A EUCLIDEAN DIRECTION BASED ALGORITHM FOR BLIND SOURCE SEPARATION USING A NATURAL GRADIENT

Glen W. Mabey, Jacob Gunther, and Tamal Bose

Center for High-speed Information Processing (CHIP),
Electrical and Computer Engineering Department,
Utah State University, Logan, Utah, USA

ABSTRACT

The development in this paper is a extension of the adaptive RLS-type algorithm proposed by Zhu and Zhang [1]. Their work uses the matrix inversion lemma to iteratively solve the equation obtained from the natural gradient of the nonlinear principle component analysis problem. This paper reduces the complexity of the solution by applying the Euclidean Direction Search concept in place of the matrix inversion lemma. The simulations performed show that the convergence rate is comparable, albeit slower, but with reduced complexity per iteration.

1. INTRODUCTION

Blind Source Separation (BSS) is a classic problem in signal processing, and has been generalized under the name of Independent Component Analysis (ICA). For BSS, this problem consists of separating multiple inputs to a system when only outputs are observable. The objectives in ICA are more generic; a signal model is not assumed, but rather the procedure aims to identify underlying variables of some process. However, the formulation and solution for each are practically identical. Requirements for these algorithms to succeed are that the sources be independent statistically and that there be at least as many observed signals as there are sources.

A wide variety of approaches have been taken toward this problem, with considerable success [2]. However, a persistent issue that remains pertinent is the computational requirements of the algorithm. Several important advances have been made recently in improving convergence with applying an RLS-type solution to the adaptive case [3], with the use of the natural gradient approach [4], and with using both [1]. These techniques build on a nonlinear principle component analysis problem posed in [5].

The present work takes this problem formulation and applies the Euclidean Direction Search (EDS) algorithm to the recursive solution. In this, we use the same application of the natural gradient used in [1], and simply substitute the EDS methodology in calculating the update vector. By doing so, we reduce the complexity of the solution for each iteration.

2. GRADIENT DERIVATION

Assume the form of a multiple-input multiple-output mixing system to be

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

This work was supported in part by the Utah Centers of Excellence program and in part by NASA Grant NAG-5-10716. Corresponding author can be reached at Glen.Mabey@usu.edu. Prepared with LyX.

where $\mathbf{s}_t = [s_1(t) \cdots s_n(t)]^T$ is a vector of mutually-independent scalar-valued inputs to the mixing system $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$. The BSS problem is to find an "un-mixing" matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$ such that $\mathbf{y}_t = \mathbf{B}\mathbf{x}_t$ has components that are as independent as possible.

In our development, we assume that the generalized form in (1) has already been followed by a whitening filter $\mathbf{V} \in \mathbb{R}^{n\times m}$ with $\mathbf{v}_t = \mathbf{V}\mathbf{x}_t$ such that the components of $\mathbf{v}_t$ are unit variance and uncorrelated. The total separating matrix is given as $\mathbf{B} = \mathbf{W}\mathbf{V}$, and we seek to find $\mathbf{W} \in \mathbb{R}^{n\times n}$. Due to the number of unknown variables which are involved in the mixing process, the un-mixer is only able to restore the original sources up to an order permutation and scaling; they will be output at unit variance. As the observed signals are already whitened by $\mathbf{V}$, $\mathbf{W}$ should be orthogonal ($\mathbf{W}\mathbf{W}^T = \mathbf{I}$).

The basic PCA cost function given in [5] is

$$J(\mathbf{W}_t) = E\left\{\left\|\mathbf{v}_t - \mathbf{W}_t^T\mathbf{g}(\mathbf{W}_t\mathbf{v}_t)\right\|^2\right\} \tag{2}$$

where $\mathbf{g}(\mathbf{y}) = [g(y_1), \ldots, g(y_n)]^T$ with nonlinear function $g(\cdot)$. Following the development in [3,6], we replace the expectation of the mean-squared error in (2) by an exponentially weighted sum, and we replace $\mathbf{g}(\mathbf{W}_t\mathbf{v}_t)$ with $\mathbf{z}_t = \mathbf{g}(\mathbf{W}_{t-1}\mathbf{v}_t)$:

$$J(\mathbf{W}_t) = \sum_{k=1}^{t} \beta^{t-k}\left\|\mathbf{v}_k - \mathbf{W}_t^T\mathbf{z}_k\right\|^2$$

where $0 < \beta \leq 1$. This yields the matrix of partial derivatives

$$\nabla J(\mathbf{W}_t) = \sum_{k=1}^{t} \beta^{t-k}\left\{-\mathbf{z}_k\mathbf{v}_k^T + \mathbf{z}_k\mathbf{z}_k^T\mathbf{W}_t\right\}. \tag{3}$$

Conventional gradient descent algorithms update the parameters in the direction of the partial derivatives. However, if the parameter space has the structure of a differentiable manifold other than $\mathbb{R}^N$, then the matrix of partial derivatives does not point in the "true gradient direction." Amari [7] showed that on the space of invertible matrices, the true gradient direction can be calculated as a linear transformation on the matrix of partial derivatives: $\tilde{\nabla}J(\mathbf{W}) = \nabla J(\mathbf{W}) \cdot \mathbf{W}^T\mathbf{W}$. In this paper, the parameter space (the manifold) is the set of orthonormal matrices. For this manifold, Edelman *et al.* [8] showed the relationship between $\tilde{\nabla}J$ and ∇J to be

$$\tilde{\nabla}J(\mathbf{W}) = \mathbf{W}\mathbf{W}^T \cdot \nabla J(\mathbf{W}) - \mathbf{W} \cdot [\nabla J(\mathbf{W})]^T \cdot \mathbf{W}.$$

In their development, Zhu and Zhang apply this general form to (3), which, after some simplification yields

$$\tilde{\nabla} J(\mathbf{W}_t) = \sum_{k=1}^{t} \beta^{t-k} \left\{ -\mathbf{z}_k \mathbf{v}_k^T + \mathbf{y}_k \mathbf{z}_k^T \mathbf{W}_t \right\}. \tag{4}$$

Incidentally, this natural gradient modification somewhat resembles the Recursive Instrumental Variable (RIV) twist on the RLS algorithm in [9].

3. EUCLIDEAN DIRECTION SEARCH

The Euclidean Direction Search (EDS) was first proposed by Xu and Bose [10–12] as an alternative to the well-known Recursive Least Squares (RLS) iterative solution. A good introduction to this technique is presented in Chapter 8 of [13], where it is derived alongside the RLS algorithm. The key idea in the EDS approach is that a major simplification (resulting in order-n computations instead of order-n^2 for RLS) can be made by calculating the optimal step size in a gradient descent along a single component, $\mathbf{e}_i$.

In a one-dimensional signal processing context, the RLS algorithm uses the matrix inversion lemma to iteratively solve the normal equation

$$\mathbf{Q}(t)\mathbf{w}(t) = \mathbf{r}(t)$$

for $\mathbf{w}(t)$ without explicitly inverting $\mathbf{Q}$, since $\mathbf{Q}$ is modified at each time step with a rank-one update. In [13], the EDS algorithm is derived with the same objectives, but takes the approach of finding the best step size α for minimizing the value of the cost function $J_t(\mathbf{w}+\alpha\mathbf{h})$. When the direction $\mathbf{h}$ is chosen to be one of $\mathbf{e}_i = [0 \cdots 0 \; 1 \; 0 \cdots 0]$, where the 1 appears in the i-th position, the formula which yields this optimal step size

$$\nabla_\alpha \triangleq \frac{\partial J_t(\mathbf{w}+\alpha\mathbf{h})}{\partial \alpha} = 0$$

is greatly simplified.

Ideally, at each time step n, this optimal step size and update would be performed for each of the parameters which are being estimated. However, the result is that the complexity of the solution remains order-n^2. A variation of the EDS making it suitable for real-time implementations is known as the Fast Euclidean Direction Search (FEDS). This algorithm differs in that at each time step, only a single parameter is updated. There are also some minor computational adjustments which need to be made. This results in moderately degraded performance, but also produces an order-n solution. A more thorough comparison between EDS and FEDS as well as simulations can be found in [13].

