BLIND SEPARATION OF MULTIPLE BINARY SOURCES FROM ONE NONLINEAR MIXTURE

Konstantinos Diamantaras\textsuperscript{a}, Theophilos Papadimitriou\textsuperscript{b}, Gabriela Vranou\textsuperscript{a}

\textsuperscript{a}Technological Education Institute of Thessaloniki
Department of Informatics
Sindos 57400, Greece
\{kdiamant.gvranou\}@it.teithe.gr

\textsuperscript{b}Democritus University of Thrace
Dept. of Int. Econ. Relat. & Devel.
Komotini 69100, Greece
papadimi@ierd.duth.gr

ABSTRACT

We propose a new method for the blind separation of multiple binary signals from a single general nonlinear mixture. In addition to the usual independence assumption on the input signals our key hypothesis is the asymmetry of the source probabilities. This condition allows us to express the output probability distribution as a linear mixture of the sources. We then proceed to solve the problem using known linear BSS methods for the binary underdetermined case. The method is based on clustering avoiding costly iterative optimization. Our simulations demonstrate successful separation for up to four sources. The problem however grows exponentially with the number of sources \( n \), and the dataset size required for accurate estimation may become prohibitively large for large \( n \).

Index Terms— Blind Source Separation, BSS, Nonlinear BSS, Underdetermined BSS

1. INTRODUCTION

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 [4], FastICA [13]. 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 [15]. The recovery inconsistency has been attacked by adding further \textit{a priori} information directly in the model or as a regularization term in the optimization processing.

In his pioneering paper, Burel proposed, in 1992 [16], 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 [3] use a genetic algorithm to recover the sources, while Zhang e.a. 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 [17] 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 [18] proposed a nonlinear ICA based on a multi-layer perceptron network for the separation. Valpola e.a. in [19] 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.

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 [20] proposed different approaches based on the kernel theory.

In this paper we address the general nonlinear mixture case for binary sources in 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 2 describes the problem and the working assumptions, Section 3 proposes a novel method using a linear BSS approach based on the values of the output probability distribution, while Section 4 presents simulations using artificial signals and real images. Section 5 concludes.

2. PROBLEM FORMULATION AND ASSUMPTIONS

Consider a general real nonlinear mixture

\[ x(k) = f(s_1(k), \ldots, s_n(k)) \]  

(1)

of \( n \) binary sources \( s_1, \ldots, s_n \in \{-1, +1\} \). Our goal is to separate the sources using solely the output signal \( x \) and to investigate a suitable set of conditions under which this is possible. Eq. (1) represents a generic instantaneous nonlinear multi-input single-output (MISO) problem format. The distribution of each source \( s_i \) is completely determined by the probability \( p_i = \text{Prob}[s_i = 1] \). The case of unbiased sources \( p_i = 0.5 \) is intractable, in general, unless some strict constraints are imposed, such as, for example, that \( f \) is an odd function and the number of sources is no more than three [5, 6]. In some cases, however, the sources are biased, for instance when the inputs are binary images where the number of black and white pixels are not necessarily equal. Such cases are of interest here. We specifically make the following assumption:
Assumption 1: The sources are such that all probabilities $p_i$ and $(1 - p_i)$ are distinct, i.e.: $p_i \neq p_j$ for all $i \neq j$, and $p_i \neq 1 - p_j$ for all $i, j$ (including $i = j$).

Assuming that $f$ is analytic, its Taylor expansion is
\[
f(s_1, \ldots, s_n) = f(0, \cdots, 0) + \sum_{i=1}^{n} \frac{\partial f}{\partial s_i} s_i + \sum_{p=2}^{\infty} \sum_{a_1 + \cdots + a_p = p} \frac{\partial^p f}{\partial a_1 a_{2} \cdots a_p} s_1^{a_1} \cdots s_n^{a_n} \frac{1}{a_1! \cdots a_n!} \tag{2}
\]

For any binary number $b = \cdots b_1 b_2$, if $b_\alpha$ is even and $b_{\alpha+1}$ is odd.

The coefficients $a_i$ are real but otherwise unknown.

Without loss of generality assume that
\[
p_1 > p_2 > \cdots > p_n > 0.5,
\]
so
\[
0.5 > 1 - p_n > \cdots > 1 - p_2 > 1 - p_1.
\]  

Indeed, inequality (6) can be always forced to hold by rearranging the order and/or reversing the signs of the sources.

We also use the following typical BSS assumption:

Assumption 2: The sources are statistically independent.

