ORTHOGONAL AND NON-ORTHOGONAL JOINT BLIND SOURCE SEPARATION IN THE LEAST-SQUARES SENSE

Marco Congedo, Ronald Phlypo, Jonas Chatel-Goldman

Equipe ViBS (Vision and Brain Signal Processing), GIPSA-lab, CNRS, Grenoble University, France

ABSTRACT

We present two new algorithms for the orthogonal and non-orthogonal joint blind source separation (JBSS), a flexible framework that extends the well-known blind source separation to the case when multiple datasets are decomposed simultaneously. The algorithms minimize the total sum of squares of the off-diagonal terms by means of very simple gradient ascent iterations.

Index Terms— Joint Blind Source Separation, Hyperscanning, Brain Coupling, Synchronization.

1. INTRODUCTION

Blind Source Separation (BSS) is a well-known framework including Independent Component Analysis and other decomposition methods aiming at recovering source signals in MIMO systems [1]. BSS nowadays encompasses a wide range of engineering applications such as speech enhancement, image processing, geophysical data analysis, wireless communication and biomedical signal analysis. Generally speaking, BSS takes as input a multivariate linearly mixed signal received over N sensors and outputs a $P\leq N$ multivariate demixed signal. More recently the joint blind source separation (JBSS) approach has been proposed in biomedical imaging [2-5]. Whenever M datasets are available and a relation exists among the sources of the M sets, this approach exploits the joint statistics of the M datasets by solving the M BSS problems simultaneously rather than independently. Several algorithms for performing JBSS in the case of orthogonal mixing matrices have been proposed [2-5]. In this paper we extend previous work performed on BSS [5-7] and we describe a least-squares gradient approach to JBSS allowing a simple solution for both the orthogonal and nonorthogonal case.

2. METHOD

2.1. Problem Statement

Suppose we are given M datasets, with m=1,…,M. As usual, we suppose that each data set is multidimensional, such as $\boldsymbol{x}_m(t)=\left[x_m^{(1)}(t),...,x_m^{(N)}(t)\right]^T$, wherein N random variables hereafter indexed by n=1,…,N, unfold over the discrete dimension t=1,…,T. Note that n and t may refer to, for instance, space and time, respectively, as it is the case for the human electroencephalogram (EEG), but this is not important for the sequel. Suppose further that K observations are available for each dataset, indexed by k=1,…,K, yielding KM groups of N variables, hereafter denoted compactly as $\boldsymbol{x}_{m,k}(t)$. To give just a few examples, in the second-order statistics framework the K observations may refer to K experimental conditions, to recordings in K different times (or trials), or to an expansion of the original data in K discrete frequencies or K time-frequency regions. We assume the following generative model for the data:

$$\boldsymbol{x}_{m,k}(t)=\boldsymbol{A}_m\boldsymbol{s}_{m,k}(t)+\boldsymbol{\eta}_{m,k}(t) \tag{1.1}$$

where $\boldsymbol{A}_m\in\mathbb{R}^{NxP}$ is a (t, k)-invariant full column rank mixing matrix, $\boldsymbol{s}_{m,k}(t)\in\mathbb{R}^P$, with $P\leq N$, holds the source components over the t dimension and $\boldsymbol{\eta}_{m,k}(t)\in\mathbb{R}^N$ is spatially uncorrelated additive noise, assumed also uncorrelated with $\boldsymbol{s}_{m,k}(t)$. Notice that the mixing matrix is specific to each dataset, but is the same for each dataset along the K observations. This model is an extension to multiple datasets of the typical model found in the BSS literature. It has been proposed already several times [2-5, 8]. It reduces to the very common model used in BSS $\boldsymbol{x}_k(t)=\boldsymbol{A}\boldsymbol{s}_k(t)+\boldsymbol{\eta}_k(t)$ when only one dataset is available. In JBSS we require to find the M matrices $\boldsymbol{B}_m\in\mathbb{R}^{NxP}$, m=1,…,M, yielding the source estimates $\hat{\boldsymbol{s}}_{m,k}(t)=\boldsymbol{B}_m^T\boldsymbol{x}_{m,k}(t)$, where, under mild assumptions on the noise, the demixing matrices $\boldsymbol{B}_m$ estimate the Moore-Penrose pseudo-inverse of the mixing matrices up to a sign, scale and permutation indeterminacy, as in the BSS case. However in JBSS *we require the permutation be the same for all M datasets*, otherwise the analysis of the corresponding sources in the M datasets becomes difficult. This is, among others, a key advantage of the JBSS approach. Notice that for the sake of simplicity we suppose hereafter that N=P, this quantity being the same across datasets.

