Independent Vector Analysis: An Extension of ICA to Multivariate Components

Taesu Kim[1,2], Torbjørn Eltoft[3], and Te-Won Lee[1]

[1] Institute for Neural Computation, UCSD, USA
{taesu, tewon}@ucsd.edu
[2] Department of BioSystems, KAIST, Korea
[3] Department of Physics, University of Tromsø, Norway
torbjorn.eltoft@phys.uit.no

Abstract. In this paper, we solve an ICA problem where both source and observation signals are multivariate, thus, vectorized signals. To derive the algorithm, we define dependence between vectors as Kullback-Leibler divergence between joint probability and the product of marginal probabilities, and propose a vector density model that has a variance dependency within a source vector. The example shows that the algorithm successfully recovers the sources and it does not cause any permutation ambiguities within the sources. Finally, we propose the frequency domain blind source separation (BSS) for convolutive mixtures as an application of IVA, which separates 6 speeches with 6 microphones in a reverberant room environment.

1 Introduction

Independent component analysis (ICA) is proposed as a method to find statistically independent sources from mixture observations by utilizing higher-order statistics [1]. In its simplest form, the ICA model assumes linear, instantaneous mixing without sensor noise, the number of sources being equal to the number of sensors, and so on. Before considering these assumptions, there is more fundamental assumption, which is that *every component is independent of the others.* Of course, it is. However, what if the sources are multivariate or vectorized signal? Let's consider some examples such as complex-valued signal, time-frequency representation of audio signal, color image signal, etc. Are the components still independent? Usually not. Elements within a source vector are sometimes correlated or sometimes uncorrelated but dependent.

In this paper, we consider an algorithm for solving the following problem.

Independent Vector Analysis (IVA)
Given observations $\mathbf{x}_i$,

$$\mathbf{x}_i = \sum_{j}^{L} \mathbf{a}_{ij} \circ \mathbf{s}_j \tag{1}$$

finding source vectors $\mathbf{s}_j$ by

J. Rosca et al. (Eds.): ICA 2006, LNCS 3889, pp. 165–172, 2006.

$$\mathbf{s}_i \approx \hat{\mathbf{s}}_i = \sum_j^M \mathbf{w}_{ij} \circ \mathbf{x}_j \tag{2}$$

where $\circ$ denotes element-wise product, and L and M is the number of sources and observations, respectively. Notation used in this paper is defined in the footnote.[1]

Assumptions
1. Elements of a source vector are mutually independent of elements of the other source vectors.
2. Within a source vector, the elements are highly dependent on the others.
3. The number of sources is less than or equal to the number of observations.

Easily, one can treat this problem as several numbers of ICA problems, because (1) can be rewritten as

$$\mathbf{x}^{(1)} = A^{(1)}\mathbf{s}^{(1)}, \qquad \mathbf{x}^{(2)} = A^{(2)}\mathbf{s}^{(2)}, \qquad \cdots, \qquad \mathbf{x}^{(D)} = A^{(D)}\mathbf{s}^{(D)} \tag{3}$$

However, once the ICA algorithm is separately applied to each element of a vector, the elements of the recovered source vectors would be randomly ordered. In this case, afterwards, it should be decided which component belongs to which source vector. It causes another clustering problem, which is not easy to solve when the number of sources is large. Instead of applying ICA separately, we tackle the problem by defining dependence between multivariate components and deriving an algorithm for the IVA problem directly.

