Blind Signal Separation with Kernel Probability Density Estimation Based on MMI Criterion Optimized by Conjugate Gradient

Wei Li
Anhui Key Laboratory of Electric Drive and Control, Anhui Polytechnic University, Wuhu 241000, PR China
Email: liweilwlcw@sina.com

***Abstract*—**This paper presents a novel blind signal separation (BSS) approach based on the theory of independent component analysis. In the proposed BSS approach, the learning rule is derived by the conjugate gradient optimization algorithm rather than the ordinary gradient and natural gradient algorithm based on the minimum mutual information (MMI) criterion. The score function is a key point in solving the BSS problem. Instead of choosing nonlinear activity functions empirically, a kernel probability density function estimation method is used in order to estimate the probability density functions and their derivatives of the separated signals. Thus the score function is then estimated directly. The proposed BSS approach is applied to separate the mixtures of sub-Gaussian and super-Gaussian source signals simultaneously. Computer simulations are provided to demonstrate the superior learning performance of the proposed BSS approach.

***Index Terms*—**Blind signal separation, minimum mutual information criterion, probability density estimate, conjugate gradient optimization algorithm

I. Introduction

Blind signal separation (BSS) has received considerable attentions by a number of researchers due to its various potential applications in many scientific fields of multi-dimensional signal processing. The general idea to solve the BSS problem is to exploit some statistical assumptions of the source signals, such as non-Gaussianity, mutual independence, sparsity, spatio-temporal decorrelation, smoothness and linear predictability etc., to identify the true sources or to find a representation of the observed signals with physical meaning [1], [2].

In this paper, it is supposed that the source signals are mutually independent and are mixed linearly and instantaneously, the static mixing model of the BSS problem can be formulated as follows:

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

where $\mathbf{x}$ is the N-dimensional observed sensor signals, $\mathbf{s}$ is the underlying N-dimensional source signals, $\mathbf{A}$ denotes the unknown mixing matrix with $N \times N$ entries and $\mathbf{n}$ represents the additive sensor noise. As the name implies, the purpose of the BSS problem is to reveal the unobservable source signals from the observed possibly noisy sensor signals without any a priori information about the mixing process [3]. In other words, a $N \times N$ dimensional separating matrix $\mathbf{W}$ is to be estimated and applied to the sensor signals $\mathbf{x}$ to obtain the separated signals

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2}$$

such that $\mathbf{y}$ are the estimation of the source signals, i.e., $\mathbf{y} = \mathbf{P}\mathbf{D}\mathbf{s}$ holds for a permutation matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{D}$. The mixing/separating system of BSS is shown in Fig. 1.

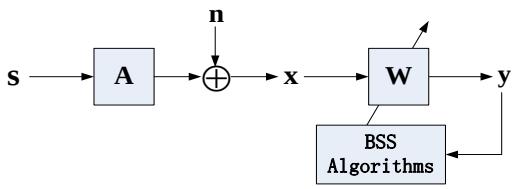

Fig. 1 The block diagram of the BSS mixing/separating system

Up to now, a number of approaches with different behavior have been proposed for solving the BSS problem [4]-[6]. The most widely used method for BSS problem is the so-called independent component analysis (ICA). ICA was first proposed by Comon, who presented the definition of ICA issue as well as its resolution framework, and applied it to BSS [7]. Since then, The BSS problem was transformed to that of finding the appropriate cost functions of some statistics and the proper algorithms for optimization.

In the ICA-type BSS algorithms, mutual information (MI) is the most popular measurement of the statistical dependence among data and signals. For the linear BSS (LBSS) problems, Comon *et al*. [7] and Yang *et al*. [8] have used the Edgeworth expansion and the truncated Gram-Charlier expansion to estimate the MI of the separated signals respectively. Hild II *et al*. [9] employed Renyi's entropy with nonparametric probability density function estimation instead of Kullback-Leibler divergence to approximate the MI. Luo and Lu [10] combined the MI criterion with the Newton's method to realize the separation on the Stiefel manifold. Mokhtari *et al*. [11] separated the bilinear mixtures exploiting a recurrent network adjusted by the MI criterion. Pham

Manuscript received March 24, 2014; revised July 28, 2014.
Corresponding author: liweilwlcw@sina.com
doi:10.12720/jcm.9.7.579-587.

derived a Gaussian mutual information cost function for separating the instantaneous mixtures [12], [13], and then generalized it to the blind signal deconvolution cases [14], [15]. Rhabi *et al*. also considered the convolutive mixing case with MI [16].

In addition to LBSS, Yang [17] extended the MMI to solve the nonlinear BSS (NLBSS) problem. Taleb and Jutten [18] and Achard *et al*. [19] applied the MMI to a specific nonlinear case, namely post-nonlinear mixture model. Almeida [20], [21] proposed another efficient nonlinear BSS method, namely MISEP, which used a neural network as the separating system based on MI.

As to the optimization algorithm, the natural gradient algorithm (NGA) has been proven to be efficient for training the separating matrix [22]-[25], which outperforms the ordinary gradient descent algorithm. Still, there exists a tough task for selecting the nonlinear activity function empirically [26], [27].

In this paper, a novel adaptive linear BSS approach based on the MMI criterion is developed. In the proposed approach, the MI is used as the cost function for deriving the adaptive learning rules, and the conjugate gradient searching algorithm is exploited to find the optimal separating matrix. The score function in the adaptive learning rules is not replaced by certain nonlinear activity functions empirically. Instead, a kernel density estimation method is used to estimate the score function directly. Some computer simulations are conducted to confirm the performance of the proposed conjugate gradient based BSS algorithm over that of the ordinary gradient algorithm and the natural gradient one.

The rest of this paper is organized as follows: The MMI cost function and its gradient are presented in Section 2. In Section 3, the conjugate gradient optimization algorithm is introduced and applied to search the optimal separating matrix in the BSS problem. The score function is estimated through a kernel density estimation method in Section 4. Some simulations for validating the proposed BSS method are presented in Section 5. Finally, Section 6 concludes the paper.

