A Clustering Approach for the Blind Separation of Multiple Finite Alphabet Sequences from a Single Linear Mixture

Konstantinos I. Diamantaras

Department of Informatics, Technological Education Institute of Thessaloniki, Sindos, GR 57400, Greece. E-mail: kdiamant@it.teithe.gr.

Abstract

In this paper we treat the blind separation problem of binary signals and multi-level PAM signals from a single real mixture or a single complex mixture, respectively. Our approach is based on the clustering of the observation values and the close relationship between the position of the cluster centers and the mixing coefficients. Under mild assumptions, our mathematical formulation yields two deterministic algorithms for the blind estimation of the mixing operator. In the real mixture case we derive a finite, recursive algorithm exploiting the arrangement of the centers along the 1-D line, while in the complex mixture case we exploit the properties of the convex hull of the 2-D data cloud to estimate the mixing parameters. In the absence of noise and for any number of sources, both methods yield perfect results. Following the parameter estimation step, the source symbols can be estimated using a nearest neighbor rule. In the noisy case, our error analysis shows that the parameter estimation error increases smoothly with the noise power, while the source estimate bit error rate depends on relative size of the noise power and the minimum distance between the cluster centers.

1 Introduction

The blind separation of signals from a set of observed linear mixtures with constant coefficients is known as Blind Signal Separation from Instantaneous Mixtures (BSS-IM). This problem finds applications under a variety of contexts. For example, in the reverse link model of wireless, multiuser communications, n user signals arrive at the base station impinging on an array of $M_a > 1$ antennas. Each antenna records a mixture of the signals affected by multipath results such as delay spread, frequency spread, and angle spread. A core problem in multiuser blind equalization/signal separation is the estimation of the sources using just the antenna baseband data sampled at a

suitable rate. If the delay spread is negligible and the sources are synchronized then the baseband data model takes the form of a linear combination of the sources with constant, complex mixing coefficients [32]. Even though the BSS-IM problem seems simplistic it is important because realistic FIR-MIMO equalization scenarios allowing large delay spread, unsynchronized sources and arbitrary modulation functions can be reduced to the above problem. An important special case is the one where the sources transmit symbols from a Finite Alphabet (FA). In particular, binary antipodal symbols $\{-1, +1\}$ appear in many digitally modulated schemes such as, Binary Phase Shift Keying (BPSK) [34], Direct-Sequence CDMA [38], e.a. BSS-IM for FA signals has attracted a lot of attention and various algorithms have been proposed in the literature, especially for binary sources, including, ILSE, ILSP, [35], ACMA [37], SD [1,22], AMiSRoF [15], etc.

Digital communications is not the only reason for our interest in BSS-IM. In fact, there is a fast growing literature on the problem of blind source separation (BSS) from real or complex instantaneous mixtures. Traditional approaches are usually based on appropriate statistical problem formulations. Higher order statistics are used in the Independent Component Analysis (ICA) method which assumes that the sources are white, independent random sequences [8,6]. Second order statistics can be also used provided that the source signals are colored [5]. These methods have found applications in array signal processing [27], speech separation [36], medical signal processing [21], industrial fault detection [33], feature extraction for image and speech data [23,2,3], etc. For a thorough discussion of ICA, BSS and related topics see [18,7].

In this paper we study the BSS-IM problem for binary antipodal sources or M-ary PAM sources under the constraint that there is only one available mixture although there are $n > 1$ sources. The importance of the problem is obvious since the reduction of the number of sensors affects the cost and the complexity of the blind signal separation system. The separation of more sources than mixtures is attracting increasing attention in recent years. One of the first approaches was proposed by Belouchrani and Cardoso [4]. This method is based on the FA property of the sources and offers the maximum likelihood estimation of the mixing parameters using the EM algorithm. Although it is primarily concerned with the case where the number of observations is greater or equal to the number of sources the authors present an example where the algorithm works with 2 observations and 3 sources. However, our simulations show that with 1 mixture and $n > 2$ sources the method does not perform well because the EM algorithm is very often trapped in local minima. Since good initial conditions are difficult to derive the algorithm does not seem to be a promising candidate for this case. Another approach based on Self-Organizing Maps (SOMs) was presented in [17]. Again the method is designed primarily for problems with equal numbers of sources and mixtures. Performance decreases rapidly when the number of sensors m becomes smaller

than the number of sources n. For example, after 1000 trials of the algorithm with $m = 1$ and $n = 4$ no source signal was recovered for 989 times. The problem of nonlinear BSS was treated in [30] using a maximum likelihood approach. In the case of binary sources the method can work for $n > m$ and it becomes similar to [4]. The same drawbacks as discussed above apply to this method as well. More recently, Lee e.a. [24] proposed a gradient learning algorithm for the maximization of the data likelihood function. The method applies to continuous signals and can successfully separate up to 4 speech signals from 2 mixtures. For binary sources the method becomes similar to the maximum likelihood approach in [4].

Our approach is not based on the optimization of signal statistics but on the clustering of the observed data. We use the FA property of the sources to obtain a theory which relates the position of the cluster centers with the mixing coefficients. We'll find that the problem is more combinatorial in nature rather than stochastic. Preliminary results of this research have been presented elsewhere [9,10] for the special case of binary antipodal sources and one real mixture. Li, e.a. [25] have also presented a modification of [9] for the same type of sources. Here we extend this approach to the important case of PAM sources with M-levels and complex mixing coefficients, which is of interest in digital communications. In the real case we obtain a finite, recursive algorithm which, under very mild assumptions and in the absence of noise, provides perfect mixing vector estimation and source separation (Section 2). Extensive performance analysis under noisy conditions is provided in Section 3. The extension to the case of multilevel PAM sources and complex mixing coefficients is treated in Section 4. Simulation results for both real and complex cases are presented in Section 5. We conclude in Section 6.

1.1 Problem formulation

Let $x(k)$ denote a scalar observed sequence, where $k = 1, \cdots, N$, is the discrete time index. The data model is described by a mixing vector $\mathbf{a} = [a_1, \cdots, a_n]^T$ and additive noise $e(k)$:

$$x(k) = \mathbf{a}^T \mathbf{s}(k) + e(k) \tag{1}$$

where $\mathbf{s}(k) = [s_1(k), \cdots, s_n(k)]^T$ is the vector source sequence. First, we shall treat the BSS-IM problem with real mixing coefficients in Section 2. The complex case will be discussed in Section 4. The samples of each individual source are i.i.d. variables: $s_i(k)$, for all k, drawn from the binary antipodal alphabet $\mathsf{A}_b = \{-1, 1\}$, in the real case, or from a real multisymbol alphabet A, in the complex case. All sources are mutually independent and the sequence $\{\mathbf{s}(1), \cdots, \mathbf{s}(N)\}$ is assumed to be rich enough so that every combination vec-

tor in A_b^n or A^n appears at least once for some k. We make no assumptions regarding the probability distribution of the sources, or any other statistical property of $\mathbf{s}(k)$. The noise $e(k)$ is assumed to be zero-mean Gaussian with unknown variance σ^2. The problem is to find both the mixing vector $\mathbf{a}$ and the sources $s_i(k)$ up to a permutation and sign. The scaling ambiguity, typical of most BSS problems, is not present here since the source samples can take only values of certain magnitude.

2 Binary sources and real mixture

2.1 Mathematical preliminaries

Consider the case of n binary antipodal sources s_i, so every vector sample $\mathbf{s}(k)$ can take $M = 2^n$ values. Let $\mathcal{S} = \{\mathbf{b}_1, \cdots, \mathbf{b}_M\}$ be the set of possible values for $\mathbf{s}(k)$, where $\mathbf{b}_i$, $i = 1, \cdots, M$, are the combination vectors $[\pm 1, \cdots, \pm 1]$. We shall also define $\mathbf{B}^{(n)}$, be the $2^n \times n$ matrix with rows $\mathbf{b}_1$, ..., $\mathbf{b}_M$:

$$\mathbf{B}^{(n)} \triangleq \begin{bmatrix} -1 & -1 & \cdots & -1 & -1 \\ -1 & -1 & \cdots & -1 & 1 \\ -1 & -1 & \cdots & 1 & -1 \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & 1 & \cdots & -1 & 1 \\ 1 & 1 & \cdots & 1 & -1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

The rows of $\mathbf{B}^{(n)}$ are symmetric in the following sense: if $\mathbf{b}_i^T$ and $\mathbf{b}_{n-i+1}^T$ are the i-th and $(n-i+1)$-th rows, respectively, then $\mathbf{b}_i = -\mathbf{b}_{n-i+1}$.

