Beyond Pham's algorithm for joint diagonalization

Pierre Ablin, Jean-François Cardoso, Alexandre Gramfort

Beyond Pham's algorithm for joint diagonalization

Pierre Ablin[1], Jean-François Cardoso[2] and Alexandre Gramfort[1] *

1- INRIA - Parietal team
1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau - France

2- CNRS - Institut d'Astrophysique de Paris
98bis boulevard Arago, 75014, Paris - France

Abstract. The approximate joint diagonalization of a set of matrices consists in finding a basis in which these matrices are as diagonal as possible. This problem naturally appears in several statistical learning tasks such as blind signal separation. We consider the diagonalization criterion studied in a seminal paper by Pham (2001), and propose a new quasi-Newton method for its optimization. Through numerical experiments on simulated and real datasets, we show that the proposed method outperforms Pham's algorithm. An open source Python package is released.

1 Introduction

The task of joint diagonalization arises in several formulations of the blind source separation problem. In [1], independent component analysis is performed by joint-diagonalization of a set of cumulant matrices. In [2], the separation of non-stationnary signals is carried by joint-diagonalization of a set of autocorrelation matrices. Finally, in [3], joint-diagonalization of a set of covariance matrices separates Gaussian sources that have non-stationnary power.

Consider a set of n symmetric square matrices $\mathcal{C} \triangleq (C^1, \cdots, C^n)$ of size $p \times p$. Its approximate joint diagonalization consists in finding a matrix $B \in \mathbb{R}^{p \times p}$ such that the matrix set $B\mathcal{C}B^\top \triangleq (BC^1B^\top, \ldots, BC^nB^\top)$ contains matrices that are *as diagonal as possible*, as measured by some joint-diagonality criterion. This paper considers the joint diagonalization of positive matrices, defined as the minimization of the (non-convex) criterion

$$\mathcal{L}(B) = \frac{1}{2n} \sum_{i=1}^{n} \Big[\log\det \operatorname{diag}(BC^iB^\top) - \log\det(BC^iB^\top) \Big] \ , \tag{1}$$

This criterion was introduced by Pham in [3] who derived it as the negative log-likelihood of a source separation model for Gaussian stationary sources.

Pham [4] proposes in its seminal work a block coordinate descent approach for the minimization of $\mathcal{L}$. Each iteration of this method guarantees a decrease of $\mathcal{L}$. Further, when there exists a matrix B_* such that $B_*\mathcal{C}B_*^\top$ contains only diagonal matrices (that is, if the set is exactly jointly diagonalizable), then in the vicinity

*This work was supported by the Center for Data Science, funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02, and the European Research Council (ERC SLAB-YStG-676943).

of B_*, the algorithm converges quadratically. Since then, only a few other papers have focused on minimizing this criterion. In [5], the Newton method is studied. However, this method is not practical as it requires solving a $p^2 \times p^2$ linear system at each iteration. In [6], it is proposed to minimize an approximation of the cost function. The authors do so by alternating minimization over columns. This method minimizes an approximation of $\mathcal{L}$, and scales in $O(p^4)$.

In this paper, we propose a quasi-Newton method for the minimization of $\mathcal{L}$. We use sparse approximations of the Hessian which are cheap to compute and match the true Hessian when the set is jointly diagonalized, granting quadratic convergence. The algorithm as a cost per iteration of $O(n \times p^2)$, which is the natural scaling of the problem since it is the size of the dataset. Through experiments, we show that the proposed method outperforms Pham's algorithm, on both synthetic and real data.

Notation: The identity matrix of size p is denoted I_p. The Frobenius scalar product between two $p \times p$ matrices is noted as $\langle M|M'\rangle \triangleq \sum_{a,b} M_{ab}M'_{ab}$. The corresponding norm is $||M||_F = \sqrt{\langle M|M\rangle}$. Given a $p \times p \times p \times p$ tensor $\mathcal{H}$, the weighted scalar product is $\langle M|\mathcal{H}|M'\rangle \triangleq \sum_{a,b,c,d} \mathcal{H}_{abcd}M_{ab}M'_{cd}$. The Kronecker symbol δ_{ab} is 1 is $a = b$, 0 otherwise.

