

# Multi-Input Single-Output Nonlinear Blind Separation of Binary Sources

Konstantinos Diamantaras, *Senior Member, IEEE*, Gabriela Vranou, *Student Member, IEEE*,  
and Theophilos Papadimitriou, *Member, IEEE*

## Abstract

The problem of blindly separating multiple binary sources from a single nonlinear mixture is addressed through a novel clustering approach without the use of any optimization procedure. The method is based on the assumption that the source probabilities are asymmetric in which case the output probability distribution can be expressed as a linear mixture of the sources. We are then able to solve the problem by using a known linear Multiple-Input Single-Output (MISO) blind separation method. The overall procedure is very fast and, in theory, it works for any number of independent binary sources and for a wide range of nonlinear functions. In practice, the accuracy of the method depends on the estimation accuracy of the output probabilities and the cluster centers. It can be quite sensitive to noise especially as the number of sources increases or the number of data samples is reduced. However, in our experiments we have been able to demonstrate successful separation of up to four sources.

## Index Terms

Blind Source Separation, BSS, Nonlinear BSS, Underdetermined BSS

Blind Source Separation (BSS) methods aim at recovering the  $n$  unobservable sources using  $m$  mixture signals (observations). The fact that both the sources and the underlying mixing operator are assumed unknown justifies the term “*blind*”. Depending on the mixing process, BSS methods can be divided into linear (instantaneous or convolutive) and nonlinear ones. The research in the field of linear BSS has provided many powerful and well-established methods such as AMUSE [1], SOBI [2], JADE [3], FastICA [4]. For nonlinear mixing models the same is not true. Even though the nonlinear case describes a common scenario, the indeterminacies it imposes are very difficult to handle. Linear BSS suffers for the inability to recover the sources scale and order. However, scale is often determined by the application, while the order may not be important. Nonlinear BSS suffers from the inability a) to recover the scale and b) to estimate a stable and unique solution. In fact, any nonlinear transformation of the true sources forms a potential solution to the nonlinear BSS problem [5]. The recovery inconsistency has been attacked by adding further *a priori* information directly in the model or as a regularization term in the optimization processing.

In his pioneering paper, Burel proposed, in 1992 [6], a nonlinear blind source separation algorithm using two-layer perceptron. Following a classic procedure in linear BSS methods, Burel aimed at the independence of the reconstructed signals by minimizing a cost function based on the mutual information. In the same scheme Tan e.a. in [7] use a genetic algorithm to recover the sources, while Zhang e.a. [8] use a Radial Basis Function network.

Since, the research on the linear BSS problem is more advanced, than in the nonlinear scenario, many researchers tried to modify well known linear BSS methods to the nonlinear framework. Almeida in [9] proposes a generalization of the INFOMAX method, which is able to deal with nonlinear mixtures. The method creates a single multilayer network performing ICA, recovering the sources simultaneously from a single maximization procedure. Lappalainen and Honkela in [10] proposed a nonlinear ICA method based on a multi-layer perceptron network for the separation. Valpola e.a. in [11] attack the BSS problem of a nonlinear system using unsupervised Bayesian modeling. The necessary posterior pdf's of the unknown variables for the Bayesian estimation are approximated using the Variational Bayesian Learning. In [12] Babaie-Zadeh and Jutten proposed a non-parametric gradient of the mutual information which was then used for solving linear BSS problems as well as Post-nonlinear (PNL) instantaneous and convolutive BSS problems. More recently, Levin addressed the problem of blindly separating

K. Diamantaras and G. Vranou are with the Department of Informatics, Technological Education Institute of Thessaloniki, 57400, Thessaloniki, Greece, e-mail: {kdiamant,gvrano}@it.teithe.gr.

Th. Papadimitriou is with the Department of International Economic Relations & Development, Democritus University of Thrace, 69100, Komitini, Greece, e-mail: papadimi@ierd.duth.gr.

multiple continuous-time sources from their instantaneous nonlinear mixtures using the data velocity in the so called (state, state-velocity)-space [13].

A different perspective can be established from the kernel-based theory. The basic idea is to project the system into a feature space, where the nonlinear mixtures become linear, and then solve the simpler problem with an established linear BSS method. Martinez and Bray in [14], and Harmeling e.a. in [15] proposed different approaches based on the kernel theory.

This paper is an extension of our earlier conference paper [16]. Here we improve our original clustering method, we provide new insights on our approach and present new simulation results. We address the general nonlinear mixture case for binary sources in the strictest underdetermined framework assuming any number of sources and only one output signal. To our knowledge this problem has not been treated in the past. Section I describes the problem and the working assumptions, Section II proposes a novel method using a linear BSS approach based on the values of the output probability distribution, while Section IV presents simulations using artificial signals and real images. Section V concludes.

## I. PROBLEM FORMULATION AND ASSUMPTIONS

Consider  $n$  binary sequences  $s_i(k) \in \{-1, +1\}$ ,  $i = 1, \dots, n$ ,  $k = 1, \dots, N$ , mixed by a general real nonlinear function  $f$  into a single output sequence

$$x(k) = f(s_1(k), \dots, s_n(k)). \quad (1)$$

The extraction of the sources  $s_1, \dots, s_n$ , using only the output samples  $x(k)$ ,  $k = 1, \dots, N$ , is classified as a nonlinear Multi-Input Single-Output (MISO) blind source separation problem. If  $f$  is constrained, for example, to be an odd function, then one can find a solution for a limited number of sources ( $n \leq 3$ ) [17], [18]. In this paper our aim is to investigate the solution of the problem under the most general conditions for the nonlinear function  $f$  and without posing any theoretical limitations on  $n$ .