The above assumption allows us to write the log probability of the vector $s = [s_1, \ldots, s_n]^T$ as
\[
\log P(s) = \log P(s_1) + \log P(s_2) + \cdots + \log P(s_n).
\]  

3. PROPOSED METHOD

Our method will be based on the analysis of the log probability (8). We start by observing that $\log P(s_i = 1) = \log(p_i)$ while $\log P(s_i = -1) = \log(1 - p_i)$. If we define
\[
m_i = \log([p_i(1 - p_i)]^{1/2}) \quad \text{and} \quad h_i = \log([p_i/(1 - p_i)]^{1/2}),
\]
then clearly,
\[
\log P(s_i) = m_i + h_i s_i.
\]  

Note that due to (6) and (10) we have
\[
h_1 > h_2 > \cdots > h_n > 0.
\]

Call $\tilde{z}$ the log probability of the source vector:
\[
\tilde{z}(s) = \log P(s)
\]

From (8) and (11) it follows that $\tilde{z}$ is a linear combination of the sources
\[
\tilde{z}(s) = m + \sum_{i=1}^{n} h_i s_i
\]
including a bias term $m = \sum_{i=1}^{n} m_i$. There are $M = 2^n$ distinct values $s^{(1)}, \ldots, s^{(n)}$, of $s$ with probabilities $\tilde{z}(s^{(1)}) = \tilde{z}_1, \ldots, \tilde{z}(s^{(n)}) = \tilde{z}_n$. The values $\tilde{z}_i$ can be estimated from the data and then used to estimate $m$ by simple averaging:
\[
m = \frac{1}{M} \sum_{i=1}^{M} \tilde{z}_i.
\]

Subtracting $m$ from $\tilde{z}$ we obtain the centered log probability
\[
\tilde{z}(s) = \tilde{z}(s) - m = \sum_{i=1}^{n} h_i s_i
\]
which is now a purely linear mixture of the sources albeit, with unknown mixing parameters $h_i$. Let
\[
\zeta_i = \tilde{z}_i - m
\]
be the values taken by $z$. Now the problem has been transformed into the following task:

- Find the source values $s_i(k)$ from $z(s)$ in (16).

The problem is equivalent to a linear binary BSS problem with $n$ inputs and 1 output; this problem has been treated in [8, 7]. A key assumption for the feasibility of the solution is that $z$ takes on $2^n$ distinct values:

Assumption 3: There are no repeated values in the set $\{\zeta_1, \ldots, \zeta_M\}$. There is only a finite set $H \subset \mathbb{R}^n$ for which Assumption 3 fails iff $h = [h_1, \ldots, h_n]^T \in H$. The assumption implies that the mixing parameters $h_i$ are in “general position”, i.e. $h \in \mathbb{R}^n - H$. Equivalently, since $h_i$ is generated by $p_i$ according to (10) the probabilities $p_i$ should be in general position.

Under these assumptions, the problem is solved in a recursive manner [7]. The algorithm is summarized below

Algorithm 1 (Linear binary BSS) [7]

Step 1. Sort the values $\zeta_i$ into the increasing sequence $\hat{\zeta}_i$.

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 $dc_1 = 2h_{n}$, $dc_2 = 2h_{n-1}$, $dc_3 = 2(\hat{h}_{n-1} + \hat{h}_n)$.

Step 4. Remove the set $\{\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_1 + dc_1\}$ from the sequence $\{\hat{c}_i\}$. Set $\hat{c}_1' = \hat{c}_1 + \hat{h}_n + \hat{h}_{n-1}$ as the first element of a new sequence $\{\hat{c}_i'\}$.

Step 5. Repeat until all elements have been removed:

Find the smallest element $\hat{c}_i$ of the remaining sequence $\{\hat{c}_i\}$.
Remove the set $\{\hat{c}_i, \hat{c}_i + dc_1, \hat{c}_i + dc_2, \hat{c}_i + dc_3\}$ from $\{\hat{c}_i\}$.
Add $\hat{c}_i + \hat{h}_n + \hat{h}_{n-1}$ as the next element of $\{\hat{c}_i'\}$.

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

Step 6. Recursively repeat the algorithm for the new sequence $\{\hat{c}_i'\}$ 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 $h_2 = (\hat{c}_2 - \hat{c}_1)/2$, $h_1 = (\hat{c}_3 - \hat{c}_1)/2$, for $n' = 2$, or $h_3 = (\hat{c}_2 - \hat{c}_1)/2$, $h_2 = (\hat{c}_3 - \hat{c}_1)/2$, $h_1 = -(\hat{c}_2 + \hat{c}_3)/2$, for $n' = 3$.

Step 7. Estimate source vector corresponding to $\zeta_i$
\[
\hat{s}^{(1)} = \arg \min_s \|\zeta_i - \hat{h}_s^T s\|^2.
\]
Based on algorithm 1 the overall nonlinear BSS algorithm is stated below:

Algorithm 2 (Nonlinear binary BSS)

Step 1. Collect the data \( x(k), \ k = 1, \ldots, K \), which should be clustered around \( M \) distinct values \( \chi_1, \chi_2, \ldots, \chi_M \).

