A NEW DIAGONAL HESSIAN ALGORITHM FOR BLIND SIGNAL SEPARATION

M. Elsabrouty, T. Aboulnasr, and M. Bouchard

School of Information Technology and Engineering (SITE), University of Ottawa,
800 King Edward, Ottawa (Ontario), K1N 6N5, Canada,
e-mail: {melsabro, aboulnas, bouchard}@site.uottawa.ca

ABSTRACT

A new algorithm for blind signal separation of speech signals that does not require pre-whitening is proposed in this paper. The algorithm is based on second order optimization using Riemannian geometry. The algorithm employs several practical approximations to the Hessian matrix of the maximum-likelihood blind separation cost function, to produce a computationally efficient algorithm that is capable of working on-line. Simulation results show the improved performance of the proposed algorithm with different mixing data.

1. INTRODUCTION

Blind Source Separation (BSS) is intended to separate or recover a set of unobservable sources from another set of mixtures, without having access to the mixing matrix coefficients. The main category of cost functions for blind separation of instantaneous mixtures is information theory based methods [1]. This category is preferred due to its robustness against outliers, meaning that a single or a few highly erroneous observations should have little impact on the overall estimate of the sources. Information theory approaches aim at separating the mixtures into their basic components using estimates of the sources probability densities. One of the main cost functions in the information theory category is the Maximum-Likelihood (ML) cost function.

In [2], the idea of applying differential geometry to the LMS-update of the maximum-likelihood cost function for blind separation lead to the well-known natural gradient update that exhibits equivariant properties, meaning that the gradient is better directed to the minima of the cost function on the curved optimization manifold, i.e. Riemannian manifold. Not only does the natural gradient provide a better convergence behavior, it is also simpler than the original LMS-update as it does not require matrix inversion. The algorithm proposed in this paper builds upon the natural gradient in an attempt to provide an improved on-line algorithm with faster convergence and a low additional computational load.

The rest of this paper is organized as follows. In section 2, the principle of signal separation based on maximum-likelihood is presented, along with a review of the natural gradient. Section 3 presents the improved on-line algorithm. Section 4 is dedicated to the test setup and the presentation of the results. We finally conclude the paper in Section 5.

2. MAXIMUM-LIKELIHOOD ALGORITHMS

Let $s_i(n), i=1\cdots N$ be the scalar inputs for mixing matrix $\mathbf{A}$ at a time n. For simplicity, it is assumed that the mixing is linear and that the mixing matrix $\mathbf{A}$ is square, i.e. of size $N\times N$. The mixing model can be expressed as:

$$\mathbf{x}(n) = \mathbf{A}\times\mathbf{s}(n) \tag{1}$$

The purpose of blind signal separation algorithms is to estimate a matrix $\mathbf{W}$ such that $\mathbf{W}\times\mathbf{A}=\mathbf{P}$, where $\mathbf{P}$ is a permutation of the identity matrix $\mathbf{I}$. If this condition is met then the outputs of the separation process referred to as $y_i(n), i=1\cdots N$ are equal to the independent sources $s_i(n), i=1\cdots N$, except for order and scale ambiguity.

Maximum-likelihood attempts to perform the separation via maximizing the joint entropy $H(\mathbf{y};\mathbf{W})$. The concept has been introduced in [3]. The cost function of the log-likelihood $L(\mathbf{W})$ of the $\mathbf{W}$ can be expressed as:

$$L(\mathbf{W}) = \log\left|\det(\mathbf{W})\right| + E\left\{\sum_{i=1}^{N}\log p_i(y_i)\right\} \tag{2}$$

where $E\{.\}$ refers to the expected value, y_i is the i[th] output and $p_i(.)$ is the probability density function of y_i. The gradient of $L(\mathbf{W})$ on the Euclidean space is:

$$\nabla L(\mathbf{W}) = (\mathbf{W}^{-1})^T + E\left\{\mathbf{g}(\mathbf{y})\mathbf{x}^T\right\} \tag{3}$$

where $g(y_i) = -\dfrac{p(y_i)'}{p(y_i)}$. The operation $(.)'$ denotes the derivative. A good choice for the nonlinearity $g(y_i)$ is:

$$g(y_i) = \begin{cases} \tanh(y_i), & \text{if } y_i \text{ is supergaussian} \\ y_i^{\,3}, & \text{if } y_i \text{ is subgaussian} \end{cases} \tag{4}$$