We shall start by assuming that the sources are stationary<sup>1</sup> and we shall define the probability that source  $s_i(k)$  takes the value 1

$$p_i \doteq \text{Prob}\{s_i = 1\}. \quad (2)$$

Due to stationarity,  $p_i$  is independent of  $k$  while, obviously,  $\text{Prob}\{s_i = -1\} = 1 - p_i$ , thus the distribution of  $s_i$  is completely determined by  $p_i$ . We are not aware of any solution when the sources are unbiased, i.e. if  $p_1 = p_2 = \dots = p_n = 0.5$ , and at the same time  $f$  is totally unconstrained. In this case the problem is ill-posed. On the other hand, if the sources are biased we can exploit this asymmetry to achieve separation, as shown in this paper. Binary images is a typical case of biased sources: in a general binary image the probability that a pixel is white is not equal to the probability that it is black. Our assumptions are formulated as follows:

**Assumption 1:** We have

$$\begin{aligned} p_i &\neq p_j \text{ for all } i \neq j, \text{ and} \\ p_i &\neq 1 - p_j \text{ for all } i, j \text{ (including } i = j) \end{aligned}$$

i.e. all probabilities  $p_i$  and  $(1 - p_i)$  are distinct.

**Assumption 2:** The sources are statistically independent.

**Assumption 3:** Each unique input vector  $\mathbf{s}^{(p)}$  corresponds to a unique output value  $x^{(p)} = f(\mathbf{s}^{(p)})$ .

Due to Assumption 1 we have  $p_i \neq 0.5$ , for all  $i$ . Furthermore, without loss of generality, we can assume

$$p_1 > p_2 > \dots > p_n > 0.5, \quad (3)$$

so

$$0.5 > 1 - p_n > \dots > 1 - p_2 > 1 - p_1. \quad (4)$$

<sup>1</sup>Since the source  $s_i(k)$  is a binary stochastic sequence, strict and wide-sense stationarity are equivalent.

Indeed, inequality (3) can be always forced to hold by rearranging the order and/or reversing the signs of the sources.

Assumption 2 is typical for BSS problems. It allows us, however, to write the log probability of the input vector  $\mathbf{s} = [s_1, \dots, s_n]^T$  as the sum of the log probabilities of the individual sources:

$$\log P(\mathbf{s}) = \log P(s_1) + \log P(s_2) + \dots + \log P(s_n). \quad (5)$$

Eq. (5) will be the key formula for the development of the blind separation algorithm described next.

## II. NONLINEAR MISO SEPARATION

Taking advantage of the additive nature of the log probabilities in (5) we shall attempt to transform the problem into a known linear MISO problem. Note that we can write

$$\log P(s_i) = m_i + h_i s_i. \quad (6)$$

where

$$m_i = \log([p_i(1-p_i)]^{1/2}) \quad (7)$$

$$h_i = \log([p_i / (1-p_i)]^{1/2}) \quad (8)$$

Indeed, then

$$\log P(s_i = 1) = m_i + h_i = \log(p_i)$$

and

$$\log P(s_i = -1) = m_i - h_i = \log(1-p_i).$$

Combining (5) and (6) we obtain that  $[\log P(\mathbf{s})]$  is an affine combination of the sources

$$\log P(\mathbf{s}) = \sum_{i=1}^n h_i s_i + m = \mathbf{h}^T \mathbf{s} + m \quad (9)$$

where  $\mathbf{h} = [h_1, \dots, h_n]^T$  and  $m$  is a constant offset equal to

$$m = \sum_{i=1}^n m_i.$$

Since  $\mathbf{s} = [\pm 1, \pm 1, \dots, \pm 1]^T$  is a binary vector of length  $n$ , there are  $M = 2^n$  distinct values  $\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(M)}$ , of  $\mathbf{s}$ . Clearly, the points  $\mathbf{s}^{(i)}$  are symmetric around zero, ie. for each  $\mathbf{s}^{(i)}$  there exists some  $j$  such that  $\mathbf{s}^{(j)} = -\mathbf{s}^{(i)}$ . So

$$\sum_{p=1}^M \mathbf{s}^{(p)} = \mathbf{0}. \quad (10)$$

Call  $\bar{z}_i$  their respective log probabilities:

$$\bar{z}_1 \doteq \log P(\mathbf{s}^{(1)}), \dots, \bar{z}_M \doteq \log P(\mathbf{s}^{(M)}).$$

Assumption 3 imposes a constraint on the nonlinear mixing function  $f$ :

$$f(\mathbf{s}^{(p)}) \neq f(\mathbf{s}^{(q)}), \text{ for all } p \neq q. \quad (11)$$

Although the class of functions that satisfy (11) is very large it excludes certain types of functions, for example, the even functions, since  $f(\mathbf{s}) \neq f(-\mathbf{s})$ .

At the same time, Assumption 3 gives us the opportunity to estimate the set  $\{\bar{z}_1, \dots, \bar{z}_M\}$  from the observations  $x(k)$ . Since there is an one-to-one correspondence between input and output values, we have

$$\bar{z}_p = \log P(\mathbf{s}^{(p)}) = \log P(x^{(p)})$$

In order to compute  $\log P(x^{(p)})$  we first need to cluster the output data into  $M$  distinct classes  $\mathcal{X}_1, \dots, \mathcal{X}_M$ , which correspond to the distinct output values  $x^{(1)}, \dots, x^{(M)}$ , and then we need to compute the probabilities of each class.

After we have estimated the values  $\bar{z}_p$  we can use them to estimate  $m$  by simple averaging:

$$\begin{aligned} \frac{1}{M} \sum_{p=1}^M \bar{z}_p &= \frac{1}{M} \sum_{p=1}^M \log P(\mathbf{s}^{(p)}) \\ &= \frac{1}{M} \sum_{p=1}^M m + \frac{1}{M} \mathbf{h}^T \sum_{p=1}^M \mathbf{s}^{(p)} \\ &= m \end{aligned} \quad (12)$$

The last equation is true due to (10).

Subtracting  $m$  from  $\bar{z}_p$  we obtain

$$z_p = \bar{z}_p - m = \mathbf{h}^T \mathbf{s}^{(p)}, \quad p = 1, \dots, M \quad (13)$$

The signal  $z$ , as a function of  $p$ , is now a purely linear mixture of the source vector signal  $\mathbf{s}^{(p)}$ . The problem is equivalent to a linear binary BSS problem with  $n$  inputs and 1 output. Therefore, we may employ the linear MISO BSS method for binary signals proposed in [19], [20].

