

NATURAL GRADIENT APPROACH TO BLIND SEPARATION OF OVER- AND UNDER-COMPLETE MIXTURES

L.-Q. Zhang, S. Amari and A. Cichocki

Brain-style Information Processing Group, BSI
The Institute of Physical and Chemical Research
Saitama 351-0198, Wako shi, JAPAN

ABSTRACT

In this paper we study natural gradient approaches to blind separation of over- and under-complete mixtures. First we introduce Lie group structures on the manifolds of the under- and over-complete mixture matrices respectively, and endow Riemannian metrics on the manifolds based on the property of Lie groups. Then we derive the natural gradients on the manifolds using the isometry of the Riemannian metric. Using the natural gradient, we present a new learning algorithm based on the minimization of mutual information. Finally we apply the natural gradient approach to the state-space model and develop a novel learning algorithm for dynamic component analysis.

1. INTRODUCTION

Recently blind separation of independent sources have become an increasing important research area due to its similarity to the separation feature in human brain, as well as its rapidly growing applications in various fields, such as telecommunication systems, sonar and radar systems, audio and acoustics, image enhancement and biomedical signal processing. Several neural networks and statistical signal processing methods have been developed for blind signal separation. These methods include the Hebbian learning algorithm (Jutten, Herault, 1986 and 1991), the independent component analysis (ICA)(Comon, 1994), robust adaptive algorithm(Cichocki et al. 1994, 1996), nonlinear principal component analysis(Oja, Kahrunen,1995), entropy maximization(Bell and Sejnowski, 1995), equivariant adaptive algorithm and relative gradient(Cardoso and Laheld, 1996) and the natural gradient approach (Amari et al, 1995, 1998).

It has been proven that the natural gradient improves greatly the learning efficiency in blind separation. For special case when the number of sources is equal to the number of sensors, the natural gradient

algorithm has been developed by Amari and independently as relative gradient by Cardoso [7]. In many cases, the number of the source signals is changing over time. Therefore the mixture matrix and demixture matrix are not square and not invertible. For special non-square mixture case, Cardoso and Amari [6] introduced a Lie group and present a novel learning algorithm with uniform performance. Recently Amari [1] extended the natural gradient approach to the over and under-complete cases when the sensor signals are prewhitened.

The main objective of this paper is to extend the idea of natural gradient to the case when the separating matrix is nonsquare, and apply the natural gradient to derive effective learning algorithms to update the separating matrices. First we introduce Lie group structures on the manifolds of the under- and over-complete mixture matrices, and endow Riemannian metrics on the manifolds based on the property of Lie groups. Then we derive the natural gradients on the manifolds using the isometry of the Riemannian metric. Based on the minimization of mutual information, we present a new learning algorithm using the natural gradient on the manifold. Finally we apply the natural gradient approach to the state-space model and develop a novel learning algorithm for dynamic component analysis. It is worthy noting that both under- and over-complete mixtures produce a new ancillary term in natural gradient compared with square matrix mixtures.

2. UNDER- AND OVER-COMPLETE BSS PROBLEM

Assume that the source signals are stationary zero-mean processes and mutually statistically independent. Let $\mathbf{s}(t) = (s_1(t), \dots, s_n(t))$ be an unknown independent source vector and $\mathbf{x}(t) = (x_1(t), \dots, x_m(t))$ a sensor vector, which is linear instantaneous mixture of sources by

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{v}(t), \quad (1)$$

where $\mathbf{A} \in \mathbf{R}^{m \times n}$ is an unknown mixture matrix of full rank, $\mathbf{v}(t)$ is the vector of noises. The blind separation problem is to recover original signals from observations $\mathbf{x}(t)$ without prior knowledge on the source signals and mixture except for independence of the source signals. The demixing model here is a linear transformation of form

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t), \quad (2)$$

where $\mathbf{y}(t) = (y_1(t), \dots, y_n(t))$, $\mathbf{W} \in \mathbf{R}^{n \times m}$ is a demixture matrix to be determined. If $n > m$, i.e. the number of sensor signals is less than the one of source signals, the mixture is called overcomplete. And if $n < m$, the mixture is called undercomplete. The term overcomplete is borrowed from data representation, refer to [8] and [13] for more detail. The general solution to the blind separation is to find a matrix \mathbf{W} such that

$$\mathbf{W}\mathbf{A} = \mathbf{\Lambda}\mathbf{P}, \quad (3)$$

where $\mathbf{\Lambda}$ is a diagonal matrix and \mathbf{P} is a permutation. In the overcomplete case, some components in the diagonal of the matrix $\mathbf{\Lambda}$ could be zero. Denote

$$Gl(n, m) = \{\mathbf{W} \in \mathbf{R}^{n \times m} | \text{rank}(\mathbf{W}) = \min(n, m)\},$$

to the set of $n \times m$ matrices of full rank.