4. EDS-TYPE SOLUTION TO BSS

The work of Zhu and Zhang [1] does not precisely apply the traditional RLS algorithm to the BSS problem, but rather uses the matrix inversion lemma to solve $\tilde{\nabla} J = 0$ efficiently. In this paper, computational efficiency is achieved using the concept of an optimal step size in a Euclidean direction.

Consider minimizing with respect to α the function

$$J(\mathbf{W}+\alpha\mathbf{E}_{ij}),$$

with $\mathbf{E}_{ij}$ an $n \times n$ matrix whose elements are all 0 except for a 1 at element (i,j). As such, it is an update to a single element of $\mathbf{W}$. Then, applying the chain rule,

$$\begin{aligned} \frac{\partial J(\mathbf{W}+\alpha\mathbf{E}_{ij})}{\partial \alpha} &= \sum_{k,l} \left[\left. \frac{\partial J}{\partial \mathbf{W}_{kl}} \right|_{\mathbf{W}+\alpha\mathbf{E}_{ij}} \cdot \underbrace{\frac{\partial (\mathbf{W}+\alpha\mathbf{E}_{ij})_{kl}}{\partial \alpha}}_{=\delta_{k-i}\delta_{l-j}} \right] \\ &= \left[\left. \frac{\partial J}{\partial \mathbf{W}} \right|_{\mathbf{W}+\alpha\mathbf{E}_{ij}} \right]_{(i,j)}. \end{aligned}$$

So, each element (i,j) of the gradient matrix is simply element (i,j) of $\nabla J(\mathbf{W}+\alpha\mathbf{E}_{ij})$. Applying this to (3), we have

$$\begin{aligned} &[\nabla J(\mathbf{W}+\alpha\mathbf{E}_{ij})]_{ij} \\ &\quad = \left[\sum_{k=1}^{t} \beta^{t-k} \left\{ -\mathbf{z}_k \mathbf{v}_k^T + \mathbf{z}_k \mathbf{z}_k^T \left(\mathbf{W}_t + \alpha \mathbf{E}_{ij} \right) \right\} \right]_{ij} \\ &\quad = \sum_{k=1}^{t} \beta^{t-k} \Big\{ - [\mathbf{z}_k]_i \left[\mathbf{v}_k^T \right]_j + [\mathbf{z}_k]_i \left[\mathbf{z}_k^T \mathbf{W}_t \right]_j \\ &\qquad\qquad + \alpha [\mathbf{z}_k]_i \left[\mathbf{z}_k^T \right]_i \Big\} \\ &\quad = 0 \end{aligned}$$

which we set equal to zero in our search for the value of α which yields minimum cost. Solving for α leads to

$$\alpha = \frac{\sum_{k=1}^{t} \beta^{t-k} \left([\mathbf{z}_k]_i \left[\mathbf{v}_k^T \right]_j - [\mathbf{z}_k]_i \mathbf{z}_k^T \mathbf{W}_t^{(j)} \right)}{\sum_{k=1}^{t} \beta^{t-k} [\mathbf{z}_k]_i^2},$$

where $\mathbf{W}_t^{(j)}$ indicates the j-th column of $\mathbf{W}_t$, and we get the value for the update of element (i,j) of $\mathbf{W}_t$. For the natural gradient formulation in (4) we get,

$$\alpha = \frac{\sum_{k=1}^{t} \beta^{t-k} \left([\mathbf{z}_k]_i \left[\mathbf{v}_k^T \right]_j - [\mathbf{y}_k]_i \mathbf{z}_k^T \mathbf{W}_t^{(j)} \right)}{\sum_{k=1}^{t} \beta^{t-k} [\mathbf{y}_k]_i [\mathbf{z}_k]_i}.$$