II. MMI Cost Function and the Learning Algorithms

In this section, the MMI cost function is presented, and the corresponding ordinary gradient and natural gradient are derived respectively.

A. The MMI Cost Function for BSS

The basic idea of MMI is to minimize the statistical dependence among the components of the output signals $\mathbf{y}$ of the separating system. The dependence can be measured by the Kullback-Leibler divergence $KL(p(\mathbf{y})|\prod_{i=1}^{N} p_i(y_i))$ between the joint probability density function (PDF) $p(\mathbf{y})$ of $\mathbf{y}$ and the product of the marginal PDF $p_i(y_i)$ of y_i:

$$KL\left(p(\mathbf{y})\left|\prod_{i=1}^{N} p_i(y_i)\right.\right)=\int p(\mathbf{y})\ln\frac{p(\mathbf{y})}{\prod_{i=1}^{N} p_i(y_i)}d\mathbf{y} \quad (3)$$

Therefore, the MI $I(\mathbf{y},\mathbf{W})$ can be written in terms of the differential entropy:

$$I(\mathbf{y},\mathbf{W})=\sum_{i=1}^{N} H(y_i)-H(\mathbf{y}) \quad (4)$$

where $H(\mathbf{y})=-E\{\ln p(\mathbf{y})\}=-\int p(\mathbf{y})\ln p(\mathbf{y})d\mathbf{y}$ and $H(y_i)=-E\{\ln p_i(y_i)\}=-\int p_i(y_i)\ln p_i(y_i)dy_i$ are the joint and the marginal entropy respectively.

From formula (2), we have

$$H(\mathbf{y})=H(\mathbf{x})+\ln\left|\det(\mathbf{W})\right| \quad (5)$$

where $|\bullet|$ is an operator of the absolute value, det(•) denotes a determinant of a matrix. Since the entropy $H(\mathbf{x})$ does not depend on the separating matrix $\mathbf{W}$, bring the above formula in to (4) except for the first term in the right hand side of (5), then the MI cost function can be rewritten as

$$I(\mathbf{y},\mathbf{W})=\sum_{i=1}^{N} H(y_i)-\ln\left|\det(\mathbf{W})\right| \quad (6)$$

B. The Ordinary Gradient and the Natural Gradient

In order to find the optimal parameter of the separating system, the BSS problem is equivalent to the following optimization problem

$$\mathbf{W}=\arg\min_{\mathbf{W}} I(\mathbf{y},\mathbf{W}) \quad (7)$$

For realizing the unsupervised adaptive training of the separating matrix $\mathbf{W}$, the gradient of the cost function $I(\mathbf{y},\mathbf{W})$ with respect to $\mathbf{W}$ is calculated subsequently.

The gradient of the first term in the right hand side of formula (6) with respect to arbitrary (i,j) element w_{ij} is computed as

$$\begin{aligned}\nabla_{w_{ij}}\sum_{i=1}^{N} H(y_i)&=-\frac{\partial\sum_{i=1}^{N} E\{\ln p_i(y_i)\}}{\partial y_i}\frac{\partial y_i}{\partial w_{ij}}\\&=E\{\phi_i(y_i)x_j\}\end{aligned} \quad (8)$$

where $\nabla_u v$ denotes the gradient of v with respect to u, and

$$\phi_i(y_i)=-\frac{\partial E\{\ln p_i(y_i)\}}{\partial y_i}=-\frac{p'(y_i)}{p(y_i)} \quad (9)$$

is called the score function of y_i. Moreover, since the determinant $\det(\mathbf{W})$ can be expanded as a summation of the product of its entries w_{ij} and the corresponding algebraic cofactor F_{ij} for any row i in all columns $j=1,2,\cdots,N$, i.e, $\det(\mathbf{W})=\sum_{i=1}^{N} w_{ij}F_{ij}$. According the definition of the inverse of a matrix,

$\mathbf{W}^{-1} = \mathbf{W}_{adj} / \det(\mathbf{W})$, where $\mathbf{W}_{adj} = \left(F_{ij}\right)_{N\times N}$ is the adjoint matrix of $\mathbf{W}$. Hence, for each entry w_{ij}, we have

$$\nabla_{w_{ij}} \ln\left|\det(\mathbf{W})\right| = \frac{F_{ij}}{\det(\mathbf{W})} = \left(\left(\mathbf{W}^{-1}\right)^{T}\right)_{ij} = \left(\mathbf{W}^{-1}\right)_{ji} \quad (10)$$

where $(\bullet)_{ij}$ denotes the *i*th row-*j*th column component of a matrix.

Combining (8) and (9) together, we obtain the component-wise gradient as

$$\nabla_{w_{ij}} I(\mathbf{y},\mathbf{W}) = E\left\{\phi_i(y_i)x_j\right\} - \left(\mathbf{W}^{-1}\right)_{ji} \quad (11)$$

And its matrix form is

$$\nabla_{\mathbf{W}} I(\mathbf{y},\mathbf{W}) = E\left\{\mathbf{\Phi}(\mathbf{y})\mathbf{x}^{T}\right\} - \left(\mathbf{W}^{\mathbf{T}}\right)^{-1} \quad (12)$$

The corresponding stochastic gradient algorithm is

$$\Delta\mathbf{W} = -\mu\nabla_{\mathbf{W}} I(\mathbf{y},\mathbf{W}) = \mu\left(\left(\mathbf{W}^{\mathbf{T}}\right)^{-1} - \mathbf{\Phi}(\mathbf{y})\mathbf{x}^{T}\right) \quad (13)$$

where μ is the step size bounded in the area from 0 to 1, and $\mathbf{\Phi}(\mathbf{y}) = \left[\phi_1(y_1), \phi_1(y_1), \cdots, \phi_N(y_N)\right]^{T}$ is the vector of the score functions. Generally, in the implementation of the algorithm, the score functions are often chosen to be some nonlinear activity functions heuristically.