With these definitions at hand we can write the probability distribution of $x(k)$ as follows

$$p(x(k)) = \sum_{\mathbf{s}(k) \in \mathcal{S}} p(x(k) \,|\, \mathbf{s}(k))\, p(\mathbf{s}(k)) = \sum_{i=1}^{M} p(x(k) \,|\, i) P_i \tag{2}$$

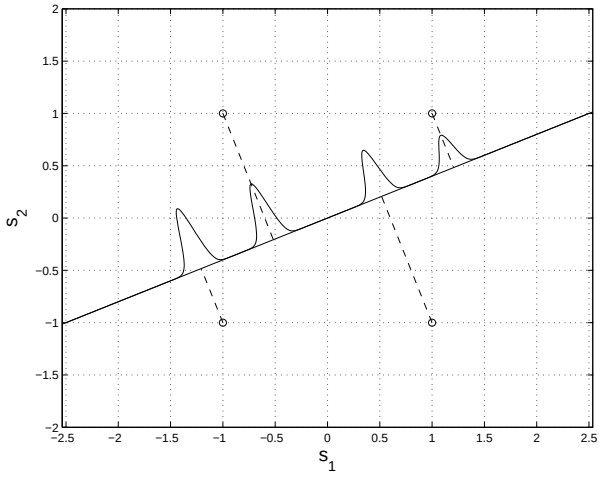


Fig. 1. The distribution $p(x)$ of x for two antipodal sources s_1, s_2 is the mixture of 4 Gaussian "bells" of equal variances but different amplitudes. If $\|\mathbf{a}\| = 1$ then the bell centers are the projections of the vectors $[-1,-1]$, $[-1,+1]$, $[+1,-1]$, $[+1,+1]$, on the direction $\mathbf{a}$ as shown in the figure.

where $p(x(k) \mid i)$ and P_i stand for $p(x(k)|\mathbf{s}(k) = \mathbf{b}_i)$ and $P(\mathbf{s}(k) = \mathbf{b}_i)$ respectively. For Gaussian noise with variance σ^2 we have

$$p(x(k) \mid i) = \frac{1}{(2\pi)^{1/2}\sigma} exp\{-\frac{1}{2\sigma^2}(x(k) - c_i)^2\} \tag{3}$$

where

$$c_i \stackrel{\triangle}{=} \mathbf{b}_i^T \mathbf{a} \tag{4}$$

is the *center* (or the mean) of class i. From (2), (3) it follows that $p(x(k))$ is a mixture of $M = 2^n$ Gaussians with the same variance parameter σ^2, with different centers (in general) at $\mathbf{b}_i^T\mathbf{a}$, and with amplitudes P_i (see Fig. 1). If the sources are independent with equal probabilities for the symbols $+1$ and -1 then the input vectors $\mathbf{b}_i$ will be equiprobable, so $P_1 = \cdots = P_M = 1/M$.

2.2 Structure of the center vector

Equation (4) can be written as

$$[c_1 \cdots c_M]^T = [\mathbf{b}_1 \cdots \mathbf{b}_M]^T \mathbf{a} \tag{5}$$

or

$$\mathbf{c} = \mathbf{B}^{(n)}\mathbf{a}$$

where we made the obvious definition for $\mathbf{c}$. The special structure of $\mathbf{B}^{(n)}$ induces a special structure on the center vector $\mathbf{c}$. As we shall see next, this allows us to find the mixing parameters in a recursive way using a finite number of steps.

In the subsequent discussion we shall make the following assumption

A1: The mixing parameters are positive and arranged in decreasing order:

$$a_1 > a_2 > \cdots > a_n > 0. \tag{6}$$

Assumption **A1** does not hurt generality. For example, if some parameter a_i is negative we can change its sign, together with the sign of the associated signal s_i, without affecting the observation signal x. Then we can rearrange the positive parameters a_i in decreasing order, while at the same time we rearrange the order of the sources, and still obtain the same x. Thus any set of real parameters a_i we can forced to satisfy (6) leaving x unaltered. It follows that the sign and order of the original source signals are inherently unobservable. No method can retrieve them without any further knowledge regarding the source signals.

We shall treat the cases for $n = 2$ and $n > 2$ separately. First, we'll show that for $n = 2$ the true centers c_i are arranged in increasing order.

Lemma 1 *Let* $\alpha_1 > \alpha_2 > 0$ *and* $[\gamma_1, \cdots, \gamma_4]^T = \mathbf{B}^{(2)}[\alpha_1, \alpha_2]^T$. *Then* $\gamma_1 < \gamma_2 < \gamma_3 < \gamma_4$.

PROOF. See the appendix.

Next, for $n > 2$ we'll show that the following statements are true:

(a) the sequence c_1, ..., c_M can be partitioned in consecutive quadruples, each arranged in increasing order;

(b) the first 3 centers $c_1 < c_2 < c_3$ are the three smallest values in the sequence c_i;

(c) the last 3 centers $c_{M-2} < c_{M-1} < c_M$ are the three largest values in the sequence c_i;

(d) The two smallest mixing parameters a_{n-1}, a_n, can be retrieved from c_1, c_2, and c_3;

(e) The differences $c_{4i+j} - c_{4i+1}$, $j = 2, 3, 4$, in the i-th quadruple are functions of a_{n-1}, a_n and are independent of i.

Lemma 2 *Let $n > 2$, $M = 2^n$, $\boldsymbol{\gamma} = [\gamma_1, \cdots, \gamma_M]^T = \mathbf{B}^{(n)}\boldsymbol{\alpha}$, where the elements α_1, ..., α_n, of the vector $\boldsymbol{\alpha}$ are positive reals, arranged in decreasing order. Let $\overline{\gamma}_1$, ..., $\overline{\gamma}_M$ be the sequence γ_i arranged in increasing order. Then*

$$\begin{aligned}
\overline{\gamma}_1 = \gamma_1 &= -\gamma_M &&= -\overline{\gamma}_M \\
\overline{\gamma}_2 = \gamma_2 &= -\gamma_{M-1} &&= -\overline{\gamma}_{M-1} \\
\overline{\gamma}_3 = \gamma_3 &= -\gamma_{M-2} &&= -\overline{\gamma}_{M-2} \ .
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\alpha_n &= (\overline{\gamma}_2 - \overline{\gamma}_1)/2 \\
\alpha_{n-1} &= (\overline{\gamma}_3 - \overline{\gamma}_1)/2 \ .
\end{aligned}$$

PROOF. See the appendix.

Lemma 3 *Let $n > 2$, and $\boldsymbol{\gamma}$, $\boldsymbol{\alpha}$, M as in Lemma 2. Then*

(a) the sequence γ_j, $j = 1, \cdots, M$, is partitioned in consecutive quadruples where each one is arranged in increasing order, i.e. for all $i = 0, \cdots, 2^{n-2} - 1$, $\gamma_{4i+1} < \gamma_{4i+2} < \gamma_{4i+3} < \gamma_{4i+4}$.

(b) The differences $\gamma_{4i+2} - \gamma_{4i+1} = 2\alpha_n$, $\gamma_{4i+3} - \gamma_{4i+1} = 2\alpha_{n-1}$, $\gamma_{4i+4} - \gamma_{4i+1} = 2(\alpha_{n-1} + \alpha_n)$, are independent of i.

PROOF. See the appendix.

In the following we shall assume that our center estimates $\hat{c}_i$ are arranged in increasing order. If we further assume that they are equal to the true centers c_i, except for an unknown permutation π, $\hat{c}_i = c_{\pi(i)}$. then from Lemmata 1, 2, we have the following corollaries:

Corollary 4 *Let $n = 2$ and let the estimated centers $\hat{c}_i$, $i = 1, \cdots, 4$, be arranged in increasing order. Then $\hat{c}_i = c_i$, for all i, and*

$$a_2 = (\hat{c}_2 - \hat{c}_1)/2 \ , \tag{7}$$
$$a_1 = (\hat{c}_3 - \hat{c}_1)/2 \ . \tag{8}$$

Corollary 5 *Let $n = 3$ and let the estimated centers $\hat{c}_i$, $i = 1, \cdots, 8$, be arranged in increasing order. Then $\hat{c}_1 = c_1$, $\hat{c}_2 = c_2$, $\hat{c}_3 = c_3$ and*

Algorithm 1. Clustering
Step 1. Collect N samples of the 1-D real sequence $x(k)$.
Step 2. Sort $x(1)$, ..., $x(N)$ in ascending order to obtain the sequence $x_s(1)$, ..., $x_s(N)$.
Step 3. Form M groups of sorted data $\{x_s(1), \cdots, x_s(T)\}$, ..., $\{x_s((M-1)T+1), \cdots, x_s(MT)\}$, where $T = N/M$, and estimate the cluster centers by the median values $\hat{c}_i = x_s(\lfloor iT + T/2 \rfloor)$, $i = 1, \cdots, M$, of these groups.

$$a_3 = (\hat{c}_2 - \hat{c}_1)/2 \ , \tag{9}$$
$$a_2 = (\hat{c}_3 - \hat{c}_1)/2 \ , \tag{10}$$
$$a_1 = -(\hat{c}_2 + \hat{c}_3)/2 \ . \tag{11}$$

2.3 Estimating the centers of the mixture model

The estimation of a mixture density from data samples is a classical statistical problem [13][chapter 6],[29]. Especially Gaussian mixtures have received a lot of attention due to their applications in pattern recognition and signal processing. Different methods for estimating the parameters in a Gaussian mixture include: Radial Basis Function (RBF) neural networks [28], neural mixtures of experts [19,20], and the Expectation-Maximization (EM) algorithm [12,14]. See [16] for a comprehensive treatment of most of these methods.

