SPARSE GAUSSIAN NOISY INDEPENDENT COMPONENT ANALYSIS

Frosti Palsson, Magnus O. Ulfarsson and Johannes R. Sveinsson

Dept. of Electrical Eng., University of Iceland, Reykjavik, ICELAND

ABSTRACT

There are two main approaches to independent component analysis (ICA); maximization of non-Gaussianity of the sources and the exploitation of temporal correlation in Gaussian sources. In this paper, we present a novel sparse noisy ICA model where we have introduced temporal correlation in the sources, described by a first order auto regressive (AR(1)) process. The correlation structure of the sources eliminates the rotational invariance of the estimates, enabling their separation. Using simulated data, we demonstrate both source separation and denoising, where we compare our results to a sparse PCA method and the fastICA method. Additionally, we apply the method on a real hyperspectral dataset.

Index Terms— Independent Component Analysis, Sparsity, Noisy Principal Component Analysis, Source Separation, Denoising

1. INTRODUCTION

Principal component analysis (PCA) [1], also known as the Karhunen-Loeve expansion, plays an important role in signal processing, e.g., for exploratory signal analysis and dimensionality reduction. PCA decomposes a signal into principal components (PCs) which are orthogonal and ordered according to their variance. The first PC explains most of the variance of the signal, while the next PC is orthogonal to the first PC and explains second most of the variance of the signal, and so on.

For PCA there is an underlying signal processing model [2],[3] called noisy PCA (nPCA). Recent generalizations of nPCA are, e.g., smooth nPCA [4] and sparse variable nPCA (svnPCA) [5]. The main idea of svnPCA is the incorporation of a sparseness vector penalty for automatic variable selection. This is achieved by maximizing a vector ℓ_0 penalized log-likelihood function using the Expectation-Maximization (EM) algorithm [6],[7].

Independent component analysis (ICA) [8] is a technique to separate mixed signals (sources) based on their statistical independence. There are two main approaches to ICA. One is the maximization of the non-Gaussianity of the estimated sources and the other approach is to exploit sample dependence in Gaussian sources, i.e., that the samples of each source are correlated. Methods that fall into this category are called Gaussian noisy ICA [9].

The main contribution of this paper is a novel sparse Gaussian noisy ICA method, where the sources (PCs) are assumed to have temporal correlation which is described by a first order auto regressive (AR(1)) process. The model is related to the svnPCA model, however, the source estimates in svnPCA are invariant under rotation, making separation impossible. The correlation structure of the sources in the new method eliminates the rotation invariance and makes the sources separable.

A further generalization of the model is achieved by assuming that the signal under study is sparse when expressed in a basis such as the orthogonal wavelet basis, which is the choice of basis in this work. We call the new method sparse Gaussian noisy ICA (sgnICA).

The proposed method is demonstrated using simulated data and we compare its source separation and denoising performance to the sparse PCA (sPCA) method presented in [10] and the fastICA [11] method.

In the sPCA method, the data is transformed to a basis in which the PCs are sparse, using the orthogonal wavelet transform. In [10], it is shown that the PCs can be consistently estimated by restricting the PCA to a subset of the variables with variances above a threshold. Instead of adaptively selecting the threshold, the top k variables are chosen according to their variance. The next step is performing reduced PCA on this subset, retaining the leading r PCs. Finally, the data is reconstructed using the inverse PCA transform and returning to the original basis via the inverse wavelet transform.

The organization of the paper is as follows. In Section 2 we derive the sgnICA algorithm. In Section 3 we discuss parameter selection for the proposed method. Section 4 describes the experiments using simulated data and in Section 5 we present denoising of a real hyperspectral remote sensing dataset. Finally, conclusions are drawn in Section 6.

2. THE sgnICA MODEL

The sgnICA model is given by

$$\boldsymbol{y}_t = \boldsymbol{G}\boldsymbol{u}_t + \boldsymbol{\epsilon}_t \quad (1)$$

$$\boldsymbol{u}_t = \rho\boldsymbol{u}_{t-1} + \boldsymbol{\eta}_t, \quad t = 1, ..., T \quad (2)$$

This work was partly supported by the Research Fund of the University of Iceland and the Icelandic Research Fund (130635-051).

where $\boldsymbol{y}_t$ is an $M \times 1$ (zero mean) vector of observations, $\boldsymbol{G}$ is an $M \times r$ mixing matrix, $\boldsymbol{u}_t$ is an $r \times 1$ matrix of independent components, ρ is the AR(1) parameter, $\boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_M)$, $\boldsymbol{\eta}_t \sim N(\boldsymbol{0}, \boldsymbol{I}_r)$ and $\boldsymbol{u}_t$ and $\boldsymbol{\epsilon}_t$ are independent.