The solution is feasible if there are no repeated values in the sequence  $z_1, \dots, z_M$ . So the following assumption is needed:

**Assumption 4:** The set  $\{z_1, \dots, z_M\}$  contains  $2^n$  distinct values.

This assumption implies that there is no pair  $p, q$ , ( $p \neq q$ ) such that  $\mathbf{h}^T (\mathbf{s}^{(p)} - \mathbf{s}^{(q)}) = \mathbf{h}^T \mathbf{d}^{(p,q)} = 0$  (with the obvious definition for  $\mathbf{d}^{(p,q)}$ ). Therefore,  $\mathbf{h}$  should not belong to the null space  $\mathcal{N}_{p,q}$  of  $\mathbf{d}^{(p,q)}$ . Note that due to (3) and (8) we have

$$h_1 > h_2 > \dots > h_n > 0. \quad (14)$$

Also  $\mathbf{d}^{(p,q)} = [d_1^{(p,q)}, \dots, d_n^{(p,q)}]^T$ , where  $d_i^{(p,q)} \in \{-2, 0, 2\}$ , and at least one  $d_i^{(p,q)} \neq 0$  because  $\mathbf{s}^{(p)} \neq \mathbf{s}^{(q)}$ . Let  $n_+$  and  $n_-$  be, respectively, the number of positive and negative elements of  $\mathbf{d}^{(p,q)}$ . If  $n_+ = M$  and  $n_- = 0$ , ie.  $d_i^{(p,q)} \geq 0$ , for all  $i$ , then  $\mathbf{h}^T \mathbf{d}^{(p,q)} > 0$  and thus  $\mathbf{h} \notin \mathcal{N}_{p,q}$ . A similar argument holds for  $n_+ = 0, n_- = M$ . In order to have  $\mathbf{h} \in \mathcal{N}_{p,q}$  we must have  $n_+, n_- \geq 1$ . In fact, we can formulate an even stricter condition noticing that if  $n_+ = n_- = 1$ , so, for example,  $d_i^{(p,q)} = 2$  and  $d_j^{(p,q)} = -2$ , then  $\mathbf{h}^T \mathbf{d}^{(p,q)} = 2(h_i - h_j) \neq 0$ , since  $h_i \neq h_j$ . So in order to exist an  $\mathbf{h}$  satisfying (14) and at the same time belonging to the null space of  $\mathbf{d}^{(p,q)}$  we must have

$$n_+, n_- \geq 1 \text{ and } (n_+ > 1 \text{ or } n_- > 1). \quad (15)$$

Call  $\mathcal{D}$  the set of index pairs  $(p, q)$  for which (15) holds. Then Assumption 4 implies that

$$\mathbf{h} \notin \cup_{(p,q) \in \mathcal{D}} \mathcal{N}_{p,q}. \quad (16)$$

The null spaces  $\mathcal{N}_{p,q}$  are  $(n-1)$ -dimensional. Consequently, if the vector  $\mathbf{h}$  is in ‘‘general position’’ in  $\mathbb{R}^n$  then the probability that Assumption 4 is not satisfied is 0.

Under the above assumptions, the linear problem (13) can be solved in a recursive manner [20]. The overall algorithm is summarized below

*Algorithm 1 (Nonlinear binary MISO BSS):*

**Step 1.** Cluster the observed data  $x(k)$ ,  $k = 1, \dots, N$ , into  $M = 2^n$  distinct classes  $\mathcal{X}_1, \dots, \mathcal{X}_M$ , represented by the values  $x^{(1)}, x^{(2)}, \dots, x^{(M)}$ .

**Step 2.** Estimate the log-probabilities  $\bar{z}_p = \log \text{Prob}(x \in \mathcal{X}_p)$ . Use (12), (13) to obtain  $z_p$ .

**Step 3.** Estimate the linear mixing coefficients  $h_1, \dots, h_n$  and the input vector  $\mathbf{s}^{(p)}$  corresponding to the value  $z_p$  using algorithm 2.

**Step 4.** For each  $x(k) \in \mathcal{X}_p$  estimate the input vector  $\hat{\mathbf{s}}(k)$  by  $\mathbf{s}^{(p)}$ . ■

For the sake of completeness we also present the linear binary MISO BSS algorithm used in step 3:

*Algorithm 2 (Linear binary MISO BSS):* [20]

**Step 1.** Sort the values  $z_p$  into the increasing sequence  $\hat{c}_p$

**Step 2.** Let  $\hat{h}_n = (\hat{c}_2 - \hat{c}_1)/2$ ,  $\hat{h}_{n-1} = (\hat{c}_3 - \hat{c}_1)/2$ .

**Step 3.** Compute the values  $\hat{d}c_1 = 2\hat{h}_n$ ,  $\hat{d}c_2 = 2\hat{h}_{n-1}$ ,  $\hat{d}c_3 = 2(\hat{h}_{n-1} + \hat{h}_n)$

**Step 4.** Remove the set  $\{\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_1 + \hat{d}c_3\}$  from the sequence  $\{\hat{c}_p\}$ . Set  $\hat{c}'_1 = \hat{c}_1 + \hat{h}_n + \hat{h}_{n-1}$  as the first element of a new sequence  $\{\hat{c}'_p\}$ .

**Step 5.** Repeat until all elements have been removed:

Find the smallest element  $\hat{c}_q$  of the remaining sequence  $\{\hat{c}_p\}$ ;

Remove the set  $\{\hat{c}_q, \hat{c}_q + \hat{d}c_1, \hat{c}_q + \hat{d}c_2, \hat{c}_q + \hat{d}c_3\}$  from  $\{\hat{c}_p\}$ ;

Add  $\hat{c}_q + \hat{h}_n + \hat{h}_{n-1}$  as the next element of  $\{\hat{c}'_p\}$ .

At the end, the new sequence  $\{\hat{c}'_p\}$  will be 4 times shorter than the original sequence  $\{\hat{c}_p\}$ .