2.2. The Joint Blind Source Separation (JBSS)

In order to apply JBSS we extract $K[M(M+1)/2]$ matrices of second-order statistics from the K observations of M multivariate datasets [2, 5, 8]. From Eq. (1.1) these matrices follow the theoretical model

$$\boldsymbol{C}_{ij,k} = \boldsymbol{A}_i \boldsymbol{\Lambda}_{ij,k} \boldsymbol{A}_j^T + \boldsymbol{N}_{ij,k}\text{, with } i,j=1,\ldots,M, \qquad (1.2)$$

where the $\boldsymbol{\Lambda}_{ij,k}$ matrices, the unknown source statistics, are supposed all diagonal, surely non-null if $i=j$ (auto-statistics within datasets) and possibly non-null even for $i\neq j$ (cross-statistics between corresponding sources of the i and j datasets). The matrices $\boldsymbol{N}_{ij,k}$ are unknown noise matrices holding additive measurement noise and sample estimation errors. In order to estimate the M demixing matrices we seek matrices making all $K[M(M+1)/2]$ distinct products

$$\boldsymbol{Q}_{ij,k} = \boldsymbol{B}_i^T \boldsymbol{C}_{ij,k} \boldsymbol{B}_j \qquad (1.3)$$

as diagonal as possible. This implies that the output (source) statistics within datasets are diagonalized (for $i=j$), as in the BSS framework. In addition, the output cross-statistics between datasets are also diagonalized, thus corresponding sources across datasets may be correlated. Notice that $\boldsymbol{C}_{ij,k}=\boldsymbol{C}^T_{ji,k}$, thus it suffices to consider $K[M(M+1)/2]$ products instead of all KM^2 products.

2.3. Least-Squares Functional

We want to find matrices $\boldsymbol{B}^T_m$ minimizing the sum of squares of the elements outside the diagonals of all $\boldsymbol{Q}_{ij,k}$, that is

$$\min_{\boldsymbol{B}_1,\ldots,\boldsymbol{B}_M} \sum_{i,j,k} \left\| off\left(\boldsymbol{B}_i^T \boldsymbol{C}_{ij,k} \boldsymbol{B}_j\right)\right\|_F^2,$$

where the *off* operator nullifies the diagonal elements of the matrix argument. The overall strategy is to sequentially search for each matrix $\boldsymbol{B}_i$, for $i=1,\ldots,M$ and iterate such sequential search until convergence. Let us denote with $\boldsymbol{B}$ the set of all demixing matrices. In the sequel, following [3], let us define the functional of interest for any given $i=1,\ldots,M$ as

$$\Psi^{off}_{\boldsymbol{B}_i|\boldsymbol{B}} = 2\sum_{j\neq i}\sum_k \left\|off\left(\boldsymbol{Q}_{ij,k}\right)\right\|_F^2 + \sum_k \left\|off\left(\boldsymbol{Q}_{ii,k}\right)\right\|_F^2, \qquad (1.4)$$

wherein we have used Eq. (1.3) and we have separated the partitions for $i\neq j$ (firt Frobenius norm) and the partition for $i=j$ (second Frobenius norm). Equation (1.4) can also be written such as