For our experiments, we chose a simple and efficient clustering algorithm which is only applicable to real, 1-D sequences where all clusters containing approximately the same number of samples (Algorithm 1). This algorithm is not iterative, and therefore it is very fast, it can not be trapped in local minima and it does not require any initial conditions. This is the clustering method we use in our simulations in Section 5.

2.4 Recursive Algorithm for estimating the mixing parameters

For $n = 2$ or $n = 3$ the mixing parameters can be retrieved immediately using Corollaries 4 and 5. For $n > 3$ our preceding analysis suggests the construction of a recursive algorithm (see Algorithm 2).

We'll show next that Algorithm 2 yields perfect estimates $\hat{a}_i$ if the estimated centers $\hat{c}_i$ are equal to the true centers c_i except for some unknown ordering. Step 1 arranges the estimated centers in increasing order. Step 2 estimates a_{n-1} and a_n according to 2. Using $\hat{a}_n = a_n$ and $\hat{a}_{n-1} = a_{n-1}$ in Step 3 we obtain the differences $\hat{dc}_1 = \hat{c}_2 - \hat{c}_1$, $\hat{dc}_2 = \hat{c}_3 - \hat{c}_1$, $\hat{dc}_3 = \hat{c}_4 - \hat{c}_1$, which will be used in

Algorithm 2. Obtaining the mixing parameters from the cluster centers.
Step 1. Sort the estimated centers $\hat{c}_i$ in increasing order
Step 2. Estimate a_n, a_{n-1} from

$$\hat{a}_n = (\hat{c}_2 - \hat{c}_1)/2 \tag{12}$$

$$\hat{a}_{n-1} = (\hat{c}_3 - \hat{c}_1)/2 \tag{13}$$

according to Lemma 2.
Step 3. According to Lemma 3 the sequence $\{c_i\}$ is partitioned in quadruples of the form $[y, y + dc_1, y + dc_2, y + dc_3]$. Estimate the differences dc_i:

$$\hat{dc}_1 = 2\hat{a}_n$$
$$\hat{dc}_2 = 2\hat{a}_{n-1}$$
$$\hat{dc}_3 = 2(\hat{a}_{n-1} + \hat{a}_n)$$

Step 4. Remove the set $\{\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_1 + \hat{dc}_3\}$ from the sequence $\{\hat{c}_i\}$. If these numbers are not present in the sequence, remove the values that are closest to them. Set $\hat{c}'_1 = \hat{c}_1 + \hat{a}_n + \hat{a}_{n-1}$ as the first element of a new sequence $\{\hat{c}'_i\}$.
Step 5.
Repeat until all elements have been removed:
Find the smallest element $\hat{c}_j$ of the
remaining sequence $\{\hat{c}_i\}$;
Remove the set
$\{\hat{c}_j, \hat{c}_j + \hat{dc}_1, \hat{c}_j + \hat{dc}_2, \hat{c}_1 + \hat{dc}_3\}$ from $\{\hat{c}_i\}$;
If these numbers are not present in $\{\hat{c}_i\}$,
remove the values closest to them;
Add $\hat{c}_j + \hat{a}_n + \hat{a}_{n-1}$ 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 $\hat{a}_{n'} = \hat{a}_{n-2}$, $\hat{a}_{n'-1} = \hat{a}_{n-3}$. Eventually we will get $n' = 2$ or $n' = 3$ and the parameters a_1 and a_2, or a_1, a_2, and a_3 can be estimated using Corollaries 4 or 5.

conjunction with Lemma 3. Steps 4 and 5 create a new sequence of centers $\{\hat{c}'_i\}$ from the first elements of the quadruples of c_i shifted by $\hat{a}_n + \hat{a}_{n-1} = a_n + a_{n-1}$. Indeed, the minimum element $\hat{c}_j$ of the sequence $\{\hat{c}_i\}$ must be the first element of a quadruple, since $\hat{dc}_1, \hat{dc}_2, \hat{dc}_3 > 0$. Therefore $\hat{c}_j$ has the following form

$$\hat{c}_j = [\pm 1 \cdots \pm 1 \ -1 \ -1][a_1 \cdots a_{n-2} \ a_{n-1} \ a_n]^T$$

By removing one-by-one all the quadruples and adding the numbers

$$\hat{c}'_i = \hat{c}_j + a_{n-1} + a_n$$

for every minimum element $\hat{c}_j$ of each quadruple we are able to collect numbers of the form $[\pm 1 \cdots \pm 1][a_1 \cdots a_{n-2}]^T$ for all combinations of $+1$ and -1. Therefore, we are able to reduce the problem to a similar problem of smaller dimension, i.e. $\{\hat{c}'_i\}$ corresponds to the vector $\mathbf{B}^{(n-2)}[a_1 \cdots a_{n-2}]^T$. The new dimension n' is less 2 than the dimension n of the original problem. So every recursion in Step 6 will estimate a new pair of mixing parameters: (a_{n-2}, a_{n-3}), (a_{n-4}, a_{n-5}), ... The recursion terminates in the cases where $n' = 3$ or $n' = 2$ where the problem is solved using Corollaries 4 or 5.

Algorithm 2 has been originally presented in [9,10]. Later, Li, e.a. [25], presented a clustering-based method which is similar to this algorithm. Their method starts by using Eq. (12) to estimate the smallest mixing parameter a_n. Then the centers are grouped in pairs $(c_{i_k}, c_{i_{(k+1)}})$ with distance $c_{i_k} - c_{i_{(k+1)}} = 2a_n$ between elements. Subsequently, the set $C^{11} = \{c_{i_1}, c_{i_3}, \cdots c_{i_{(M/2)}}\}$, is formed, which includes the centers that correspond to the even lines of $\mathbf{B}^{(n)}$, ie, those lines whose last bit is equal to 1. Then we have a new problem of size $n-1$ and the process can be recursively applied to obtain all mixing parameters a_{n-1}, ..., a_1.

It is obvious that any algorithm that solves the blind source separation of many binary sources from a single linear mixture can be applied to the blind deconvolution of an FIR filter with binary input since the equation describing the filter

$$x(k) = \sum_{i=1}^{n} h_i \bar{s}(k - i + 1)$$

is identical to Eq. (1): simply substitute h_i with a_i and $\bar{s}(k-i+1)$ with $s_i(k)$. Li, e.a., have applied their method to handle the blind deconvolution of a linear FIR system with binary inputs in [26].

2.5 Example

Consider a problem with $n = 4$ sources and mixing vector $\mathbf{a} = [0.88\ 0.45\ 0.34\ 0.27]^T$. The properly ordered true centers are

$$\begin{aligned}\mathbf{c} &= \mathbf{B}^{(4)}\mathbf{a} \\ &= [-1.94,\ -1.40,\ -1.26,\ -0.72,\end{aligned}$$

$$
\begin{aligned}
&-1.04,\ -0.50,\ -0.36,\ 0.18,\\
&-0.18,\ 0.36,\ 0.50,\ 1.04,\\
&0.72,\ 1.26,\ 1.40,\ 1.94]^T
\end{aligned}
\tag{14}
$$

Note, that each quadruple in $\mathbf{c}$ is sorted in increasing order. Suppose now, that the centers are perfectly estimated. Even so, their proper order is lost. Instead, we may produce a sorted list in ascending order (Step 1):

$$
\begin{aligned}
\hat{\mathbf{c}} = [&-1.94,\ -1.40,\ -1.26,\ -1.04,\\
&-0.72,\ -0.50,\ -0.36,\ -0.18,\\
&0.18,\ 0.36,\ 0.50,\ 0.72,\\
&1.04,\ 1.26,\ 1.40,\ 1.94]^T
\end{aligned}
$$

Then the parameters a_3, a_4, are readily estimated from $\hat{c}_1 = -1.94$, $\hat{c}_2 = -1.40$, $\hat{c}_3 = -1.26$ (Step 2):

$$
\begin{aligned}
\hat{a}_4 &= (\hat{c}_2 - \hat{c}_1)/2 = 0.27 = a_4\\
\hat{a}_3 &= (\hat{c}_3 - \hat{c}_1)/2 = 0.34 = a_3
\end{aligned}
$$

Next, we compute the following differences (Step 3):

$$
\begin{aligned}
dc_1 &= 2\hat{a}_4 = 0.54\\
dc_2 &= 2\hat{a}_3 = 0.68\\
dc_3 &= 2(\hat{a}_3 + \hat{a}_4) = 1.22
\end{aligned}
$$

It is easy to verify that every quadruple in $\mathbf{c}$ can be written as $[y, y + dc_1, y + dc_2, y + dc_3]$ for some y (ref Eq. (14)).