The natural gradient $\widetilde{\nabla}L(\mathbf{W})$ can be computed by modifying the Euclidean gradient, post-multiplying it by $\mathbf{W}^T\mathbf{W}$ to yield:

$$\widetilde{\nabla} L(\mathbf{W}) = \left[I - E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\right]\mathbf{W} \quad (5)$$

A stochastic form of this algorithm can also be developed. This is achieved by replacing the expectation by the instantaneous value and in each step of the algorithm a small step size $\mu << 1$ is used to add the update to the demixing matrix $\mathbf{W}$:

$$\mathbf{W} = \mathbf{W} + \mu\left[I - \mathbf{g}(\mathbf{y})\mathbf{y}^T\right]\mathbf{W} \quad (6)$$

The natural gradient update can also be obtained directly by finding the steepest descent direction of the cost function $L(\mathbf{W})$ on the Lie-algebra space associated with $\mathbf{W}$ [4]. The modified differential along this tangent space is [5]:

$$d\mathbf{X} = d\mathbf{W}\,\mathbf{W}^{-1} \quad (7)$$

such that:

$$d\mathbf{y} = d\mathbf{W}\,\mathbf{x} = d\mathbf{W}\,\mathbf{W}^{-1}\,\mathbf{y} = d\mathbf{X}\,\mathbf{y} \quad (8)$$

Differentiation of the cost function $L(\mathbf{W})$ with respect to $d\mathbf{X}$ results in two terms:

$$d\left|\log\left(\det(\mathbf{W})\right|\right| = -tr(d\mathbf{X}), \text{ such that } \frac{d\left|\log\left(\det(\mathbf{W})\right|\right|}{d\mathbf{X}} = \mathbf{I} \quad (9)$$

and the second term is $\dfrac{\partial\sum_{i=1}^{N} E\{\log|p_i(y_i)|\}}{\partial\mathbf{X}}$ which can be estimated as follows:

$$\begin{aligned}\frac{\partial\sum_{i=1}^{N} E\{\log|p_i(y_i)|\}}{\partial\mathbf{X}} &= -E\left\{g(y_1)\frac{\partial y_1}{\partial\mathbf{X}} + \cdots + g(y_N)\frac{\partial y_N}{\partial\mathbf{X}}\right\} \\ &= -E\{g(y_1)\begin{pmatrix} y_1 & \cdots & y_N \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \end{pmatrix} + \cdots + g(y_N)\begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ y_1 & \cdots & y_N \end{pmatrix}\} \\ &= E\left\{g(\mathbf{y})\mathbf{y}^T\right\} \end{aligned} \quad (10)$$

The update along the Lie-algebra is given by:

$$\Delta\mathbf{X} = (I - g(\mathbf{y})\mathbf{y}^T) \quad (11)$$

This update is mapped to the Riemannian space of $\mathbf{W}$ via exponential mapping [6]:

$$\mathbf{W} = \exp(-\mu\,\Delta\mathbf{X})\,\mathbf{W} \quad (12)$$

At the steady state point, i.e. when separation is reached, the output signals $\mathbf{y}$ are independent and as such:

1. The output signals $\mathbf{y}$ are uncorrelated, meaning:

$$\mathrm{E}\{\mathbf{y}\,\mathbf{y}^{\mathbf{T}}\} = \begin{pmatrix} E\{{y_1}^2\} & & \\ & \ddots & \\ & & E\{{y_N}^2\} \end{pmatrix} \quad (13)$$

2. They are uncorrelated in the more general sense:

$$\mathrm{E}\{\mathbf{g}(\mathbf{y})\,\mathbf{y}^{\mathrm{T}}\} = \begin{pmatrix} E\{g(y_1)y_1\} & & \\ & \ddots & \\ & & E\{g(y_N)y_N\} \end{pmatrix} \quad (14)$$

This means that at separation the gradient along the tangent space given by equation (11) reduces to a diagonal matrix. This diagonal matrix would have a mere effect of scaling the outputs $\mathbf{y}$ without affecting the separation.

Using the natural gradient as the starting point, we propose an algorithm based on the second order differentiation along the Lie-algebra space. The algorithm proposed in the following section preserves the stability of the separation by ensuring that the update at the optimal separation point stays diagonal. The proposed algorithm is capable of producing improved acceleration of separation with a low computational cost.

