Dimensionality reduction of hyperspectral images using an ICA-based stochastic second-order optimization algorithm

Daniela Lupu[1], Ion Necoara[1], Tudor C. Ionescu[1], Liliana Ghinea[2], Joseph Garrett[3] and Tor A. Johansen[3]

***Abstract*— Hyperspectral imaging is one of the advanced remote sensing techniques whose goal is to obtain the spectrum for each pixel in the image of a scene, with the purpose of finding objects, identifying materials or detecting processes. However, the high dimensional nature of hyperspectral images makes their analysis complex. Various methods have been developed to reduce the dimension of hyperspectral images. Most commonly used dimension reduction techniques are Principal Component Analysis (PCA) and Independent Component Analysis (ICA). PCA is a method to reduce the dimensionality by removing the correlation among the bands, while ICA finds additively independent components. FastICA is one of the most used ICA algorithms. It is based on maximizing a loss derived from the fourth order statistical moment (kurtosis) or negentropy, which are both non-convex functions. Moreover, FastICA can find irrelevant stationary points (no maxima) and is not scalable as it uses at each iteration the whole set of pixels. In this paper, we present a stochastic second-order Taylor-based algorithm adapted to such ICA non-convex loss functions. Our algorithm guarantees ascent, hence it usually identifies (local) maxima. Moreover, the algorithm since it is stochastic, is scalable. Detailed numerical simulations show the superior performance of our method compared to FastICA.**

I. Introduction

The hyperspectral imaging consists of acquiring and processing information from the electromagnetic spectrum by targeting a subset of wavelenghts that span beyond the usual RGB spectrum. The hyperspectral data is becoming a valuable tool for monitoring the Earth's surface or human body and are used in many applications ranging from environment, health, agriculture to astronomy and chemical imaging [6]. In the last two decades, a vast number of techniques were proposed for image processing. Still, the majority of these methods have been designed for application to colour and/or grayscale images; therefore, they have limited success when they are applied to hyperspectral images. This is partially owing to large hyperspectral datasets being difficult to collect, process and analyze, and also to the heavy computations associated with images captured using many spectral bands.

The fast growth of the Artificial Intelligence (AI) and Big Data fields, with support from hardware development, facilitates the emergence of the next generation of Hyperspectral Imaging Systems (HypIS), which are expected to learn from streams of data and take optimal decisions in real-time on the process at hand, leading to increased performance. However, in order to move such technologies into HypIS, bottlenecks have to be overcome, e.g., today computation for HypIS is not locally embedded due to the huge dimension of the hyperspectral data cubes which yields enormous computational load for the existing algorithms, requiring powerful data centres or cloud infrastructures. Moreover, in hyperspectral imaging, we usually deal with a small set of samples and a large feature space, and the curse of dimensionality (Hughes effect) becomes inevitable. Finally, high-dimensional data is a source of variation and redundancy in hyperspectral images. Some of the information in the image may be redundant for analysis, due to strong correlation in the bands. Therefore, dimensionality reduction is needed for hyperspectral image analysis, see also [4], [6], [16]. The main methods for dimensionality reduction of hyperspectral images are Principal Component Analysis (PCA) [17] and Independent Component Analysis (ICA) [7].

PCA is a method to reduce the dimensionality by removing the correlation among the bands and identifying the optimum linear combination of the original bands accounting for the variation of pixel values in an image [10]. More precisely, the principal components are ordered according to variance, such that the first components carry more information with respect to the full data space than the later components [17]. If the variables or spectral signal under observation carry additive independent normally distributed noise, PCA is an optimal method for noise filtering. However, since the criterion used by PCA is variance, which is used to measure second-order statistics, such method may not be effective when the noise is not normally distributed, see [4].

Similar to PCA, ICA is another method to reduce the dimensionality of data, but finds additively separable components rather than successive approximations [7]. More precisely, ICA attempts to decompose a multivariate signal into independent non-Gaussian signals, i.e. a decomposition that provides statistical independence between the estimated components. When the statistical independence assumption is correct, ICA separation of a mixed signal gives good results [4]. Whilst PCA is computed based on second order statistical moments when estimating the subspace, ICA utilizes higher order statistical moments, e.g. kurtosis (the fourth order statistical moment). Due to this, ICA has good performance in reducing the effects of noise and other

*The research leading to these results has received funding from the NO Grants 2014–2021, under project ELO-Hyp, contract no. 24/2020.

[1]Automatic Control and System Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania, Emails: {daniela.lupu, ion.necoara, tudor.ionescu}@upb.ro.

