Stochastic higher-order independent component analysis for hyperspectral dimensionality reduction

Daniela Lupu[1], Ion Necoara[1,3], Joseph L. Garrett[2] and Tor Arne Johansen[2]

***Abstract*—Hyperspectral imaging is a remote sensing technique that measures the spectrum of each pixel in the image of a scene. It can be used to detect objects or classify materials based on their optical reflectance spectra. Various methods have been developed to reduce the spectral dimension of hyperspectral images in order to facilitate their analysis. Independent Component Analysis (ICA) is a class of algorithms which extract statistically independent features. FastICA, is one of the most used ICA algorithms because it is simple and fast. However, FastICA often finds irrelevant stationary points (e.g., minima instead of maxima) and is not scalable as it uses at each iteration the whole set of pixels. In this paper, we present a new stochastic algorithm, called SHOICA, which smoothly approximates the non-convex loss functions of ICA using higher-order Taylor minorizers. Because SHOICA guarantees ascent of its objective function, it identifies (local) maxima. Moreover, because SHOICA is stochastic, it facilitates minibatching and thus is scalable and appropriate for large datasets. The quality of features extracted, as well as the time and epochs required by both FastICA and SHOICA are compared on dimensionality reduction and classification tasks of real hyperspectral images.**

I. Introduction

Hyperspectral imaging acquires information from the electromagnetic spectrum by recording a continue of wavelengths rather than a few discrete bands. It is becoming a valuable tool for studying e.g. the Earth's surface, industrial product quality, and the human body, with applications ranging from environmental monitoring, healthcare evaluation, agriculture quality assurance to astronomy and chemical imaging [1], [2]. In the last two decades, many techniques have been proposed for hyperspectral image processing. The obstacles for hyperspectral image processing are quite different from those of color or greyscale images. Therefore, algorithms designed for the latter have limited success when they are applied to hyperspectral images. First, because hyperspectral images are often about 50 times larger than color images with the same number of pixels, they inevitably require more memory to store. The memory requirements to process them can exceed the needs of greyscale images by an even larger factor if coupling between the bands is considered. Second, because the number of samples is often quite small relative to the size of the feature space, the accuracy of classification can actually decrease as the number of bands increases, in what has been called the Hughes effect. Third, the large, tagged online databases which have been so critical for modern image processing with convolutional neural networks do not exist for hyperspectral images. Moreover, hyperspectral images are often taken of complex scenes in which the ground truth is sampled sparsely if at all.

Dimensionality reduction (DR) is a group of techniques which can mitigate the difficulties associated with hyperspectral images [2]–[4]. First, by selecting only the most information-rich data, DR reduces the size of the data to be processed. Second, the decreased dimensionality of the feature space helps to reduce the consequences of the Hughes effect. The third difficulty is only partially mitigated by DR. Many DR techniques present both a spatial and a spectral representation for each dimension. The physical origin of different signals can then be investigated by comparing each dimensions spectral representation to spectra from compiled databases.

State of the art. Dimensionality reduction algorithms can be partitioned into two groups depending on whether they are optimization based or machine learning based [5], [6]. Although both groups have an objective function, the optimization-based techniques are designed to be applied to an image as a whole, undivided into a training and test sets. Moreover, the machine learning

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

[2]Center for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, Emails: {joseph.garrett, tor.arne.johansen}@ntnu.no.

[3]Gheorghe Mihoc-Caius Iacob Institute of Mathematical Statistics and Applied Mathematics of the Romanian Academy, 050711 Bucharest, Romania.

based approaches result in a trained network, which can be immediately applied to additional images. Detailed comparisons between these techniques and their performances on classification tasks can be found e.g., in [5], [7]–[9]. In this paper, we follow the optimization based approach. The advantages of optimization based techniques such as Principal Component Analysis (PCA) [10], [11] and Independent Component Analysis (ICA) [12], [13] include more interpretable output, invertibility, and generality.

Among the optimization-based algorithms, PCA is one of the most common techniques to reduce the number of features. It consists of identifying the orthonormal basis which decorrelates the bands, ordered according to the variance of each [11], [14]. If the variables or spectral signal under observation carry additive independent normally distributed noise, PCA is an optimal method for noise filtering. However, variance, which is used to measure second-order statistics, may not effectively characterize the signal when the noise is not normally distributed. In particular, when the spectral data contains subtle signals that cannot be captured by the second-order statistics, PCA does not extract them [3], [10].

Independent Component Analysis (ICA) extracts subtle signals that are undetected by PCA because it incorporates higher-order moments, such as kurtosis [12]. It attempts to decompose a multivariate signal into independent signals, i.e. a decomposition that provides statistical independence between the estimated components. When the statistical independence assumption is reasonable, ICA separation of a mixed signal gives good results [3]. Due to this, ICA has good performance in reducing the effects of noise and other forms of undesired interference with the observed spectral signatures, enhancing the classification and detection rate, see e.g. [3], [4], [15]–[17]. The ICA problem is usually formulated as the maximization of a non-convex finite sum objective function subject to a simple quadratic constraint [12]. In the literature there are many optimization algorithms available for solving this finite sum optimization problem, i.e. finding the independent components. Commonly used one, including in industrial applications, is the FastICA algorithm, see [12]. It maximizes the kurtosis and is based on fixed point iterations derived from the KKT conditions of the ICA optimization problem.

Other methods for computing the ICs include Joint Approximation Diagonalization of Eigenmatrices (JADE) [18], Infomax [19] and stochastic majorization-minimization [20]. Infomax is based on a loss function which is a non-convex log-likelihood. In [20], a majorization-minimization optimization algorithm is developed, which is adapted to the Infomax loss function and guarantees a decrease of the objective at each iteration. The different ICA algorithms have varying sets of advantages and disadvantages. 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 the JADE formulation is more robust. The 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, calling for more scalable algorithms.

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 [21]–[23]. Additionally, FastICA, JADE and Infomax do not always find optimal points (i.e., maxima). It is thus of great importance to develop ICA solvers which are scalable, easy to use and with strong convergence guarantees. The computational complexity of ICA algorithms typically scales linearly in the number of data points, the number of signals to be extracted, and the number of iterations required to reach a given accuracy [24]. A scalable algorithm requires either faster convergence per iteration or minibatching, in which the dataset is subsampled at each iteration.

Contributions. In this paper we develop a new stochastic higher-order Taylor-approximation based algorithmic framework adapted to the loss functions used in ICA in order to improve scalability and guarantee objective function ascent. Our development starts from the observation that the loss functions in ICA have the second- (third-) order derivatives bounded over the feasible set and thus they can be minorized by a first- (second-) order Taylor approximation with a proper regularization term. For instance, the simplest variant of our method in each iteration needs to compute only the first/second-order derivatives of a single randomly selected function from the finite sum. Hence, our stochastic framework is based on the notion of stochastic first-/second order Taylor lower bound approximations of the ICA finite-sum objective function and minibatching, called *SHOICA*.

Our algorithm builds on the foundation of FastICA, but provides two additional benefits which assist the processing of hyperspectral data. First, due to its stochastic nature, SHOICA is *scalable* and appropriate for large

datasets. Unlike FastICA, SHOICA works with any batch size. Hence, our method is faster in terms of cpu time than FastICA for large images. Second, theoretical convergence guarantees are provided for SHOICA. This is in contrast to FastICA, for which the updating rule is a simplified Newton type iteration (see [12] for its derivation), for which there are no global converges guarantees (see e.g., [25]). In particular, it is proved that SHOICA algorithm guarantees ascent for the objective function along iterations. FastICA, because it solves the KKT optimality conditions, does not, and sometimes may determine (local) minimal points instead of finding maximal points. The solutions found by SHOICA are guaranteed to be local maxima of the objective function. Hence, in practice the extracted features yielded by SHOICA are better than by FastICA (see also our numerical section). Experiments on hyperspectral datasets demonstrate the efficiency and performance of our method both for dimensionality reduction itself and as pre-processing for classification tasks. More specifically, we demonstrate the superiority of our method against other state-of-the-art dimensionality reduction algorithms on several benchmark hyperspectral data sets.

Content. The paper is organized as follows. In Section II we formulate the ICA problem and present one of the most known algorithms for solving it, FastICA. We continue in Section III by presenting our approach, SHOICA, and provide convergence guarantees and implementation details. In Section IV, we present detailed experiments in order to emphasize the superior performance of SHOICA and test the quality of the reduced data in the context of classification.

Notations and preliminaries. We consider the Euclidean space $\mathbb{R}^b$ and denote the unit sphere with $\mathcal{B} = \{x \in \mathbb{R}^b : \|x\| = 1\}$. For a p-multilinear form H in $\mathbb{R}^b$, where p is a positive integer, its value in $x_1, \ldots, x_N \in \mathbb{R}^b$ is denoted with $H[x_1, \ldots, x_N]$. The abbreviation $H[x]^p$ is used when $x_1 = \cdots = x_p = x$ for some $x \in \mathbb{R}^b$. The norm of a symmetric p-multilinear form H is defined in the standard way [26]:

$$\|H\| := \max_{\|x\|=1} |H[x]^p|.$$

We use the short notation $i = 1 : p$ for $i \in \{1, \cdots, p\}$. For a p times continuously differentiable function $G : \mathbb{R}^b \to \mathbb{R}$, its derivatives of order $i = 1 : p$ at some $w \in \mathbb{R}^b$ is denoted with $\nabla^i G(w)$. The i-directional derivative of a function G at w along the direction $x \in \mathbb{R}^b$ is denoted by:

$$\nabla^i G(w)[x]^p \quad \text{for } i \geq 1.$$

Based on this, we can write the Taylor approximation of the function G around v of order p as follows:

$$T_p^G(w; v) = G(v) + \sum_{j=1}^{p} \frac{1}{j!} \nabla^j G(v)[w-v]^j \quad \forall w, v \in \mathbb{R}^b.$$

Further, we introduce the p Lipschitz derivative notion.