$$\Psi^{off}_{\boldsymbol{B}_i|\boldsymbol{B}} = \Psi^{tot-diag}_{\boldsymbol{B}_i|\boldsymbol{B}} = \Psi^{tot}_{\boldsymbol{B}_i|\boldsymbol{B}} - \Psi^{diag}_{\boldsymbol{B}_i|\boldsymbol{B}}, \qquad (1.5)$$

where the "total" and "diagonal" parts are

$$\Psi^{tot}_{\boldsymbol{B}_i|\boldsymbol{B}} = \sum_k \left[\sum_{j\neq i} 2tr\left(\boldsymbol{Q}_{ij,k}\boldsymbol{Q}^T_{ij,k}\right) + tr\left(\boldsymbol{Q}^2_{ii,k}\right)\right] \text{ and} \qquad (1.6)$$

$$\Psi^{diag}_{\boldsymbol{B}_i|\boldsymbol{B}} = \sum_k \left[2\sum_{j\neq i}\sum_n \left(\boldsymbol{b}^T_{i(n)}\boldsymbol{C}_{ij,k}\boldsymbol{b}^T_{j(n)}\right)^2 + \sum_n\left(\boldsymbol{b}^T_{i(n)}\boldsymbol{C}_{ii,k}\boldsymbol{b}^T_{i(n)}\right)^2\right] \qquad (1.7)$$

respectively. In Eq. (1.7), $\boldsymbol{b}_{i(n)}$ is the n^{th} column vector of $\boldsymbol{B}_i$, with n=1,…,N and $\boldsymbol{b}^T_{i(n)}$ its transpose.

2.4. The Orthogonal Mixing Matrices Case

In this case the exact estimation of $\boldsymbol{B}_i$ is equal to $\boldsymbol{A}_i$ up to permutation and scale indeterminacy, for all $m=1,\ldots,M$ and it is well known that the "total" function is invariant with respect to $\boldsymbol{B}$. Thus we are left with the problem of maximizing iteratively the "diag" functional in (1.7), for $i=1,\ldots,M$. Let us rewrite the objective function as

$$\Psi^{diag}_{\boldsymbol{B}_i|\boldsymbol{B}} = \sum_k \left(2\sum_{j\neq i}\left\|diag\left(\boldsymbol{Q}_{ij,k}\right)\right\|_F^2 + \sum_{j\neq i}\left\|diag\left(\boldsymbol{Q}_{ii,k}\right)\right\|_F^2\right),$$

where the *diag* operator nullifies the off-diagonal elements of the matrix argument, and then as

$$\Psi^{diag}_{\boldsymbol{B}_i|\boldsymbol{B}} = \sum_k \begin{bmatrix} 2\sum_{j\neq i}\left(tr\sum_n\left(\boldsymbol{E}_n\boldsymbol{Q}_{ij,k}\boldsymbol{E}_n\boldsymbol{E}_n\boldsymbol{Q}_{ji,k}\boldsymbol{E}_n\right)\right) \\ +tr\sum_n\left(\boldsymbol{E}_n\boldsymbol{Q}_{ii,k}\boldsymbol{E}_n\boldsymbol{E}_n\boldsymbol{Q}_{ii,k}\boldsymbol{E}_n\right)\end{bmatrix},$$

where matrix $\boldsymbol{E}_n$ is the elementary matrix filled with entry 1 at position (n,n) and 0 elsewhere. The above function is a matrix polynomial of second degree in $\boldsymbol{B}_i$. The derivative is of first degree in $\boldsymbol{B}_i$ for the first trace and of third degree in $\boldsymbol{B}_i$ for the second trace. However, using the symmetry of matrices $\boldsymbol{C}_{ii,k}$, the gradient simplifies to

$$\frac{\partial\Psi^{diag}_{\boldsymbol{B}_i|\boldsymbol{B}}}{\partial\boldsymbol{B}_i} = \sum_k \begin{bmatrix} 4\sum_{j\neq i}\sum_n\left(\boldsymbol{C}_{ij,k}\boldsymbol{B}_j\boldsymbol{E}_n\boldsymbol{E}_n\boldsymbol{Q}_{ji,k}\boldsymbol{E}_n\right) \\ +4\sum_n\left(\boldsymbol{C}_{ii,k}\boldsymbol{B}_i\boldsymbol{E}_n\boldsymbol{E}_n\boldsymbol{Q}_{ii,k}\boldsymbol{E}_n\right)\end{bmatrix}.$$