3. LIE GROUP AND NATURAL GRADIENT FOR UNDER-COMPLETE MIXTURE

Assume that $n < m$. For $\mathbf{W} \in Gl(n, m)$, there exists an orthogonal matrix $\mathbf{Q} \in \mathbf{R}^{m \times m}$ such that

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2]\mathbf{Q}, \quad (4)$$

where $\mathbf{W}_1 \in \mathbf{R}^{n \times n}$ is nonsingular. For simplicity, we assume that the orthogonal matrix $\mathbf{Q} = \mathbf{I}$ in the following discussion. Appendix gives the results for general case.

3.1. Lie Group $Gl(n, m)$

The Lie group plays a crucial role in deriving natural gradient of the manifold $Gl(n, n)$, whose element \mathbf{W} is square and nonsingular matrix. Here we introduce the Lie group structure on the manifold $Gl(n, m)$. It is easy to see that $Gl(n, m)$ is a C^∞ manifold of dimension nm . The operations on the manifold $Gl(n, m)$ are define as follows

$$\mathbf{X} * \mathbf{Y} = [\mathbf{X}_1\mathbf{Y}_1, \mathbf{X}_1\mathbf{Y}_2 + \mathbf{X}_2], \quad (5)$$

$$\mathbf{X}^\dagger = [\mathbf{X}_1^{-1}, -\mathbf{X}_1^{-1}\mathbf{X}_2], \quad (6)$$

where $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]$ and $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2]$ are in $Gl(n, m)$, $*$ is the multiplication operator of two matrices \mathbf{X} and

\mathbf{Y} in $Gl(n, m)$ and \dagger is the inverse operator on $Gl(n, m)$. The identity is defined by $\mathbf{E} = [\mathbf{I}_n, \mathbf{0}]$. It is easy to verify that both multiplication and inverse mappings are C^∞ mappings. The inverse operator satisfies the following relation

$$\mathbf{X} * \mathbf{X}^\dagger = \mathbf{X}^\dagger * \mathbf{X} = \mathbf{E} \quad (7)$$

Therefore the manifold $Gl(n, m)$ with the above operations forms a Lie Group.

3.2. Natural Gradient

Lie Groups have a favorite property that they admit an invariant Riemannian metric [5]. Let $T_{\mathbf{W}}$ be the tangent space of $Gl(n, m)$, \mathbf{X} and $\mathbf{Y} \in T_{\mathbf{W}}$ be the tangent vectors. We introduce the inner product with respect to \mathbf{W} as

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}} \quad (8)$$

Since $Gl(n, m)$ is a Lie group, any $\mathbf{Z} \in Gl(n, m)$ defines an onto-mapping: $\mathbf{W} \rightarrow \mathbf{W} * \mathbf{Z}$. The multiplication transformation maps a tangent vector \mathbf{X} at \mathbf{W} to a tangent vector $\mathbf{X} * \mathbf{Z}$ at $\mathbf{W} * \mathbf{Z}$. Therefore we can define a Riemannian metric on $Gl(n, m)$, such that the right multiplication transformation is isometric, that is, it preserves the Riemannian metric. Explicitly we write it as follows

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}} = \langle \mathbf{X} * \mathbf{Z}, \mathbf{Y} * \mathbf{Z} \rangle_{\mathbf{W} * \mathbf{Z}}. \quad (9)$$

If we define the inner product at the identity \mathbf{E} , then $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}}$ is automatically induced by

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}} = \langle \mathbf{X} * \mathbf{W}^\dagger, \mathbf{Y} * \mathbf{W}^\dagger \rangle_{\mathbf{E}}. \quad (10)$$

The inner product at \mathbf{E} is naturally defined by

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{E}} = \text{tr}(\mathbf{X}\mathbf{Y}^T) \quad (11)$$

