

Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold

Fabian J. Theis¹, Thomas P. Cason², and P.-A. Absil²

¹ CMB, Institute of Bioinformatics and Systems Biology,
Helmholtz Zentrum München, Germany, and
MPI for Dynamics and Self-Organization, Göttingen, Germany
fabian.theis@helmholtz-muenchen.de, <http://cmb.helmholtz-muenchen.de>

² Department of Mathematical Engineering, Université catholique de Louvain,
B-1348 Louvain-la-Neuve, Belgium (<http://www.inma.ucl.ac.be/~{cason,absil}>)

Abstract. Joint diagonalization for ICA is often performed on the orthogonal group after a pre-whitening step. Here we assume that we only want to extract a few sources after pre-whitening, and hence work on the Stiefel manifold of p -frames in \mathbb{R}^n . The resulting method does not only use second-order statistics to estimate the dimension reduction and is therefore denoted as soft dimension reduction. We employ a trust-region method for minimizing the cost function on the Stiefel manifold. Applications to a toy example and functional MRI data show a higher numerical efficiency, especially when p is much smaller than n , and more robust performance in the presence of strong noise than methods based on pre-whitening.

1 Introduction

The common approach to blind source separation of a set of multivariate data is to first whiten the data and to then search on the more restricted orthogonal group. This has two advantages: (i) instead of optimizing a cost function on n^2 , the optimization takes place on a $n(n-1)/2$ dimensional manifold. (ii) During whitening via PCA, the dimension can already be reduced. However a serious drawback of this sometimes called *hard-whitening* technique is that the resulting method is biased towards the data correlation (which is used in the PCA step). Using the empirical correlation estimator, the method perfectly trusts the correlation estimate, whereas it ‘mistrusts’ any later sample estimates.

In contrast to hard-whitening, *soft-whitening* tries to avoid the bias towards second-order statistics. In algorithms based on joint diagonalization (JD) of a set of source conditions, reviewed for example in [1], this implies using a non-orthogonal JD algorithm [2–5]. It jointly diagonalizes both the source conditions together with the mixture covariance matrix. Then possible estimation errors in the second-order part do not influence the total error disproportionately high.

Soft-whitening essentially does away with issue (i). In this contribution, we propose a method to deal with issue (ii) instead: The above argument of bias towards correlation with respect to source estimation also applies to the bias

with respect to dimension reduction. This can be solved by a subspace extraction algorithm followed by an ICA algorithm, see e.g. [6, 7], which however may lead to an accumulation of errors in the two-step procedure. Hence, we propose the following integrated solution implementing a *soft dimension reduction*: We will first whiten the data, so we will assume (i) and hard-whitening. However we do not reduce the data dimension beforehand. Instead we propose to search for a non-square pseudo-orthogonal matrix (Stiefel matrix) such that it minimizes the JD cost function. An efficient minimization procedure will be proposed in the following. Examples to speech and fMRI data confirm the applicability of the method. In future research a combination of (i) soft-whitening and (ii) soft dimension reduction may be attractive.

2 Joint Diagonalization on the Stiefel manifold

Let $\text{St}(p, n) = \{Y \in \mathbb{R}^{n \times p} : Y^T Y = I\}$ denote the Stiefel manifold of orthogonal p -frames in \mathbb{R}^n for some $p \leq n$. The JD problem on the Stiefel manifold consists of minimizing the cost function

$$f_{\text{diag}} : \text{St}(p, n) \rightarrow \mathbb{R} : Y \mapsto f_{\text{diag}}(Y) = - \sum_{i=1}^N \|\text{diag}(Y^T C_i Y)\|_F^2, \quad (1)$$

where $\|\text{diag}(X)\|_F^2$ returns the sum of the squared diagonal elements of X . In the context of ICA, the matrices C_i can for example be cumulant matrices (as in the JADE algorithm [8]) or time-lagged covariance matrices (as in SOBI [9]).