Thus $$\frac{\partial\Psi^{diag}_{\boldsymbol{B}_i|\boldsymbol{B}}}{\partial\boldsymbol{B}_i} = 4\left[\boldsymbol{M}_{i(1)}\boldsymbol{b}_{i(1)},\ldots,\boldsymbol{M}_{i(N)}\boldsymbol{b}_{i(N)}\right], \qquad (1.8)$$

where $$\boldsymbol{M}_{i(n)} = \sum_k\sum_j\left(\boldsymbol{C}_{ij,k}\boldsymbol{b}_{j(n)}\boldsymbol{b}^T_{j(n)}\boldsymbol{C}^T_{ij,k}\right). \qquad (1.9)$$

In words, the gradient of the N vectors of $\boldsymbol{B}_i$ should be taken as the eigenvectors of corresponding matrices $\boldsymbol{M}_{i(n)}$ associated with their largest eigenvalue, for all $n=1,\ldots,N$. In order to update these vectors we limit ourselves to a single power iteration [6-7]. After updating all vectors of $\boldsymbol{B}_i$ we need to orthogonalize $\boldsymbol{B}_i$ so as to ensure that at each step $\boldsymbol{B}_i$ stays in the orthogonal group [5]. Therefore, we have the following simple updating rule:

for all i=1,…,M do $\begin{cases} \boldsymbol{B}_i \leftarrow \left[\boldsymbol{M}_{i(1)}\boldsymbol{b}_{i(1)},...,\boldsymbol{M}_{i(N)}\boldsymbol{b}_{i(N)} \right] \\ orthogonalize\ \boldsymbol{B}_i \end{cases}$.

The iterative algorithm is summarized here below:

> **Algorithm OJoB (Orthogonal Joint BSS)**
> Initialize $\boldsymbol{B}_1,\ldots,\boldsymbol{B}_M$ as orthogonal matrices (e.g., identity)
> Repeat
> For i=1,…,M do
> Compute the N matrices $\boldsymbol{M}_{i(n)}$ using (1.9)
> For n=1,…,N do $\boldsymbol{b}_{i(n)} \leftarrow \boldsymbol{M}_{i(n)}\, \boldsymbol{b}_{i(n)}$
> $\boldsymbol{B}_i \leftarrow (\boldsymbol{B}_i^T\boldsymbol{B}_i)^{-1/2}\boldsymbol{B}_i$
> Until Convergence

Note that in practice the orthogonalization is computed faster as $\boldsymbol{B}_i \leftarrow \boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{B}_i$ [5].

2.5. The Non-Orthogonal Mixing Matrices Case

In this case the "total" function is not invariant to $\boldsymbol{B}$, hence we need to explicitly minimize the "off" functional in (1.4). Furthermore we need to avoid the trivial solution $\boldsymbol{B}^T{}_m=\boldsymbol{0}$, for any m=1,…,M. Therefore, we minimize the "off" functional in (1.5)-(1.7) with a constraint (w.c.) on the norm of the column vectors of $\boldsymbol{B}_m$, such as

$$\Psi^{tot-diag}_{\boldsymbol{B}_i|\boldsymbol{B}}, \text{w.c. } \boldsymbol{b}^T_{i(n)}\boldsymbol{M}_{i(n)}\boldsymbol{b}_{i(n)} = 1,\ \forall n = 1,...,N \,, \quad (1.10)$$

where matrices $\boldsymbol{M}_{i(n)}$ are given in (1.9). Note that this is an "intrinsic" constraint, as proposed first in [6-7]. We use the method of Lagrange multipliers to turn (1.10) into an unconstrained optimization problem. The method leads us to minimize