2 Study of the cost function

The cost function $\mathcal{L}(B)$ is defined on the group of invertible matrices, with $\log\det(BC^iB^\top)$ acting as a barrier. Its minimization is performed by iterative algorithms. To exploit the multiplicative structure of the group of invertible matrices, we perform *relative* updates on B [7]. The neighborhood of an iterate $B^{(t)}$ is parameterized by a small $p \times p$ matrix $\mathcal{E}$ as $B^{(t+1)} = (I_p + \mathcal{E})B^{(t)}$. The second-order (in $\mathcal{E}$) Taylor expansion of the loss function

$$\mathcal{L}((I_p + \mathcal{E})B) = \mathcal{L}(B) + \langle G|\mathcal{E}\rangle + \frac{1}{2}\langle \mathcal{E}|\mathcal{H}|\mathcal{E}\rangle + o(||\mathcal{E}||^2) \ . \tag{2}$$

where the $p \times p$ matrix $G(B)$ is the relative gradient and the $p \times p \times p \times p$ tensor $\mathcal{H}(B)$ is the relative Hessian. Simple algebra yields:

$$G_{ab} = \frac{1}{n}\sum_{i=1}^{n} \frac{D^i_{ab}}{D^i_{aa}} - \delta_{ab}, \ \text{ and } \ \mathcal{H}_{abcd} = \delta_{ac}\frac{1}{n}\sum_{i=1}^{n}\left[\frac{D^i_{bd}}{D^i_{aa}} - 2\frac{D^i_{ab}D^i_{ad}}{{D^i_{aa}}^2}\right] + \delta_{ad}\delta_{bc} \tag{3}$$

where we define $D^i = BC^iB^\top$ for $i = 1, \ldots, n$. Eq. (3) shows that $H_{aaaa} = 0$, consistent with the scale-indeterminacy of the criterion: $\mathcal{L}(\Lambda B) = \mathcal{L}(B)$ for any diagonal matrix Λ. The criterion is flat in the direction of the scale matrices.

Complexity: The cost of computing a gradient is $O(p^2 \times n)$. It is the natural complexity of an iterative algorithm as it is the same cost as computing the set $B\mathcal{C}B^\top$. Computing the Hessian is $O(p^3 \times n)$, and computing $H^{-1}G$ is $O(p^6)$. This is prohibitively costly when p is large compared to a gradient evaluation. As a consequence, Newton's method is extremely costly to setup.

3 Relative Quasi-Newton method

In this section, we first introduce an approximation of the Hessian and then derive a quasi-Newton algorithm to minimize (1).

3.1 Hessian approximation

The accelerated algorithm presented in this article is based on the massive sparsification of the Hessian tensor when matrices D^i are all diagonal. Indeed, it that case, it becomes:

$$\tilde{\mathcal{H}}_{abcd} = \delta_{ac}\delta_{bd}\Gamma_{ab} + \delta_{ad}\delta_{bc} - 2\delta_{abcd} \qquad \Gamma_{ab} = \frac{1}{n}\sum_{i=1}^{n}\frac{D^i_{bb}}{D^i_{aa}} \tag{4}$$

This Hessian approximation has three key properties. First, it is cheap to compute, at cost $O(p^2 \times n)$, just like a gradient. Then, it is sparse and structured. It only has $\simeq 2p^2$ non-zero coefficients, and can be seen as a block-diagonal operator, with blocks of size 2. Indeed, for a $p \times p$ matrix M, we have:

$$\begin{pmatrix}[\tilde{\mathcal{H}}M]_{ab} \\ [\tilde{\mathcal{H}}M]_{ba}\end{pmatrix} = \begin{pmatrix}\Gamma_{ab} & 1-2\delta_{ab} \\ 1-2\delta_{ab} & \Gamma_{ba}\end{pmatrix}\begin{pmatrix}M_{ab} \\ M_{ba}\end{pmatrix} \quad . \tag{5}$$

The following lemma establishes the positivity of the approximate Hessian:

Lemma 1 *The approximation $\tilde{\mathcal{H}}$ is positive with p zero eigenvalues. If the matrices C^i are independently sampled from continuous densities, with probability one, the other $p^2 - p$ eigenvalues are strictly positive.*

Proof: Using eq. (4), one has $\tilde{\mathcal{H}}E_{ii} = 0$, where E_{ii} is the matrix with 1 for its (i,i) coefficient, and 0 elsewhere. Thus $\tilde{\mathcal{H}}$ has p zero eigenvalues, and the associated eigenvectors are the E_{ii} for $i = 1 \cdots p$. The $p^2 - p$ other eigenvalues are the eigenvalues of the 2×2 blocks introduced in eq. (5). The diagonal coefficients of the blocks are the $\Gamma_{ab} > 0$. The determinant of a block is given by $\Gamma_{ab}\Gamma_{ba} - 1$. Cauchy-Schwartz inequality yields $\Gamma_{ab}\Gamma_{ba} \geq 1$, with equality if and only if $D^i_{aa} \propto D^i_{bb}$. This happens with probability 0, concluding the proof.

