JOINT DIAGONALIZATION VIA SUBSPACE FITTING TECHNIQUES

Alle-Jan van der Veen

Delft University of Technology, Dept. Electrical Engineering/DIMES, 2628 CD Delft, The Netherlands

Joint diagonalization problems of Hermitian or non-Hermitian matrices occur as the final parameter estimation step in several blind source separation problems such as ACMA, JADE, PARAFAC, and SOBI. Previous approaches have been Jacobi iteration schemes and alternating projections. Here we show how the joint diagonalization problem can be formulated as a (weighted) subspace fitting problem so that it can be solved using the efficient Gauss-Newton optimization algorithm proposed for that problem. Since a good initial point is usually available, the algorithm converges very fast.

1. INTRODUCTION

Suppose that we are given K complex Hermitian matrices $\hat{\mathbf{Y}}_k$ of the form

$$\hat{\mathbf{Y}}_k = \mathbf{A}\mathbf{\Lambda}_k\mathbf{A}^{\mathrm{H}} + \mathbf{E}_k\,, \qquad k = 1,\cdots,K, \tag{1}$$

where the $\mathbf{\Lambda}_k$ are diagonal and real, and $\mathbf{E}_k$ represents additive noise. The joint diagonalization problem we consider is, given the $\hat{\mathbf{Y}}_k$, to estimate the common factor $\mathbf{A}$. We assume that all $\hat{\mathbf{Y}}_k$ are square $d\times d$ matrices, and that $\mathbf{A}$ is square $d\times d$ with full rank d.

An extension of this problem is, for complex non-Hermitian matrices,

$$\hat{\mathbf{Y}}_k = \mathbf{A}\mathbf{\Lambda}_k\mathbf{B}^{\mathrm{H}} + \mathbf{E}_k\,, \qquad k = 1,\cdots,K, \tag{2}$$

where $\mathbf{A}$ and $\mathbf{B}$ can be different, and the $\mathbf{\Lambda}_k$ are diagonal but not necessarily real.

Joint diagonalization of either type turns up in several recently proposed blind source separation problems with data models $\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N}$, where $\mathbf{X}$ is the observation matrix, $\mathbf{A}$ is the mixing matrix, the rows of $\mathbf{S}$ contain the source signals, and $\mathbf{N}$ is additive noise. Depending on the assumptions on $\mathbf{A}$ and/or $\mathbf{S}$, the following types of algebraic source separation techniques have been proposed:

- Diagonalization of fourth order cumulant matrices, as in JADE [1] where $K = d$ and $\mathbf{A}$ is considered unitary.
- Algebraic Constant Modulus Algorithm (ACMA) [2]. Typically we have problem (1) with $K = d$ but $\mathbf{A}$ not unitary.
- Second order techniques for separating sources based on differences in their spectra, such as SOBI [3]. $\hat{\mathbf{Y}}_k$ are covariance matrices for several lags.
- Multi-dimensional ESPRIT [4, 5], and related applications such as joint angle-delay estimation [6], giving rise to problem (2) with $K = 3$ or more.
- Multilinear source separation problems, going under the name of PARAFAC models [7], also resulting in problem (2).

The joint diagonalization problem is a generalization of an eigenvalue problem. For two matrices, we can form $\hat{\mathbf{Y}}_2\hat{\mathbf{Y}}_1^{-1}$, assuming that $\hat{\mathbf{Y}}_1$ is invertible. Without noise, $\mathbf{Y}_2\mathbf{Y}_1^{-1} = \mathbf{A}\mathbf{\Lambda}_2\mathbf{\Lambda}_1^{-1}\mathbf{A}^{-1}$ and $\mathbf{A}$ follows from a standard eigenvalue decomposition. In the presence of noise, $\hat{\mathbf{Y}}_2\hat{\mathbf{Y}}_1^{-1}$ can still be diagonalized. When $K > 2$ matrices are available, the problem becomes overdetermined and we cannot find an exact diagonalizing factor.

The joint diagonalization problem is often solved by iterative Jacobi techniques (one-sided or two-sided; see [8, 9] for comparisons), or by iterations such as Alternating Least Squares [7, 10].

Here, we consider $\mathbf{A}$ (and $\mathbf{B}$) full rank but not necessarily unitary, and we derive an efficient Gauss-Newton iteration.

Notation Overbar $(\overline{\ })$ denotes complex conjugation, $^{\mathrm{T}}$ is the matrix transpose, $^{\mathrm{H}}$ the matrix complex conjugate transpose, † the matrix pseudo-inverse (Moore-Penrose inverse). $\mathbf{I}$ or $\mathbf{I}_p$ is the $(p\times p)$ identity matrix; $\mathbf{e}_i$ is its i-th column. $\mathrm{vec}(\mathbf{A})$ is a stacking of the columns of a matrix $\mathbf{A}$ into a vector. $\otimes$ is the Kronecker product, $\circ$ is the Khatri-Rao product, which is a column-wise Kronecker product: $\mathbf{A}\circ\mathbf{B} = [\mathbf{a}_1\otimes\mathbf{b}_1 \quad \mathbf{a}_2\otimes\mathbf{b}_2 \quad \cdots]$.