We assume that the signal $\boldsymbol{G}\boldsymbol{u}_t$ is sparse when expressed in the basis $\boldsymbol{\Phi}$ and thus we write $\boldsymbol{G} = \boldsymbol{\Phi}\boldsymbol{B}$, where $\boldsymbol{\Phi}$ a 2D orthogonal wavelet transform. The normalized frequency domain log-likelihood function [12],[13] for this model is given by

$$l_{\boldsymbol{\theta}}(\tilde{\boldsymbol{Y}}) = -\frac{1}{2}\sum_{k=1}^{T}\left(\text{tr}(\boldsymbol{\Omega}_k^{-1}\boldsymbol{S}_k) + \log|\boldsymbol{\Omega}_k|\right),$$

where $\boldsymbol{\Omega}_k = \boldsymbol{\Phi}\boldsymbol{B}\boldsymbol{F}_k\boldsymbol{B}^H\boldsymbol{\Phi}^T + \sigma^2\boldsymbol{I}_M$, $\boldsymbol{S}_k = \frac{\tilde{\boldsymbol{y}}_k\tilde{\boldsymbol{y}}_k^H}{T}$, $\tilde{\boldsymbol{y}}_k$ and $\tilde{\boldsymbol{u}}_k$ contain the Fourier transforms of $\boldsymbol{y}_t$ and $\boldsymbol{u}_t$, respectively, $\boldsymbol{\theta} = (\boldsymbol{B}, \sigma^2, \boldsymbol{F}_1, ..., \boldsymbol{F}_T)$, $\tilde{\boldsymbol{Y}} = [\tilde{\boldsymbol{y}}_k^T]$ and

$$\begin{aligned}\boldsymbol{F}_k &= \text{diag}(F_{kj}), \quad k = 1, ..., T\\ F_{kj} &= \frac{1}{|1-\rho_j e^{-j\omega_k}|^2}, \quad j = 1, ..., r\end{aligned}$$

where $\omega_k = \frac{2\pi}{T}k$. To enforce sparseness we introduce a penalized log-likelihood function

$$J_{\boldsymbol{\theta}}(\tilde{\boldsymbol{Y}}) = l_{\boldsymbol{\theta}}(\tilde{\boldsymbol{Y}}) - \frac{h}{2}\sum_{v=1}^{M}|||\boldsymbol{b}_v|||_0,$$

where $\boldsymbol{B} = [\boldsymbol{b}_v^T]$ and h is a tuning parameter. The ℓ_0 penalty is frequently used in wavelet analysis where it leads to hard-thresholding [14]. The notation $|||\boldsymbol{b}_v|||_0 = I(||\boldsymbol{b}_v|| > 0)$ is used to indicate that this is a vector ℓ_0 penalty, where the indicator function I is 1 if $\|\boldsymbol{b}_v\| > 0$ and is zero otherwise.

2.1. Estimation

The EM algorithm [6] offers a reliable way for maximizing the frequency domain log-likelihood. The main idea behind the algorithm is to use a surrogate function instead of directly maximizing the log-likelihood function. The algorithm consists of two steps. In the E-step, the surrogate function, which is called the EM functional is constructed, while in the M-step, the previously constructed EM-functional is maximized. The algorithm iterates between E- and M-steps until it converges.

