FAST APPROXIMATE JOINT DIAGONALIZATION OF POSITIVE DEFINITE HERMITIAN MATRICES

Koby Todros and Joseph Tabrikian

Dept. of Electrical and Computer Engineering,
Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

ABSTRACT

In this paper, a new efficient iterative algorithm for approximate joint diagonalization of positive-definite Hermitian matrices is presented. The proposed algorithm, named as SVDJD, estimates the diagonalization matrix by iterative optimization of a maximum likelihood based objective function. The columns of the diagonalization matrix is not assumed to be orthogonal, and they are estimated separately by using iterative singular value decompositions of a weighted sum of the matrices to be diagonalized. The performance of the proposed SVDJD algorithm is evaluated and compared to other existing state-of-the-art algorithms for approximate joint diagonalization. The results imply that the SVDJD algorithm is computationally efficient with performance similar to state-of-the-art algorithms for approximate joint diagonalization.

Index Terms— Joint diagonalization, BSS, SVD

1. INTRODUCTION

Consider a set $\mathcal{R}$ of M positive-definite, Hermitian matrices $\mathbf{R}_1, \ldots, \mathbf{R}_M \in \mathbb{C}^{K\times K}$. The set $\mathcal{R}$ is said to be simultaneously diagonalizable if there exists a nonsingular matrix $\mathbf{B} \in \mathbb{C}^{K\times K}$ and M congruent diagonal matrices $\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_M \in \mathbb{R}^{K\times K}$, such that

$$\mathbf{B}\mathbf{R}_m\mathbf{B}^H = \mathbf{\Lambda}_m, \;\; m = 1, \ldots, M. \tag{1}$$

In practice, the set $\mathcal{R}$ is usually unknown and it is estimated from a finite sample size. Due to estimation errors, the estimate of $\mathcal{R}$, denoted by $\hat{\mathcal{R}} = \{\hat{\mathbf{R}}_m\}_{m=1}^M$, is perturbed and exact joint diagonalization of $\mathcal{R}$ may not be achievable. Hence, the problem of approximate joint diagonalization seeks for a matrix $\mathbf{B}$ such that $\{\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H\}_{m=1}^M$ are "as diagonal as possible" in a sense that a measure of deviation from diagonality of $\{\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H\}_{m=1}^M$ is minimized w.r.t. $\mathbf{B}$.

A variety of algorithms for approximate joint diagonalization have been proposed in the literature [1]-[7]. Many blind source separation (BSS) techniques utilize approximate joint diagonalization algorithms [8]-[12]. According to these techniques, a set of unknown matrices, which obey exact joint diagonalization, is estimated from the observed data. A diagonalization matrix $\mathbf{B}$, which is usually the separation matrix, is estimated (up to scaling and permutation of columns) by minimizing an objective function that measures the deviation of $\{\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H\}_{m=1}^M$ from diagonality.

In this paper, a new efficient iterative algorithm for approximate joint diagonalization of positive-definite Hermitian matrices, named as SVDJD, is proposed. The positive-definite assumption is motivated by the fact that in many applications [10], [11], the set $\mathcal{R}$ consists of covariance matrices of some random variables. According to the proposed algorithm, a diagonalization matrix $\mathbf{B}$, whose columns are not constrained to be orthogonal, is estimated by optimization of a maximum likelihood (ML) based objective function, used also by Pham [3]. The columns of $\mathbf{B}$ are estimated separately using iterative singular value decompositions (SVD) of a weighted sum of the matrices to be diagonalized. This property enables low computational load of order $O(MK + K^4)$ per iteration, which is useful especially in cases of large amount of matrices.