Definition 1. *Let $G : \mathbb{R}^b \to \mathbb{R}$ be p times continuously differentiable. Then, the p derivative is Lipschitz continuous on a set $\mathcal{B}$ if there exist $L_p^G > 0$ such that the following relation holds:*

$$\|\nabla^p G(w) - \nabla^p G(w)\| \leq L_p^G \|w - v\| \quad \forall w, v \in \mathcal{B}. \quad (1)$$

Note that if the $p+1$ derivative of G is bounded, i.e

$$\|\nabla^{p+1} G(w)\| \leq L_p^G \quad \forall w \in \mathcal{B},$$

then (1) holds, see [23]. It is well-known that if (1) holds, then the following bounds on G are valid [23], [26]:

$$|G(w) - T_p^G(w; v)| \leq \frac{L_p^G}{(p+1)!} \|w - v\|^{p+1} \quad (2)$$

$$\forall w, v \in \mathcal{B}. \quad (3)$$

For a random variable x, $\mathbb{E}[x]$ denotes its expectation.

II. Independent Component Analysis

Independent component analysis (ICA) is a procedure that solves the Blind Source Separation (BSS) problem by recovering statistically independent signals from a linear mixture, see [12] for more details. ICA is motivated by a model that consists of a set of observations $X = [x_1\ x_2 \ldots x_N] \in \mathbb{R}^{b \times N}$, which is a linear combination of separated independent signals $U = [u_1\ u_2 \ldots u_r] \in \mathbb{R}^{r \times N}$, i.e.:

$$X = VU,$$

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

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

The fundamental issue in formulating ICA is determining how to describe statistical independence. Two signal u_1 and u_2 are statistically independent if and only if the joint PDF can be expressed as:

$$p(u_1, u_2) = p_1(u_1) p_2(u_2),$$

where the joint probability density function (PDF) of the signals is $p(u_1, u_2)$ and the marginal PDF of a signal is $p_i(u_i)$, for $i = 1, 2$. However, this condition is difficult to incorporate directly into an objective function

because it involves computing the relationships between signals. Because the central limit theorem states that the distribution of a sum of independent signals with arbitrary distributions tends toward a Gaussian distribution, a signal which is non-Gaussian should not be the sum of many other signals. In this sense, a non-Gaussian signal can be considered independent. A common metric for the non-Gaussianity of a random signal u is kurtosis [12], known as the fourth central moment:

$$\kappa(u) = \mathbb{E}[(u - \mu_u)^4 / \sigma_u^4], \tag{4}$$

where μ_u is the mean of u and σ_u is the standard deviation (to make kurtosis dimensionless we need to normalize it, dividing by σ_u^4). Distributions with kurtosis less than three are called sub-gaussian while those above three are called super-gaussian. Hence, we maximize:

$$\max \kappa(u) = \max_{\|w\|=1} \mathbb{E}[(w^T x - \mu_u)^4], \tag{5}$$

where we consider $u = w^T x$ and unit variance. Note that the problem is formulated for only one vector, i.e., finding only one row of W, denoted w, at a time. Because kurtosis is disproportionately sensitive to outliers, other metrics of non-Gaussianity have been developed. One of them, negentropy, estimates the fourth moment in a way that is more insensitive to outliers.

From the information theory concept, entropy measures the randomness of a signal. Because a Gaussian variable has the largest entropy among all random variables of equal variance, entropy can be used as a measure of non-Gaussianity. In contrast, the entropy is small for distributions that are concentrated on certain values. One measure of non-Gaussianity that is zero for a Gaussian variable and non-negative otherwise is called negentropy, $\mathcal{J}$, and is defined as follows:

$$\mathcal{J}(u) = H(u_G) - H(u),$$

where u_G is a given random Gaussian signal with the same covariance matrix as u and H is the entropy (see Section 5.1.1 in [12].) Although negentropy is robust, it requires an estimation of the PDF and thus some approximations are used in practice. One approximation of negentropy is:

$$\mathcal{J}(u) \approx \left(\mathbb{E}[g(u)] - \mathbb{E}[g(\vartheta)]\right)^2,$$

where ϑ is a zero mean unit variance Gaussian variable and the function g is some non-quadratic function which leads to the approximation always being non-negative (or zero if u has a Gaussian distribution). For the estimator to be robust, it is necessary to choose a function that grows less quickly than kurtosis with increasing u. Some typical choices for g are:

$$g(u) = \frac{1}{\alpha} \ln \cosh(\alpha u), \text{ with } 1 \leq \alpha \leq 2,$$
$$g(u) = -e^{-\frac{u^2}{2}} \quad \text{or} \quad g(u) = u^3. \tag{6}$$

Usually the first example is used in applications, but if robustness is very important or if the independent components are highly super-Gaussian, then we should choose the second example for g, see [12].

Before formulating the optimization problem, some preprocessing step of the data is necessary. The first step consist in centring the data, i.e we subtract from x it's mean, $\mathbb{E}[x]$. Then, the observed signals, X, will have zero mean and implicitly U as well. The next step requires to whiten the data, i.e., we apply a linear transformation Q on X such that $\mathbb{E}[XX^T] = I_b$. Thus, we decorrelate the observed signals and make their variance to be unity. Since the ICA framework is insensitive to the variances of the independent components, we can assume without loss of generality that the sources are white as well, i.e., $\mathbb{E}[UU^T] = I_r$. There are many options for whitening, but for dimensionality reduction, whitening by way of PCA is simple and general. Data can be whitened by PCA by computing $XX^T = C\Lambda C^T$ with either of these two alternatives (see Section 6.4 in [12]):

$$Q = \begin{cases} Q_1 = C\Lambda^{-\frac{1}{2}} C^T \\ Q_2 = \Lambda^{-\frac{1}{2}} C^T, \end{cases} \tag{7}$$

and then $X_w = QX$, where we denoted with X_w the whitened data. In what follows, we consider the data preprocessed and still denote it with X. After preprocessing, one searches for a maximal point of the optimization problem:

$$\arg\max_{\|w\|=1} \mathcal{J}(w^T x) = \arg\max_{\|w\|=1} |\mathbb{E}[g(w^T x)] - \mathbb{E}[g(\vartheta)]|.$$

Taking into account the fact that most independent components encountered in practice are super-Gaussian (see Section 3.2.1 [27]), the following ICA optimization problem is finally solved (see also (5)):

$$\max_{\|w\|=1} \mathbb{E}[g(w^T x)]. \tag{8}$$

One popular choice for solving this problem is FastICA given in Algorithm 1 below, which is based on Newton type iterations for the KKT conditions of the ICA optimization problem (8), see [12], [27], [28]. Note that in Algorithm 1, the expectation in step 1 can be approximated with a finite sum using the empirical data, see also (9). The procedure described in FastICA finds only one unit, i.e finds one row w such

that the projection $w^T x$ maximizes non-Gaussianity. To estimate several independent components, we need to find the maxima of optimization problem (8) 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 two primary methods to decorrelate, the symmetric scheme in which a Gram-Schmidt-like decorrelation is applied to all the sources simultaneously [12] and the deflation scheme based on sequentially estimating the sources one by one [24]. After $W = [w_1; \ldots; w_r] \in \mathbb{R}^{r \times b}$ is found, the reduced data are obtained as:

$$X_{\text{reduced}} = WX \in \mathbb{R}^{r \times N}, \quad \text{with} \quad r \ll b.$$

Algorithm 1 FastICA for one unit

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

Note that FastICA (Algorithm 1) is a full batch method and consequently it can perform poorly for large datasets. Additionally, 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 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 a Newton type method, a proper cubic regularization is needed [25].

III. A STOCHASTIC HIGHER-ORDER TAYLOR-BASED ICA ALGORITHM

In this section we propose a stochastic higher-order Taylor-based ICA algorithmic framework, called SHOICA, that removes some of the drawbacks described in the previous section. Our method makes use of the structure present in the ICA formulation. First, given a set of N i.i.d. samples $[x_1\ x_2 \cdots x_N]$ of the random variable x one can approximate the stochastic optimization problem (8) through the empirical risk formulation (also called finite sum problem, see [23]):

$$\max_{\|w\|=1} G(w) := \frac{1}{N} \sum_{i=1}^{N} g_i(w), \tag{9}$$

where $g_i(w) = g(x_i^T w)$. Further, we assume that each individual function g_i is p times differentiable and has the p derivative Lipschitz continuous with constant L_p on the unit sphere $\mathcal{B}$, for $p = 1, 2$. Note that all contraction functions in (6) comply with our assumptions for $p = 1, 2$. In particular, the first three directional derivatives along a direction ν are computed as:

$$\begin{aligned} &\nabla g(x_j^T w) = g_j'(x_j^T w) \cdot x_j \\ &\nabla^2 g(x_j^T w)[\nu] = g_j''(x_j^T w) x_j^T \nu \cdot x_j \\ &\nabla^3 g(x_j^T w)[\nu]^2 = g_j'''(x_j^T w)(x_j^T \nu)^2 \cdot x_j. \end{aligned} \tag{10}$$

For such functions $g(x_i^T w)$, we only need to keep track of scalar values, avoiding matrices or tensors storage when computing higher-order derivatives. Further, for $p = 1, 2$ the Lipschitz constants for the contraction function examples from (6) are:

(I) For the function $g_i(w) = \log\cosh(x_i^T w)$, where for simplicity we consider $\alpha = 1$, the expressions of the first three derivatives along a given direction ν are:

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

Since $\tanh(\cdot) \in [-1, 1]$ and $\operatorname{sech}(\cdot) \in [0, 1]$, we can bound the third derivative as follows:

$$\begin{aligned} \|\nabla^3 g_i(w)\| &= \max_{\|\nu\| \leq 1} |\nabla^3 g_i(w)[\nu]^3| \\ &= \max_{\|\nu\| \leq 1} 2 \operatorname{sech}^2(x_i^T w) |\tanh(x_i^T w)(x_i^T \nu)^3| \\ &\leq \max_{\|\nu\| \leq 1} 2 |x_i^T \nu|^3 \leq 2 \|x_i\|^3 \quad \forall \|w\| \leq 1, \end{aligned}$$

where in the last inequality we used Cauchy-Schwartz inequality $\left|x_i^T \nu\right| \leq \|x_i\| \, \|\nu\|$. Thus, the Hessian $\nabla^2 g_i$ is Lipschitz continuous with Lipschitz constant:

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

Using the same reasoning as for $p = 2$, we can easily show that the gradient ∇g_i is Lipschitz continuous with the Lipschitz constant:

$$L_1^{g_i} = \|x_i\|^2.$$

(II) For the second example $g_i(w) = -e^{-\frac{(x_i^T w)^2}{2}}$, the expressions of the first three derivatives along a given direction ν are:

$$\begin{aligned} &\nabla g_i(w)[\nu] = \epsilon \left(w^T x_i\right) \left(x_i^T \nu\right) \\ &\nabla^2 g_i(w)[\nu]^2 = \epsilon \left[1 - \left(w^T x_i\right)^2\right] \left(x_i^T \nu\right)^2 \\ &\nabla^3 g_i(w)[\nu]^3 = \epsilon \left[\left(w^T x_i\right)^3 - 3 \left(w^T x_i\right)\right] \left(x_i^T \nu\right)^3, \end{aligned}$$

where we denote $\epsilon = e^{-\frac{\left(w^T x_i\right)^2}{2}}$. Following a similar reasoning as in the first example, we find the third derivative bounded:

$$\|\nabla^3 g_i(w)\| \leq \left(3\|x_i\| + \|x_i\|^3\right)\|x_i\|^3 \quad \forall \|w\| \leq 1,$$

and implicitly an estimate of the Lipschitz constant of the Hessian: $L_2^{g_i} = \left(3\|x_i\| + \|x_i\|^3\right)\|x_i\|^3$. We can also show that the gradient is Lipschitz continuous with the constant: $L_1^{g_i} = (1 + \|x_i\|^2)\|x_i\|^2$.

(III) Finally, for the function $g_i(w) = (x_i^T w)^3$, proceeding in the same manner as above, we find that:

$$\begin{aligned}\|\nabla^3 g_i(w)\| &\leq \max_{\|\nu\|\leq 1} |\nabla^3 g_i(w)[\nu]^3| \\ &\leq \max_{\|\nu\|\leq 1} |6(x_i^T \nu)^3| \leq 6\|x_i\|^3 \ \ \forall \|w\| \leq 1.\end{aligned}$$

Thus, the Lipschitz constants for the Hessian and for the gradient are in this case:

$$L_2^{g_i} = L_1^{g_i} = 6\|x_i\|^3,$$

respectively. Finally, the overall Lipschitz constant of the p derivatives of function G in (9), for $p = 1, 2$, is:

$$L_p = \frac{1}{N}\sum_{i=1}^{N} L_p^{g_i}.$$

Our algorithmic approach consists of finding a lower bound for each function g_i which is simpler to maximize than the original one. For functions g_i that have the p derivative Lipschitz continuous we can use the Taylor approximation plus a proper regularization to bound from below the objective function (see (2)):

$$\begin{aligned} g_i(w) &\geq \phi_i(w; v) \qquad (11)\\ &:= T_p^{g_i}(w; v) - \frac{M_p}{(p+1)!}\|w - v\|^{p+1}, \quad p = 1, 2,\end{aligned}$$

where the constant $M_p \geq L_p$. Summing for $i = 1 : N$, we obtain a lower bound on G:

$$G(w) \geq \phi(w; v) := \frac{1}{N}\sum_{i=1}^{N} \phi_i(w, v). \qquad (12)$$

Now, we are ready to derive a new Stochastic Higher Order ICA-based optimization algorithm for (9), which we call SHOICA, that is based on the Taylor approximations of the individual functions g_i, see also [23].

Note that for the contraction functions from (6) it is easy to compute the higher-order derivatives (see the discussion above) and consequently we can update the model of the subproblem $\phi(w; \hat{w}_k)$ in an efficient manner as:

$$\begin{aligned}\phi(w; \hat{w}_k) &= \frac{1}{N}\sum_{i=1}^{N} \phi_i(w; w_k^i) \qquad (13)\\ &= \phi(w; \hat{w}_{k-1}) + \frac{\sum_{i\in S_k} \phi_i(w; w_k) - \phi_i(w; w_{k-1}^i)}{N}.\end{aligned}$$

Algorithm 2 SHOICA for one unit

Given w_0, compute functions $\phi_i(w; w_0^i)$ of g_i near $w_0^i = w_0 \quad \forall i = 1 : N$.

while $\delta \geq \epsilon$:

1. Chose uniformly random a subset (minibatch) $S_k \subseteq \{1, \cdots, N\}$ of size $\tau \in [1, N]$.
2. For each $i \in S_k$, compute $\phi_i(w; w_k)$ of g_i near w_k as in (11) and keep the previous minorizers for $i \notin S_k$
3. Find:

$$w_{k+1} \in \arg\max_{\|w\|=1} \phi(w; \hat{w}_k) := \frac{1}{N}\sum_{i=1}^{N} \phi_i(w; w_k^i)$$

such that the following increase holds

$$\phi(w_{k+1}; \hat{w}_k) \geq \phi(w_k; \hat{w}_k), \qquad (14)$$

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

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

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

Moreover, regardless of the convexity properties of g_i, in our algorithm it is sufficient to compute only a w_{k+1} that satisfies the ascend property $\phi(w_{k+1}; \hat{w}_k) \geq \phi(w_k; \hat{w}_k)$, see (14). Hence, we do not need to solve the subproblem $\max_{\|w\|=1} \phi(w; \hat{w}_k)$ exactly. Consequently, our algorithm is simple to implement. Moreover, one can see that for $\tau = N$, the algorithm becomes a deterministic one. However, our algorithm is flexible as it allows to work with minibatches of any size $\tau \in [1 : N]$. Finally, our algorithm is an ascent method in expectation, i.e., $\mathbb{E}[G(w_{k+1})] \geq \mathbb{E}[G(w_k)]$. Hence, compared to FastICA our method generally yields (local) maxima. This property is proved in Theorem 1 below and it is also supported by the numerical experiments from Section IV. It is important to note that we can also use a line search procedure in SHOICA in step 3. To preserve the ascent property of our algorithm it is sufficient to choose for the stepsize M_p an adaptive one, M_p^k, that ensures at each iteration k the increase condition:

$G(w_{k+1}) \geq \phi(w_k; \hat{w}_k)$. More precisely, at each iteration k we can apply the following *line search procedure* to find w_{k+1} in step 3:

i. Let $\beta > 1$, $M_p^{k,0} < M_p$ and $l = 0$.
ii. Until $G(w_{k+1}) \geq \phi(w_k; \hat{w}_k)$ holds, do:
Set $M_p^{k,l+1} \leftarrow \beta M_p^{k,l}$ and find corresponding w_{k+1}^{l+1}.
Increment $l \leftarrow l + 1$.
iii. Set $w_{k+1} = w_{k+1}^l$

This procedure finishes in a finite number of steps (12).

A. Solving SHOICA's subproblem

Our algorithmic framework depends on the choice of $p = 1, 2$ and requires computing the solution of a subproblem at each iteration, see step 3 of Algorithm 2 (SHOICA). In this section, we provide implementation details for this step. Let us consider first the deterministic case, i.e. when $\tau = N$. Then, for $p = 1$, the subproblem becomes:

$$\begin{aligned} w_{k+1} &\in \arg\max_{\|w\|=1} \phi(w; \hat{w}_k), \\ &= \arg\max_{\|w\|=1} \langle \nabla G(w_k), w - w_k \rangle - \frac{M_1}{2}\|w - w_k\|^2 \\ &= \arg\min_{\|w\|=1} \left\| w - \left(w_k + M_1^{-1}\nabla G(w_k)\right)\right\|^2, \end{aligned}$$

where the last expression is just a term rearrangement and $\nabla G(w_k) = \frac{1}{N}\sum_{i=1}^N \nabla g_i(w_k)$. Hence, the solution is just a gradient step projected on the set $\mathcal{B}$, which has an explicit expression:

$$\tilde{w}_{k+1} = w_k + M_1^{-1}\nabla G(w_k), \quad w_{k+1} = \tilde{w}_{k+1}/\|\tilde{w}_{k+1}\|.$$

Let us compare our previous iteration (i.e., the update in SHOICA with $p = 1$ and $\tau = N$) with the FastICA iteration. Approximating the expectation with the empirical risk, we get that the update in FastICA has the following expression after rearranging the terms:

$$\tilde{w}_{k+1} = w_k + \left(-\frac{1}{N}\sum_{i=1}^N g''\left(w_k^T x_i\right)\right)^{-1} \nabla G(w_k).$$

The only difference between SHOICA (with $p = 1$ and $\tau = N$) and FastICA iterations consists in the choice of the stepsizes, i.e. M_1^{-1} versus $\left(-\frac{1}{N}\sum_{i=1}^N g''\left(w_k^T x_i\right)\right)^{-1}$, respectively. Note that we can also use a line search procedure at iteration k of SHOICA (see previous section), i.e., we can replace M_1 with some adaptive step M_1^k that ensures the increase condition $G(w_{k+1}) \geq \phi(w_k; \hat{w}_k)$. This condition is sufficient for SHOICA to converge and identify local maxima, while the stepsize choice in FastICA does not guarantee any convergence for its iterations (see also Section IV).