From definition, for $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2] \in GL(n, m)$ we have

$$\mathbf{W}^\dagger = [\mathbf{W}_1^{-1}, -\mathbf{W}_1^{-1}\mathbf{W}_2], \quad (12)$$

and

$$\mathbf{X} * \mathbf{W}^\dagger = [\mathbf{X}_1\mathbf{W}_1^{-1}, -\mathbf{X}_1\mathbf{W}_1^{-1}\mathbf{W}_2 + \mathbf{X}_2], \quad (13)$$

$$\mathbf{Y} * \mathbf{W}^\dagger = [\mathbf{Y}_1\mathbf{W}_1^{-1}, -\mathbf{Y}_1\mathbf{W}_1^{-1}\mathbf{W}_2 + \mathbf{Y}_2]. \quad (14)$$

For a function $l(\mathbf{W})$ defined on the manifold $Gl(n, m)$, the natural gradient $\tilde{\nabla}l(\mathbf{W})$ is the contravariant form of $\nabla l(\mathbf{W}) = \left(\frac{\partial l(\mathbf{W})}{\partial \mathbf{W}_{ij}} \right)_{n \times m}$, denoting the steepest direction of the function $l(\mathbf{W})$ as measured by the Riemannian metric of $Gl(n, m)$, which is defined by

$$\langle \mathbf{X}, \tilde{\nabla}l(\mathbf{W}) \rangle_{\mathbf{W}} = \langle \mathbf{X}, \nabla l(\mathbf{W}) \rangle_{\mathbf{E}}, \quad (15)$$

for any $\mathbf{X} \in Gl(n, m)$. Using definition (10), and comparing the both side of (15), we have

$$\tilde{\nabla}l(\mathbf{W}) = \nabla l(\mathbf{W})\mathbf{W}^T\mathbf{W} + \nabla l(\mathbf{W})\mathbf{N}_I, \quad (16)$$

where

$$\mathbf{N}_I = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m-n} \end{bmatrix} \in \mathbf{R}^{m \times m}. \quad (17)$$

It is worthy noting that the natural gradient on the manifold $Gl(n, m)$ has an additional term compared with the one on the manifold $Gl(n, n)$. In the undercomplete case, the matrix $\mathbf{W}^T\mathbf{W}$ is singular, while $\mathbf{W}^T\mathbf{W} + \mathbf{N}_I$ is positive definite for any $\mathbf{W} \in Gl(n, m)$. The property ensures that natural gradient descent algorithm keeps the same kind of equilibria of systems as ordinary gradient descent one. When $m = n$, the natural gradient reduces to the one derived in [3].

Remark 1. If the matrix \mathbf{W}_1 is singular for decomposition $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2]$, we should replace the decomposition with (4). In general case the natural gradient induced by the Lie group is given by

$$\tilde{\nabla}l(\mathbf{W}) = \nabla l(\mathbf{W})\mathbf{W}^T\mathbf{W} + \nabla l(\mathbf{W})\mathbf{N}_I, \quad (18)$$

where

$$\mathbf{N}_I = \mathbf{Q}^T \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m-n} \end{bmatrix} \mathbf{Q} \in \mathbf{R}^{m \times m}. \quad (19)$$

See Appendix 7.1 for detail derivation. The result indicates that the natural gradient for under-complete mixture is not unique, which depends on the orthogonal matrix \mathbf{Q} .

4. LEARNING ALGORITHM

Assume that $p(\mathbf{y}, \mathbf{W})$, $p_i(y_i, \mathbf{W})$ are the joint probability density function of \mathbf{y} and marginal pdf of y_i , ($i = 1, \dots, m$) respectively. In order to separate independent sources by demixing model, we formulate the blind deconvolution problem into an optimization problem. Our target is to make the components of \mathbf{y} as mutually independent as possible. To this end, we employ the Kullback-Leibler divergence as a risk function [3]

$$l(\mathbf{W}) = -H(\mathbf{y}, \mathbf{W}) + \sum_{i=1}^n H(y_i, \mathbf{W}), \quad (20)$$

where

$$H(\mathbf{y}, \mathbf{W}) = - \int p(\mathbf{y}, \mathbf{W}) \log p(\mathbf{y}, \mathbf{W}) d\mathbf{y},$$

$$H(y_i, \mathbf{W}) = - \int p_i(y_i, \mathbf{W}) \log p_i(y_i, \mathbf{W}) dy_i.$$