3. DIAGONAL HESSIAN ALGORITHM

3.1. Second Order Differentiation

The second order differentiation of the cost function $L(\mathbf{W})$ of the matrix $\mathbf{W}$ of size $N \times N$, denoted by $\nabla^2 L(\mathbf{W})$ is of size $N^2 \times N^2$. $\nabla^2 L(\mathbf{W})$ results from the differentiation $\dfrac{\partial\sum_{i=1}^{N} E\{\log|p_i(y_i)|\}}{\partial\mathbf{X}\,\partial\mathbf{X}^T} = -\dfrac{\partial E\{g(\mathbf{y})\,\mathbf{y}^T\}}{\partial\mathbf{X}^T}$. Converting the matrix $\mathbf{X}$ and the first derivative row-by-row into vectors of size $N^2 \times 1$. The differentiation is composed of two terms:

1. The first results from the application of a differentiation operator on the nonlinearity $\mathbf{g}(\mathbf{y})$: $-\sum_{i=1}^{N}\dfrac{\partial g(y_i)}{\partial(y_i)}\dfrac{\partial y_i}{\partial\mathbf{X}^T}$. The differentiation of this term, called **Hessian1**, results in a block diagonal matrix of size $N^2 \times N^2$ with blocks along the diagonal of size $N \times N$ of the from:

$$(\mathbf{Hessian\,1})_i = -E\left\{g'(y_i)\,\mathbf{y}\,\mathbf{y}^T\right\}, \quad i = 1\cdots N \quad (15)$$

2. The second term of the differentiation results from applying the differentiation operator on the output $\mathbf{y}$. This term is called **Hessian2** and is also of size $N^2 \times N^2$.

$$\mathbf{Hessian\ 2} = -E\{\begin{pmatrix} g(y_1)\begin{pmatrix} y^T & \mathbf{0}_{-1\times(N^2-N)} \end{pmatrix} \\ g(y_1)\begin{pmatrix} \mathbf{0}_{-1\times N} & y^T & \mathbf{0}_{-1\times(N^2-N)} \end{pmatrix} \\ g(y_1)\begin{pmatrix} \mathbf{0}_{-1\times(N^2-N)} & & y^T \end{pmatrix} \\ \vdots \\ g(y_N)\begin{pmatrix} y^T & \mathbf{0}_{-1\times(N^2-N)} \end{pmatrix} \\ g(y_N)\begin{pmatrix} \mathbf{0}_{-1\times N} & y^T & \mathbf{0}_{-1\times(N^2-N)} \end{pmatrix} \\ g(y_N)\begin{pmatrix} \mathbf{0}_{-1\times(N^2-N)} & & y^T \end{pmatrix} \end{pmatrix}\} \quad (16)$$

The second order differentiation, i.e. Hessian matrix is given by:

$$\nabla^2 L(\mathbf{W}) = \mathbf{Hessian\ 1} + \mathbf{Hessian\ 2} \quad (17)$$

The second order update can thus be of the following form:

$$\mathbf{W}_{\text{vec}} = (-\nabla^2 L(\mathbf{W}))^{-1}\left[(I - E\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\})\mathbf{W}\right]_{\text{vec}} \quad (18)$$

Where vec(.) is an operation transforming a matrix of size $N \times N$ to a vector of size $N^2 \times 1$.

Equation (18) requires the inversion of the Hessian matrix to calculate the update. However, the Hessian matrix $\nabla^2 L(\mathbf{W})$ is of size $N^2 \times N^2$ and thus its inversion would be computationally expensive. It would be advantageous to apply some realistic approximations so that the inversion of the Hessian matrix becomes easier to compute.

3.2. The Diagonal Hessian Algorithm

In order to approximate the Hessian matrix with a simpler structure, the conditions of separation presented in equations (13) and (14) are revisited. Equation (15), which describes the block diagonal matrices of the first term of the Hessian matrix, **Hessian1**, can thus be approximated by applying equation (13). The approximated **Hessian1** is given the name **H1Diag** and can be written as:

$$(\mathbf{H1Diag})_i = -E\left\{g'(y_i)\begin{pmatrix} {y_1}^2 & & \\ & \ddots & \\ & & {y_N}^2 \end{pmatrix}\right\}, i = 1 \cdots N \quad (19)$$

As the name indicates, it is clear that the approximation in equation (13) reduces the block diagonal matrices $\mathbf{Hessian\ 1}_i$, $i = 1 \cdots N$ to mere diagonal matrices. The second term **Hessian2**, given by equation (16), can also be approximated by applying equation (14) and taking into account that at the separation:

$$\frac{\partial y_i}{\partial X} = \begin{pmatrix} 0 & \cdots & \cdots & \cdots \\ \vdots & \cdots & y_i & \vdots \\ \vdots & \vdots & 0 & \vdots \\ \cdots & \cdots & \vdots & \cdots \end{pmatrix} \quad (20)$$