[2]Automatic Control Department, Dunarea de Jos University of Galati, Romania, Email: liliana.ghinea@ugal.ro.

[3]Norwegian University of Science and Technology, Trondheim, Norway, Emails: {joseph.garrett, tor.arne.johansen}@ntnu.no.

forms of undesired interference with the observed spectral signatures, enhancing the classification and detection rate, see e.g. [4], [15], [16].

In what follows, we explore and analyse optimization-based methods for ICA. In the literature there are many iterative algorithms available for finding Independent Components (ICs). Commonly used one, including in industrial applications, is the FastICA algorithm, developed in [7], which maximizes the kurtosis and is based on fixed point iterations derived from the KKT conditions. Other methods for computing the independent components are e.g., Joint Approximation Diagonalization of Eigenmatrices (JADE) [5], Infomax [2] and stochastic majorization-minimization [1]. Infomax is based on a loss function which is a non-convex log-likelihood. In [1], a new majorization-minimization optimization algorithm is developed, which is adapted to the Infomax loss function and guarantees a decrease of the objective at each iteration. In [15], a comparative study was conducted on different types of ICA algorithms (FastICA, Jade and Infomax) for dimensionality reduction of hyperspectral images. From this study it appears that JADE formulation is more robust. FastICA algorithm is comparable to JADE from the perspective of accuracy/precision. However, when more features are considered, JADE demands a higher computational power than FastICA. Moreover, FastICA, JADE and Infomax are full batch methods and consequently they can perform poorly for large datasets (i.e., number of pixels is very large in the given hypercube), calling for more scalable algorithms. Additionally, these methods do not always find good optimal points (maxima). It is thus of importance to develop ICA solvers which are fast, easy to use and with strong convergence guarantees. Various strategies have been recently proposed to scale-up inferential problems from big datasets. Besides parallelized and distributed approaches exploiting hardware architectures, several variants of the stochastic gradient descent method have been designed for accelerating the optimization [3], [11].

We develop a stochastic second-order Taylor-based algorithm adapted to the loss functions used in ICA, inspired from [11] (see also the extended version [12] of this paper). In particular, we show that the ICA loss functions have the third derivatives bounded on the unit ball. Then, we derive an upper bound approximation of the objective function in ICA using a regularized second order Taylor-based approximation. At each iteration our algorithm minimizes this approximation. This algorithm, due to his stochastic nature, facilitates minibatching and thus is scalable and appropriate for large datasets. Moreover, our algorithm guarantees ascent of the loss function along iterations, hence it is able to identify local maxima. We prove experimentally that FastICA does not always find a local maximum point, hence it may provide a direction where the non-Gaussianity is minimized instead of maximizing it. On the other hand, our method being an ascent method always maximizes the non-Gaussianity. Experiments on a real hyperspectral dataset demonstrate the efficiency of our method.

II. Independent component analysis

In the signal processing field, the Independent Component Analysis (ICA) technique becomes popular due to its efficiency in solving the blind signal separation (BSS) problem. The goal of the BSS problem is to separate a set of source signals from a set of mixed observed signals, using little to none information of the sources signals. In what follows, we detail ICA technique, our presentation follows [7]. More precisely, let us consider a set of observed signals, denoted with $X = [x_1\, x_2 \ldots x_N] \in \mathbb{R}^{d\times N}$. We assume the set is a linear combination of separated independent signals $S = [s_1\, s_2 \ldots s_r] \in \mathbb{R}^{r\times N}$, i.e:

$$X = AS,$$

where $A \in \mathbb{R}^{d\times r}$ is called the mixing matrix and is unknown. The goal is to recover the source signals by estimating the unmixing matrix $W = A^+$ (pseudoinverse), i.e.:

$$\hat{S} = WX.$$

Note that this model is simple and we don't consider noise. Further, we formulate the basic assumptions for ICA:

1) Sources are considered statistically independent.
2) Independent components (ICs) are non-Gaussian.
3) Mixing matrix is (pseudo)invertible.