2 Method

2.1 Objective Function

In order to separate multivariate components from multivariate observations, we need to define the objective function for multivariate random variables. Here, we define Kullback-Leibler divergence between two functions as the measure of dependence. One is an exact joint probability density function, $p(\mathbf{s}_1, \cdots, \mathbf{s}_L)$ and the other is a nonlinear function which is the product of approximated marginal probability distribution functions, $\prod_i q(\mathbf{s}_i)$.

$$\begin{aligned}\mathcal{C} &= \mathcal{KL}\left(p(\mathbf{s}_1, \cdots, \mathbf{s}_L) \,\|\, \prod_i q(\mathbf{s}_i)\right) \\ &= const. + \sum_d \log|\det A^{(d)}| - \sum_i E_{\mathbf{s}_i} \log q(\mathbf{s}_i)\end{aligned} \tag{4}$$

[1] **Notation.** We use lower-cased, bold-faced letters to denote vector variables, upper cased letters to denote matrix variables, e.g. $\mathbf{s}_i = [s_i^{(1)}, \cdots, s_i^{(D)}]^\mathsf{T}$. $\mathbf{x}_i = [x_i^{(1)}, \cdots, x_i^{(D)}]^\mathsf{T}$, and $\mathbf{a}_{ij} = [a_{ij}^{(1)}, \cdots, a_{ij}^{(D)}]^\mathsf{T}$, where $a_{ij}^{(d)}$ is the ith row, jth column element of the dth mixing matrix $A^{(d)}$.

Note that the random variables in above equations are multivariate. The interesting parts of this objective function are that each source is multivariate and it would be minimized when dependency between the source vectors is removed, but dependency between the components of each vector does not need to be removed. Therefore, the objective function preserves the inherent dependency within each source vector, although it removes dependency between the source vectors.

2.2 Learning Algorithm: A Gradient Descent Method

Now that we have defined the objective function for IVA, derivation of the learning algorithm is straightforward. Here, we are using a gradient descent method to minimize the objective function. By differentiating the objective function $\mathcal{C}$ with respect to the coefficients of unmixing matrices $w_{ij}^{(d)}$, we can derive the learning rule as follows.

$$\begin{aligned} \Delta w_{ij}^{(d)} &= -\frac{\partial \mathcal{C}}{\partial w_{ij}^{(d)}} \\ &= a_{ji}^{(d)} - E\varphi^{(d)}\left(\hat{\mathbf{s}}_i^{(1)}, \cdots, \hat{\mathbf{s}}_i^{(D)}\right)\mathbf{x}_j^{(d)} \end{aligned} \tag{5}$$

By multiplying scaling matrices, $W^{(d)^{\mathsf{T}}}W^{(d)}$, the natural gradient learning rule [2], which is well known as a fast convergence method, can be obtained as

$$\Delta w_{ij}^{(d)} = \sum_{l=1}^{L}\left(I_{il} - E\varphi^{(d)}\left(\hat{\mathbf{s}}_i^{(1)}, \cdots, \hat{\mathbf{s}}_i^{(D)}\right)\hat{\mathbf{s}}_l^{(d)}\right)w_{lj}^{(d)} \tag{6}$$

where I_{il} is one when $i = l$, otherwise zero, and a multivariate score function is given by

$$\varphi^{(d)}\left(\hat{\mathbf{s}}_i^{(1)}, \cdots, \hat{\mathbf{s}}_i^{(D)}\right) = -\frac{\partial \log q\left(\hat{\mathbf{s}}_i^{(1)}, \cdots, \hat{\mathbf{s}}_i^{(D)}\right)}{\partial \hat{s}_i^{(d)}} \tag{7}$$

3 Vector Density Model

In order to minimize the objective function, defining an optimal form of the function $q(\cdot)$ as an approximated marginal probability density function is the most critical part. Here, the function $q(\cdot)$ has to be characterized as a vector density model that has dependency within a source vector. We define a vector density model as a scale mixture of Gaussians distribution.