2. ALGORITHM DERIVATION

2.1. Cost function

Consider $\{\hat{\mathbf{Y}}_k\}$ of the form (1). If we assume that the entries of the additive noise matrix $\mathbf{E}_k$ are independent and identically distributed,[1] then it makes sense to consider the following Least Squares problem:

$$\{\hat{\mathbf{A}},\{\hat{\mathbf{\Lambda}}_k\}\} \;=\; \underset{\mathbf{A},\{\mathbf{\Lambda}_k\}}{\mathrm{argmin}} \sum_k \|\,\hat{\mathbf{Y}}_k - \mathbf{A}\mathbf{\Lambda}_k\mathbf{A}^{\mathrm{H}}\,\|_{\mathrm{F}}^2 \tag{3}$$

where the $\mathbf{\Lambda}_k$ are real and diagonal. This is the cost function usually considered for joint diagonalization. Without loss of generality, we can put a norm constraint on the columns of $\mathbf{A}$, say $\|\mathbf{a}_k\| = 1$.

The cost function can be rewritten as

$$\begin{aligned}\textstyle\sum_k \|\,\hat{\mathbf{Y}}_k - \mathbf{A}\mathbf{\Lambda}_k\mathbf{A}^{\mathrm{H}}\,\|_{\mathrm{F}}^2 \;&=\; \textstyle\sum_k \|\,\hat{\mathbf{y}}_k - (\bar{\mathbf{A}}\circ\mathbf{A})\mathbf{m}_k\,\|_{\mathrm{F}}^2 \\ &=\; \|\,\hat{\mathbf{Y}} - (\bar{\mathbf{A}}\circ\mathbf{A})\mathbf{M}\,\|_{\mathrm{F}}^2\end{aligned}$$

where $\hat{\mathbf{y}}_k = \mathrm{vec}(\hat{\mathbf{Y}}_k)$, $\mathbf{m}_k = \mathrm{diag}(\mathbf{\Lambda}_k)$, $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1,\cdots,\hat{\mathbf{y}}_K]$, $\mathbf{M} = [\mathbf{m}_1\,,\cdots,\ \mathbf{m}_K]$. Thus the joint diagonalization problem is equivalent to

$$\{\hat{\mathbf{A}},\hat{\mathbf{M}}\} \;=\; \mathrm{argmin}\ \|\,\hat{\mathbf{Y}} - \utilde{\mathbf{A}}\mathbf{M}\,\|_{\mathrm{F}}^2, \qquad \utilde{\mathbf{A}} \;:=\; \bar{\mathbf{A}}\circ\mathbf{A}, \tag{4}$$

which is recognized as a Subspace Fitting problem where the K columns of $\hat{\mathbf{Y}}$ are considered to span a subspace, and we seek to model this subspace by d unit-norm vectors of the form $\bar{\mathbf{a}}\otimes\mathbf{a}$. $\mathbf{M}$ is a full rank (real) $d\times K$ matrix that relates the two bases.

We immediately note that in noise-free conditions, if the model holds then $\mathbf{Y}$ has maximal rank d. If $K > d$, then $\mathbf{Y}$ must be rank deficient and its dimension can be reduced if so desired.

To solve the problem, we can follow entirely similar procedures as for subspace fitting, viz. [11]. First note that the optimization problem is separable, since $\hat{\mathbf{M}} = \utilde{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}}$. (It is automatically real, see Appendix B.) Thus, we can eliminate $\mathbf{M}$ and reduce (4) to

$$\hat{\mathbf{A}} \;=\; \mathrm{argmin}\ \|\,\hat{\mathbf{Y}} - \utilde{\mathbf{A}}\utilde{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}}\,\|_{\mathrm{F}}^2 \;=\; \mathrm{argmin}\ \|\,\mathbf{P}_{\utilde{\mathbf{A}}}^{\perp}\hat{\mathbf{Y}}\,\|_{\mathrm{F}}^2 \tag{5}$$

where $\mathbf{P}_{\utilde{\mathbf{A}}}^{\perp} := \mathbf{I} - \utilde{\mathbf{A}}\utilde{\mathbf{A}}^{\dagger}$.

[1] More precisely, in view of the Hermitian symmetry of $\mathbf{E}_k$, this is understood for the real and imaginary parts of the upper triangular part of $\mathbf{E}_k$.

2.2. Minimizing the cost function via Gauss-Newton

Assume that $\mathbf{A}$ is parametrized by a uniquely identifiable parametrization $\mathbf{A} = \mathbf{A}(\boldsymbol{\theta})$, and consider the cost function

$$J(\boldsymbol{\theta}) \;=\; \tfrac{1}{2}\|\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\hat{\mathbf{Y}}\|_{\mathrm{F}}^{2} \;=\; \tfrac{1}{2}\|\mathrm{vec}(\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\hat{\mathbf{Y}})\|^{2} \;=\; \tfrac{1}{2}\mathbf{f}(\boldsymbol{\theta})^{\mathrm{H}}\mathbf{f}(\boldsymbol{\theta}) \qquad (6)$$

where $\mathbf{f}(\boldsymbol{\theta}) = \mathrm{vec}(\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\hat{\mathbf{Y}})$. This is a quadratic minimization problem suitable for the Gauss-Newton optimization scheme [12]. Define the Jacobian

$$\mathbf{F}(\boldsymbol{\theta}) \;=\; [\frac{\mathrm{d}\mathbf{f}(\boldsymbol{\theta})}{\mathrm{d}\theta_1}\,,\; \frac{\mathrm{d}\mathbf{f}(\boldsymbol{\theta})}{\mathrm{d}\theta_2}\,,\; \cdots].$$

