Second-Order Blind Signal Separation for Convolutional Mixtures Using Conjugate Gradient

Hai Huyen Dam, Antonio Cantoni, Sven Nordholm, and Kok Lay Teo

Abstract—This letter presents a new computational procedure for the second-order gradient-based blind signal separation (BSS) problem with convolutional mixtures that has improved convergence characteristics over the steepest descent algorithm. The BSS problem is formulated as a constrained optimization problem with complex unmixing weight matrices where the constraints are formulated to overcome the permutation effects. This problem is then transformed into an unconstrained optimization problem, so that the conjugate gradient algorithm can be applied. The convergence of the proposed procedure is compared with the steepest descent algorithms in real and simulated environments.

Index Terms—Blind signal separation, conjugate gradient, convolutional mixtures, decorrelation, non-stationary, unmixing.

I. INTRODUCTION

BLIND SIGNAL SEPARATION (BSS) is an area of great interest due to its ability to separate sources from the observed mixtures without requiring a priori knowledge of the location of sources or the geometry of sensor array. Such flexibility has made BSS a popular technique in many practical applications.

BSS techniques can be broadly classified into two classes, namely, higher-order-based BSS [1] and second-order-based BSS [2]–[4]. Higher-order-based BSS requires assumption about the source’s density function. Second-order-based BSS, on the other hand, requires assumptions about the source second-order statistics such as non-stationary or non-whiteness. Here, we consider the second-order gradient-based BSS problem for convolutional mixtures and the approach described in [2] for obtaining the unmixing matrices.

In [5], a steepest descent algorithm with projected gradient in combination with a step size search procedure is employed for the second-order gradient-based BSS. The algorithm improves the convergence characteristics compared to the fixed step size gradient-based BSS [2], [4]. In other words, it results in lower computational complexity for convergence by incorporating a step size search for each iteration.

In this letter, we develop a new computational procedure for the second-order gradient-based BSS to improve further the convergence over the steepest descent algorithm [5] without any computational complexity penalty. The key contribution of this letter is the approach to achieve the transformation of the constrained optimization problem with complex unmixing matrices into an unconstrained optimization problem so that the conjugate gradient algorithm [6]–[8] can be applied. The convergence and the complexity in terms of the number of real multiplications of the new procedure is compared with the steepest descent algorithm.

This letter is organized as follows. The second-order BSS problem and the constrained optimization problem are presented in Section II. The equivalent unconstrained optimization problem is obtained in Section III. A computational procedure for the optimization problem is developed in Section IV. Simulation results are presented in Section V, and finally, conclusions are given in Section VI.

II. SECOND-ORDER BLIND SIGNAL SEPARATION AND OPTIMIZATION PROBLEM

Consider the model used in [2] for a convolutional mixture of N sources in which the received signal vector at L microphones is

$$x(n) = h(n) \ast s(n) + v(n)$$

(1)

where $n$ is the sampled time index at the time instant $t = nT$, and $T$ denotes the sampling period. The notations $h(n), s(n)$, and $v(n)$ represent the mixing matrix, the sources, and the noise components, respectively, while $\ast$ denotes the convolution operation.

The received signal is passed through $Q$ unmixing matrices $w(q), 0 \leq q \leq Q - 1$, with each of size $N \times L$ to produce an output signal vector $y(n)$. The problem becomes to estimate the unmixing matrices so as to recover the sources (up to an arbitrary scaling and permutation). The exact number of sources is usually unknown and generally assumed to be the same as the number of microphones, i.e., $N = L$. Consequently, the unmixing matrix $w(q)$ is assumed to have the dimension $L \times L$.

The BSS method exploits the non-stationarity of signals, and thus, the received signal is processed in blocks with the processing carried out in the frequency domain. As such, the problem is to estimate the frequency domain unmixing matrix $W(k), 0 \leq k \leq K - 1$, of size $L \times L$ that decorrelates the cross-power spectrum matrices $R_x(k, m)$ for $0 \leq m \leq M - 1$, where $M$ is the number of blocks and $K$ is the number of DFT.
points [2]. We investigate the case where $K$ is even. The case where $K$ is odd can be developed similarly.