The implementation which follows should be executed in its entirety for each i, j:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{W}_{t-1} \mathbf{v}_t \\ \mathbf{z}_t &= \mathbf{g}(\mathbf{y}_t) \\ \mathbf{Q}_t &= \beta \mathbf{Q}_{t-1} + \mathbf{z}_t \mathbf{v}_t^T \\ \mathbf{P}_t &= \beta \mathbf{P}_{t-1} + \mathbf{y}_t \mathbf{z}_t^T \\ \alpha &= \left[(\mathbf{Q}_t)_{ij} + (\mathbf{P}_t)_i \mathbf{W}_{t-1}^{(j)} \right] / (\mathbf{P}_t)_{ii} \\ \mathbf{W}_t &= \mathbf{W}_{t-1} + \alpha \mathbf{E}_{ij}. \end{aligned}$$

When the FEDS algorithm is implemented, it requires $2n^2 + 3n$ operations per iteration, compared to $9n^2 + 2n$ for [1]. This is considered to be a useful improvement, particularly for higher values of n.

The quest for a true order-n algorithm, where n is the number of sources, was the original aim of this work. However, for n sources there are n^2 parameters which need to be determined, in the $\mathbf{W}$ matrix (or at least $n(n-1)/2$, since $\mathbf{W}$ should be orthogonal). Hence, an order-n algorithm may not be reasonable.

Table 1. Operations per Iteration

Algorithm	Operations	For $n = 5$
Douglas [4]	$7n^2$	175
Pajunen [3]	$5n^2 + 3n$	140
Zhu [1]	$9n^2 + 2n$	235
EDS	$(2n^2 + 3n) \cdot n^2$	1625
FEDS	$2n^2 + 3n$	65

Other algorithms which laid an essential foundation for this work were developed by Pajunen [3] and Douglas [4]. Pajunen established how a least-squares approach can solve the nonlinear PCA criterion, providing a better convergence rate for the BSS problem than the traditional gradient approaches. Douglas's work is a rather purist approach to the adaptation which needs to be done, and with the use of the natural gradient, the un-mixing matrix $\mathbf{W}$ is completely orthogonal at every iteration. In Table 1, the relevant algorithms are compared with respect to the number of operations (multiplications) required to complete a single iteration. In addition, the general expression for the operations per iteration is evaluated at $n = 5$.

5. SIMULATION RESULTS

A simulation setting identical to that used in [1] was run, with

$$\mathbf{s}_t = \begin{bmatrix} \text{sign}(\cos(2\pi 155t)) \\ \sin(2\pi 800t) \\ \sin(2\pi 300t + 6\cos(2\pi 60t)) \\ \sin(2\pi 90t) \\ n(t) \end{bmatrix}$$

where $n(t) \sim \mathcal{U}(-1, 1)$. The signals were mixed with a random matrix and sampled at 10kHz. The results are shown in Figs. 1–4. The criterion for evaluating the performance of each algorithm is also taken from [1]:

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

where $\mathbf{C} = \mathbf{W}\mathbf{A} = \{c_{ij}\}$ is the combined mixing-separating matrix.

Several different variations on the value for β were experimented with. Since the signals used were stationary, instead of a single value for β, a time-varying $\beta(t)$ was used. The exception to this approach was for the EDS-gradient, which used $\beta = 0.983$ for all t. For Zhu [1] and the EDS-natural gradient,

$$\beta(t) = \lambda\beta(t-1) + (1-\lambda)$$

with $\beta(0) = 0.94$ and $\lambda = 0.995$, borrowing this form from [9]. For Pajunen [3], FEDS-gradient and FEDS-natural gradient,

$$\beta(t) = \begin{cases} \lambda\beta(t-1) + (1-\lambda) & \text{while } \beta(t-1) \leq .99 \\ 0.99 & \text{ever after.} \end{cases}$$

It is known (see [14]) that the RLS algorithm may become unstable, and this is assumed to be the phenomena observed in the