$$L\left(\Psi^{tot-diag}_{\boldsymbol{B}_i|\boldsymbol{B}}\right) = \sum_k \left[2\sum_{j\neq i} tr\left(\boldsymbol{Q}_{ij,k}\boldsymbol{Q}_{ji,k}\right) + tr\left(\boldsymbol{Q}^2_{ij,k}\right)\right] -$$
$$\sum_k \left[4\sum_{j\neq i}\sum_n \delta_n\left(\boldsymbol{C}_{ij,k}\boldsymbol{B}_j\boldsymbol{E}^2_n\boldsymbol{Q}_{ji,k}\boldsymbol{E}_n\right) + 4\sum_n \delta_n\left(\boldsymbol{C}_{ii,k}\boldsymbol{B}_i\boldsymbol{E}^2_n\boldsymbol{Q}_{ii,k}\boldsymbol{E}_n\right)\right]$$

where the Lagrangian multipliers δ_n are adjusted in order to satisfy the constraint. Using again the symmetry of the matrices $\boldsymbol{C}_{ii,k}$ and exploiting the previous results in (1.8), the gradient of the Lagrangian reads

$$\frac{\partial L\left(\Psi^{tot-diag}_{\boldsymbol{B}_i|\boldsymbol{B}}\right)}{\partial \boldsymbol{B}_i} = \sum_k \left[4\sum_{j\neq i}\left(\boldsymbol{C}_{ij,k}\boldsymbol{B}_j\boldsymbol{Q}_{ij,k}\right) + 4\left(\boldsymbol{C}_{ii,k}\boldsymbol{B}_i\boldsymbol{Q}_{ii,k}\right)\right] - \frac{\partial \Psi^{diag}_{\boldsymbol{B}_i|\boldsymbol{B}}}{\partial \boldsymbol{B}_i}$$

or $\dfrac{\partial L\left(\Psi^{tot-diag}_{\boldsymbol{B}_i|\boldsymbol{B}}\right)}{\partial \boldsymbol{B}_i} = 4\boldsymbol{M}_i\boldsymbol{B}_i - 4\left[\boldsymbol{M}_{i(1)}\boldsymbol{b}_{i(1)},...,\boldsymbol{M}_{i(N)}\boldsymbol{b}_{i(N)}\right]$,

where $\boldsymbol{M}_i = \sum_k\sum_j\left(\boldsymbol{C}_{ij,k}\boldsymbol{B}_j\boldsymbol{B}^T_j\boldsymbol{C}^T_{ij,k}\right) = \sum_n \boldsymbol{M}_{i(n)}$,

that is, the sum of matrices already found in (1.9). Setting the gradient to zero gives us a stationary point for our optimization, which for each $\boldsymbol{B}_i$ is given by $\boldsymbol{M}_i\boldsymbol{B}_i$=[$\boldsymbol{M}_{i(1)}\boldsymbol{b}_{i(1)}$,…,$\boldsymbol{M}_{i(n)}\boldsymbol{b}_{i(n)}$]. This is a generalized eigenvalue-eigenvector problem, where each $\boldsymbol{b}_{i(n)}$ is the eigenvector of $\boldsymbol{M}_{i(n)}$ in the metric of $\boldsymbol{M}_i$. Again, limiting ourselves to one single power iteration per update step, this yields the simple updating rule:

For i=1,…,M do $\begin{cases} \boldsymbol{B}_i \leftarrow \boldsymbol{M}_i^{-1}\left[\boldsymbol{M}_{i(1)}\boldsymbol{b}_{i(1)},...,\boldsymbol{M}_{i(N)}\boldsymbol{b}_{i(N)}\right] \\ \text{for } n=1,...,N \text{ do} \\ \qquad \boldsymbol{b}_{i(n)} \leftarrow \boldsymbol{b}_{i(n)}\left(\boldsymbol{b}^T_{i(n)}\boldsymbol{M}_{i(n)}\boldsymbol{b}_{i(n)}\right)^{-1/2} \end{cases}$.

