Median-Based Clustering for Underdetermined Blind Signal Processing

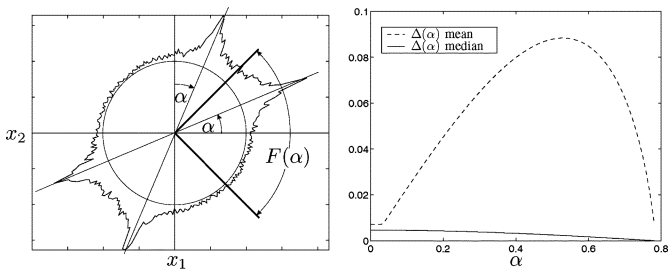
Fabian J. Theis, *Member, IEEE*, Carlos G. Puntonet, *Member, IEEE*, and Elmar W. Lang

***Abstract*—In underdetermined blind source separation, more sources are to be extracted from less observed mixtures without knowing both sources and mixing matrix. k-means-style clustering algorithms are commonly used to do this algorithmically given sufficiently sparse sources, but in any case other than deterministic sources, this lacks theoretical justification. After establishing that mean-based algorithms converge to wrong solutions in practice, we propose a median-based clustering scheme. Theoretical justification as well as algorithmic realizations (both online and batch) are given and illustrated by some examples.**

***Index Terms*—Blind source separation (BSS), independent component analysis (ICA).**

BLIND source separation (BSS), mainly based on the assumption of independent sources, is currently the topic of many researchers [1], [2]. Given an observed m-dimensional mixture random vector $\mathbf{x}$, which allows an unknown decomposition $\mathbf{x} = \mathbf{A}\mathbf{s}$, the goal is to identify the mixing matrix $\mathbf{A}$ and the unknown n-dimensional source random vector $\mathbf{s}$. Commonly, first $\mathbf{A}$ is identified, and only then are the sources recovered. We will therefore denote the former task by *blind mixing model recovery (BMMR)* and the latter (with known $\mathbf{A}$) by *blind source recovery (BSR)*.

In the difficult case of *underdetermined* or *overcomplete* BSS, where fewer mixtures than sources are observed $(m < n)$, BSR is nontrivial (see Section II). However, our main focus lies on the usually more elaborate matrix recovery. Assuming statistically independent sources with existing variance and at most one Gaussian component, it is well known that $\mathbf{A}$ is determined uniquely by $\mathbf{x}$ [3]. However, how to do this algorithmically is far from obvious, and although quite a few algorithms have been proposed recently [4]–[6], performance is yet limited. The most commonly used overcomplete algorithms rely on sparse sources (after possible sparsification by preprocessing), which can be identified by clustering, usually by k-means or some extension [5], [6]. However, apart from the fact that theoretical justifications have not been found, mean-based clustering only identifies the correct $\mathbf{A}$ if the data density approaches a delta distribution. In Fig. 1, we illustrate the deficiency of *mean*-based clustering; we get an error of up to 5° per mixing angle, which is rather substantial considering the sparse density and the simple, complete case of $m = n = 2$. Moreover, the figure indicates that *median*-based clustering performs much better. Indeed, mean-based clustering does not possess any equivariance property (performance independent of $\mathbf{A}$). In the following, we propose a novel median-based clustering method and prove its equivariance (Lemma 1.2) and convergence. For brevity, the proofs are given for the case of arbitrary n, but $m = 2$, although they can be readily extended to higher sensor signal dimensions. Corresponding algorithms are proposed and experimentally validated.



(a) circle histogram $\alpha = 0.4$ (b) comparison of mean and median

Fig. 1. Mean- versus median-based clustering. We consider the mixture $\mathbf{x}$ of two independent gamma-distributed sources ($\gamma = 0.5$, 10^5 samples) using a mixing matrix $\mathbf{A}$ with columns inclined by α and $(\pi/2 - \alpha)$, respectively. (a) Mixture density for $\alpha = 0.4$ after projection onto the circle. (b) For $\alpha \in [0, \pi/4)$, we compare the error when estimating $\mathbf{A}$ by the mean and the median of the projected density in the receptive field $F(\alpha) = (-\pi/4, \pi/4)$ of the known column $\mathbf{a}_1$ of $\mathbf{A}$. The former is the k-means convergence criterion.

I. Geometric Matrix Recovery

Without loss of generality, we assume that $\mathbf{A}$ has pairwise linearly independent columns, and $m \leq n$. BMMR tries to identify $\mathbf{A}$ in $\mathbf{x} = \mathbf{A}\mathbf{s}$ given $\mathbf{x}$, where $\mathbf{s}$ is assumed to be statistically independent. Obviously, this can only be done up to equivalence [3], where $\mathbf{B}$ is said to be *equivalent* to $\mathbf{A}$, $\mathbf{B} \sim \mathbf{A}$, if $\mathbf{B}$ can be written as $\mathbf{B} = \mathbf{A}\mathbf{P}\mathbf{L}$ with an invertible diagonal matrix $\mathbf{L}$ (scaling matrix) and an invertible matrix $\mathbf{P}$ with unit vectors in each row (permutation matrix). Hence, we may assume the columns $\mathbf{a}_i$ of $\mathbf{A}$ to have unit norm.