The divergence $l(\mathbf{W})$ is a nonnegative functional, which measures the mutual independence of the output signals $y_i(k)$. The output signals \mathbf{y} are mutually independent if and only if $l(\mathbf{W}) = 0$. We apply the stochastic gradient descent method to obtain a learning algorithm. In order to obtain efficient on-line learning algorithm, we simplify the risk function into the following loss function $l(\mathbf{y}, \mathbf{W})$ for the undercomplete mixture, i.e. $n < m$

$$l(\mathbf{y}, \mathbf{W}) = -\log(|\det(\mathbf{W}\mathbf{E}^T)|) - \sum_{i=1}^n \log p_i(y_i(k), \mathbf{W}), \quad (21)$$

where \mathbf{E} is the identity element of Lie group $Gl(n, m)$, and $\det(\mathbf{W}\mathbf{E}^T)$ is the determinant of matrix $\mathbf{W}\mathbf{E}^T$. In the following discussion, we use the following decomposition

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2], \quad \mathbf{W}_1 \in \mathbf{R}^{n \times n}, \quad (22)$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}, \quad \mathbf{x}_1 \in \mathbf{R}^n, \quad (23)$$

then we have $\mathbf{W}\mathbf{E}^T = \mathbf{W}_1$.

For the gradient of $l(\mathbf{y}, \mathbf{W})$ with respect to \mathbf{W} , we calculate the total differential $dl(\mathbf{y}, \mathbf{W})$ of $l(\mathbf{y}, \mathbf{W})$ when we take a differential $d\mathbf{W}$ on \mathbf{W}

$$dl(\mathbf{y}, \mathbf{W}) = l(\mathbf{y}, \mathbf{W} + d\mathbf{W}) - l(\mathbf{y}, \mathbf{W}). \quad (24)$$

Following the derivation for the natural gradient learning algorithm [3], we have

$$dl(\mathbf{y}, \mathbf{W}) = -tr(d\mathbf{W}_1 \mathbf{W}_1^{-1}) + \varphi^T(\mathbf{y})d\mathbf{y}, \quad (25)$$

where tr is the trace of a matrix and $\varphi(\mathbf{y})$ is a vector of nonlinear activation functions

$$\varphi_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i} = -\frac{p_i'(y_i)}{p_i(y_i)}. \quad (26)$$

From (25), we can easily obtain the standard gradient of $l(\mathbf{y}, \mathbf{W})$ with respect to \mathbf{W} .

$$\frac{dl(\mathbf{y}, \mathbf{W})}{d\mathbf{W}_1} = -\mathbf{W}_1^{-T} + \varphi(\mathbf{y})\mathbf{x}_1^T, \quad (27)$$

$$\frac{dl(\mathbf{y}, \mathbf{W})}{d\mathbf{W}_2} = \varphi(\mathbf{y})\mathbf{x}_2^T. \quad (28)$$

Therefore the natural gradient learning algorithm on $Gl(n, m)$ can be implemented as follows

$$\begin{aligned} \Delta\mathbf{W} &= -\eta \nabla l(\mathbf{W}) (\mathbf{W}^T\mathbf{W} + \mathbf{N}_I) \\ &= \eta ((\mathbf{I} - \varphi(\mathbf{y})\mathbf{y}^T)\mathbf{W} - \varphi(\mathbf{y})\mathbf{x}^T\mathbf{N}_I), \end{aligned} \quad (29)$$

or we write it in another form

$$\Delta\mathbf{W} = \eta [\mathbf{I} - \varphi(\mathbf{y})\mathbf{y}^T, -\varphi(\mathbf{y})\mathbf{x}_2^T] * \mathbf{W}. \quad (30)$$

Remark 2. It is worthy noting that if we let $m = n + 1$ and $x(m) = 1$, (30) reduces to the learning algorithm developed in [6], where Lie group was introduced in the special case. In this case, Cardoso and Amari proved that the learning algorithm (30) is of uniform performance [6].