Finally, the approximation is straightforwardly inverted by inverting each 2×2 blocks. The Moore-Penrose pseudoinverse of $\tilde{\mathcal{H}}$, $\tilde{\mathcal{H}}^+$, satisfies:

$$[\tilde{\mathcal{H}}^+G]_{ab} = \frac{\Gamma_{ba}G_{ab} - G_{ba}}{\Gamma_{ab}\Gamma_{ba} - 1}, \ \ \forall a \neq b \ , \tag{6}$$

and $[\tilde{\mathcal{H}}^+G]_{aa} = 0$. The cost of inversion is thus $O(p^2)$.

3.2 Algorithm

The proposed quasi-Newton method uses $-\tilde{\mathcal{H}}^+G$ as search direction. Following from the positivity of $\tilde{\mathcal{H}}$, this is a descent direction. Unfortunately, there is no

Algorithm 1: Quasi-Newton method for joint-diagonalization

Input : Set of matrices $\mathcal{C}$, number of iterations T.
Initialize $B = I_p$.
for $t = 1, \cdots, T$ **do**
  Compute the gradient G using eq.(3)
  Compute the Hessian approximation $\tilde{\mathcal{H}}$ using eq.4
  Compute the search direction $-\tilde{\mathcal{H}}^{+}G$ using eq. (6)
  Do a backtracking line search in that direction to find a step size α decreasing $\mathcal{L}$
  Set $B \leftarrow (I_p - \alpha\tilde{\mathcal{H}}^{+}G)B$
end
Output: B

guarantee that the iteration $B \leftarrow (I_p - \tilde{\mathcal{H}}^{+}G)B$ decreases the cost function. Therefore, we resort to a line-search to find a step $\alpha > 0$ ensuring $\mathcal{L}((I_p - \alpha\tilde{\mathcal{H}}^{+}G)B) < \mathcal{L}(B)$ (condition $(*)$). This is done using backtracking, starting from $\alpha = 1$ and iterating $\alpha \leftarrow \frac{\alpha}{2}$ until the condition $(*)$ is met. The full algorithm is summarized in Algorithm. 1.

Quadratic convergence: Like Pham's algorithm, the proposed algorithm enjoys quadratic convergence when the matrix set is jointly diagonalizable. Indeed, assume that there exists a matrix B_* such that $B_*\mathcal{C}B_*^{\top}$ contains only diagonal matrices. Then, by construction, $\mathcal{H}(B_*) = \tilde{\mathcal{H}}(B_*)$. It follows that the method converges quadratically in the vicinity of B_*.

4 Experiments

4.1 Setting

For the experiments, three data sets are used – coming either from synthetic or real data – and we compare our approach to Pham's algorithm. The code to reproduce the experiments is available online[1]. We set $n = 100$ and $p = 40$.

Initialization: For a dataset $\mathcal{C}$, the algorithms start from a *whitener* (whitening matrix): writing $PDP^{\top} = \frac{1}{n}\sum_{i=1}^{n} C^i$ with P orthogonal and D diagonal, the initial matrix is taken as $B^{(0)} = D^{-\frac{1}{2}}P^{\top}$.

Metrics: To compare the speed of convergence of the algorithms, we monitor the diagonalization error $\mathcal{L}(B)$, and the gradient norm $||G(B)||$. The first quantity goes to 0 if the dataset is perfectly diagonalizable, while the second should always converge to 0 since the algorithm should reach a local minimum.