Because the update parameter is a matrix, the solution space is Riemannian space. Amari [22] introduces the natural gradient

$$\begin{aligned}\nabla_{\mathbf{W}}^{nat} I(\mathbf{y},\mathbf{W}) &= \nabla_{\mathbf{W}} I(\mathbf{y},\mathbf{W})\mathbf{W}^{T}\mathbf{W} \\ &= \left(E\left\{\mathbf{\Phi}(\mathbf{y})\mathbf{y}^{T}\right\} - \mathbf{I}\right)\mathbf{W}\end{aligned} \quad (14)$$

to replace the ordinary gradient $\nabla_{\mathbf{W}} I(\mathbf{y},\mathbf{W})$, and points out that the natural gradient in Riemannian space indicates the steepest descending direction just like the ordinary gradient in Euclidian space. The corresponding natural gradient algorithm is

$$\Delta\mathbf{W} = -\mu\nabla_{\mathbf{W}}^{nat} I(\mathbf{y},\mathbf{W}) = \mu\left(\mathbf{I} - \mathbf{\Phi}(\mathbf{y})\mathbf{y}^{T}\right)\mathbf{W} \quad (15)$$

where $\mathbf{I}$ is an identity matrix. It can be observed that the natural gradient algorithm does not need to compute the inverse of a matrix not like the ordinary gradient in formula (13). Consequently, it is supposed that the natural gradient algorithm has greater stability and faster convergence speed. The separating matrix and the estimation signals can be calculated as

$$\begin{cases}\mathbf{W}_k = \mathbf{W}_{k-1} + \Delta\mathbf{W}_k \\ \mathbf{y}_k = \mathbf{W}_k\mathbf{x}\end{cases} \quad (16)$$

where k is the number of iteration steps.

III. The Conjugate Gradient Based BSS Algorithm

The conjugate gradient searching algorithm has been validated to be an efficient learning algorithm with applications in the neural network field for training the weights between layers [28]. In this section, the proposed conjugate gradient based BSS algorithm, short for CGBSS, is deduced. The algorithm updates the separating matrix $\mathbf{W}$ along the conjugate gradient directions alternatively. For the convenience to the derivation of the algorithm, some preliminaries are first elucidated.

A. Some Preliminaries

Before elaborating the CGBSS algorithm, a useful definition is given first.

Definition 1. [28] Suppose that there exists a space of $n\times n$ dimensional matrices. When two arbitrary points of the space, i.e., two matrices $\mathbf{A}$ and $\mathbf{B}$, are not very far from each other, the shortest trajectory from $\mathbf{A}$ to $\mathbf{B}$ is defined as the geodesic denoted by $\mathbf{G}(\xi)$, which can be formulated by

$$\mathbf{G}(\xi) = \exp\left(\xi\mathbf{T}_{\mathbf{A}}\mathbf{A}^{-1}\right)\mathbf{A} \quad (17)$$

where the parameter ξ is bounded in the interval $[0,1]$ such that $\mathbf{G}(0) = \mathbf{A}$ and $\mathbf{G}(1) = \mathbf{B}$, and $\mathbf{T}_{\mathbf{A}} = \mathbf{G}'(0)$ is a tangent vector at the point $\mathbf{A}$, which indicates the direction of the geodesic.

From the above definition of the geodesic, the following proposition is presented.

Proposition 1. [28] Suppose that $\mathbf{A}$ and $\mathbf{B}$ are two elements within a matrix space, and $\mathbf{T}_{\mathbf{A}}$ is the tangent vector at the point $\mathbf{A}$ along the trajectory of the geodesic, similarly, $\mathbf{T}'_{\mathbf{B}}$ is the tangent vector at the point $\mathbf{B}$. Then, when moving the tangent vector from $\mathbf{A}$ to $\mathbf{B}$, the relationship between $\mathbf{T}_{\mathbf{A}}$ and $\mathbf{T}'_{\mathbf{B}}$ can be expressed as

$$\mathbf{T}'_{\mathbf{B}} = \mathbf{T}_{\mathbf{A}}\mathbf{A}^{-1}\mathbf{B} \quad (18)$$

B. The Conjugate Gradient Algorithm for Training the Separating Matrix

Combining the preliminaries in the previous subsection and the cost function (6), the conjugate gradient method is detailed presented in the following.

The conjugate gradient searching method mainly contains two key procedures in each iteration: 1) Calculate the tangent vector of the current solution point, which is conjugate to the former searching direction, thus the next searching direction is determined; 2) Solve a one-dimensional optimization problem to find the new iterative point based on the newly formed trajectory of geodesic.

At the beginning of the algorithm, the separating matrix is initialized as $\mathbf{W} = \mathbf{W}_0$, and then the gradient $\nabla I(\mathbf{y},\mathbf{W}_0)$ of the cost function (6) is computed out. Note, $\nabla I(\mathbf{y},\mathbf{W}_0)$ may be chosen as either the ordinary gradient

$\nabla_{\mathbf{W}} I(\mathbf{y}, \mathbf{W}_0)$ in (12) or the natural gradient $\nabla_{\mathbf{W}}^{nat} I(\mathbf{y}, \mathbf{W}_0)$ in (14).