We remove the set $\{\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_3 + dc_3\} = \{-1.94,\ -1.40,\ -1.26,\ -1.04\}$ from the sequence $\{\hat{c}\}$ (Step 4) and set the first element of a new sequence $\{\hat{c}'\}$ to be

$$
\hat{c}'_1 = \hat{c}_1 + \hat{a}_3 + \hat{a}_4 = -1.33\ .
$$

Repeatedly, we remove elements from $\{\hat{c}\}$ adding new elements to $\{\hat{c}'\}$ as

shown below (Step 5):

Iter.	Remove from $\{\hat{c}\}$	Add to $\{\hat{c}'\}$
1	$-1.04,\ -0.50,\ -0.36,\ 0.18$	$\hat{c}'_2 = -0.43$
2	$-0.18,\ 0.36,\ 0.50,\ 1.04$	$\hat{c}'_3 = 0.43$
3	$0.72,\ 1.26,\ 1.40,\ 1.94$	$\hat{c}'_4 = 1.33$

Now, we are faced with a smaller problem with fewer centers $\hat{\mathbf{c}} = [-1.33,\ -0.43,\ 0.43,\ 1.33]^T$ and dimension $n' = n - 2 = 2$. The algorithm is recursively repeated on the new problem (Step 6). In this case the parameters a_1, a_2, can be estimated using Corollary 4:

$$\hat{a}_2 = (\hat{c}'_2 - \hat{c}'_1)/2 = 0.45 = a_2$$
$$\hat{a}_1 = (\hat{c}'_3 - \hat{c}'_1)/2 = 0.88 = a_1$$

and the algorithm terminates. Our final estimate is identical with the actual mixing parameters

$$\hat{\mathbf{a}} = [\hat{a}_1\ \hat{a}_2\ \hat{a}_3\ \hat{a}_4]^T = [0.88\ 0.45\ 0.34\ 0.27]^T = \mathbf{a}\ .$$

2.6 Recovering the sources.

Having an estimate $\hat{\mathbf{a}}$ of the mixing vector, we may proceed to estimate the sources as follows:

$$\hat{\mathbf{s}}(k) = \arg\min_{\mathbf{s}\in\mathcal{S}} \|x(k) - \hat{\mathbf{a}}^T\mathbf{s}\|^2 = \arg\min_{\mathbf{b}_i} \|x(k) - \hat{\mathbf{a}}^T\mathbf{b}_i\|^2\ . \tag{15}$$

For the solution to be unique there must not exist two different binary vectors that yield the same center value. Formally,

A2: No two centers are identical:

$$i \neq j\ \Rightarrow\ \mathbf{b}_i^T\hat{\mathbf{a}} \neq \mathbf{b}_j^T\hat{\mathbf{a}} \tag{16}$$

This condition was first proposed in [25]. In this case the vector $\hat{\mathbf{a}}$ is called *bi-independent*.

For each time index k, Eq. (15) corresponds to a separate optimization problem. Therefore, we are faced with N independent such problems. Given the bi-independence of $\hat{\mathbf{a}}$, the solution of each problem is unique and the solution

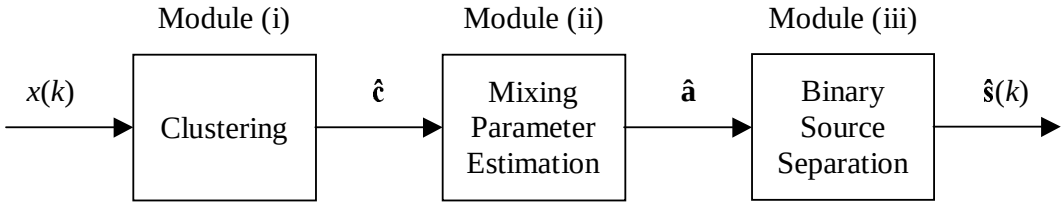


Fig. 2. The proposed binary source separation system.

process is simple, since we only need to search in a small set $\mathcal{S}$ of possible values of $\mathbf{s}$.

3 Error analysis

The proposed blind source separation system consists of three parts: (i) a clustering module which estimates c_i, σ, (ii) a module for the estimation of the mixing parameters a_i based on $\hat{\mathbf{c}}$, and (iii) a nearest neighbor estimator of the binary sources (see Fig. 2). Each stage of the system contributes an error term in the total source estimation performance.

Module (i) is implemented using Algorithm 1. Figure 3 shows the normalized center estimation error defined as follows

$$J_c = \|\hat{\mathbf{c}} - \mathbf{c}_s\|/\|\mathbf{c}_s\| \tag{17}$$

where $\mathbf{c}_s = [c_{s,1} \cdots c_{s,M}]^T$ is the true center vector sorted in increasing order. The error is clearly a function of the observation signal to noise ratio (SNR).

Module (ii) uses Algorithm 2 with the estimated cluster centers as inputs. Although the analysis for the general case is difficult, the error analysis is straightforward for $n = 2$. Let w_i and v_i be the center and mixing parameter estimation errors, respectively,

$$\hat{c}_i = c_{s,i} + w_i$$
$$\hat{a}_i = a_i + v_i$$

For $n = 2$ using Corollary 4 we obtain

$$\hat{a}_1 = (\hat{c}_3 - \hat{c}_1)/2 = a_1 + (w_3 - w_1)/2 \tag{18}$$
$$\hat{a}_2 = (\hat{c}_2 - \hat{c}_1)/2 = a_2 + (w_2 - w_1)/2 \tag{19}$$

so

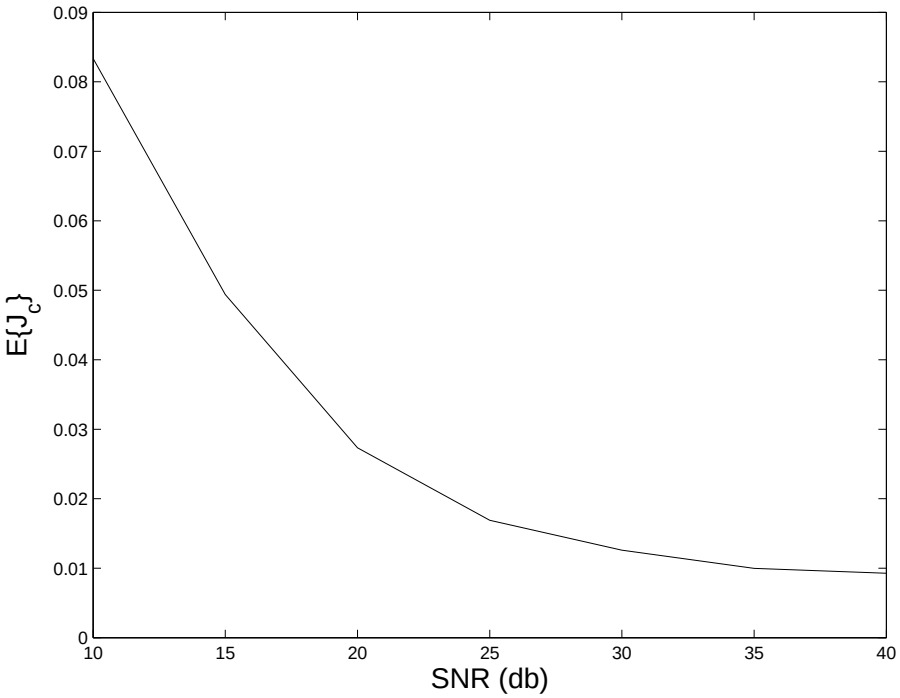


Fig. 3. The average normalized center estimation error from 5000 Monte Carlo experiments obtained by the clustering algorithm.

$$v_1 = (w_3 - w_1)/2 \tag{20}$$
$$v_2 = (w_2 - w_1)/2 \tag{21}$$

Similarly, for $n = 3$ we obtain (cf. Corollary 5)

$$v_1 = -(w_2 + w_3)/2 \tag{22}$$
$$v_2 = (w_3 - w_1)/2 \tag{23}$$
$$v_3 = (w_2 - w_1)/2 \tag{24}$$

Thus, for $n = 2$ and $n = 3$, there is a linear relationship between the input and output errors in Algorithm 2. Taking the errors w_i to be independent, zero mean, with equal variance σ_w^2, we have

$$E\{v_i^2\} = \sigma_w^2/2 \ . \tag{25}$$

Defining the absolute center and mixing vector errors

$$D_c^2 = \|\hat{\mathbf{c}} - \mathbf{c}_s\|^2 = \sum_{i=1}^{M} w_i^2 \tag{26}$$
$$D_a^2 = \|\hat{\mathbf{a}} - \mathbf{a}\|^2 = \sum_{i=1}^{n} v_i^2 \tag{27}$$

and using (25) we obtain

$$E\{D_c^2\} = M\sigma_w^2 \tag{28}$$
$$E\{D_a^2\} = n\sigma_w^2/2 \tag{29}$$

Since $M = 2^n$ we have

$$E\{D_a^2\} = \frac{n}{2^{n+1}} E\{D_c^2\} \tag{30}$$

In analogy to the normalized center estimation error define the normalized mixing vector estimation error as follows:

$$J_a = \|\hat{\mathbf{a}} - \mathbf{a}\|/\|\mathbf{a}\| \tag{31}$$

Clearly then, $J_c^2 = D_c^2\|\mathbf{c}_s\|^2$ and $J_a^2 = D_a^2\|\mathbf{a}\|^2$, so

$$E\{J_a^2\} = \frac{n\|\mathbf{c}_s\|^2}{2^{n+1}\|\mathbf{a}\|^2} E\{J_c^2\} \tag{32}$$

Since $c_{s,i}^2 = (a_1 \pm a_2 \cdots \pm a_n)^2$, it is straightforward to verify that

$$\|\mathbf{c}_s\|^2 = 2^n\|\mathbf{a}\|^2 \tag{33}$$

so, for $n = 2$ or 3,

$$E\{J_a^2\} = \frac{n}{2} E\{J_c^2\} \ . \tag{34}$$

For $n > 3$ an analytical expression between v_i and w_i becomes difficult to derive due to the combinatorial nature of Algorithm 2. We can, however, study the performance of the algorithm by simulation. Figure 4 plots the normalized parameter estimation error J_a as a function of the normalized center estimation error J_c, both averaged over 5000 Monte Carlo experiments. We see that the general trend is smooth (almost linear or sublinear) for $n > 3$ and linear for $n = 2$, $n = 3$, as theoretically expected.