2. MAXIMUM LIKELIHOOD BASED OBJECTIVE FUNCTION

In this section, an ML-based objective function for estimation of the diagonalization matrix $\mathbf{B}$ is derived under the following measurements model. Let $\mathbf{X}_m = \left[\mathbf{x}_1^{(m)}, \ldots, \mathbf{x}_{N_m}^{(m)}\right],\ m = 1, \ldots, M$ denote M statistically independent populations of K-variate complex random vectors, where $\mathbf{x}_n^{(m)}$ and N_m denote the nth vector and the sample size of the mth population, respectively. In each population, the vectors $\mathbf{x}_n^{(m)},\ n = 1, \ldots, N_m,\ m = 1, \ldots, M$ are independent and $\mathbf{x} \sim N^c(\boldsymbol{\eta}_m, \mathbf{R}_m)$, where N^c denotes the circular complex Gaussian distribution, $\{\boldsymbol{\eta}_m\}_{m=1}^M$, denotes the mean vectors and the set $\mathcal{R} = \{\mathbf{R}_m\}_{m=1}^M$ denotes a simultaneously diagonalizable family of complex covariance matrices such that (1) is satisfied. Assuming that $\boldsymbol{\eta}_m = 0,\ m = 1, \ldots, M$, the ML estimate of $\mathbf{R}_m$ is given by $\hat{\mathbf{R}}_m = \frac{1}{N_m}\mathbf{X}_m\mathbf{X}_m^H$.

Since $\{\mathbf{X}_m\}_{m=1}^M$ are statistically independent, $\{\hat{\mathbf{R}}_m\}_{m=1}^M$ are also statistically independent and therefore, their joint log-likelihood function is given by

$$L(\mathbf{R}_1, \ldots, \mathbf{R}_M) = -N_m \sum_{m=1}^{M} \left(\log|\mathbf{R}_m| + tr\left(\mathbf{R}_m^{-1}\hat{\mathbf{R}}_m\right)\right) + const. \tag{2}$$

Since the set $\mathcal{R}$ is a simultaneously diagonalizable family, then using (1), the joint log-likelihood of $\mathbf{B}, \mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_M$, after normalization and dropping the constant term, is

$$Q(\mathbf{B}, \mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_M) = \sum_{m=1}^{M} w_m \left(tr\left(\mathbf{B}^H\mathbf{\Lambda}_m^{-1}\mathbf{B}\hat{\mathbf{R}}_m\right) - \log|\mathbf{B}^H\mathbf{\Lambda}_m^{-1}\mathbf{B}|\right) \tag{3}$$

where $w_m = \frac{N_m}{N}$ and $N = \sum_{m=1}^M N_m$ denotes the total sample size. It can be shown that

$$Q(\mathbf{B}, \mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_M) = \sum_{m=1}^{M} w_m KL_{norm}\left(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H|\mathbf{\Lambda}_m\right) \tag{4}$$

where $KL_{norm}\left(\Sigma_1|\Sigma_2\right)$ denotes the Kullback-Leibler divergence between two zero-mean K-variate circular complex normal densities with covariance matrices Σ_1 and Σ_2. The Pythagorean property of the Kullback-Leibler divergence [13] implies that (4) can be decomposed as

$$\begin{aligned} Q(\mathbf{B},\mathbf{\Lambda}_1,\ldots,\mathbf{\Lambda}_M) = \quad & \sum_{m=1}^{M} w_m \\ & \left[KL_{norm}\left(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H|diag(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H)\right)\right. \\ & \left. + KL_{norm}\left(diag(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H)|\mathbf{\Lambda}_m\right)\right] \end{aligned} \tag{5}$$

where $diag(\cdot)$ denotes a diagonal matrix with the same diagonal elements of its argument. Thus, the objective function in (5) is minimized for a fixed value of $\mathbf{B}$ when $\mathbf{\Lambda}_m = diag(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H)$ and the attained minimum is

$$Q^*(\mathbf{B}) = \sum_{m=1}^{M} w_m \left[KL_{norm}\left(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H|diag(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H)\right)\right]. \tag{6}$$

Therefore, we conclude that the normalized log-likelihood of $\mathbf{B}$, $\mathbf{\Lambda}_1,\ldots,\mathbf{\Lambda}_M$ given $\hat{\mathcal{R}}$, leads to an objective function, which measures the deviation of $\{\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H\}_{m=1}^M$ from diagonality. It can be shown that the objective function in (6) can be expressed as

$$Q^*(\mathbf{B}) = \sum_{m=1}^{M} w_m \left[\log\left|diag(\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H)\right| - \log\left|\mathbf{B}\hat{\mathbf{R}}_m\mathbf{B}^H\right|\right]. \tag{7}$$

We note that the same ML-based objective function was obtained in [10] and [11], in the context of blind separation of statistically independent mixed sources for nonstationary and Gaussian mixture model distributed sources, respectively.