2.1 Diagonal maximization versus off-diagonal minimization

In the case $p = n$, minimizing f_{diag} is equivalent to minimizing the sum of the squared off-diagonal elements

$$f_{\text{off}} : \text{St}(p, n) \rightarrow \mathbb{R} : Y \mapsto f_{\text{off}}(Y) = \sum_i \|\text{off}(Y^T C_i Y)\|_F^2.$$

Indeed, $\|\text{off}(Y^T C_i Y)\|_F^2 = \|Y^T C_i Y\|_F^2 - \|\text{diag}(Y^T C_i Y)\|_F^2$ and $\|Y^T C_i Y\|_F^2$ does not depend on $Y \in \text{St}(n, n) = \text{O}(n)$. When $p < n$, we can still observe that if Y_* minimizes f_{diag} , then it minimizes f_{off} over $\{Y_* Q : Q \in \text{O}(p)\} \subset \text{St}(p, n)$; this follows from the same argument applied to the function $Q \mapsto f_{\text{diag}}(Y_* Q)$. Note that minimizing f_{off} is clearly not sufficient for minimizing f_{diag} . As an illustration, consider the case $N = 1$, i.e., there is only one target matrix, C , assumed to be symmetric positive definite with distinct eigenvalues. Then the minimizers of f_{off} are all the matrices $Y \in \text{St}(p, n)$ such that $Y^T C Y$ is diagonal (when $p < n$, there are infinitely many such Y), whereas the optimizers of f_{diag} are $Y_* = [v_1 \dots v_p] \pi$, where v_1, \dots, v_p are the p dominant eigenvectors of C and π denotes any signed permutation matrix.

2.2 A trust-region method for minimizing f_{diag}

Minimizing f_{diag} is an optimization problem over the Stiefel manifold. Recently, several methods have been proposed to tackle optimization problems on manifolds; see, e.g., [10,11] and references therein. In this paper, we use a trust-region approach, which combines favorable global and local convergence properties with a low numerical cost.

In \mathbb{R}^n , trust-region methods proceed as follows. At the current iterate x_k , a model m_{x_k} is chosen to approximate a cost function f . The model is “trusted” within a ball of radius Δ_k around x_k , termed the trust region. A candidate for x_{k+1} is selected as an (approximate) solution of the trust-region subproblem, namely, the minimization of m_{x_k} under the trust-region constraint. The new iterate is accepted and the trust-region radius Δ_k updated according to the agreement between the values of f and m_{x_k} at the candidate. We refer to [12] for more information.

The concept of trust-region was generalized to Riemannian manifolds in [13]. On a manifold \mathcal{M} , the trust-region subproblem at x_k becomes a subproblem on the tangent space $T_{x_k}\mathcal{M}$. Since the tangent space is a linear space, the classical techniques for solving trust-region subproblems apply as well. The correspondence between the tangent spaces and the manifold \mathcal{M} is specified by a mapping R , called retraction, that is left to the user’s discretion but for some rather lenient requirements. The retraction makes it possible to pull back the cost function f on \mathcal{M} to a cost function $\hat{f}_{x_k} = f \circ R_{x_k}$ on $T_{x_k}\mathcal{M}$, where R_{x_k} denotes the restriction of R to $T_{x_k}\mathcal{M}$.

More specifically, the Riemannian trust-region method proceeds along the following steps.

1. Consider the local approximation of the pulled back cost function \hat{f}_{x_k}

$$m_{x_k}(\xi) = f(x_k) + \langle \xi, \text{grad}f(x_k) \rangle + \frac{1}{2} \langle \xi, \text{Hess}f(x_k) [\xi] \rangle,$$

where $\text{grad}f$ and $\text{Hess}f$ stand for the gradient and the Hessian of f , respectively, and obtain ξ_k by (approximately) solving

$$\min_{\xi \in T_{x_k}\mathcal{M}} m_{x_k}(\xi) \quad \text{s.t.} \quad \|\xi_k\| \leq \Delta_k,$$

where Δ_k denotes the radius.