Module (iii) deals with the estimation of the sources using the nearest neighbor approach expressed by Eq. (15). Let

$$\hat{g}_i = \hat{\mathbf{a}}^T \mathbf{b}_i$$

$$\hat{g}_h(i) = \arg\min_{g_j}\{|\hat{g}_j - \hat{g}_i| \, ; \, \hat{g}_j > \hat{g}_i\}$$

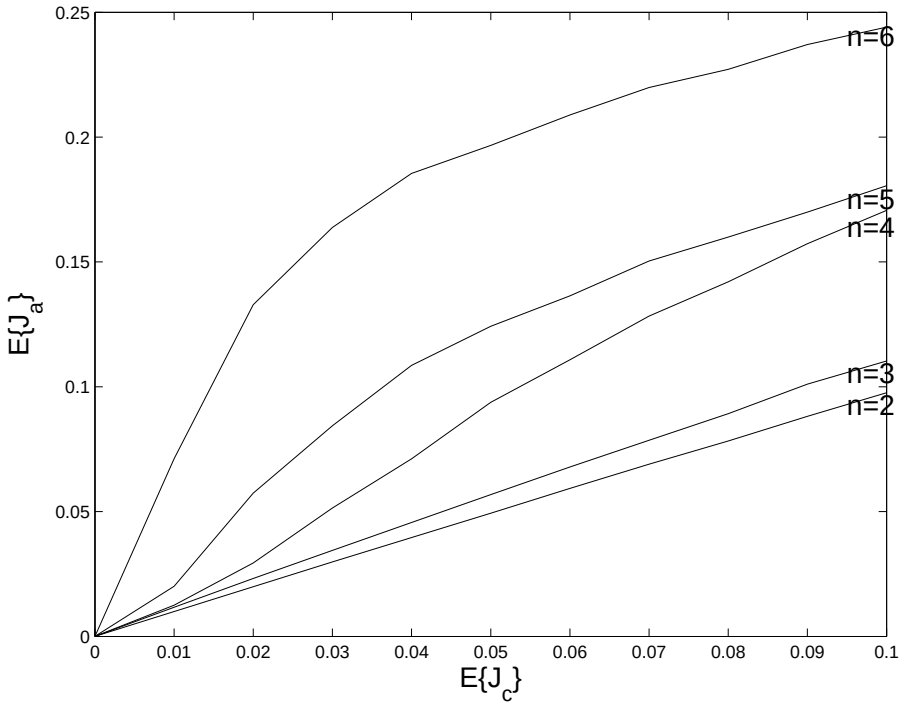


Fig. 4. The average normalized parameter estimation error after 5000 Monte Carlo runs vs. the average normalized center estimation error for various numbers of sources n.

$$\hat{g}_l(i) = \arg\min_{g_j}\{|\hat{g}_j - \hat{g}_i|\,;\, \hat{g}_j < \hat{g}_i\}$$

According to the nearest neighbor rule $\hat{\mathbf{s}}(k)$ is assigned the value $\mathbf{b}_i$ if $x(k)$ belongs to the bin $B_i = (\hat{g}_i - \delta_i^l, \hat{g}_i + \delta_i^h)$ where

$$\delta_i^h = |\hat{g}_h(i) - \hat{g}_i|/2, \tag{35}$$
$$\delta_i^l = |\hat{g}_l(i) - \hat{g}_i|/2\ . \tag{36}$$

Assume that the sample $x(k)$ is generated from a source vector $\mathbf{s}(k) = \mathbf{b}_i$, so $x(k) = \mathbf{a}^T\mathbf{b}_i + e(k)$. Then the vector error probability for the vector $\hat{\mathbf{s}}(k)$ is defined as

$$\begin{aligned} p_e^{vec}(k;i) &\triangleq \mathrm{Prob}(\hat{\mathbf{s}}(k) \neq \mathbf{b}_i \,|\, \mathbf{s}(k) = \mathbf{b}_i) \\ &= 1 - \mathrm{Prob}(\hat{g}_i - \delta_i^l < x(k) < \hat{g}_i + \delta_i^h) \\ &= 1 - \mathrm{Prob}(\hat{\mathbf{a}}^T\mathbf{b}_i - \delta_i^l < \mathbf{a}^T\mathbf{b}_i + e(k) < \hat{\mathbf{a}}^T\mathbf{b}_i + \delta_i^h) \\ &= 1 - \mathrm{Prob}(\mathbf{v}^T\mathbf{b}_i - \delta_i^l < e(k) < \mathbf{v}^T\mathbf{b}_i + \delta_i^h) \end{aligned} \tag{37}$$

where $\mathbf{v} = \hat{\mathbf{a}} - \mathbf{a} = [v_1 \cdots v_n]^T$. The above vector error probability is the same for all times k for which $x(k)$ is generated by $\mathbf{s}(k) = \mathbf{b}_i$. Thus, for the sake of simplicity, we write $p_e^{vec}(i)$ instead of $p_e^{vec}(k;i)$. The error $e(k)$

follows the Gaussian distribution with zero mean and variance σ^2. Using the corresponding cdf $F(x)$ and the error function erf, we can write

$$\begin{aligned} p_e^{vec}(i) &= F(\mathbf{v}^T\mathbf{b}_i - \delta_i^l) + 1 - F(\mathbf{v}^T\mathbf{b}_i + \delta_i^h) \\ &= 1 + \frac{1}{2}\mathrm{erf}(\frac{\mathbf{v}^T\mathbf{b}_i - \delta_i^l}{\sqrt{2}\sigma}) - \frac{1}{2}\mathrm{erf}(\frac{\mathbf{v}^T\mathbf{b}_i + \delta_i^h}{\sqrt{2}\sigma}) \ . \end{aligned} \tag{38}$$

Even without mixing parameter estimation error ($\mathbf{v} = 0$) the vector error probability $p_e^{vec}(i)$ is non-zero. The ratios $\delta_i^l/(\sqrt{2}\sigma)$, $\delta_i^h/(\sqrt{2}\sigma)$ determine the quality of the estimator. This is not surprising because these ratios correspond to the distance between the consecutive centers $\hat{g}_l$, $\hat{g}_i$, $\hat{g}_h$. For a given noise power σ^2, the larger the distance between the centers the smaller the probability that $x(k)$ falls outside the bin B_i. Similarly, for given distances between consecutive centers, the larger the noise variance the larger the probability that $x(k)$ falls out of B_i.

The total vector error probability is given by

$$\begin{aligned} P_e^{vec} &= \mathrm{Prob}(\hat{\mathbf{s}}(k) \neq \mathbf{s}(k)) \\ &= 1 - \sum_{i=1}^{M} \mathrm{Prob}(\hat{\mathbf{s}}(k) = \mathbf{b}_i \,|\, \mathbf{s}(k) = \mathbf{b}_i)\, \mathrm{Prob}(\mathbf{s}(k) = \mathbf{b}_i) \\ &= 1 - \sum_{i=1}^{M} [1 - p_e^{vec}(i)]\, P_i \\ &= \sum_{i=1}^{M} p_e^{vec}(i) P_i \end{aligned} \tag{39}$$

For $P_1 = \cdots = P_M = 1/M$ we obtain $P_e^{vec} = 1/M \sum_i p_e^{vec}(i)$. P_e^{vec} is an upper bound for the Bit-Error-Rate (BER) which corresponds to the probability of error for each individual bit in the vector $\hat{\mathbf{s}}(k)$. When the source vector $\mathbf{s}(k) = \mathbf{b}_i$ is wrongly estimated by $\hat{\mathbf{s}}(k) = \mathbf{b}_j$, $i \neq j$, not all the bits of $\mathbf{b}_i$ and $\mathbf{b}_j$ need be different. Assuming equal probabilities for the events (i) that the m-th bits of $\mathbf{b}_i$ and $\mathbf{b}_j$ are equal and (ii) that they are not equal, then a closer estimate of BER would be

$$BER \approx \frac{1}{2} P_e^{vec} \ . \tag{40}$$

Figure 5 shows the total vector probability error P_e^{vec} as a function of the *spreading factor*:

$$q = \frac{\sqrt{2}\sigma}{\min_i\{\delta_i^h, \delta_i^l\}} \ . \tag{41}$$

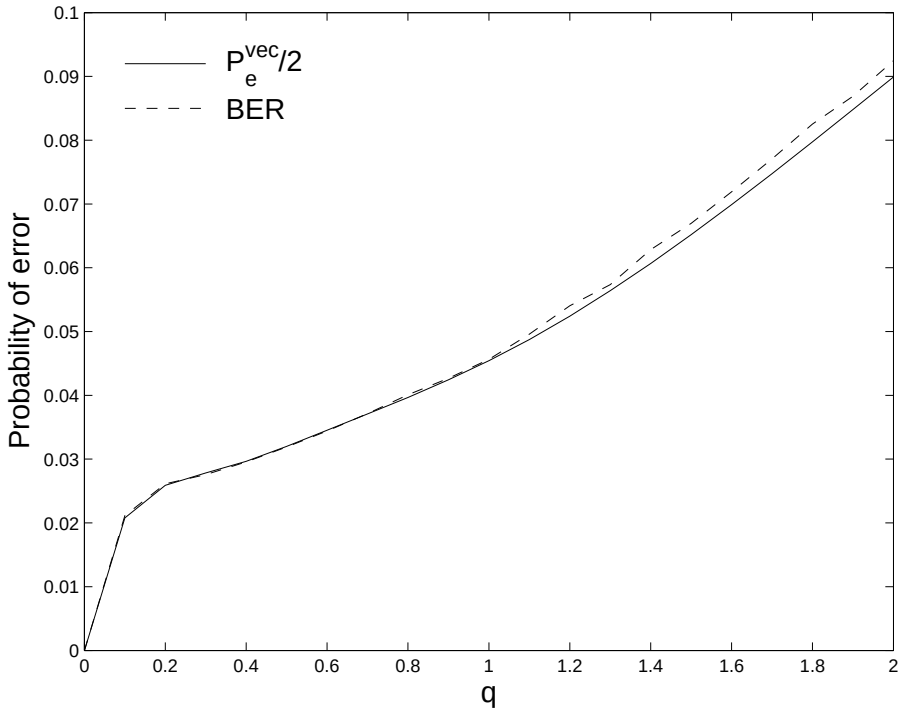


Fig. 5. The average Bit-Error-Rate from 100 Monte Carlo experiments compared to $P_e^{vec}/2$.

Our simulations prove that the estimate (40) is quite accurate. Compare the average BER, computed from 100 Monte Carlo runs, with the theoretically computed $P_e^{vec}/2$.

4 Complex mixing parameters and multilevel PAM sources

In some applications the data model of Eq. (1) involves complex mixing parameters a_i. In digital communications, for example, this is the baseband data model for the receiving antenna, under the assumptions of synchronized sources, negligible multipath effects, and sampling at the bit rate. This communications scenario, although simple, has attracted a lot of attention [37,35,22] because more realistic FIR-MIMO scenarios are usually separated into two sub-problems: a blind multi-user equalization problem and a blind separation problem of the same form as (1). In all the methods proposed in the literature however, the source signals are assumed to impinge on an array of sensors with $M_a > 1$ elements. If we have only one sensor and there are more than two binary sources none of the above methods apply.

By decomposing the complex problem into real and imaginary subproblems

we obtain the following real, two-output model:

$$\mathbf{x}(k) = \mathbf{a}_1 s_1(k) + \cdots + \mathbf{a}_n s_n(k) + \mathbf{e}(k) \tag{42}$$

where $\mathbf{x}(k) = [Re\{x(k)\}, Im\{x(k)\}]^T$, $\mathbf{a}_i = [Re\{a_i\}, Im\{a_i\}]^T$, and $\mathbf{e}(k) = [Re\{e(k)\}, Im\{e(k)\}]^T$. Clearly, a simple solution is to approach Eq. (42) as two distinct single-output problems

$$\begin{aligned} x_R(k) &= Re\{x(k)\} = \mathbf{a}_R^T \mathbf{s}(k) \\ x_I(k) &= Im\{x(k)\} = \mathbf{a}_I^T \mathbf{s}(k) \end{aligned}$$

where $\mathbf{a}_R = [Re\{a_1\}, \cdots, Re\{a_n\}]^T$, $\mathbf{a}_I = [Im\{a_1\}, \cdots, Im\{a_n\}]^T$. These can be solved separately, as discussed in Section 2 and then the two estimated source signals can be combined to obtain a unique source estimate. This approach, however, combines the information from the two subproblems only at the end and not prior to the estimation of the two mixing vectors. Furthermore, it does not apply to multilevel PAM sources.

An alternative approach is to use the geometrical properties of the constellation of $\mathbf{x}(k)$. It is known [11] that the convex hull H, of a 2-D linear mixture $\mathbf{x}$, of n sources s_i, with symbols drawn from the real multilevel alphabet A, ignoring noise, has the following properties:

(1) Every edge $\mathbf{h}_i$ of H is parallel to some mixing vector $\mathbf{a}_j$, $j \in \{0, 1, \cdots, n\}$. Also, $\mathbf{h}_i$ has length $d\|\mathbf{a}_j\|$, where

$$d = \max \mathsf{A} - \min \mathsf{A}$$

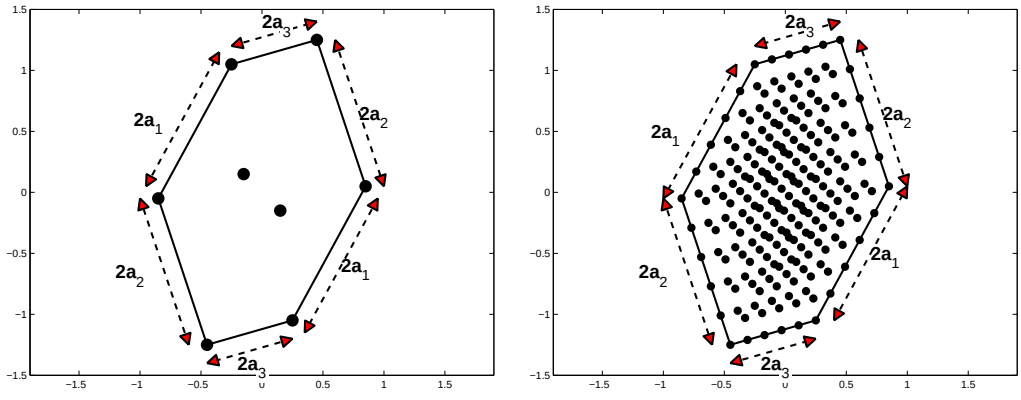
for the given input alphabet A. For the binary antipodal alphabet $\mathsf{A}_b = \{-1, 1\}$, we have $d = 2$.

(2) Every vector $\mathbf{a}_j$ corresponds to a pair of edges, i.e. it is parallel to two edges $\mathbf{h}_i$ and $\mathbf{h}_i'$ of equal length $d\|\mathbf{a}_i\|$. It follows that H has $2n$ edges.

(3) H is symmetric. If the alphabet is symmetric around 0 (e.g. A_b) then the center of symmetry is the point $\mathbf{x}_O = 0$. Otherwise, the center of symmetry is a non-zero point $\mathbf{x}_O' \in \mathbb{R}^2$.

Figure 6 shows the convex hull of a 2-D mixture of $n = 3$ binary sources (Fig. 6a) and the corresponding convex hull for $n = 3$ PAM sources with $L = 6$ levels: $\{-1, -0.6, -0.2, 0.2, 0.6, 1\}$ (Fig. 6b). The convex hull is easily computed using standard libraries (for example, the `convhull` command in MATLAB). From the edges $\mathbf{h}_i$ of the convex hull, the mixing vectors are readily estimated as

$$\hat{\mathbf{a}}_j = \mathbf{h}_i / d.$$



(a) (b)

Fig. 6. (a) Convex hull of the constellation of a 2-D linear mixture of $n = 3$ binary sources. (b) Convex hull of the constellation of $n = 3$ 6-level PAM sources.

In the noise-free case the estimation is perfect except for the unknown sign and permutation of the vectors. In the noisy case, each point in the constellation becomes the center of a cloud of points with spread proportional to the noise variance. Clustering methods, such as the k-means algorithm, the EM algorithm, etc, can be employed again for estimating the cluster centers. From the convex hull of those we estimate the mixing vectors as before. Finally, the sources can be estimated using Eq. (15).