For $p = 2$, the objective function in the subproblem of step 3 of SHOICA has the expression:

$$\begin{aligned} \phi(w; w_k) &= \langle \nabla G(w_k), w - w_k \rangle \\ &+ \frac{1}{2}\langle \nabla^2 G(w_k)(w - w_k), w - w_k \rangle - \frac{M_2}{6}\|w - w_k\|^3, \end{aligned}$$

where $\nabla G(w_k)$ is defined as before and $\nabla^2 G(w_k) = \frac{1}{N}\sum_i^N \nabla^2 g_i(w_k)$. For finding w_{k+1} in this case we can use (augmented) Lagrangian methods. For example, we can search for a scalar multiplier λ, corresponding to the constraint $w^T w - 1 = 0$, that minimizes the convex dual function $d(\lambda)$ obtained from the Lagrangian:

$$\begin{aligned} d(\lambda) &= \max_{w \in \mathbb{R}^b} \mathcal{L}(w, \lambda) \\ &:= -\frac{\lambda}{2} + \langle \nabla G(w_k) - \nabla^2 G(w_k) w_k, w \rangle \\ &+ \frac{1}{2}\langle (\nabla^2 G(w_k) + \lambda I_b) w, w \rangle - \frac{M_2}{6}\|w - w_k\|^3, \end{aligned}$$

using e.g., the bisection method for minimization in λ, $\min_{\lambda \in \mathbb{R}} d(\lambda)$, and the numerical scheme in [25] for maximization in w, $\max_{w \in \mathbb{R}^b} \mathcal{L}(w, \lambda)$. An alternative algorithm for solving the subproblem in this case is given in the Appendix and it is based on the alternating direction method of multipliers (ADMM) [29].

SHOICA becomes scalable in the stochastic case, i.e. $\tau < N$. In this setting, for $p = 1$ the subproblem in step 3 of SHOICA has the expression:

$$\begin{aligned} w_{k+1} &\in \arg\max_{\|w\|=1} \frac{1}{N}\sum_{i=1}^N \phi_i(w; w_k^i) \\ &= \arg\max_{\|w\|=1} \frac{1}{N}\sum_{i=1}^N \langle \nabla g_i(w_k^i), w - w_k^i \rangle - \frac{M_1}{2}\|w - w_k^i\|^2 \\ &= \arg\min_{\|w\|=1} \sum_{i=1}^N \left\| w - \left(w_k^i + M_1^{-1}\nabla g_i(w_k^i)\right)\right\|^2 \\ &= \arg\min_{\|w\|=1} \left\| w - \left(\bar{w}_k + M_1^{-1}\bar{\nabla G}(\hat{w}_k)\right)\right\|^2, \end{aligned}$$

where we denoted $\bar{w}_k = \frac{1}{N}\sum_{i=1}^N w_k^i$ and $\bar{\nabla G}(\hat{w}_k) = \frac{1}{N}\sum_{i=1}^N \nabla g_i(w_k^i)$. Hence, the solution of this subproblem is a stochastic gradient step projected on the set $\mathcal{B}$, which has the explicit expression:

$$\tilde{w}_{k+1} = \bar{w}_k + M_1^{-1}\bar{\nabla G}(\hat{w}_k), \quad w_{k+1} = \tilde{w}_{k+1}/\|\tilde{w}_{k+1}\|.$$

Hence, in the stochastic settings, there are differences between SHOICA and FastICA updates in both, stepsizes (M_1^{-1} versus $\left(-\frac{1}{N}\sum_{i=1}^N g''\left(w_k^T x_i\right)\right)^{-1}$) and directions

($\bar{\nabla G}(\hat{w}_k)$ versus $\nabla G(w_k)$). Note that in SHOICA the average sequence $\bar{w}_k$ and the average gradient $\bar{\nabla G}(\hat{w}_k)$ are updated in an efficient manner:

$$\bar{w}_k = \bar{w}_{k-1} + \frac{\sum_{i\in S_k}(w_k - w^i_{k-1})}{N}$$
$$\bar{\nabla G}(\hat{w}_k) = \bar{\nabla G}(\hat{w}_{k-1}) + \frac{\sum_{i\in S_k}(\nabla g_i(w_k) - \nabla g_i(w^i_{k-1}))}{N}.$$

Recall that we can easily compute the gradients of the g_i functions since we only need to keep track of scalar values (see (10)). It follows that the stochastic variant of SHOICA (with $p = 1$ and $\tau < N$) is indeed scalable (since computing $\bar{\nabla G}(\hat{w}_k)$ in SHOICA is much easier than computing $\nabla G(w_k) = \frac{1}{N}\sum_{i=1}^N \nabla g_i(w_k)$ in FastICA, when N is large).

Finally, for $p = 2$ and $\tau < N$ the objective function of the subproblem becomes:

$$\phi(w; \hat{w}_k) = \frac{1}{N}\sum_{i=1}^N \big(\langle \nabla g_i(w^i_k), w - w^i_k\rangle \tag{15}$$
$$+ \frac{1}{2}\langle \nabla^2 g_i(w^i_k)(w - w^i_k), w - w^i_k\rangle - \frac{M_2}{6}\|w - w^i_k\|^3\big).$$

First, recall that we can easily compute the higher-order directional derivatives along a direction (see (10)) and we only need to keep track of scalar values. Further, for solving the subproblem (15) one can use stochastic methods that are based on proximal operators [26] or variant-reduced types methods [30], since the objective function of the subproblem, $\phi(w; \hat{w}_k)$, has Lipschitz gradient over the compact set $\mathcal{B}$.

B. Convergence properties of SHOICA

In this section we provide convergence guarantees for SHOICA under quite general assumptions on the objective function G of the finite sum problem (9). In particular, our algorithm is an ascent method in expectation and consequently it finds (local) maxima.

Theorem 1. *Assume that the individual objective functions g_i (possibly nonconvex), with $i = 1 : N$, of problem* (9) *have the second and third derivatives bounded over $\mathcal{B} = \{w \in \mathbb{R}^b : \|w\| = 1\}$. Additionally, assume that G is bounded from above by $G_* < \infty$. Then, the sequence $(w_k)_{k>0}$ generated by SHOICA satisfies:*

$$G(w_{k+1}) \geq \phi(w_{k+1}; \hat{w}_k) \geq \phi(w_k; \hat{w}_{k-1}) \geq G(w_0),$$
$$G_* \geq \mathbb{E}[G(w_{k+1})] \geq \mathbb{E}[G(w_k)],$$

and consequently the sequence $(\mathbb{E}[G(w_k)])_{k>0}$ converges. Moreover, any limit point of $(w_k)_{k>0}$ is a stationary point of (9) *and the sequence $(G(w_{k+1}) - \phi(w_{k+1}; \hat{w}_k))_{k\geq 0}$ converges to 0 almost surely.*

Proof. First, let us note that the approximation model ϕ is bounded from above by the objective function G, when both are evaluated in w_{k+1}:

$$\phi(w_{k+1}; \hat{w}_k) = \frac{1}{N}\sum_{i=1}^N \phi_i(w_{k+1}; w^i_k)$$
$$\overset{(11)}{\leq} \frac{1}{N}\sum_{i=1}^N g_i(w_{k+1}) = G(w_{k+1}). \tag{16}$$

Further, note that for $i \in S_k$, we have:

$$\phi_i(w_k; \hat{w}_k) = \phi_i(w_k; w_k) = g_i(w_k) \geq \phi_i(w_k; \hat{w}_{k-1}).$$

Similarly, for $i \notin S_k$, we get the following relation:

$$\phi_i(w_k; \hat{w}_k) = \phi_i(w_k; \hat{w}_{k-1}).$$

Therefore, summing all these inequalities, we obtain:

$$\phi(w_k; \hat{w}_{k-1}) \leq \phi(w_k; \hat{w}_k) \leq \phi(w_{k+1}; \hat{w}_k), \tag{17}$$

where the last inequality comes from the ascent property of w_{k+1} in SHOICA. Also note that $G(w_0) = \phi(w_0; \hat{w}_0)$. Hence, the first set of inequalities is proved. Furthermore, using (17) and basic properties of conditional expectation, we have:

$$\mathbb{E}[\phi(w_{k+1}; \hat{w}_k)] \geq \mathbb{E}[\phi(w_k; \hat{w}_k)] = \mathbb{E}[\mathbb{E}[\phi_i(w_k; w_k)|S_k]]$$
$$= \mathbb{E}[\mathbb{E}[g_i(w_k)|S_k]] = \mathbb{E}[G(w_k)]. \tag{18}$$

Taking expectation w.r.t. the whole set of minibatches $\{S_0, \cdots S_k\}$ in (16) and combining with (18), we also get $\mathbb{E}[G(w_{k+1})] \geq \mathbb{E}[G(w_k)]$, i.e. the second set of inequalities is proved. It follows that the sequence $(\mathbb{E}[G(w_k)])_{k\geq 0}$ is monotonically nondecreasing and bounded above, since by our assumption G is bounded from above. Consequently, $(\mathbb{E}[G(w_k)])_{k\geq 0}$ converges to a finite value.