For $0 \leq l_1, l_2 \leq L - 1$, the coefficient vector

$$[W_{l_1,l_2}(0), \ldots, W_{l_1,l_2}(K - 1)]^T$$

(2)

does the source of the time domain unmixing vector $w_{l_1,l_2}$ from the $l_1$th input to the $l_2$th output. Note that $W_{l_1,l_2}(k)$, $0 \leq k \leq K - 1$, denotes the $(l_1, l_2)$th element of the matrix $W$. From the DFT property of a real sequence, we have:

- $W_{l_1,l_2}(0)$ and $W_{l_1,l_2}(K/2)$ are real-valued;
- $W_{l_1,l_2}(k) = W_{l_1,l_2}^*(K - k)$ for all $1 \leq k \leq K/2 - 1$, where $^*$ denotes the complex conjugate.

Thus, the problem becomes to estimate $K/2 + 1$ unmixing matrices $W(0), \ldots, W(K/2)$. This problem can be formulated as minimizing the following weighted cost function:

$$f(W(0), \ldots, W(K/2)) = \sum_{k=0}^{K/2} \sum_{m=0}^{M-1} \|E(k, m)\|_F^2$$

(3)

where $E(k, m)$ is a diagonal matrix error

$$E(k, m) = W(k) R_x(k, m) W^H(k) - \Lambda_k(m, m).$$

(4)

$\| \cdot \|_F^2$ is the Frobenius norm and $H$ denotes the Hermitian operation. The matrix $\Lambda_k(m, m)$ is a diagonal matrix with elements equal to $R_x(k, m) W^H(k)$ along the diagonal. The weighting function $\gamma(k)$ is obtained from the correlation matrices as in [5].

To avoid the degenerate solution of matrices with zero entries, it is necessary to include additional constraints on the elements of the unmixing matrices. Here, we restrict the diagonal elements of the matrix $W$ to be one for all $k$. For convenience, denote by $L \subset \mathbb{N}^2$

$$L = \{(l_1, l_2) : 0 \leq l_1, l_2 \leq L - 1 ; l_1, l_2 \in \mathbb{N} ; l_1 \neq l_2\}.$$

The minimization of the weighted cost function in (3) can lead to an arbitrary permutation of the frequency bins. A method to overcome this problem is to constrain the corresponding time domain unmixing weight $w_{l_1,l_2}$, $\forall (l_1, l_2) \in L$, to be of length $D$, $D \ll K$ [2]. Denote by $G$ a $(K - D) \times K$ matrix consisting of the $K - D$ last rows of the IDFT matrix, i.e.,

$$G = \begin{bmatrix} 1 & \cdots & e^{2\pi i D(K - 1)/K} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & e^{2\pi i (K - 1)(K - 1)/K} \end{bmatrix}. \quad (5)$$

Then, the constraint on the time domain unmixing vector can be given in terms of the vector in (2) as

$$G[W_{l_1,l_2}(0) \ldots W_{l_1,l_2}(K - 1)]^T = 0_{K-D \times 1}. \quad (6)$$

The problem is to minimize (3) subject to (6) and constraints on the diagonal of the unmixing matrices. This problem is equivalently represented in terms of a set of $L(L - 1)$ real unknown vectors $v_{l_1,l_2}$, $(l_1, l_2) \in L$

$$v_{l_1,l_2} = [\Re \{W_{l_1,l_2}(0)\}, \ldots, \Re \{W_{l_1,l_2}(K/2)\}]$$

$$\Im \{W_{l_1,l_2}(1)\}, \ldots, \Im \{W_{l_1,l_2}(K/2 - 1)\}]^T \quad (7)$$

where $\Re \{\cdot\}$ and $\Im \{\cdot\}$ denote, respectively, the real and imaginary parts of $\{\cdot\}$. The cost function in (3) is expressed in terms of a real cost function $g$ of $L(L - 1)$ real vectors $v_{l_1,l_2}$, $(l_1, l_2) \in L$.

