

# TECHNICAL RESEARCH REPORT

A Novel Non-Orthogonal Joint Diagonalization Cost Function for ICA

*by Bijan Afsari, P.S. Krishnaprasad*

TR 2005-106



*ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.*

*ISR is a permanent institute of the University of Maryland, within the Glenn L. Martin Institute of Technology/A. James Clark School of Engineering. It is a National Science Foundation Engineering Research Center.*

**Web site <http://www.isr.umd.edu>**

# A Novel Non-orthogonal Joint Diagonalization Cost Function for ICA

Bijan Afsari and P.S. Krishnaprasad

Institute for Systems Research , University of Maryland  
College Park, Maryland 20742, USA  
{bijan, krishna}@isr.umd.edu

**Abstract.** We present a new scale-invariant cost function for non-orthogonal joint-diagonalization of a set of symmetric matrices with application to Independent Component Analysis (ICA). We derive two gradient minimization schemes to minimize this cost function. We also consider their performance in the context of an ICA algorithm based on non-orthogonal joint diagonalization.

## 1 Introduction

Simultaneous or Joint Diagonalization (JD) of a set of estimated statistics matrices is a part of many algorithms especially for ICA both in its standard and non-stationary formulations. Historically, the first methods developed for JD were those that assume the joint diagonalizer to belong to the compact Lie group of orthogonal matrices  $O(n)$ [5] and see also [1]. Accordingly the JD problem is defined as a minimization of a function like:

$$J_1(\theta) = \sum_{i=1}^n \|\theta C_i \theta^T - \text{diag}(\theta C_i \theta^T)\|_F^2 \quad (1)$$

where  $\{C_i\}_{i=1}^N$  are the set of symmetric matrices to be diagonalized and  $\theta \in O(n)$  is the joint diagonalizer sought. We remind that due to compactness of  $O(n)$  we know

in advance that  $J_1(\Theta)$  has a minimum on  $O(n)$ . Different methods for minimization of this cost function in the context of Jacobi methods [5],[3] and optimization on manifolds have been proposed among them [11],[13].

In many cases it is believed that non-orthogonal JD is more efficient in the context of noisy ICA. In fact for the standard ICA model:

$$\mathbf{x}_{n \times 1} = A_{n \times n} \mathbf{s}_{n \times 1} + \mathbf{n} \quad (2)$$

with  $\mathbf{n}$  a Gaussian noise vector, we know that all the cumulant slices of  $\mathbf{x}$  of order higher than two are diagonalizable by an un-mixing matrix in congruence manner. Here by un-mixing matrix (with some abuse of terminology) we mean any matrix  $B$  such that  $BA = \Pi A$  where  $\Pi$  is a permutation and  $A$  is a non-singular diagonal matrix. If  $\{C_i\}_{i=1}^N$  is a subset of cumulant matrix slices of  $\mathbf{x}$  of order higher than two (which are symmetric  $n \times n$  matrices) and  $B$  is an un-mixing matrix belonging to the Lie group of non-singular matrices  $GL(n)$  then  $BC_i B^T$ 's all are diagonal. The celebrated JADE algorithm [5] uses this fact for the whitened signal for which the un-mixing matrix is forced or assumed to be orthogonal.

Defining a suitable cost function for non-orthogonal JD seems to be difficult due to non-compactness of  $GL(n)$ . For a suitable cost function  $J(B)$  we expect scale-invariance that is:  $J_1(\Lambda B) = J_1(B)$  for any non-singular diagonal matrix  $\Lambda$ . Note that mutual information also has this property: for any random vector  $\mathbf{x}_{n \times 1}$  and  $\Lambda \mathbf{x}_{n \times 1}$  have the same mutual information. In [14] and [15] a cost function of the form

$$J_2(W, \{A_i\}_{i=1}^N) = \sum_{i=1}^N \|C_i - W A_i W^T\|_F^2 \quad (3)$$

is introduced where  $A_i$  are diagonal and  $W$  is the estimated mixing matrix. Note that the minimization would be over  $W$  and  $A_i$ , therefore  $n^2 + Nn$  variables are involved whereas the original problem is in fact a search in  $n^2 - n$  variables (i.e.