2. Evaluate the quality of the model m_{x_k} through the quotient

$$\rho_k = \frac{\hat{f}_{x_k}(0) - \hat{f}_{x_k}(\xi_k)}{m_{x_k}(0) - m_{x_k}(\xi_k)}.$$

If ρ_k is exceedingly small, then the model is very inaccurate, the trust-region radius is reduced and $x_{k+1} := x_k$. If ρ_k is small but less dramatically so, then $Y_{k+1} = R_{x_k}(\xi_k)$ but the trust-region radius is reduced. Finally, if ρ_k is close to 1, then there is good agreement between the model and the function, and the trust-region radius can be expanded.

Note that trust-region algorithms (such as steepest-descent, Newton, and conjugate gradient algorithms) are *local methods*: they efficiently exploit information from the derivatives of the cost function, but they are not guaranteed to find the global minimizer of the cost function. (This is not dishonorable, as computing the global minimizer is a very hard problem in general.) Nevertheless, they can be shown to converge *globally* (i.e., for every initial point) to stationary points of the cost function; moreover, since they are descent methods, they only converge to minimizers (local or global), except in maliciously-crafted cases. More details on the Riemannian trust-region method can be found in [11, 13]. We also refer to [14] for recent developments.

2.3 Implementing the Riemannian trust-region method

A generic MATLAB code for the Riemannian trust-region method can be obtained from <http://www.scs.fsu.edu/~cbaker/GenRTR/>. The optimization method utilizes MATLAB function handles to access user-provided routines for the objective function, gradient, Hessian, retraction, etc. This allows the encapsulation of a problem into a single driver. In the remainder of this section, we describe the essential elements of the driver that we have created for minimizing f_{diag} (1).

The driver must contain a routine that returns the inner product of two vectors of $T_Y \text{St}(p, n)$, so as to specify the Riemannian structure of $\text{St}(p, n)$. We choose $\langle \xi_1, \xi_2 \rangle = \text{tr}(\xi_1^T \xi_2)$, which makes $\text{St}(p, n)$ a Riemannian submanifold of $\mathbb{R}^{n \times p}$. The retraction is chosen as

$$R_Y : T_Y \text{St}(p, n) \rightarrow \text{St}(p, n) : \xi \mapsto R_Y \xi := \text{qf}(Y + \xi)$$

where $\text{qf}(Y)$ denotes the Q factor of the QR decomposition of Y . We further need a formula for the gradient of f_{diag} . We have $\text{grad} f_{\text{diag}}(Y) = P_Y \text{grad} \hat{f}_{\text{diag}}(Y)$, where $\text{grad} \hat{f}_{\text{diag}}(Y) = -\sum_i 4C_i Y \text{ddiag}(Y^T C_i Y)$ is the gradient of

$$\hat{f}_{\text{diag}} : \mathbb{R}^{n \times p} \rightarrow \mathbb{R} : Y \mapsto \hat{f}_{\text{diag}}(Y) = -\sum_i \|\text{diag}(Y^T C_i Y)\|_F^2,$$

and where P_Y denotes the orthogonal projection onto $T_Y \text{St}(p, n)$ i.e. $P_Y \xi := \xi - Y \text{sym}(Y^T \xi)$. Finally, the Hessian of f_{diag} is given by

$$\text{Hess} f(Y) [\xi] = \nabla_{\xi} \text{grad} f(Y)$$

where ∇ is the Riemannian connection on \mathcal{M} (see [11, Section 5.3.2]). In our case we choose $\nabla_{\eta} \xi := P_Y (\text{D}\xi(Y)[\eta])$ where Y denotes the foot of η . Therefore, the Hessian of f_{diag} is given by

$$\text{Hess} f_{\text{diag}}(Y) [\xi] = P_Y \text{Dgrad} \hat{f}_{\text{diag}}(Y) [\xi] - \xi \text{sym}(Y^T \text{grad} \hat{f}_{\text{diag}}(Y))$$

where $\text{Dgrad} \hat{f}_{\text{diag}}(Y)$ is

$$\text{Dgrad} \hat{f}_{\text{diag}}(Y) [\xi] = -\sum_i 4C_i \begin{pmatrix} \xi \text{ddiag}(Y^T C_i Y) \\ +Y \text{ddiag}(\xi^T C_i Y) \\ +Y \text{ddiag}(Y^T C_i \xi) \end{pmatrix}.$$