$$g(v_{l_1,l_2}, (l_1, l_2) \in L) = f(W(0), \ldots, W(K/2)). \quad (8)$$

The constraint (6) is reformulated in terms of $v_{l_1,l_2}$ as

$$Q v_{l_1,l_2} = 0_{K-D \times 1}, \quad \forall (l_1, l_2) \in L.$$ 

Following simplifications, the $(K - D) \times K$ real matrix $Q$ can be given by

$$Q = [Q^1 \quad Q^2]^T \quad (9)$$

where $Q^1$ is a $(K - D) \times (K/2 + 1)$ matrix with its $(l_1, l_2)$th element, $0 \leq l_1 \leq K - D - 1, 0 \leq l_2 \leq K/2$, given by

$$Q^1_{l_1,l_2} = \begin{cases} 1, & \text{if } l_2 = 0 \\ 2 \cos\left(2\pi (l_1 + D)/(2K)\right), & \text{if } 1 \leq l_2 \leq K/2 - 1 \\ (-1)^{l_1 + D}, & \text{if } l_2 = K/2. \end{cases} \quad (10)$$

The $(K - D) \times (K/2 - 1)$ real matrix $Q^2$ has its $(l_1, l_2)$th element, $0 \leq l_1 \leq K - D - 1, 0 \leq l_2 \leq K/2 - 2$, given as

$$Q^2_{l_1,l_2} = 2 \sin(2\pi (l_1 + D)/(2K)). \quad (11)$$

Since the columns of the matrix $Q$ can be obtained from the columns of the matrix $G$ in (5) and vice versa, we have the following.

Property II.1: The rank of matrix $Q$ is $K - D$.

The optimization problem is reformulated as the following real optimization problem with linear constraints:

$$\min \{G[v_{l_1,l_2} \in \mathbb{R}^{K-D \times 1} : (l_1,l_2) \in L] \}$

$$Q v_{l_1,l_2} = 0_{K-D \times 1}, \quad \forall (l_1, l_2) \in L. \quad (12)$$

III. TRANSFORMATION TO AN UNCONSTRAINED OPTIMIZATION PROBLEM

The optimization problem (12) is a linear constrained optimization problem. This problem is now equivalently transformed into an unconstrained optimization problem. For all $(l_1, l_2) \in L$, the constraint $Q v_{l_1,l_2} = 0_{K-D \times 1}$ indicates that the vector $v_{l_1,l_2}$ is in the null space of the matrix $Q$. Since the rank of $Q$ is $K - D$, the rank of the null space matrix is $D$. The matrix $Q$ can be expressed in terms of its singular value decomposition (SVD) as

$$Q = C_1 X C_2^T \quad (13)$$

where $C_1$ is a $(K - D) \times (K - D)$ orthogonal matrix. The matrix $X = [X_1 \quad 0_{(K-D) \times D}]$ has dimension $(K - D) \times K$, where $X_1$ is a $(K - D) \times (K - D)$ diagonal matrix with the singular values of $Q$ that lie on its diagonal in decreasing order and $0_{(K-D) \times D}$ is a $(K - D) \times D$ zero matrix. The $K \times K$ matrix $C_2$ is expressed as

$$C_2 = [C_{21} \quad B] \quad (14)$$
where $C_{21}$ is a $K \times (K - D)$ matrix containing an orthonormal basis of vectors for the row space, while $B$ is a $K \times D$ orthonormal null space matrix of $Q$. Thus, we have $B^T B = I_{D \times D}$. In addition, any vector $v_{i+1}^{(i)}$ in the null space of the matrix $Q$ can be expressed uniquely in terms of a vector $u_{i+1}^{(i)} \in \mathbb{R}^{D \times 1}$ and vice versa, i.e., $v_{i+1}^{(i)} = Bu_{i+1}^{(i)}$ and $u_{i+1}^{(i)} = B^T v_{i+1}^{(i)}$.