Moreover, from (17) it follows that the sequence $(\phi(w_{k+1}; \hat{w}_k))_{k\geq 0}$ is monotonically nondecreasing and bounded above with probability 1, since ϕ is lower bounded by G, which by our assumption is bounded from above. Taking expectation w.r.t. the whole set of minibatches $\{S_0, \cdots S_k\}$, we obtain that the nondecreasing sequence $(\mathbb{E}[\phi(w_{k+1}; \hat{w}_k)])_{k\geq 0}$ converges. Further, let us prove that the nonnegative quantity

$\mathbb{E}\left[g_i\left(w_{k+1}\right) - \phi_i\left(w_{k+1}; w_k\right)\right]$ is the summand of a converging sum. Indeed:

$$
\begin{aligned}
0 \leq \mathbb{E}&\left[\sum_{k=0}^{\infty}\sum_{i\in S_{k+1}} g_i(w_{k+1}) - \phi_i(w_{k+1}; w_k)\right] \\
&= \sum_{k=0}^{\infty}\mathbb{E}\left[\sum_{i\in S_{k+1}} g_i\left(w_{k+1}\right) - \phi_i\left(w_{k+1}; w_k\right)\right] \\
&= \frac{\tau}{N}\sum_{k=0}^{\infty}\mathbb{E}\left[g_i\left(w_{k+1}\right) - \phi_i\left(w_{k+1}; w_k\right)\right] \\
&= \frac{\tau}{N}\sum_{k=0}^{\infty}\mathbb{E}\left[G(w_{k+1}) - \phi\left(w_{k+1}; \hat{w}_k\right)\right],
\end{aligned}
$$

where we used the Beppo-Lèvy theorem to interchange the expectation and the sum in front of nonnegative quantities. Combining this with (18), we further get:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{k=0}^{+\infty}\sum_{i\in S_{k+1}} g_i\left(w_{k+1}\right) - \phi_i\left(w_{k+1}; w_k\right)\right] \\
&\leq \frac{\tau}{N}\sum_{k=0}^{+\infty}\mathbb{E}\left[G(w_{k+1}) - G(w_k)\right] \\
&\leq \frac{\tau}{N}\left(G_* - G(w_0)\right) < \infty,
\end{aligned}
$$

where for the last inequality we used that G is assumed upper bounded by the constant $G_* < \infty$. As a result, the nonnegative sequence $(G(w_{k+1}) - \phi(w_{k+1}; \hat{w}_k))_{k\geq 0}$ converges to 0 almost surely. □

From the previous theorem we see that SHOICA is an ascent method in expectation. Moreover, in the deterministic case (i.e., $\tau = N$) one can easily see that

$$G(w_k) = \phi(w_k; w_k) \leq \phi(w_{k+1}; w_k) \leq G(w_{k+1}),$$

and consequently SHOICA is a pure ascent method:

$$G(w_k) \geq G(w_{k-1}) \geq \cdots \geq G(w_0),$$

yielding local maxima. On the other hand, FastICA does not enjoy this important property (see also next section).

IV. Numerical experiments

In this section experiments on hyperspectral image feature reduction are shown to evaluate the performance of SHOICA algorithm relative to FastICA in terms of both quality of the solution (i.e., identifying correctly maximal points), and scalability and speed of convergence. We also validate the quality of the SHOICA solution by using the reduced hyperspectral data in classification tasks and compare the performance with other dimensionality reduction techniques such as PCA and FastICA. For experiments we use three hyperspectral images from [31]:

I.) `Indian Pines`: The spatial dimensions of this hyperspectral image are 145×145 ($N = 21025$). It has 220 bands with 20 water absorption bands being discarded, hence $b = 200$. In Figure 1, we display a colour image of Indian Pines and the groundtruth, from which we can differentiate 16 classes. See also Table I for the number of samples available in each class.

II.) `Pavia University`: The spatial dimensions of this hyperspectral image are 610×340 ($N = 207400$) and the number of bands is $b = 103$. The groundtruth image differentiates 9 classes (see Figure 2) and one can find in Table I the number of samples for each class.

III.) `Pavia`: The spatial dimensions of this hyperspectral image are 1096×715 ($N = 783640$) and the number of bands is $b = 102$. This image is used in the first set of experiments to show the scalability of SHOICA.

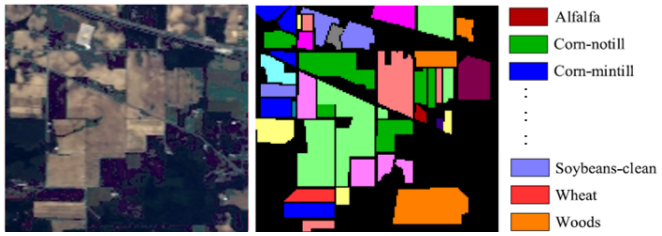

Fig. 1. Colour image and the ground truth of Indian Pines dataset.

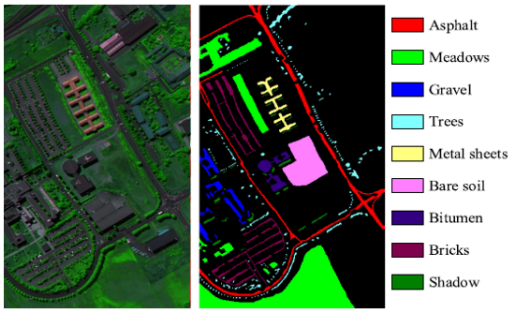

Fig. 2. Colour image and the ground truth of Pavia University dataset.

In our experiments, we work with the matrix representation of the hyperspectral cube, denoted by $X \in \mathbb{R}^{b\times N}$. For all the methods we implement, a preprocessing step (whitening) as described in Section II is applied before using the data. For the optimization problem (9), we choose from (6) the following contraction function:

$$g(w^T x_i) = \ln(\cosh(w^T x_i)) \quad \forall i = 1 : N.$$

Similar behaviours were observed for the algorithms discussed in this paper when using the other contraction

TABLE I
NUMBER OF SAMPLES BY CLASS USED IN THE EXPERIMENTS.

Data Set	No.	Class	Samples
Indian Pines	1	Alfalfa	46
	2	Corn-notill	1428
	3	Corn-mintill	830
	4	Corn	237
	5	Grass-Pasture	483
	6	Grass-Trees	730
	7	Grass-pasture-mowed	28
	8	Hay-windrowed	478
	9	Oats	20
	10	Soybean-notills	972
	11	Soybean-mintills	2455
	12	Soybean-clean	593
	13	Wheat	205
	14	Woods	1265
	15	Buildings-Grass-Trees-Drives	386
	16	Stone-Steel-Tower	93
Pavia University	1	Asphalt	6631
	2	Meadows	18649
	3	Gravel	2099
	4	Trees	3064
	5	Metal sheets	1345
	6	Bare Soil	5029
	7	Bitumen	21330
	8	Bricks	3682
	9	Shadows	947

functions in (6), when considering other initializations w_0 (including random initial points) or when different values for the reduced dimension r are considered. The algorithms are implemented in Matlab and all experiments are conducted on an Intel Core i7 and 16GB RAM. In particular, for FastICA algorithm we use the Matlab package from [28].

A. Dimensionality reduction experiments

As FastICA is the state of the art dimensional reduction method for the ICA formulation, in this section we compare our new algorithm with this method. We first evaluate the scalability of SHOICA and FastICA on previous three datasets, with dimensions ranging from $N \approx 10^4$ to $N \approx 10^6$, by inspecting the CPU time and the number of epochs (i.e., the number of passes through all the pixels N of the given dataset). First, the data is whitened and then standardised. After removing the mean, we whitened the data using both approaches described in (7). For SHOICA we consider three choices for the minibatch size: $\tau = 1$, a minibatch variant $\tau^* \in (1, N)$ (we choose different values depending on the dataset) and $\tau = N$ (i.e., deterministic variant), respectively. Both algorithms, SHOICA and FastICA, are initialized with the same random point and stopped with $\epsilon = 10^{-6}$. The results given in Tables II and III are for one row in W. From these tables, one can notice that SHOICA with $p = 1, 2$ and appropriate minibatch size τ is superior to FastICA in terms of both, number of epochs and cpu time. In particular, for small data the two methods are comparable, while when the data size increases, our algorithm is considerable faster.

TABLE II
SHOICA P=1 AND FASTICA PERFORMANCE IN TERMS OF TIME AND NUMBER OF EPOCHS FOR THREE DIFFERENT DATASETS.

Data set	Perf.	SHOICA p=1			FastICA
		$\tau = 1$	τ^*	$\tau = N$	
Inidian Pines $N \approx 10^4$	time(s)	5.12	0.695	1.707	1.21
	epochs	8	14	149	84
Pavia U $N \approx 10^5$	time(s)	76.65	21.93	28.59	35.82
	epochs	14	297	320	549
Pavia $N \approx 10^6$	time(s)	338.17	109.8	137.9	148.91
	epochs	9	108	282	598

TABLE III
SHOICA P=2 AND FASTICA PERFORMANCE IN TERMS OF NUMBER OF EPOCHS FOR THREE DIFFERENT DATASETS.