Step 2. Estimate the log-probabilities \( \zeta_i = \log \text{Prob}(\chi_i) \). Use (15), (17) to obtain \( \zeta_i \).

Step 3. Estimate the linear mixing coefficients \( h_1, \ldots, h_n \) and the input vector \( s^{(1)} \) corresponding to the value \( \zeta_i \) and to the cluster \( \chi_i \) using algorithm 1

Step 4. For each \( x(k) \) belonging to cluster \( \chi_i \), estimate the input vector \( s(k) \) by \( s^{(1)} \)

3.1. Discussion

In the absence of noise, the performance of the proposed method is determined by the following factors

(a) the dataset size, \( N \), which affects the accuracy of the estimates of the input probabilities \( p_i \);

(b) the dispersion of the input probabilities which relates to how well assumption 1 is satisfied. The dispersion can be measured by the index \( \delta = \min_{i,j} \left| p_i' - p_j' \right| \) where \( p_i' \in \{0.5, p_1, \ldots, p_n\} \);

(c) the number of sources, \( n \), since the number of clusters increases exponentially with it. Increasing \( n \) implies (i) fewer samples per cluster (for a given \( N \)) and therefore poorer probability estimates and (ii) more crowded clusters making their estimation more sensitive to noise.

Due to lack of space proper sensitivity analysis of our method (taking into account all the above factors) is not presented here but it will appear in a future full paper. Images appear to be good candidate sources for our approach because a typical image has a large dataset size (\( N \) is in the order of \( 10^5 \) or \( 10^6 \) pixels). Moreover, depending on the application, the ratio between black and white pixels can vary a lot among different images and therefore we may get a good dispersion of the input probabilities. It should be noted, however, that the method is not successful with any arbitrary sets of images since the independence assumption may not hold in the general case. In the next section we show simulation experiments illustrating the points raised above. A possible application of this method lies in digital communications where we aim at the separation of binary signals propagating through nonlinear channels.

4. SIMULATIONS

We have run a number of Monte-Carlo experiments with different numbers of randomly generated binary sources \( n \). The nonlinear mixtures were generated using model equations (3)-(5) choosing the mixing parameters also randomly from the normal distribution. Figures 1 – 4, show the Bit Error Rate averaged over all reconstructed sources after 1000 Monte Carlo runs, for \( n = 2, 3, 4, \) and 5, respectively. The BER performance is obtained for various source probability sets and it is plotted against \( N \). Clustering is done by identifying the unique values of \( x \) (within tolerance = \( 10^{-6} \)) and probability estimation is done by simple voting of each sample to the closest cluster center. We note that the results improve as the probabilities dispersion increases. The number of sources also has a severe effect on the performance as the number of samples, \( N \), required to achieve same BER, grows exponentially with \( n \). For \( n = 5 \) BER is still quite significant even for \( N = 50000 \).

![Figure 1. Average Bit Error Rate (BER) for \( n = 2 \) sources accumulated over 1000 Monte Carlo simulations. Plotted as a function of the data-set size. Mixing parameters: \( \alpha = [-0.6314, -0.9890, -0.1815] \).](image1)

![Figure 2. Average BER for \( n = 3 \) sources. Mixing parameters: \( \alpha = [-0.2615, -2.1343, 0.4836, 1.0264, 1.0835, 1.0658, -0.0404] \).](image2)

Figure 5 shows the result of the blind separation of three inherently binary or binarized images (“cameraman”, “testpattern” and “text”) from a single nonlinear mixture (top-left image). For these images the source probabilities are \( p_1 = 0.7359 \), \( p_2 = 0.6783 \), and \( p_3 = 0.8929 \). In this case we obtained perfect reconstruction (BER = 0).

5. CONCLUSION

We have presented a new blind method for the separation of multiple binary sources from a single general nonlinear mixture. The method assumes biased sources with diverse probabilities and takes advantage of this asymmetry to transform the problem into a linear BSS problem involving the log probabilities of the output clusters. We then proceed to solve the problem using previous results on underdetermined linear binary BSS. The proposed method is computationally efficient as it based on 1-d clustering and does not involve lengthy iterative optimization. Its disadvantages include the large dataset sizes required for accurate probability estimation and exponential growth of the number of clusters with respect to the number of sources. In our future work we shall present a systematic study of the sensitivity of the approach in the presence of noise and also the possible extension to the multiple output case.
Fig. 3. Average BER for $n = 4$ sources. Mixing parameters: $\alpha = [0.388, -1.870, 0.768, 0.307, -0.294, 0.182, -0.068, -1.455, 0.284, 1.657, 1.655, 1.000, -1.212, 1.282, 0.176]$. 

6. REFERENCES