The penalized complete log-likelihood is given by

$$\begin{aligned}J_{\boldsymbol{\theta}}(\tilde{\boldsymbol{Y}}, \tilde{\boldsymbol{U}}) &= \sum_{k=1}^{T}\left(-\frac{M}{2}\log\sigma^2 - \frac{\|\tilde{\boldsymbol{y}}_k - \boldsymbol{\Phi}\boldsymbol{B}\tilde{\boldsymbol{u}}_k\|^2}{2\sigma^2}\right.\\ &- \left.\frac{1}{2}\tilde{\boldsymbol{u}}_k^T\boldsymbol{F}_k^{-1}\tilde{\boldsymbol{u}}_k - \frac{1}{2}\log|\boldsymbol{F}_k|\right)\\ &- \frac{hT}{2}\sum_{v=1}^{M}|||\boldsymbol{b}_v|||_0,\end{aligned}$$

where $\tilde{\boldsymbol{U}} = [\tilde{\boldsymbol{u}}_k^T]$.

In the E-step, we construct the penalized EM functional

$$\begin{aligned}\text{EM}(\boldsymbol{\theta}_0, \boldsymbol{\theta}) &= E_0(J_{\boldsymbol{\theta}}(\tilde{\boldsymbol{Y}}, \tilde{\boldsymbol{U}})|\tilde{\boldsymbol{Y}})\\ &= -\frac{\text{tr}(\tilde{\boldsymbol{S}}_y)}{2\sigma^2} + \frac{\text{tr}(\boldsymbol{B}\boldsymbol{C}_0^H)}{\sigma^2} - \frac{\text{tr}(\boldsymbol{B}\boldsymbol{A}_0\boldsymbol{B}^H)}{2\sigma^2}\\ &- \frac{1}{2}\sum_{k=1}^{T}\text{tr}(\boldsymbol{F}_k^{-1}\boldsymbol{A}_k) - \frac{1}{2T}\sum_{k=1}^{T}\log|\boldsymbol{F}_k|\\ &- \frac{M}{2}\log\sigma^2 - \frac{h}{2}\sum_{v=1}^{M}|||\boldsymbol{b}_v|||_0,\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{A}_k &= \frac{1}{T}E_0[\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{u}}_k^H|\tilde{\boldsymbol{y}}_k]\\ &= \frac{1}{T}\left(\sigma_0^2\boldsymbol{W}_k^{-1} + \langle\tilde{\boldsymbol{u}}_k\rangle\langle\tilde{\boldsymbol{u}}_k\rangle^H\right)\\ \boldsymbol{A}_0 &= \sum_{k=1}^{T}\boldsymbol{A}_k\\ \boldsymbol{W}_k &= \boldsymbol{F}_{k0}^{-1}\sigma_0^2 + \boldsymbol{B}_0^H\boldsymbol{B}_0\\ \langle\tilde{\boldsymbol{u}}_k\rangle &= E[\tilde{\boldsymbol{u}}_k|\tilde{\boldsymbol{y}}_k] = \boldsymbol{W}_k^{-1}\boldsymbol{B}^H\boldsymbol{\Phi}^T\tilde{\boldsymbol{y}}_k\\ \boldsymbol{C}_0 &= [\boldsymbol{c}_{v0}^T] = \frac{1}{T}\sum_{k=1}^{T}\boldsymbol{\Phi}^T\tilde{\boldsymbol{y}}_k\langle\tilde{\boldsymbol{u}}_k\rangle^H\end{aligned}$$

and $\boldsymbol{\theta}_0 = (\boldsymbol{B}_0, \sigma_0^2, \boldsymbol{F}_{10}, ..., \boldsymbol{F}_{T0})$ denotes the current iterate in the EM algorithm. We denote the current iterate of a variable by subscript 0 and the next iterate by subscript 1.

In the M-step, we maximize the penalized EM functional. First we maximize with respect to $\boldsymbol{G}$, this is equivalent to minimizing

$$f(\boldsymbol{B}) = \sum_{v=1}^{M}\left(\frac{1}{2}\boldsymbol{b}_v^H\boldsymbol{A}_0\boldsymbol{b}_v - \boldsymbol{b}_v^H\boldsymbol{c}_{v0} + \frac{h}{2}|||\boldsymbol{b}_v|||_0\right).$$

The solution to this optimization problem is

$$\boldsymbol{b}_{v1} = \boldsymbol{A}_0^{-1}\boldsymbol{c}_{v0}I(\boldsymbol{c}_{v0}^H\boldsymbol{A}_0^{-1}\boldsymbol{c}_{v0} \geq h), \quad v = 1, .., M.$$

Maximization of the EM function w.r.t. σ^2 yields

$$\sigma_1^2 = \frac{1}{M}\left(\text{tr}(\tilde{\boldsymbol{S}}_y) - \sum_{v\in I_a}\boldsymbol{c}_{v0}^H\boldsymbol{A}_0^{-1}\boldsymbol{c}_{v0}\right),$$

where I_a is an index set for the active (non-zero) variables. Finally we get $F_{kj1} = \frac{1}{|1-\rho_{j1}e^{-j\omega_k}|^2}$.