**Step 6.** Recursively repeat the algorithm for the new sequence  $\{\hat{c}'_p\}$  and for a new  $n' = n - 2$  to obtain  $\hat{h}_{n'} = \hat{h}_{n-2}$ ,  $\hat{h}_{n'-1} = \hat{h}_{n-3}$ . Eventually we will get  $n' = 2$  or  $n' = 3$  and

$\hat{h}_2 = (\hat{c}'_2 - \hat{c}'_1)/2$ ,  $\hat{h}_1 = (\hat{c}'_3 - \hat{c}'_1)/2$ , for  $n' = 2$ , or

$\hat{h}_3 = (\hat{c}'_2 - \hat{c}'_1)/2$ ,  $\hat{h}_2 = (\hat{c}'_3 - \hat{c}'_1)/2$ ,  $\hat{h}_1 = -(\hat{c}'_2 + \hat{c}'_3)/2$ , for  $n' = 3$ .

**Step 7.** Estimate source vector corresponding to  $z_p$  as

$$\hat{\mathbf{s}}^{(p)} = \arg \min_{\mathbf{s}} \|z_p - \hat{\mathbf{h}}^T \mathbf{s}\|^2. \quad \blacksquare$$

#### A. Remarks

There is a number of issues that need to be addressed before the practical implementation of the proposed algorithm. First, we need to select an appropriate clustering method for grouping the observed samples into  $2^n$  classes. In the noise-free case each class  $\mathcal{X}_p$  contains just one point  $x^{(p)}$ , so the clustering task boils down to finding the  $2^n$  unique values in the set  $\{x(1), x(2), \dots, x(N)\}$ . The selection of clustering method is most critical in the presence of noise. As we shall see in Section III, due to the center-oriented and mixture of distributions nature of the clusters the EM algorithm would be the most suitable method for the particular problem.

Second, a factor that affects the performance is the spread of the source probabilities  $p_i$ . Due to the biased sources assumption (Assumption 1) the probabilities of the different output classes are not equal. For example, assume that there are  $n = 3$  sources with  $p_1 = 0.52$ ,  $p_2 = 0.58$ , and  $p_3 = 0.61$ . Then the different classes and their probabilities are:

Class	Center	$Prob(\mathcal{X}_p)$
$\mathcal{X}_1$	$x^{(1)} = f([-1, -1, -1])$	0.078624
$\mathcal{X}_2$	$x^{(2)} = f([-1, -1, 1])$	0.122976
$\mathcal{X}_3$	$x^{(3)} = f([-1, 1, -1])$	0.108576
$\mathcal{X}_4$	$x^{(4)} = f([-1, 1, 1])$	0.169824
$\mathcal{X}_5$	$x^{(5)} = f([1, -1, -1])$	0.085176
$\mathcal{X}_6$	$x^{(6)} = f([1, -1, 1])$	0.133224
$\mathcal{X}_7$	$x^{(7)} = f([1, 1, -1])$	0.117624
$\mathcal{X}_8$	$x^{(8)} = f([1, 1, 1])$	0.183976

In this (conservative) example, the most probable class ( $\mathcal{X}_8$ ) has probability 0.183976 which is almost two and a half times higher than the probability 0.078624 of the least probable class ( $\mathcal{X}_1$ ). This discrepancy will be accentuated if the spread of the initial probabilities  $p_i$  is higher or if the number of sources is increased. If the least probable class is much less likely than the most probable one, the clustering algorithm can have difficulty locating it, when noise is present. This effect shall be demonstrated in our experiments. However, we should note that the increase of the spread of the  $p_i$ 's has a positive effect in the blind separation algorithm since the values  $z_p$  are better separated.

A third parameter that affects the performance of the algorithm is the dataset size  $N$ . The magnitude of  $N$  should be big enough as to have an accurate estimate of the output class probabilities  $Prob(\mathcal{X}_p)$  and consequently of their logarithms  $\bar{z}_p$ .

Finally, the number of sources has a fundamental effect on the performance since the number of clusters increases exponentially with  $n$ , and so cluster estimation becomes more sensitive to noise. This is to be expected however, since the problem is “difficult” having only one output (observed) signal and multiple input (source) signals.

### III. ESTIMATING CLUSTER CENTERS AND PROBABILITIES

In the presence of zero-mean Gaussian observation noise  $e$ , the output sequence becomes:

$$x(k) = f(\mathbf{s}(k)) + e(k) \quad (17)$$

The distribution of  $x$  is a mixture of Gaussians centered at the points  $x^{(i)}$ :

$$p_x(x) = \sum_{i=1}^M P(x^{(i)}) g_\sigma(x - x^{(i)}) \quad (18)$$

where  $g_\sigma(\cdot)$  is the Gaussian kernel

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x)^2}{2\sigma^2}\right\} \quad (19)$$

and  $\sigma^2$  is the noise variance. Therefore, by estimating the centers  $x^{(i)}$  and amplitudes  $P(x^{(i)})$  of the Gaussian components, we have all the elements required to run Algorithm 1. The EM algorithm [22] has been suggested in [16] as an appropriate method for this estimation. However, the algorithm often fails to estimate the parameters even with modest SNR (e.g.  $\leq 20$  dB). The difficulty is exacerbated when small Gaussian components are close to large ones. Other potential methods include the k-means algorithm [23] and the mean-shift algorithm [21]. However, both methods demonstrate poor performance especially when different size clusters are in close proximity, a scenario which frequently occurs in our problem.