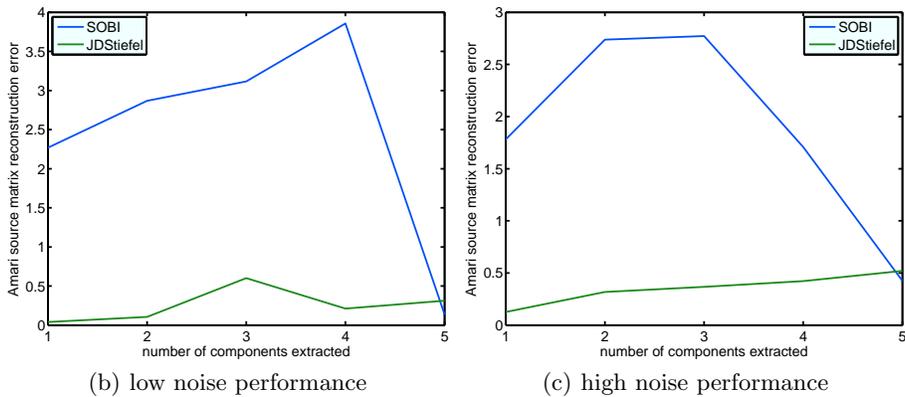
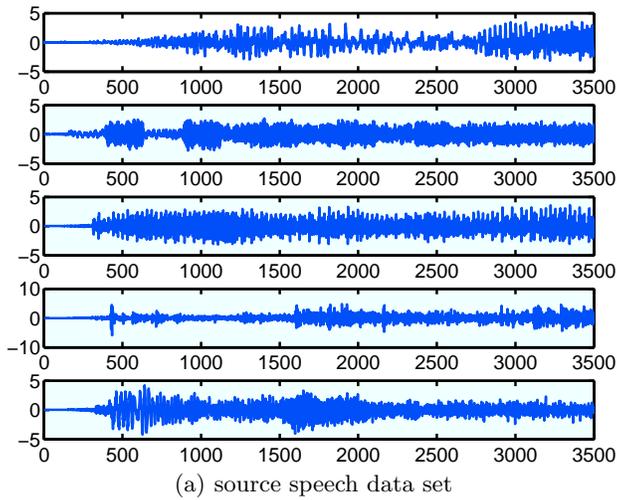


Fig. 1. Application of the algorithm to speech data:

3 Simulations

We propose two applications of the above ‘JD Stiefel’ algorithm in the following. As source conditions, we will follow the SOBI algorithm [9] and calculate lagged auto-covariance matrices.

3.1 Artificial data

In the first example, we apply the JD Stiefel algorithm to the SOBI cost function in order to separate artificially mixed speech data. $n = 5$ speech sources with 3500 samples were chosen³. These were embedded using a matrix chosen with

³ Data set ‘acspeech16’ from ICALAB <http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/benchmarks/>

independent normal random elements into a $m = 10$ -dimensional mixture space. White Gaussian noise was added with varying strength.

Algorithmically, the noisy mixture data was first whitened. Then either SOBI [9] with dimension reduction was applied or the JD Stiefel algorithm. For both algorithms, $N = 20$ lagged autocovariance matrices were calculated (with lags 2, 4, 6, ..., 40). We are interested in the performance of the algorithm when recovering parts of the mixing matrix \mathbf{A} . We only recover part of the data in order to simulate the situation of larger dimension than source dimension of interest. This is realized by extracting only a $n' \leq n$ dimensional subspace, either by PCA and SOBI or by the non-squared JD Stiefel algorithm.