5. OVERCOMPLETE MIXTURE CASE

For the overcomplete mixture case ($n > m$), we write the demixing model as follows

$$\mathbf{y} = \mathbf{V}\mathbf{x}, \quad \mathbf{V} \in \mathbf{R}^{n \times m}. \quad (31)$$

Suppose that \mathbf{V} is of full rank, i.e. $\text{rank}(\mathbf{V}) = m$. In this case we can also introduce directly a Lie group structure on manifold $Gl(n, m)$, ($n > m$) (See Appendix 7.2). But here we simply derive the natural gradient on $Gl(n, m)$, ($n > m$) from (16). For $\mathbf{V} \in Gl(n, m)$, let $\mathbf{U} = \mathbf{V}^T \in Gl(m, n)$. Using the Lie group structure of $Gl(m, n)$, we have

$$\tilde{\nabla}l(\mathbf{U}) = \nabla l(\mathbf{U})\mathbf{U}^T\mathbf{U} + \nabla l(\mathbf{U})\mathbf{N}_I, \quad (32)$$

For a function $l(\mathbf{V})$ defined on the manifold $Gl(n, m)$, by definition, we have

$$\tilde{\nabla}l(\mathbf{V}) = \tilde{\nabla}l(\mathbf{U})^T, \quad (33)$$

$$\nabla l(\mathbf{V}) = \nabla l(\mathbf{U})^T. \quad (34)$$

Therefore we easily deduce the natural gradient $\nabla l(\mathbf{V})$ on the manifold $Gl(n, m)$ from (32)- (34),

$$\tilde{\nabla}l(\mathbf{V}) = \mathbf{V}\mathbf{V}^T\nabla l(\mathbf{V}) + \mathbf{N}_I\nabla l(\mathbf{V}). \quad (35)$$

As soon as we obtain the natural gradient on manifold $GL(n, m)$ in overcomplete case, we can develop a learning algorithm to learn demixture matrix \mathbf{V} . Here we define the loss function in the overcomplete case as follows

$$l(\mathbf{y}, \mathbf{V}) = -\log(|\det(\mathbf{V}_e)|) - \sum_{i=1}^n \log p_i(y_i(k), \mathbf{W}), \quad (36)$$

where \mathbf{V}_e is a $n \times n$ matrix extended from \mathbf{V} to a square matrix, such that \mathbf{V}_e is nonsingular. If we use the following decomposition

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}, \quad \mathbf{y}_1, \mathbf{V}_1 \in \mathbf{R}^{m \times m}, \quad (37)$$

and \mathbf{V}_1 is nonsingular, we define \mathbf{V}_e as

$$\mathbf{V}_e = \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} \\ \mathbf{V}_2 & \mathbf{I}_{n-m} \end{bmatrix} \in \mathbf{R}^{n \times n}. \quad (38)$$

Following the same procedure as deriving (27) and (28), we obtain the ordinary gradient of $l(\mathbf{y}, \mathbf{V})$ with respect to \mathbf{V}

$$\frac{dl(\mathbf{y}, \mathbf{V})}{d\mathbf{V}_1} = \mathbf{V}_1^{-T} + \varphi_1(\mathbf{y}_1)\mathbf{x}^T, \quad (39)$$

$$\frac{dl(\mathbf{y}, \mathbf{V})}{d\mathbf{V}_2} = \varphi_2(\mathbf{y}_2)\mathbf{x}^T, \quad (40)$$

where $\varphi(\mathbf{y})$ is the vector of activation functions defined in (26), and $\varphi(\mathbf{y}) = \begin{bmatrix} \varphi_1(\mathbf{y}_1) \\ \varphi_2(\mathbf{y}_2) \end{bmatrix}$. Therefore the natural gradient learning algorithm on $Gl(n, m)$ can be implemented as follows

$$\begin{aligned} \Delta\mathbf{V} &= -\eta \left(\mathbf{V}\mathbf{V}^T + \mathbf{N}_I \right) \nabla l(\mathbf{y}, \mathbf{V}) \\ &= \eta\mathbf{V} \left(\mathbf{I}_m - \mathbf{V}^T\varphi(\mathbf{y})\mathbf{x}^T \right) - \eta\mathbf{N}_I\varphi(\mathbf{y})\mathbf{x}^T. \end{aligned} \quad (41)$$

It is easily seen that the equilibria of learning algorithm (41) include

$$E(\varphi(\mathbf{y}_1)\mathbf{y}_1^T) - \mathbf{I}_m = \mathbf{0}, \quad (42)$$

$$E(\varphi(\mathbf{y}_2)\mathbf{y}_2^T) = \mathbf{0}. \quad (43)$$

It is inferred that the learning algorithm can decompose m signals in such a way that they are as independent as possible if the activation functions are suitably chosen.