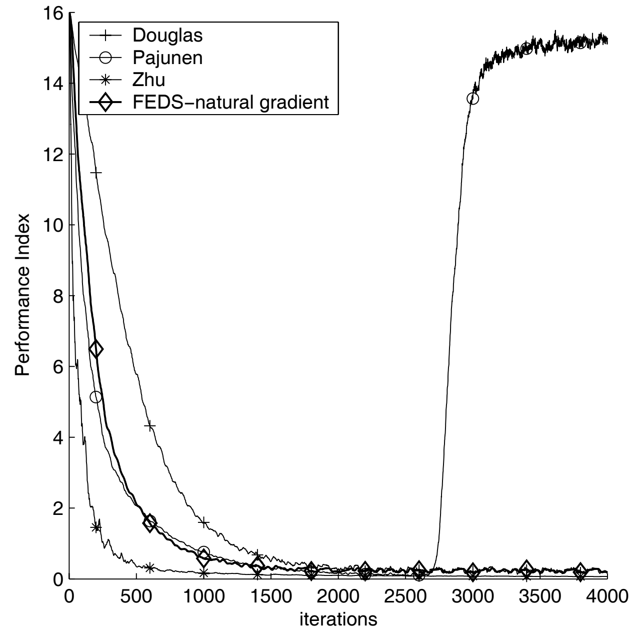

Fig. 1. Averaged results over 1000 runs of the 4 algorithms.

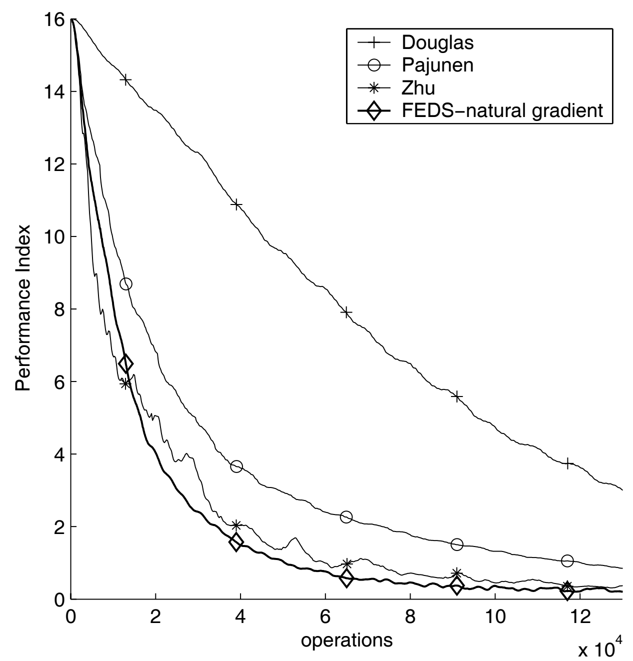

Fig. 2. Simulation results with the x-axis scaled differently for each algorithm according to the number of operations required for each iteration, as given in Table 1.

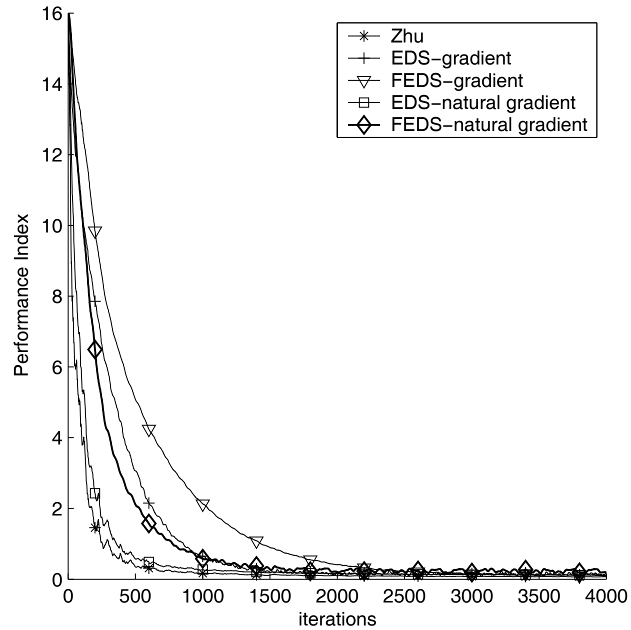

Fig. 3. Averaged results over 1000 runs of the various EDS algorithms.