In order to improve our ability to locate the peaks of the distribution we employ sharpening using the derivatives of the pdf. In particular, we start by computing an estimate  $\hat{p}_x$  of the distribution  $p_x$  from the data  $x(k)$  using the Parzen window method using the Gaussian window and some small width  $w$ :

$$\hat{p}_x(x) = \frac{1}{N} \sum_{k=1}^N g_w(x - x(k)) \quad (20)$$

Then, we estimate the second derivative  $p_x''(x) = d^2 p_x / dx^2$ :

$$p_x''(x) = \sum_{i=1}^M P(x^{(i)}) g_\sigma^{(2)}(x - x^{(i)}) \quad (21)$$

$$g_\sigma^{(2)}(u) = \frac{1}{\sigma^3\sqrt{2\pi}} \left[ \frac{u^2}{\sigma^2} - 1 \right] \exp\left\{-\frac{u^2}{2\sigma^2}\right\} \quad (22)$$

The function  $g_\sigma^{(2)}(x - x^{(i)})$  has a sharp negative peak with value  $-1/(\sigma^3\sqrt{2\pi})$  at  $x = x^{(i)}$ . Therefore, the locations of the negative peaks of  $p_x''(x)$  are good estimates of  $x^{(i)}$ . Of course,  $p_x''$  is not readily available. Instead, we work with the estimate  $\hat{p}_x$  to obtain

$$\hat{p}_x''(x) = (\hat{p}(x + \delta) - 2\hat{p}(x) + \hat{p}(x - \delta))/\delta^2 \quad (23)$$

where  $\delta$  is a very small positive value. Provided that the estimate  $\hat{p}_x$  is smooth enough so that there are no spikes due to noise, then we can obtain good estimates of the cluster centers. The smoothness of  $\hat{p}_x$  is controlled dynamically by adjusting the Parzen window width  $w$ , until exactly  $2^n$  negative peaks are found.

Next we shall demonstrate the effect of the other three parameters discussed in Section II-A, namely (a) the spread of the source probabilities  $p_i$  (b) the dataset size  $N$  and (c) the number of sources  $n$ .

## IV. SIMULATIONS

Two sets of experiments have been conducted: in the first set the sources were artificially generated 1-D binary sequences while in the second set the sources were binary images. The mixing model for both sets was Eq. (17). In the first set we varied the following parameters: (a) the number of sources from  $n$ , (b) the mixing function  $f$ , (c) the Signal to Noise Ratio (SNR), and (d) the number of data samples  $N$ . The mixing function has the general form:

for  $n = 2$ ,

$$f(s_1, s_2) = a_0 + a_1s_1 + a_2s_2 + a_3s_1s_2 \quad (24)$$

for  $n = 3$ ,

$$f(s_1, s_2, s_3) = a_0 + a_1s_1 + a_2s_2 + a_3s_3 + a_4s_1s_2 + a_5s_1s_3 + a_6s_2s_3 + a_7s_1s_2s_3 \quad (25)$$

for  $n = 4$ ,

$$f(s_1, s_2, s_3, s_4) = a_0 + a_1s_1 + a_2s_2 + a_3s_3 + a_4s_4 + a_5s_1s_2 + a_6s_1s_3 + a_7s_1s_4 + a_8s_2s_3 + a_9s_2s_4 + a_{10}s_3s_4 + a_{11}s_1s_2s_3 + a_{12}s_1s_2s_4 + a_{13}s_1s_3s_4 + a_{14}s_2s_3s_4 + a_{15}s_1s_2s_3s_4 \quad (26)$$

etc.

We present the results from three experiments with the following setup:

	Mixing parameters: $a_i$ / Source probabilities: $p_i$
$n = 2$ :	0, 0.8855, -0.2634, 0.1404 $p_1 = 0.62, p_2 = 0.55$
$n = 3$ :	0, -0.2370, -0.5216, 0.9854, 1.2145, -0.3449, 0.1728, -0.1639 $p_1 = 0.62, p_2 = 0.58, p_3 = 0.53$
$n = 4$ :	0, 1.0049, -0.0621, 0.0992, -0.8508, 1.4947, 1.0521, -0.6112, 0.1241, -1.3060, 0.2194, -0.3608, -0.2069, -0.2870, 1.0089, -0.1375 $p_1 = 0.70, p_2 = 0.64, p_3 = 0.58, p_4 = 0.53$

Note that in all the experiments for simplicity we have  $a_0 = 0$ , without lack of generality, since any non-negative value  $a_0 = c$  simply shifts the clusters by a constant amount  $c$  without further effect on the algorithm. The performance measure is the average bit error rate (BER) between source and reconstructed signals after 100 Monte Carlo experiments.

As shown in Figures 1–3 the results of the algorithm depend heavily on the number of sources  $n$  and the SNR level and it also depends on the dataset size  $N$ . For  $n = 2$  (Fig. 1) the average BER is reasonably low for SNR=20 dB and reduces quickly to zero as SNR increases to 30 dB. Practically the same performance is achieved for  $N = 5,000$  as for  $N = 100,000$ . For  $n = 3$  (Fig. 2), we see that a large SNR ( $\approx 30$  dB) is required for obtaining reasonably low BER. Also the performance improves with  $N$  approaching 50,000 samples. For  $n = 4$  (Fig. 3) the noise tolerance is even lower and reasonably low BER's are achieved for very high SNR ( $\approx 40$  dB) and again performance improves with the increase of  $N$ . In order to demonstrate the effect of the number of sources  $n$  on the performance, Fig. 4 shows the change in the average BER as a function of  $N$  keeping the SNR level constant at 30 dB. Clearly, there is dramatic improvement on the performance as  $n$  decreases, attributed to the improved identification of the pdf peaks. Figure 5 shows the average MATLAB execution time of the algorithm as a function of the dataset size  $N$ , the number of sources  $n$  and the SNR level, running on a 2.3 GHz, Core i5 (4 core) PC, with 6GB RAM. We see that execution time scales linearly with respect to  $N$  and improves considerably on very high

SNR, mainly because the clustering problem becomes easier to solve. Moreover, for  $n = 2$  the execution times are less than 10 sec even for  $N = 100,000$  although it scales quickly with  $n$ . As an example, we report the execution times for SNR=30dB and  $N = 50,000$  to be 3.1 sec ( $n = 2$ ), 9.6 sec ( $n = 3$ ), and 25.2 sec ( $n = 4$ ).