We assume that the values of ρ_j are known. The sgnICA algorithm is given in Algorithm 1.

3. TUNING PARAMETER SELECTION

The proposed algorithm has two tuning parameters, r which is the number of components and h, which is the sparsity

Algorithm 1: The sgnICA algorithm

Input: Data matrix $\boldsymbol{Y}$, r, $\boldsymbol{\Phi}$, and h

Initialization: $\boldsymbol{B}_0, \sigma_0^2,\ \rho_j,\ j = 1, ..., r.$

while $(\|\boldsymbol{B}_1 - \boldsymbol{B}_0\|_F^2 / \|\boldsymbol{B}_0\|_F^2 > \delta)$ **do**

$F_{kj} = \frac{1}{|1-\rho_{j0}e^{-j\omega_k}|^2},\ j = 1, ..., r;\ k = 1, ..., T$

$\boldsymbol{W}_k = \boldsymbol{F}_{k0}^{-1}\sigma_0^2 + \boldsymbol{B}_0^H\boldsymbol{B}_0,\ k = 1, ..., T$

$\langle\tilde{\boldsymbol{u}}_k\rangle = \boldsymbol{W}_k^{-1}\boldsymbol{B}^H\boldsymbol{\Phi}^T\tilde{\boldsymbol{y}}_k,\ k = 1, ..., T$

$\boldsymbol{C}_0 = [\boldsymbol{c}_{v0}^T] = \frac{1}{T}\sum_{k=1}^T \boldsymbol{\Phi}^T\tilde{\boldsymbol{y}}_k\langle\tilde{\boldsymbol{u}}_k\rangle^H,\ v = 1, ..., M$

$\boldsymbol{A}_k = \frac{1}{T}\left(\sigma_0^2\boldsymbol{W}_k^{-1} + \langle\tilde{\boldsymbol{u}}_k\rangle\langle\tilde{\boldsymbol{u}}_k\rangle^H\right),\ k = 1, ..., T$

$\boldsymbol{A}_0 = \sum_{k=1}^T \boldsymbol{A}_k$

$\boldsymbol{b}_{v1} = \boldsymbol{A}_0^{-1}\boldsymbol{c}_{v0}I(\boldsymbol{c}_{v0}^H\boldsymbol{A}_0^{-1}\boldsymbol{c}_{v0} \geq h),\ v = 1, .., M.$

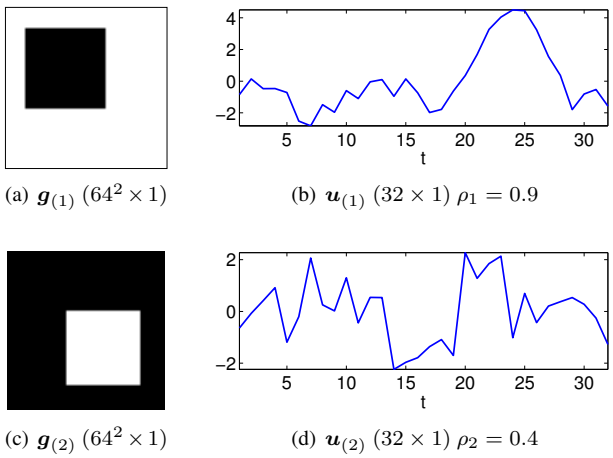
$\sigma_1^2 = \frac{1}{M}\left(\text{tr}(\tilde{\boldsymbol{S}}_y) - \sum_{v\in I_a}\boldsymbol{c}_{v0}^H\boldsymbol{A}_0^{-1}\boldsymbol{c}_{v0}\right)$

Output: $\hat{\boldsymbol{G}} = \boldsymbol{\Phi}\hat{\boldsymbol{B}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{Y}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{G}}^T$ and σ^2

penalty. The Bayesian Information Criterion (BIC) [15] is a classical choice for parameter selection for this kind of model. It is based on the likelihood function and a term that penalizes for the number of parameters in the model. It is given by

$$\text{BIC}_{r,h} = -2\ell_\theta(\tilde{\boldsymbol{Y}}) + \dim(\hat{\boldsymbol{\theta}})\log(T)$$

where $\dim(\hat{\boldsymbol{\theta}}) = M_h r - r(r-1)/2 + 1$ is the effective number of variables and M_h is the number of active variables which are kept by the algorithm. The values of the tuning parameters that correspond to the minimum of BIC are chosen.