For geometric matrix-recovery, we use a generalization [7] of the geometric independent component analysis (ICA) algorithm [8]. Let $\mathbf{s}$ be an independent n-dimensional, Lebesgue-continuous, random vector with density $p_{\mathbf{s}}$ describing the sources. As $\mathbf{s}$ is independent, $p_{\mathbf{s}}$ factorizes into $p_{\mathbf{s}}(s_1, \ldots, s_n) = p_{s_1}(s_1) \cdots p_{s_n}(s_n)$ with the one-dimensional marginal source density functions p_{s_i}. We assume symmetric sources, i.e., $p_{s_i}(s) = p_{s_i}(-s)$ for $s \in \mathbb{R}$ and $i \in [1:n] := \{1, \ldots, n\}$, in particular $\mathbf{E}(\mathbf{s}) = \mathbf{0}$.

The *geometric blind mixing model recovery (BMMR) algorithm* for symmetric distributions goes as follows [7]: Pick $2n$ starting vectors $\mathbf{w}_1, \mathbf{w}'_1, \ldots, \mathbf{w}_n, \mathbf{w}'_n$ on the unit sphere

Manuscript received July 14, 2005; revised September 3, 2005. Some preliminary results were reported at the conferences ESANN 2002, SIP 2002, and ICA 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yue (Joseph) Wang.

F. J. Theis and E. W. Lang are with the Institute of Biophysics, University of Regensburg, 93040 Regensburg, Germany (e-mail: fabian@theis.name).

C. G. Puntonet is with the Dept. Arqitectura y Tecnología de Computadores, Universidad de Granada, 18071 Granada, Spain.

Digital Object Identifier 10.1109/LSP.2005.861590

1070-9908/$20.00 © 2006 IEEE

$S^{m-1} \subset \mathbb{R}^m$ such that the $\mathbf{w}_i$ are pairwise linearly independent and $\mathbf{w}_i = -\mathbf{w}_i'$. Often, these $\mathbf{w}_i$ are called *neurons* because they resemble the neurons used in clustering algorithms and in Kohonen's self-organizing maps. Furthermore, fix a learning rate $\eta : \mathbb{N} \to \mathbb{R}$. The usual hypothesis in competitive learning is $\eta(t) > 0$, $\sum_{t\in\mathbb{N}} \eta(t) = \infty$, and $\sum_{t\in\mathbb{N}} \eta(t)^2 < \infty$. Then iterate the following steps until an appropriate abort condition has been met: Choose a sample $\mathbf{x}(t) \in \mathbb{R}^m$ according to the distribution of $\mathbf{x}$. If $\mathbf{x}(t) = 0$, pick a new one—note that this case happens with probability zero. Project $\mathbf{x}(t)$ onto the unit sphere, and get $\mathbf{y}(t) := \boldsymbol{\pi}(\mathbf{x}(t))$, where $\boldsymbol{\pi}(\mathbf{x}) := \mathbf{x}/|\mathbf{x}| \in S^{m-1}$. Let $i(t) \in [1 : n]$ such that $\mathbf{w}_{i(t)}(t)$ or $\mathbf{w}'_{i(t)}(t)$ is the neuron closest to $\mathbf{y}(t)$. Then set $\mathbf{w}_{i(t)}(t+1) := \boldsymbol{\pi}(\mathbf{w}_{i(t)}(t) + \eta(t)\boldsymbol{\pi}(\sigma\mathbf{y}(t) - \mathbf{w}_{i(t)}(t)))$ and $\mathbf{w}'_{i(t)}(t+1) := -\mathbf{w}_{i(t)}(t+1)$, where $\sigma := 1$ if $\mathbf{w}_{i(t)}(t)$ is closest to $\mathbf{y}(t)$, $\sigma := -1$ otherwise. All other neurons are not moved in this iteration. This update rule equals online k-means on S^{n-1}, except for the fact that the winner neuron is not moved proportional to the sample but only in its direction due to the normalization. We will see that instead of finding the mean in the receptive field (as in k-means), the algorithm searches for the corresponding median.

A. Model Verification

In this section, we first calculate the densities of the random variables of our clustering problem. Then we will prove an asymptotic convergence result. For the theoretical analysis, we will restrict ourselves to $m = 2$ mixtures for simplicity. As above, let $\mathbf{x}$ denote the sensor signal vector and $\mathbf{A}$ the mixing matrix such that $\mathbf{x} = \mathbf{A}\mathbf{s}$. We may assume that $\mathbf{A}$ has columns $\mathbf{a}_i = (\cos\alpha_i, \sin\alpha_i)^\top$ with $0 \le \alpha_1 <, \ldots, < \alpha_n < \pi$.