In practice, we sought to avoid the computation of the matrix inverse, therefore we proceed more efficiently, albeit equivalently, in the following way :

> **Algorithm NOJoB (Non-Orthogonal Joint BSS)**
> Initialize $\boldsymbol{B}_1,\ldots,\boldsymbol{B}_M$ so as to satisfy constraint in (1.10)
> Repeat
> For i=1,…,M do
> Get the N matrices $\boldsymbol{M}_{i(n)}$ using (1.9) and their sum $\boldsymbol{M}_i$
> Do Cholesky decomposition $\boldsymbol{M}_i=\boldsymbol{L}\boldsymbol{L}^T$.
> For n=1,…,N do
> Solve $\boldsymbol{Lx}=\boldsymbol{M}_{i(n)}\boldsymbol{b}_{i(n)}$ for $\boldsymbol{x}$ and $\boldsymbol{L}^T\boldsymbol{y}=\boldsymbol{x}$ for $\boldsymbol{y}$
> $\boldsymbol{b}_{i(n)} \leftarrow \boldsymbol{y}(\boldsymbol{y}^T\boldsymbol{M}_{i(n)}\,\boldsymbol{y})^{-1/2}$
> Until Convergence

3. RESULTS

In this section the behavior of the proposed algorithms is assessed by means of simulations. Input matrices $\boldsymbol{C}_{ij,k}$ are generated according to the model in Eq.(1.2); matrices $\boldsymbol{\Lambda}_{ij,k}$ are generated as square diagonal matrices with each diagonal entry randomly distributed as a chi-squared with M degrees of freedom and divided by M; noise matrices $\boldsymbol{N}_{ij,k}$ are symmetric and have entries randomly Gaussian distributed with zero mean and σ standard deviation (sd). The parameter σ controls the signal to noise ratio of the input matrices. Several different values of σ will be considered in the simulations. The mixing matrices $\boldsymbol{A}_m$, m=1,…,M, are generated as orthogonal or non-orthogonal. Orthogonal matrices are generated by first generating a matrix with entries randomly drawn from a Gaussian distribution with zero mean and sd=1 and then taking its left singular vector matrix. In this case the conditioning of the mixing matrices does not jeopardize the performance of the algorithms and we can evaluate their robustness with respect to noise. In order to generate non-orthogonal matrices, the matrices generated as above are perturbed by adding to each entry a number randomly drawn from a Gaussian distribution with zero mean and sd=1/2. In this case the mixing matrices have variable conditioning and we can evaluate the behavior of the algorithms with respect to the conditioning of the mixing matrices.

The algorithms estimate the demixing matrices $\boldsymbol{B}^T_1,\ldots,\boldsymbol{B}^T_M$, which should approximate the pseudo-inverse of actual mixing matrices $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_M$ up to row scaling (including sign) and global permutation. Then, matrices $\mathbf{G}^m = \boldsymbol{B}^T_m\boldsymbol{A}_m$ should approximate as much as possible a scaled permutation matrix [1]. For each estimated demixing matrix we consider the Amari-like performance index [1], which is computed as

$$\pi_m = \sum_x \left[\frac{\sum_y \left|g^m_{xy}\right|}{\max_y \left|g^m_{xy}\right|} - 1 \right] + \sum_y \left[\frac{\sum_x \left|g^m_{xy}\right|}{\max_x \left|g^m_{xy}\right|} - 1 \right] \Big/ 2N\left(N-1\right),$$

where indexes x and y run over $1,\ldots,N$ (rows and columns of matrices $\boldsymbol{G}^m$), g^m_{xy} is the (x,y) entry of matrix $\boldsymbol{G}^m$, and $|.|$ denotes the absolute value of the argument. We define the composite performance as a function of the geometric mean of the performance indexes obtained over the M matrices, such as

$$\pi = -\log_{10}\left(1 - \sqrt[M]{\prod\nolimits_m \left(1-\pi_m\right)}\right). \quad (1.11)$$