Remark 3. It should be noted that Lewicki and Sejnowski [13] proposed a learning algorithm to estimate the mixture matrix \mathbf{A} in following form

$$\Delta\mathbf{A} = \mathbf{A}\mathbf{A}^T\nabla \log p(\mathbf{y}|\mathbf{A}), \quad (44)$$

which can be considered as a special form of (41), when we take $\mathbf{N}_I = \mathbf{0}$ and $l(\mathbf{y}, \mathbf{W}) = -\log p(\mathbf{y}|\mathbf{A})$. Therefore natural gradient (35) gives an insight into the learning algorithm (44) from a geometrical point of view.

6. STATE-SPACE MODELS

The state-space models provides a new approach to the blind separation and deconvolution [15], which is easy to exploit common features of systems that may be present in the real dynamic systems. It is natural to extend mixture models to nonlinear ones by using nonlinear state-space models. Here we assume that mixture model is described a linear state-space system

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}}\bar{\mathbf{x}}(k) + \bar{\mathbf{B}}\mathbf{s}(k) + \bar{\mathbf{P}}\boldsymbol{\xi}(k), \quad (45)$$

$$\mathbf{u}(k) = \bar{\mathbf{C}}\bar{\mathbf{x}}(k) + \bar{\mathbf{D}}\mathbf{s}(k), \quad (46)$$

$\mathbf{s}(k) \in \mathbf{R}^n$ is a vector of source signals with mutually components, $\mathbf{u}(k) \in \mathbf{R}^m$ ($m \geq n$) is the available vector of sensor signals, and $\bar{\mathbf{x}} \in \mathbf{R}^r$ is the vector of state.

And $\xi \in \mathbf{R}^n$ is the process noises. If the process noises are negligible small, the transfer function matrix of the linear system (45) and (46) is a $m \times n$ matrix of the form

$$\mathbf{H}(z) = \overline{\mathbf{C}}(z\mathbf{I} - \overline{\mathbf{A}})^{-1}\overline{\mathbf{B}} + \overline{\mathbf{D}}, \quad (47)$$

where z^{-1} is a delay operator.

The blind separation problem is formulated as recovering source signals $\mathbf{s}(k)$ from mixture $\mathbf{u}(k)$ in some sense (accepting unavoidable indeterminacy, such as arbitrary scaling, permutation and delays of original sources), without knowing parameters of matrices $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, $\overline{\mathbf{C}}$ and $\overline{\mathbf{D}}$ in the model, but only some statistic features on source signals.

Since the mixture model is supposed to be a linear state-space system, we employ another linear state-space system with similar structure as a demixture model, which is described as follows,

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}\xi_R(k), \quad (48)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k), \quad (49)$$

where the input $\mathbf{u}(k) \in \mathbf{R}^m$ of the demixing model is just the output (sensor signals) of the mixing model, $\mathbf{y}(k) \in \mathbf{R}^n$ is the output of the demixture system to recover source signals, and $\xi_R(k)$ is the reference model noises. The matrix set $\mathbf{W} = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ are parameters to be determined in learning. For simplicity we assume that the reference model noise is zero, i.e. $\xi_R(k) = 0$.

The transfer function of the demixing model is

$$\mathbf{W}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \quad (50)$$

The output $\mathbf{y}(k)$ is designed to recover the source signals in the following sense

$$\mathbf{y}(k) = \mathbf{W}(z)\mathbf{H}(z)\mathbf{s}(k) = \mathbf{P}\mathbf{\Lambda}(z)\mathbf{s}(k), \quad (51)$$

where \mathbf{P} is a permutation matrix and $\mathbf{\Lambda}(z)$ is a diagonal matrix with $\lambda_i z^{-\tau_i}$ in diagonal entry (i,i), here λ_i is a nonzero constant and τ_i is any nonnegative integer. Refer to [15] for more details.