For the kth iteration in the algorithm, denote $\mathbf{W}_k$ the k th searching result, $\mathbf{T}_{\mathbf{W}_k}$ the current searching direction at the point $\mathbf{W}_k$, which is calculated by $\mathbf{T}'_{\mathbf{W}_k} = \mathbf{T}_{\mathbf{W}_{k-1}} \mathbf{W}_{k-1}^{-1} \mathbf{W}_k$. The new searching direction, which is conjugate to former searching direction $\mathbf{T}_{\mathbf{W}_k}$, should be first determined. In order to find the conjugate searching direction, the gradient $\nabla I(\mathbf{y}, \mathbf{W}_k)$ at the point $\mathbf{W}_k$ is computed. The new searching direction is then given by the tangent vector

$$\mathbf{T}_{\mathbf{W}_k} = \nabla I(\mathbf{y}, \mathbf{W}_k) + \lambda_k \mathbf{T}'_{\mathbf{W}_k} \tag{19}$$

where λ_k is selected such that the new searching direction is conjugate to the former one satisfying that the Hessian matrix $\mathbf{H}(\mathbf{T}_{\mathbf{W}_k}, \mathbf{T}_{\mathbf{W}_{k-1}}) = \mathbf{0}$. In the practical implementation, to save the computational load of the algorithm, the Hessian matrix is replaced by the finite difference approximation. Therefore, the parameter λ_{k+1} can be computed in the following form

$$\lambda_k \approx \frac{tr\left\{\left(\nabla I(\mathbf{y}, \mathbf{W}_k) - \nabla I(\mathbf{y}, \mathbf{W}_{k-1})\right) \nabla I(\mathbf{y}, \mathbf{W}_k)^T\right\}}{tr\left\{\nabla I(\mathbf{y}, \mathbf{W}_{k-1}) \nabla I(\mathbf{y}, \mathbf{W}_{k-1})^T\right\}} \tag{20}$$

where the finite difference approximation reduces the computational load greatly. After the new searching direction $\mathbf{T}_{\mathbf{W}_k}$ is determined, the next iterative point $\mathbf{W}_{k+1}$ can be obtained by solving the following one-dimensional optimization problem along the geodesic

$$\mathbf{W}_{k+1} = \arg \min_{\xi} I(\mathbf{y}, \mathbf{G}_{\mathbf{W}_k}(\xi)) \tag{21}$$

where the geodesic is given by $\mathbf{G}_{\mathbf{W}_k}(\xi) = \exp(\xi \mathbf{T}_{\mathbf{W}_k} \mathbf{W}_k^{-1}) \mathbf{W}_k$, as defined in the previous subsection.

IV. The Kernel Density Estimation to Estimate the Score Function

It has been proved that the score function $\mathbf{\Phi}(\mathbf{y})$ in formula (13) and (15) can be replaced by some nonlinear activity functions, when the statistics of the source signals satisfies some special conditions. For any stochastic signal x, define $K(x) = E\{x^4\} - 3E^2\{x^2\}$ as the kurtosis, $k(x) = E\{x^4\} / E^2\{x^2\} - 3$ as the normalized kurtosis. If $k(x) > 0$, x is called super-Gaussian signal, and the nonlinear activity function can be chosen to be $\phi_{\sup}(x) = \alpha x + \tanh(\beta x)$, $\alpha \geq 0, \beta \geq 2$; while If $k(x) < 0$, x is called sub-Gaussian signal, and the nonlinear function can be chosen to be $\phi_{\text{sub}}(x) = \alpha x + x|x^2|$, $\alpha \geq 0$. From this fact, we can obtain three conclusions: 1) Although the PDF of signals are unknown thus the SF can not be computed out, the BSS problem can still be solved by selecting the non-linear functions correctly; 2) When the statistical property of the sources can not be distinguished, there is no basis for selecting the non-linear functions, as a result, the BSS can not be realized successfully; 3) when super-Gaussian signal and sub-Gaussian signal exist simultaneously in the sources, only one nonlinear activity function can not work effectively for the blind separation task.

When the statistical property of the signal is unknown, an intuitional method is to estimate the kurtosis on-line. According to the estimated kurtosis, the nonlinear activity function is then selected properly. This method can solve the BSS problem to a certain degree, but it needs to calculate the kurtosis repeatedly, which leads to heavy computational load as well as the bad robustness.

In this paper, instead of choosing nonlinear functions empirically and estimating the kurtosis on-line, an adaptive score function estimation approach is proposed. In the proposed approach, a kernel density estimation method is used to estimate the PDFs of signals and their derivatives directly. Thus, form formula (9), the score functions $\mathbf{\Phi}(\mathbf{y})$ can also be estimated directly.

It is assumed that there are T realizations $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$ of the observation signals $\mathbf{x}$. According to the equation (2), T realizations $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T$ of the output signals $\mathbf{y}$ can also be obtained. Therefore, the kernel density estimator [29,30] $\hat{p}_{i,h}(y_i)$, used to estimate the true marginal PDF $p_i(y_i)$, is given by

$$\hat{p}_{i,h}(y_i) = \frac{1}{T} \sum_{t=1}^{T} K_h(y_i - y_{i,t}) \tag{22}$$

where $K_h(u) = \frac{1}{h} K\left(\frac{u}{h}\right)$, $K(\bullet)$ is a kernel function, and h is the bandwidth. The kernel function usually has the following properties

$$\int_{-\infty}^{+\infty} K(u)\, du = 1 \tag{23}$$

$$\int_{-\infty}^{+\infty} u^j K(u)\, du \begin{cases} = 0, & \text{for } j = 1, \cdots, k-1 \\ > 0, & \text{for } j = k \end{cases} \tag{24}$$

Note that the property (23) makes $\hat{p}_{i,h}(y_i)$ become a density function, i.e., $\int_{-\infty}^{+\infty} \hat{p}_{i,h}(y_i)\, dy_i = 1$. One of the most popular kernels is the Gaussian kernel density function $K_G(v) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{v^2}{2}\right)$. It has been proven that the choice of the bandwidth h is much more important than the choice of $K(\bullet)$ for the estimation accuracy of the

kernel density estimator $\hat{p}_{i,h}(y_i)$ [31], [32]. Small values of h give rise to spurious features, whereas a very large value of h leads to an estimate that is too smooth to reveal structural features of the true density $p_i(y_i)$.