Data set	Perf.	SHOICA p=2			FastICA
		$\tau = 1$	τ^*	$\tau = N$	
Inidian Pines $N \approx 10^4$	time(s)	7.39	1.09	1.57	1.21
	epochs	9	11	83	84
Pavia U $N \approx 10^5$	time(s)	69.56	20.05	31.95	35.82
	epochs	12	145	295	549
Pavia $N \approx 10^6$	time(s)	299.71	129.4	146.19	158.91
	epochs	7	89	227	598

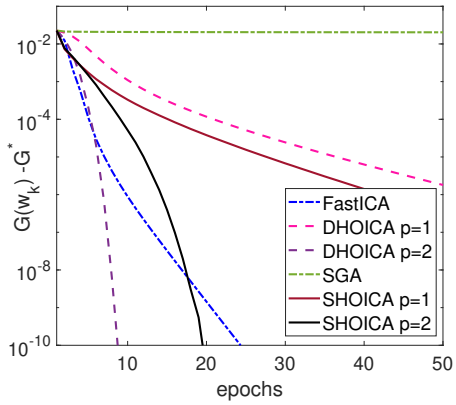

Fig. 3. Behavior of SHOICA ($\tau = 145$) and DHOICA for $p = 1, 2$, FastICA and SGA for the starting point $w_0 = e_1$ along epochs (data whitened by Q_1).

Next, we analyze the speed of convergence of several optimization algorithms for solving the ICA optimization problem (9) on the `Indian Pines` dataset: SHOICA, the deterministic variant of SHOICA (called DHOICA), Stochastic Gradient Ascent (called SGA) and FastICA. The results in terms of epochs (number of passes through data) for data whitened by the matrix Q_1 from (7) are

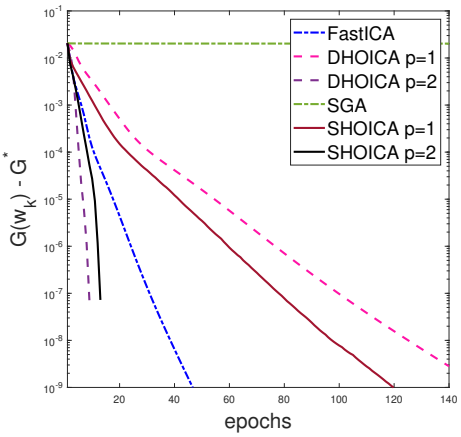

Fig. 4. Behavior of SHOICA ($\tau = 145$) and DHOICA for $p = 1, 2$, FastICA and SGA for the starting point $w_0 = e_b$ along epochs (data whitened by Q_2).

shown in Figure 3, while by the matrix Q_2 from (7) in Figure 4, respectively. Figures 3 and 4 display the behaviour of SHOICA (with a minibatch size $\tau = 145$) and DHOICA for $p = 1, 2$, FastICA, and SGA, in terms of epochs. For both $p = 1$ and $p = 2$, both algorithms, SHOICA and DHOICA, are comparable to FastICA when low accuracies are required, but our methods with $p = 2$ perform better when we target high accuracies. SGA performs poorly. Therefore, these tables and figures show the scalability feature of SHOICA.

We further analyze the quality of the solution given by SHOICA and FastICA, using different initialization for w_0. First, we use the matrix Q_1 from (7) to whiten the data. One can notice from Figure 5 that FastICA sometimes finds local minima, i.e., it minimizes the objective function G instead of maximizing it (see bottom plot). On the other hand, our ascent algorithms will always find a local maxima. In Figure 6, we display two independent components (ICs) found by SHOICA and FastICA with the initialization $w_0 = \frac{\mathbf{1}}{\|\mathbf{1}\|}$. Note that FastICA's IC is just noise and we can't distinguish any structure in the image, unlike the IC provided by SHOICA $p = 2$. Hence, typically the extracted features yielded by SHOICA are better than by FastICA. Then, we use the matrix Q_2 from (7) to whiten the data. For this whitening procedure we observe the same behaviour as before, see Figure 7. Hence, our algorithm is also robust with respect to different preprocessing procedures. Finally, in Figure 8 the first 5 most important components by PCA, FastICA and SHOICA on `Indian Pines` data set are depicted. Because PCA calculates components in the order of descending variance, we likewise sort the components

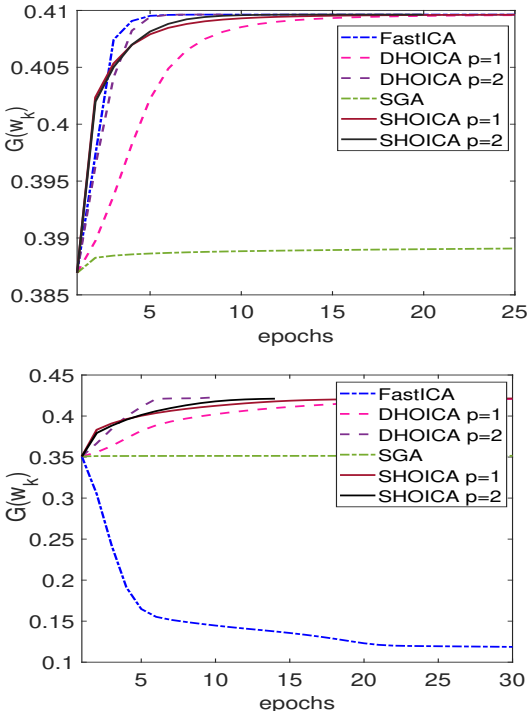

Fig. 5. Objective function G along epochs: comparison between SHOICA and DHOICA with $p = 1, 2$, FastICA and SGA on Indian Pines dataset for different initializations: top $w_0 = e_1$, bottom $w_0 = \mathbf{1}/\|\mathbf{1}\|$. In the bottom plot we observe that FastICA minimizes G instead of maximizing it (data whitened by Q_1).

from ICA according to descending objective function values.

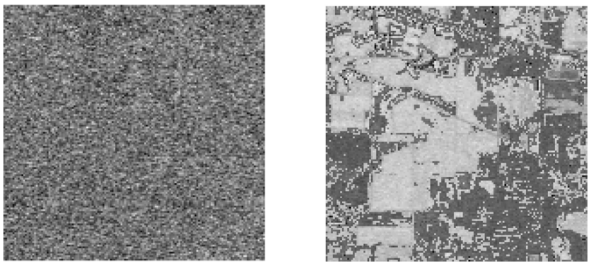

Fig. 6. Two ICs found for $w_0 = \mathbf{1}/\|\mathbf{1}\|$ on the Indian Pines dataset: FastICA on the left and SHOICA $p = 2$ on the right.

B. SVM Classification experiments

As ICA is a linear feature extraction method, when exploring the quality of the reduced data in a classification task, we consider another dimensionality reduction technique from the same category for comparison, i.e. PCA. Hence, in this section, we explore the quality of

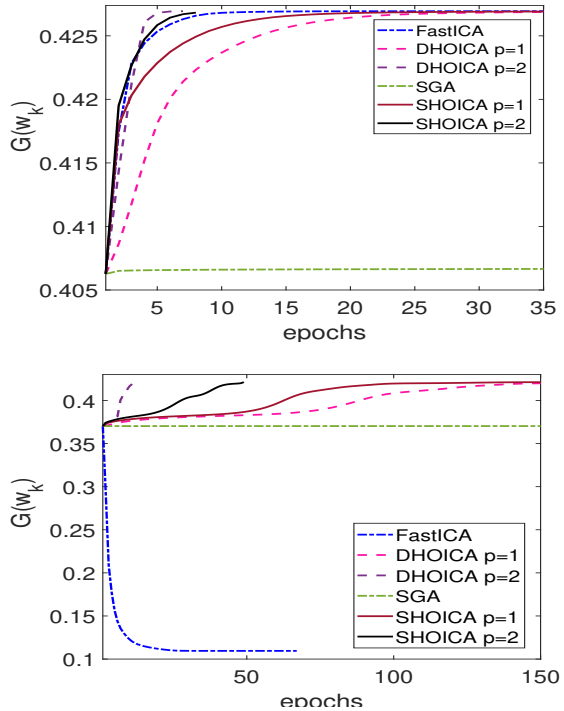

Fig. 7. Objective function G along epochs: comparison between SHOICA and DHOICA with $p = 1, 2$, FastICA and SGA on Indian Pines dataset [31] for different initializations: top $w_0 = e_b$, bottom $w_0 = \mathbf{1}/\|\mathbf{1}\|$. In the bottom plot we observe that FastICA minimizes G instead of maximizing it (data whitened by Q_2).

the reduced data obtained with SHOICA, FastICA, and PCA for `Indian Pines` and `Pavia University` datasets in a classification problem. For PCA procedure the reduced whitened data is $X_{\text{reduced}} = \Lambda^{-\frac{1}{2}}[1:r, 1:r]C[:, 1:r]^T X$, see (7). From our numerical experiments we observed that whitening in PCA, i.e., multiplication with the term $\Lambda^{-\frac{1}{2}}$, is also beneficial in classification as in ICA. We chose for this task a supervised technique called Support Vector Machine (SVM) classifier and use the Python's scikit-learn library [32] for the experiments. For a training set with M samples of dimension b, i.e $x_i \in \mathbb{R}^b$ with $i = 1 : M$, each sample has an associated label y_i that can take the values $\{-1\,, 1\}$. Then, the linear SVM problem is formulated as:

$$\begin{aligned} &\min_{t,z,\zeta\geq 0} \frac{1}{2}\|t\|^2 + C\sum_{i=1}^{M}\zeta_i \qquad (19)\\ &\text{s.t.: } y_i\left(t^T\psi(x_i) + z\right) \geq 1 - \zeta_i \quad \forall i = 1, \ldots, M, \end{aligned}$$

where in linear SVM ψ is the identity function and t and z are the hyperplane parameters that separates the data and ζ_i are slack variables that account for nonseparable data. The regularization parameter C controls the penalty