The gradient of the cost function at $\boldsymbol{\theta}$ is $\mathbf{g} = \mathrm{Re}(\mathbf{F}(\boldsymbol{\theta})^{\mathrm{H}}\mathbf{f}(\boldsymbol{\theta}))$. According to the Gauss-Newton scheme, the Hessian of the cost function is approximated by $\mathbf{H} \approx \mathrm{Re}(\mathbf{F}^{\mathrm{H}}\mathbf{F})$, and the Gauss-Newton update step is

$$\boldsymbol{\theta}^{(k+1)} \;=\; \boldsymbol{\theta}^{(k)} - \mu_k \mathbf{H}^{-1}\mathbf{g}.$$

$\mu_k \in [0,1]$ is a step size; with a good initial point we can take $\mu_k = 1$. To apply Gauss-Newton, it remains to (*i*) select a suitable parametrization, (*ii*) give an explicit form of $\mathbf{F}$ in terms of the problem variables, and (*iii*) compute an initial point.

2.3. Parametrization

Recall that $\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_d]$, where each $\mathbf{a}_i$ is normalized to unit norm. We can furthermore constrain the first entry of each $\mathbf{a}_i$ to be positive real. Let p be the number of (real-valued) parameters per $\mathbf{a}$-vector. An important property of the parametrization is that it is minimal, otherwise the Hessian will become singular, leading to problems in the optimization. A convenient parametrization with $p = 2(d-1)$ real parameters is given in Appendix A.

Let $\boldsymbol{\theta}_i$ be the parameter vector for $\mathbf{a}_i$,

$$\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_d] : p \times d, \qquad \boldsymbol{\theta} = \mathrm{vec}(\boldsymbol{\Theta}).$$

The entries of $\boldsymbol{\Theta}$ will be denoted by θ_{ij}, $(i = 1, \cdots, p,\; j = 1, \cdots d)$, the entries of $\boldsymbol{\theta}$ by θ_η. We will collect the derivatives of $\mathbf{A}$ and $\underset{\sim}{\mathbf{A}}$ in vectors and matrices, defined as follows:

$$\begin{array}{lcl@{\qquad}lcl}
\mathbf{d}_{ij} &=& \frac{\mathrm{d}\mathbf{a}_j}{\mathrm{d}\theta_{ij}}(\boldsymbol{\theta}) & \underset{\sim}{\mathbf{d}}_{ij} &=& \frac{\mathrm{d}\underset{\sim}{\mathbf{a}}_j}{\mathrm{d}\theta_{ij}}(\boldsymbol{\theta}) \\
\mathbf{D}_j &=& [\mathbf{d}_{1j} \cdots \mathbf{d}_{pj}] & \underset{\sim}{\mathbf{D}}_j &=& [\underset{\sim}{\mathbf{d}}_{1j} \cdots \underset{\sim}{\mathbf{d}}_{pj}] \\
\mathbf{D} &=& [\mathbf{D}_1 \;\cdots\; \mathbf{D}_d] & \underset{\sim}{\mathbf{D}} &=& [\underset{\sim}{\mathbf{D}}_1 \;\cdots\; \underset{\sim}{\mathbf{D}}_d].
\end{array} \qquad (7)$$

Since $\underset{\sim}{\mathbf{a}}_j = \bar{\mathbf{a}}_j \otimes \mathbf{a}_j$, we obtain

$$\begin{array}{rcl}
\underset{\sim}{\mathbf{d}}_{ij} &=& \bar{\mathbf{a}}_j \otimes \mathbf{d}_{ij} + \bar{\mathbf{d}}_{ij} \otimes \mathbf{a}_j \\
\underset{\sim}{\mathbf{D}}_j &=& [\bar{\mathbf{a}}_j \;\cdots \bar{\mathbf{a}}_j] \circ \mathbf{D}_j \;+\; \bar{\mathbf{D}}_j \circ [\mathbf{a}_j \;\cdots \mathbf{a}_j] \\
\underset{\sim}{\mathbf{D}} &=& \mathbf{A}_e \circ \mathbf{D} + \mathbf{D} \circ \mathbf{A}_e
\end{array} \qquad (8)$$

where $\mathbf{A}_e := \mathbf{A} \otimes \mathbf{1}_p^{\mathrm{T}} = [\mathbf{a}_1 \;\cdots\; \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_d \;\cdots\; \mathbf{a}_d]$.

2.4. Explicit form of F

Recall that $\mathbf{f} = \mathrm{vec}(\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\hat{\mathbf{Y}})$. Let η be the index of one of the parameters θ_{ij} in $\boldsymbol{\theta}$, and let $\underset{\sim}{\mathbf{A}}_\eta := \frac{\mathrm{d}\underset{\sim}{\mathbf{A}}}{\mathrm{d}\theta_\eta}$. The derivative of $\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}$ is [11]

$$\underset{\sim}{\mathbf{P}}_\eta \;:=\; \frac{\mathrm{d}\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}}{\mathrm{d}\theta_\eta} \;=\; -\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\underset{\sim}{\mathbf{A}}_\eta\underset{\sim}{\mathbf{A}}^{\dagger} \;-\; (\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\underset{\sim}{\mathbf{A}}_\eta\underset{\sim}{\mathbf{A}}^{\dagger})^{\mathrm{H}}.$$