3.1 Scale Mixture of Gaussians Distribution

Suppose that there is a D-dimensional random variable, which is defined by

$$\mathbf{s} = \sqrt{v}\,\mathbf{z} + \mu, \tag{8}$$

where v is a scalar random variable, $\mathbf{z}$ is a D-dimensional random variable, and μ is a deterministic bias. Here, the random variable, $\mathbf{z}$, has a Gaussian distribution with mean 0 and covariance matrix Σ.

$$\mathbf{z} \sim \mathcal{N}(0, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{\mathbf{z}^{\mathsf{T}}\Sigma^{-1}\mathbf{z}}{2}\right) \tag{9}$$

Obviously, the random variable, v, is non-negative. We assume that v has a Gamma distribution, which is a commonly used distribution for non-negative random variables.

$$v \sim \mathcal{G}(\alpha, \lambda) = \frac{\lambda^{\alpha} v^{\alpha-1}}{\Gamma(\alpha)} \exp(-\lambda v), \tag{10}$$

where α and λ are the parameters of a Gamma distribution, and $\Gamma(\cdot)$ is a complete Gamma function. Then, the random variable $\mathbf{s}$ given v has a Gaussian distribution. The mean and variance of this distribution are $E\mathbf{s} = \sqrt{v}E\mathbf{z} + \mu = \mu$ and $E(\mathbf{s}-\mu)(\mathbf{s}-\mu)^{\mathsf{T}} = \sqrt{v}Ezz^{\mathsf{T}}\sqrt{v} = v\Sigma$, respectively.

$$\mathbf{s}|v \sim \mathcal{N}(\mu, v\Sigma) = \frac{1}{(2\pi v)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{(\mathbf{s}-\mu)^{\mathsf{T}}\Sigma^{-1}(\mathbf{s}-\mu)}{2v}\right) \tag{11}$$

In this model, each component of $\mathbf{s}$ is not only correlated to others, but also has variance dependency generated by v. Even though we assume the covariance matrix Σ is identity, that is, each element of $\mathbf{s}$ is uncorrelated, it is dependent on the others. We can obtain probability distribution function of variance dependent random variable $\mathbf{s}$, by integrating joint distribution of $\mathbf{s}$ and v over v.

$$p(\mathbf{s}) = \int_0^{\infty} p(\mathbf{s}|v)p(v)\mathrm{d}v \tag{12}$$

Let $\delta = \sqrt{((\mathbf{s}-\mu)^{\mathsf{T}}\Sigma^{-1}(\mathbf{s}-\mu))}$ and $\gamma = \sqrt{2\lambda}$. Now, we rearrange the joint p.d.f as a form of Inverse Gaussian distribution [3] as follows.

$$\begin{aligned}(12) = & \frac{\lambda^{\alpha}}{(2\pi)^{D/2}\Gamma(\alpha)|\Sigma|^{1/2}} \frac{(2\pi)^{1/2}}{\delta} \exp(-\gamma\delta) \\ & \times \int_0^{\infty} v^{\alpha-(D-1)/2} \underbrace{\frac{\delta}{(2\pi)^{1/2}} \exp(\gamma\delta)\, v^{-3/2} \exp\left(-\frac{1}{2}\left(\frac{\delta^2}{v} + \gamma^2 v\right)\right)}_{\text{Inverse Gaussian p.d.f.}} \mathrm{d}v \end{aligned} \tag{13}$$

Then, the integral in (13) is the $(\alpha - (D-1)/2)$-th order moment of Inverse Gaussian. Therefore, the variance dependent source p.d.f is obtained as

$$p(\mathbf{s}) = c\left((\mathbf{s}-\mu)^{\mathsf{T}}\Sigma^{-1}(\mathbf{s}-\mu)\right)^{\alpha/2-D/4} \mathcal{K}_{\alpha-D/2}\left(\sqrt{2\lambda(\mathbf{s}-\mu)^{\mathsf{T}}\Sigma^{-1}(\mathbf{s}-\mu)}\right), \tag{14}$$

where c is a normalization term and $\mathcal{K}_{\nu}(z)$ is the modified Bessel function of the second kind, which is approximated as

$$\mathcal{K}_{\nu}(z) \approx \sqrt{\frac{\pi}{2z}} e^{-z}\left(1 + \frac{4\nu^2-1}{8z} + \frac{(4\nu^2-1)(4\nu^2-9)}{2!(8z)^2} + \cdots\right) \tag{15}$$