1) Neural Update Rule on the Sphere: Due to the symmetry of $\mathbf{s}$, we can identify the two neurons $\mathbf{w}_i$ and $\mathbf{w}_i'$. For this, let $\iota(\varphi) := (\varphi \bmod \pi) \in [0, \pi)$ identify all angles modulo π, and set $\theta(\mathbf{v}) := \iota(\arctan(v_2/v_1))$; then $\theta(\mathbf{w}_i) = \theta(\mathbf{w}_i')$ and $\theta(\mathbf{a}_i) = \alpha_i$. We are interested in the essentially one-dimensional projected sensor signal random vector $\boldsymbol{\pi}(\mathbf{x})$, so using θ, we may approximate $y := \theta(\boldsymbol{\pi}(\mathbf{x})) \in [0, \pi)$ measuring the argument of $\mathbf{x}$. Note that the density p_y of y can be calculated from $p_\mathbf{x}$ by $p_y(\varphi) = \int_{-\infty}^{\infty} p_\mathbf{x}(r\cos\varphi, r\sin\varphi) r dr$. Now let the $(n \times n)$-matrix $\mathbf{B}$ be defined by

$$\mathbf{B} := \begin{array}{|c|c|}\hline \multicolumn{2}{|c|}{\mathbf{A}} \\ \hline \mathbf{0} & \mathbf{I}_{n-2} \\ \hline \end{array}$$

where $\mathbf{I}_{n-2}$ is the $(n-2)$-dimensional identity matrix; so, $\mathbf{B}$ is invertible. The random vector $\mathbf{B}\mathbf{s}$ has the density $p_{\mathbf{B}\mathbf{s}} = |\det \mathbf{B}|^{-1} p_\mathbf{s} \circ \mathbf{B}^{-1}$. $\mathbf{A}$ equals $\mathbf{B}$, followed by the projection from $\mathbb{R}^n$ onto the first two coordinates; hence

$$p_y(\varphi) = \frac{2}{|\det \mathbf{B}|} \int_0^\infty r dr \int_{\mathbb{R}^{n-2}} d\mathbf{x} p_\mathbf{s}\big(\mathbf{B}^{-1}(r\cos\varphi, r\sin\varphi, \mathbf{x})^\top\big) \tag{1}$$

for any $\varphi \in [0, \pi)$, where we have used the symmetry of $p_\mathbf{s}$.

The geometric learning algorithm induces the following n-dimensional Markov chain $\boldsymbol{\omega}(t)$ defined recursively by a starting point $\boldsymbol{\omega}(0) \in \mathbb{R}^n$ and the iteration rule $\boldsymbol{\omega}(t+1) = \iota^n(\boldsymbol{\omega}(t) + \eta(t)\boldsymbol{\zeta}(y(t)\mathbf{e} - \boldsymbol{\omega}(t)))$, where $\mathbf{e} = (1, \ldots, 1)^\top \in \mathbb{R}^n$, $\boldsymbol{\zeta}(x_1, \ldots, x_n) \in \mathbb{R}^n$ such that

$$\zeta_i(x_1, \ldots, x_n) = \begin{cases} \operatorname{sgn}(x_i), & |x_i| \le |x_j|, \text{ for all } j \\ 0, & \text{otherwise} \end{cases}$$

and $y(0), y(1), \ldots$ is a sequence of i.i.d. random variables with density p_y representing the samples in each online iteration. Note that the "modulo π" map ι is only needed to guarantee that each component of $\boldsymbol{\omega}(t+1)$ lies in $[0, \pi)$.

Furthermore, we can assume that after enough iterations, there is one point $\mathbf{v} \in S^1$ that will not be traversed any more, and without loss of generality, we assume $\theta(\mathbf{v}) = 0$ so that the above algorithm simplifies to the planar case with the recursion rule $\boldsymbol{\omega}(t+1) = \boldsymbol{\omega}(t) + \eta(t)\boldsymbol{\zeta}(y(t)\mathbf{e} - \boldsymbol{\omega}(t))$. This is k-means-type learning with an additional sign function. Without the sign function and the additional condition that p_y is log-concave, it has been shown [9] that the process $\boldsymbol{\omega}(t)$ converges to a unique constant process $\boldsymbol{\omega}(\infty) \in \mathbb{R}^n$ such that $\omega_i(\infty) = E(p_y|[\beta(i), \beta'(i)])$, where $F(\omega_i(\infty)) := \{\varphi \in [0, \pi) | \iota(|\varphi - \omega_i(\infty)|) \le \iota(|\varphi - \omega_j(\infty)|)$, for all $j \ne i\}$ denotes the *receptive field* of the neuron $\omega_i(\infty)$, and $\beta(i), \beta'(i)$ designate the receptive field borders. This is precisely the k-means convergence condition illustrated in Fig. 1.

2) Limit Points Analysis: We now want to study the limit points of geometric matrix-recovery, so we assume that the algorithm has already converged. The idea, generalizing our analysis in the complete case [7], then is to formulate a condition that the limit points will have to satisfy and to show that the BMMR solutions are among them.

