Beyond Pham’s algorithm for joint diagonalization
Pierre Ablin, Jean-François Cardoso, Alexandre Gramfort

To cite this version:
Pierre Ablin, Jean-François Cardoso, Alexandre Gramfort. Beyond Pham’s algorithm for joint diagonalization. 2018. hal-01936887

HAL Id: hal-01936887
https://hal.archives-ouvertes.fr/hal-01936887
Submitted on 27 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Beyond Pham’s algorithm for joint diagonalization

Pierre Ablin\textsuperscript{1}, Jean-François Cardoso\textsuperscript{2} and Alexandre Gramfort\textsuperscript{1} \textsuperscript{*}

\textsuperscript{1} INRIA - Parietal team
1 Rue Honoré d’Estienne d’Orves, 91120 Palaiseau - France
\textsuperscript{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-stationary 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-stationary power.

Consider a set of $n$ symmetric square matrices $\mathcal{C} \equiv (C^1, \ldots, C^n)$ 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 $BCB^\top \equiv (BC^1B^\top, \ldots, BC^nB^\top)$ 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

$$\mathcal{L}(B) = \frac{1}{2n} \sum_{i=1}^{n} \left[ \log \det \text{diag}(BC^iB^\top) - \log \det(BC^iB^\top) \right], \quad (1)$$

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 $B^*CB^*_\top$ contains only diagonal matrices (that is, if the set is exactly jointly diagonalizable), then in the vicinity

\textsuperscript{*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 $||M||_F = \sqrt{\langle M | M \rangle}$.

2 Study of the cost function

The cost function $\mathcal{L}(B)$ is defined on the group of invertible matrices, with $\log \det(B C B^\top)$ 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 $E$ as $B(t+1) = (I_p + E) B(t)$. The second-order (in $E$) Taylor expansion of the loss function

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

(2)

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

$$G_{ab} = \frac{1}{n} \sum_{i=1}^{n} \frac{D_{ab}^i}{D_{aa}^i} - \delta_{ab}, \quad \text{and} \quad H_{abcd} = \delta_{ac} \frac{1}{n} \sum_{i=1}^{n} \left[ \frac{D_{ad}^i}{D_{aa}^i} - 2 \frac{D_{ab}^i D_{bd}^i}{D_{aa}^i} \right] + \delta_{ad} \delta_{bc}$$

(3)

where we define $D^i = BC^i B^\top$ 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 $\Lambda$. 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 $BCB^\top$. 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:

\[
\tilde{H}_{abcd} = \delta_{ac} \delta_{bd} \Gamma_{ab} + \delta_{ad} \delta_{bc} - 2 \delta_{abcd} \Gamma_{ab} = \frac{1}{n} \sum_{i=1}^{n} \frac{D_{ib}^i}{D_{ia}^i} \quad \Gamma_{ab} = \frac{1}{n} \sum_{i=1}^{n} \frac{D_{ib}^i}{D_{ia}^i}.
\] (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 \( \sim 2p^2 \) 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:

\[
\left( \begin{array}{cc} \tilde{H} M_{ab} & \tilde{H} M_{ba} \\ \tilde{H} M_{ab} & \tilde{H} M_{ba} \end{array} \right) = \left( \begin{array}{cc} \Gamma_{ab} & 1 - 2 \delta_{ab} \\ 1 - 2 \delta_{ab} & \Gamma_{ba} \end{array} \right) \left( \begin{array}{c} M_{ab} \\ M_{ba} \end{array} \right).
\] (5)

The following lemma establishes the positivity of the approximate Hessian:

**Lemma 1** The approximation \( \tilde{H} \) 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 \( \tilde{H} E_{ii} = 0 \), where \( E_{ii} \) is the matrix with 1 for its \( (i,i) \) coefficient, and 0 elsewhere. Thus \( \tilde{H} \) has \( p \) zero eigenvalues, and the associated eigenvectors are the \( E_{ii} \) for \( i = 1 \ldots p \). The \( p^2 - p \) other eigenvalues are the eigenvalues of the \( 2 \times 2 \) 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 \( D_{ia}^i \propto D_{ib}^i \). This happens with probability 0, concluding the proof.

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

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

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

3.2 Algorithm

The proposed quasi-Newton method uses \(-\tilde{H}^+ G\) as search direction. Following from the positivity of \( \tilde{H} \), this is a descent direction. Unfortunately, there is no
Algorithm 1: Quasi-Newton method for joint-diagonalization

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

guarantee that the iteration $B \leftarrow (I_p - \tilde{H}G)B$ decreases the cost function. Therefore, we resort to a line-search to find a step $\alpha > 0$ ensuring $\mathcal{L}((I_p - \alpha \tilde{H}G)B) < \mathcal{L}(B)$ (condition $(\ast)$). This is done using backtracking, starting from $\alpha = 1$ and iterating $\alpha \leftarrow \frac{\alpha}{2}$ until the condition $(\ast)$ 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 $B_*CB_*)^\top$ contains only diagonal matrices. Then, by construction, $\mathcal{H}(B_*) = \tilde{H}(B_*)$. 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 $C$, the algorithms start from a whitener (whitening matrix): writing $PD\!P^\top = \frac{1}{n} \sum_{i=1}^n C^i$ 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
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 \(C = (C_1, \cdots, C_n)\) with:
\[
C_i = AD_iA^\top + \sigma^2 (R_i)(R_i)^\top,
\]
where \(\sigma\) controls the noise level. In practice we take \(\sigma = 0\) (experiment (a), 
perfectly 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, \(X_1, \cdots, X_n \in \mathbb{R}^{p \times T}\), cor-
responding 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.

---

Fig. 1: Comparison of the algorithms on three datasets. (a): Jointly diagonal-
izable 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.
References