By using the matrix $B$, the optimization problem (12) can be reformulated as the following unconstrained optimization problem in terms of $u_{i+1}^{(i)}$:

$$
\min_{u_{i+1}^{(i)} \in \mathbb{R}^{D \times 1}, v_{i+1}^{(i)} \in \mathbb{L}} G(u_{i+1}^{(i)}, v_{i+1}^{(i)}) \quad (15)
$$

where

$$
G(u_{i+1}^{(i)}, v_{i+1}^{(i)}) = g(Bu_{i+1}^{(i)}, v_{i+1}^{(i)}) = g(v_{i+1}^{(i)}, u_{i+1}^{(i)}) \quad (16)
$$

Denote by $u$ and $v$, respectively, a $L(L-1)D \times 1$ vector consisting of $\{u_{i+1}^{(i)}, (i, l_2) \in \mathbb{L}\}$ and a $L(L-1)K \times 1$ vector consisting of $\{v_{i+1}^{(i)}, (i, l_2) \in \mathbb{L}\}$. The cost functions in (12) and (15) can be written in a simplified form as $g(v)$ and $G(u)$.

We now obtain the gradient of the cost function for the unconstrained problem (15). From the chain rule, we have

$$
\nabla_u G(u) = \frac{\partial G(u)}{\partial u_{i+1}^{(i)}} = B^T \frac{\partial g(v)}{\partial u_{i+1}^{(i)}} = B^T \nabla_v g(v) \quad \forall (l_1, l_2) \in \mathbb{L} \quad (17)
$$

In addition, following from (7), we have

$$
\nabla_v g(v) = 2 [\mathcal{R} \{D_{l_2}(0)\}, \ldots, \mathcal{R} \{D_{l_2}(K/2)\}] \\
\quad \mathcal{I} \{D_{l_2}(1)\}, \ldots, \mathcal{I} \{D_{l_2}(K/2 - 1)\} \quad (18)
$$

where $D_{l_2}(k)$ is the gradient of $f(W(0), \ldots, W(K/2))$ with respect to $W_{l_1,l_2}(k)$ for all $(l_1, l_2) \in \mathbb{L}$.

**IV. OPTIMIZATION PROCEDURE**

The problem (15) is an unconstrained optimization problem over real variables $\{u_{i+1}^{(i)}, (i, l_2) \in \mathbb{L}\}$. Here, the conjugate gradient method [6] based on a quadratic model is employed for the unconstrained optimization problem (15). The procedure for second-order gradient-based BSS utilizing the conjugate gradient algorithm is outlined as follows.

**Procedure IV.1:** Proposed procedure for second-order gradient-based BSS.

Step 1) Initialize the iteration $i = 0$. Obtain the $K \times D$ null space matrix $B$ for the linear constraints in (12) according to (13) and (14). Initialize the time domain unmixing matrices $W^{(0)}(\tau)$ as in [5]. The equivalent initialization for $u^{(0)}$ in the reduced domain is $u^{(0)} = 0_{L(L-1)D \times 1}$.

Step 2) Obtain the gradient with respect to the real vector $u$ at $u^{(i)}$ as in (17) and (18).

Step 3) If $i$ is a multiple of the number of variables, i.e., $\text{mod}(i, L(L-1)D) = 0$ where $\text{mod}()$ denotes the modulus operator, then obtain the conjugate gradient direction $\{s_{i, l_2}^{(i)}, (l_1, l_2) \in \mathbb{L}\}$, $s_{i, l_2}^{(i)} \in \mathbb{R}^{D \times 1}$ based only on the gradient direction of (15), i.e.,

$$
s_{i, l_2}^{(i)} = -\nabla u_{i+1}^{(i)} G(u^{(i)}) \quad \forall (l_1, l_2) \in \mathbb{L} \quad (19)
$$

Otherwise, the direction is obtained by using the Fletcher–Reeves formula [6]

$$
s_{i, l_2}^{(i)} = -\nabla u_{i+1}^{(i)} G(u^{(i)}) + \beta(i)s_{i, l_2}^{(i-1)} \quad \forall (l_1, l_2) \in \mathbb{L} \quad (20)
$$

The choice of $\beta(i)$ does not require the Hessian matrix information and is given by

$$
\beta(i) = \frac{||\nabla u_{i} G(u^{(i)})||^2}{||\nabla u_{i-1} G(u^{(i-1)})||^2} \quad (21)
$$

where $||\cdot||$ denotes the norm of a vector.