Thus, the derivative of $\mathbf{f}$ to θ_η is

$$\mathbf{f}'_\eta := \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\theta_\eta} = \mathrm{vec}(\frac{\mathrm{d}\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}}{\mathrm{d}\theta_\eta}\hat{\mathbf{Y}}) = -\mathrm{vec}\left(\left(\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\underset{\sim}{\mathbf{A}}_\eta\underset{\sim}{\mathbf{A}}^{\dagger} + (\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\underset{\sim}{\mathbf{A}}_\eta\underset{\sim}{\mathbf{A}}^{\dagger})^{\mathrm{H}}\right)\hat{\mathbf{Y}}\right).$$

At this point, we propose to ignore the second term in this expression, since the factor $\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\hat{\mathbf{Y}}$ occuring in that term corresponds to the residual and is typically very small in the neighborhood of the optimum (moreover, it cancels in forming $\mathbf{F}^{\mathrm{H}}\mathbf{f}$). Thus, in first order approximation

$$\mathbf{f}'_\eta \;\approx\; -\mathrm{vec}(\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\underset{\sim}{\mathbf{A}}_\eta\underset{\sim}{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}}) \;=\; -[(\underset{\sim}{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}})^{\mathrm{T}} \otimes \underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}]\mathrm{vec}(\underset{\sim}{\mathbf{A}}_\eta). \qquad (9)$$

The Jacobian is thus given by

$$\mathbf{F} \;=\; [\mathbf{f}'_1\,,\; \cdots,\; \mathbf{f}'_{pd}] \;=\; -[(\underset{\sim}{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}})^{\mathrm{T}} \otimes \underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}]\underset{\sim}{\mathbf{D}}_e \qquad (10)$$

where

$$\underset{\sim}{\mathbf{D}}_e \;:=\; [\mathrm{vec}(\underset{\sim}{\mathbf{A}}_\eta)]_{\eta=1,\cdots,pd} \;=\; [\frac{\mathrm{dvec}(\underset{\sim}{\mathbf{A}})}{\mathrm{d}\theta_{11}}\,,\; \frac{\mathrm{dvec}(\underset{\sim}{\mathbf{A}})}{\mathrm{d}\theta_{21}}\,,\; \cdots].$$

Recall that we have defined

$$\underset{\sim}{\mathbf{D}} = [\frac{\mathrm{d}\underset{\sim}{\mathbf{a}}_1}{\mathrm{d}\theta_{11}}\,,\; \frac{\mathrm{d}\underset{\sim}{\mathbf{a}}_1}{\mathrm{d}\theta_{21}}\,,\; \cdots,\; \frac{\mathrm{d}\underset{\sim}{\mathbf{a}}_2}{\mathrm{d}\theta_{12}}\,,\; \cdots].$$

$\underset{\sim}{\mathbf{D}}_e$ merely augments each column of $\underset{\sim}{\mathbf{D}}$ with many zero entries since each parameter affects only one column of $\underset{\sim}{\mathbf{A}}$. Thus

$$\underset{\sim}{\mathbf{D}}_e = [\mathbf{e}_1 \cdots \mathbf{e}_1 \mid \mathbf{e}_2 \cdots \mathbf{e}_2 \mid \cdots \mid \mathbf{e}_d \cdots \mathbf{e}_d] \circ \underset{\sim}{\mathbf{D}} = (\mathbf{I}_d \otimes \mathbf{1}_p^{\mathrm{T}}) \circ \underset{\sim}{\mathbf{D}}$$

and after substitution of this in (10)

$$\mathbf{F} = -[(\underset{\sim}{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}} \otimes \mathbf{1}_p)^{\mathrm{T}} \circ \underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\underset{\sim}{\mathbf{D}}].$$

At this point, we note that due to the Hermitian symmetry of the problem (see Appendix B), $\mathbf{F}^{\mathrm{H}}\mathbf{f}$ and $\mathbf{F}^{\mathrm{H}}\mathbf{F}$ are automatically real, so that we have $\mathbf{H}^{-1}\mathbf{g} = (\mathbf{F}^{\mathrm{H}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{H}}\mathbf{f} = \mathbf{F}^{\dagger}\mathbf{f}$. Given an estimate $\boldsymbol{\theta}^{(k)}$, one iteration of the Gauss-Newton scheme thus becomes as follows.

$$\boxed{\begin{array}{rcl@{\qquad}r}
\mathbf{A} &:=& \mathbf{A}(\boldsymbol{\theta}^{(k)}) & [d \times d] \\
\mathbf{D} &:=& \mathbf{D}(\boldsymbol{\theta}^{(k)}) & [d \times dp] \\
\mathbf{A}_e &:=& \mathbf{A} \otimes \mathbf{1}_p^{\mathrm{T}} & [d \times dp] \\
\underset{\sim}{\mathbf{D}} &:=& \bar{\mathbf{A}}_e \circ \mathbf{D} + \bar{\mathbf{D}} \circ \mathbf{A}_e & [d^2 \times dp] \\
\underset{\sim}{\mathbf{A}} &:=& \bar{\mathbf{A}} \circ \mathbf{A} & [d^2 \times d] \\
\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}} &:=& \mathbf{I} - \underset{\sim}{\mathbf{A}}\underset{\sim}{\mathbf{A}}^{\dagger} & [d^2 \times d^2] \\
\boldsymbol{\theta}^{(k+1)} &:=& \boldsymbol{\theta}^{(k)} + \mu_k[(\underset{\sim}{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}} \otimes \mathbf{1}_p)^{\mathrm{T}} \circ \underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\underset{\sim}{\mathbf{D}}]^{\dagger}\mathrm{vec}(\underset{\sim}{\mathbf{P}}^{\perp}_{\mathbf{A}}\hat{\mathbf{Y}}) &
\end{array}}$$

The complexity of an iteration is $\mathcal{O}(d^2K(pd)^2)$, with $p = 2(d-1)$.