Equation (20) and equation (14) help to reduce the second term of the Hessian matrix, **Hessian2**, to a very sparse diagonal matrix, named **H2Diag**, given by the equation:

$$\mathbf{H2Diag} = -\begin{pmatrix} \begin{pmatrix} E\{g(y_1)y_1\} & 0 & \cdots \\ 0 & \vdots & 0 \\ \vdots & \vdots & \vdots \end{pmatrix} & & \\ & \ddots & \\ & & \begin{pmatrix} 0 & \cdots & \cdots \\ 0 & \vdots & 0 \\ \vdots & \vdots & E\{g(y_N)y_N\} \end{pmatrix} \end{pmatrix} \quad (21)$$

The two approximations in equations (19) and (21) are crucial. They help to approximate the Hessian matrix $\nabla^2 L(\mathbf{W})$ by a diagonal matrix of size $N^2 \times N^2$, which is given the name **HDiag**:

$$\mathbf{HDiag} = \mathbf{H1Diag} + \mathbf{H2Diag} \quad (22)$$

This is especially valuable in calculating the inverse of the Hessian matrix. The inversion process will thus reduce to obtaining the reciprocal of the diagonal elements:

$$\mathbf{HDiag}^{-1} = \begin{pmatrix} \frac{1}{\mathbf{HDiag}_{11}} & & \\ & \ddots & \\ & & \frac{1}{\mathbf{HDiag}_{N^2N^2}} \end{pmatrix} \quad (23)$$

A further simplification in the update can be achieved by re-arranging the diagonal $N^2 \times N^2$ $\mathbf{HDiag}^{-1}$ into an $N \times N$ matrix $\mathbf{H}$ via column stacking. This will eliminate the need to vectorize the first gradient $(I - E\{g(\mathbf{y})\mathbf{y}^T\})$. The update along the Lie algebra is given by:

$$\Delta\mathbf{X} = -\mathbf{H} \circ (I - E\{g(\mathbf{y})\mathbf{y}^T\}) \quad (24)$$

where the operation "$\circ$" denotes element by element multiplication. The update $\Delta\mathbf{X}$ can be calculated on-line from sample t-1 to sample t by replacing the term $(I - E\{g(\mathbf{y})\mathbf{y}^T\})$ by its instantaneous value $(I - g(\mathbf{y}(t))\mathbf{y}(t)^T)$ and applying a moving window, that we chose to be exponential, to replace the statistical expectation of the **HDiag** matrix by the following:

$$\mathbf{HDiag}(t) = \lambda\, \mathbf{HDiag}\,(t-1) + \mathbf{HDiag}_{\text{instantaneous}} \quad (25)$$

where the forgetting factor λ is chosen close to 1.

The update along the Riemannian space of the matrix $\mathbf{W}$ can be estimated via exponential mapping of $\Delta\mathbf{X}$, and it can be implemented on-line as:

$$\mathbf{W}(t) = \exp(-\mu\, \Delta\mathbf{X}(t))\, \mathbf{W}(t-1)$$
$$\approx \mathbf{W}(t-1) - \mu\, \mathbf{H}(t) \circ (I - \mathbf{g}(\mathbf{y}(t))\mathbf{y}(t)^T)\, \mathbf{W}(t-1) \quad (26)$$

The reduction in complexity of the Diagonal Hessian algorithm compared to using the direct form of the Hessian is clear. The most computationally demanding operation, namely the inversion of the $N^2 \times N^2$ Hessian matrix, is reduced to scalar divisions and element-by-element multiplications as indicated by equations (23),(24). The additional complexity of the Diagonal Hessian algorithm will thus be in the order of N^2 additional operations, which is fairly low.

4. PERFORMANCE OF THE PROPOSED ALGORITHM

In order to assess the improvement produced by the proposed algorithm given by equation (26), it is compared to the natural gradient algorithm, given by equation (6). The step size of the natural gradient algorithm is set to 0.0005, while the step size of the Diagonal Hessian algorithm is set to 0.25. The forgetting factor λ of the Diagonal Hessian

algorithm is set to change from 0.994 at sample t=0 to 0.999 at t=25000. The nonlinear function $g(\mathbf{y}) = \tanh(\mathbf{y})$.

The test was performed on audio data files from [7]. These files are sampled at 8 kHz and are of duration 6.5 sec each, in the following two groups:

(a) Miscellaneous_4: two males, one female, and soft music.

(b) Male_4: 4 male speakers with similar characteristics.

