A proximal approach to IVA-G with convergence guarantees

Clément Cosserat, Ben Gabrielson, Emilie Chouzenoux, Jean-Christophe Pesquet, Tülay Adali

A PROXIMAL APPROACH TO IVA-G WITH CONVERGENCE GUARANTEES

Clément Cosserat[†], *Ben Gabrielson*[‡], *Emilie Chouzenoux*[†], *Jean-Christophe Pesquet*[†], *Tülay Adali*[‡]

† Université Paris-Saclay, CentraleSupélec, Inria, CVN, Gif-sur-Yvette, France
‡Dept of CSEE, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

ABSTRACT

Independent vector analysis (IVA) generalizes independent component analysis (ICA) to multiple datasets, and when used with a multivariate Gaussian model (IVA-G), provides a powerful tool for joint analysis of multiple datasets in an array of applications. While IVA-G enjoys uniqueness guarantees, the current solution to the problem exhibits significant variability across runs necessitating the use of a scheme for selecting the most consistent one, which is costly. In this paper, we present a penalized maximum-likelihood framework for the problem, which enables us to derive a non-convex cost function that depends on the precision matrices of the source component vectors, the main mechanism by which IVA-G leverages correlation across the datasets. By adding a quadratic regularization, a block-coordinate proximal algorithm is shown to offer a suitable solution to this minimization problem. The proposed method also provides convergence guarantees that are lacking in other state-of-the-art approaches to the problem. This also allows us to obtain overall slightly better performance, and in particular, we show that our method yields better estimation in average than the current IVA-G algorithm for various source numbers, datasets, and degrees of correlation across the data.

Index Terms— blind source separation, IVA-G, proximal algorithms, joint ISI, non-convex optimisation

1. INTRODUCTION

Joint blind source separation (JBSS) enables recovery of information across multiple datasets through latent variables, and has proven powerful for extracting useful features from related datasets in a large number of application areas. In particular, by enabling multiple datasets to fully interact, it leverages the joint information across them and has been attractive for data fusion, [1,2]. Independent vector analysis (IVA) [2,3] generalizes independent component analysis (ICA) to multiple datasets. It captures the joint information across the datasets through selection of an appropriate multivariate probability density model, for the source component vectors (SCVs), the key element of the IVA formulation. When a multivariate Gaussian density model is used, it is termed IVA-G, and the method enables effective fusion of multiple datasets using only second-order statistics [2, 4], and also generalizes multiset canonical correlation analysis (MCCA), which has been widely used in an array of applications [5]. In addition, IVA provides identifiability under rather general conditions [4,6]. However, one of its main difficulties is that the cost function is non convex, which makes it necessary to use of a criterion to evaluate the reproducibility (stability) of the solutions and pick one as the best run among many obtained with random initializations [7]. The variability in the solution space becomes a bigger issue when operating in the neighborhood of the local stability conditions, increasing the variability in the solution space. Hence, a solution with convergence guarantees is highly desirable.

In this paper, we propose and analyze a new estimator for performing IVA-G by using a penalized maximum likelihood approach. A proximal alternating minimization algorithm is used to minimize the resulting cost function, which alternates over updates on the covariance matrices of the SCVs and on the mixing coefficients. Convergence guarantees are provided based on recent tools from non smooth analysis. As we demonstrate through simulations in a number of relevant scenarios, the new algorithm exhibits a greater level of accuracy than IVA-G in the estimation of the ground truth sources.

The paper is organized as follows. In Section 2, we formulate the problem and describe the data model and the parameters involved. We derive a cost function starting from the maximization of a likelihood criterion. Section 3 introduces our proximal alternating algorithm, details its updates, and discusses its convergence properties. Finally, we describe our experimental protocol, and discuss the results we obtained in Section 4, before concluding in Section 5.

2. JBSS PROBLEM FORMULATION

We first formulate the problem of JBSS, and consider the case when the latent sources are modeled as random vectors. In this model, a signal is a sequence of $T \in \mathbb{N}$ realisations of its source. There are K observed datasets ($k \in \{1, \dots, K\}$). Within each dataset, there exist N observed signals ($n \in \{1, \dots, N\}$). We denote the N observed signals for a sample $t \in \{1, \dots, T\}$ within the k-th dataset by the vector $x^{[k]}(t) = [x_1^{[k]}(t), \dots, x_N^{[k]}(t)]^\top \in \mathbb{R}^N$ (here, $(.)^\top$ denotes the transpose). These signals are obtained by the linear mixing of N latent source signals $s^{[k]}(t) = [s_1^{[k]}(t), \dots, s_N^{[k]}(t)]^\top \in \mathbb{R}^N$, which we aim to jointly estimate.

The linear mixing of sources within each dataset is represented by unknown invertible matrix $A^{[k]} = \left(a_{i,j}^{[k]}\right)_{1 \le i,j \le N} \in \mathbb{R}^{N \times N}$. The generative model is given as

$$x^{[k]}(t) = A^{[k]}\, s^{[k]}(t), \quad k \in \{1, \dots, K\}. \tag{1}$$

Since the K datasets are observed over T samples of the data, the datasets are themselves represented by matrices $X^{[k]} = [x_1^{[k]}, \dots, x_N^{[k]}]^\top \in \mathbb{R}^{N \times T}$ where, for every $k \in \{1, \dots, K\}$ and $n \in \{1, \dots, N\}$, $x_n^{[k]} = [x_n^{[k]}(1), \dots, x_n^{[k]}(T)]^\top \in \mathbb{R}^T$. Using a similar notation, the corresponding source signals are given by $S^{[k]} = [s_1^{[k]}, \dots, s_N^{[k]}]^\top \in \mathbb{R}^{N \times T}$.

The goal of JBSS is to estimate the source signals within each dataset, which is done by estimating demixing matrices

E.C. acknowledges support from the ERC Starting Grant MAJORIS ERC-2019-STG-850925. Part of the work of J.-C. P. was supported by the ANR Chair in AI, BRIDGEABLE. This work was also supported in part by NSF-NCS 1631838, and NIH grants R01 MH118695, R01 MH123610, R01 AG073949.

$W^{[k]} \in \mathbb{R}^{N\times N}$ that demix each dataset $X^{[k]}$ into the source signals estimates $Y^{[k]} = W^{[k]}X^{[k]} = W^{[k]}A^{[k]}S^{[k]}$, with $Y^{[k]} = [y_1^{[k]}, \ldots, y_N^{[k]}]^\top \in \mathbb{R}^{N\times T}$. Source signals are perfectly estimated when the mixing-demixing matrices $W^{[k]}A^{[k]}$ can be expressed as permuted diagonal matrices, indicating perfect estimation subject to permutation and scale ambiguities. The nth row of $W^{[k]}$ is given by the vector $w_n^{[k]} \in \mathbb{R}^N$, which is used to obtain the nth estimated source signal in the kth dataset, given by $y_n^{[k]\top} = w_n^{[k]\top}X^{[k]}$, with $y_n^{[k]} \in \mathbb{R}^T$.

JBSS exploits dependence across the datasets by modeling dependence through the sources. Across the K datasets, sources of the same index n are modeled as dependent to other sources of index n, thus forming N sets of K dependent sources. In IVA terminology, each of these sets is referred to as a "source component vector" (SCV). The nth SCV is given by $\mathbf{s}_n = [\mathbf{s}_n^{[1]}, \ldots, \mathbf{s}_n^{[K]}]^\top$[1], which is a K-variate random vector. The observed signals are thus realisations of the random vector $\mathbf{x}_n = [\mathbf{x}_n^{[1]}, \ldots, \mathbf{x}_n^{[K]}]^\top$ where $\forall k \in \{1, \ldots, K\} : \mathbf{x}_n^{[k]} = \sum_{m=1}^N a_{n,m}^{[k]}\mathbf{s}_m^{[k]}$. Sources $\mathbf{s}_n$ are estimated by $\mathbf{y}_n = [\mathbf{y}_n^{[1]}, \ldots, \mathbf{y}_n^{[K]}]^\top$ which are similarly modeled by the mixing of the $\mathbf{x}_n$ by the demixing matrices. Over T samples of data (i.e., T realizations), the nth SCV $\mathbf{s}_n$ is represented through the matrix $S_n = [s_n^{[1]}, \ldots, s_n^{[K]}]^\top \in \mathbb{R}^{K\times T}$, and is estimated from $Y_n = [y_n^{[1]}, \ldots, y_n^{[K]}]^\top \in \mathbb{R}^{K\times T}$. Each SCV is modeled as independent from all other SCVs, thus any two sources across the datasets are modeled as dependent only if they correspond to the same source index n (nth SCV).

For each of the SCVs $\mathbf{s}_n, n \in \{1, \ldots, N\}$, we denote covariance $\Sigma_n = \mathrm{E}\{\mathbf{s}_n\,\mathbf{s}_n^\top\} \in \mathbb{R}^{K\times K}$ as the covariance of $\mathbf{s}_n$, and $C_n = \Sigma_n^{-1}$ as the corresponding precision matrix of $\mathbf{s}_n$. We similarly define $\hat{\Sigma}_n \in \mathbb{R}^{K\times K}$ as the covariance of $\mathbf{y}_n$, and $\hat{C}_n = \hat{\Sigma}_n^{-1}$ as the precision matrix of $\mathbf{y}_n$.

2.1. IVA with multivariate Gaussian model

IVA is a method of JBSS that estimates $W^{[k]}$, for $k \in \{1, \ldots, K\}$, such that the corresponding SCV estimates are maximally independent. IVA does this by minimizing the mutual information across the SCVs, which is dependent on the probability distribution function (PDF) assumed on each of the sources. Typical implementations model each SCV by a single multivariate distribution, allowing mutual information of the SCVs $\mathcal{I}\{\mathbf{y}_n\}$ to be written with respect to the marginal entropy of each individual source $\mathcal{H}\{\mathbf{y}_n^{[k]}\}$ and the joint entropy of the entire SCV $\mathcal{H}\{\mathbf{y}_n\}$, via $\mathcal{I}\{\mathbf{y}_n\} = \sum_{k=1}^K \mathcal{H}\{\mathbf{y}_n^{[k]}\} - \mathcal{H}\{\mathbf{y}_n\}$.

IVA assuming a multivariate Gaussian PDF for each SCV (IVA-G) [8] can be considered the simplest of the IVA algorithms, in that it exploits only correlation between sources as the measure of dependence. By denoting $\boldsymbol{\mathcal{W}}$ as the collection of matrices $W^{[k]}$ for $k \in \{1, \ldots, K\}$, the IVA-G cost function is given by:

$$\mathcal{J}_{\text{IVA-G}}(\boldsymbol{\mathcal{W}}) = \frac{NK\log(2\pi e)}{2} + \frac{1}{2}\sum_{n=1}^N \log\left|\det\left(\hat{\Sigma}_n\right)\right| - \sum_{k=1}^K \log\left|\det\left(W^{[k]}\right)\right|. \quad (2)$$

[1] throughout this paper, bold characters represent random variables, and upper cases letters represent matrices

To derive our own variant of IVA-G for our proximal approach, we revisit the general cost for maximum likelihood IVA where now we additionally estimate the corresponding parameters that parametrize the PDFs of the SCVs. For the Gaussian case, these parameters are the covariances $\hat{\Sigma}_n$, or alternatively represented by the corresponding precision matrices $\hat{C}_n$. It is straightforward to show that by applying a change of variable to the general IVA cost, assuming a multivariate Gaussian PDF for each SCV, then applying independence across the SCVs and across the T samples, the IVA-G cost function can alternatively be expressed by the following up to additive constant:

$$\mathcal{J}_{\text{IVA-G}}(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}}) = -\sum_{k=1}^K \log\left|\det\left(W^{[k]}\right)\right| + \sum_{n=1}^N \frac{1}{2}\left(\mathrm{tr}\left(\hat{C}_n W_n \hat{\Lambda} W_n^\top\right) - \log\left|\det\left(\hat{C}_n\right)\right|\right). \quad (3)$$

where we use the following notation:

- $\boldsymbol{\mathcal{C}}$ is the collection of all $\hat{C}_n$, for $n \in \{1, \ldots, N\}$.
- $W_n \in \mathbb{R}^{K\times KN}$ is a matrix corresponding to the nth SCV, defined as $W_n = \left(W_n^{[1]}, \ldots, W_n^{[K]}\right)$, a block matrix obtained by horizontal concatenation of the $W_n^{[k]} \in \mathbb{R}^{K\times N}$ where $W_n^{[k]}$ has $w_n^{[k]}$ as its k-th line and zeros elsewhere.
- $\hat{\Lambda} \in \mathbb{R}^{KN\times KN}$ is the empirical covariance matrix of vector $\mathrm{vec}(\mathbf{x}) \triangleq [(\mathbf{x}^{[1]})^\top, \ldots, (\mathbf{x}^{[K]})^\top]^\top \in \mathbb{R}^{NK}$, where $\mathbf{x}^{[k]} \triangleq [(\mathbf{x}_1^{[k]})^\top, \ldots, (\mathbf{x}_N^{[k]})^\top]^\top$ is the random vector whose realisations produce the columns of the dataset $X^{[k]}$ and $\mathrm{vec}(\mathbf{x})$ is the vertical concatenation of these random vectors.

This yields an alternative but equivalent cost to (2) where we also measure cost as a function of $\hat{C}_n$. Here we assume that $\log|\det(\hat{C}_n)| = \infty$ if $\hat{C}_n$ is not positive definite.

Both the cost functions in (2) and (3) are non-convex, which makes minimization challenging. In [8], the authors minimize (2) by proposing a block-Newton descent method for iteratively updating $W^{[k]}$. Despite this method's practical performance, it does not benefit from proven convergence guarantees, and may suffer from instabilities dependent on the data. We thus introduce a regularized version of the problem by instead optimizing over (3), and build a structured optimization method to tackle the problem with the explicit goal of proven convergence.

2.2. Taking into account ambiguities

As noted in Section 2, there exists scaling ambiguity that is unresolved by demixing matrices $W^{[k]}$, that is, for any $\alpha_n^{[k]} \in \mathbb{R}\setminus\{0\}$, simultaneously multiplying $\mathbf{s}_n^{[k]}$ and $w_n^{[k]}$ by $\alpha_n^{[k]}$ does not change the cost. This scaling ambiguity ultimately results in an unbounded set of minima of the cost function (3). This may introduce some misbehaviour of minimization algorithms as the norm of the estimated $W^{[k]}$ may tend to zero or infinity. A way to reduce this to only a sign ambiguity thus consists of constraining the diagonal components of $\hat{C}_n$ to be equal to a given value (e.g. 1). To this end, we add a quadratic regularization term in our cost function, thus leading to a regularized version of (3):

$$\mathcal{J}_{\text{IVA-G}}(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}}) = -\sum_{k=1}^{K} \log \left| \det\left(W^{[k]}\right) \right| + \sum_{n=1}^{N} \frac{1}{2} \left(\text{tr}\left(\hat{C}_n W_n \hat{\Lambda} W_n^\top\right) - \log \left| \det\left(\hat{C}_n\right) \right| + \sum_{k=1}^{K} \alpha \left(\hat{C}_n(k,k) - 1\right)^2 \right) \quad (4)$$

where we denote $\hat{C}_n(k,k)$ as the kth diagonal element of $\hat{C}_n$, and $\alpha > 0$ is a regularization hyperparameter.

3. OPTIMIZATION ALGORITHM

We now present our optimization scheme to minimize (4). Our method relies on the PALM algorithm (Proximal Alternating Linearized Minimization) [9]. We first present the general form of PALM, and then specify it applied to (4).

3.1. General form

The PALM algorithm aims at finding a critical point of a function of the form:

$$(x_1, \ldots, x_p) \mapsto \Psi(x_1, \ldots, x_p) = H(x_1, \ldots, x_p) + \sum_{k=1}^{p} f_k(x_k)$$

where H is a smooth function defined on $\mathbb{R}^{d_1 \times \ldots \times d_p}$ and each f_k is a function defined on $\mathbb{R}^{d_k}$. An advantage of PALM is that none of the involved functions need to be convex.

Given an initialization $(x_1^0, \ldots, x_p^0) \in \mathbb{R}^{d_1 \times \ldots \times d_p}$, the PALM algorithm iterates, for iteration index i, over the blocks of variables x_k for $k \in \{1, \ldots, p\}$. Specifically, it generates $x^i = (x_1^i, \ldots, x_p^i)$ such that, for $k \in \{1, \ldots, p\}$,

$$x_k^{i+1} \in \text{prox}_{c_k^i}^{f_k}\left(x_k^i - \frac{1}{c_k^i} \nabla_k H(x_1^{i+1}, \ldots, x_{k-1}^{i+1}, x_k^i, \ldots, x_p^i)\right)$$

where $c_k^i = \gamma_k L_k^i$, L_k^i is a Lipschitz modulus of the gradient function

$$x \in \mathbb{R}^{d_k} \mapsto \nabla_k H(x_1^{i+1}, \ldots, x_{k-1}^{i+1}, x, x_{k+1}^i, \ldots, x_p^i),$$

and $\gamma_k > 1$. Under some mild technical assumptions [9], if Ψ is regular enough, the sequence generated by PALM has two interesting properties: the sequence $(x^i)_{i \geq 1}$ converges toward a critical point of Ψ and the sequence of their cost decreases. Note that variable metrics variants of the PALM algorithms have been proposed in [10] and shown to be effective to solve various inverse problems [11, 12].

3.2. Application to IVA-G

PALM can be applied to the minimization of (4), by defining the functions within (4) operating on $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{C}}$:

$$H(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}}) = \sum_{n=1}^{N} \frac{1}{2} \left(\text{tr}\left(\hat{C}_n W_n \hat{\Lambda} W_n^\top\right) + \sum_{k=1}^{K} \alpha \left(\hat{C}_n(k,k) - 1\right)^2 \right), \quad (5)$$

$$f(\boldsymbol{\mathcal{W}}) = -\sum_{k=1}^{K} \log \left| \det\left(W^{[k]}\right) \right|, \quad (6)$$

$$g(\boldsymbol{\mathcal{C}}) = -\frac{1}{2} \sum_{n=1}^{N} \log \left| \det\left(\hat{C}_n\right) \right|. \quad (7)$$

Let us discuss the practical implementation of PALM in our context. First, we can express the gradient of (5) with respect to $\boldsymbol{\mathcal{W}}$, or $\nabla_{\boldsymbol{\mathcal{W}}} H(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}})$, by considering the $W^{[k]}$ blocks separately as:

$$\frac{\partial H(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}})}{\partial W^{[k]}} = \left(\frac{\partial H(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}})}{\partial w_{n,m}^{[k]}}\right)_{1 \leq n,m \leq N} = \begin{bmatrix} [\hat{C}_1 W_1 \hat{\Lambda}]_{k,(k-1)N+1} & \ldots & [\hat{C}_1 W_1 \hat{\Lambda}]_{k,(k-1)N+N} \\ \vdots & & \vdots \\ [\hat{C}_N W_N \hat{\Lambda}]_{k,(k-1)N+1} & \ldots & [\hat{C}_N W_N \hat{\Lambda}]_{k,(k-1)N+N} \end{bmatrix}. \quad (8)$$

Similarly, $\nabla_{\boldsymbol{\mathcal{C}}} H(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}})$ is calculated for each $\hat{C}_n$ block as:

$$\frac{\partial H(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}})}{\partial \hat{C}_n} = \frac{1}{2} W_n \hat{\Lambda} W_n^\top + \alpha \left(\text{diag}(\hat{C}_n) - I_K\right). \quad (9)$$

From these expressions, we can derive the Lipschitz moduli $L_{\boldsymbol{\mathcal{W}}}^i = \|\hat{\Lambda}\|_S \max_{1 \leq n \leq N} \|\hat{C}_n^i\|_S$ and $L_{\boldsymbol{\mathcal{C}}}^i = \alpha$ where $\hat{C}_n^i$ is the estimate of $\hat{C}_n$ at iteration i and $\|\cdot\|_S$ denotes the spectral norm, that is the largest spectral value of the symmetric definite positive matrix it applies to.

Let us discuss the calculation of the proximal operators of (6) and (7), respectively. We can separate the blocks into sub-blocks and calculate, for every k and n, the proximal operator of $f_k(W^{[k]}) = -\log|\det W^{[k]}|$ and $g_n(\hat{C}_n) = -\frac{1}{2}\log\det\hat{C}_n$. Both these functions are spectral, i.e., they only depend on the singular values of the matrices. Function g_n corresponds to a convex function of a symmetric matrix for which the proximity operator is known [13]. Function f_k is non-convex, but its proximity operator can be derived very similarly to the previous one by performing a singular value decomposition of $W^{[k]}$. We thus have all the analytical formulas needed to implement the iterations of PALM algorithm.

Regarding the convergence analysis, we can easily establish the existence of a minimizer. In addition, the above formulas combined with Hadamard and Jensen inequalities allow to prove that (4) satisfies assumptions for the convergence of the PALM algorithm. We can thus guarantee that the generated sequence, assuming it is bounded, will converge to a critical point of the cost function.

4. EXPERIMENTS

We tested the performance of our algorithm on simulated data over K equal to 2 or 5 datasets, and a number of sources N equal to 3, 5, or 10. For each pair (K, N), we randomly generated 100 simulations of data defined by N covariance matrices and K mixing matrices. The models were used to simulate SCVs $\mathbf{s}_n$, and sample the source datasets $S^{[k]}$, which were then mixed by $A^{[k]} \in \mathbb{R}^{N \times N}$ to generate the observed datasets $X^{[k]}$. Each datasets was composed of $T = 10^4$ samples. The coefficients of the $A^{[k]}$ were generated according to random realizations of the zero mean unit variance normal distribution. The SCV covariance matrices Σ_n were generated according to random realizations from the Wishart distribution $W_p(V, n)$, where

$n = K+10$ and $V \in \mathbb{R}^{K\times K}$ is a matrix whose diagonal coefficients are 1 and off-diagonal coefficients are all equal to $\rho \in [0, 1]$. Those forms for V in Wishart distribution allow us to control the typical correlation within the SCVs across the datasets, which is high if ρ is close to 1 and low if ρ is close to 0.

For each simulation of the data, we compared the performance of PALM-IVA with $\alpha = 1$ to IVA-G with a vector-gradient update (IVA-G-V), and IVA-G with a block-Newton update (IVA-G-B), both of which are introduced in [8] and have matlab implementation available at https://mlsp.umbc.edu/resources.html. We use joint inter-symbol-interference (joint-ISI) used in [8] as the metric to compare separation performance of JBSS algorithms' demixing matrices $W^{[k]}$ when the true mixing matrices $A^{[k]}$ are known. Joint-ISI is normalized in [0 1], and jointly measures distance of each dataset's "mixing-demixing" matrix $W^{[k]}A^{[k]}$ to a permuted diagonal matrix, with 0 joint-ISI representing perfect separation performance. The three algorithms were used with a stopping criteria of 10^{-4} to ensure that the results were not biased, and we also added a limit of 5000 iterations.

We repeated twice the experiment described above: once with $\rho = 0.2$ to simulate low correlation within an SCV, representing a lower level of dependence across the datasets, and once with $\rho = 0.8$ to simulate high correlation within an SCV (and across the datasets). Indeed, it is known that IVA-G algorithms usually perform better if the datasets are strongly correlated, that is why we also tested the influence of the correlation across dataset on the performance of PALM-IVA. These experiments have been implemented in Python 3.10.8.

The results of the 100 simulations are displayed in Tables 1 and 2. We indicated for each algorithm and for each pair (K, N) the average computation time t in seconds, the mean ISI score μ_{ISI} and the standard deviation of the scores σ_{ISI}. For the mean and the standard deviation, we noted the best performance in bold characters.

Table 1: Results for low correlation across datasets ($\rho = 0.2$).

	IVA-G-B					
	$K=2$			$K=5$		
	$N=3$	$N=5$	N=10	$N=3$	$N=5$	N=10
t (s)	0.05	0.16	0.55	0.13	0.35	4.05
μ_{ISI}	0.408	0.308	0.279	0.035	0.021	0.016
σ_{ISI}	0.189	0.124	0.065	0.061	0.018	0.007
	IVA-G-V					
	$K=2$			$K=5$		
	$N=3$	$N=5$	N=10	$N=3$	$N=5$	N=10
t (s)	0.37	0.61	1.06	0.40	0.9	3.41
μ_{ISI}	0.137	0.126	0.115	**0.015**	0.015	**0.013**
σ_{ISI}	0.097	0.072	0.034	0.033	**0.007**	**0.003**
	PALM-IVA					
	$K=2$			$K=5$		
	$N=3$	$N=5$	N=10	$N=3$	$N=5$	N=10
t (s)	1.03	2.06	4.96	0.66	1.57	13.96
μ_{ISI}	**0.100**	**0.102**	**0.113**	**0.015**	**0.014**	0.014
σ_{ISI}	**0.084**	**0.056**	**0.033**	**0.021**	0.011	0.006

We see that PALM-IVA always achieves a lower ISI score than IVA-G-B in average, especially if there are few datasets (e.g., $K = 2$). Generally speaking, a joint ISI score greater than 0.1 indicates poor performance to estimate $W^{[k]}$. Therefore, our method enables to find a proper solution to the IVA-G problem when there are only 2 datasets which are barely correlated. The standard deviation is also smaller for PALM-IVA, which confirms that this method is more consistent than IVA-G-B. PALM-IVA also achieves overall better performance than IVA-G-V, but these algorithms perform similarly when $K = 5$. The computation time however is the downside of PALM-IVA which, in our Python implementation, is about 10 times slower than IVA-G-B and 3 times slower than IVA-G-V. Despite this, computation time using PALM-IVA may be significantly improved by initializing PALM-IVA with an efficient JBSS algorithm, such as MCCA or the faster variants of IVA-G.

Table 2: Results for high correlation across datasets ($\rho = 0.8$).

	IVA-G-B					
	$K=2$			$K=5$		
	$N=3$	$N=5$	N=10	$N=3$	$N=5$	N=10
t (s)	0.08	0.23	0.83	0.20	0.52	4.47
μ_{ISI}	0.252	0.210	0.178	0.011	0.009	0.008
σ_{ISI}	0.195	0.108	0.057	0.006	0.009	0.001
	IVA-G-V					
	$K=2$			$K=5$		
	$N=3$	$N=5$	N=10	$N=3$	$N=5$	N=10
t (s)	0.59	0.98	1.76	0.99	1.83	4.83
μ_{ISI}	0.094	0.081	**0.070**	0.009	0.008	0.007
σ_{ISI}	0.099	0.054	**0.024**	0.002	0.001	**0.006**
	PALM-IVA					
	$K=2$			$K=5$		
	$N=3$	$N=5$	N=10	$N=3$	$N=5$	N=10
t (s)	1.37	2.93	7.54	1.74	3.53	22.07
μ_{ISI}	**0.064**	**0.071**	0.074	**0.007**	**0.007**	**0.007**
σ_{ISI}	**0.071**	**0.051**	0.029	**0.002**	**0.001**	0.001

In the case of high correlation where JBSS is expected to be most useful, PALM-IVA still performs better than IVA-G-B and IVA-G-V in most scenarios. Namely, the mean ISI score is 2 to 3 times smaller with PALM-IVA than with IVA-G-B if $K = 2$.

5. CONCLUSION

We proposed a new algorithm based on alternating block proximal descent, which provides convergence guarantees to a critical point of the proposed IVA-G cost function, and performs competitively with respect to previous state-of-the art approaches to IVA-G.

In our future work, we would like to improve the computational efficiency of our approach, and try to adapt the PALM algorithm to versions of IVA with other assumed multivariate distributions, such as the multivariate Laplacian distribution assumed within IVA-L. Another lead would be to consider more general constrained formulations as alternative solutions to the regularization on the diagonal coefficients of the precision matrices we used. Finally, we would also like to consider applications of PALM-IVA to real data, such as fMRI datasets typically modeled by IVA.

6. REFERENCES

[1] D. Lahat, T. Adalı, and C. Jutten, "Multimodal data fusion: A methodological overview methods, challenges and prospectives," *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015.

[2] T. Adalı, M. Anderson, and G.-S. Fu, "Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging," *IEEE Signal Proc. Mag.*, vol. 31, no. 3, pp. 18–33, May 2014.

[3] T. Kim, I. Lee, and T.-W. Lee, "Independent vector analysis: Definition and algorithms," in *Proc. 40th Asilomar Conf. Signals, Systems, Comput.*, 2006, pp. 1393–1396.

[4] M. Anderson, X.-L. Li, and T. Adalı, "Joint blind source separation with multivariate Gaussian model: Algorithms and performance analysis," *IEEE Trans. Signal Processing*, vol. 60, no. 4, pp. 2049–2055, April 2012.

[5] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, Dec. 1971.

[6] J. Via, M. Anderson, X.-L. Li, and T. Adalı, "Joint blind source separation from second-order statistics: Necessary and sufficient identifiability conditions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011, pp. 2520 – 2523.

[7] T. Adali, F. Kantar, M. A. B. S. Akhonda, S. C. Strother, V. D. Calhoun, and E. Acar, "Reproducibility in matrix and tensor decompositions: Focus on model match, interpretability, and uniqueness," *IEEE Signal Processing Magazine*, 2022.

[8] M. Anderson, T. Adali, and X.-L. Li, "Joint Blind Source Separation With Multivariate Gaussian Model: Algorithms and Performance Analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1672–1683, Apr. 2012.

[9] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1, pp. 459–494, 2014.

[10] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "A block coordinate variable metric forward–backward algorithm," *Journal of Global Optimization*, vol. 66, no. 3, pp. 457–485, 2016.

[11] A. Repetti, M. Q. Pham, L. Duval, E. Chouzenoux, and J.-C. Pesquet, "Euclid in a taxicab: Sparse blind deconvolution with smoothed l1/l2 regularization," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 539–543, 2014.

[12] F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot, and L. Laborelli, "An alternating proximal approach for blind video deconvolution," *Signal Processing: Image Communication*, vol. 70, pp. 21–36, 2019.

[13] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2 edition, 2017.