2.5. Initial point and preprocessing

A suitable initial estimate for $\mathbf{A}$ is usually obtained from an eigenvalue decomposition of $\hat{\mathbf{Y}}_2\hat{\mathbf{Y}}_1^{-1}$, since in the noise free case

$$\mathbf{Y}_2\mathbf{Y}_1^{-1} = \mathbf{A}(\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_1^{-1})\mathbf{A}^{-1}.$$

The initial point $\boldsymbol{\theta}^{(0)}$ is derived from $\mathbf{A}$ as described in Appendix A. With this initialization, the iteration converges very fast, typically within two steps. The initialization assumes that (*i*) the inverse of $\hat{\mathbf{Y}}_1$ exists, (*ii*) the eigenvalues $\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_1^{-1}$ are not repeated, and (*iii*) they are real. The latter requirement sometimes gives problems: with strong noise, it may happen that the eigenvalues of $\hat{\mathbf{Y}}_2\hat{\mathbf{Y}}_1^{-1}$ become complex. One can prove that if either $\hat{\mathbf{Y}}_1$ or $\hat{\mathbf{Y}}_2$ is positive definite, then the eigenvalues of $\hat{\mathbf{Y}}_2\hat{\mathbf{Y}}_1^{-1}$ are real. Thus, we can search among the $\hat{\mathbf{Y}}_k$ for a matrix that is positive definite and use this matrix in the initialization. We may also try to find a linear combination of the $\hat{\mathbf{Y}}_k$ such that the result is positive definite.

To generalize the model, suppose that $\mathbf{A}$ has size $d \times r$. If $r < d$, then r is the rank of the $\mathbf{Y}_k$, and these matrices are rank deficient.

To improve the initialization, it is better to first reduce the dimensions to the square ($r \times r$) case. This can be done via a singular value decomposition of $[\hat{\mathbf{Y}}_1 \ \cdots \ \hat{\mathbf{Y}}_K]$:

$$[\hat{\mathbf{Y}}_1 \ \cdots \ \hat{\mathbf{Y}}_K] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} \tag{11}$$

The rank r can be detected from the singular values. Let $\hat{\mathbf{U}}$ be the r dominant left singular vectors, then we can replace the $\hat{\mathbf{Y}}_k$ by compressed $r \times r$ matrices $\hat{\mathbf{Y}}'_k = \hat{\mathbf{U}}^{\mathrm{H}}\hat{\mathbf{Y}}_k\hat{\mathbf{U}}$, and solve the joint diagonalization problem $\hat{\mathbf{Y}}'_k = \mathbf{T}\mathbf{\Lambda}_k\mathbf{T}^{\mathrm{H}}$, initialized by the solution of

$$\hat{\mathbf{Y}}'_2(\hat{\mathbf{Y}}'_1)^{-1} = \mathbf{T}(\mathbf{\Lambda}_2\mathbf{\Lambda}_1^{-1})\mathbf{T}^{-1} \,. \tag{12}$$

After finding $\mathbf{T}$, we can set $\mathbf{A} = \hat{\mathbf{U}}\mathbf{T}$. In fact, we have a choice to either solve the joint diagonalization problem for the $\hat{\mathbf{Y}}'_k$ (this is most efficient but does not exactly solve (3)), or to find the initialial $\mathbf{T}$ from (12) and solve the original problem initialized by $\mathbf{A} = \hat{\mathbf{U}}\mathbf{T}$.

If from the singular values in (11) it turns out that $r > d$, then we are in a situation with “less sensors than sources”. For this situation, there appears to be no closed form solution available yet. Although unelegant, we can try a random initialization, and follow the iteration with a reduced step size (e.g. $\mu_k = 0.5$). Convergence now takes longer, in the order of 15 iterations.

If $K > d$, then $\mathbf{Y}$ is necessarily rank deficient (rank d). This is not a problem for the algorithm, but it is possible to reduce the dimension to d, by replacing $\hat{\mathbf{Y}}$ by its dominant d left singular vectors. Thus, we can reduce cases with $K > d$ and $d > r$ to a generic case of $K = d = r$.

2.6. Remarks

A requirement for the iteration is that $\mathbf{F}$ is a tall matrix of full column rank. If $\mathbf{A} : d \times r$, then $\mathbf{F}$ has size $Kd^2 \times pr$, where for the current parametrization $p = 2(d-1)$, so that $Kd^2 \geq 2r(d-1)$. If $r = d$, then we need $K \geq 2$. Increasing K or d increases the tallness of $\mathbf{F}$ and improves its conditioning and thus the robustness of the optimization. For sufficiently large K, it is seen that $r > d$ is permitted. However, only $r \leq d$ gives convenient initialization.