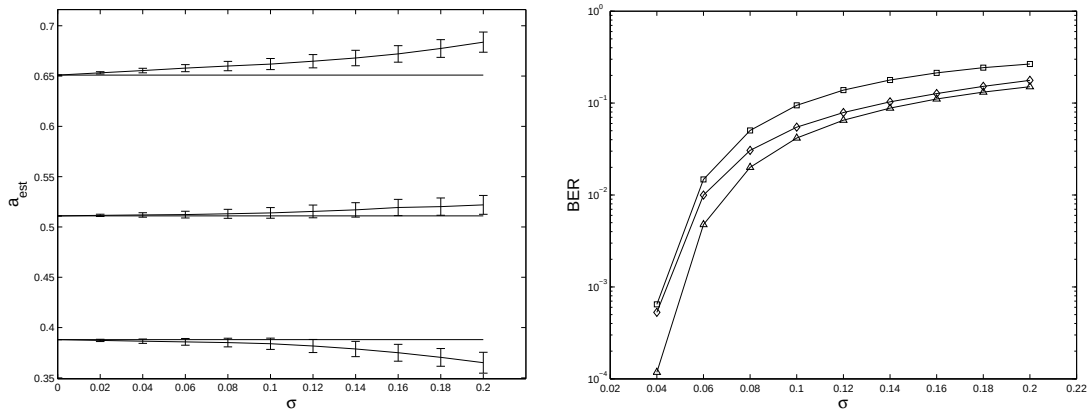
5 Simulation results

5.1 Real mixture

In this section we show the results of a noisy experiment with 1 real observation, $n = 3$ binary sources, $N = 1000$ data samples, and observation noise variance σ varying from 0 to 0.2. The mixing vector and the true center vector arranged in increasing order is given below

$$\mathbf{a} = [0.651,\ 0.511,\ 0.388]^T$$
$$\mathbf{c} = [-1.550, -0.774, -0.528, 0.248, -0.248, 0.528, 0.774, 1.550]^T$$

The minimum half-distance between the centers c_i is $\delta = 0.123$. The method was tested for different noise levels in a series of Monte-Carlo experiments. Figure 7a plots the mean and the variance of the estimates $\hat{a}_i$, $i = 1, 2, 3$, against the observation noise variance σ, for 1000 MC experiments. We see that there is a gradual estimate deterioration as the noise power increases. We don't see however, any abrupt change in the error as σ approaches δ. In



(a) (b)

Fig. 7. Separation of 3 binary sources from one real mixture. (a) Estimates of the mixing parameters as a function of the noise variance σ. (b) Separate Bit Error Rates for the 3 source signals as a function of σ. Both plots (a) and (b) are averaged over 1000 Monte Carlo simulations.

comparison, Figure 7b shows the average Bit-Error-Rate (BER) against σ, for the same 1000 MC runs. Here the BER drops drastically as σ gets below 0.8, or approximately, $\delta/\sqrt{2}$.

5.2 Complex mixture

In this experiment we simulated a complex linear mixture with $n = 3$ binary sources. This corresponds to a 3-input 2-output real system. The mixing vectors and the cluster centers are the columns of the following matrices $\mathbf{A}$ and $\mathbf{C}$, respectively:

$$\mathbf{A} = \begin{bmatrix} -1.6396 & 1.6896 & -0.8289 \\ 0.9802 & 0.1798 & -0.4011 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0.7789 & -0.8789 & 4.1581 & 2.5003 & -2.5003 & -4.1581 & 0.8789 & -0.7789 \\ -0.7589 & -1.5611 & -0.3993 & -1.2015 & 1.2015 & 0.3993 & 1.5611 & 0.7589 \end{bmatrix}$$

The clustering approach is a simple peak-selection procedure from the 2-d histogram of the observation vector $\mathbf{x}$. The average BER for all sources together is shown in Fig. 8, for various levels of the noise variance σ. We note that for $\sigma = 0$ and 0.02 the BER is equal to zero (not shown in the logarithmic plot). Compared to the real 1-D example, the BER increases more slowly simply because there is more "room" in 2-D space for the clusters to be effectively separated.

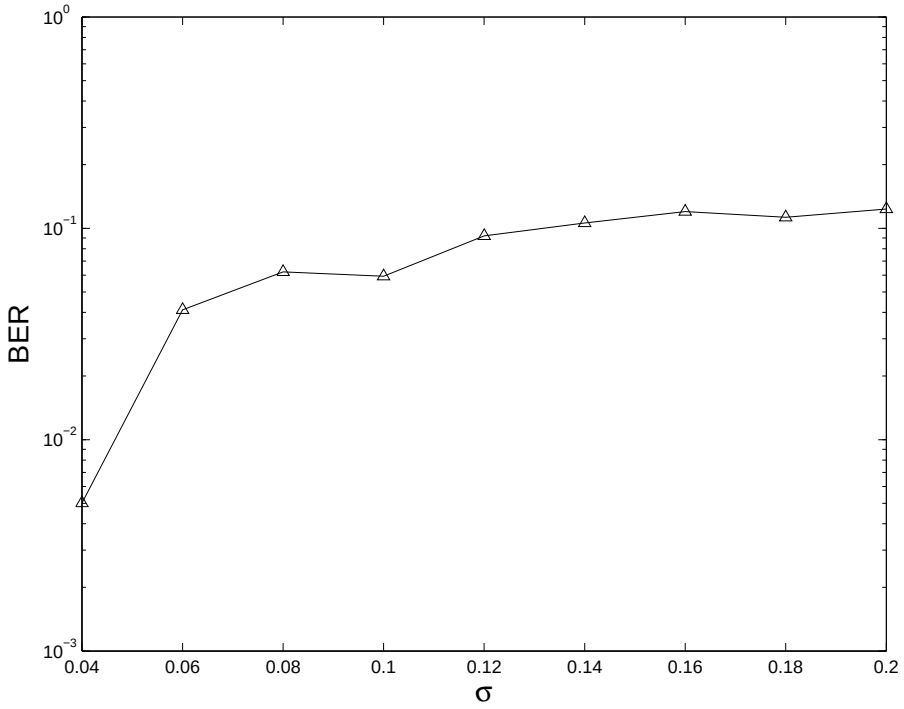


Fig. 8. Average Bit Error Rate of 3 binary inputs separated from a complex mixture with observation noise variance σ. Average values from 100 Monte Carlo simulations per noise level.

6 Conclusions

In this paper we introduced two blind methods for separating multiple finite alphabet sources from a single real or complex mixture. The problem appears in a variety of applications, including digital communications. Both methods are deterministic in nature and are based on the relation between the position of the cluster centers of the observation data and the mixing parameters. The first method is 1-dimensional and it applies to a real mixture of binary antipodal sources whereas the second method is 2-dimensional and it applies to complex mixtures of multilevel PAM sources. The source symbols are estimated in a second step, following the estimation of the mixing operator, using a nearest neighbor rule. In the absence of noise both methods produce perfect estimates of the mixing system and the source signals, for any number of sources. In the noisy case the system estimation deteriorates smoothly with the increase of the noise variance. The bit error rate of the sources estimates increases quickly as the noise variance approaches the minimum distance between the clusters.

References

[1] K. Anand, G. Mathew, and V. U. Reddy, "Blind Separation of Multiple Co-Channel BPSK Signals Arriving at an Antenna Array", *IEEE Signal Processing Letters*, vol. 2, pp. 176–178, Sept. 1995.

[2] A. Bell and T. Sejnowski, "Learning the Higher Order Structure of a Natural Sound", *Network: Comput. Neural Syst.*, vol. 7, pp. 261–267, 1996.

[3] A. Bell and T. Sejnowski, "Edges are the Independent Components of Natural Scenes", in *Advances in Neural Information Processing Systems 9*, MIT Press, Cambridge, MA, 1997.

[4] A. Belouchrani, and J.-F. Cardoso, "Maximum Likelihood Source Separation for Discrete Sources", *Proc. EUSIPCO'94*, pp. 768–771, Edinburgh, Scotland, Elsevier, Sept. 1994.

[5] 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.

[6] J.-F. Cardoso, 'Blind Signal Separation: Statistical Principles", *Proc. IEEE*, vol. 86, pp. 2009–2025, Oct. 1998.

[7] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*, John Wiley, Chichester, 2002.

[8] P. Comon, "Independent Component Analysis, A New Concept?", *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[9] K. I. Diamantaras and E. Chassioti, "Blind Separation of n Binary Sources from one Observation: A Deterministic Approach", in *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Source Separation (ICA-2000)*, pp. 93-98, Helsinki, Finland, June 2000.

[10] K. I. 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.

[11] K. I. Diamantaras, "Blind channel identification based on the geometry of the received signal constellation", *IEEE Transactions on Signal Processing*, pp. 1133-1143, vol. 50, no. 5, May 2002.

[12] 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*, vol. 39, pp. 1–38, 1977.

[13] R.O. Duda and P.E. Hart, *Patetrn Classification and Scene Analysis*, John Wiley, NY, 1973.

[14] Z. Ghahramani and M. Jordan, "Supervised Learning from Incomplete Data via an EM Approach", in *Advances in Neural Information Processing Systems 6*, J.D. Cowan, G. Tesauro, and J. Alspector (eds.), pp. 120-127, 1994.

[15] O. Grellier and P. Comon, "Blind Separation of Discrete Sources", *IEEE Signal Proc. Letters*, vol. 5, no. 8, pp. 212-214, August 1998.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*, IEEE Press, 1994.

[17] M. Hermann and H. H. Yang, "Perspectives and Limitations of Self Organizing Maps in Blind Separation of Source Signals", *Proc. ICONIP'96*, 1996.

[18] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley, NY, 2001.

[19] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive Mixture of Local Experts", *Neural Computation*, vol. 3, pp. 79–87, 1991.

[20] M. Jordan and R. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm", *Neural Computation*, vol. 6, pp. 181–214, 1994.

[21] T.-P. Jung, C. Humphries, T.-W. Lee, S. Makeig, M. J. McKeown, V. Iragui, and T. Sejnowski, "Removing Electroencephalographic Artifacts: Comparison between ICA and PCA", *Proc. IEEE Workshop on Neural Networks for Signal Processing VIII*, pp. 63–72, IEEE Press, 1998.

[22] A. Kannan and V. U. Reddy, "Maximum Likelihood Estimation of Constellation Vectors for Blind Separation of Co-Channel BPSK Signals and its Performance Analysis", *IEEE Trans. Signal Processing*, vol. 45, no. 7, pp. 1736–1741, July 1997.