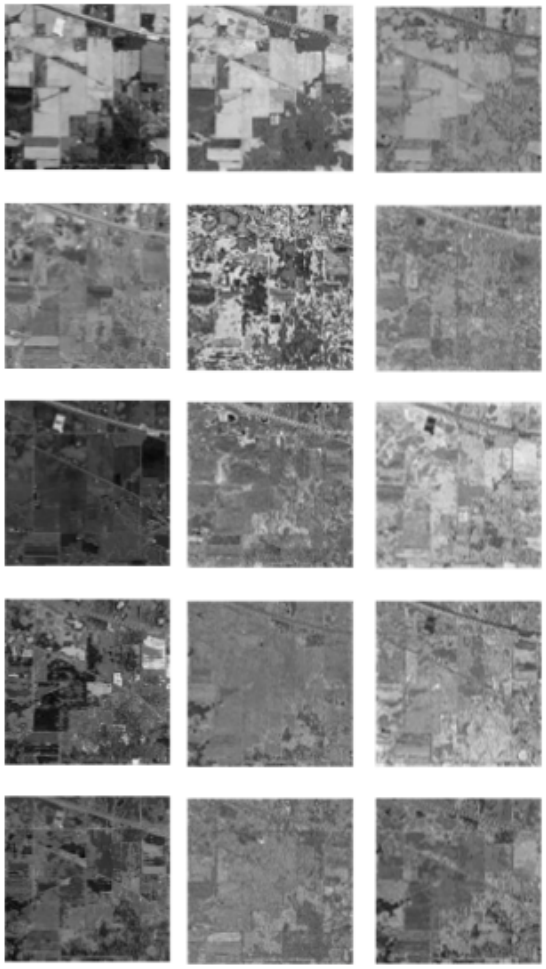

Fig. 8. First 5 most important components for Indian Pines dataset. From left to right: PCA, FastICA, SHOICA

assigned to misclassified samples. Nonlinear SVM uses a kernel formulation to map the data into a higher dimensional feature space using a transformation ψ in (19) so that the separation between the two classes which share a nonlinear boundary can be reduced to the linear case. The kernel method in the dual formulation of SVM problem is written as (19):

$$\begin{aligned} &\min_{0\leq\alpha\leq C} \frac{1}{2}\alpha^T Q\alpha - \mathbf{1}^T\alpha\\ &\text{s.t.:} \quad y^T\alpha = 0, \end{aligned}$$

where $\alpha \in \mathbb{R}^M$ is the vector of Lagrange multipliers, $Q \succeq 0$ with $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ and $K(x_i, x_j) = \psi(x_i)^T \psi(x_j)$ is the so-called kernel function. Note that one does not need the explicit form of ψ, just a kernel function. A common kernel function is the Radial Basis

TABLE IV
QUANTITATIVE COMPARISON OF DIFFERENT REDUCTION TECHNIQUES IN TERMS OF OA,AA, κ, τ ON THE INDIAN PINES DATA SET

Class	SVM Linear				SVM RBF			
No	Full	PCA	ICA	SHOICA	Full	PCA	ICA	SHOICA
1	86.66	68.88	68.88	60.86	84.44	83.33	86.66	88.88
2	82.79	50.41	52.37	54.20	78.67	76.50	77.86	78.04
3	56.20	6.50	6.62	5.78	64.15	70.36	72.10	78.97
4	61.48	0.85	0.85	1.69	76.59	70.85	69.78	72.55
5	93.09	69.89	71.44	76.85	94.12	94.63	94.53	94.43
6	98.21	97.32	96.91	97.53	98.97	97.05	97.87	96.36
7	76.00	64.00	74.00	64.28	94.00	92.00	90.00	88.00
8	99.68	99.37	98.75	99.58	99.89	99.68	99.79	99.58
9	72.50	25.00	32.50	50.00	47.50	77.50	77.50	70.00
10	70.87	32.62	30.92	33.33	74.74	76.54	75.72	81.59
11	82.11	83.80	85.11	85.09	89.42	87.39	86.74	86.49
12	81.42	4.78	4.11	4.71	83.78	75.54	74.45	75.88
13	99.02	98.53	97.80	98.03	98.04	96.34	96.58	96.34
14	96.24	97.74	97.98	96.99	97.74	98.10	97.31	95.49
15	76.23	31.16	29.35	33.67	64.80	59.87	58.57	62.46
16	96.31	94.73	93.68	93.47	95.26	92.10	90.52	90.00
OA	83.00	62.50	62.88	63.68	85.46	84.49	84.41	85.41
AA	83.05	57.85	58.83	59.75	83.88	84.23	84.12	84.69
κ	0.8053	0.5573	0.5616	0.5722	0.8249	0.8226	0.8217	0.8336
time(s)	7.25	0.17	0.17	0.17	2.99	0.90	0.92	0.91

Function (RBF):

$$K_{\text{RBF}}(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}.$$

Through the γ parameter, the influence of individual training samples on the overall algorithm can be controlled based on their proximity. In our experiments, we consider both linear SVM with parameter $C = 10$ and nonlinear SVM with the RBF kernel keeping the default value provided by Python for γ. For the multiclass case a *one against one* strategy is adopted. A total of $T(T-1)/2$ binary classifiers are trained, where T is the number of classes. The final decision is made on *winner takes it all* approach. Furthermore, the datasets are divided randomly into 80% samples from each class for the training and the remaining 20% for testing. Additionally, to reduce the influence of samples random selection, the classifier runs 10 times and we display the average results. To quantify the quality of the classifiers, we consider the average accuracies for each class, the total average accuracy (AA), the overall accuracy (OA) (see Table 2 in [33], the average training time, and the kappa coefficient (κ), which quantifies the level of agreement between the SVM output and the class labels relative to how many would be labelled correctly by chance. In our experiments we consider three dimensionality reduction techniques, PCA, FastICA and SHOICA, and the number of bands is reduced to $r = 15$ in all three approaches (see also [34], [35] for similar choices).

Tables IV and V provides a quantitative comparison of the two SVM classifiers (linear and RBF) in terms of the quality measures OA, AA, κ and τ, using Full (i.e., the original data for Indian Pines and Pavia University, respectively) and reduced data (PCA, FastICA and SHOICA). For linear SVM the best performance on both datasets is achieved when one uses the full data. However, the training time for this classifier is much longer for full data than for reduced features data. Moreover, nonlinear SVM classifiers perform better than linear SVM on both datasets. More specifically, for nonlinear SVM the best performance on both datasets is achieved when one uses reduced features data based on SHOICA. Moreover, the training times for the RBF classifiers based on data features reduction are comparable and much smaller than the training time for RBF classifier based on full data. Better overall performance is achieved on Pavia University dataset than on Indian Pines dataset (see also Figures 9 and 10, which display the classification maps for Indian Pines and Pavia University, respectively). We attribute this to the fact that on Pavia University dataset the classes are more balanced in terms of number of samples than on Indian Pines dataset. In conclusion, the training time and performance accuracies are better for SHOICA feature reduction approach, making our numerical algorithm a reliable framework for dimensionality reduction and classification of hyperspectral images.

TABLE V
QUANTITATIVE COMPARISON OF DIFFERENT REDUCTION TECHNIQUES IN TERMS OF OA, AA, κ AND TIME ON PAVIA UNIVERSITY DATASET

Class No	SVM Linear				SVM RBF			
	Full	PCA	ICA	SHOICA	Full	PCA	ICA	SHOICA
1	88.22	91.70	93.46	92.35	96.06	95.17	94.57	95.92
2	97.14	97.16	97.32	97.34	98.52	97.75	97.22	97.95
3	70.11	66.66	66.10	67.80	82.04	67.80	70.54	77.92
4	87.87	79.17	82.73	80.68	96.49	92.10	95.82	94.56
5	99.92	100.0	99.94	100.0	99.96	99.85	99.86	99.81
6	44.96	33.32	23.07	35.86	91.45	89.02	77.71	92.19
7	38.87	8.72	13.87	19.51	88.12	77.14	82.21	88.23
8	65.19	68.38	61.37	70.47	91.11	91.41	88.77	91.05
9	98.73	95.99	98.77	99.89	99.89	99.78	99.68	99.94
OA	83.19	80.86	79.84	83.10	94.87	92.37	92.05	95.04
AA	76.78	71.23	70.74	73.77	93.74	90.00	89.06	93.86
κ	0.7703	0.7359	0.7214	0.7535	0.9398	0.9118	0.8939	0.9328
time(s)	25.31	0.50	0.49	0.45	45.48	2.97	3.12	2.89

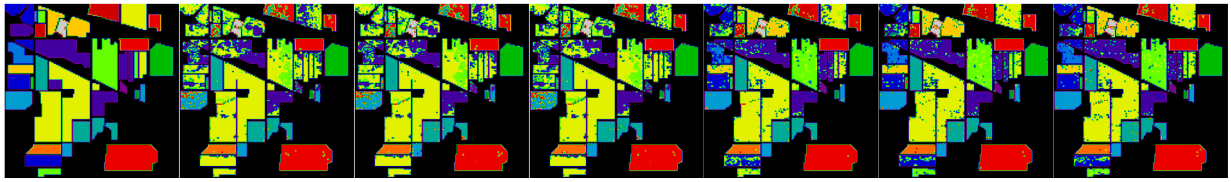

Fig. 9. Classification maps for Indian Pines, from left to right: groundtruth, PCA, ICA, SHOICA for SVM linear; PCA, ICA, SHOICA for SVM RBF.