Let us observe that, for this technique, the independence is fundamental. To define it, we introduce the notion of probability density. Thus, let us denote the joint probability density function (PDF) of two signals as $p(s_1,s_2)$ and the marginal PDF of a signal as $p_i(s_i)$. Two signals, s_1 and s_2, are statistically independent if and only if the joint PDF can be expressed as: $p(s_1,s_2) = p_1(s_1)p_2(s_2)$. In other words, the signals s_1 and s_2 are not offering information about each other. Further, we need a way to quantify statistical independence and the central limit theorem is giving us the means. This theorem affirms that the distribution of a sum of independent signals with arbitrary distributions tends toward a Gaussian distribution. Thus, a Gaussian signal x can be considered as a linear combination of many independent signals s_i, $i = 1:r$. Hence, this two notion, independence and non-Gaussianity are equivalent. The standard technique of measuring the non-Gaussianity of a random variable s is the kurtosis, known as the fourth central moment [7]:

$$\kappa(s) = E[(s-\mu_s)^4/\sigma_s^4], \tag{1}$$

where μ_s is the mean of s and σ_s is the standard deviation (to make kurtosis dimensionless we need to normalize it, dividing by σ_s^4). Kurtosis measures the relative peakedness of flatness of a distribution with respect to a Gaussian (normal) distribution. Recall that the kurtosis of a Gaussian is equal to 3. Thus, a distribution with a positive kurtosis ($\kappa > 3$ in (1)) is named a super-Gaussian, while with a negative kurtosis ($\kappa < 3$ in (1)) is termed a sub-Gaussian. In order to have independence we need to maximize the kurtosis, since in practice the signals tend to be more super-Gaussian:

$$\max \kappa(s) = \max_{\|w\|=1} E[(w^T x - \mu_s)^4], \tag{2}$$

where we consider $s = w^T x$ and unit variance. Let us note that the problem is formulated for only one vector, i.e finding only one row of W, denoted w, at a time. Even though kurtosis is theoretically a good measure of non-Gaussianity, it is extremely sensitive to changes in the distribution tail. Thus, other measures of non-Gaussianity are often used to overcome this weakness. One of them is the approximation of negentropy, which estimates the fourth moment using a non-quadratic function g:

$$\max_{\|w\|=1} E\left[g(s)\right]. \tag{3}$$

Two common choices for g are:

$$g_1(s) = \frac{1}{\alpha}\ln\cosh(\alpha s) = \frac{1}{\alpha}\ln(\alpha\frac{e^s + e^{-s}}{2}), \text{ with } 1 \leq \alpha \leq 2$$

and

$$g_2(s) = -e^{-\frac{s^2}{2}}.$$

Usually g_1 is used in applications, but if robustness is very important or if the independent components are highly super-Gaussian, then we should choose g_2 [7].

Given a set of N i.i.d. samples $[x_1\ x_2 \cdots x_N]$, the empirical risk minimization problem for (3) reads:

$$\min_{\|w\|=1} \frac{1}{N}\sum_{i=1}^{N} -g(w^T x_i). \tag{4}$$

We observe that (4) is a particular case of a finite sum optimization problem, i.e. it can be written as:

$$\min_{\|w\|=1} f(w) := \frac{1}{N}\sum_{i=1}^{N} f_i(w). \tag{5}$$

Before continuing with algorithms that solve the ICA problem (3) or its approximation (4), let us highlight that the preprocessing steps are very important. The most basic and necessary preprocessing step is to center x, i.e. subtract its mean vector $E[x]$ so as to make x a zero-mean variable. This implies that s is zero-mean as well. After centering the data, whitening the observed signals x is the next step. This means that before the application of the ICA algorithm (and after centering), we transform the observed vector x linearly so that we obtain a new vector $\tilde{x}$ which is white, i.e. its components are uncorrelated and their variances equal unity. Thus, the covariance matrix of $\tilde{x}$ equals the identity matrix:

$$E[\tilde{x}\tilde{x}^T] = I_d.$$

The whitening transformation is always possible. One popular method for whitening is to use the eigenvalue decomposition of the covariance matrix $E[xx^T]$ (see Section IV.A).

III. Optimization algorithms for ICA

Below we describe two optimization algorithms for solving ICA. FastICA is one of the most used ICA algorithms, which is based on full batch Newton type iterations [7]. The second algorithm is a stochastic Newton type method with a proper cubic regularization term that guarantees descent, originally developed in the paper [11].