finding the un-mixing or mixing matrix up to row scaling). This cost function encompasses the scale-invariability by introducing diagonal  $\Lambda_i$ . In fact in the sequel we will follow the same path, but we will choose a special  $\Lambda_i$ . Note that the single term  $\|C_i - W\Lambda_iW^T\|_F^2$  is minimized when  $W$  is a diagonalizer of  $C_i$  and  $\Lambda_i = \text{diag}(W^{-1}C_iW^{-T})$ (see (6) below). In [1] and [2] we consider using the same cost function as  $J_1(B)$  with  $B \in \text{GL}(n)$  for non-orthogonal JD. The problem with  $J_1(B)$  is that it is not scale-invariant and in fact can be reduced as  $\|B\| \rightarrow 0$ . We showed that  $J_1(B)$  has no stationary point when  $C_i$ 's do not have an exact joint diagonalizer. Based on this observation we provide some measures to deal with this issue which essentially results in a gradient flow of the form:

$$\dot{B} = -\Delta B, \quad B(0) = I_{n \times n} \quad (4)$$

where  $\Delta$  can be derived from  $H = \left( \sum_{i=1}^N (BC_iB^T - \text{diag}(BC_iB^T))BC_iB^T \right)$  by either equating diagonal of  $H$  to zero or by equating  $\Delta = H - \text{tr}(H)/n$ . Note that in both cases  $\text{tr}(\Delta) = 0$ . The first choice is equivalent to identifying  $\Lambda B$  and  $B$  for non-singular  $\Lambda$  and the second one is equivalent to identifying  $\alpha B$  and  $B$  for  $\alpha \in \mathbb{R} - \{0\}$ . The reader is referred to the companion paper [1] for further discussions.

In [16] also  $J_1(B)$  is used and a heuristic minimization method based on the idea multiplicative updates is developed which uses the idea of zero-diagonal in the updates. In [12] the cost function:

$$J_3(B) = \sum_{i=1}^N \log \left( \frac{\det \text{diag}(BC_iB^T)}{\det BC_iB^T} \right) \quad (5)$$

is introduced where  $C_i$  are required to be positive definite. Note that in the case of high-order cumulant slices this is not the case. Note that  $J_3(\Lambda B) = J_3(B)$  hence it is scale-invariant. This cost function is derived based on the maximum likelihood estimation of correlation matrices of Gaussian vectors [12].

In the light of (3) and (5) and the discussions above we introduce this cost function:

$$J_4(B) = \sum_{i=1}^N \left\| B^{-1}(BC_iB^T - \text{diag}(BC_iB^T))B^{-T} \right\|_F^2 =$$

$$\sum_{i=1}^N \left\| C_i - B^{-1}\text{diag}(BC_iB^T)B^{-T} \right\|_F^2 \quad (6)$$

Note that  $J_4(\lambda B) = J_4(B)$  and there is no condition on  $C_i$ 's needed. Needless to say that this cost function is formed merely on two basis; first, scale-invariance and, second that in the case that  $C_i$ 's have an exact joint diagonalizer the  $J_4$  can become zero. Therefore its applicability and further justification should be investigated. Note that the first formulation for  $J_4$  shows somehow a "normalized" version of  $J_1(B)$  and the second formulation relates to  $J_2$  with a specific  $\Lambda_i$  and using the unmixing matrix instead of the mixing matrix. One immediate problem that seems to be unavoidable is the presence of  $B^{-1}$  in  $J_4(B)$  which might make the computations costly. Note that  $J_3$  also includes terms that are not easy to compute but in [12] an approximation is used to avoid this direct computation. In the remainder we will derive a gradient flow for minimization of  $J_4(B)$ . In section (3) we also derive an adjoint equation for minimization of  $J_4(B)$  with respect to  $B^{-1}$ , based on this we suggest a discrete algorithm for minimization of  $J_4(B)$  that does not require explicit computation of  $B^{-1}$  at each step. In section (4) we will use the derived JD algorithms in an ICA algorithm introduced in [2] and consider some numerical results.