We use the following performance index [8], [9]:

$$\eta = 20 \log \left(\sum_{i=1}^{N} \left(\sum_{j=1}^{N} \frac{|P_{ij}|}{\max_k |P_{ik}|} - 1 \right) + \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \frac{|P_{ij}|}{\max_k |P_{kj}|} - 1 \right) \right) \quad (27)$$

where $\mathbf{P} = \mathbf{WA}$ is a permutation matrix. The tests are performed for 100 different runs. The matrix $\mathbf{A}$ is randomly generated for each run. The following two figures illustrate the results of the simulations.

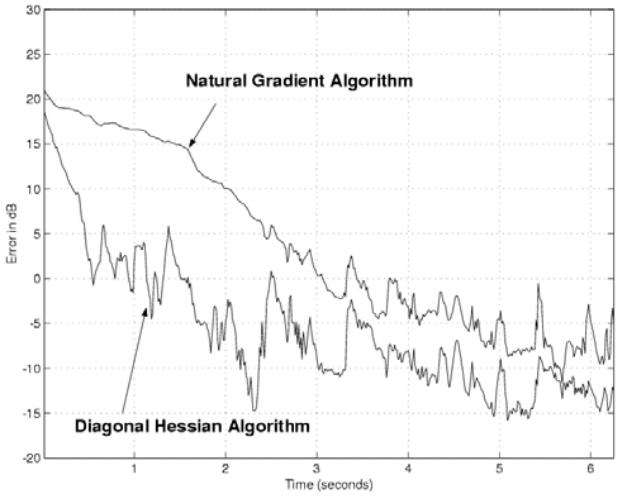

Fig.1. The average of 100 randomly generated runs for the first group of sources (Miscellaneous_4)

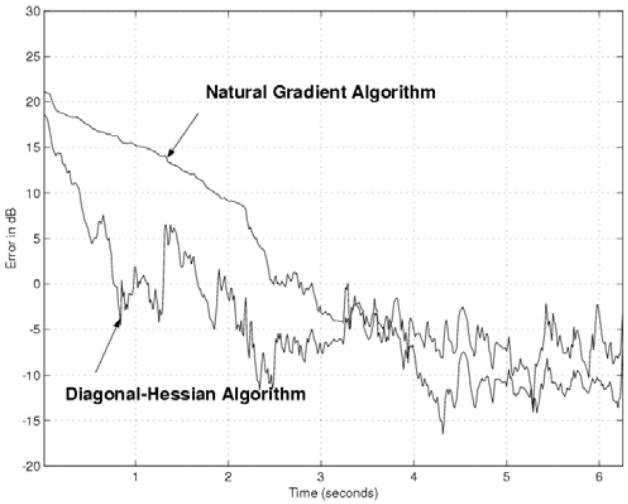

Fig.2. The average of 100 randomly generated runs for the second group of sources (Male_4)

The simulation results indicate that the proposed algorithm can actually provide an improved convergence speed and a slightly lower steady state error of separation. This improved performance is produced with a low additional complexity over the natural gradient.

5. CONCLUSION

This paper introduced an improved on-line algorithm for blind signal separation using the maximum-likelihood principle. The algorithm implements an approximation of the Hessian matrix on the Riemannian manifold and is capable of working on-line and to separate audio mixtures with good results, as confirmed by the assessment quality measure in different mixing situations. The algorithm produces an accelerated performance over the existing natural gradient algorithm with only a low additional computational load.

6. REFERENCES

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

[2] S. Amari, A. Cichocki, and H.H. Yang, "A new learning algorithm for blind source separation". In *Advances in Neural Information Processing 8,* p.p. 757-763. MIT Press, Cambridge, MA, 1996.

[3] A. Bell, and T. Sejnowski, "An information-maximization approach for blind separation and blind deconvolution". *Neural Comput*., vol.7, p.p. 1129-1159, 1995.

[4] S. Smith, *Geometric Optimization for Adaptive filtering*, PhD. Thesis, Harvard University, Cambridge, MA, 1993.

[5] A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing*, John Wiley & Sons, NY, 2002.

[6] W. Boothby, *An Introduction to Differential Manifolds and Riemannian Geometry*, 2nd Ed., Academic Press 2002.

[7] Helsinki University of Technology, Laboratory of Computer and Information Science, Neural Network Research Center, "ICA Sources and Demonstration" http://www.cis.hut.fi/projects/ica/cocktail/cocktailen.cgi

[8] H.H. Yang and S. Amari, "Adaptive online learning algorithms for blind separation: Maximum entropy and minimum mutual information," *Neural Comput.*, vol. 9, pp. 1457–1482, 1997.