The angles $\gamma_1, \ldots, \gamma_n \in [0, \pi)$ are said to satisfy the *geometric convergence condition (GCC)* if they are the medians of y restricted to their receptive fields, i.e., if γ_i is the median of $p_y|F(\gamma_i)$. Moreover, a constant random vector $\hat{\boldsymbol{\omega}} \in \mathbb{R}^n$ is called *fixed point* if $\mathbf{E}(\boldsymbol{\zeta}(y\mathbf{e} - \hat{\boldsymbol{\omega}})) = 0$. Hence, the expectation of a Markov process $\boldsymbol{\omega}(t)$ starting at a fixed point will indeed not be changed by the geometric update rule because $\mathbf{E}(\boldsymbol{\omega}(t+1)) = \mathbf{E}(\boldsymbol{\omega}(t)) + \eta(t)\mathbf{E}(\boldsymbol{\zeta}(y(t)\mathbf{e} - \boldsymbol{\omega}(t))) = \mathbf{E}(\boldsymbol{\omega}(t))$.

Lemma 1.1: Assume that the geometric algorithm converges to a constant random vector $\boldsymbol{\omega}(\infty)$. Then $\boldsymbol{\omega}(\infty)$ is a fixed point if and only if the $\omega_i(\infty)$ satisfy the GCC.

Proof: Assume $\boldsymbol{\omega}(\infty)$ is a fixed point of geometric ICA in the expectation. Without loss of generality, let $[\beta, \beta']$ be the receptive field of $\omega_i(\infty)$ such that $\beta, \beta' \in [0, \pi)$. Since $\boldsymbol{\omega}(\infty)$ is a fixed point of geometric ICA in the expectation, $E(\chi_{[\beta,\beta']}(y(t))\operatorname{sgn}(y(t) - \omega_i(\infty))) = 0$, where $\chi_{[\beta,\beta']}$ denotes the characteristic function of that interval. However, this means $\int_\beta^{\omega_i(\infty)} (-1) p_y(\varphi) d\varphi + \int_{\omega_i(\infty)}^{\beta'} 1 p_y(\varphi) d\varphi = 0$, so $\omega_i(\infty)$ satisfies GCC. The other direction follows similarly. ∎

Lemma 1.2: The angles $\alpha_i = \theta(\mathbf{a}_i)$ satisfy the GCC.

Proof: It is enough to show that α_1 satisfies GCC. Let $\beta := (\alpha_1 + \alpha_n - \pi)/2$ and $\beta' := (\alpha_1 + \alpha_2)/2$. Then the receptive field of α_1 can be written (modulo π) as $F(\alpha_1) = [\beta, \beta']$. Therefore, we have to show that α_1 is the median of $p_y|F(\alpha_1)$, which means $\int_\beta^{\alpha_1} p_y(\varphi) d\varphi = \int_{\alpha_1}^{\beta'} p_y(\varphi) d\varphi$. Using (1), the left-hand side can be expanded as $\int_\beta^{\alpha_1} p_y(\varphi) d\varphi = 2|\det \mathbf{B}|^{-1} \int_{K'} d\mathbf{x} p_\mathbf{s}(\mathbf{B}^{-1}\mathbf{x})$, where $K := \theta^{-1}[\beta, \alpha_1]$ denotes the cone of opening angle $\alpha_1 - \beta$ starting from angle β, and $K' = K \times \mathbb{R}^{n-2}$. This implies $\int_\beta^{\alpha_1} p_y(\varphi) d\varphi = 2\int_{\mathbf{B}^{-1}(K')} d\mathbf{s} p_\mathbf{s}(\mathbf{s})$. Now note that the transformed extended cone $\mathbf{B}^{-1}(K')$ is a cone ending at the s_1-axis of opening angle $\pi/4$ times $\mathbb{R}^{n-2}$, because $\mathbf{A}$ is linear. Hence, $\int_\beta^{\alpha_1} p_y(\varphi) d\varphi = 2\int_0^\infty ds_1 \int_0^{s_1} ds_2 \int_{\mathbb{R}^{n-2}} ds_3, \ldots, ds_n p_\mathbf{s}(\mathbf{s}) = \int_{\alpha_1}^{\beta'} p_y(\varphi) d\varphi$, where we have used the same calculation for $[\alpha_1, \beta']$ as for $[\beta, \alpha_1]$ at the last step. ∎

In the proof, we show that the median-condition is *equivariant*, meaning that it does no depend on $\mathbf{A}$. Hence, any algorithm based on such a condition is equivariant, as confirmed by Fig. 1. Set $\boldsymbol{\xi}(\omega) := (\cos\omega, \sin\omega)^\top$; then, $\theta(\boldsymbol{\xi}(\omega)) = \omega$. Combining both lemmata, we have therefore shown the following.