**Notation** We already used the notation  $\text{diag}(A)$  as the diagonal part of  $A$ .  $\|A\|_F$  denotes the Frobenius norm of matrix  $A$ .  $\text{tr}(A)$  is the trace of the matrix  $A$ .  $\dot{x}$  shows the time derivative of the variable  $x$ .  $T_pM$  represents the tangent space to the manifold  $M$  at point  $p$ .  $I_{n \times n}$  is the identity matrix of dimension  $n \times n$ . All random variables are in boldface small letters.

## 2 Gradient Flow for Minimization of $J_4(B)$

We consider  $GL(n)$  as a Riemannian manifold with the Riemannian metric (also known as Natural Riemannian metric [8]):

$$\langle \xi, \eta \rangle_B = \text{tr}((\xi B^{-1})^T \eta B^{-1}) = \text{tr}(B^{-T} \xi^T \eta B^{-1}) = \text{tr}(\eta (B^T B)^{-1} \xi^T) \quad (7)$$

for  $\xi, \eta \in T_B GL(n)$ . Employing the relation  $\dot{B}^{-1} = -B^{-1} \dot{B} B^{-T}$  we can show that the gradient flow for minimization of  $J_1(B)$  with respect to this Riemannian metric is:

$$\dot{B} = -\Delta B = -\left( \sum_{i=1}^N (\Psi_i \text{diag}(BC_i B^T) - \text{diag}(\Psi_i) BC_i B^T) \right) B, \quad B(0) = I_{n \times n} \quad (8)$$

where:

$$\begin{aligned} \Psi_i &= B^{-T} (C_i - B^{-1} \text{diag}(BC_i B^T) B^{-T}) B^{-1} = \\ & (BB^T)^{-1} (BC_i B^T - \text{diag}(BC_i B^T)) (BB^T)^{-1} \end{aligned} \quad (9)$$

It is interesting to note that the term  $\Delta$  in (8) which in fact belongs to the Lie algebra of  $GL(n)$ , i.e  $\mathfrak{gl}(n)$  is such that  $\text{diag}(\Delta) = 0$ . We recall that in order to use  $J_1(B)$  for non-orthogonal JD we forced the diagonal of the corresponding  $\Delta$  to be zero (See (4) and [1] or [2]) whereas using  $J_4(B)$  as a cost function we naturally reach to a flow that has this property and in fact is a flow on the group of  $n \times n$  matrices with unity determinant i.e.  $SL(n)$ . (we remind that the Lie algebra of  $SL(n)$  is the set of  $n \times n$  matrices with zero trace)

The Euler discretization of (8) with small enough step size results in the steepest descent algorithm. This can be demonstrated as:

$$B_{k+1} = (I - \mu_k \Delta_k) B_k = \left( I - \mu_k \sum_{i=1}^N \Psi_{ik} \text{diag}(B_k C_i B_k^T) - \text{diag}(\Psi_{ik}) B_k C_i B_k^T \right) B_k \quad (10)$$

where  $\Psi_{ik} = (B_k B_k^T)^{-1} (B_k C_i B_k^T - \text{diag}(B_k C_i B_k^T)) (B_k B_k^T)^{-1}$  and  $B_0 = I_{n \times n}$ . The step size  $\mu_k$  should be such that at each update the cost function is reduced and  $\det(B_k)$  is almost unity. Note that for a steepest descent algorithm on a linear space there is no restriction on the step size as long as it is such that the cost is reduced but in this case which is in fact constrained or equivalently is such that the answer is confined to the manifold  $\text{SL}(n)$  the step size should be such the updates stay on the manifold to a good extend. In [1] and [2] we proposed a discretization scheme based on the LU decomposition of  $B$  that keeps the updates on  $\text{SL}(n)$  by construction which can be applied to the present case too. Here, however we choose to pick a small and fixed step size for discretization. Of course adaptive step-size methods with an eye on keeping the updates on  $\text{SL}(n)$  will result in faster algorithms. The pseudo code for this algorithm is:

**Algorithm 1**

1. **Set**  $\mu$  and  $\epsilon$ .
2. **Set**  $B_0 = I_{n \times n}$  or to a good initial guess.
3. **While**  $\|X_k\|_F > \epsilon$  **do**  $B_{k+1} = (I - \mu \Delta_k) B_k$   
**if**  $\|B_{k+1}\|_F$  is "big" **then reduce**  $\mu$  **and goto** 2.
4. **End**