(a) $\boldsymbol{g}_{(1)}$ $(64^2 \times 1)$ (b) $\boldsymbol{u}_{(1)}$ (32×1) $\rho_1 = 0.9$

(c) $\boldsymbol{g}_{(2)}$ $(64^2 \times 1)$ (d) $\boldsymbol{u}_{(2)}$ (32×1) $\rho_2 = 0.4$

Fig. 1: The simulated dataset is generated by $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{G}^T$, where $\boldsymbol{g}_{(1)}$ and $\boldsymbol{g}_{(2)}$ are the components (columns) of the mixing matrix $\boldsymbol{G}$ while $\boldsymbol{u}_{(1)}$ and $\boldsymbol{u}_{(2)}$ are the columns of $\boldsymbol{U}$, i.e., the source signals.

4. SIMULATION

We simulated data according to (1)-(2) using 2 source signals with the AR(1) parameter ρ chosen as 0.9 and 0.4, respectively. The simulated dataset is generated by $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{G}^T$ where $\boldsymbol{G}$ is a $64^2 \times 2$ matrix and $\boldsymbol{U}$ is 32×2, giving data matrix $\boldsymbol{Y}$ of dimension 32×64^2, i.e., $T = 32$ and $M = 64^2$. The columns of $\boldsymbol{G}$ which contain an image of a rectangle of unit magnitude, -1 and 1, respectively, and the columns of $\boldsymbol{U}$, i.e., the sources, are shown in Figure 1. Figure 2 shows the BIC and signal-to-noise ratio (SNR) values as a function of h for the simulated dataset.

We perform two experiments using simulated data and compare our results to sPCA and fastICA. In the first experiment we demonstrate the separation of the sources and in the second experiment we consider denoising.

For the separation experiment we added Gaussian noise to the simulated data giving SNR of 5 dB and then we computed the estimates of $\boldsymbol{G}$ and $\boldsymbol{U}$ using all the methods. The results are shown in Figures 3 and 4, respectively. The percentage of active variables is 8.18% for sgnICA and 11.16% for sPCA.

In Figure 3, we see that sgnICA separates the components of $\boldsymbol{G}$ significantly better than the other methods. The degree of separation is measured as the angle (in degrees) between the estimate and the true signal. In Figure 4, the sgnICA method is shown to give near perfect estimates of the sources, as measured by the correlation between the estimate and the true source.

For the denoising experiment, we used the same simulatated dataset as before and we added varying amounts of Gaussian noise, ranging from SNR of -5 dB to 5 dB in steps of 1 dB. For each experiment, the denoised data is computed from the estimates of $\boldsymbol{G}$ and $\boldsymbol{U}$, i.e., $\hat{\boldsymbol{Y}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{G}}^T$. This was repeated 50 times and the final SNR value is the mean value of all the trials. The results are shown in Figure 5. Again, sgnICA significantly outperforms the other methods in every experiment.