In order to measure deviation from perfect recovery, given a projection matrix \mathbf{W} , we want the resulting matrix \mathbf{WA} to have only one large number per row. This is measured by Amari's performance index [15] $E(\mathbf{WA})$ generalized to non-square matrices:

$$E(\mathbf{C}) = \sum_{i=1}^{n'} \left(\sum_{j=1}^n \frac{|c_{ij}|}{\max_k |c_{ik}|} - 1 \right)$$

In figure 1(b,c), we show the results for a low and a high noise setting with signal-to-noise ratios of 32.4dB and 4.65dB, respectively. Clearly the JD Stiefel algorithm is able to take advantage of the full dimensionality of the data when searching the correct subspace, so it always considerably outperforms the SOBI algorithm, which only operates on the PCA-dimension-reduced data. Moreover, we see that even in the case of low signal-to-noise ratio (SNR), the JD Stiefel algorithm performs satisfactorily well.

3.2 Recording from functional MRI

Functional magnetic-resonance imaging (fMRI) can be used to measure brain activity. Multiple MRI scans are taken in various functional conditions; the extracted task-related component reveals information about the task-activated brain regions. Classical power-based methods fail to blindly recover the task-related component as it is very small with respect to the total signal, usually around one percent in terms of variance. Hence we propose to use the autocovariance structure (in this case spatial autocovariances) in combination with the soft dimension reduction to properly identify the task component.

fMRI data with 98 images ($TR/TE = 3000/60$ msec) were acquired with five periods of rest and five photic stimulation periods with rest. Simulation and rest periods comprised 10 repetitions each, i.e. 30s. Resolution was $3 \times 3 \times 4$ mm. The slices were oriented parallel to the calcarine fissure. Photic stimulation was performed using an 8 Hz alternating checkerboard stimulus with a central fixation point and a dark background with a central fixation point during the control periods [16]. The first scans were discarded for remaining saturation effects. Motion artifacts were compensated by automatic image alignment. In order to speed up computation, we reduce the 98 slices to 10 slices by PCA.

As before, in order to compare the performance of JD Stiefel versus SOBI after PCA (with $N = 100$ lagged autocovariance matrices), we analyze how well

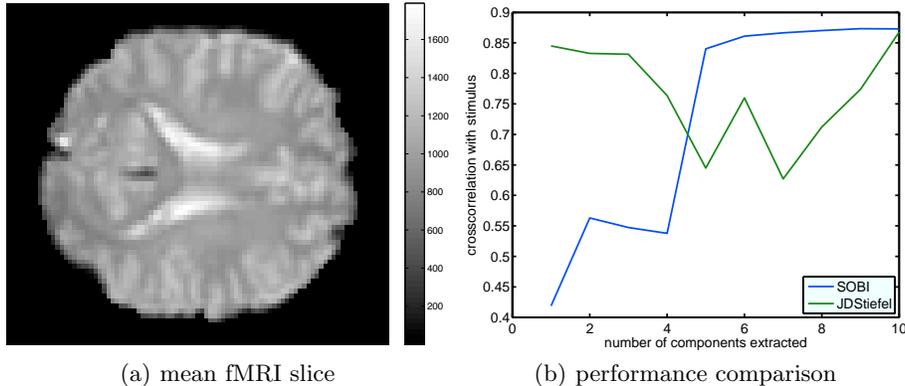


Fig. 2. Application of the algorithm to data acquired from functional MRI: (a) shows the temporal mean of the $128 \times 128 \times 98$ data set. (b) shows the comparison of JD Stiefel and PCA+SOBI algorithm when recovering the task-related component.

the task-related component with the known task vector $\mathbf{v} \in \{0, 1\}^{98}$ is contained in a component by the maximal crosscorrelation of all columns of \mathbf{A} with \mathbf{v} .