3. EXTENSIONS

3.1. Parametrized array

In some applications where joint diagonalization problems occur, the columns of $\mathbf{A}$ are not arbitrary vectors but functions of a single (or a few) parameters. For example, in array signal processing, we might have $\mathbf{a} = \mathbf{a}(\theta)$, where θ corresponds to the direction of the source. It is clear that we can directly exploit this more parsimonious parametrization. The algorithm remains the same, but we have a smaller parameter vector ($p = d$ rather than $p = 2(d-1)$), and only the functions $\mathbf{A}(\boldsymbol{\theta})$ and $\mathbf{D}(\boldsymbol{\theta})$ need to change. An example is the combination of ACMA or JADE to take a directional model into account.

3.2. Unsymmetric joint diagonalization

An extension of the joint diagonalization problem in (1) is the data model

$$\hat{\mathbf{Y}}_k = \mathbf{A}\mathbf{\Lambda}_k\mathbf{B}^{\mathrm{H}} + \mathbf{E}_k \,, \qquad k = 1, \cdots, K \,.$$

In this problem, the left and right factors $\mathbf{A}$ and $\mathbf{B}$ are not necessarily equal to each other (perhaps even with different dimensions), and $\mathbf{\Lambda}_k$ are diagonal but not necessarily real.

If our objective is again to minimize the model error, we may derive as before

$$\sum_i \|\hat{\mathbf{Y}}_i - \mathbf{A}\mathbf{\Lambda}_i\mathbf{B}^{\mathrm{H}}\|_{\mathrm{F}}^2 \;=\; \sum_i \|\hat{\mathbf{y}}_i - (\bar{\mathbf{B}} \circ \mathbf{A})\mathbf{m}_i\|_{\mathrm{F}}^2 \;=\; \|\hat{\mathbf{Y}} - (\bar{\mathbf{B}} \circ \mathbf{A})\mathbf{M}\|_{\mathrm{F}}^2$$

where $\mathbf{M}$ is an arbitrary complex matrix. Thus, we end up with a very similar subspace fitting problem, except that now we have two parameter sets: $\boldsymbol{\theta} = [\boldsymbol{\theta}_A^{\mathrm{T}}, \boldsymbol{\theta}_B^{\mathrm{T}}]^{\mathrm{T}}$, where $\mathbf{A} = \mathbf{A}(\boldsymbol{\theta}_A)$ and $\mathbf{B} = \mathbf{B}(\boldsymbol{\theta}_B)$. Since without loss of generality we can still take the columns of $\mathbf{A}$ and $\mathbf{B}$ to be unit norm and with positive real first entry, we can use the same parametrization as before. Thus, we can use a very similar Gauss-Newton iteration, now with

$$\begin{aligned}
\underset{\sim}{\mathbf{A}} &:= \bar{\mathbf{B}} \circ \mathbf{A} \\
\underset{\sim}{\mathbf{D}}_A &:= \bar{\mathbf{B}}_e \circ \mathbf{D}(\boldsymbol{\theta}_A) \,, \qquad \underset{\sim}{\mathbf{D}}_B := \bar{\mathbf{D}}(\boldsymbol{\theta}_B) \circ \mathbf{A}_e \,. \\
\mathbf{F} &= -\left[\, (\underset{\sim}{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}} \otimes \mathbf{1}_p)^{\mathrm{T}} \circ \mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\underset{\sim}{\mathbf{D}}_A \quad , \quad (\underset{\sim}{\mathbf{A}}^{\dagger}\hat{\mathbf{Y}} \otimes \mathbf{1}_p)^{\mathrm{T}} \circ \mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\underset{\sim}{\mathbf{D}}_B \, \right] \\
\mathbf{f} &= \mathrm{vec}(\mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\hat{\mathbf{Y}}) \\
\boldsymbol{\theta}^{(k+1)} &= \boldsymbol{\theta}^{(k)} - \mu_k[\mathrm{Re}(\mathbf{F}^{\mathrm{H}}\mathbf{F})]^{-1}\mathrm{Re}(\mathbf{F}^{\mathrm{H}}\mathbf{f}) \,.
\end{aligned}$$

$\mathbf{F}$ has size $Kd^2 \times 2pr$, with $p = 2(d-1)$. To have $\mathbf{H}$ invertible, we need $[\mathrm{Re}(\mathbf{F})^{\mathrm{T}} \ \ \mathrm{Im}(\mathbf{F})^{\mathrm{T}}]^{\mathrm{T}}$ to be tall, or $2Kd^2 \geq 4(d-1)r$. This leads to the same conditions as we had for the symmetric case before.

3.3. Weighted subspace fitting

Weighted subspace fitting estimates are usually obtained by inserting in (6) (but not in the original (3)) a positive definite weighting matrix $\mathbf{\Gamma}$. In its most general form, the cost function becomes

$$J(\boldsymbol{\theta}) \;=\; \|\mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\hat{\mathbf{Y}}\|^2_{\mathbf{\Gamma}} \;=\; \mathrm{vec}(\mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\hat{\mathbf{Y}})^{\mathrm{H}}\mathbf{\Gamma}\,\mathrm{vec}(\mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\hat{\mathbf{Y}}) \,. \tag{13}$$