Let $\mathbf{W} = \{\mathbf{C}, \mathbf{D}\}$. Assume that the matrices \mathbf{A}, \mathbf{B} are known or obtained by off-line training. In this case the demixture model can simply write into the following form

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k). \quad (52)$$

If we consider \mathbf{x} as a component of mixed signals, the demixture (52) becomes under-complete case. Using the following decomposition and notation

$$\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2], \quad \mathbf{D}_1 \in \mathbf{R}^{n \times n}, \quad (53)$$

$$\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2], \quad \mathbf{u}_1 \in \mathbf{R}^n, \quad (54)$$

$$\mathbf{C}_1 = [\mathbf{D}_2, \mathbf{C}], \quad \mathbf{x}_1(k) = [\mathbf{u}_2^T, \mathbf{x}(k)^T]^T, \quad (55)$$

we rewrite (52) into following form

$$\mathbf{y}(k) = \mathbf{C}_1\mathbf{x}_1(k) + \mathbf{D}_1\mathbf{u}_1(k). \quad (56)$$

Therefore we can apply the natural gradient to derive an efficient learning algorithm to update \mathbf{C}_1 and \mathbf{D}_1 . If the loss function $l(\mathbf{y}, \mathbf{W})$ is chosen as follows

$$l(\mathbf{y}, \mathbf{W}) = -\log(|\det(\mathbf{D}_1)|) - \sum_{i=1}^n \log p_i(y_i(k), \mathbf{W}), \quad (57)$$

where $\det(\mathbf{D}_1)$ is the determinant of matrix \mathbf{D}_1 , we can easily obtain the partial derivatives of $l(\mathbf{y}, \mathbf{W})$ with respect to matrices \mathbf{C}_1 and \mathbf{D}_1

$$\frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{C}_1} = \varphi(\mathbf{y}(k))\mathbf{x}_1^T(k), \quad (58)$$

$$\frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{D}_1} = \varphi(\mathbf{y}(k))\mathbf{u}_1^T(k) - \mathbf{D}_1^{-T}. \quad (59)$$

Using natural gradient (16) we derive a novel learning algorithm for updating \mathbf{C}_1 and \mathbf{D}_1

$$\Delta \mathbf{D}_1 = \eta(\mathbf{I} - \varphi(\mathbf{y})\mathbf{y}^T)\mathbf{D}_1, \quad (60)$$

$$\Delta \mathbf{C}_1 = \eta(\mathbf{I} - \varphi(\mathbf{y})\mathbf{y}^T)\mathbf{C}_1 - \eta\varphi(\mathbf{y})\mathbf{x}_1^T. \quad (61)$$

Remark 4. It should be pointed out that the demixture (52) is not real instantaneous case. We could use other state estimator, such as Kalman Filter, to update state vector $\mathbf{x}(k)$ and employ (60) and (61) to update matrices \mathbf{D}_1 and \mathbf{C}_1 .

7. APPENDIX

7.1. Natural Gradient for General Case

Assume that $n < m$. For $\mathbf{W} \in Gl(n, m)$, there exists an orthogonal matrix $\mathbf{Q} \in \mathbf{R}^{m \times m}$ such that

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2]\mathbf{Q}, \quad (62)$$

where $\mathbf{W}_1 \in \mathbf{R}^{n \times n}$ is nonsingular. We define two operations for the Lie group as follows,

$$\mathbf{X} * \mathbf{Y} = [\mathbf{X}_1\mathbf{Y}_1, \mathbf{X}_1\mathbf{Y}_2 + \mathbf{X}_2]\mathbf{Q}, \quad (63)$$

$$\mathbf{X}^\dagger = [\mathbf{X}_1^{-1}, -\mathbf{X}_1^{-1}\mathbf{X}_2]\mathbf{Q}, \quad (64)$$

where $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]\mathbf{Q}$ and $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2]\mathbf{Q}$ are in $Gl(n, m)$, $*$ is the multiplication operator of two matrices in $Gl(n, m)$ and \dagger is the inverse operator on $Gl(n, m)$. The identity is defined by $\mathbf{E} = [\mathbf{I}_n, \mathbf{0}]\mathbf{Q}$.

Lie Groups have a favorite property that they admit an invariant Riemannian metric. Let $T_{\mathbf{W}}$ be the tangent space of $Gl(n, m)$, \mathbf{X} and $\mathbf{Y} \in T_{\mathbf{W}}$ be the tangent vectors. The Riemannian metric can easily be induced by following inner product

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}} = \langle \mathbf{X} * \mathbf{W}^\dagger, \mathbf{Y} * \mathbf{W}^\dagger \rangle_{\mathbf{E}}. \quad (65)$$

The inner product at \mathbf{E} is naturally defined by

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{E}} = \text{tr}(\mathbf{X}\mathbf{Y}^T). \quad (66)$$