3.2 Multivariate Score Function

So far, we have derived an algorithm and defined a vector density model. Finally in the algorithm, one can notice that the only difference between IVA and the conventional ICA is caused by the form of a score function. If we define the multivariate score function given in (7) as a single-variate score function, $\varphi^{(d)}\left(\hat{s}_i^{(d)}\right)$, which is a function of only one variable, the algorithm is converted to the same as the conventional ICA such as InfoMax algorithm. According to the density model we defined, we can obtain a form of a multivariate score function by differentiating log prior (14) with respect to each element of a source vector, because $q\left(\hat{\mathbf{s}}_i\right)$ in the objective function is an approximated probability density function of a source vector, that is, $q\left(\mathbf{s}_i\right) \approx p\left(\mathbf{s}_i\right)$. Therefore, we can obtain following form of a multivariate score function.

$$\varphi^{(k)}\left(\hat{s}_i^{(1)}, \cdots, \hat{s}_i^{(D)}\right) = \frac{\mathcal{K}_{\alpha-D/2-1}\left(\delta\right)}{\mathcal{K}_{\alpha-D/2}\left(\delta\right)} \frac{\hat{s}_i^{(d)}}{\delta} = \xi(\delta)\frac{\hat{s}_i^{(d)}}{\delta} \tag{16}$$

where $\xi(\delta) \approx 1$ for large δ. To obtain a simplified score function, we may approximate the Bessel function in (14) up to the 1st order, which results the following function.

$$\varphi^{(k)}\left(\hat{s}_i^{(1)}, \cdots, \hat{s}_i^{(D)}\right) \approx \left(\frac{D+1-2\alpha}{2\delta} + 1\right)\frac{\hat{s}_i^{(d)}}{\delta} \tag{17}$$

Although it is possible to estimate the mean vector μ and the covariance matrix Σ while the algorithm learns. We would, in this paper, fix them to zero mean and unit variance, and assume that the elements in a source vector are uncorrelated. Thus, simply $\delta = \sqrt{\sum_d \left|\hat{s}_i^{(d)}\right|^2}$. Although we propose above 2 forms of multivariate score functions, we believe that another form of a multivariate score function will be still possible by choosing a different vector density model that has different dependencies.

4 Example

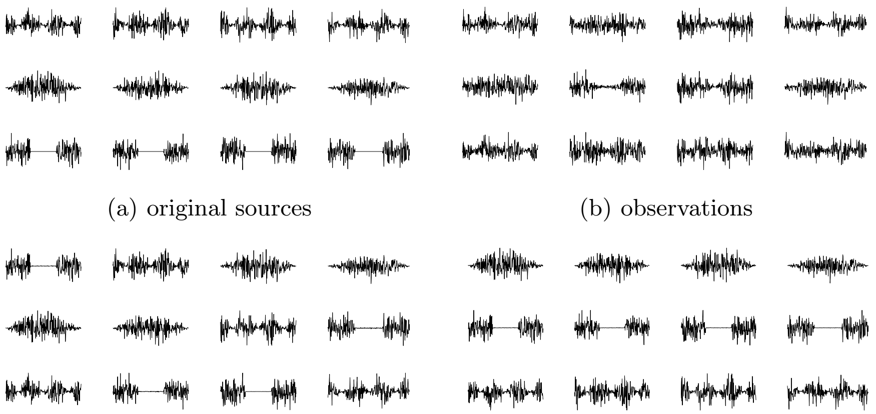
We verified our algorithm with artificially generated signals. First, we generated 3 i.i.d. Gaussian random vector signals, which were 4 dimensional vectors. Then, the same amplitude modulation was applied to the elements of each vector signal as follows.