Theorem 1.3: The set Φ of fixed points of geometric matrix-recovery contains an element $(\hat{\omega}_1, \ldots, \hat{\omega}_n)$ such that the matrix $(\boldsymbol{\xi}(\hat{\omega}_1) \ldots \boldsymbol{\xi}(\hat{\omega}_n))$ solves the overcomplete BMMR.

The stable fixed points in the above set Φ can be found by the geometric matrix-recovery algorithm. Furthermore, experiments confirm that in the special case of unimodal, symmetric, and non-Gaussian signals, the set Φ consists of only two elements: a stable and an unstable fixed point, where the stable fixed point will be found by the algorithm. Then, depending on the kurtosis of the sources, either the stable (super-Gaussian case) or the instable (sub-Gaussian case) fixed point represents the image of the unit vectors. We have partially shown this in the complete case (see [7, Theorem 5]).

Theorem 1.4: If $n = 2$ and $p_{s_1} = p_{s_2}$, then Φ contains only two elements given that $p_y|[0, \pi)$ has exactly four local extrema.

More elaborate studies of Φ are necessary to show full convergence; however, the mathematics can be expected to be difficult. This can already be seen from the complicated and in higher dimensions yet unknown proofs of convergence of the related self-organizing-map algorithm [9].

B. Turning the Online Algorithm Into a Batch Algorithm

The above theory can be used to derive a batch-type learning algorithm, by testing all different receptive fields for the overcomplete GCC after histogram-based estimation of y. For simplicity, let us assume that the cumulative distribution P_y of y is invertible. For $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_{n-1})$, define a function $\boldsymbol{\mu}(\boldsymbol{\varphi}) := ((\gamma_1(\boldsymbol{\varphi}) + \gamma_2(\boldsymbol{\varphi}))/2 - \varphi_1, \ldots, (\gamma_n(\boldsymbol{\varphi}) + \gamma_1(\boldsymbol{\varphi}))/2 - \varphi_n)$, where $\gamma_i(\boldsymbol{\varphi}) := P_y^{-1}((P_y(\varphi_i) + P_y(\varphi_{i+1})/2)$ is the median of $y|[\varphi_i, \varphi_{i+1}]$ in $[\varphi_i, \varphi_{i+1}]$ for $i \in [1 : n]$ and $\varphi_n := (\varphi_{n-1} + \varphi_1 + \pi)/2$, $\varphi_{n+1} := \varphi_1$.

Lemma 1.5: If $\boldsymbol{\mu}(\boldsymbol{\varphi}) = \mathbf{0}$, then the $\gamma_i(\boldsymbol{\varphi})$ satisfy the GCC.

Proof: By definition, the receptive field of $\gamma_i(\boldsymbol{\varphi})$ is given by $[(\gamma_{i-1}(\boldsymbol{\varphi}) + \gamma_i(\boldsymbol{\varphi}))/2, (\gamma_i(\boldsymbol{\varphi}) + \gamma_{i+1}(\boldsymbol{\varphi}))/2]$. Since $\boldsymbol{\mu}(\boldsymbol{\varphi}) = \mathbf{0}$ implies $\gamma_i(\boldsymbol{\varphi}) + \gamma_{i+1}(\boldsymbol{\varphi}))/2 = \varphi_i$, the receptive field of $\gamma_i(\boldsymbol{\varphi})$ is precisely $[\varphi_i, \varphi_{i+1}]$, and by construction, $\gamma_i(\boldsymbol{\varphi})$ is the median of y restricted to the above interval. ∎

Algorithm (overcomplete FastGeo): Find the zeros of $\boldsymbol{\mu}(\boldsymbol{\varphi})$.

Algorithmically, we may simply estimate y using a histogram and search for the zeros exhaustively or by discrete gradient descent. Note that for $m = 2$, this is precisely the complete FastGeo algorithm [7]. Again, $\boldsymbol{\mu}$ always has at least two zeros representing the stable and the unstable fixed point of the neural algorithm, so for super-Gaussian sources, we extract the stable fixed point by maximizing $\sum_{i=1}^{n} p_y(\gamma_i(\varphi_i))$.

Similar to the complete case, histogram-based density approximation results in a quite "ragged" distribution. Hence, zeros of $\boldsymbol{\mu}$ are split up into multiple close-by zeros. This can be improved by smoothing the distribution using a kernel with sufficiently small radius. Too large kernel radii result in lower accuracy because the calculation of the median is only roughly independent of the kernel radius. In Fig. 2, we use a polynomial kernel of radius 5°(zero everywhere else); one can see that indeed this smooths the distribution nicely.