$\mathbf{\Gamma}$ can be used to minimize the estimator variance. From the theory of WSF (viz. [11,13]) we know that the optimal weight is $\mathbf{\Gamma}^{\mathrm{opt}} = \mathbf{C}^{\dagger}$ where

$$\mathbf{C} = \mathrm{E}\{\mathrm{vec}(\mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\hat{\mathbf{Y}})^{\mathrm{H}}\mathrm{vec}(\mathbf{P}^{\perp}_{\underset{\sim}{\mathbf{A}}}\hat{\mathbf{Y}})\} \,.$$

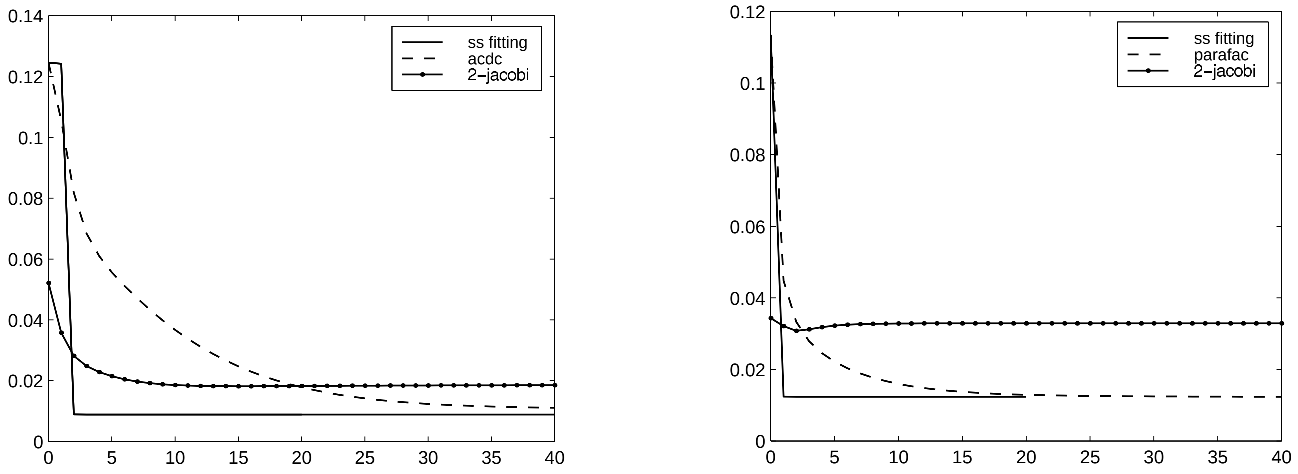
Thus, the optimal weight depends on the covariance of $\hat{\mathbf{Y}}$, and in turn on the origin of the problem.

4. SIMULATIONS

Figure 1(a) shows a test with $K = 4$ Hermitian matrices of size $d = 4$, a randomly generated complex $\mathbf{A}$ (i.i.d. entries with standard deviation 1), $\mathbf{\Lambda}_k$ (std 1), and $\mathbf{E}_k$ (std 0.05). We compare the subspace fitting technique using Gauss-Newton iterations with step size $\mu_k = 1$ to ACDC [10], which is an alternating least squares type technique that optimizes $\mathbf{A}$ and $\mathbf{\Lambda}_k$ in turn, and two-sided Jacobi iterations as in [2]. The latter is a QZ iteration that tries to solve $\mathbf{Q}\hat{\mathbf{Y}}_k\mathbf{Z} = \mathbf{R}_k$ for unitary $\mathbf{Q}$ and $\mathbf{Z}$ and upper triangular $\mathbf{R}_k$, and subsequenly derives $\mathbf{A}$ from the result. The subspace fitting and ACDC algorithms are initialized from an eigenvalue decomposition of $\hat{\mathbf{Y}}_2\hat{\mathbf{Y}}_1^{-1}$, the Jacobi iterations from a QZ decomposition. From the graph it is seen that the subspace fitting converges in two steps, ACDC converges to almost the same point in about 40 steps, whereas the Jacobi iterations quickly converge but to a different point.

Figure 1(b) shows a similar test with non-Hermitian matrices. Here we compare the subspace fitting technique ($\mu_k = 1$) to two-sided Jacobi iterations and to PARAFAC [7], which is an alternating least squares technique for the non-Hermitian case. The results are similar to the symmetric case.

The smaller number of iterations for the subspace fitting technique is to some extent offset by its larger complexity: $\mathcal{O}(d^7)$ compared to $\mathcal{O}(d^4)$ for the other iterative techniques. As is well known, the Gauss-Newton iteration with maximal step size $\mu_k = 1$ gives fastest convergence but is only robust if the initialization point is sufficiently close. For ill-conditioned $\mathbf{A}$, the eigenvalue decomposition of two matrices is not always accurate enough, and a more conservative step size has to be used for the first few steps.



(*a*) Hermitian, (*b*) non-Hermitian.