We compare the two algorithms for dimension reductions $n' \in \{1, \dots, 10\}$ in figure 2. The key result is that JD Stiefel already detects the main task component when reducing to only one dimension (crosscorrelation larger than 80%). SOBI is only able to find this when having access to at least 5 dimensions. Then SOBI outperforms JD Stiefel, which is prone to fall in local minima during its search. Multiple restarts and more extended searches should resolve this issue, as the cost functions coincide if $n = m$. More complex analyses of fMRI data using dimension reduction can now be approached, as generalization of e.g. [17].

4 Conclusions

Instead of reducing the dimension of the data and searching for independent components in two distinct steps, we have proposed a two-in-one approach which consists of optimizing the JD cost function on a Stiefel manifold. Numerical experiments on artificially mixed toy and fMRI data are promising. In analogy to soft-whitening, where the correlation estimate is weighed equally with respect to higher-order moments of the data, the proposed method implements a soft dimension reduction strategy, by using both second- and higher-order information of the data. In future work, we propose merging soft-whitening and soft dimension reduction.

Acknowledgements

Partial financial support by the Helmholtz Alliance on Systems Biology (project ‘CoReNe’) is gratefully acknowledged. This paper presents research results of

the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

References

1. Theis, F., Inouye, Y.: On the use of joint diagonalization in blind signal processing. In: Proc. ISCAS 2006, Kos, Greece (2006)
2. Yeredor, A.: Non-orthogonal joint diagonalization in the leastsquares sense with application in blind source separation. *IEEE Trans. Signal Processing* **50**(7) (2002) 1545–1553
3. Ziehe, A., Laskov, P., Mueller, K.R., Nolte, G.: A linear least-squares algorithm for joint diagonalization. In: Proc. of ICA 2003, Nara, Japan (2003) 469–474
4. Pham, D.: Joint approximate diagonalization of positive definite matrices. *SIAM Journal on Matrix Anal. and Appl.* **22**(4) (2001) 1136–1152
5. Absil, P.A., Gallivan, K.A.: Joint diagonalization on the oblique manifold for independent component analysis. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Volume 5. (2006) V–945–V–948
6. Blanchard, G., Kawanabe, M., Sugiyama, M., Spokoiny, V., Müller, K.: In search of non-gaussian components of a high-dimensional distribution. *Journal of Machine Learning Research* **7** (2006) 247–282
7. Kawanabe, M., Theis, F.: Joint low-rank approximation for extracting non-gaussian subspaces. *Signal Processing* (2007)
8. Cardoso, J.F.: High-order constraints for independent component analysis. *Neural Computation* **11**(1) (January 1999) 157–192
9. Belouchrani, A., Abed-Meraim, K., Cardoso, J.F., Moulines, E.: A blind source separation technique using secton-order statistics. *IEEE Trans. Signal Process.* **45**(2) (1997) 434–444
10. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.* **20**(2) (1998) 303–353
11. Absil, P.A., Mahony, R., Sepulchre, R.: *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ (January 2008)
12. Conn, A.R., Gould, N.I.M., Toint, P.L.: *Trust-Region Methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2000)
13. Absil, P.A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. *Found. Comput. Math.* **7**(3) (July 2007) 303–330
14. Baker, C.G., Absil, P.A., Gallivan, K.A.: An implicit trust-region method on Riemannian manifolds. *IMA Journal of Numerical Analysis*, to appear (2008)
15. Amari, S., Cichocki, A., Yang, H.: A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems* **8** (1996) 757–763
16. Wismüller, A., Lange, O., Dersch, D., Leinsinger, G., Hahn, K., Pütz, B., Auer, D.: Cluster analysis of biomedical image time-series. *International Journal on Computer Vision* **46** (2002) 102–128
17. Keck, I., Theis, F., Gruber, P., Lang, E., Specht, K., Fink, G., Tomé, A., Puntotnet, C.: Automated clustering of ICA results for fMRI data analysis. In: Proc. CIMED 2005, Lisbon, Portugal (2005) 211–216