A. FastICA method

The FastICA algorithm for one unit (row) finds a direction w such that the projection $w^T x$ maximizes non-Gaussianity, i.e. problem (3) is solved. To find the update expression of FastICA, the Kuhn-Tucker optimality conditions are applied to the equality constrained problem (3), see [7]:

$$E[xg'\left(w^T x\right)] - \lambda w = 0 \quad \text{and} \quad \|w\| = 1, \tag{6}$$

where λ is the Lagrange multiplier for the constraint $\|w\| = 1$. We can solve this system of equations by Newton's method. Denoting the function on the left-hand side of (6) by F, $F(w) = E[xg'\left(w^T x\right)] - \lambda w$, we obtain its Jacobian matrix $\nabla F(w)$ as:

$$\nabla F(w) = E[xx^T g''\left(w^T x\right)] - \lambda I_d. \tag{7}$$

To simplify the inversion of this matrix, we will approximate the first term in (7). Since the data is on the unit ball, a reasonable approximation is:

$$\begin{aligned} &E[xx^T g''\left(w^T x\right)] \\ &\approx E[xx^T]\cdot E[g''\left(w^T x\right)] = E[g''\left(w^T x\right)]\cdot I_d. \end{aligned}$$

Thus the Jacobian matrix becomes diagonal and can be easily inverted. We obtain the following *approximate* Newton iteration (recall that we are using an approximation of the Jacobian):

$$w^+ = w - \left(E[xg'\left(w^T x\right)] - \lambda w\right) / \left(E[g''\left(w^T x\right)] - \lambda\right).$$

At optimality the optimal multiplier is given by $\lambda_* = E[w_*^T x g'\left(w_*^T x\right)]$, where w_* is an optimal point. Therefore, we also approximate the Lagrange multiplier at each iteration using the current estimate of w:

$$\lambda = E[w^T x g'\left(w^T x\right)].$$

This algorithm can be further simplified by multiplying both sides in the previous relation by $\lambda - E[g''\left(w^T x\right)]$. After few algebraic simplifications, we obtain the FastICA iteration (step 1 from Algorithm 1).

Algorithm 1: FastICA

Data: Choose a random w_0 and normalize it.
while $\delta \geq \varepsilon$ **do**
1. Update:
$$w_{k+1} = E[xg'\left(w_k^T x\right)] - E[g''\left(w_k^T x\right)]w_k$$
2. Normalize: $w_{k+1} \leftarrow w_{k+1}/\|w_{k+1}\|$
3. Update stopping criterion $\delta = |w_{k+1}^T w_k - 1|$
4. $w_k \leftarrow w_{k+1}$ and increase k.

end

To estimate several independent components, we need to run FastICA algorithm using several units (rows) with weight vectors $w_1, \ldots, w_r$. To prevent different vectors from converging to the same maxima we must decorrelate the outputs $w_1^T x, \ldots, w_r^T x$ after every iteration. There are different methods to decorrelate, one example being a deflation scheme

based on a Gram-Schmidt-like decorrelation. Despite its simplicity and fast convergence (as confirmed by many experiments), FastICA may fail to find local maxima due to the approximations used in the derivation of the Newton iteration for solving the KKT system and since it is well known that the convergence of the Newton method may be rather uncertain outside of the quadratic convergence ball. It is known that in order to guarantee global convergence for the Newton method, a proper cubic regularization is needed [14]. Hence, in the next section we describe a stochastic variant of the cubic regularized Newton method for solving the finite sum problem (4). The algorithm, in full generality, was developed in [11], [12].

B. Stochastic second-order Taylor-based method

We consider the empirical risk minimization problem (4) or equivalently the finite sum problem (5) coming from the ICA formulation. For simplicity of the exposition we choose:

$$f_i(w) = -g_1(w^T x_i) = \ln\cosh(w^T x_i),\ \forall i = 1:N.$$

We prove below that this loss function has the third derivatives bounded on the unit ball. Then, we derive an upper bound approximation of the objective function in ICA using a regularized second-order Taylor-based approximation. Our algorithm below, at each iteration minimizes this approximation. Indeed, the derivatives of f_i along a given direction v have the following expressions:

$$\begin{aligned}
&\nabla f_i(w)[v] = -\tanh(w^T x_i) x_i^T v\\
&\nabla^2 f_i(w)[v]^2 = -\operatorname{sech}^2(w^T x_i)(x_i^T v)^2,\\
&\nabla^3 f_i(w)[v]^3 = 2\operatorname{sech}^2(w^T x_i)\tanh(w^T x_i)(x_i^T v)^3.
\end{aligned}$$