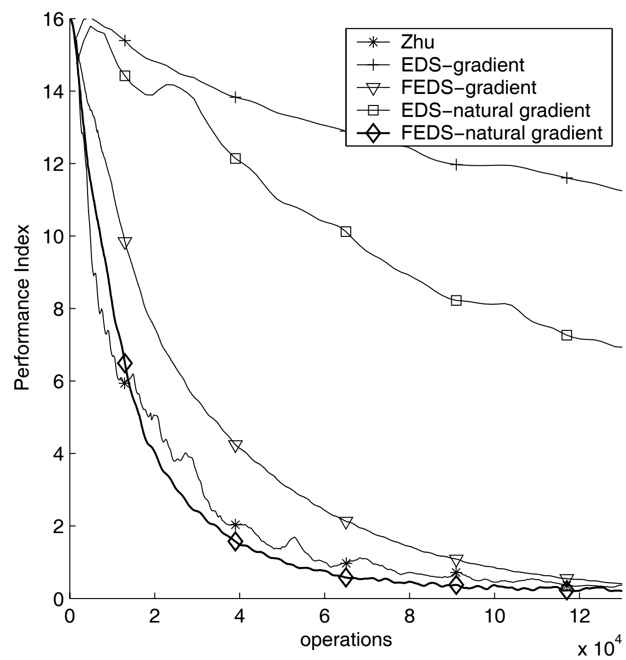

Fig. 4. Simulation results for the EDS and FEDS algorithms, with and without the natural gradient. The x-axis is scaled according to the operations per iteration in Table 1.

Pajunen algorithm in Fig. 1. Since [1] is also RLS-based, it is conceivable that instability may also result with this approach, but EDS-type algorithms are not subject to this type of instability.

In short, the EDS algorithm should be added to the list of algorithms which "can be used for solving the weight matrix $\mathbf{W}(t)$ iteratively" [3, page 6].

6. FUTURE WORK

Using the jargon of Differential Geometry, the EDS algorithm consists of searching for a minimum along a one-dimensional submanifold (formed by taking a single element of the preferred coordinate patch) of the cost space. This problem should be couched completely in a Differential Geometry setting, uniting the concepts of a search along a one-dimensional submanifold with the natural gradient, both of which belong in that realm.

7. REFERENCES

[1] X.-L. Zhu and X.-D. Zhang, "Adaptive RLS algorithm for blind source separation using a natural gradient," *IEEE Sig. Proc. Let.*, vol. 9, no. 12, pp. 432–435, Dec. 2002.

[2] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, 2001.

[3] P. Pajunen and J. Karhunen, "Least-squares methods for blind source separation based on nonlinear PCA," *Int. J. of Neural Systems*, vol. 8, no. 5-6, pp. 601–612, October/December 1997.

[4] S. C. Douglas, "Self-stabilized gradient algorithms for blind source separation with orthogonality constraints," *IEEE Trans. on Neural Networks*, vol. 11, pp. 1490–1497, November 2000.

[5] E. Oja, "The nonlinear PCA learning rule and signal separation: Mathmatical analysis," *Neurocomputing*, vol. 17, no. 1, pp. 25–46, 1997.

[6] J. Karhunen, P. Pajunen, and E. Oja, "The nonlinear PCA criterion in blind source separation: Relations with other approaches," *Neurocomputing*, vol. 22, pp. 520, 1998.

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

[8] A. Edelman, T. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, pp. 303–353, 1998.

[9] F. Fnaiech and L. Ljung, "Recursive identification of bilinear systems," *Int. J. of Control*, vol. 45, no. 2, pp. 453–470, 1987.

[10] G.F. Xu, T. Bose, W. Kober, and J. Thomas, "A fast adaptive algorithm for image restoration," *IEEE Trans. on Circ. and Sys. -I*, January 1999.

[11] G.F. Xu and T. Bose, "Analysis of the euclidean direction search set adaptive algorithm," in *ICASSP*, 1998, pp. 1689–1692.

[12] G.F. Xu, T. Bose, and J. Schroeder, "Channel equalization using a euclidean direction search based adaptive algorithm," in *Proc. of the Globecom*, November 1998.

[13] T. Bose, *Digital Signal and Image Processing*, John Wiley & Sons, New York, 2003.

[14] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, 3rd edition, 1996.