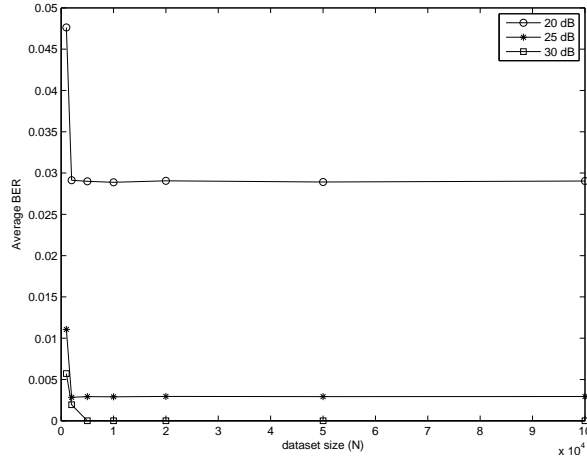


Fig. 1. Average Bit Error Rate (BER) for  $n = 2$  sources accumulated over 100 Monte Carlo simulations. Plotted as a function of the data-set size.

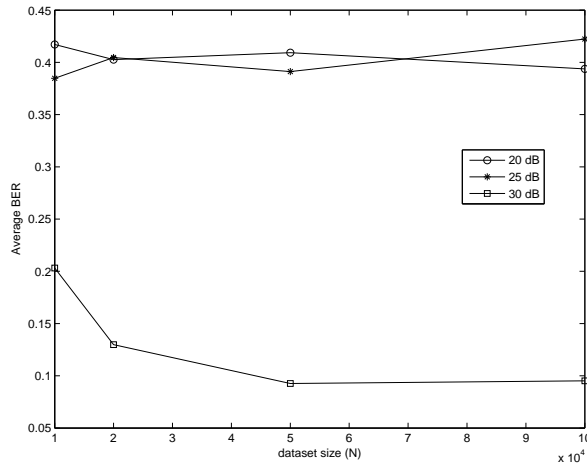


Fig. 2. Average Bit Error Rate (BER) for  $n = 3$  sources after 100 Monte Carlo simulations.

In the second set of experiments three standard test images (“blobs”, “mandrill” and “boat”) were binarized to  $-1/1$  and scaled to size  $256 \times 256$  from their original size  $512 \times 512$  (see Fig. 6). The “blobs” image was already binary. Binarization of the other two images was done by thresholding the luminance component with thresholds  $\theta = 125$  (“mandrill”) and  $\theta = 148$  (“boat”). Scaling for all images was done by simple subsampling by a factor 2. For the binarized images the probabilities were  $p_1 = 0.5501$ ,  $p_2 = 0.6091$ ,  $p_3 = 0.8303$ . Each image was transformed into an 1-D signal by stacking the columns on top of each other to form a long vector. The nonlinear mixing function has the general form of Eq. (25) where the mixing parameters  $a_i$  were randomly generated and then scaled so that the final mixture is in the range  $0 - 255$  (for viewing purposes only). We ran 100 Monte Carlo experiments, with the same source images but with different mixing parameters. In each case, the mixed image was generated by applying the quantization  $Q(\cdot)$  on  $f(\cdot)$  (rounding it to closest integer from below). Thus a small amount of quantization noise  $e_Q$  was introduced:

$$\begin{aligned}
 x(k) &= Q(f(s_1(k), s_2(k), s_3(k))) \\
 &= f(s_1(k), s_2(k), s_3(k)) + e_Q(k)
 \end{aligned}$$



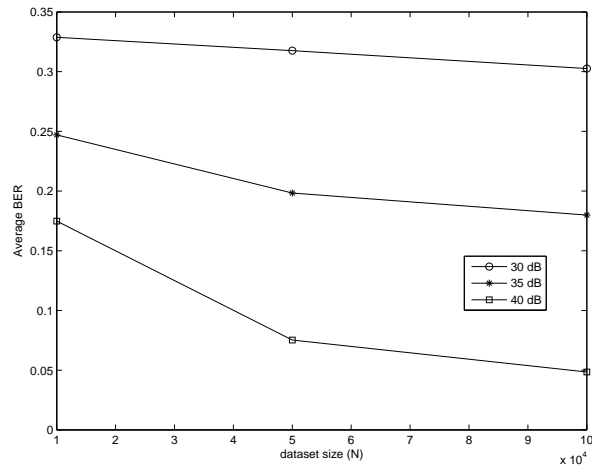


Fig. 3. Average Bit Error Rate (BER) for  $n = 4$  sources after 100 Monte Carlo simulations.

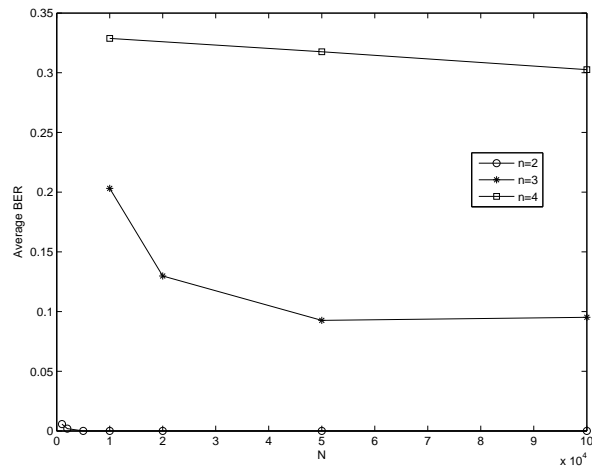


Fig. 4. Comparison of the Average Bit Error Rate (BER) as a function of the dataset size  $N$  for different number of inputs  $n$  and for the same SNR level at 30dB.

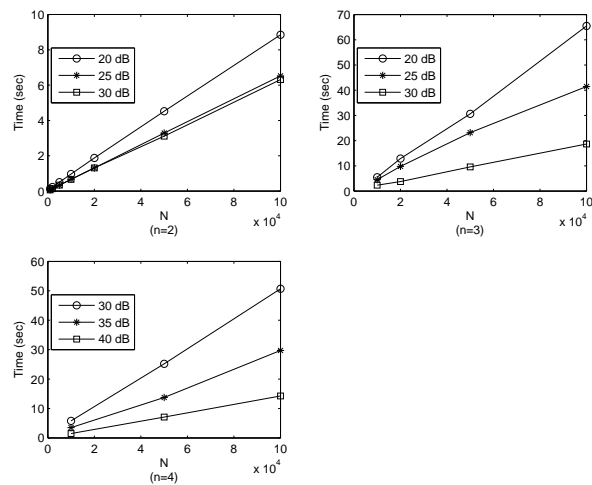


Fig. 5. Average execution time (sec) as a function of the dataset size  $N$  for different number of inputs  $n$  and SNR levels. Platform: 2.3 GHz, Core i5 Win7 PC, 6GB RAM, running MATLAB.