Values of π above 2 indicate a very good performance. Note that the composite performance defined this way is dominated by the worst performance over the M performances. The higher the value of the composite index (1.11), the higher the performance. Likewise, the composite conditioning with respect to matrix inverse of the mixing matrices is defined as

$$\chi = \log_{10}\left(\prod\nolimits_m \left|\max eig\left(\boldsymbol{A}_m\right) / \min eig\left(\boldsymbol{A}_m\right)\right|\right), \quad (1.12)$$

where max*eig* and min*eig* are the largest and smallest eigenvalue of the argument, respectively. Figure 1 shows the performance (1.11) obtained by the OJoB and NoJoB algorithms with orthogonal mixing (input) matrices, N=P=3 sensors/sources and several combinations of M (number of datasets), K (number of observations), and σ (noise level). One hundred simulations have been performed for each algorithm and for each combination of M, K and σ. Each dot represent the intersection of the performance obtained in one simulation when the algorithm is initialized with identity matrices (x-axis), that is, possibly far from the optimal solution, and with the exact solutions (y-axis). Dots lying on the 45° line indicate that the algorithms have a stable attractor, despite the added noise. Dots lying above the 45° line indicate that the algorithm gets far from the exact solution. These results show that for both algorithms the degradation engendered by noise is mitigated by increasing either K or M. Also, when either M or K are much larger than N no divergence of the algorithms is noticed. Overall, NoJoB appears more stable than OJoB.

Figure 2 shows the performance of the NoJoB algorithm (y-axis) vs. the mixing matrices conditioning (1.12) with non-orthogonal mixing (input) matrices, N=P=3 and several combinations of M, K, and σ. Results show that the more noise there is in the system the more the conditioning affects the performance. Furthermore, the degradation is not mitigated by increasing K and only moderately mitigated by increasing M.

4. CONCLUSION

We have presented two new algorithms performing JBSS in a least-squares framework. Preliminary results encourage further investigation. Before NoJoB, only one non-orthogonal JBSS algorithm has been proposed [8].

5. AKNOWLEDGEMENTS

This research has been partially funded by the Grenoble Institute of Technology with a BQR (Bonus Qualité Recherche) and by the ANR (Agence Nationale de la Recherche), through projects Gaze&EEG, RoBIK and OpenViBE2.

6. REFERENCES

[1] P. Comon, C. Jutten, Handbook of Blind source separation. Independent component analysis and applications, Academic Press, 2010.
[2] J. Vía, M. Anderson, X.-L. Li, T. Adali, "Joint blind source separation from second-order statistics: Necessary and sufficient identifiability conditions," ICASSP 2011, 2011, pp. 2520-23.
[3] X.-L. Li, T. Adali, M. Anderson, "Joint blind source separation by generalized joint diagonalization of cumulant matrices," Signal Process., vol. 91(10), 2011, pp. 2314-2322.
[4] Y.-O. Li, T. Adali, W. Wang, and V. D. Calhoun, "Joint blind source separation by multi-set canonical correlation analysis," IEEE Trans. Signal Process., vol. 57, no. 10, 2009, pp. 3918-29.
[5] M. Congedo, R. Phlypo, D.-T. Pham, "Approximate joint singular value decomposition of an asymmetric rectangular matrix set," IEEE Trans. Signal Process., vol. 59(1), 2011, pp. 415-424.
[6] M. Congedo, D.-T. Pham, "Least-squares joint diagonalization of a matrix set by a congruence transformation," SinFra 2009, 2009.
[7] D.-T. Pham, M. Congedo, "Least square joint diagonalization of matrices under an intrinsic scale constraint," ICA 2009, 2009, pp. 298-305.
[8] M. Anderson, T. Adali, X. Li, "Joint Blind Source Separation With Multivariate Gaussian Model: Algorithms and Performance Analysis," IEEE Trans. Signal Process., vol. 60(4), 2012, pp. 1672-1683.