$$\mathbf{s}_2(t) = \cos\left(2\pi t/3\right)\mathbf{z}_1(t) \tag{18}$$

$$\mathbf{s}_1(t) = \sin\left(2\pi t\right)\mathbf{z}_2(t) \tag{19}$$

$$\mathbf{s}_3(t) = \mathbf{U}\left(\sin\left(2\pi t/3\right)\right)\mathbf{z}_3(t), \tag{20}$$

where $\mathbf{z}_i$ is 4 dimensional i.i.d. Gaussian random vector, and $\mathbf{U}(\cdot)$ denotes a unit step function. Mixing matrices were randomly generated. Fig. 1 shows the original sources, observations signals, and recovered sources by both of ICA and IVA.



(a) original sources

(b) observations

(c) recovered sources by ICA

(d) recovered sources by IVA

Fig. 1. The original sources, observations, and recovered sources by ICA and IVA. Each row is corresponding to a single source vector, which is 4 dimensional in the example. In contrast to ICA, IVA does not suffer the inter-element permutation problem as well as it separates sources properly.

Each row is corresponding to a single source vector, which is 4 dimensional in the example. As shown in the figure, ICA solution disorders elements in a source vector, whereas IVA does not suffer the inter-element permutation problem as well as it separates the sources properly. Following matrices show the product of the unmixing matrix and the mixing matrix, which should be identity matrix with permutations. Those obtained by ICA was

$$W^{(1)}A^{(1)}=\begin{bmatrix} 0.031 & -0.034 & \boxed{1.557} \\ 0.012 & \boxed{1.3695} & 0.039 \\ \boxed{1.395} & -0.081 & 0.019 \end{bmatrix} \quad W^{(2)}A^{(2)}=\begin{bmatrix} \boxed{1.360} & -0.065 & 0.060 \\ 0.054 & \boxed{-1.399} & 0.013 \\ -0.005 & 0.021 & \boxed{-1.536} \end{bmatrix}$$

$$W^{(3)}A^{(3)}=\begin{bmatrix} 0.018 & \boxed{1.422} & -0.019 \\ \boxed{-1.391} & 0.058 & 0.075 \\ 0.013 & -0.054 & \boxed{-1.538} \end{bmatrix} \quad W^{(4)}A^{(4)}=\begin{bmatrix} 0.011 & \boxed{1.357} & 0.018 \\ -0.002 & 0.001 & \boxed{-1.557} \\ \boxed{-1.428} & -0.047 & 0.029 \end{bmatrix}$$

In contrast to ICA, IVA provided a well-ordered solution, which has the same permutations in a source vector as follows.

$$W^{(1)}A^{(1)}=\begin{bmatrix} -0.006 & \boxed{-2.388} & -0.048 \\ -0.011 & 0.099 & \boxed{-2.592} \\ \boxed{2.370} & -0.079 & 0.092 \end{bmatrix} \quad W^{(2)}A^{(2)}=\begin{bmatrix} 0.022 & \boxed{-2.395} & 0.036 \\ 0.015 & 0.066 & \boxed{2.610} \\ \boxed{-2.306} & 0.1270 & -0.076 \end{bmatrix}$$

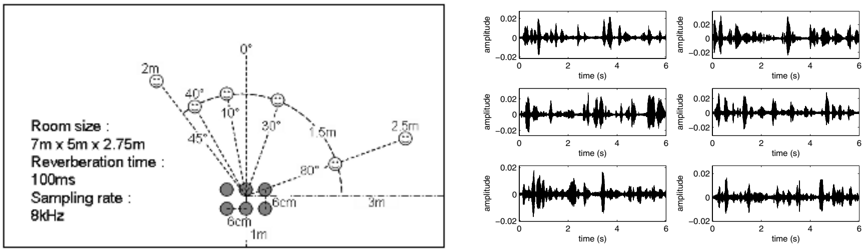
$$W^{(3)}A^{(3)}=\begin{bmatrix} -0.009 & \boxed{2.409} & 0.007 \\ -0.011 & -0.077 & \boxed{-2.609} \\ \boxed{-2.386} & 0.033 & 0.103 \end{bmatrix} \quad W^{(4)}A^{(4)}=\begin{bmatrix} -0.003 & \boxed{-2.338} & -0.009 \\ 0.027 & -0.083 & \boxed{2.587} \\ \boxed{2.421} & -0.012 & 0.013 \end{bmatrix}$$