**3 An Inverse-Free Algorithm**

In implementing the discretized expression (10) we ought to compute  $B_k^{-1}$  or  $(B_k B_k^T)^{-1}$  which can be costly. However, one should notice that in the context of ICA the number of matrices  $C_i$  i.e.  $N$  is quite large compared to  $n$  and in fact if all the fourth order cumulant slices are used  $N = n^2$ . Even when a subset of cumulant slices is used usually  $N \gg n$ . Therefore the cost of computing  $B_k^{-1}$  is comparable (in order of magnitude) to that of the rest of (10). In the case  $N = n^2$  the complexity of com-

puting (10) is of order  $O(n^4)$  whereas the complexity of computing  $B_k^{-1}$  is of order  $O(n^3)$ . Hence we may conclude that in the context of JD for ICA, the computation of inverse is not a significant burden compared to the rest of computations required in (10). Still we may find it interesting to develop a scheme for minimization of  $J_4$  which is free of computing inverses. To this end here we propose a method of adjoint equations. It is possible to show that the gradient flow for minimization of  $J_4(B)$  with respect to  $B^{-1}$  is:

$$\dot{B}^{-1} = -B^{-1}\Delta \tag{11}$$

where  $\Delta$  is as in (8). This flow is derived with respect to a Riemannian metric defined as:

$$\langle \xi, \eta \rangle_B = \text{tr}((B^{-1}\xi)^T B^{-1}\eta) = \text{tr}(\xi^T B^{-T} B^{-1}\eta) = \text{tr}(\eta(BB^T)^{-1}\xi^T) \tag{12}$$

The reason for selecting this metric is that considering  $J_4$  as a function of  $B^{-1}$  we expect  $J_4(B^{-1}A) = J_4(B^{-1})$ , so at the point  $B^{-1}$  a tangent vector of the form  $B^{-1}\Delta$  is suitable. Notice that this flow can found from (8) via the equation  $\frac{d}{dt}B^{-1} = -B^{-1}\dot{B}B^{-1}$ .

We consider the Euler discretization of (11):

$$B_{k+1}^{-1} = B_k^{-1}(I + \mu_k\Delta_k), \quad B_0^{-1} = I \tag{13}$$

where  $\Delta_k$  has the same expression as in (10). The idea then is to use (10) to update  $B_{2^{k-1}}$  and (13) to update  $B_{2^k}^{-1}$ . In practice this leads to an update for the inverse that is not exactly equal to inverse of  $B_k$  therefore for clarity we replace  $B_k^{-1}$  with  $Q_k$  in the expressions. It was noted in practice that the second formulation for  $\Psi_i$  in (9) performs better than the first one therefore in the algorithm the former is used.

Here is a pseudo code for this algorithm:

**Algorithm 2**

Consider the set  $\{C_i\}_{i=1}^N$  of symmetric matrices.



1. **Set**  $\mu$  **and**  $\epsilon$ .
2. **Set**  $B_{-1} = Q_{-1} = B_0 = Q_0 = I_{n \times n}$ .
3. **do**{
 
$$\Psi_{i2k+1} = Q_{2k}^T Q_{2k} (B_{2k-1} C_i B_{2k-1}^T - \text{diag}(B_{2k-1} C_i B_{2k-1}^T)) Q_{2k}^T Q_{2k}$$

$$\Delta_{2k+1} = \sum_{i=1}^N \Psi_{i2k+1} \text{diag}(B_{2k-1} C_i B_{2k-1}^T) - \text{diag}(\Psi_{i2k+1}) B_{2k-1} C_i B_{2k-1}^T$$

$$B_{2k+1} = (I - \mu \Delta_{2k+1}) B_{2k-1}$$

$$\Psi_{i2k+2} = Q_{2k}^T Q_{2k} (B_{2k+1} C_i B_{2k+1}^T - \text{diag}(B_{2k+1} C_i B_{2k+1}^T)) Q_{2k}^T Q_{2k}$$

$$\Delta_{2k+2} = \sum_{i=1}^N \Psi_{i2k+2} \text{diag}(B_{2k+1} C_i B_{2k+1}^T) - \text{diag}(\Psi_{i2k+2}) B_{2k+1} C_i B_{2k+1}^T$$

$$Q_{2k+2} = Q_{2k} (I + \mu \Delta_{2k+2}) \}$$
**if**  $\|B_{2k+1}\|_F$  **or**  $\|Q_{2k+2}\|_F$  **is "big"** **then reduce**  $\mu$  **and goto** 2.  
**While**  $\|\Delta_{2k+1}\|_F$  **or**  $\|\Delta_{2k+2}\|_F > \epsilon$
4. **End**