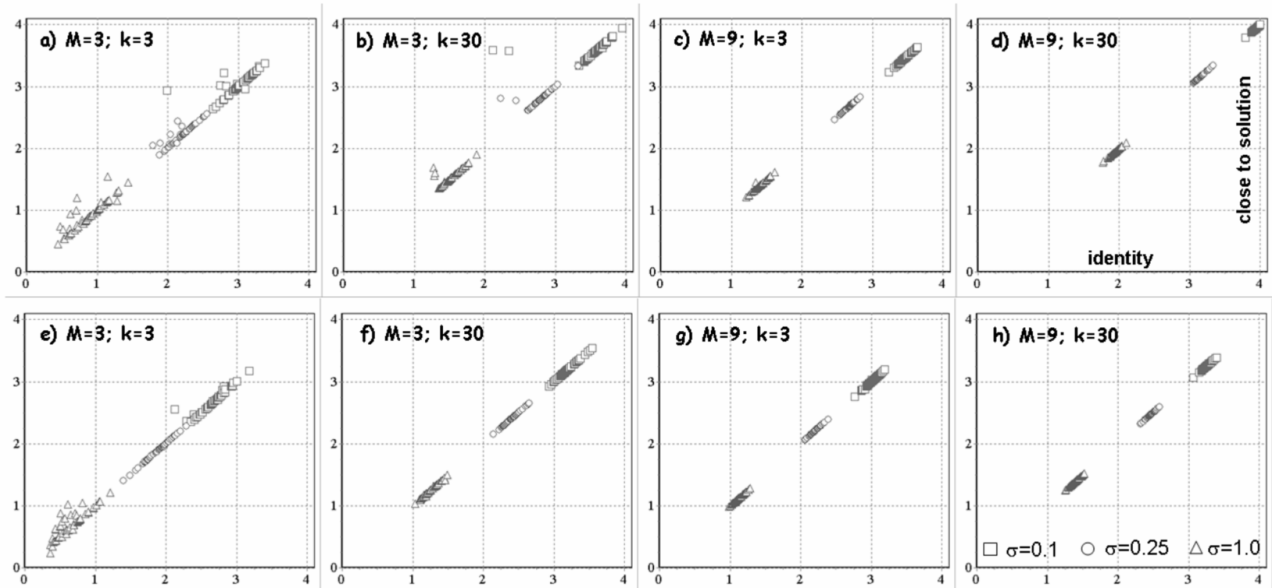

Figure 1. Composite performance of the OJoB (top row: a to d) and NOJoB (bottom row: e to h) algorithms (1.11) when initialized with the identity matrices (x-axis) vs. when initialized with the inverse of the actual mixing matrices (y-axis), for N=P=3, three noise levels (σ) and several combinations of M and K.

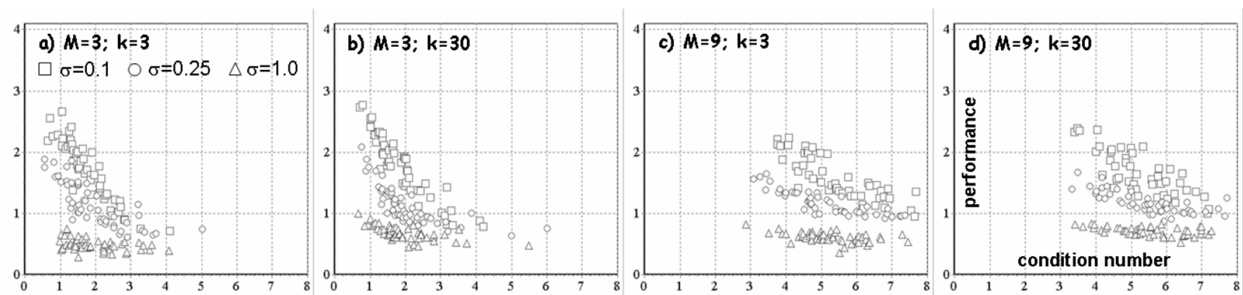

Figure 2. Composite performance of the NOJoB algorithm (y-axis) vs. composite condition number (1.12) of mixing matrices. Same parameters as Fig. 1.