For a function $l(\mathbf{W})$ defined on the manifold $Gl(n, m)$, the natural gradient $\tilde{\nabla}l(\mathbf{W})$ is the contravariant form of $\nabla l(\mathbf{W})$ denoting the steepest direction of the function $l(\mathbf{W})$ as measured by the Riemannian metric of $Gl(n, m)$, which is defined by

$$\langle \mathbf{X}, \tilde{\nabla}l(\mathbf{W}) \rangle_{\mathbf{W}} = \langle \mathbf{X}, \nabla l(\mathbf{W}) \rangle_{\mathbf{E}}, \quad (67)$$

for any $\mathbf{X} \in Gl(n, m)$. Using definition (65), and comparing the both side of (67), we have

$$\tilde{\nabla}l(\mathbf{W}) = \nabla l(\mathbf{W})\mathbf{W}^T\mathbf{W} + \nabla l(\mathbf{W})\mathbf{N}_I, \quad (68)$$

where

$$\mathbf{N}_I = \mathbf{Q}^T \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m-n} \end{pmatrix} \mathbf{Q} \in \mathbf{R}^{m \times m}. \quad (69)$$

7.2. Lie Group and Riemannian metric for Overcomplete Mixtures

Using decomposition (37), we define Lie group operations as follows

$$\mathbf{X} * \mathbf{Y} = \begin{bmatrix} \mathbf{X}_1\mathbf{Y}_1 \\ \mathbf{X}_2\mathbf{Y}_1 + \mathbf{Y}_2 \end{bmatrix}, \quad \mathbf{X}^\dagger = \begin{bmatrix} \mathbf{X}_1^{-1} \\ -\mathbf{X}_2\mathbf{X}_1^{-1} \end{bmatrix}. \quad (70)$$

The identity in the group is $\mathbf{E} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}$. The Riemannian metric at \mathbf{W} is implemented by

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}} = \langle \mathbf{W}^\dagger * \mathbf{X}, \mathbf{W}^\dagger * \mathbf{Y} \rangle_{\mathbf{E}}. \quad (71)$$

It is easy to verify that the natural gradient in the Riemannian metric is given by (35).

8. REFERENCES

- [1] S. Amari. Natural gradient for over- and undercomplete bases in ica. *Submitted to Neural Computation*, 1998.
- [2] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.
- [3] S. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind signal separation. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 8 (NIPS'95)*, pages 757–763, Cambridge, MA, 1996. The MIT Press.
- [4] A.J. Bell and T.J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
- [5] W. M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry*. Academic Press, Inc., 1986.
- [6] J. Cardoso and S. Amari. Maximum likelihood source separation: Equivalence and adaptivity. In *Proceedings of SYSID,97*, pages 1063–1068, 1997.
- [7] J.F. Cardoso and B. Laheld. Equivariant adaptive source separation. *IEEE Trans. Signal Processing*, 44(12):3017–3030, December 1996.
- [8] S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. Technical report, Technical Report, Dept. Stat., Stanford Univ., Stanford, CA, 1886.
- [9] A. Cichocki and R. Unbehauen. Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Trans Circuits and Systems I: Fundamentals Theory and Applications*, 43:894–906, 1996.
- [10] A. Cichocki, R. Unbehauen, and E. Rummert. Robust learning algorithm for blind separation of signals. *Electronics Letters*, 30(17):1386–1387, 1994.
- [11] P. Comon. Independent component analysis – a new concept? *Signal Processing*, 36(3):287–314, 1994.
- [12] C. Jutten and J. Herault. Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.
- [13] M. S. Lewicki and T. Sejnowski. Learning nonlinear covercomplete representations for efficient coding. In *NIPS,10*, 1998.
- [14] E. Oja and J. Karhunen. Signal separation by nonlinear hebbian learning. In M. Palaniswami, Y. Attkiouzel, R. Marks II, D. Fogel, and T. Fukuda, editors, *Computational Intelligence - A Dynamic System Perspective*, pages 83–97, New York, NY, 1995. IEEE Press.
- [15] L. Zhang and A. Cichocki. Blind deconvolution/equalization using state-space models. In *Proceeding of NNSP'98*, page in printing, 1998.
- [16] L. Zhang and A. Cichocki. Blind separation/deconvolution using canonical stable state-space models. In *Proceeding of NOLTA '98*, page in printing, 1998.