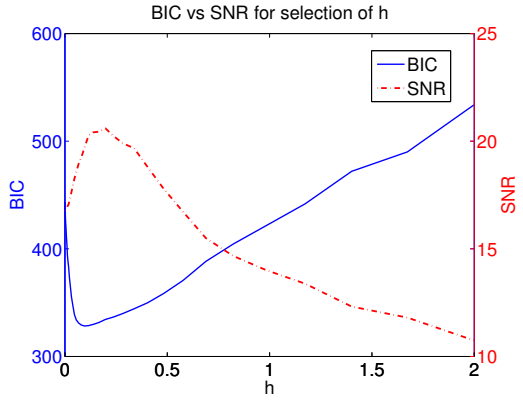
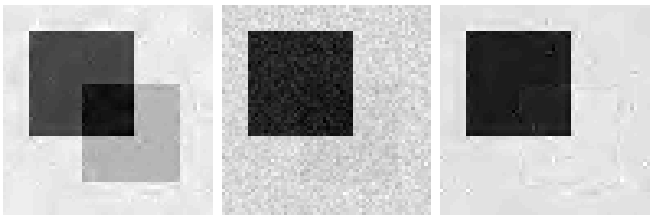
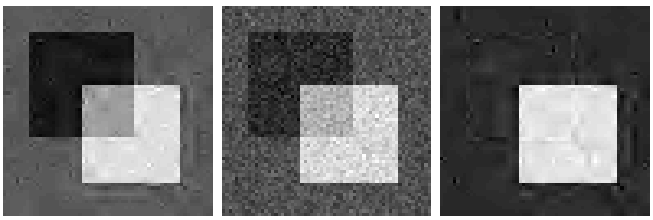


Fig. 2: Comparison of BIC and SNR for selection of the sparsity penalty parameter h for the first experiment.

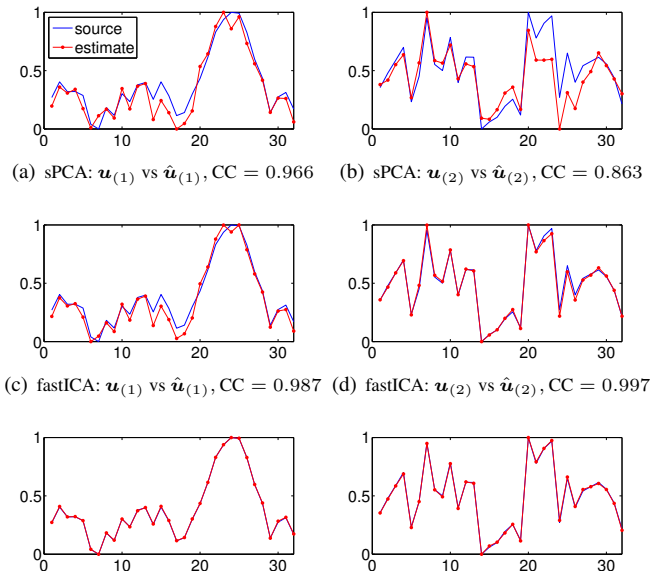


(a) sPCA $\hat{\boldsymbol{g}}_{(1)} \angle 15.3°$ (b) fastICA $\hat{\boldsymbol{g}}_{(1)} \angle 7.2°$ (c) sgnICA $\hat{\boldsymbol{g}}_{(1)} \angle 2.8°$



(d) sPCA $\hat{\boldsymbol{g}}_{(2)} \angle 31.1°$ (e) fastICA $\hat{\boldsymbol{g}}_{(2)} \angle 21.6°$ (f) sgnICA $\hat{\boldsymbol{g}}_{(2)} \angle 4.7°$

Fig. 3: The estimated components of $\boldsymbol{G}$ and their separation as given by the angle (in degrees, denoted $\angle$) between the estimate and the true signal.



(a) sPCA: $\boldsymbol{u}_{(1)}$ vs $\hat{\boldsymbol{u}}_{(1)}$, CC $= 0.966$ (b) sPCA: $\boldsymbol{u}_{(2)}$ vs $\hat{\boldsymbol{u}}_{(2)}$, CC $= 0.863$

(c) fastICA: $\boldsymbol{u}_{(1)}$ vs $\hat{\boldsymbol{u}}_{(1)}$, CC $= 0.987$ (d) fastICA: $\boldsymbol{u}_{(2)}$ vs $\hat{\boldsymbol{u}}_{(2)}$, CC $= 0.997$

(e) sgnICA: $\boldsymbol{u}_{(1)}$ vs $\hat{\boldsymbol{u}}_{(1)}$, CC $= 1.000$ (f) sgnICA: $\boldsymbol{u}_{(2)}$ vs $\hat{\boldsymbol{u}}_{(2)}$, CC $= 0.999$

Fig. 4: Normalized estimates of the sources for all methods. CC is the correlation coefficient between the estimate and the true source.