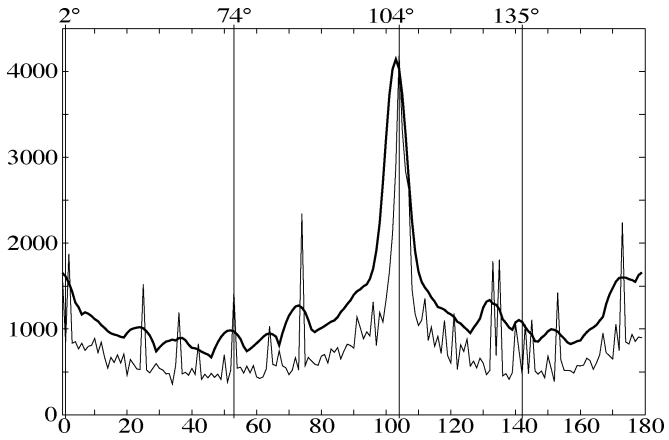


Fig. 2. Approximated probability density function of a mixture of four speech signals using the mixing angles $\alpha_i = 2°, 74°, 104°, 135°$. Plotted is the approximated density using a histogram with 180 bins; the thick line indicates the density after smoothing with a 5° radius polynomial kernel.

C. Clustering in Higher Mixture Dimensions

We now extend any BSS algorithm working in lower mixing dimension $m' < m$ to dimension m using the simple idea of projecting the mixtures $\mathbf{x}$ onto different subspaces and then estimating $\mathbf{A}$ from the recovered projected matrices. We eliminate scaling indeterminacies by normalization and permutation indeterminacies by comparing the source correlation matrices.

1) Equivalence After Projections: Let $m' \in \mathbb{N}$ with $1 < m' < m$, and let M denote the set of all subsets of $[1 : m]$ of size m'. For an element $\tau \in M$, let $\tau = \{\tau_1, \ldots, \tau_{m'}\}$ such that $1 \leq \tau_1 <, \ldots, < \tau_{m'} \leq m$ and let $\boldsymbol{\pi}_\tau$ denote the ordered projection from $\mathbb{R}^m$ onto those coordinates. Consider the projected mixing matrix $\mathbf{A}_\tau := \boldsymbol{\pi}_\tau \mathbf{A}$. We will study how scaling-equivalence behaves under projection, where two matrices $\mathbf{A}, \mathbf{B}$ are said to be scaling equivalent, $\mathbf{A} \sim_s \mathbf{B}$ if there exists an invertible diagonal matrix $\mathbf{L}$ with $\mathbf{A} = \mathbf{BL}$.

Lemma 1.6: Let $\tau^1, \ldots, \tau^k \in M$ such that $\bigcup_i \tau^i = [1 : m]$ and $j \in \bigcap_i \tau^i$. If $\mathbf{A}_{\tau^i} \sim_s \mathbf{B}_{\tau^i}$ for all i and $a_{jl} \neq 0$ for all l, then $\mathbf{A} \sim_s \mathbf{B}$.

Proof: Fix a column $l \in [1 : n]$, and let $\mathbf{a} := \mathbf{a}_l$, $\mathbf{b} := \mathbf{b}_l$. By assumption, for each $i \in [1 : k]$, there exist $\lambda_i \neq 0$ such that $\mathbf{b}_{\tau^i} = \lambda_i \mathbf{a}_{\tau^i}$. Index j occurs in all projections, so $b_j = \lambda_i a_j$ for all i. Hence, all $\lambda_i =: \lambda$ coincide and $\mathbf{b} = \lambda \mathbf{a}$. ∎

This lemma gives the general idea how to combine matrices; now, we will construct specific projections. Assume that the first row of $\mathbf{A}$ does not contain any zeros. This is a very mild assumption because $\mathbf{A}$ was assumed to be full rank, and the set of $\mathbf{A}$'s with first row without zeros is dense in the set of full-rank matrices. As usual, let $\lceil \lambda \rceil$ denote the smallest integer larger or equal to $\lambda \in \mathbb{R}$. Then let $k := \lceil (m-1)/(m'-1) \rceil$, and define $\tau^i := \{1, 2 + (m'-1)(i-1), \ldots, 2 + (m'-1)i - 1\}$ for $i < k$ and $\tau^k := \{1, m - m' + 2, \ldots, m\}$. Then $\bigcup_i \tau^i = [1 : m]$ and $1 \in \bigcap_i \tau^i$. Given $(m' \times n)$-matrices $\mathbf{B}^1, \ldots, \mathbf{B}^k := (b^k_{i,j})$, define $\mathbf{A}_{\mathbf{B}^1, \ldots, \mathbf{B}^k}$ to be composed of the columns $(j \in [1 : n])$

$$\mathbf{a}_j := \left(1, \frac{b^1_{2,j}}{b^1_{1,j}}, \ldots, \frac{b^1_{m',j}}{b^1_{1,j}}, \quad \frac{b^2_{2,j}}{b^2_{1,j}}, \ldots, \frac{b^{k-1}_{m',j}}{b^{k-1}_{1,j}}, \quad \frac{b^k_{3+k(m'-1)-m,j}}{b^k_{1,j}}, \ldots, \frac{b^k_{m',j}}{b^k_{1,j}}\right)^\top.$$

Lemma 1.7: Let $\mathbf{B}^1, \dots, \mathbf{B}^k$ be $(m' \times n)$-matrices such that $\mathbf{A}_{\tau^i} \sim_s \mathbf{B}^i$ for $i \in [1:k]$. Then $\mathbf{A}_{\mathbf{B}^1,\dots,\mathbf{B}^k} \sim_s \mathbf{A}$.