To evaluate the behavior of the kernel density estimator $\hat{p}_{i,h}(y_i)$, it is necessary to choose an appropriate distance measure between the true density $p_i(y_i)$ and the estimator $\hat{p}_{i,h}(y_i)$. One of the commonly used measure is the asymptotic mean integrated squared error [31], defined by

$$M(h)=\frac{\sigma_K^4 h^4 R\left(p^{(2)}{}_i(y_i)\right)}{4}+\frac{R(K(y_i))}{Th} \tag{25}$$

where $R(K(y_i))=\int K^2(y_i)dy_i$ and $\sigma_K^2=\int y_i^2 K(y_i)dy_i$. The measure $M(h)$ can be minimized explicitly to obtain a proper asymptotic representation for minimizer as

$$h_{\min}=T^{-\frac{1}{5}}\left\{\frac{R(K(y_i))}{\sigma_K^4 R\left(p^{(2)}{}_i(y_i)\right)}\right\}^{\frac{1}{5}} \tag{26}$$

To acquire the best bandwidth, the unknown true density function $p_i(y_i)$ in (26) is replaced by an estimated density function $\hat{p}_{i,h'}(y_i)$, where h' is an initial bandwidth used to estimate $R\left(p^{(2)}{}_i(y_i)\right)$. Note, the bandwidth h' should be chosen to be distinct from h, due to the selection of h' has much less influence on the performance of the kernel density estimator $\hat{p}_{i,h}(y_i)$ than that of h.

A plug-in method from [33] is exploited to select the proper bandwidth denoted by $\bar{h}$ based on equation (26) to ensure that the bandwidth $\bar{h}$ satisfies

$$\bar{h}=T^{-\frac{1}{5}}\left\{\frac{R(K)}{\sigma_K^4 R\left(\hat{p}_i^{(2)}\left(g_i(\bar{h})\right)\right)}\right\}^{\frac{1}{5}} \tag{27}$$

where $g_i(\bar{h})=C(K(y_i))D(p_i(y_i))\bar{h}^{\frac{5}{7}}$ for some appropriate $C(K(y_i))$ and $D(p_i(y_i))$. $C(K(y_i))$ can be just simply replaced by an proper constant. Usually, $C(K(y_i))$ is selected as $\left(6\sqrt{2}\right)^{\frac{1}{7}}$. $D(p_i(y_i))$ is a function of $R\left(p^{(2)}{}_i(y_i)\right)\Big/R\left(p^{(3)}{}_i(y_i)\right)$ and can also be estimated by the plug-in method. Substituting the bandwidth h in (22) with the solution value of $\bar{h}$ in (27), the kernel density estimator $\hat{p}_{i,\bar{h}}(y_i)$ is obtained. Therefore, the score function can be estimated as $\hat{\phi}_i(y_i)=-\frac{\hat{p}'_{i,\bar{h}}(y_i)}{\hat{p}_{i,\bar{h}}(y_i)}$, where $\hat{p}'_{i,\bar{h}}(y_i)$ the first-order derivative of $\hat{p}_{i,\bar{h}}(y_i)$.

V. Simulation Results

To show the performance of the proposed BSS algorithm, we perform some simulations in this section.

A. Validity of the Algorithm for Separating Sub-Gaussian and Super-Gaussian Sources

In this subsection, five signals with 6000 sample points, which are derived from the benchmark signals in the MATLAB toolbox ICALAB developed by Cichocki *et al* [34], are used as the source signals, shown in Fig. 2(a), to test the proposed BSS algorithm. The sources contain two sub-Gaussian artificial signals, two super-Gaussian speech signals and one Gaussian noise signal, whose kurtosis are separately -1.4550, -1.5441, 3.5126, 5.3750 and 0.0087.

The source signals are mixed by random generated matrix, whose elements are drawn from a standardized Gaussian distribution. Hence, the mixing matrix **A** is created with 5 rows and 5 columns. The mixture signals, shown in Fig. 2(b), are generated from formula (1). From Fig. 2(b), it can be observed that the sources have been highly mixed, thus we can not acquire any information about the sources from the mixtures.

To validate the superior performance of the proposed conjugate gradient based BSS algorithm (CGBSS) over that of the ordinary stochastic gradient BSS algorithm (SGBSS) in (13) and the natural gradient BSS algorithm (NGBSS) in (15), we perform all these algorithms separately on the generated mixture signals. Note, in the CGBSS algorithm, the gradient direction can be chosen to be either the ordinary gradient or the natural gradient. We denote these two cases as SG-CGBSS and NG-CGBSS. Hence, four types of BSS algorithms are available here. Additionally, to evaluate the affect of the score function, we implement the SG-CGBSS and the NG-CGBSS algorithms using the proposed score function kernel density estimation method, the SGBSS algorithm with the nonlinear activity functions $\phi_{\text{sub}}(x)$ and the NGBSS algorithm with $\phi_{\text{sup}}(x)$ presented in Section 4 for comparison. At the beginning, the initial separating matrix for each algorithm is chosen as the identity matrix $\mathbf{W}_0=\mathbf{I}$; the learning rate $\mu=0.01$. The algorithms are performed adaptively and refresh the parameters of separating system every time a new mixture sample arrives. After finishing the learning, the separated signals of four algorithms are obtained and shown in Fig. 2(c~f).

From Fig. 2(c) and Fig. 2(d), we find that the algorithms with only one specific nonlinear activity function can not achieve the separation task. However, from Fig. 2(e) and Fig. 2(f), it can be intuitionally seen that the waveform of the source signals has been

recovered apart from different permutation and scaling. Upon this fact, it is proven that the proposed BSS algorithm with kernel density function estimation can separate the mixtures with sub-Gaussian and super-Gaussian signals simultaneous successfully, which overcomes the deficiency of the algorithms using some specific nonlinear activity functions. Compare Fig. 2(e) and Fig. 2(f) with Fig. 2(a) separately, it can be observed that the recovered signals from the NG-CGBSS algorithm are more similar to the true source signals than those from the SG-CGBSS algorithm. Therefore, the learning algorithm with the natural gradient is more efficient than that with the ordinary gradient in Riemannian space.