3. MINIMIZATION ALGORITHM

In this section, an iterative algorithm for minimization of (7) w.r.t. $\mathbf{B}$ is derived. Direct minimization of (7) w.r.t. $\mathbf{B}$ is analytically cumbersome. Therefore, $\mathbf{B}$ is decomposed into $\mathbf{B} = \breve{\mathbf{B}}\hat{\mathbf{W}}$, where $\hat{\mathbf{W}}$ is a whitening matrix of $\hat{\mathbf{R}} = \sum_{m=1}^{M} w_m\hat{\mathbf{R}}_m$, and

$$\begin{aligned} Q^*(\breve{\mathbf{B}}) = \sum_{m=1}^{M} w_m \left[\log\left|diag(\breve{\mathbf{B}}(\hat{\mathbf{W}}\hat{\mathbf{R}}_m\hat{\mathbf{W}}^H)\breve{\mathbf{B}}^H)\right| - \right. \\ \left. \log\left|(\breve{\mathbf{B}}(\hat{\mathbf{W}}\hat{\mathbf{R}}_m\hat{\mathbf{W}}^H)\breve{\mathbf{B}}^H)\right|\right] \end{aligned} \tag{8}$$

is minimized w.r.t. $\breve{\mathbf{B}}$. Since $\hat{\mathbf{W}}$ and $\hat{\mathbf{R}}_m$ are independent of $\breve{\mathbf{B}}$ (8) can be reduced to the following form

$$Q'(\breve{\mathbf{B}}) = \sum_{m=1}^{M} w_m \left[\log\left|diag(\breve{\mathbf{B}}(\hat{\mathbf{W}}\hat{\mathbf{R}}_m\hat{\mathbf{W}}^H)\breve{\mathbf{B}}^H)\right| - \log\left|\breve{\mathbf{B}}\breve{\mathbf{B}}^H\right|\right]. \tag{9}$$

In [14], it is shown that since $\hat{\mathbf{W}}$ is a whitening matrix of $\hat{\mathbf{R}}$, then $\breve{\mathbf{B}}$ is approximately unitary and thus

$$\begin{aligned} Q'(\breve{\mathbf{B}}) \approx \sum_{m=1}^{M} w_m \left[\log\left|diag(\breve{\mathbf{B}}(\hat{\mathbf{W}}\hat{\mathbf{R}}_m\hat{\mathbf{W}}^H)\breve{\mathbf{B}}^H)\right| - \right. \\ \left. \log\left|diag(\breve{\mathbf{B}}\breve{\mathbf{B}}^H)\right|\right] = Q''(\breve{\mathbf{B}}). \end{aligned} \tag{10}$$

Let $\breve{\mathbf{B}} = \left[\breve{\mathbf{b}}_1,\ldots,\breve{\mathbf{b}}_K\right]$, then (10) can be expressed as

$$Q''(\breve{\mathbf{B}}) = \sum_{m=1}^{M} w_m \sum_{k=1}^{K} \log\left(\tilde{\mathbf{b}}_k^H\hat{\mathbf{W}}\hat{\mathbf{R}}_m\hat{\mathbf{W}}^H\tilde{\mathbf{b}}_k\right), \tag{11}$$

where $\tilde{\mathbf{b}}_k = \frac{\breve{\mathbf{b}}_k}{\|\breve{\mathbf{b}}_k\|_2}$ and $\left\|\tilde{\mathbf{b}}_k\right\|_2 = 1 \ \ \forall k = 1,\ldots,K$. Let $\breve{\mathbf{R}} \triangleq \hat{\mathbf{W}}\hat{\mathbf{R}}_m\hat{\mathbf{W}}^H$, then the objective function becomes

$$Q'''(\tilde{\mathbf{B}}) = \sum_{m=1}^{M}\sum_{k=1}^{K}\left[\log\left(\tilde{\mathbf{b}}_k^H\breve{\mathbf{R}}\tilde{\mathbf{b}}_k\right) - \lambda_k\left(\tilde{\mathbf{b}}_k^H\tilde{\mathbf{b}}_k - 1\right)\right], \tag{12}$$

where $\{\lambda_k\}_{k=1}^K$ are the Lagrange multipliers. Minimization of Q''' w.r.t. $\tilde{\mathbf{B}}$ causes a scaled estimation of $\mathbf{B}$. In BSS applications this inherent limitation is well known and usually tolerable. Equating the partial derivatives of Q''' w.r.t. $\{\tilde{\mathbf{b}}_k^H\}_{k=1}^K$ to zero yields

$$\sum_{m=1}^{M} w_m\left(\tilde{\mathbf{b}}_k^H\breve{\mathbf{R}}_m\tilde{\mathbf{b}}_k\right)^{-1}\breve{\mathbf{R}}_m\tilde{\mathbf{b}}_k = \lambda_k\tilde{\mathbf{b}}_k. \tag{13}$$

Left multiplication of (13) by $\tilde{\mathbf{b}}_k^H$, and using the unity norm constraint on $\tilde{\mathbf{b}}_k$, we obtain $\lambda_k = 1, \ \forall k = 1,\ldots,K$. Therefore, (13) can be rewritten in the form

$$\mathbf{G}(\tilde{\mathbf{b}}_k)\tilde{\mathbf{b}}_k = \tilde{\mathbf{b}}_k, \tag{14}$$

where $\mathbf{G}(\tilde{\mathbf{b}}_k) \triangleq \sum_{m=1}^{M} w_m\left(\tilde{\mathbf{b}}_k^H\breve{\mathbf{R}}_m\tilde{\mathbf{b}}_k\right)^{-1}\breve{\mathbf{R}}_m$ is a Hermitian matrix. Direct solution of (14) is analytically cumbersome. Therefore, an iterative algorithm which minimizes the L_2 norm of the difference between the l.h.s and r.h.s of (14) w.r.t. $\tilde{\mathbf{b}}_k$, is derived. The L_2 norm of $\mathbf{G}(\tilde{\mathbf{b}}_k)\tilde{\mathbf{b}}_k - \tilde{\mathbf{b}}_k$ is

$$\xi(\tilde{\mathbf{b}}_k) = \tilde{\mathbf{b}}_k^H\left(\mathbf{G}(\tilde{\mathbf{b}}_k) - \mathbf{I}\right)^2\tilde{\mathbf{b}}_k. \tag{15}$$

In order to enforce unit L_2 norm on $\tilde{\mathbf{b}}_k$, the following expression is minimized w.r.t. $\tilde{\mathbf{b}}_k$, and $\alpha_k, \ k = 1,\ldots,K$

$$\xi'(\tilde{\mathbf{b}}_k,\alpha_k) = \tilde{\mathbf{b}}_k^H\left(\mathbf{G}(\tilde{\mathbf{b}}_k) - \mathbf{I}\right)^2\tilde{\mathbf{b}}_k - \alpha_k\left(\tilde{\mathbf{b}}_k^H\tilde{\mathbf{b}}_k - 1\right) \tag{16}$$

where α_k is the Lagrange multiplier. Let $\tilde{\mathbf{b}}_k^*$ be defined as

$$\tilde{\mathbf{b}}_k^* = \arg\min_{\breve{\mathbf{b}}_k}\xi'(\tilde{\mathbf{b}}_k,\alpha_k). \tag{17}$$

The direct minimization in (17) is analytically cumbersome. Therefore, Minimization of (16) w.r.t. $\tilde{\mathbf{b}}_k$ is carried out by iterative minimization of the following auxiliary function

$$\psi(\tilde{\mathbf{b}}_k,\tilde{\mathbf{b}}_k^*,\alpha_k) = \tilde{\mathbf{b}}_k^H\left(\mathbf{G}(\tilde{\mathbf{b}}_k^*) - \mathbf{I}\right)^2\tilde{\mathbf{b}}_k - \alpha_k\left(\tilde{\mathbf{b}}_k^H\tilde{\mathbf{b}}_k - 1\right). \tag{18}$$

The matrix $\mathbf{G}(\tilde{\mathbf{b}}_k^*)$ is the zero-order Taylor approximation of $\mathbf{G}(\tilde{\mathbf{b}}_k)$ around $\tilde{\mathbf{b}}_k^*$. Note that (18) is tangent to (16) when $\tilde{\mathbf{b}}_k = \tilde{\mathbf{b}}_k^*$ and therefore, share the same global minimum. Since $\tilde{\mathbf{b}}_k^*$ is unknown, $\tilde{\mathbf{b}}_k^* = \tilde{\mathbf{b}}_k^{old}$ is initially guessed and

$$\tilde{\mathbf{b}}_k^{new} = \arg\min_{\breve{\mathbf{b}}_k}\min_{\alpha_k}\psi(\tilde{\mathbf{b}}_k,\tilde{\mathbf{b}}_k^*,\alpha_k). \tag{19}$$

is obtained. This process is iterated until convergence, where $\tilde{\mathbf{b}}_k^{old}$ of the next iteration is $\tilde{\mathbf{b}}_k^{new}$ of the current one. In each iteration, minimization of $\psi(\tilde{\mathbf{b}}_k,\tilde{\mathbf{b}}_k^*,\alpha_k)$ is carried out by equating its derivative w.r.t. $\tilde{\mathbf{b}}_k$ to zero. Hence, the following equation is solved

$$\left(\mathbf{G}(\tilde{\mathbf{b}}_k^{old}) - \mathbf{I}\right)^2\tilde{\mathbf{b}}_k^{new} = \alpha_k\tilde{\mathbf{b}}_k^{new}. \tag{20}$$

According to (19) and (20), $\tilde{\mathbf{b}}_k^{new}$ is the eigenvector of $\left(\mathbf{G}(\tilde{\mathbf{b}}_k^{old}) - \mathbf{I}\right)^2$, corresponding to the minimal eigenvalue, α_k.

Finally, the iterative algorithm for the estimation of $\tilde{\mathbf{b}}_k$, named as SVDJD comprises the following steps:

1. **Set $k = 1$.**
2. **Let $l = 1$. Initialize $\tilde{\mathbf{b}}_k^{(0)}$.**
3. **Solve $\left(\mathbf{G}(\tilde{\mathbf{b}}_k^{(l-1)}) - \mathbf{I}\right)^2 \tilde{\mathbf{b}}_k^{(l)} = \alpha_k^{(l)} \tilde{\mathbf{b}}_k^{(l)}$ and pick $\tilde{\mathbf{b}}_k^{(l)}$, corresponding to the minimal eigenvalue, $\alpha_k^{(l)}$.**
4. **If $\alpha_k^{(l)} > \epsilon$, then $l = l + 1$ and go to step 3.**
5. **If $\alpha_k^{(l)} \leq \epsilon$, then if $k < K$, then $k = k + 1$ and go to step 2, else stop.**

Convergence and efficient initialization of this algorithm are discussed in detail in [14].

In order to examine the typical averaged convergence patterns of $\xi(\tilde{\mathbf{b}}_k^{(l)})$, a data set with 1000 random realizations of the complex three-dimensional matrix set $\left\{\mathbf{A}\mathbf{\Lambda}_m\mathbf{A}^H + \sigma^2\mathbf{E}_m\mathbf{E}_m^H\right\}_{m=1}^M$ was generated. The elements of the matrix $\mathbf{A}$ are drawn from a complex standard normal distribution and the matrices $\{\mathbf{\Lambda}_m\}_{m=1}^M$ are real diagonal with elements uniformly distributed in $(0, 1]$. The matrices $\{\mathbf{E}_m\}_{m=1}^M$ are perturbation matrices, randomized from a complex normal standard distribution and the scalar σ^2 was set to 0.01. The number of matrices, M, was set to 25 and the matrices weights, $\{w_m\}_{m=1}^M$, were equally set to $\frac{1}{M}$. Fig. 1 depicts the average values of $\log\left(\xi(\tilde{\mathbf{b}}_k^{(l)})\right)$ for $k = 1, 2, 3$, as a function of the iteration index l. One can notice that the typical averaged convergence pattern is exponential and that the algorithm converges after approximately 20-25 iterations in each dimension.

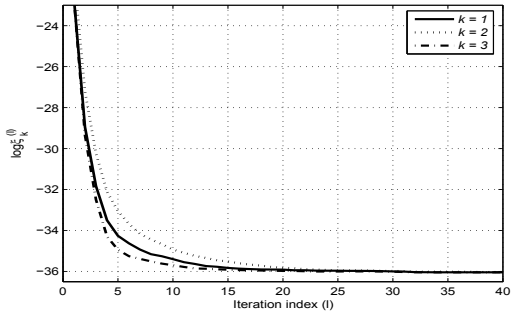


Fig. 1. Typical averaged convergence patterns of the minimization algorithm.

4. COMPUTATIONAL COMPLEXITY

In this section, an asymptotic computational load analysis of the proposed algorithm is presented and compared to other existing state-of-the-art techniques for approximate joint diagonalization. According to the minimization algorithm described in Section 3, for each dimension, denoted by k $(k = 1, \ldots, K)$, in each iteration, the matrix $\left(\mathbf{G}(\tilde{\mathbf{b}}_k) - \mathbf{I}\right)^2$ is calculated and its eigenvectors and eigenvalues are computed using SVD. The computational load of calculating $\left(\mathbf{G}(\tilde{\mathbf{b}}_k) - \mathbf{I}\right)^2$ is of order $O(M)$, and according to [15], the computational load of the SVD algorithm is $O(K^3)$. Therefore, assuming identical number of iterations per dimension, the asymptotic computational load of the algorithm per iteration is $O(MK + K^4)$.

The computational loads per iteration of the proposed method and of other existing techniques for approximate joint diagonalization, like Pham's algorithm [3], the AC/DC [5] and FFDIAG [7] algorithms are presented in Table 1. Observing Table 1, one can notice that the SVDJD is computationally the most efficient algorithm in comparison to Pham's, AC/DC and FFDIAG algorithms, whenever $M > \frac{K^3}{K-1}$. This case may be encountered, for example, when $K = 5$ and $M = 100$.

Table 1. Asymptotic computational load per iteration of the SVDJD, Pham's, AC/DC, and FFDIAG

Algorithm	Computational load
SVDJD	$O(MK + K^4)$
PHAM	$O(MK^2)$
AC/DC	$O(MK^3)$
FFDIAG	$O(MK^2)$

5. SIMULATION RESULTS

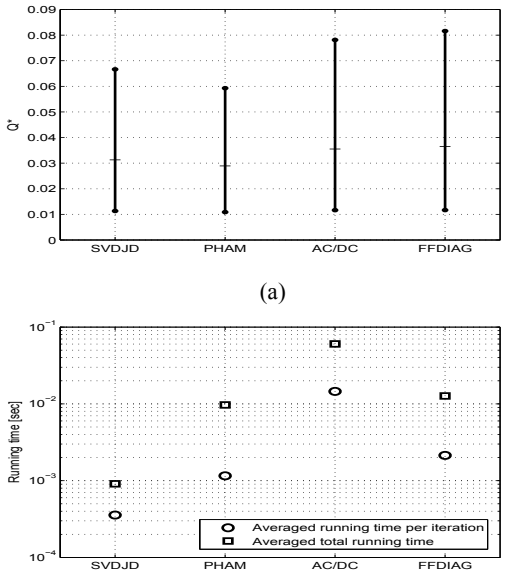
In this section, the performance of the SVDJD algorithm is compared with state-of-the-art algorithms for approximate joint diagonalization, like Pham's [3], AC/DC [5] and the FFDIAG algorithm [7]. The first trial evaluates the performance of the SVDJD algorithm in a scenario of orthogonal joint diagonalization problem. The second trial evaluates the performance of the proposed method in the case of non-orthogonal joint diagonalization problem.

5.1. Orthogonal joint diagonalization

In this test, the performance of the proposed algorithm is evaluated for orthogonal joint diagonalization. The data set consisted of 1000 random realizations of the real two-dimensional matrix set $\left\{\mathbf{A}\left(\mathbf{E}_m\mathbf{\Lambda}_m\mathbf{E}_m^H\right)\mathbf{A}^H\right\}_{m=1}^M$, where $\mathbf{A}$ is a rotation matrix with rotation angle uniformly distributed in $(-180°, 180°]$, the matrices $\{\mathbf{\Lambda}_m\}_{m=1}^M$ are diagonal with elements uniformly distributed in $(0, 1]$ and the matrices $\{\mathbf{E}_m\}_{m=1}^M$ are perturbation rotation matrices with rotation angles uniformly distributed in $(-10°, 10°]$. The number of matrices, M, was set to 50 such that the condition $M > \frac{K^3}{K-1}$ is satisfied. The matrix weights, $\{w_m\}_{m=1}^M$, were equally set to $\frac{1}{M}$. In each realization, the values of the objective function $Q^*(\mathbf{B})$, total running time and total running time per iteration were calculated for the SVDJD, Pham's, AC/DC and FFDIAG algorithms. Calculation of the running period per iteration of the SVDJD algorithm is given by $T_{It} = \frac{T_{Tot}K}{\sum_{k=1}^K L_k}$ where T_{It}, T_{Tot}, and L_k denote the running time per iteration, the total running time, and the number of iterations in the kth dimension, respectively. Calculation of the running time per iteration of Pham's, AC/DC and FFDIAG algorithms was performed by dividing the total running time by the number of iterations. The mean values, 5th and 95th percentiles of $Q^*(\mathbf{B})$, evaluated for each algorithm, are depicted in Fig. 2.a. The average running time per iteration and total running time of each algorithm are depicted in Fig. 2.b. This figure shows that the average total running time of the SVDJD algorithm is significantly shorter than in the other algorithms.

5.2. Non-orthogonal joint diagonalization

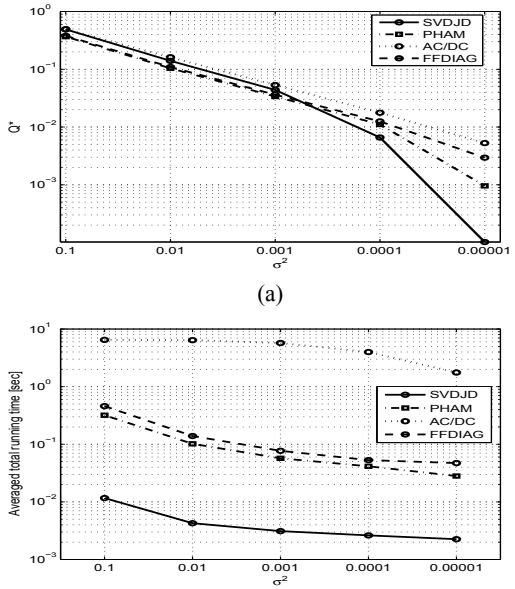
In this test, the performance of the proposed algorithm was evaluated in a scenario of non-orthogonal joint diagonalization problem. The data set consisted of 1000 random realizations of the real five-dimensional matrix set $\left\{\mathbf{A}\mathbf{\Lambda}_m\mathbf{A}^H + \sigma^2\mathbf{E}_m\mathbf{E}_m^H\right\}_{m=1}^M$, where $\mathbf{A}$ is



(a)

(b)

Fig. 2. Orthogonal joint diagonalization a) The mean values, 5th and 95th percentiles of the cost function, $Q^*(\mathbf{B})$. The '–' mark denotes the mean value, and the lower and upper 'o' marks denote the 5th and 95th percentiles, respectively; b) The averaged running time per iteration and the averaged total running time in seconds of the SVDJD, Pham's, AC/DC and FFDIAG algorithms.



(a)

(b)

Fig. 3. Non-orthogonal joint diagonalization: a) The mean values of the cost function, $Q^*(\mathbf{B})$; b) The average total running time in seconds of the SVDJD, Pham's, AC/DC and FFDIAG algorithms.

a random matrix with elements drawn from a standard normal distribution, the matrices $\{\mathbf{\Lambda}_m\}_{m=1}^M$ are diagonal with elements uniformly distributed in $(0, 1]$ and the matrices $\{\mathbf{E}_m\}_{m=1}^M$ are random perturbation matrices drawn from a normal standard distribution. The scalar σ^2 is aimed to control the perturbation level and M was set to 100 such that the condition $M > \frac{K^3}{K-1}$ is satisfied. The matrix weights, $\{w_m\}_{m=1}^M$, were equally set to $\frac{1}{M}$. The mean values, 5th and 95th percentiles of $Q^*(\mathbf{B})$, evaluated for each algorithm, are depicted in Fig. 3.a. It can be seen that in low perturbation levels, the SVDJD algorithm outperforms the other algorithms. The total running time of each algorithm as a function of the perturbation level, σ^2, are depicted in Fig. 3.b. This figure shows that for different perturbation levels, the averaged total running time obtained by the SVDJD algorithm is significantly shorter in comparison to Pham's, AC/DC and FFDIAG algorithms.

6. CONCLUSION

A new efficient algorithm, named SVDJD, for approximate joint diagonalization of positive-definite Hermitian matrices is proposed. The diagonalization matrix, $\mathbf{B}$, is not constrained to be unitary and it is estimated by iterative optimization of an -based objective function. Each column in $\mathbf{B}$ is estimated independently using iterative SVDs of a weighted sum of the matrices to be diagonalized. The fact that each column in $\mathbf{B}$ is estimated independently, enables low computational load, which is practical especially in cases of large amount of matrices. The algorithm is easy to implement and demonstrates good diagonalization performance together with low computational load in comparison to existing state-of-the-art algorithms for approximate joint diagonalization.

7. REFERENCES

[1] B.N. Flury and W.G. Gautschi, "An algorithm for orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form," *SIAM J. Sci. Stat. Comp.*, vol. 7, 1, pp. 169-184, January 1986.

[2] J.F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Mat. Anal. Appl.*, vol. 17, 1, pp. 161-164, January 1996.

[3] D.T. Pham, "Joint approximate diagonalization of positive definite matrices," *SIAM J. Matrix Anal. Appl.*, vol 22, 4, pp. 1136-1152, 2001.

[4] A.J. van der Veen, "Joint diagonalization via subspace fitting techniques," in *Proc. ICASSP'01*, vol. 5, pp. 2773-2776, 2001.

[5] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. on Signal Processing*, vol. 50, 7, pp. 1545-1553, July 2002.

[6] M. Joho and K. Rahbar, "Joint diagonalization of correlation matrices by using Newton methods with application to blind signal separation," *Proc. of IEEE Sensor Array and Multichannel Signal Processing Workshop - SAM*, pp. 403-407, 2002.

[7] A. Ziehe, P. Laskov, G. Nolte, and K.-R Muller, "A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation," *J. of Machine Learning Research*, vol. 5, pp. 777-800, July 2004.

[8] J.F. Cardoso, "Blind signal separation: statistical principles," *Proc. of the IEEE*, vol. 86, 10, pp. 2009-2025, 1998.

[9] J.F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, 1, pp. 157-192, January 1999.

[10] D.T. Pham and J.F. Cardoso, "Blind separation of instantaneous mixtures of non-stationary sources," *IEEE Trans. Signal Processing*, vol. 49, 9, pp. 1837-1848, 2001.

[11] K. Todros and J. Tabrikian, "Blind separation of independent sources using Gaussian mixture model," *IEEE Trans. Signal Processing*, in press.

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

[13] S.I. Amari, "Differential-geometrical methods in statistics," *Lect. Notes in Stat.*, vol. 28 Springer-Verlag, 1985.

[14] K. Todros and J. Tabrikian, "Fast approximate joint diagonalization of positive-definite Hermitian matrices with application in blind source separation," submitted for publication in the *IEEE Trans. Signal Processing*.

[15] R.A. Horn and C.R. Johnson, *Matrix analysis*. Cambridge University Press, Cambridge, 1985.