## 4 Simulations

Here we consider the performance of the developed non-orthogonal JD methods in the context of an ICA algorithm introduced in [2] which its salient feature is that although it whitens the data it does not confine the search space afterwards to  $O(n)$ . Consider the data model (2) A simplified version of that algorithm is:

1. Whiten  $\mathbf{x}$ , let  $W$  be the whitening matrix, compute  $\mathbf{y} = W\mathbf{x}$ .
2. Estimate  $C = \{C_i\}_{i=1}^N$  a subset of the fourth order cumulant slice matrices of  $\mathbf{y}$ .
3. Jointly diagonalize  $C = \{C_i\}_{i=1}^N$  by a non-orthogonal matrix  $B_{JDN}$  (using any algorithm like Algorithms 1 or 2) and set  $B = B_{JDN}W$ .
4. Compute  $\hat{\mathbf{x}} = B\mathbf{x}$

**Example:** Consider

$$\mathbf{x} = A\mathbf{s}_{n \times 1} + \sigma\mathbf{n} \tag{14}$$

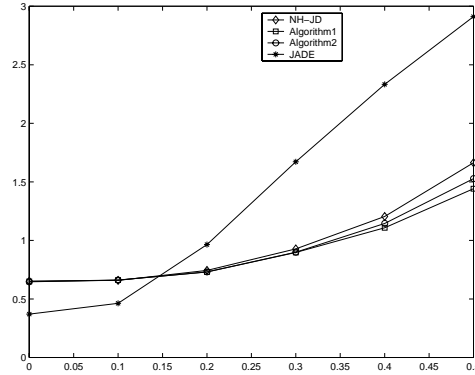
where  $\mathbf{n}$  is zero mean Gaussian noise with identity correlation matrix then  $\sigma^2$  indicates the power of noise. We consider  $n = 5$  sources all of them uniformly distributed in  $[-\frac{1}{2}, \frac{1}{2}]$ . The matrix  $A$  is randomly generated and truncated to integer entries:

$$A = \begin{bmatrix} 8 & -3 & 6 & -16 & 5 \\ 12 & 6 & 11 & 2 & 2 \\ -15 & 8 & -12 & -10 & -9 \\ -14 & 7 & 0 & 14 & -21 \\ 5 & 12 & -1 & -8 & 0 \end{bmatrix}$$

We generate  $T = 3500$  samples of data and mix the data through  $A$ . Next we run four ICA algorithms. Three algorithms NH-JD [2] and Algorithm 1 and Algorithm 2 in addition to the standard JADE are applied to the data.  $N = n^2$  fourth order cumulant matrix slices are used. The NH-JD algorithm is basically the above ICA algorithm whose JD part is the implementation of minimization of  $J_1(B)$  via the flow (4) with the nonholonomic constraint  $\Delta = H - \text{diag}(H)$ . The discretization of the JD part in NH-JD has exactly the same structure as Algorithm 1. For NH-JD, Algorithm 1 and Algorithm 2,  $\mu = .01$  and  $\epsilon = .01$  are used. These values are not optimal, they were chosen based on few tries. Implementations all are in *MATLAB*<sup>®</sup> code and the *MATLAB*<sup>®</sup> code for JADE was downloaded from: “<http://tsi.enst.fr/~cardoso/icacentral/Algos>”. The performance measure used is the distance of the product of the estimated un-mixing and the mixing matrix, i.e.  $P = BA$ , from essential diagonality:

$$\text{Index}(P) = \sum_{i=1}^n \left( \sum_{j=1}^n \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1 \right) + \sum_{j=1}^n \left( \sum_{i=1}^n \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1 \right) \quad (15)$$