Proof: By assumption, there exist $\lambda_l^i \in \mathbb{R} \setminus \{0\}$ such that $b_{j,l}^i = \lambda_l^i (\mathbf{A}_{\tau^i})_{j,l}$ for all $i \in [1:k], j \in [1:m']$ and $l \in [1:n]$; hence, $b_{j,l}^i / b_{1,l}^i = (\mathbf{A}_{\tau^i})_{j,l} / (\mathbf{A}_{\tau^i})_{1,l}$. One can check that due to the choice of the τ^i's, we then have $(\mathbf{A}_{\mathbf{B}^1,\dots,\mathbf{B}^k})_{j,l} = a_{j,l}/a_{1,l}$ for all $j \in [1:m]$, and therefore, $\mathbf{A}_{\mathbf{B}^1,\dots,\mathbf{B}^k} \sim \mathbf{A}$. ■

2) Reduction Algorithm: The dimension reduction algorithm now is very simple. Pick k and $\tau^1, \dots, \tau^k$ as in the previous section. Perform overcomplete BMMR with the projected mixtures $\boldsymbol{\pi}_{\tau^i}(\mathbf{x})$ for $i \in [1:k]$ and get estimated mixing matrices $\mathbf{B}^i$. If this recovery has been carried out without any error, then every $\mathbf{B}^i$ is equivalent to $\mathbf{A}_{\tau^i}$. Due to permutations, they might however not be scaling-equivalent. Therefore, do the following iteratively for each $i \in [1:k]$: Apply the overcomplete source-recovery (see the next section) to $\boldsymbol{\pi}_{\tau^i}(\mathbf{x})$ using $\mathbf{B}^i$ and get recovered sources $\mathbf{s}^i$. For all $j < i$, consider the absolute cross-correlation matrices $(|\mathrm{Cor}(s_r^i, s_s^j)|)_{r,s}$. The row positions of the maxima of this matrix are pairwise different because the original sources were chosen to be independent. Thereby, we get a permutation matrix $\mathbf{P}^i$ indicating how to permute $\mathbf{B}^i$, $\mathbf{C}^i := \mathbf{B}^i \mathbf{P}^i$, so that the new source correlation matrices are diagonal. Finally, we have constructed matrices $\mathbf{C}^i$ such that there exists a permutation $\mathbf{P}$ independent of i with $\mathbf{C}^i \sim_s \mathbf{A}_{\tau^i}\mathbf{P}$ for all $i \in [1:k]$. Now we can apply Lemma 1.7 and get a matrix $\mathbf{A}_{\mathbf{C}^1,\dots,\mathbf{C}^k}$ with $\mathbf{A}_{\mathbf{C}^1,\dots,\mathbf{C}^k} \sim_s \mathbf{AP}$, and therefore, $\mathbf{A}_{\mathbf{C}^1,\dots,\mathbf{C}^k} \sim \mathbf{A}$ as desired.

II. BSR

Using the results from the BMMR step, we can assume that an estimate of $\mathbf{A}$ has been found. In order to solve the overcomplete BSS problem, we are therefore left with the task of reconstructing the sources using the mixtures $\mathbf{x}$ and the estimated matrix (BSR). Since $\mathbf{A}$ has full rank, the equation $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ yields the $(n-m)$-dimensional affine vector space $\mathbf{A}^{-1}\{\mathbf{x}(t)\}$ as solution space for $\mathbf{s}(t)$. Hence, if $n > m$, the source-recovery problem is ill-posed without further assumptions. Using a maximum-likelihood approach [4], [5], an appropriate assumption can be derived.

Given a prior probability $p_{\mathbf{s}}^0$ on the sources, it can be seen quickly [4], [10] that the most likely source sample is recovered by $\mathbf{s} = \arg\max_{\mathbf{x}=\mathbf{A}\mathbf{s}} p_{\mathbf{s}}^0$. Depending on the assumptions on the prior $p_{\mathbf{s}}^0$ of $\mathbf{s}$, we get different optimization criteria. In the experiments, we will assume a simple prior $p_{\mathbf{s}}^0 \propto \exp(-|\mathbf{s}|_p)$ with any p-norm $|.|_p$. Then $\mathbf{s} = \arg\min_{\mathbf{x}=\mathbf{A}\mathbf{s}} |\mathbf{s}|_p$, which can be solved linearly in the Gaussian case $p = 2$ and by linear programming or a shortest-path decomposition in the sparse, Laplacian case $p = 1$ (see [5], [10]).

III. Experimental Results

In order to compare the mixture matrix $\mathbf{A}$ with the recovered matrix $\mathbf{B}$ from the BMMR step, we calculate the *generalized crosstalking error* $E(\mathbf{A}, \mathbf{B})$ of $\mathbf{A}$ and $\mathbf{B}$ defined by $E(\mathbf{A}, \mathbf{B}) := \min_{\mathbf{M} \in \Pi} \|\mathbf{A} - \mathbf{B}\mathbf{M}\|$, where the minimum is taken over the group Π of all invertible matrices having only one nonzero entry per column, and $\|.\|$ denotes some matrix norm. It vanishes if and only if $\mathbf{A}$ and $\mathbf{B}$ are equivalent [10].