V. CONCLUSIONS

Independent Component Analysis is a quick and effective way to extract signals from hyperspectral data. FastICA is an often-used optimization algorithm for solving the ICA problem, which is an efficient technique to reduce the dimension of hyperspectral images. However, FastICA can find irrelevant stationary points and is not scalable as it uses at each iteration the whole set of pixels. In this paper, we have designed a new stochastic higher-order Taylor approximation-based algorithm adapted to ICA problem. Our algorithm guarantees ascent, hence it able to identify local maxima. Moreover, the algorithm, since it is stochastic, is scalable. Detailed numerical simulations have shown the superior performance of our method compared to FastICA on both, dimensionality reduction and classification tasks.

VI. APPENDIX

In this appendix we show that ADMM algorithm can be applied easily for solving the subproblem appearing in step 3 of ScaleICA for $p = 2$ and $\tau = N$. Note that [29] shows the efficiency of ADMM on orthogonality constraint problems through numerical results. Recall that the objective function of subproblem in this case is:

$$\phi(w; \hat{w_k}) = \langle \nabla G(w_k), w - w_k \rangle + \frac{1}{2}\langle \nabla^2 G(w_k)(w - w_k), w - w_k \rangle - \frac{M_2}{6}\|w - w_k\|^3.$$

The subproblem can be reformulated equivalently as:

$$\begin{aligned} &\max_{w,u\in\mathbb{R}^b} \ \phi(w; w_k) + \mathbf{1}_{\mathcal{B}}(u) \\ &\text{s.t.: } \ w = u, \end{aligned} \tag{20}$$

where $\mathbf{1}_{\mathcal{B}}(\cdot)$ is the indicator function of the set $\mathcal{B}$. Let us denote the augumented Lagrangian for subproblem (20) with:

$$\mathcal{L}_\rho(w, u, \lambda) = \phi(w; \hat{w_k}) + \langle \lambda, w - u \rangle - \frac{\rho}{2}\|w - u\|^2 + \mathbf{1}_{\mathcal{B}}(u),$$

where $\rho > 0$ is a penalty term. Then, ADMM iterations are as follows:

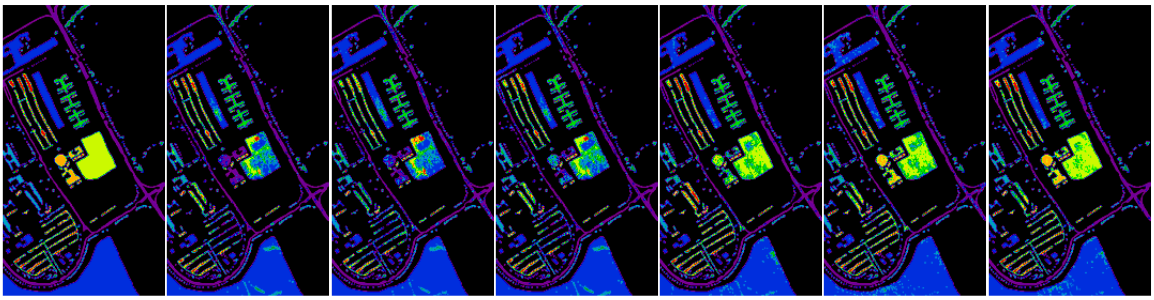

Fig. 10. Classification maps for Pavia University, from left to right: groundtruth, PCA, ICA, SHOICA for SVM linear, PCA, ICA, SHOICA for SVM RBF.

Algorithm 3 ADMM

Given w_0,u_0, λ_0, ρ, ϵ.

while $\delta \geq \epsilon$:

1. $w_{k+1} = \arg\max_w \mathcal{L}_\rho(w, u_k, \lambda_k)$
2. $u_{k+1} = \arg\max_u \mathcal{L}_\rho(w_{k+1}, u, \lambda_k)$
3. $\lambda_{k+1} = \lambda_k - \rho(w_{k+1} - u_{k+1})$
4. Update stopping criterion
   $\delta = \max(\|w_{k+1} - u_{k+1}\|, \|u_{k+1} - u_k\|)$
5. Increase k.

Let us show that steps 1 and 2 in ADMM can be performed efficiently. More precisely, if we denote $v = w - w_k$, then step 1 requires solving the following unconstrained problem:

$$\max_{v \in \mathbb{R}^b} \frac{1}{2} v^T Q v + q^T v - \frac{M_2}{6} \|v\|^3,$$

where $Q = \nabla^2 G(w_k) - \rho I_b$ and $q = \nabla G(w_k) + \lambda_k + \rho(u_k - w_k)$. Note that the objective is a sum between a quadratic term and a cubic term and this can be solved very efficiently using e.g. the technique from [25]. Moreover, step 2 can be computed explicitly, since it is just a projection onto a sphere:

$$\begin{aligned} u_{k+1} &= \arg\max_{u^T u = 1} -\frac{\rho}{2} \|w_{k+1} - u\|^2 - \langle \lambda_k, u \rangle \\ &= \arg\min_{u^T u = 1} \frac{\rho}{2} \|u - w_{k+1} - \lambda_k/\rho\|^2 \\ &= \frac{\rho w_{k+1} + \lambda_k}{\|\rho w_{k+1} + \lambda_k\|}. \end{aligned}$$

ACKNOWLEDGMENT

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

REFERENCES

[1] Muhammad Jaleed Khan, Hamid Saeed Khan, Adeel Yousaf, Khurram Khurshid, and Asad Abbas. Modern trends in hyperspectral image analysis: A review. *Ieee Access*, 6:14118–14129, 2018.

[2] Pedram Ghamisi, Naoto Yokoya, Jun Li, Wenzhi Liao, Sicong Liu, Javier Plaza, Behnood Rasti, and Antonio 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.

[3] Sivert Bakken, Milica Orlandic, and Tor Arne Johansen. The effect of dimensionality reduction on signature-based target detection for hyperspectral remote sensing. In *CubeSats and SmallSats for Remote Sensing III*, volume 111310L, 2019.

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

[5] Behnood Rasti, Danfeng Hong, Renlong Hang, Pedram Ghamisi, Xudong Kang, Jocelyn Chanussot, and Jon Atli Benediktsson. Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):60–88, 2020.

[6] Danfeng Hong, Wei He, Naoto Yokoya, Jing Yao, Lianru Gao, Liangpei Zhang, Jocelyn Chanussot, and Xiaoxiang Zhu. Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing. *IEEE Geoscience and Remote Sensing Magazine*, 9(2):52–87, 2021.

[7] Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655, 2017.

[8] Lunjun Wan, Ke Tang, Mingzhi Li, Yanfei Zhong, and Kai Qin. Collaborative active and semisupervised learning for hyperspectral remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2384–2396, 2014.

[9] Yi Chen, Nasser Nasrabadi, and Trac Tran. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10):3973–3985, 2011.

[10] Anil Cheriyadat and Lori Bruce. Why principal component analysis is not an appropriate feature extraction method for hyperspectral data. In *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium*, volume 6, pages 3420–3422, 2003.

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

[12] Erkki Oja Aapo Hyvärinen, Juha Karhunen. *Independent component analysis*. John Wiley & Sons, 2001.

[13] Jing Wang and Chein-I Chang. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1586–1600, 2006.

[14] Guangchun Luo, Guangyi Chen, Ling Tian, Ke Qin, and Shen-En 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.

[15] Nicola Falco, Lorenzo Bruzzone, and Jon Atli Benediktsson. An ica based approach to hyperspectral image feature reduction. In *2014 IEEE Geoscience and Remote Sensing Symposium*, pages 3470–3473, 2014.

[16] Farid Melgani and Lorenzo Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790, 2004.

[17] Shutao Li, Weiwei Song, Leyuan Fang, Yushi Chen, Pedram Ghamisi, and Jon Atli Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6690–6709, 2019.

[18] Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non-gaussian signals. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 362–370, 1993.

[19] Shun-ichi Amari, Andrzej Cichocki, and Howard Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8, 1995.

[20] Pierre Ablin, Alexandre Gramfort, Jean-François Cardoso, and Francis Bach. Stochastic algorithms with descent guarantees for ica. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1564–1573. PMLR, 2019.

[21] Leon Bottou. *Stochastic Gradient Descent Tricks*. In: Montavon G., Orr G.B., Muller KR. (eds), Neural Networks: Tricks of the Trade, Springer, 2012.

[22] Stephan Clemencon, Aurelien Bellet, Ons Jelassi, and Guillaume Papa. Scalability of stochastic gradient descent based on "smart" sampling techniques. *Procedia Computer Science*, 53:308–315, 2015.

[23] Daniela Lupu and Ion Necoara. Convergence analysis of stochastic higher-order majorization-minimization algorithms. *arXiv preprint:2103.07984*, 2021.

[24] Vicente Zarzoso, Pierre Comon, and Mariem Kallel. How fast is fastica? In *2006 14th European Signal Processing Conference*, pages 1–5, 2006.

[25] Yurii Nesterov and Boris Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[26] Yurii Nesterov. Inexact basic tensor methods. *CORE Discussion Papers*, 23, 2019.

[27] Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

[28] FastICA software. https://research.ics.aalto.fi/ica/fastica.

[29] Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, 2014.

[30] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26, 2013.

[31] Database of Hyperspectral images. http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

[32] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

[34] Nicola Falco, Lorenzo Bruzzone, and Jon Atli Benediktsson. A comparative study of different ica algorithms for hyperspectral image analysis. In *5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2013.

[35] Behnood Rasti, Magnus Orn Ulfarsson, and Johannes Sveinsson. Hyperspectral feature extraction using total variation component analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):6976–6985, 2016.