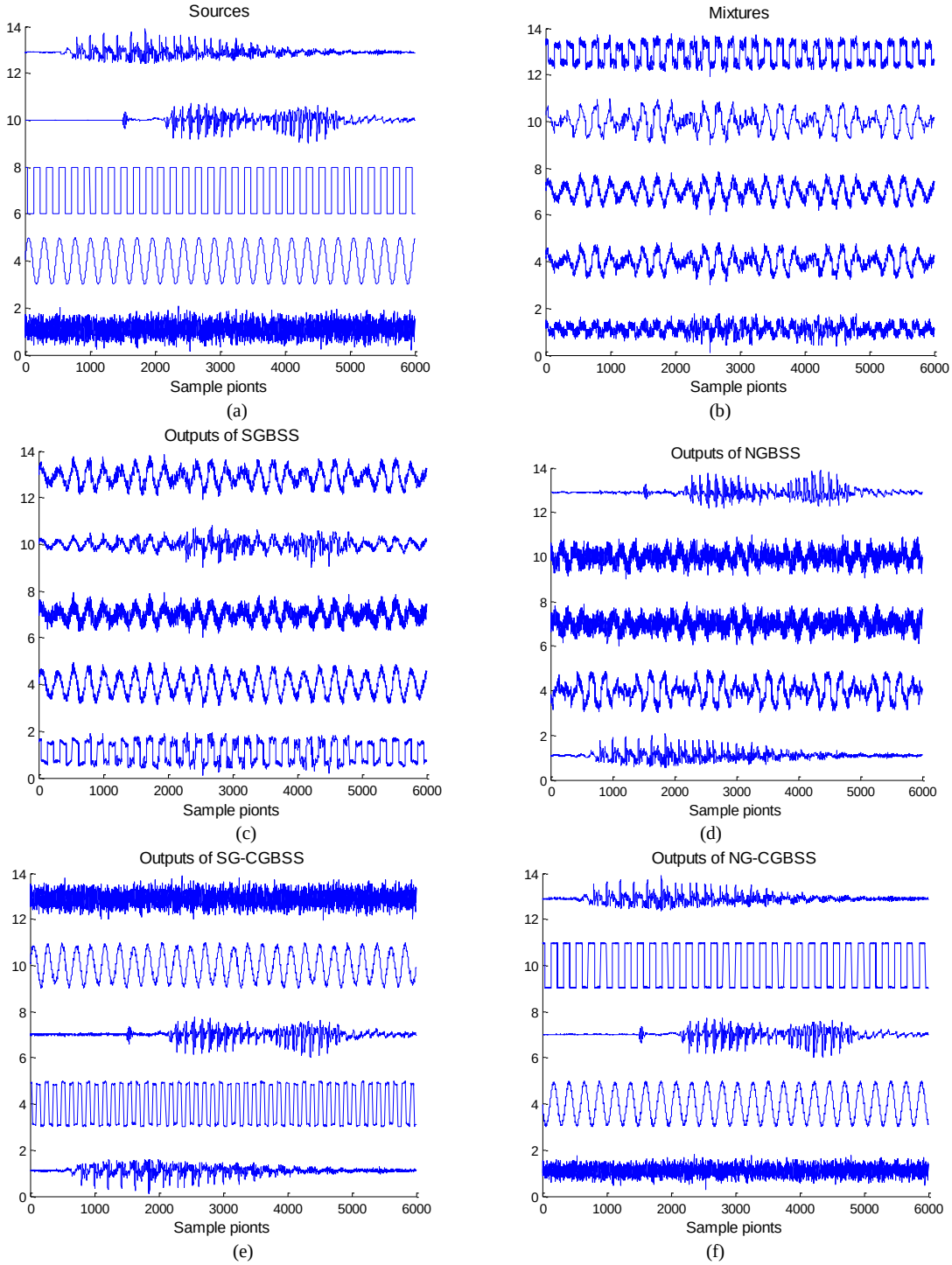

Fig. 2. The waveform of the source signals, the mixture signals and the separated signals obtained from four different BSS algorithms (a. the source signals; b. the mixture signals; c. the output signals of the SGBSS algorithm; d. the output signals of the NGBSS algorithm; e. the output signals of the SG-CGBSS algorithm; f. the output signals of the NG-CGBSS algorithm)

B. Quantitative Comparison of the Behavior of the BSS Algorithms

For the sake of quantitatively evaluating the performance of the proposed BSS algorithm, we conduct several simulations in this subsection. First of all, a classical performance index of the linear BSS problem is defined as

$$PI = \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{|c_{ij}|}{\max_k |c_{ik}|}-1\right)+\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{|c_{ij}|}{\max_k |c_{kj}|}-1\right) \quad (28)$$

where c_{ij} is the element of the globe system matrix $\mathbf{C}=\mathbf{WA}$ in the i th row and j th column. It should be pointed out that for the linear BSS problem, the smaller the value of the performance index PI is, the better the result of the separation algorithm is. While we can observed that, from formula (28), when only one element in each row or in each column of the globe system matrix $\mathbf{C}$ is in the dominant position, and the others are zeros, then PI arrives the minimum value.

We then separately apply the four BSS algorithms, SGBSS, NGBSS, SG-CGBSS and NG-CGBSS, to process the same mixture signals generated in the previous simulation. All the four algorithms are preformed for 100 times independently. The performance index PI of the algorithms are calculated for each run and averaged for all 100 runs. The averaged PI with respect to the number of iteration are drawn in Fig. 3.

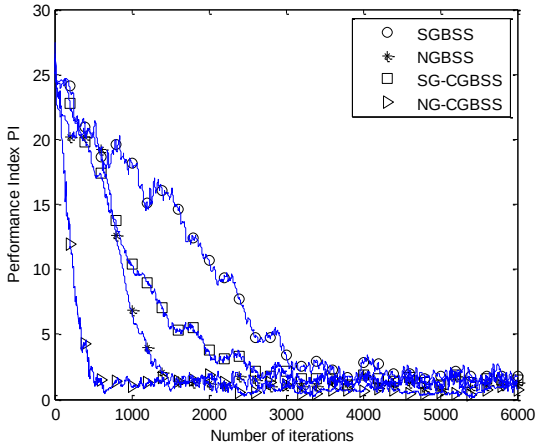

Fig. 3. The learning curves of the averaged performance index PI of the four BSS algorithm with respect to the iteration number

It should be noted that in the implementation of the SGBSS algorithm and SG-CGBSS algorithm, the results may be trapped in the local optimum, leading to the failure of the separation task. Therefore, the values of the PI are averaged only with the successful runs. Nevertheless, this situation does not exist in the other two algorithms. From Fig. 3, we observe that the conjugate gradient based BSS algorithms, SG-CGBSS and NG-CGBSS, have faster convergence speed than the traditional algorithms. The ordinary gradient algorithm has the slowest convergence speed. The reason is possibly that the parameter space of the BSS problem is the Riemannian space, while in the Riemannian space, the steepest decent direction is not the direction of the ordinary gradient any more but that of the natural gradient. Moreover, the conjugate gradient based BSS algorithms have smaller values of the index PI than the other algorithms. This means that the conjugate gradient based BSS algorithms can obtain a more accurate estimation of the sources and the inverse of the mixing matrix than the others.

For further performance comparison and investigating the influence of the mixing matrix, we conduct the following simulations. 1) Four previously mentioned algorithms are implemented, and then the values of the index PI and the signal to interference ratio (SIR, in dB) for each algorithm are recorded. The SIR is given by

$$SIR = \frac{1}{N}\sum_{i=1}^{N}10\log_{10}\frac{E[s_i^2]}{E[(y_i-s_i)^2]} \quad (29)$$

where y_i is the repermuted and normalized signal corresponding to the i th source signal, and N is the number of the sources. 2) We set a threshold value of the PI, denoted by $PI_{\min}$. Then the algorithms are performed again. When the values of the PI converges to $PI_{\min}$, the convergence steps and the time consumption of the algorithms are recorded. We set the threshold $PI_{\min}=0.5$. The above simulations are conducted for 100 independent runs and the average values are calculated under two mixing situations (refer to Tab. 1). The condition numbers of the two mixing matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are $cond(\mathbf{A}_1)=17.365$ and $cond(\mathbf{A}_2)=9.844\times10^3$ separately. Hence, $\mathbf{A}_1$ is regarded as a non-singular matrix, while $\mathbf{A}_2$ is nearly singular.

TABLE I. THE COMPARISON OF THE PERFORMANCE OF FOUR ALGORITHMS UNDER TWO MIXING CONDITIONS

	Mixing Matrix $\mathbf{A}_1$				Mixing Matrix $\mathbf{A}_2$			
	SIR (dB)	PI	Conv. steps	Time cons. (seconds)	SIR (dB)	PI	Conv. steps	Time cons. (seconds)
SGBSS	19.48	0.1558	2595	0.6521	6.71	9.3442	-	-
NGBSS	26.25	0.1063	1610	0.5414	17.55	0.3910	2441	0.7630
SG-CGBSS	24.63	0.1284	1987	0.7143	8.73	7.2105	-	-
NG-CGBSS	31.45	0.0735	1038	0.5606	22.18	0.2644	1456	0.7922

From Table. I, it can be seen that the conjugate gradient modified algorithms have more accurate estimation in terms of the final PI and the SIR and less convergence steps than the original algorithms. Still, the

time consumption has not increased too much. We also find that the ordinary gradient type algorithms spend more time for training than the natural gradient type algorithms. This is possibly because the ordinary gradient algorithm needs to compute the inverse of the separating matrix $\mathbf{W}^{-1}$ in each update step, which results in heavy computational load, while the natural gradient algorithm does not need to do this. Compare the separation results under the two mixing cases, it can be found that although the performance of all the four algorithms degenerates as the mixing condition become worse, the conjugate gradient based BSS algorithms obtain better results than the others. Moreover, it should be noted that the ordinary gradient based algorithms become invalid under the bad mixing condition.

VI. Conclusions

In this paper, we have presented a novel ICA based BSS method. The method combines the minimum mutual information principle and the conjugate gradient optimizing algorithm together to derive the adaptive learning algorithms for updating the separating matrix. In the proposed BSS algorithm, the parameter of the separating system updates along the conjugate gradient directions alternatively, which is the combination of the geodesic and the gradient (ordinary gradient or natural gradient) of the current solution point. The computational load of the algorithm is reduced obviously, when the calculation of Hessian matrix is replaced by finite difference approximation.

Instead of choosing nonlinear activity functions empirically, the kernel density estimation method is employed in order to estimate the probability density function and its derivatives of the output signals directly. Consequently, the score function of the outputs can also be estimated directly.

The simulations have shown the superior performance of the proposed conjugate gradient based BSS algorithm. The proposed algorithm has faster convergence speed and more accurate estimation of the source signals than the ordinary gradient algorithm and the natural gradient algorithm. Moreover, the proposed BSS algorithm can separate the mixtures of sub-Gaussian and super-Gaussian source signals simultaneously.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61273070, Doctor Candidate Foundation of Jiangnan University under Grant No. 1252050205135130.

References

[1] S. Choi, A. Cichocki, H. M. Park, and S. Y. Lee, "Blind source separation and independent component analysis: A review," *Neural Information Processing*, vol. 6, pp. 1–57, 2005.