5. REAL HYPERSPECTRAL DATA

In remote sensing, a hyperspectral image is a cube of images where each image represents a certain tight band of frequencies in the electromagnetic spectrum. A typical dataset can contain hundreds of bands, covering a wide band of frequencies. The dataset used here, Indian Pines [1], is of dimension 128×128 pixels and 220 spectral bands in the wavelength range of $0.4 - 2.5\mu$m. There are a number of noisy bands which make their analysis difficult. Figure 6 shows such a band and the denoised band using the proposed method.

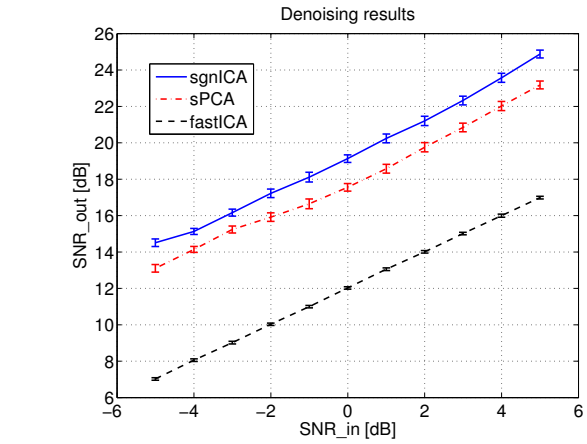


Fig. 5: Results for the denoising experiment. The standard deviation of each experiment (50 trials) is shown by errorbars.

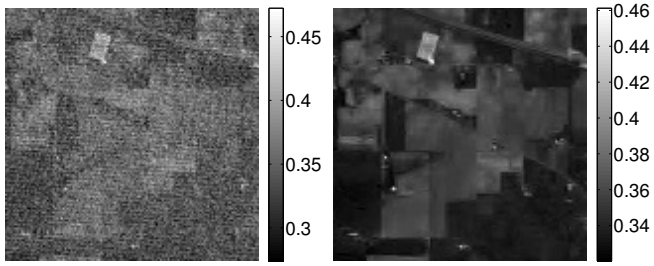


Fig. 6: A very noisy band from the hyperspectral dataset and the denoised band using the proposed method.

6. CONCLUSIONS

In this paper we have presented a novel sparse Gaussian noisy ICA algorithm, where the sources have a temporal correlation described by an AR(1) process. The proposed method was compared to a sparse PCA based method and the fastICA method, using simulated data and was demonstrated to give significantly better separation of the source signals and also shown to give significantly better denoising of the simulated data than the other methods. Finally, the practical application of the proposed method to hyperspectral image denoising was demonstrated.

[1] Is available through Purdue's University MultiSpec site

7. REFERENCES

[1] I. Jolliffe, *Principal component analysis*, Wiley Online Library, 2005.

[2] D.N. Lawley, "A modified method of estimation in factor analysis and some large sample results.," in *Uppsala symposium on psychological factor analysis.*. 1953, SEE/URISCA.

[3] M.E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[4] M. O. Ulfarsson and V. Solo, "Smooth principal component analysis with application to functional magnetic resonance imaging," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*, 2006, vol. 2, pp. II–993–996.

[5] M.O. Ulfarsson and V. Solo, "Sparse variable noisy PCA using l0 penalty," in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'10).*, 2010, pp. 3950–3953.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[7] T.K. Moon, "The Expectation-Maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[8] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley, 2001.

[9] J. Cardoso and D. Pham, "Optimization issues in noisy Gaussian ICA," in *Independent component analysis and blind signal separation*, pp. 41–48. Springer, 2004.

[10] I. M. Johnstone and A. Y. Lu, "On consistency and sparsity for principal components analysis in high dimensions," *Journal of the American Statistical Association*, vol. 104, no. 486, 2009.

[11] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[12] K.O. Dzhaparidze, *Parameter Estimation and Hypothesis Testing in Spectral Analysis of Stationary Time Series*, Springer Series in Statistics. Springer New York, 1986.

[13] P. Whittle, "Estimation and information in stationary time series," *Arkiv för matematik*, vol. 2, no. 5, pp. 423–434, 1953.

[14] J. Liu and P. Moulin, "Complexity-regularized image denoising," *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 841–851, 2001.

[15] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.