Synthetic data: We proceed as in [8] for generating synthetic datasets. We generate n diagonal matrices of size $p \times p$, $(D^1, \cdots D^n)$ for which each diagonal

[1] https://github.com/pierreablin/qndiag

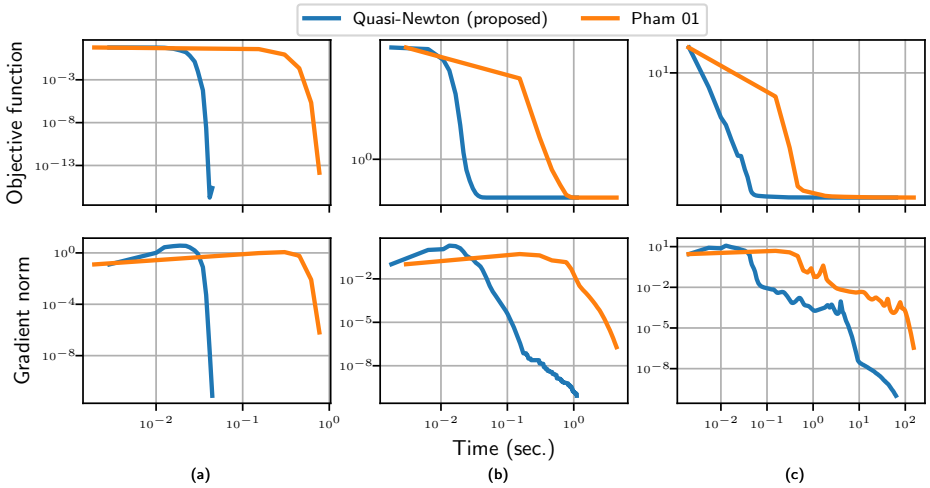

Fig. 1: Comparison of the algorithms on three datasets. **(a)**: Jointly diagonalizable synthetic dataset. **(b)**: Same dataset with added noise, rendering perfect joint diagonalization impossible.**(c)**: Real data, covariance matrices from MEG signals. Note the log-log scale.

coefficient is drawn from an uniform distribution in $[0, 1]$. Then, we generate a random 'mixing' matrix $A \in \mathbb{R}^{p\times p}$ with independent normally distributed entries. Finally, in order to add noise, we generate n matrices $R^1, \cdots, R^n \in \mathbb{R}^{p\times p}$ of normally distributed entries. The dataset is then $\mathcal{C} = (C^1, \cdots, C^n)$ with:

$$C^i = AD^iA^\top + \sigma^2(R^i)(R^i)^\top \ , \tag{7}$$

where σ controls the noise level. In practice we take $\sigma = 0$ (experiment **(a)**, perfecty jointly-diagonalizable set) or $\sigma = 0.1$ (experiment **(b)**).

Magnetoencephalography (MEG) data: We use the MNE sample dataset [9]. From n matrices containing p signals of T samples, $X_1, \cdots, X_n \in \mathbb{R}^{p\times T}$, corresponding to time segments of MEG signals, we jointly diagonalize the set of covariance matrices $C^i = \frac{1}{T}X_iX_i^\top$ (experiment **(c)**). This practical task is in the spirit of [3]. Results are displayed in Figure 1

4.2 Discussion

In experiment (a), where the dataset is exactly jointly diagonalizable, we observe the expected quadratic rate of convergence for both the proposed algorithm and Pham's method. We also observe that breaking the model (experiments (b) and (c)) makes convergence fall back to a linear rate. As expected, the cost function does not go to 0 in those cases. We observe that the proposed quasi-Newton algorithm outperforms Pham's method by about an order of magnitude on each experiment.

References

[1] J. F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proc. F - Radar and Signal Processing*, 140(6):362–370, Dec 1993.

[2] A. Belouchrani, K. Abed-Meraim, J-F Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

[3] D.T. Pham and J.F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Tr. SP*, 49(9):1837–1848, 2001.

[4] D.T. Pham. Joint approximate diagonalization of positive definite hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1136–1152, 2001.

[5] M. Joho. Newton method for joint approximate diagonalization of positive definite hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1205–1218, 2008.

[6] K. Todros and J. Tabrikian. Fast approximate joint diagonalization of positive definite hermitian matrices. In *ICASSP (3)*, pages 1373–1376, 2007.

[7] J-F Cardoso and B. H Laheld. Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, 44(12):3017–3030, 1996.

[8] A. Ziehe, G. Laskov, P.and Nolte, and K.R. Muller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5(Jul):777–800, 2004.

[9] A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. Hämäläinen. MNE software for processing MEG and EEG data. *Neuroimage*, 86:446–460, 2014.