[2] M. E. Farfoura, S. J. Horng, and X. Wang, "A novel blind reversible method for watermarking relational databases," *Journal of the Chinese Institute of Engineers*, vol. 36, pp. 87–97, 2013.

[3] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*, John Wiley, New York, 2003.

[4] J. Wen, S. Zhang, and J. Yang, "A fast algorithm for undedetermined mixing matrix identification based on mixture of guassian (MOG) sources model," *Journal of Software*, vol. 9, pp. 184–189, 2014.

[5] S. Sun, C. Peng, W. Hou, J. Zheng, Y. Jiang, and X. Zheng, "Blind source separation with time series variational Bayes expectation maximization algorithm," *Digital Signal Processing*, vol. 22 pp. 17–33, 2012.

[6] H. Zhang, L. Li, and W. Li, "Independent vector analysis for convolutive blind noncircular source separation," *Signal Processing*, vol. 92, pp. 2275–2283, 2012.

[7] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, Salt Lake City, USA, 2010.

[8] H. H. Yang and S. Amari, "Adaptive on-line learning algorithms of blind separation—maximum entropy and minimum mutual information," *Neural Computation*, vol. 9, pp. 1457–1482, 1997.

[9] K. E. Hild II, D. Erdogmus, and J. Principe, "Blind source separation using Renyi's mutual information," *IEEE Signal Processing Letters*, vol. 8, pp. 174–176, 2001.

[10] Z. Q. Luo and J. Lu, "On blind source separation using mutual information criterion," *Mathematical Programming*, vol. 97, pp. 587–603, 2003.

[11] F. Mokhtari, M. Babaie-Zadeh, and C. Jutten, "Blind separation of bilinear mixtures using mutual information minimization," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing*, MLSP 2009, pp. 1–6, 2009.

[12] D. T. Pham, "Blind separation of instantaneous mixture of sources via the Gaussian mutual information criterion," *Signal Processing*, vol. 81, pp. 855–870, 2001.

[13] D. T. Pham, "Fast algorithms for mutual information based independent component analysis," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2690–2700, 2004.

[14] D. T. Pham, "Mutual information approach to blind separation of stationary sources," *IEEE Transactions on Information Theory*, vol. 48, pp. 1935–1946, 2002.

[15] D. T. Pham, "Mutual information approach to blind separation-deconvolution," in *Proc. 13th European Signal Processing Conference*, pp. 684–687, 2007.

[16] M. E. Rhabi, G. Gelle, H. Fenniri, and G. Delaunay, "A penalized mutual information criterion for blind separation of convolutive mixtures," *Signal Processing*, vol. 84, pp. 1979–1984, 2004.

[17] H. H. Yang, S. Amari, and A. Cichocki, "Information theoretic approach to blind separation of sources in non-linear mixture," *Signal Processing*, vol. 64, pp. 291–300, 1998.

[18] A. Taleb and C. Jutten, "Source separation in post-nonlinear mixtures," *IEEE Transactions on Signal Processing*, vol. 47, pp. 2807–2820, 1999.

[19] S. Achard, D. T. Pham, and C. Jutten, "Criteria based on mutual information minimization for blind source separation in post nonlinear mixtures," *Signal Processing*, vol. 85, pp. 965–974, 2005.

[20] L. B. Almeida, "MISEP-linear and nonlinear ICA based on mutual information," *Journal of Machine Learning Research*, vol. 4, pp. 1297–1318, 2003.

[21] L. B. Almeida, "Linear and nonlinear ICA based on mutual information-the MISEP method," *Signal Processing*, vol. 84, pp. 231–245, 2004.

[22] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, 1998.

[23] S. Choi, A. Cichocki, L. L. Zhang, and S. Amari, "Approximate maximum likelihood source separation using the natural gradient," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E86-A, pp. 198–205, 2003.
[24] S. Choi, S. Amari, and A. Cichocki, "Natural gradient learning for spatio-temporal decorrelation: Recurrent network," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E83-A, pp. 2175–2722, 2000.
[25] E. Celledoni and S. Fiori, "Neural learning by geometric integration of reduced `rigid-body' equations," *Journal of Computational and Applied Mathematics*, vol. 172, pp. 247–269, 2004,
[26] S. Choi, A. Cichocki, and S. Amari, "Flexible independent component analysis," *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 26, pp. 25–38, 2000.
[27] S. Fiori, "Hybrid independent component analysis by adaptive LUT activation function neurons," *Neural Networks*, vol. 15, pp. 85–94, 2002.
[28] Y. Nishimori, S. Akaho, and M. D. Plumbley, "Natural conjugate gradient on complex flag manifolds for complex independent subspace," in *Proc. 18th International Conference on Artificial Neural Networks, Lecture Notes in Computer Science*, 2008, pp. 165–174.
[29] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 302–309.
[30] J. Karvanen and V. Koivunen, "Blind separation methods based on Pearson system and its extensions," *Signal Processing*, vol. 82, pp. 663–673, 2002.
[31] M. C. Jones, J. S. Marron, and S. J. Sheather, "Progress in data-based bandwidth selection for kernel density estimation," *Computational Statistics*, vol. 11, pp. 337–381, 1996.
[32] S. Fiori, "Probability density function learning by unsupervised neurons," *International Journal of Neural Systems*, vol. 11, pp. 399–417, 2001.
[33] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.
[34] A. Cichocki, S. Amari, K. Siwek, and T. Tanaka. ICALAB Toolboxes. [Online]. Available: http://www.bsp.brain.riken.jp/ICALAB>.

Wei Li was born in Anhui Province, China, in 1985. He received the M.S. degree in detection technology and automatic equipment from Anhui Polytechnic University in 2010, and the Ph.D degree in control theory and control engineering from Jiangnan University in 2014. He is currently a lecturer of Anhui Polytechnic University. His research interest covers data analysis and signal processing.