All 100 experiments were successful achieving zero bit error rate for the reconstructed images, i.e. we obtained perfect reconstruction although some sources were recovered with reverse sign (see an example mixture in Fig. 6d)

and the recovered sources in Fig. 6e, f, g).

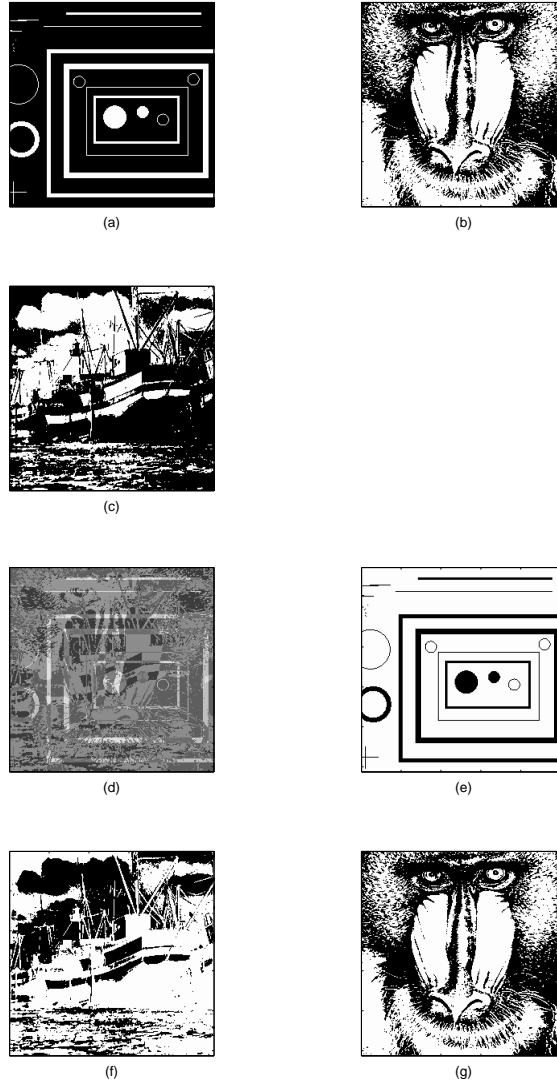


Fig. 6. Separating binary images. [Top two rows] source images: (a) “blobs”, (b) “mandrill” and (c) “boat”. [Bottom two rows]: (d) nonlinear mixture and reconstructed images (e), (f), (g).

On the other hand, the method may not work for all types of images. For example, if the sources are not independent –thus violating Assumption 2– then the separating results may contain spurious signals (see Fig. 7). As a measure of independence between two source signals  $a$  and  $b$  we use the pairwise Mutual Information (in bits) computed as

$$I_{a,b} = \sum_a \sum_b p(a,b) \log \left( \frac{p(a,b)}{p(a)p(b)} \right).$$

The pairwise mutual information between the three source images is:

$$I_{1,2} = 0,0067, I_{1,3} = 9.12 \cdot 10^{-4}, I_{2,3} = 4.17 \cdot 10^{-6}.$$

Clearly, the first pair of images (“girl” and “mandrill”) have significantly higher mutual information  $I_{1,2}$  compared to the mutual information  $I_{1,3}$ ,  $I_{2,3}$ , of those images against the third image (“boat”). We have also run 100 Monte Carlo experiments in this case with randomly varying mixing parameters. Consequently, we see from the experiments that the boat can be easily separated from the other two images. In contrast, the mutual information between the image pairs for the first experiment were:

$$I_{1,2} = 2.58 \cdot 10^{-4}, I_{1,3} = 3.44 \cdot 10^{-5}, I_{2,3} = 4.17 \cdot 10^{-6}.$$

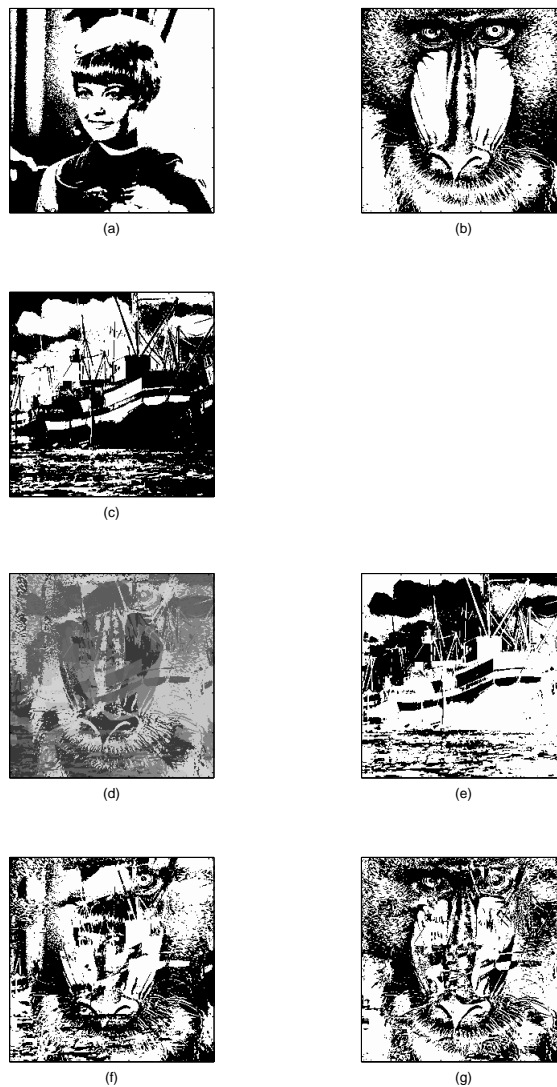


Fig. 7. Failing to separate three binary images due to lack of independence. [Top two rows] source images: (a) “girl”, (b) “mandrill” and (c) “boat”. [Bottom two rows]: (d) nonlinear mixture and reconstructed images (e), (f), (g).