A. PARAMETRIZATION OF A UNIT-NORM VECTOR a

A minimal parametrization of a complex unit-norm vector **a** with d entries and positive real first entry is provided by a sequence of Givens rotations:

$$\mathbf{a} \;=\; \mathbf{\Phi}\mathbf{R}_1(\alpha_1)\mathbf{R}_2(\alpha_2)\cdots\mathbf{R}_{d-1}(\alpha_{d-1})\mathbf{e}_1$$

where

$$\mathbf{\Phi} = \operatorname{diag}[1,\, e^{j\phi_1}, \cdots, e^{j\phi_{d-1}}], \qquad 0 \le \phi_i < 2\pi,$$

$$\mathbf{R}_i(\alpha) = \begin{bmatrix} c & & -s & \\ & \mathbf{I}_{i-1} & & \\ s & & c & \\ & & & \mathbf{I}_{d-1-i} \end{bmatrix},$$

$$c = \cos(\alpha),\; s = \sin(\alpha),\; -\tfrac{\pi}{2} \le \alpha < \tfrac{\pi}{2}$$

The parameter vector for $\mathbf{a}(\boldsymbol{\theta})$ is $\boldsymbol{\theta} = [\alpha_1, \cdots, \alpha_{d-1}, \phi_1, \cdots, \phi_{d-1}]^{\mathrm{T}}$. We will also need the derivative of $\mathbf{a}(\boldsymbol{\theta})$ to each of the $p = 2(d-1)$ parameters:

$$\mathbf{d}_k = \frac{\mathrm{d}\mathbf{a}}{\mathrm{d}\theta_k}(\boldsymbol{\theta}) = \begin{cases} \dfrac{\mathrm{d}\mathbf{a}}{\mathrm{d}\alpha_i}, & i = k, \qquad 1 \le k \le d-1, \\ \dfrac{\mathrm{d}\mathbf{a}}{\mathrm{d}\phi_i}, & i = k-d+1, \quad d \le k \le 2(d-1) \end{cases}$$

where

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}\alpha_i} \;=\; \mathbf{\Phi}\mathbf{R}_1(\alpha_1)\cdots\mathbf{R}_{i-1}(\alpha_{i-1})\mathbf{R}'_i(\alpha_i)\mathbf{R}_{i+1}(\alpha_{i+1})\cdots\mathbf{R}_{d-1}(\alpha_{d-1})\mathbf{e}_1$$

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}\phi_i} \;=\; j\mathbf{e}_{i+1}\mathbf{e}_{i+1}^{\mathrm{H}}\mathbf{a}, \qquad \mathbf{R}'_i(\alpha) = \begin{bmatrix} -s & & -c & \\ & \mathbf{0}_{i-1} & & \\ c & & -s & \\ & & & \mathbf{0}_{d-1-i} \end{bmatrix}$$

B. REAL PROCESSING

Since $\bar{\mathbf{a}} \otimes \mathbf{a} = \operatorname{vec}(\mathbf{a}\mathbf{a}^{\mathrm{H}})$, the entries of this vector have a certain Hermitian symmetry property. It follows that there exists a data-independent unitary matrix **Q** such that $\mathbf{Q}(\bar{\mathbf{a}} \otimes \mathbf{a})$ is real, for any **a**. A consequence of this is that for the Hermitian joint diagonalization problem (1), all kinds of derived matrices are real or can be mapped to real. In particular, the expressions for $\hat{\mathbf{M}}$, the gradient $\mathbf{F}^{\mathrm{H}}$ and Hessian $\mathbf{F}^{\mathrm{H}}\mathbf{F}$ are real by itself.

REFERENCES

[1] J.F. Cardoso and A. Souloumiac, “Blind beamforming for non-Gaussian signals,” *IEE Proc. F (Radar and Signal Processing)*, vol. 140, pp. 362–370, Dec. 1993.

[2] A.J. van der Veen and A. Paulraj, “An analytical constant modulus algorithm,” *IEEE Trans. Signal Processing*, vol. 44, pp. 1136–1155, May 1996.

[3] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, “A blind source separation technique using second-order statistics,” *IEEE Trans. Signal Proc.*, vol. 45, pp. 434–444, Feb. 1997.

[4] A.J. van der Veen, P.B. Ober, and E.F. Deprettere, “Azimuth and elevation computation in high resolution DOA estimation,” *IEEE Tr. Signal Proc.*, vol. 40, pp. 1828–1832, July 1992.

[5] M. Haardt and J.A. Nossek, “Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieveal problems,” *IEEE Trans. Signal Proc.*, vol. 46, pp. 161–169, Jan. 1998.

[6] A.J. van der Veen, M. Vanderveen, and A. Paulraj, “Joint angle and delay estimation using shift-invariance techniques,” *IEEE Tr. Signal Proc.*, vol. 46, pp. 405–418, Feb. 1998.

[7] N.D. Sidiropoulos, G.B. Giannakis, and R. Bro, “Parallel factor analysis in sensor array processing,” *IEEE Trans. Signal Processing*, vol. 48, pp. 2377–2388, Aug. 2000.

[8] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*. PhD thesis, KU Leuven, Leuven, Belgium, 1997.

[9] K. Abed-Meraim and Y. Hua, “A least-squares approach to joint Schur decomposition,” in *Proc. IEEE ICASSP*, pp. 2541–2544, 1998.

[10] A. Yeredor, “Approximate joint diagonalization using nonorthogonal matrices,” in *Proc. IEEE workshop on ICA*, (Helsinki), June 2000.

[11] M. Viberg, B. Ottersten, and T. Kailath, “Detection and estimation in sensor arrays using weighted subspace fitting,” *IEEE Trans. Signal Proc.*, vol. 39, pp. 2436–2449, Nov. 1991.

[12] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*. London: Academic Press, 1981.

[13] J.-F. Cardoso and E. Moulines, “Invariance of subspace based estimators,” *IEEE Trans. Signal Proc.*, vol. 48, pp. 2495 – 2505, Sept. 2000.