The overcomplete FastGeo algorithm is applied to four speech signals $\mathbf{s}$, mixed by a (2×4)-mixing matrix $\mathbf{A}$ with coefficients uniformly drawn from $[-1,1]$; see Fig. 2 for their mixture density. The algorithm estimates the matrix well with $E(\mathbf{A}, \hat{\mathbf{A}}) = 0.68$, and BSR by one-norm minimization yields recovered sources with a mean SNR of only 2.6 dB when compared with the original sources; as noted before [5], [10], without sparsification for instance by fast Fourier transform (FFT), source-recovery is difficult. To analyze the overcomplete FastGeo algorithm more generally, we perform 100 Monte Carlo runs using high-kurtotic gamma-distributed three-dimensional sources with 10^4 samples, mixed by a (2×3)-mixing matrix with weights uniformly chosen from $[-1,1]$. In Table I, the mean of the performance index depending on various parameters is presented. Noting that the mean error when using random (2×3)-matrices with coefficients uniformly taken from $[-1,1]$ is $E = 1.9 \pm 0.73$, we observe good performance, especially for a larger kernel radius and higher approximation parameter ($E = 0.29$), also compared with Soft-LOST's $E = 0.68$ [6].

TABLE I
Performance of BMMR-Algorithms ($n = 3$, $m = 2$, 100 Runs)

algorithm	mean $E(\mathbf{A}, \hat{\mathbf{A}})$	deviation σ
FastGeo (kernel $r = 5$, approx. 0.1)	0.60	0.60
FastGeo (kernel $r = 0$, approx. 0.5)	0.40	0.46
FastGeo (kernel $r = 5$, approx. 0.5)	0.29	0.42
Soft-LOST ($p = 0.01$)	0.68	0.57

As an example, in higher mixture dimension, three speech signals are mixed by a column-normalized (3×3)-mixing matrix $\mathbf{A}$. For $n = m = 3$, $m' = 2$, the projection framework simplifies to $k = 2$ with projections $\boldsymbol{\pi}_{\{1,2\}}$ and $\boldsymbol{\pi}_{\{1,3\}}$. Overcomplete geometric ICA is performed with $5 \cdot 10^4$ sweeps. The recoveries of the projected matrices $\boldsymbol{\pi}_{\{1,2\}}\mathbf{A}$ and $\boldsymbol{\pi}_{\{1,3\}}\mathbf{A}$ are quite good with $E(\boldsymbol{\pi}_{\{1,2\}}\mathbf{A}, \mathbf{B}^1) = 0.084$ and $E(\boldsymbol{\pi}_{\{1,3\}}\mathbf{A}, \mathbf{B}^2) = 0.10$. Taking out the permutations as described before, we get a recovered mixing matrix $\mathbf{A}_{\mathbf{B}^1,\mathbf{B}^2}$ with low generalized crosstalking error of $E(\mathbf{A}, \mathbf{A}_{\mathbf{B}^1,\mathbf{B}^2}) = 0.15$ (compared with a mean random error of $E = 3.2 \pm 0.7$).

References

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[2] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. New York: Wiley, 2002.

[3] J. Eriksson and V. Koivunen, "Identifiability, separability and uniqueness of linear ICA models," *IEEE Signal Process. Lett.*, vol. 11, no. 7, pp. 601–604, Jul. 2004.

[4] T. Lee, M. Lewicki, M. Girolami, and T. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representations," *IEEE Signal Process. Lett.*, vol. 6, no. 4, pp. 87–90, Apr. 1999.

[5] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Process.*, vol. 81, pp. 2353–2362, 2001.

[6] P. O'Grady and B. Pearlmutter, "Soft-LOST: EM on a mixture of oriented lines," in *Proc. ICA 2004*, ser. Lecture Notes in Computer Science Granada, Spain, 2004, vol. 3195, pp. 430–436.

[7] F. Theis, A. Jung, C. Puntonet, and E. Lang, "Linear geometric ICA: Fundamentals and algorithms," *Neural Comput.*, vol. 15, pp. 419–439, 2003.

[8] C. Puntonet and A. Prieto, "An adaptive geometrical procedure for blind separation of sources," *Neural Process. Lett.*, vol. 2, no. 5, pp. 23–27, 1995.

[9] M. Benaim, J.-C. Fort, and G. Pagés, "Convergence of the one-dimensional Kohonen algorithm," *Adv. Appl. Prob.*, vol. 30, pp. 850–869, 1998.

[10] F. Theis, E. Lang, and C. Puntonet, "A geometric algorithm for overcomplete linear ICA," *Neurocomputing*, vol. 56, pp. 381–398, 2004.