## V. CONCLUSION

In this paper we presented a novel method for the blind separation of many binary sources from a single nonlinear mixture. The method is based on clustering the observed data and it is very fast since it does not use any kind of –typically costly– iterative optimization procedure. It also works for a wide range of nonlinear functions. Our key assumption is that the sources are biased and that their probabilities are distinct. We have shown that under our assumptions, the output probabilities can be written as linear combinations of the sources, therefore, the problem can be transformed into a multi-input single-output linear binary BSS problem. We have thus been able to directly use our earlier results to solve the problem. In theory the method is able to separate any number of sources given perfect estimates of the output probabilities and the output cluster center positions. Our simulations show that in practice, the performance depends heavily on the number of sources, the number of data samples, the SNR, and the choice of clustering method. We have been able to successfully separate two sources under mild SNR conditions and medium data set sizes. As the number of sources increased up to  $n = 4$ , we were still able to achieve separation although an order of magnitude was required in the increase of the SNR level and the number of data samples. We have also demonstrated the possible application to image separation while at the same time marking the limitations of the method, especially as the sources may fail the independence assumption which is essential to most BSS problems.

## REFERENCES

- [1] L. Tong, R. Liu, V.C. Soon, and Y.F. Huang, "Indeterminacy and identifiability of blind identification," *IEEE Trans. Circuits Syst. I*, vol. 38, no. 5, pp. 499–509, 1991.
- [2] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A Blind Source Separation Technique Using Second-Order Statistics," *IEEE Trans. Signal Processing*, vol. 45, no. 2, pp. 434–444, Feb. 1997.
- [3] J.-F. Cardoso and A. Souloumiac, "Blind Beamforming for non Gaussian Signals," *IEE Proceedings-F*, vol. 140, no. 6, pp. 362–370, Dec 1993.
- [4] A. Hyvärinen, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis," *IEEE Trans. on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.
- [5] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Networks*, vol. 12, no. 3, pp. 429–439, 1999.
- [6] G. Burel, "Blind Separation of Sources: A nonlinear neural algorithm," *Neural Networks*, vol. 5, no. 6, pp. 937–947, 1992.
- [7] Y. Tan, and J. Wang, "Nonlinear Blind Source Separation Using Higher Order Statistics and a Genetic Algorithm," *IEEE Trans. Evolutionary Computation*, vol. 5, no. 6, pp. 600–612, Dec. 2001.
- [8] N. Zhang, X. Zhang, J. Lu, and T. Yahagi, "An Approach for Nonlinear Blind Source Separation of Signals with Noise Using Neural Networks and Higher-Order Cumulants," in *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS-2005)*, vol. 6, pp. 5726–5729, Kobe, Japan, May, 2005.
- [9] L.B. Almeida, "MISEP - Linear and Nonlinear ICA Based on Mutual Information," *The journal of Machine Learning Research*, vol. 4, pp. 1297–1318, 2003.
- [10] H. Lappalainen and A. Honkela, "Bayesian Nonlinear Independent Component Analysis by Multi-Layer Perceptrons," in *Adv. In Ind. Comp. Anal.*, ed. By Mark Girolami, pp. 93–121, Springer, 2000.
- [11] H. Valpola, E. Oja, A. Ilin, A. Honkela, and J. Karhunen, "Nonlinear blind source separation by variational Bayesian learning," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 86, pp. 532–541, 2003.
- [12] Massoud Babaie-Zadeh and Christian Jutten, "A general approach for mutual information minimization and its application to blind source separation," Vol. 85, Issue 5, pp. 975-995, May 2005
- [13] David N. Levin, "Performing Nonlinear Blind Source Separation With Signal Invariants," *IEEE Transactions on Signal Processing*, Vol 58, No. 4, pp. 2131–2140, 2010
- [14] D. Martinez and A. Bray "Nonlinear Blind Source Separation Using Kernels," *IEEE Trans. on Neural Networks*, vol. 14, no. 1, pp. 228–235, Jan, 2003.
- [15] S. Harmeling, A. Ziehe, and M. Kawanabe, "Kernel-Based Nonlinear Blind Source Separation," *Neurocomputing*, Vol. 15, pp. 1089–1124, 2003.
- [16] K. Diamantaras, Th. Papadimitriou and G. Vranou, "Blind separation of multiple binary sources from one nonlinear mixture, in Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP-2011), pp. 2108-2111, Prague, Czechia, May 2011.
- [17] K. Diamantaras, and Th. Papadimitriou, "Blind Separation of Three Binary Sources From One Nonlinear Mixture", in Proc. 2010 Int. Workshop on Machine Learning for Signal Processing, MLSP-2010, Kittila, Finland, August 29-Sept. 1, 2010.
- [18] K. Diamantaras, and Th. Papadimitriou, "Separating two Binary Sources from a Single Nonlinear Mixture", in Proc. Int. Conf. Acoustics Speech Signal Processing (ICASSP 2010), Dallas, TX, March 14-19, 2010.
- [19] K. Diamantaras, "Blind Separation of Multiple Binary Sources using a Single Linear Mixture," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP-2000)*, vol. V, pp. 2889–2892, Istanbul, Turkey, June 2000.
- [20] K. I. Diamantaras, "A Clustering Approach for the Blind Separation of Multiple Finite Alphabet Sequences from a Single Linear Mixture," *Signal Processing*, Vol. 86, Issue 4, pp. 877–891, Elsevier, April 2006.
- [21] K. Fukunaga and L. D. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Trans. Information Theory*, vol. 21, no. 1, pp. 32-40, 1975
- [22] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
- [23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley, NY, 2001.