In the above matrices, the values covered by rectangles to the other values ratio was used to calculate the performance measure. ICA and IVA result 28.5dB and 30dB, respectively.

5 Application to the Frequency Domain BSS

We applied the proposed IVA algorithm to separate convolutive mixture in the frequency domain, because the convolution is equivalent to multiplication at each frequency bin, which is the same as the model given by (1). Although one can use the conventional ICA algorithm to separate each frequency bin separately, it causes another problem which is called the frequency permutation problem. Thus, the permutations of separating matrices at each frequency should be corrected so that the separated signal in the time domain is reconstructed properly. Various algorithms have been proposed to solve the permutation problem, e.g. method that limits the filter length in the time domain [4], uses direction of arrival estimation [5], and uses inter-frequency correlation [6]. Although these algorithms perform well in some cases, they are sometimes very sensitive to the parameters or mixing conditions. However, IVA algorithm we proposed in this paper does not suffer the permutation problem at all as well as it separates sources properly.

We tested the proposed algorithm to separate 6 speeches with 6 microphones in a reverberant room environment. In this experiment, we used 8kHz sampling rate, a 2048 point FFT and a hanning window to convert time domain signal to the frequency domain. The length of window was 2048 samples and shift size was 512 samples. The condition of the room was illustrated in Fig. 2(a), and the separated sources are shown in Fig. 2(b). The improvement of signal to interference ratio (SIR) was 18dB. More intensive experiments are included in our web site [2] and another work [7].



(a) Reverberant room environment. A case of 6 mics and 6 sources

(b) Separated speeches in the time domain

Fig. 2. Room environment and the separated speeches. 2048 sample sized hanning window and 2048 FFT point was used. SIR improvement was approximately 18dB.

[2] *http://ergo.ucsd.edu/∼taesu/source_separation.html*

6 Conclusions

We have extended the conventional ICA problem to multivariate components, which we termed IVA. While ICA algorithm has a single-variate score function, IVA algorithm has a multivariate score function, which is caused by higher-order dependency within source vectors. To model a vector density, we have used scale mixture of Gaussians distribution, which models variance dependency. The results have shown that the proposed algorithm successfully recovers the sources not only in a simple example but also real world problem such as frequency domain BSS. Further, researches on various kinds of higher-order dependency models and multivariate score functions would be important to separate multivariate components.

References

1. Hyvärinen, A., Oja, E.: Independent Component Analysis. John Wiley and Sons (2002)
2. Amari, S.I., Cichocki, A., Yang, H.H.: A new learning algorithm for blind signal separation. In: Adv. Neural information Processing Systems. Volume 8. (1996)
3. Barndorff-Nielsen, O.E.: Normal inverse gaussian distributions and stochastic volatility modeling. Scand. J. Statist. **24** (1997) 1–13
4. Parra, L., Spence, C.: Convolutive blind separation of non-stationary sources. IEEE Trans. Speech Audio Processing **8** (2000) 320–327
5. Ikram, M.Z., Morgan, D.R.: A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation. In: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. (2002) 881–884
6. Murata, N., Ikeda, S., Ziehe, A.: An approach to blind source separation based on temporal structure of speech signals. Neurocomputing **41** (2001) 1–24
7. Kim, T., Attias, H., Lee, S.Y., Lee, T.W.: Frequency domain blind source separation based on variance dependencies. In: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. (2006)