, S. Makeig, and T. J. Sejnowski, Eds., San Diego, CA, Dec. 9–13, 2001, pp. 629–632.

[8] L. Xu, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks*, vol. 6, no. 5, pp. 627–648, 1993.

[9] G. F. Harpur and R. W. Prager, "Development of low entropy coding in a recurrent network," *Network: Computation in Neural Systems*, vol. 7, pp. 277–284, 1996.

[10] D. Charles and C. Fyfe, "Modeling multiple-cause structure using rectification constraints," *Network: Computation in Neural Systems*, vol. 9, pp. 167–182, 1998.

[11] G. F. Harpur, "Low Entropy Coding with Unsupervised Neural Networks," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 1997.

[12] P. Paatero, P. K. Hopke, X.-H. Song, and Z. Ramadan, "Understanding and controlling rotations in factor analytic models," *Chemometrics Intell. Lab. Syst.*, vol. 60, no. 1–2, pp. 253–264, 2002.

[13] P. O. Hoyer and A. Hyvärinen, "A multi-layer sparse coding network learns contour coding from natural images," *Vis. Res.*, vol. 42, no. 12, pp. 1593–1605, 2002.
[14] M. D. Plumbley, "Adaptive lateral inhibition for nonnegative ICA," in *Proc. Int. Conf. Independent Component Anal. Signal Separation*, T.-W. Lee, T.-P. Jung, S. Makeig, and T. J. Sejnowski, Eds., San Diego, CA, Dec. 9–13, 2001, pp. 516–521.
[15] ——, "Conditions for nonnegative independent component analysis," *IEEE Signal Processing Lett.*, vol. 9, pp. 177–180, June 2002.
[16] J. Karhunen, E. Oja, L. Wang, R. Vigário, and J. Joutsensalo, "A class of neural networks for independent component analysis," *IEEE Trans. Neural Networks*, vol. 8, pp. 486–504, May 1997.
[17] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math. Anal. Applicat.*, vol. 106, pp. 69–84, 1985.
[18] E. Oja, "Principal components, minor components and linear neural networks," *Neural Networks*, vol. 5, pp. 927–935, 1992.
[19] R. J. Williams, "Feature Discovery Through Error-Correction Learning," Inst. Cognitive Sci., Univ. California, San Diego, ICS Rep. 8501, 1985.
[20] E. Oja, "The nonlinear PCA learning rule in independent component analysis," *Neurocomputing*, vol. 17, no. 1, pp. 25–45, 1997.
[21] M. D. Plumbley and E. Oja, A 'nonnegative PCA' algorithm for independent component analysis, 2002, submitted for publication.
[22] S. Fiori, "'Mechanical' neural learning for blind source separation," *Electron. Lett.*, vol. 35, no. 22, pp. 1963–1964, October 1999.
[23] W. H. Press, S. A. Teukolsky, W. T. Vettering, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[24] S. C. Douglas and S.-Y Kung, "An ordered-rotation kuicnet algorithm for separating arbitrarily-distributed sources," in *Proc. Int. Workshop Independent Component Analysis Blind Signal Separation*, Aussois, France, Jan. 11–15, 1999, pp. 81–86.
[25] S. C. Douglas, "Self-stabilized gradient algorithms for blind source separation with orthogonality constraints," *IEEE Trans. Neural Networks*, vol. 11, pp. 1490–1497, Nov. 2000.
[26] Y. Nishimori, "Learning algorithm for ICA by geodesic flows on orthogonal group," in *Proc. Int. Joint Conf. Neural Networks*, vol. 2, Washington, DC, July 10–16, 1999, pp. 933–938.
[27] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Applicat.*, vol. 20, no. 2, pp. 303–353, 1998.
[28] S. Fiori, "A theory for learning by weight flow on Stiefel-Grassman manifold," *Neural Comput.*, vol. 13, pp. 1625–1647, 2001.
[29] Y. Nishimori, "Multiplicative learning algorithm via geodesic flows," in *Proc. 2001 Int. Symp. Nonlinear Theory Applications*, Miyagi, Japan, 2001, pp. 529–532.
[30] A. Cichocki *et al.*, "Neural network approach to blind separation and enhancement of images," in *Signal Processing VIII: Theories and Applications*, G. Ramponi *et al.*, Eds. Trieste, Italy: EURASIP/LINT Publ., 1996, vol. I, pp. 579–582.
[31] T. Akuzawa, "Multiplicative Newton-like algorithm and independent component analysis," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks*, vol. 4, S.-I. Amari, C. L. Giles, M. Gori, and V. Piuri, Eds., Como, Italy, July 24–27, 2000, pp. 79–82.
[32] ——, "New fast factorization method for multivariate optimization and its realization as ICA algorithm," in *Proc. Int. Conf. Independent Component Analysis Signal Separation*, T.-W. Lee, T.-P. Jung, S. Makeig, and T. J. Sejnowski, Eds., San Diego, CA, Dec. 9–13, 2001, pp. 114–119.
[33] J.-F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 3017–3030, Dec. 1996.
[34] S. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, Feb. 1998.
[35] A. Edelman and S. T. Smith, "On conjugate gradient-like methods for eigen-like problems," *BIT Numer. Math.*, vol. 36, pp. 494–508, 1996.
[36] R. Martin-Clemente, C. G. Puntonet, and J. I. Acha, "Blind signal separation based on the derivatives of the output cumulants and a conjugate gradient algorithm," in *Proc. Int. Conf. Independent Component Analysis Signal Separation*, T.-W. Lee, T.-P. Jung, S. Makeig, and T. J. Sejnowski, Eds., San Diego, CA, Dec. 9–13, 2001, pp. 390–393.

Mark D. Plumbley (S'88–M'90) received the B.A. (now M.A.) degree in electrical sciences from Churchill College, University of Cambridge, Cambridge, U.K., in 1984 the graduate diploma in digital systems design from Brunel University, Brunel, U.K., and the Ph.D. degree in information theory and neural networks from the Department of Engineering, University of Cambridge, in 1991.

He began research on neural networks in 1987, as a Research Student with the Engineering Department, Cambridge University, Cambridge, U.K., continuing as a Research Associate using genetic algorithms to modify neural networks. He joined the Centre for Neural Networks at King's College, London, U.K., in 1991, moving to the Department of Electronic Engineering in 1995. In 2002, he joined the new DSP and Multimedia Group at Queen Mary University of London, and is currently working on independent component analysis (ICA), with particular interest on applications to the analysis and separation of audio and music signals.