Since the rows of unmixing matrix W satisfy $\|w\| \le 1$, it follows that f_i is concave function over the unit ball $\mathscr{B} = \{w : \|w\| \le 1\}$, provided that the data satisfy $\|x_i\| \le 1$. Moreover, f_i has the third derivative bounded on the unit ball $\mathscr{B}$. Indeed, since $\tanh(\cdot) \in [-1,1]$ and $\operatorname{sech}(\cdot) \in [0,1]$, we have:

$$\begin{aligned}
\|\nabla^3 f_i(w)\| &= \max_{\|v\|\le 1} |\nabla^3 f_i(w)[v]^3|\\
&= \max_{\|v\|\le 1} 2\operatorname{sech}^2(x_i^T w)|\tanh(x_i^T w)(x_i^T v)^3|\\
&\le \max_{\|v\|\le 1} 2|x_i^T v|^3 \le 2\|x_i\|^3,
\end{aligned}$$

where in the last inequality we used Cauchy-Schwartz inequality $|w^T x_i| \le \|w\|\|x_i\|$. From here we conclude that the hessian of f_i, $\nabla^2 f_i$, is Lipschitz continuous with the Lipschitz constant:

$$L_2^{f_i} = 2\|x_i\|^3.$$

In conclusion, the following inequality holds for each f_i:

$$\left\|\nabla^2 f_i(w) - \nabla^2 f_i(v)\right\| \le L_2^{f_i}\|w - v\| \quad \forall w, v \in \mathscr{B}. \tag{8}$$

If we further define the second-order Taylor approximation of the function f_i around a point v:

$$\begin{aligned}
&T_2^{f_i}(w;v)\\
&= f_i(v) + \nabla^T f_i(v)(w-v) + \frac{1}{2}(w-v)^T \nabla^2 f_i(v)(w-v),
\end{aligned}$$

then from (8) we can easily derive a *majorizer* of f_i [13]:

$$f_i(w) \le \phi_i(w;v) := T_2^{f_i}(w;v) + \frac{M_2^{f_i}}{6}\|w-v\|^3 \quad \forall w, v \in \mathscr{B},$$

valid for any $M_2^{f_i} \ge L_2^{f_i}$. Now, we are ready to derive a new algorithm for solving the negentropy-based ICA problem (4) (or equivalently the finite sum problem (5)). We propose a stochastic second-order minibatch Taylor-based (SSOM) algorithm based on the majorizers ϕ_i's. Our method belongs to the class of majorization-minimization algorithms, which consist of successively minimizing a sequence of upper bounds of the objective function so that along the iterations the objective function decreases [9], [13]. Our scheme is given in Algorithm 2. Note that the update of the global majorizer ϕ can be done very efficiently as:

$$\phi(w;\hat{w}_k) = \phi(w;\hat{w}_{k-1}) + \frac{\phi_{i_k}(w;w_k) - \phi_{i_k}(w;w_{k-1}^{i_k})}{N}.$$

We can also use the Hessian approximation used in FastICA when computing the expression of the global majorizer ϕ. Note that when $\tau = N$, the previous algorithm (SSOM) becomes a deterministic second order method with cubic regularization [14], called DSOM. To estimate several independent components, we need to run SSOM for the rows of W, $w_1, \ldots, w_r$, and use the same decorrelation procedure as in FastICA. For this stochastic method one can derive the following convergence and descent results (see [11], [12]).

Algorithm 2: Algorithm SSOM

Data: Given w_0, compute surrogate functions $\phi_j(w;w_0^j)$ of f_j near $w_0^j = w_0\ \forall j = 1:N$.

while $\delta \ge \varepsilon$ **do**

1. Chose uniformly random a subset (minibatch) $S_k \subseteq \{1,\cdots,N\}$ of size $\tau \in [1,N]$.
2. For each $i_k \in S_k$, compute the majorizer $\phi_{i_k}(w;w_k)$ of f_{i_k} near w_k and keep the previous majorizers for $j \notin S_k$
3. Update:

$$w_{k+1} \in \arg\min_{\|w\|=1} \phi(w;\hat{w}_k) := \frac{1}{N}\sum_{j=1}^{N} \phi_j(w;w_k^j),$$

where $\hat{w}_k = [w_k^j]_{j=1:N}$ is defined as

$$w_k^j = \begin{cases} w_k, & j \in S_k.\\ w_{k-1}^j, & j \notin S_k. \end{cases}$$

4. Update stoping criterion $\delta = |w_{k+1}^T w_k - 1|$
5. $w_k \leftarrow w_{k+1}$ and increase k.

end

Theorem 1: Assume that the functions f_i have the second derivatives Lipschitz over the unit ball $\mathscr{B}$. Then, the sequence $(w_k)_{k\ge 0}$ generated by SSOM is bounded, $\nabla f(w_k)$ converges to 0 (hence, any limit point of $(w_k)_{k\ge 0}$ is a stationary point) and the sequence of function values $(f(w_k))_{k\ge 0}$ monotonically decreases in expectation, i.e.:

$$E[f(w_{k+1})] \le E[f(w_k)] \quad \forall k \ge 0.$$

In conclusion, our algorithm SSOM is scalable (due to its stochastic nature) and has mathematical guarantees for the decrease (increase) of the finite sum function f (of the loss function g_1) along iterations. Hence, it has more chances to find local maxima than FastICA.

IV. ICA-BASED DIMENSIONALITY REDUCTION OF HYPERSPECTRAL IMAGES

A hyperspectral image consists of a three-dimensional hyperspectral data cube $m \times n \times d$, having $m \times n$ pixels, in which $N = m \cdot n$ is the number of pixels in each spectral channel and d represents the number of spectral channels (see Figure 1). From the spectral perspective, a hyperspectral data cube is composed of $m \times n$ pixels, where each pixel is a vector of d values. Each pixel corresponds to the reflected radiation of the specific region of the Earth and has multiple values in spectral bands. This detailed spectral information can be used to analyze different materials with precision. From the spatial perspective, a hyperspectral data cube consists of d gray scale images with a size of $m \times n$. The values of all pixels in one spectral band shape a grayscale image with two dimensions: wavelength and reflectance.

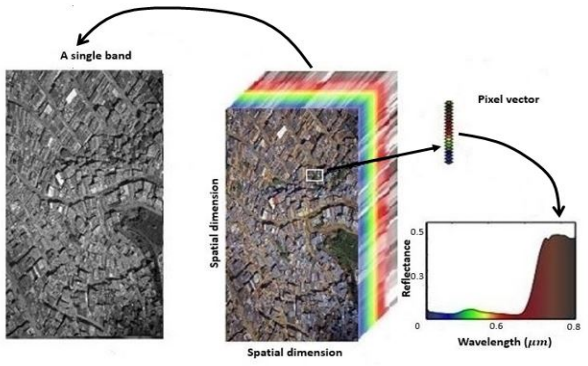

Fig. 1. A hyperspectral datacube: grayscale image (left), hyperspectral data cube (middle), pixel vector and its corresponding spectral signature (right).

However, relatively few bands can represent most of the information in hyperspectral images [8], making dimensionality reduction useful for storage, transmission, classification, target detection and visualization of remote sensing data [4], [6], [16]. As discussed in Section II, ICA is an unsupervised blind source separation technique, which identifies statistically independent components by considering only the observation of mixture signals. Based on this property, ICA can be applied to hyperspectral images which can be seen as a mixture of signals (bands), aiming at identifying and eliminating statistical redundancies of hyperspectral data while keeping as much spectral information as possible. We usually represent a hyperspectral image as a matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{d1} & x_{d2} & \cdots & x_{dN} \end{bmatrix} \in \mathbb{R}^{d \times N},$$

where $x_{i,j}$ is the pixel j on band i. We aim at reducing the number of bands in the hypespectral image from d to $r < d$ using the negentropy-based ICA framework. In the next section we present results obtained with the algorithms SSOM and FastICA for solving the negentropy-based ICA problem (4) (or equivalently the finite sum problem (5)).

A. Numerical results

For numerical simulations we used the hyperspectral image from the dataset `Indian Pines` [18]. This scene was gathered by AVIRIS sensor over the Indian Pines test site in North-Western Indiana. The spatial dimensions of the hyperspectral image are 145×145. It has 220 bands with 20 water absorption bands being discarded, hence $d = 200$. We choose $r = 15$. We denote the matrix representation of the image with $X^{200 \times 21025}$. Further, we start the preprocessing step by extracting the mean from each column of X. We denote the new matrix with $\tilde{X}$ and whiten it. This linear transformation can be achieved by an infinite number of whitening matrices Q. Since our goal is data reduction, we choose a PCA whitening procedure:

$$\tilde{X}\tilde{X}^T = U\Lambda U^T, \quad Q = U\Lambda^{-\frac{1}{2}}U^T, \quad \hat{X} = Q \cdot \tilde{X},$$

where the first relation is the eigendecomposition of the covariance matrix $\tilde{X}\tilde{X}^T$ and $\hat{X}$ is the whitened data. Further, for solving problem (4), we choose the loss function:

$$g_1(w^T\hat{x}_i) = \ln(\cosh(w^T\hat{x}_i)).$$

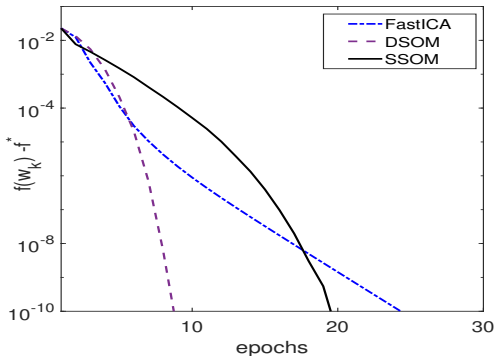

Fig. 2. Behavior of FastICA, SSOM and DSOM on Indian Pines dataset [18] for $\tau = 145$ and $w_0 = e_1$ along full iterations (epochs).

For FastICA, we use Matlab package from [19]. In Figure 2, we depict the behavior of FastICA, SSOM and DSOM. For high accuracies, our methods are better than FastICA, while for lower accuracies the three methods are comparable. Also note that SSOM is faster in CPU time than DSOM. In Figure 3, we plot the behavior of the three algorithms for two initial points. We observe that for certain initializations (e.g. $w_0 = \mathbf{1}/\|\mathbf{1}\|$) FastICA doesn't maximize the objective function (non-Gaussianity), while our methods being of ascent nature find a local maximum. The fact that FastICA can yield ICs that are not relevant can be seen in Figure 4 (left side). Finally, the three most relevant ICs are depicted in figure 5: clearly the reduced hyperspectral image obtained with SSOM is better than the one obtained with FastICA algorithm. The reader should note that similar behavior was observed for these algorithms on other initial points w_0 (including random choices) and on other loss functions. For the impact of dimension reduction on the image classification see [12].

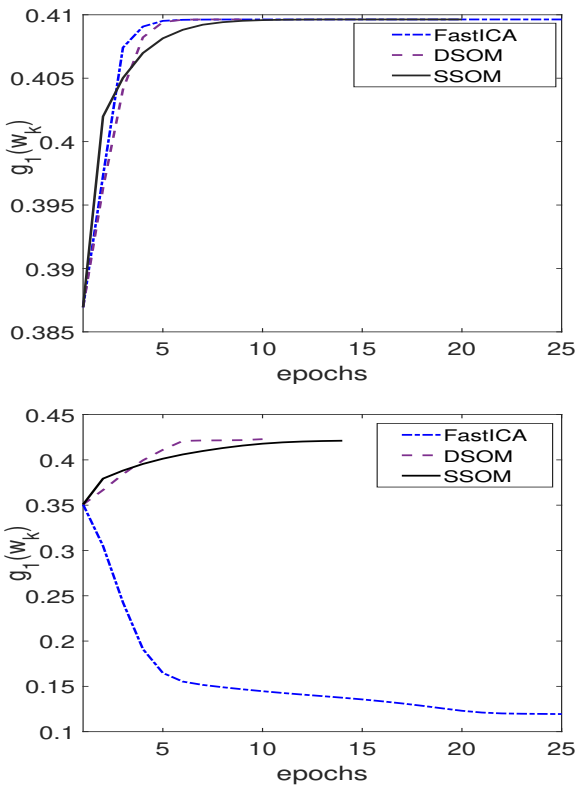

Fig. 3. Objective function g_1 along full iterations (epochs): comparison between FastICA, SSOM and DSOM on Indian Pines dataset [18] for different initializations: top $w_0 = e_1$, bottom $w_0 = \mathbf{1}/\|\mathbf{1}\|$. From the second plot we observe that FastICA minimizes g_1 instead of maximizing it.

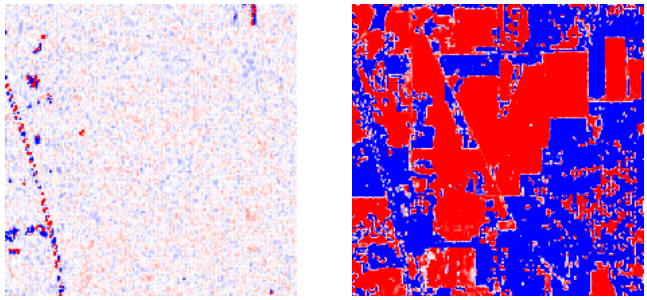

Fig. 4. The IC found with the initialization $w_0 = \mathbf{1}/\|\mathbf{1}\|$ on the Indian Pines dataset [18]: FastICA on the left, SSOM on the right.

V. Conclusions

In this paper we have present the ICA framework to reduce the dimension of hyperspectral images. FastICA is one of the most used ICA algorithms. It is based on a loss function derived from the fourth order statistical moment (kurtosis) or negentropy, which are non-convex functions. In this paper we have presented an alternative optimization algorithm to FastICA, which is a stochastic second-order majorization-minimization algorithm adapted to this loss function. Since our algorithm is stochastic in nature, it has the advantage of scalability and also it guarantees a decrease of the loss function along iterations.

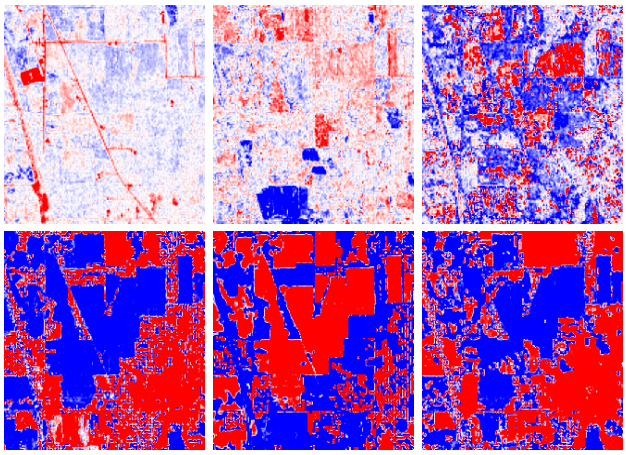

Fig. 5. Three most relevant ICs found with FastICA (top), SSOM (bottom).

References

[1] P. Ablin, A. Gramfort, J.F. Cardoso and F. Bach. *Stochastic algorithms with descent guarantees for ICA*, International Conference on Artificial Intelligence and Statistics, 1564–1573, 2019.

[2] S. Amari, A. Cichocki and H. Yang, *A new learning algorithm for blind signal separation*, Advances in Neural Information Processing Systems, 757–763, 1996.

[3] L. Bottou, *Stochastic Gradient Descent Tricks*, In Neural Networks: Tricks of the Trade, Springer, 2012.

[4] S. Bakken, M. Orlandic and T.A. Johansen, *The effect of dimensionality reduction on signature-based target detection for hyperspectral imaging*, SPIE Optical Engineering and Applications, 2019.

[5] J. Cardoso and A. Souloumiac, *Blind beamforming for non-gaussian signals*, IEEE Proceedings-F, 140: 362–370, 1993.

[6] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti and A. Plaza, *Advances in hyperspectral image and signal processing: a comprehensive overview of the state of the art*, IEEE Geoscience and Remote Sensing Magazine, 5(4): 37–78, 2017.

[7] A. Hyvärinen, J. Karhunen and E. Oja, *Independent component analysis*, Wiley, 2001.

[8] X. Jia, B. Kuo and M. Crawford, *Feature mining for hyperspectral image classification*, Proceedings of the IEEE, 101(3): 676–697, 2013.

[9] D.R. Hunter and K. Lange, *A tutorial on MM algorithms*, The American Statistician, 58: 30–37, 2004.

[10] G. Luo, G. Chen, L. Tian, K. Qin and S. Qian, *Minimum noise fraction versus principal component analysis as a preprocessing step for hyperspectral imagery denoising*, Canadian Journal of Remote Sensing 42(2): 106–116, 2016.

[11] D. Lupu and I. Necoara, *Convergence analysis of stochastic higher-order majorization-minimization algorithms*, arXiv preprint: 2103.07984, 2021.

[12] D. Lupu, I. Necoara, J. L. Garrette and T. A. Johansen, *Stochastic higher-order independent component analysis for hyperspectral dimensionality reduction*, IEEE Transactions on Computational Imaging, 8: 1184 - 1194, 2022.

[13] I. Necoara and D. Lupu, *General higher-order majorization-minimization algorithms for (non)convex optimization*, arXiv preprint: 2010.13893, 2020.

[14] Yu. Nesterov and B. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108: 177–205, 2006.

[15] F. Nicola, B. Bruzzone and J. Benediktsson, *An ICA based approach to hyperspectral image feature reduction*, IEEE Geoscience and Remote Sensing Symposium, 2014.

[16] M. Swarna, V. Sowmya and K. P. Soman, *Effect of dimensionality reduction on sparsity based hyperspectral unmixing*, Int. Conference on Soft Computing and Pattern Recognition, 429–439, 2016.

[17] H. Zou and L. Xue, *A selective overview of sparse principal component analysis*, Proceedings of the IEEE, 106(8): 1311–1320, 2018.

[18] `http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes`

[19] `https://research.ics.aalto.fi/ica/fastica/`