[23] J. Karhunen, A. Hyvärinen, R. Vigário, J. Hurri, and E. Oja, "Applications of Neural Blind Separation to Signal and Image Processing", in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP'97)*, Munich, Germany, 1997.

[24] T.-W. Lee, M.S. Lewicki, M. Girolami, and T.J. Sejnowski, "Blind Source Separation of More Sources than Mixtures Using Overcomplete Representations", *IEEE Singal Processing Letters*, vol. 6, no. 4, April 1999.

[25] Y. Li, A. Cichocki, and L. Zhang, "Blind Separation and Extraction of Binary Sources", IEICE Trans. on Fundamentals, vol. E86-A, no. 3, March 2003.

[26] Y. Li, A. Cichocki, and L. Zhang, "Blind Source Estimation of FIR Channels for Binary Sources: A Grouping Decision Approach", Signal Processing, vol. 84, pp. 2245-2263, 2004.

[27] R.-W. Liu, "Blind Signal Processing: an Introduction", in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol 2, pp81–84, Atlanta GA, May 1996.

[28] D. Lowe, "What Have Neural Networks to Offer Statistical Pattern Processing?", in *Proc. SPIE Conf. Adaptive Signal Processing*, pp. 460–471, San Diego, CA, 1991.

[29] G.J. McLachlan and K.E. Basford, *Mixture Models: Inference and Applications to Clustering*, Marcel Dekker, 1988.

[30] P. Pajunen, "Blind Separation of Binary Sources with Less Sensors than Sources", in *Proc. of the 1997 Int. Conf. Neural Networks (ICNN-97)*, pp. 1994–1997, Houston, TX, USA, June 1997.

[31] P. Pajunen and J. Karhunen, "A Maximum Likelihood Approach to Nonlinear Blind Source Separation", *Proc. ICANN'97*, pp. 541–546, Lausanne, Switzerland, 1997.

[32] A. Paulraj and C. Papadias, "Space-Time Processing for Wireless Communications", *IEEE Signal Processing Magazine*, vol. 14, no. 6, pp. 49–83, Nov. 1997.

[33] L. Parra, G. Deco, and S. Miesbach, "Statistical Independence and Novelty Detection with Information Preserving Nonlinear Maps", *Neural Computation*, vol. 8, pp. 260–269, 1996.

[34] J. Proakis, *Digital Communications*, third edition, McGraw-Hill, NY, 1995.

[35] S. Talwar, M. Viberg, and A. Paulraj, "Blind Estimation of Synchronous Co-channel Digital Signals using an Antenna Array. Part I: Algorithms", *IEEE Trans. Signal Processing*, vol. 44, no. 5, pp. 1184–1197, May 1996.

[36] K. Torkkola, "Blind Separation of Convolved Sources Based on Information Maximization", in *Proc. IEEE Workshop on Neural Networks for Signal Processing VI*, pp. 423–432, IEEE Press, 1996.

[37] A.-J. van der Veen, "Analytical Method for Blind Binary Signal Separation", *IEEE Trans. Signal Processing*, vol. 45, no. 4, pp. 1078–1082, April 1997.

[38] S. Verdú, *Multiuser Detection*, Cambridge University Press, UK, 1998.

Appendix

A Proof of Lemma 1

It is obvious since

$$\mathbf{B}^{(2)} = \begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}$$

and so $\gamma_2 - \gamma_1 = 2\alpha_2 > 0$, $\gamma_3 - \gamma_2 = 2(\alpha_1 - \alpha_2) > 0$, $\gamma_4 - \gamma_3 = 2\alpha_2 > 0$.

B Proof of Lemma 2

By definition we have,

$$\begin{aligned}
\gamma_1 &= [-1 \cdots -1 \ -1 \ -1][\alpha_1 \cdots \alpha_{n-2} \ \alpha_{n-1} \ \alpha_n]^T \\
\gamma_2 &= [-1 \cdots -1 \ -1 \ +1][\alpha_1 \cdots \alpha_{n-2} \ \alpha_{n-1} \ \alpha_n]^T \\
\gamma_3 &= [-1 \cdots -1 \ +1 \ -1][\alpha_1 \cdots \alpha_{n-2} \ \alpha_{n-1} \ \alpha_n]^T
\end{aligned}$$

First we prove that γ_1 is the smallest element in the sequence $\{\gamma_i\}$. Indeed, $\gamma_1 = -\sum_{i=1}^{n} \alpha_i$ and so for all $j > 1$:

$$\gamma_j - \gamma_1 = \sum_{i=1}^{n} (t_{j,i} + 1)\alpha_i > 0$$

since $\alpha_i > 0$, $t_{j,i} \geq -1$ for all i, and $t_{j,m} = 1 > -1$ for at least one m.

Next we show that γ_2 is the second last element of the sequence $\{\gamma_i\}$. Clearly $\gamma_2 > \gamma_1$. We need then to show that $\gamma_j > \gamma_2$, for all $j > 2$. We have $\gamma_2 = -\sum_{i=1}^{n-1} \alpha_i + \alpha_n$, and so

$$\gamma_j - \gamma_2 = \sum_{i=1}^{n-1} (t_{j,i} + 1)\alpha_i + (t_{j,n} - 1)\alpha_n$$

The number $t_{j,n}$ is either equal to $+1$ or -1. If $t_{j,n} = 1$ then $(t_{j,n} - 1)\alpha_n = 0$ and $\gamma_j - \gamma_2 > 0$ by a similar argument as before. If, on the other hand, $t_{j,n} = -1$ and since $j > 2$, then $t_{j,m} = 1$ for some $m < n$ and

$$\gamma_j - \gamma_2 = \sum_{i<n,\, i\neq m} (t_{j,i} + 1)\alpha_i + 2(\alpha_m - \alpha_n)$$

Since the sequence $\{\alpha_i\}$ is arranged in decreasing order we have $(\alpha_m - \alpha_n) > 0$ and $\sum_{i<n,\, i\neq m} (t_{j,i} + 1)\alpha_j \geq 0$ so $\gamma_j - \gamma_2 > 0$.

Similarly, for γ_3 we have $\gamma_3 > \gamma_2 > \gamma_1$, and we need to prove that $\gamma_j - \gamma_3 > 0$ for all $j > 3$. By definition, $\gamma_3 = -\sum_{i\neq n-1} \alpha_i + \alpha_{n-1}$, so for all $j > 3$

$$\gamma_j - \gamma_3 = \sum_{i\neq n-1} (t_{j,i} + 1)\alpha_i + (t_{j,n-1} - 1)\alpha_{n-1}$$

As before, we treat the cases $t_{j,n-1} = 1$ and $t_{j,n-1} = -1$ separately. If $t_{j,n-1} = 1$ then $(t_{j,n-1} - 1)\alpha_{n-1} = 0$ and $\gamma_j - \gamma_i > 0$ since $t_{j,m} > -1$ for some m. On the

other hand, $t_{j,n-1} = -1$ and the fact that $j > 3$, implies that there exists at least one $m < n-1$ such that $t_{j,m} = 1$, so

$$\gamma_j - \gamma_3 = \sum_{i \neq n-1, i \neq m} (t_{j,i} + 1)\alpha_i + 2(\alpha_m - \alpha_{n-1}) > 0.$$

So $\gamma_1 < \gamma_2 < \gamma_3$ are the three smallest elements of the sequence $\{\gamma_i\}$. By the antisymmetric property $\gamma_i = -\gamma_{M-i+1}$, it follows that $\gamma_M > \gamma_{M-1} > \gamma_{M-2}$ are the three largest elements of the same sequence.

Finally, from the definitions of γ_1, γ_2, γ_3, it follows that

$$\begin{aligned}\gamma_2 - \gamma_1 &= 2\alpha_n \\ \gamma_3 - \gamma_1 &= 2\alpha_{n-1}\end{aligned}$$

C Proof of Lemma 3

For all $i = 0, \cdots, 2^{n-2} - 1$ we have

$$\begin{bmatrix} \gamma_{4i+1} \\ \gamma_{4i+2} \\ \gamma_{4i+3} \\ \gamma_{4i+4} \end{bmatrix} = \begin{bmatrix} b_1 \cdots b_{n-2} & -1 & -1 \\ b_1 \cdots b_{n-2} & -1 & 1 \\ b_1 \cdots b_{n-2} & 1 & -1 \\ b_1 \cdots b_{n-2} & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{n-2} \\ \alpha_{n-1} \\ \alpha_n \end{bmatrix}$$

where $b_1 \cdots b_{n-2}$, corresponds to the binary antipodal representation of i.

Defining $B = [b_1, \cdots, b_{n-2}][\alpha_1, \cdots, \alpha_{n-2}]^T$, we can rewrite the above equation as follows

$$\begin{bmatrix} \gamma_{4i+1} - B \\ \gamma_{4i+2} - B \\ \gamma_{4i+3} - B \\ \gamma_{4i+4} - B \end{bmatrix} = \mathbf{B}^{(2)} \begin{bmatrix} \alpha_{n-1} \\ \alpha_n \end{bmatrix}$$

Then part (a) of the Lemma follows immediately using Lemma 1 while part (b) is easily shown using the structure of $\mathbf{B}^{(2)}$.