Step 4) Once the direction $\{s_{i, l_2}^{(i)}, (l_1, l_2) \in \mathbb{L}\}$ is obtained, the problem becomes how to chose the step size $\alpha$. This step size can be chosen to minimize the cost function along the conjugate gradient search direction

$$
\alpha = \arg \min_{\alpha} G(u^{(i)} + \alpha s_{i, l_2}^{(i)}) \quad (19)
$$

Here, the optimum step size is optimized by using a Golden search method [9] in combination with Brent’s method [10], where a parabola is fitted between the points where the minimum step size is located [5]. As the cost function in (19) requires to be calculated a number of times, it is more efficient to search for the step size $\alpha$ in terms of the cost function $g$. Thus, (19) can equivalently be expressed as

$$
\alpha = \arg \min_{\alpha} g\left(\mathbf{v}^{(i)} + \alpha \mathbf{s}^{(i)}\right) \quad (20)
$$

where $\mathbf{s}^{(i)} = B\mathbf{s}^{(i)}(l_1, l_2) \in \mathbb{L}$. Stop the procedure.

**V. SIMULATION RESULTS**

Simulations are performed with $K = 512$, $D = 128$, and $M = 4$. Here, we only compare the convergence and complexity between the conjugate gradient and the steepest descent algorithm [5]. Both algorithms are tested and compared for the real room and simulated environments as in [5]. The real room environment has the dimension of $3.5 \times 3.1 \times 2.3$ m$^3$ with babble noise.

The value $\epsilon_\delta$ employed in the stopping criteria in Procedure IV.1 is $\epsilon_\delta = 10^{-3}$. For each iteration, the step size $\alpha(i)$ is searched as in [5].
Table I shows the number of iterations required for convergence of both algorithms in various situations. The table also shows the average cost function $10 \log_{10}(f(W(0), \cdots, W(K/2)))$ [dB] at convergence. For the real room environment, the number of microphones increases from 2 to 4. In all cases, the BSS with conjugate gradient algorithm results in a significantly faster convergence than the steepest descent algorithm while achieving approximately the same cost function. As an example, Fig. 1 shows the convergence of both algorithms for four sets of mixing signals in real room environment with two microphones. The steepest descent algorithm requires in average 101 iterations for convergence while the conjugate gradient algorithm requires only 24 iterations with approximately the same cost function.

The table also compares the number of real multiplications required per iteration and for convergence between the conjugate gradient and the steepest descent algorithms. The number of real multiplications required for each iteration of the steepest descent algorithm, $\eta_s$, is given in [5]. The number of real multiplications required for each iteration of the conjugate gradient algorithm can be estimated as

$$
\eta_c = r_c (4K_1 (1 + M(2L_2^3 + L_2^2)) + KL(L - 1)) + 4K_1 (1 + 4MM^3) + (2DK + K + 3D)L(L - 1)
$$

where $r_c$ is the number of cost function calculations required to obtain the optimized step size for the conjugate gradient algorithm in each iteration. The complexity ratio per iteration between the conjugate gradient and the steepest descent algorithms is defined as $\eta_c/\eta_s$. The complexity ratio for convergence between the two algorithms is $(I_s/\eta_s)/(I_c/\eta_c)$, where $I_c$ and $I_s$ denote the number of iterations required by the conjugate and steepest descent algorithms for convergence. For the conjugate gradient algorithm, the number of multiplications required for convergence is significantly less than the steepest descent algorithm as it has a faster convergence, even though it requires slightly more multiplications in each iteration. Informal listening tests show that the output quality for both methods are approximately the same.

VI. CONCLUSIONS

Simulation results evaluated in real and simulated environments show that the conjugate gradient algorithm achieves in a faster convergence with lower total complexity required for convergence than the steepest descent algorithm while achieving approximately the same cost function. Further work is required to provide theory to support the improved convergence characteristics of the conjugate gradient method in the BSS application.

REFERENCES