For each value of  $\sigma$  the experiment is run  $k = 100$  times and the performance measure is averaged over the trials. Figure (1) shows the results. We can see that the introduced algorithms all have almost the same performance and out-perform the standard JADE in high level Gaussian noise. The run-time for these algorithms (in *MATLAB*<sup>®</sup> code) is much higher than JADE’s, although we expect faster perfor-



**Fig. 1.** Average in-noise-performance index (every point is averaged over 100 trials) of different JD based ICA algorithms. The average  $\text{Index}(P)$  is plotted versus  $\sigma$ .

mance in low-level codes or DSPs. Part of this slower convergence can be attributed to the nature of gradient based methods which have linear convergence.

## 5 Conclusion

We introduced a new scale-invariant cost function for non-orthogonal JD. We also derived a gradient minimization scheme for that cost function. To avoid computing matrix inverses we introduced an inverse-free version of the algorithm developed. We examined the performance of the developed JD in the context of an ICA algorithm introduced in [2] and compared the performance with JADE's. The developed methods out-perform JADE in high level Gaussian noise.

## 6 Acknowledgments

This research was supported in part by Army Research Office under ODDR&E MURI01 Program Grant No. DAAD19-01-1-0465 to the Center for Communicating Networked Control Systems (through Boston University).

## References

1. B. Afsari “Gradient Flow Based Matrix Joint Diagonalization for Independent Component Analysis”, Master’s Thesis, Electrical and Computer Engineering Department, University of Maryland, College Park, May 2004.(Available at:[http://techreports.isr.umd.edu/reports/2004/MS\\_2004-4.pdf](http://techreports.isr.umd.edu/reports/2004/MS_2004-4.pdf))
2. B. Afsari, P.S. Krishnaprasad, “Some Gradient Based Joint Diagonalization Methods for ICA”, Proceedings of the Fifth International Conference on Independent Component Analysis and Blind Source Separation, Springer LNCS series, 2004.
3. A. Bunse-Gerstner, R. Byers and V. Mehrmann, “Numerical Methods For Simultaneous Diagonalization”, SIAM Journal on Matrix Analysis and Applications, vol. 4, pp. 927-949, 1993.
4. M.P. Calvo, A. Iserles and A. Zanna, “Runge-Kutta methods for orthogonal and isospectral flows”, Appl. Num. Maths 22 (1996)
5. J.F. Cardoso and A. Solumiac, “Blind Beamforming For Non-Gaussian Signals”, IEE-Proceedings, Vol.140, No 6, Dec 1993
6. J.F. Cardoso, “Perturbation of joint diagonalizers”, Technical report, Telecom Paris, 1994.
7. J.F. Cardoso, “On the performance of orthogonal source separation algorithms”, EUSIPCO-94, Edinburgh.
8. S. Haykin (Editor), “Unsupervised Adaptive Filtering, Volume I, Blind Source Separation”, Wiley Interscience, 2000.
9. U.Helmke and J.B. Moore, “Optimization and Dynamical Systems”, Springer-Verlag, 1994
10. G. Hori, J.H. Manton, “Critical Point Analysis of Joint Diagonalization Criteria”, 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan

11. J.H Manton, "Optimization Algorithms Exploiting Unitary Constraints", IEEE Transactions on Signal Processing, VOL. 50, NO. 3, March 2002
12. D.T. Pham, "Joint Approximate Diagonalization of Positive Definite Hermitian Matrices", SIAM Journal of Matrix Analysis and Applications, Vol. 22, No. 4, pp. 136-1152.
13. I. Yamada, T. Ezaki, "An Orthogonal Matrix Optimization by Dual Cayley Parametrization Technique", 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan
14. A.Yeredor, "Non-Orthogonal Joint Diagonalization in the Least-Squares Sense With Application in Blind Source Separation", IEEE Transactions on Signal Processing, Vol 50, No.7.July 2002.
15. A. Van Der Veen, "Joint Diagonalization via Subspace Fitting Techniques", Proc. ICASSP, Salt Lake City, UT, 2001, vol. V, pp. 2773-2776
16. A.Ziehe, P. Laskov, K. Muller, G. Nolte, "Linear Least-squares Algorithm for Joint Diagonalization", 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan