Shoji Makino
Te-Won Lee
Hiroshi Sawada

Blind Speech Separation

SPIN Springer's internal project number, if known

Springer
Berlin Heidelberg New York
Barcelona Budapest Hong Kong
London Milan Paris
Santa Clara Singapore Tokyo

Contents

14 Sparsification for Monaural Source Separation

Hiroki Asari[1], Rasmus K. Olsson[2], Barak A. Pearlmutter[3], and
Anthony M. Zador[4]

[1] Watson School of Biological Sciences, Cold Spring Harbor Laboratory
One Bungtown Road, Cold Spring Harbor, NY 11724, USA
E-mail: asari@cshl.edu
[2] Informatics and Mathematical Modelling, Technical University of Denmark
2800 Lyngby, Denmark
E-mail: rko@imm.dtu.dk
[3] Hamilton Institute, NUI Maynooth, Co. Kildare, Ireland
E-mail: barak@cs.nuim.ie
[4] Cold Spring Harbor Laboratory
One Bungtown Road, Cold Spring Harbor, NY 11724, USA
E-mail: zador@cshl.edu

Abstract. We explore the use of sparse representations for separation of a monaural mixture signal, where by a sparse representation we mean one where the number of non-zero elements is smaller than might be expected. This is a surprisingly powerful idea, as the ability to express a signal sparsely in some known, and potentially overcomplete, basis constitutes a strong model, while also lending itself to efficient algorithms. In the framework we explore, the representation of the signal is linear in a vector of coefficients. However, because many coefficient values could represent the same signal, the mapping from signal to coefficients is nonlinear, with the coefficients being chosen to simultaneously represent the signal and maximize a measure of sparsity. This conversion of the signal into the coefficients using L_1-optimization is viewed not as a pre-processing step performed before the data reaches the heart of the algorithm, but rather as *itself* the heart of the algorithm: after the coefficients have been found, only trivial processing remains to be done. We show how, by suitable choice of overcomplete basis, this framework can use a variety of cues (*e.g.*, speaker identity, differential filtering, differential attenuation) to accomplish monaural separation. We also discuss two radically different algorithms for finding the required overcomplete dictionaries: one based on non-negative matrix factorization of isolated sources, and the other based on end-to-end optimization using automatic differentiation.

14.1 Introduction

This chapter reviews the concept of sparsity in the context of single-channel signal separation. The key idea is to impose restrictions on the decompositions of data: while the codebooks/dictionaries are allowed to have a very large number of components, the encodings are constrained to be sparse, *i.e.*, to contain only a small number of non-zero values. Enforcing sparsity helps ensure a unique decomposition, and, more importantly, can be used to extract the individual source signals from the mixture. In a learning phase, dictionaries are adapted for each source class. By concatenating such dictionaries for all sources in a given mixture, separation can then be achieved in the joint sparse encoding if the sources are exclusively distributed and thus discriminable enough in the adapted dictionaries of each source class. Detailed prior knowledge is sometimes available on the sources, in which case sparse coding is sometimes able to isolate the sources even without going through the learning phase. However, here we will give a special focus on full adaptive methods, which includes learning from data a set of dictionaries that allows for sparse representations.

The inclusion of sparsity objectives in a machine learning task is very much biologically inspired. A striking feature of many sensory processing problems in humans is that by far more neurons appear to be engaged in the internal representations of the signal than in its transduction. The auditory (or visual) cortex has in fact orders of magnitude more neurons than the cochlear (or optic nerve), and thus the neural representation of an acoustic (or visual) stimulus is *overcomplete* in the sense that many more neurons are available than are needed to represent the stimulus with high fidelity. How does the brain then choose a unique representation if many different patterns of auditory (or visual) cortical activity could all faithfully represent any given pattern of cochlear (or optic nerve) activity? It is biologically appealing to sparsely encode the patterns because such representations are metabolically efficient [1, 2], and the principle of sparse (or "efficient") coding has been used to predict receptive field properties of both auditory and visual neurons [3–7].

In single-channel separation of musical and speech signals, mimicking the sparsity of neural representations has yielded good results [8–11]. The methods fail, however, when the signals are too similar, *e.g.*, in the cases of similarly sounding voices or two trumpets in a duet. In such adverse conditions, improved performance can be expected from exploiting grouping cues such as time-continuity and common-onset known to be employed by the auditory system [12]. For instance, Asari et al. [13] use the information provided by the differential filtering imposed on a source by its path from its origin to the cochlea (the head-related transfer function, or HRTF).

In section 14.2, we formulate a general framework for monaural source separation using dictionary methods. We also discuss learning algorithms for finding such dictionary elements suitable for sparse representations of given sources [14]. In section 14.3, we describe methods to achieve a sparse representation for given signals in an overcomplete basis, *i.e.*, L_1-norm minimization by linear programming. We then introduce one particular monaural segregation cue, the HRTF, and reformulate the model accordingly in section 14.5. In section 14.6 it is demonstrated that the HRTF cues lead to improved separation when the source signals originate from different directions. Note that in contrast to much previous work, the HRTF is used here to separate auditory streams rather than to localize them in space; the model assumes that the locations of the sources have already been determined by other mechanisms. Finally, we close with a brief comment on the perspectives of sparse overcomplete representations in section 14.7.

14.2 Problem Formulation

We consider a situation, where the observable is the sum of the source signals. For instance, this is a reasonable assumption in an acoustic setup, where sound waves from various emitters superpose at the microphone. While it is a common theme in techniques for blind source separation (BSS) to exploit the strong information provided by multiple sensors, here, only a single sensor is available for the estimation of the sources. Hence,

$$y(t) = \sum_{i=1}^{P} x_i(t), \tag{14.1}$$

where $y(t)$ and $x_i(t)$ are the time-domain mixture and source signals, respectively. While the problem cannot be solved in general for all classes of sources, solutions may be obtained for certain types of source distributions. For instance, humans at large possess the ability to isolate what is being said by a single speaker in a cocktail party situation, whereas a special training is required to listen out a single instrument from a musical piece, say, transcribe the bass from a rock 'n roll track [15]. Hence, the key to achieve the separation of the sources lies in learning features of the source distributions that are sufficiently discriminative to achieve separation and invertible such that the source signals can be reconstructed.

Inspired by the human auditory system, we will proceed to work in a time-frequency representation, $\mathbf{Y} = \mathrm{TF}\{y(t)\}$, since a number of advantages are associated with performing the computations in the transformed domain. We restrict TF such that $\mathbf{Y}$ is a real-valued matrix with spectral vectors, $\mathbf{y}$, as columns. The result of such a mapping is that certain classes of sources will become less overlapped in the transformed domain, which in turn facilitates the separation of the signals. More generally, if the sources can be assumed *sparsely* distributed in the frequency domain, additivity is approximately preserved in the transformed mixture,

$$\mathbf{y} = \sum_{i=1}^{P} \mathbf{x}_i \tag{14.2}$$

where $\mathbf{x}_i$ is the transformed source signal.

A class of algorithms, here denoted 'dictionary methods,' generally relies on learning factorizations of $\mathbf{x}_i$ from a training set of isolated source ensembles in terms of dictionaries $\mathbf{d}_{ij}$ and its encodings c_{ij},

$$\mathbf{x}_i = \sum_{j=1}^{N_i} \mathbf{d}_{ij} c_{ij} = \mathbf{D}_i \mathbf{c}_i \tag{14.3}$$

where the j-th column of $\mathbf{D}_i$ consists of $\mathbf{d}_{ij}$, and the j-th element of $\mathbf{c}_i$ holds the corresponding coefficient c_{ij}. Combining models (14.2) and (14.3) results in,

$$\mathbf{y} = \sum_{i=1}^{P} \mathbf{D}_i \mathbf{c}_i = \mathbf{D}\mathbf{c} \tag{14.4}$$

We allow the number of dictionary elements, $\sum_i N_i$, to be larger than the dimensionality of $\mathbf{y}$, meaning that $\mathbf{D}$ is potentially overcomplete, *i.e.*, many possible decompositions exist. This has been shown to result in more natural and compact representations [4–7, 16–18].

The application of a factorization in terms of dictionaries to the problem of signal separation fundamentally consists of two steps: first, a set of dictionaries, $\mathbf{D}_i$, is learned from a training set of unmixed $\mathbf{x}_i$. Second, the combined encoding, $\mathbf{c}$, is mapped onto the concatenation of the pre-learned source dictionaries, $\mathbf{D}$. Finally, the sources are estimated, re-synthesizing according to Eq. (14.3). In section 14.4, we provide examples of applications.

The method relies on the premise that the dictionaries of the sources in the mixture are sufficiently different such that $\mathbf{D}_1$ almost exclusively encode $\mathbf{x}_1$ but not $\mathbf{x}_2$, etc. Alternatively, it has been shown that source signals from identical distributions can be separated provided that information about the signal path is available [13]. This is described in more detail in section 14.5.

Different matrix factorization methods can be conceived based on various a priori assumptions of the dictionaries and encodings. Since computing $\mathbf{c}$ (given $\mathbf{D}$) from Eq. (14.4) is generally ill-posed, the model should at least impose sufficient constraints for the inversion to produce a well-defined solution. In section 14.2.1, we will proceed to describe criteria for learning dictionaries from training data. An important tool in this regard is linear programming, which can be employed to (i) learn the dictionaries, and (ii) compute the sparse decomposition required in Eq. (14.4) for the separation of the sources. The relevant aspects of linear programming are covered in section 14.3.

14.2.1 Dictionary Learning

We are concerned with devising a machine learning solution to acquire a set of dictionaries that can be used for source separation as formalized in Eq. (14.4). In order to be relevant in this regard, a dictionary should easily encode its class of signal, but at the same time be discriminative, meaning that encodings of other signals can be attributed with a low likelihood. In the following, it is described how to exploit inherent properties of the source signals to derive a learning algorithm producing dictionaries of the mentioned sort.

Non-negativity The particular choice of time-frequency representation determines the set of algorithms that can be employed. In audio applications, TF is often selected in a way so to mimic features of the early processing performed by the human auditory system. A common choice is to use a compressed version (*e.g.*, cube-root) of the power spectrogram, as computed by the short-time Fourier transform. This is often motivated by the fact that loudness perception in hearing can be approximated by a power law model [19]. An important feature of this representation is that it is non-negative, reflecting a property of neuronal signaling in terms of spike rates, which by definition are non-negative.

Lee and Seung [20] derived an efficient algorithm which minimizes the Euclidean distance between the data and the factorization, subject to non-negativity constraints. In terms of learning the dictionaries, $\mathbf{D}_i$, the objective is to optimize the function,

$$E_{\mathrm{NMF}} = \|\mathbf{X}_i - \mathbf{D}_i\mathbf{C}_i\|_{\mathrm{F}}^2 \quad \text{for} \quad \mathbf{D}_i \geq \mathbf{0}, \quad \mathbf{C}_i \geq \mathbf{0} \tag{14.5}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm, and $\mathbf{X}_i$ and $\mathbf{C}_i$ are matrices with data points and corresponding encodings as columns, respectively. From a probabilistic point of view, we can interpret the optimizer of (14.5) as a maximum posterior (MAP) estimator, assuming additive i.i.d. Gaussian noise and heaviside/uniform non-negative (improper) a priori distributions.

A closed-form solution is not available, but an effective gradient descent method emerges when the step-size is associated with a certain function of $\mathbf{X}_i$, $\mathbf{D}_i$, and $\mathbf{C}_i$. Starting from random non-negative matrices,

$$\mathbf{C}_i \leftarrow \mathbf{C}_i \bullet \frac{\mathbf{D}_i^\top \mathbf{X}_i}{\mathbf{D}_i^\top \widetilde{\mathbf{X}}_i}, \tag{14.6}$$

$$\mathbf{D}_i \leftarrow \mathbf{D}_i \bullet \frac{\mathbf{X}_i \mathbf{C}_i^\top}{\widetilde{\mathbf{X}}_i \mathbf{C}_i^\top}, \tag{14.7}$$

converges to a local minimum of (14.5), where $\widetilde{\mathbf{X}}_i = \mathbf{D}_i\mathbf{C}_i$ and the operators $\bullet$ and $\div$ indicate elementwise multiplication and division, respectively.[1] Non-negativity constraints have been used to learn signal dictionaries for single-channel separation of audio signals [9], and a convolutive extension of NMF has been particularly effective in this regard [21, 22], a technique reviewed by

[1] The NMF updates are derived from a steepest descend starting point, *i.e.*: $\mathbf{C}_i \leftarrow \mathbf{C}_i - \mathit{\Delta}_C \bullet \nabla_E$, where $\nabla_E = -2\mathbf{D}_i^\top(\mathbf{X}_i - \widetilde{\mathbf{X}}_i)$ is the gradient with respect to $\mathbf{C}_i$ and $\mathit{\Delta}_C$ is a step-size matrix. Setting $\mathit{\Delta}_C = \frac{\mathbf{C}_i}{2\mathbf{D}_i^\top \widetilde{\mathbf{X}}_i}$, we arrive at the stated learning rule for $\mathbf{C}_i$. The derivation for the rule regarding $\mathbf{D}_i$ is similar.

Smaragdis in Chap. 15 in this volume. Virtanen [23] provides a comprehensive review of NMF and related methods in audio analysis.

Sparsity In the following we describe how to apply the principle of sparsity to the learning of a decomposition in terms of a dictionary and its encoding as formalized in Eq. (14.3). Insisting on the sparsity of the encodings can be viewed as applying the principle of Occam's razor to the model, which states that the simplest explanation in some sense is to be preferred. The implication in a factorization setup is that $\mathbf{c}_i$ should be optimized as to contain as few non-zero entries as possible. In mathematical terms, minimize the L_0-norm, $\|\mathbf{u}\|_0 = \sum_i u_i^0$ for all $u_i \neq 0$.

Furthermore, sparseness is motivated from a perceptional, neural computational point of view. In neural terms, we could interpret c_{ij} as the neural activities (*e.g.*, spike rates) of the corresponding neurons characterized by their features $\mathbf{d}_{ij}$. The sparseness assumption then corresponds to representing the acoustic stimulus $\mathbf{y}$ in terms of the minimum number of spikes (Figure 14.1), a biologically appealing constraint which leads to an energy-efficient representation [1, 2]. Also note that the sparse coding is compatible with the "efficient coding hypothesis" [24], according to which the goal of sensory processing is to construct an efficient representation of the sensory environment.

The problem of finding an overcomplete signal dictionary tuned to a given stimulus ensemble, so that signals drawn from that ensemble will have sparse representations in the constructed dictionary, has received increasing attention, due to applications in both neuroscience and in the construction of efficient practical codes [25, 26]. Unfortunately, it is not computationally tractable to optimize directly the L_0-norm of the encoding. In fact, the problem is NP-complete [27]. As an alternative to the L_0-norm, the L_1-norm, defined $\|\mathbf{u}\|_1 = \sum_i |u_i|$, can be applied. In many situations, the L_1-norm solution approximates the L_0-norm solution, leading to equally sparse solutions [28], particularly in the presence of a noise model. The objective function balances the norm of the encoding with the accuracy of the fit,

$$E_{\mathrm{L1}} = \|\mathbf{c}_i\|_1 \quad \text{for} \quad \|\mathbf{x}_i - \mathbf{D}_i\mathbf{c}_i\|_p \leq \beta. \tag{14.8}$$

where β is proportional to the noise level and with $p = 1, 2$, or ∞. The optimization of (14.8) with respect to $\mathbf{c}_i$ can be viewed as a MAP estimator where an i.i.d. exponential a priori distribution is assumed for $\mathbf{c}_i$, and additive i.i.d. noise whose distribution is specified by β and p. Letting $\beta \to 0$ is equivalent to assuming that the noise is very small, and the solution converges to the zero-noise solution. The Gaussian noise case, $p = 2$, can be solved by semidefinite programming methods [29]. Both $p = 1$ and $p = \infty$ can be solved using linear programming, the details of which are covered in section 14.3.

When the objective is to learn dictionaries, *i.e.*, learning $\mathbf{D}$ from training data, one option is to optimize Eq. (14.8) with respect to $\mathbf{c}$ *and* $\mathbf{D}$ [14]. This

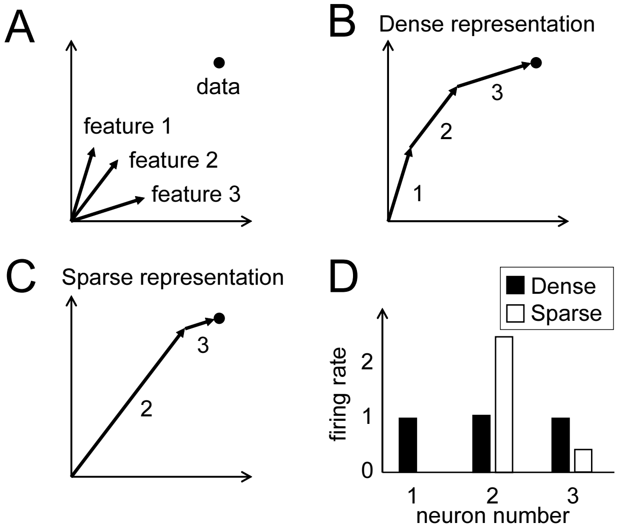

Fig. 14.1. Overcomplete representation in two dimensions. (A) Three non-orthogonal basis vectors (neural features) $\mathbf{d}_{ij}$ in two dimensions constitute an overcomplete representation, offering many possible ways to represent a data point $\mathbf{y}$ with no error. **(B)** The conventional solution is given by the pseudoinverse, yielding a *dense* neural representation where the squared sum of the coefficients (neural activities), $\|\mathbf{c}\|_2^2 = \sum_{ij} c_{ij}^2$, is minimized. This representation invokes all neural features about evenly. **(C)** The sparse solution invokes at most two neural features because it minimizes $\|\mathbf{c}\|_1 = \sum_{ij} |c_{ij}|$. **(D)** Comparison of neural activity for the two cases. For the dense representation, all three neurons participate about equally, whereas for the sparse representation activity is concentrated in neuron 2. [From 13, with permission.]

is likewise described in section 14.3. Benaroya et al. [9] combined sparsity and non-negativity constraints in order to learn audio dictionaries, whereas Jang and Lee [11] applied independent component analysis (ICA) which can also be seen as sparsity-inducing, depending on the source prior distribution.

14.3 Sparse Representation by Linear Programming

Linear programming solvers (LP) are often used as subroutines within larger systems, in both operations research and machine learning [30, 31]. One very simple example of this is in sparse signal processing, where it is common to represent a vector as sparsely as possible in an overcomplete basis; this representation can be found using LP, and the sparse representation is then used in further processing [25, 32–37]. In this section we explain how to use linear programming for two related tasks, namely (i) performing a sparse decomposition, as defined by the L_1-norm, of Eq. (14.8), and (ii) learning dictionaries by optimizing on a training set the L_1 sparsity of the decomposition.

In order to do so, we develop in section 14.3.1 a useful notation and formulate the sparse decomposition in terms of linear programming. Then, in section 14.3.2, we describe how to efficiently compute derivatives of functions where linear program solution is used as inputs. Finally, these theoretical foundations allow us to formulate a learning rule for obtaining dictionaries optimized for sparsity in section 14.3.3.

14.3.1 Basics

In order to develop a notation for LP, consider the general LP problem

$$\arg\min_{\mathbf{z}} \; \mathbf{w}^\top \mathbf{z} \text{ s.t. } \mathbf{A}\mathbf{z} \leq \mathbf{a} \text{ and } \mathbf{B}\mathbf{z} = \mathbf{b} \tag{14.9}$$

We will denote the linear program solver lp, and write the solution as $\mathbf{z} = \mathsf{lp}(\mathbf{w}, \mathbf{A}, \mathbf{a}, \mathbf{B}, \mathbf{b})$. It is important to see that $\mathsf{lp}(\cdot)$ can be regarded as either a mathematical function which maps LP problems to their solutions, or as a computer program which actually solves LP problems. Our notation deliberately does not distinguish between these two closely related notions.

Assuming feasibility, boundedness, and uniqueness, the solution to this LP problem will satisfy a set of linear equalities consisting of a subset of the constraints: the *active* constraints [38–40]. An LP solver calculates two pieces of information: the solution itself, and the identity of the active constraints. We will find it convenient to refer to the active constraints by defining some very sparse matrices that extract the active constraints from the constraint matrices. Let $\alpha_1 < \cdots < \alpha_n$ be the indices of the rows of $\mathbf{A}$ corresponding to active constraints, and $\beta_1 < \cdots < \beta_m$ index the active rows of $\mathbf{B}$. Without loss of generality, we assume that the total number of active constraints is equal to the dimensionality of the solution, $n + m = \dim \mathbf{z}$. We let $\mathbf{P}_\alpha$ be a matrix with n rows, where the i-th row is all zeros except for a one in the α_i-th column, and $\mathbf{P}_\beta$ similarly have m rows, with its i-th row all zeros except for a one in the β_i-th column. So $\mathbf{P}_\alpha \mathbf{A}$ and $\mathbf{P}_\beta \mathbf{B}$ hold the active rows of $\mathbf{A}$ and $\mathbf{B}$, respectively. These can be combined into a single matrix,

$$\mathbf{P} \equiv \begin{bmatrix} \mathbf{P}_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\beta \end{bmatrix}$$

Using these definitions, the solution $\mathbf{z}$ to (14.9), which presumably is already available having been computed by the algorithm that identified the active constraints, must be the unique solution of the system of linear constraints

$$\mathbf{P} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{z} = \mathbf{P} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$$

or

$$\mathsf{lp}(\mathbf{w}, \mathbf{A}, \mathbf{a}, \mathbf{B}, \mathbf{b}) = \mathsf{lq}(\mathbf{P} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}, \mathbf{P} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}) \tag{14.10}$$

where lq is a routine that efficiently solves a system of linear equations, $\mathsf{lq}(\mathbf{M}, \mathbf{m}) = \mathbf{M}^{-1}\mathbf{m}$. For notational convenience we suppress the identity of the active constraints as an output of the lp routine. Instead we assume that it is available where necessary, so any function with access to the solution $\mathbf{z}$ found by the LP solver is also assumed to have access to the corresponding $\mathbf{P}$.

14.3.2 Automatic Differentiation

Automatic differentiation (AD) is a process by which a numeric calculation specified in a computer programming language can be mechanically transformed so as to calculate derivatives (in the differential calculus sense) of the function originally calculated [41]. There are two sorts of AD transformations: forward accumulation [42] and reverse accumulation [43]. (A special case of reverse accumulation AD is referred to as backpropagation in the machine learning literature [44].) If the entire calculation is denoted $\mathbf{y} = h(\mathbf{x})$, then forward accumulation AD arises because a perturbation $d\mathbf{x}/dr$ induces a perturbation $d\mathbf{y}/dr$, and reverse accumulation AD arises because a gradient $dE/d\mathbf{y}$ induces a gradient $dE/d\mathbf{x}$. The Jacobian matrix $\mathbf{J}$ whose i, j-th entry is dh_i/dx_j plays a dominant role in reasoning about this process: forward AD calculates $\acute{\mathbf{y}} = \mathbf{J}\acute{\mathbf{x}} = \overrightarrow{h}(\mathbf{x}, \acute{\mathbf{x}})$, and reverse AD calculates $\grave{\mathbf{x}} = \mathbf{J}^\top \grave{\mathbf{y}} = \overleftarrow{h}(\mathbf{x}, \grave{\mathbf{y}})$. The difficulty is that, in high dimensional systems, the matrix $\mathbf{J}$ is too large to actually calculate. In AD the above matrix-vector products are found directly and efficiently, without actually calculating the Jacobian.

The central insight is that calculations can be broken down into a chained series of assignments $v := g(u)$, and transformed versions of these chained together. The transformed version of the above internal assignment statement would be $\acute{v} := \overrightarrow{g}(u, \acute{u}, v)$ in forward mode [42], or $\grave{u} := \overleftarrow{g}(u, v, \grave{v})$ in reverse mode [43]. The most interesting property of AD, which results from this insight, is that the time consumed by the adjoint calculations can be the same as that consumed by the original calculation, up to a small constant factor. (This naturally assumes that the transformations of the primitives invoked also obey this property, which is in general true.)

We will refer to the adjoints of original variables introduced in forward accumulation (perturbations) using a forward-leaning accent $v \mapsto \acute{v}$; to the adjoint variables introduced in the reverse mode transformation (sensitivities) using a reverse-leaning accent $v \mapsto \grave{v}$; and to the forward- and reverse-mode transformations of functions using forward and reverse arrows, $h \mapsto \overrightarrow{h}$ and $h \mapsto \overleftarrow{h}$, respectively. A detailed introduction to AD is beyond the scope of this chapter, but one form appears repeatedly in our derivations, *i.e.*, $\mathbf{V} := \mathbf{AUB}$ where $\mathbf{A}$ and $\mathbf{B}$ are constant matrices and $\mathbf{U}$ and $\mathbf{V}$ are matrices as well. This transforms to

$$\acute{\mathbf{V}} := \mathbf{A}\acute{\mathbf{U}}\mathbf{B} \tag{14.11}$$

$$\grave{\mathbf{U}} := \mathbf{A}^\top \grave{\mathbf{V}} \mathbf{B}^\top. \tag{14.12}$$

AD of a Linear Equation Solver We first derive AD equations for a simple implicit function, namely a linear equation solver. We consider a subroutine lq which finds the solution $\mathbf{z}$ of $\mathbf{Mz} = \mathbf{m}$, written as $\mathbf{z} = \mathsf{lq}(\mathbf{M}, \mathbf{m})$. This assumes that $\mathbf{M}$ is square and full-rank, just as a division operation $z = x/y$ assumes that $y \neq 0$. We will derive formulae for both forward mode AD (the $\acute{\mathbf{z}}$ induced by $\acute{\mathbf{M}}$ and $\acute{\mathbf{m}}$) and reverse mode AD (the $\grave{\mathbf{M}}$ and $\grave{\mathbf{m}}$ induced by $\grave{\mathbf{z}}$).

For forward propagation of perturbations, we will write $\acute{\mathbf{z}} = \overset{\rightharpoonup}{\mathsf{lq}}(\mathbf{M}, \acute{\mathbf{M}}, \mathbf{m}, \acute{\mathbf{m}}, \mathbf{z})$. Using Eq. (14.11), we have that $(\mathbf{M}+\acute{\mathbf{M}})(\mathbf{z}+\acute{\mathbf{z}}) = \mathbf{m}+\acute{\mathbf{m}}$ which reduces to $\mathbf{M}\acute{\mathbf{z}} = \acute{\mathbf{m}} - \acute{\mathbf{M}}\mathbf{z}$. Hence, we conclude that

$$\overset{\rightharpoonup}{\mathsf{lq}}(\mathbf{M}, \acute{\mathbf{M}}, \mathbf{m}, \acute{\mathbf{m}}, \mathbf{z}) = \mathsf{lq}(\mathbf{M}, \acute{\mathbf{m}} - \acute{\mathbf{M}}\mathbf{z}).$$

Note that lq is linear in its second argument, where the perturbations enter linearly. For reverse propagation of sensitivities, we will write

$$\left[\grave{\mathbf{M}}\ \grave{\mathbf{m}}\right] = \overset{\leftharpoonup}{\mathsf{lq}}(\mathbf{M}, \mathbf{m}, \mathbf{z}, \grave{\mathbf{z}}). \tag{14.13}$$

First observe that $\mathbf{z} = \mathbf{M}^{-1}\mathbf{m}$ and hence $\grave{\mathbf{m}} = \mathbf{M}^{-\top}\grave{\mathbf{z}}$ so

$$\grave{\mathbf{m}} = \mathsf{lq}(\mathbf{M}^{\top}, \grave{\mathbf{z}}).$$

For the remaining term we start with our previous forward perturbation $\acute{\mathbf{M}} \mapsto \acute{\mathbf{z}}$, namely $\acute{\mathbf{z}} = -\mathbf{M}^{-1}\acute{\mathbf{M}}\mathbf{z}$, and note that the reverse must be the transpose of this linear relationship (*i.e.*, using Eq. (14.11) and Eq. (14.12)), $\grave{\mathbf{M}} = -\mathbf{M}^{-\top}\grave{\mathbf{z}}\mathbf{z}^{\top}$, which is the outer product

$$\grave{\mathbf{M}} = -\grave{\mathbf{m}}\mathbf{z}^{\top}.$$

AD of Linear Programming We apply Eq. (14.13) followed by some bookkeeping, yields

$$\begin{aligned}
\begin{bmatrix} \grave{\mathbf{A}} & \grave{\mathbf{a}} \\ \grave{\mathbf{B}} & \grave{\mathbf{b}} \end{bmatrix} &= \overset{\leftharpoonup}{\mathsf{lp}}(\mathbf{w}, \mathbf{A}, \mathbf{a}, \mathbf{B}, \mathbf{b}, \mathbf{z}, \grave{\mathbf{z}}) \\
&= \mathbf{P}^{\top}\, \overset{\leftharpoonup}{\mathsf{lq}}(\mathbf{P}\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}, \mathbf{P}\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \mathbf{z}, \grave{\mathbf{z}}) \\
\grave{\mathbf{w}} &= \mathbf{0}
\end{aligned}$$

Forward accumulation is similar, but is left out for brevity.

Constrained L_1 Optimization We can find AD equations for linearly constrained L_1-norm optimization via reduction to LP. Consider

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \text{ s.t. } \mathbf{Dc} = \mathbf{y}.$$

Although $\|\mathbf{c}\|_1 = \sum_i |c_i|$ is a nonlinear objective function, a change in parametrization allows optimization via LP. We name the solution $\mathbf{c} = \mathsf{L1opt}(\mathbf{y}, \mathbf{D})$ where

$$\mathsf{L1opt}(\mathbf{y}, \mathbf{D}) = \left[\mathbf{I} \; -\mathbf{I}\right] \mathsf{lp}(\mathbf{1}, -\mathbf{I}, \mathbf{0}, \mathbf{D}\left[\mathbf{I} \; -\mathbf{I}\right], \mathbf{y})$$

in which $\mathbf{0}$ and $\mathbf{1}$ denote column vectors whose elements all contain the indicated number, and each $\mathbf{I}$ is an appropriately sized identity matrix. The reverse-mode AD transformation follows immediately,

$$\overleftharpoon{\mathsf{L1opt}}(\mathbf{y}, \mathbf{D}, \mathbf{c}, \grave{\mathbf{c}}) = \left[\grave{\mathbf{D}} \; \grave{\mathbf{y}}\right] = \\ \left[\mathbf{0}' \; \mathbf{I}\right] \overleftharpoon{\mathsf{lp}}(\mathbf{1}, -\mathbf{I}, \mathbf{0}, \mathbf{D}\left[\mathbf{I} \; -\mathbf{I}\right], \mathbf{y}, \mathbf{z}, \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \grave{\mathbf{c}}) \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

where $\mathbf{z}$ is the solution of the internal LP problem and $\mathbf{0}'$ is an appropriately sized matrix of zeros.

14.3.3 Dictionaries Optimized for Sparsity

A major advantage of the LP differentiation framework, and more specifically the reverse accumulation of the constrained L_1-norm optimization, is that it provides directly a learning rule for finding sparse representations in overcomplete dictionaries.

We assume an overcomplete dictionary in the columns of $\mathbf{D}$, which is used to encode a signal represented in the column vector $\mathbf{y}$ using the column vector of coefficients $\mathbf{c} = \mathsf{L1opt}(\mathbf{y}, \mathbf{D})$ where each dictionary element has unit L_2 length. We will update $\mathbf{D}$ so as to minimize $E = \langle \|\mathsf{L1opt}(\mathbf{y}, \mathbf{D})\|_1 \rangle$ while keeping the columns of $\mathbf{D}$ at unit length. This can be regarded a special case of ICA [45], where measures of independence across coefficients are optimized. We wish to use a gradient method so we calculate $\nabla_{\mathbf{D}} E_{\mathbf{y}}$ where $E_{\mathbf{y}} = \|\mathsf{L1opt}(\mathbf{y}, \mathbf{D})\|_1$ making $E = \langle E_{\mathbf{y}} \rangle$. Invoking AD,

$$\begin{aligned} \nabla_{\mathbf{D}} E_{\mathbf{y}} &= \grave{\mathbf{D}} = \left[\grave{\mathbf{D}} \; \grave{\mathbf{y}}\right] \begin{bmatrix} \mathbf{I} \\ \mathbf{0}^\top \end{bmatrix} \\ &= \overleftharpoon{\mathsf{L1opt}}(\mathbf{y}, \mathbf{D}, \mathbf{c}, \operatorname{sign}(\mathbf{c})) \begin{bmatrix} \mathbf{I} \\ \mathbf{0}^\top \end{bmatrix} \end{aligned} \tag{14.14}$$

where $\operatorname{sign}(x) = +1/0/-1$ for x positive/zero/negative, and applies elementwise to vectors.

We are now in a position to perform stochastic gradient optimization [46], modified by the inclusion of a normalization step to maintain the columns of $\mathbf{D}$ at unit length and non-negative.

```
Draw $\mathbf{y}$ from signal distribution.
Calculate $E_{\mathbf{y}}$.
Calculate $\nabla_{\mathbf{D}} E_{\mathbf{y}}$ by (14.14).
Step $\mathbf{D} := \mathbf{D} - \eta \nabla_{\mathbf{D}} E_{\mathbf{y}}$.
Set any negative element of $\mathbf{D}$ to zero.
Normalize the columns $\mathbf{d}_i$ of $\mathbf{D}$ to unit $L_2$-norm.
Repeat to convergence of $\mathbf{D}$.
```

This procedure can be regarded as an efficient exact maximum likelihood treatment of the posterior calculated by Lewicki and Sejnowski [25] using a Gaussian approximation. It is interesting to note that the formulation here can be easily and mechanically generalized to other objectives.

14.4 Source Separation Example

We will now demonstrate an application of the framework laid out in sections 14.2 and 14.3. More specifically, we will attempt to separate two speech signals of equal power from a single mixture as was proposed in Pearlmutter and Olsson [14]. This is a relevant task in hearing aids, as a speech recognition pre-processor, and in other applications which might benefit from better noise reduction. For this reason, there has been a flurry of interest in the problem.

A common trait of many approaches is that speaker-dependent models have been learned from a training set of isolated recordings and subsequently a combination of these have been applied to the mixture. Roweis [47] learned hidden Markov models (HMM) of individual speakers and combined them in a factorial HMM, separating a mixture. The high dimensionality of the combined state space prohibited direct inference, but an approximate solution was obtained. A Bayesian solution to inference in the factorial HMM, applying a set of milder assumptions, was provided by Kristjansson et al. [48], achieving a very good (super-human) performance on a word recognition task. Bach and Jordan [49] devised a clustering algorithm based on specific features of speech, which does not learn models for each speaker. Dictionary methods, which do not require combinatorial searches, have been based on a priori assumptions of sparsity and/or non-negativity (see section 14.2).

In the following, we will twice evoke the assumption of L_1 sparsity: first in order to learn the dictionaries, *i.e.*, inverting Eq. (14.3); second, to compute the separating decomposition of Eq. (14.4).

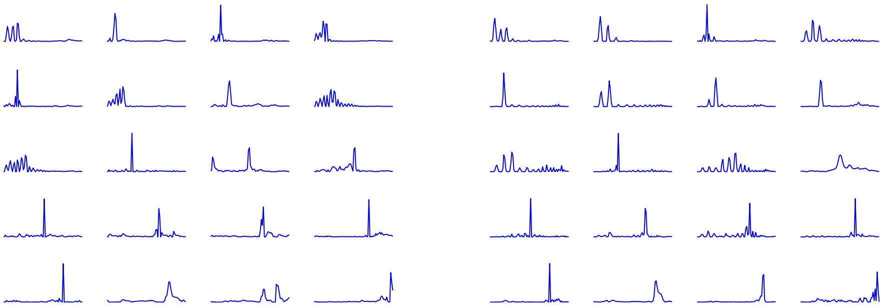

Fig. 14.2. A sample of learnt dictionary entries for male (left) and female (right) speech in the Mel spectrum domain. Harmonic features have clearly been found from the data, but broad and narrow noise spectra are also visible.

14.4.1 Dictionary Learning

A set of personalized speech dictionaries was learned by sparsity optimization using the method described in section 14.3.[2] Defining the time-frequency transformation (TF), the speech was preprocessed and represented to (essentially) transform the audio signals into an amplitude Mel time-frequency representation [51]. The stochastic gradient optimization of the linearly constrained L_1-norm was run for 40,000 iterations. The step-size η was decreased throughout the training. The $N = 256$ columns of the dictionaries were initialized with narrow pulses distributed evenly across the spectrum and non-negativity was enforced following each iteration. In figure 14.2 is displayed a randomly selected sample of learnt dictionary elements of one male and one female speaker. The dictionaries clearly capture a number of characteristics of speech, such as quasi-periodicity and dependencies across frequency bands.

14.4.2 Source Separation

In order to separate the sources, we assume the additive mixture model of Eq. (14.4) and perform a sparse decomposition by minimizing the L_1-norm. Thus, a linear program is used to compute $\mathbf{c} = \texttt{L1opt}(\mathbf{y}, \mathbf{D})$, where $\mathbf{y}$ is the mixture, $\mathbf{D} = [\mathbf{D}_1 \quad \mathbf{D}_2]$ is the concatenated dictionary, and $\mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}$ is the joint source encoding. Assuming that the $\mathbf{D}_1$ and $\mathbf{D}_2$ are different in some sense, it can be expected that a sparse representation in the basis $\mathbf{D}$ coincides with the separation of the sources. The degree of success depends on the level to which the signals (and dictionaries) are different. The source estimates in the Mel spectrum domain are then re-synthesized according as Eq. (14.3):

[2] The GRID corpus was used [50]. It contains 1000 short sentences recorded for each of 34 speakers.

Genders	SNR (dB)
M/M	4.9±1.2
M/F	7.8±1.3
F/F	5.1±1.4

Table 14.1. Monaural two-speaker signal-to-noise separation performance (mean±stderr of SNR), by speaker gender. The simulated test data consisted of all possible combinations, $T = 6\,\mathrm{s}$, of the 34 speakers. Clearly, it is an easier task to separate the speech signals in the case of opposite-gender speakers. This indicates the required level of contrast between the source signals for the method to work.

$\hat{\mathbf{x}}_1 = \mathbf{D}_1\mathbf{c}_1$ and $\hat{\mathbf{x}}_2 = \mathbf{D}_2\mathbf{c}_2$. The conversion back to the time-domain consists of mapping to the amplitude spectrogram and subsequently reconstructing the time-domain signal using the noisy phase of the mixture. Due to the sparsity of speech in the transformed domain, the degree of overlap of the sources is small, which causes the approximation to be fairly accurate.

The quality of the source estimates was evaluated in the time-domain simply as the ratio of powers of the target to reconstruction error, here termed the signal-to-noise ratio (SNR). Table 14.1 lists the performance of the method on the GRID database.

14.5 Convolutional Mixing and Head-Related Transfer Function

One limitation of the BSS model as described in Eqs. (14.1)–(14.4) is that source signals from identical distributions (or from different distributions with the same statistics) can hardly be separated because the performance depends on the “personalized” dictionaries that exclusively encode one source signal but not the others. In this section, we will then describe how we could extend the model to exploit additional separation cues that “tag” the dictionaries so they can be assigned to the appropriate sources in the framework of sparse overcomplete representations. Specifically, we will consider the monaural source separation of convolutive sources, *i.e.*, separating multiple *pre-filtered* signals combined at a single sensor, using biologically inspired spectral cues that segregate the auditory scene based on the source locations [13].

14.5.1 Head-Related Transfer Function

The auditory system uses a wide variety of psychophysical cues to segregate auditory streams [12], including both binaural and monaural cues. Many monaural cues have been identified, such as common onset time or comodulation of stimulus power in different parts of the spectrum.

For simplicity, here we focus on just one set of cues: those provided by the differential filtering imposed on a source by its path from its origin in space to the cochlea. This filtering or ‘spectral coloring’ is caused both by the head and the detailed shape of the ear (the head-related transfer function, or HRTF), and by the environment on sources at different positions in space. The HRTF depends on the spatial position—both the relative azimuth and elevation—of the source. At some frequencies, the HRTF can attenuate sound from one location by as much as 40 dB more than from another, and such HRTF cues, when present, help in source separation [52].

The HRTF is also important for generating a three-dimensional experience of sound, so that acoustic sources that bypass the HRTF (*e.g.*, those presented with headphones) are typically perceived unnaturally, as though arising inside the head [53, 54]. Note however that the HRTF is used here to *separate* auditory streams rather than to *localize* them in space, in contrast to much previous work on the role of the HRTF in sound localization [53, 55–57].

It is often reasonable to assume that sound arriving from different locations should be treated as arising from distinct sources. We thus assume that all sounds from a given position are *defined* to belong to the same source, and any sounds from a different position are defined to belong to different sources. We emphasize that although sound localization (the process by which an animal determines where in space a source is located) is related to source separation (the process by which an animal extracts different auditory streams from a single waveform), the two computations are distinct; neither is necessary nor sufficient for the other. Here we focus on the separation problem, and assume that source localization occurs by other mechanisms.

14.5.2 Reformulation

Here we will reformulate the BSS model in Eqs. (14.1)–(14.4) for the monaural source separation problem of convolutive sources. Suppose there are P acoustic sources located at known distinct positions in space, with $x_i(t)$ being the time course of the stimulus sound pressure of the i-th source at its point of origin. Associated with each position is a known filter given by $h_i(t)$. In what follows we will refer to $h_i(t)$ as the HRTF, but in general $h_i(t)$ will include not just the filtering of the head and external ear, but also the filter function of the acoustic environment (reverberation, etc.)

The signal $y(t)$ at the ear is then the sum of the filtered signals,

$$y(t) = \sum_{i=1}^{P} h_i(t) * x_i(t) = \sum_{i=1}^{P} \widetilde{x}_i(t) \tag{14.1'}$$

where $*$ indicates convolution and $\widetilde{x}_i(t) = h_i(t) * x_i(t)$ is the i-th source in isolation following filtering. (We can say that $x_i(t)$ is the i-th source measured in source space, while $\widetilde{x}_i(t)$ is the same source measured in sensor space.) The goal is then to recover the underlying sources $x_i(t)$ from the signal $y(t)$, using knowledge of the directional filters $h_i(t)$. Note that the actual spatial locations of the sources are not computed during the separation in this model but we assume the locations (and thus associated directional filters $h_i(t)$) have already been identified by other mechanisms.

In the TF domain, we have

$$\mathbf{y} = \sum_{i=1}^{P} \mathbf{h}_i \bullet \mathbf{x}_i = \sum_{i=1}^{P} \widetilde{\mathbf{x}}_i \tag*{(14.2’)}$$

where $\bullet$ indicates elementwise multiplication. As in Eq. (14.3), we then assume that each source $\mathbf{x}_i$ can be expressed as a linear combination of dictionary elements $\mathbf{d}_j$:

$$\mathbf{x}_i = \sum_{j=1}^{N_i} \mathbf{d}_j c_{ij} = \mathbf{D}\mathbf{c}_i \tag*{(14.3’)}$$

Note that we no longer have to use “personalized” dictionaries for each source but we could use any dictionary set $\mathbf{D}$ that captures the spectral correlations in the sources and permits sparse representations, *i.e.*, where only a small number of coefficients c_{ij} are significantly non-zero. By further assuming that the dictionaries in sensor space, $\widetilde{\mathbf{d}}_{ij}$, are related to the dictionaries in source space, $\mathbf{d}_j$, by convolution with each filter $\mathbf{h}_i$:

$$\widetilde{\mathbf{d}}_{ij} = \mathbf{h}_i \bullet \mathbf{d}_j, \tag*{(14.15)}$$

the signal $\mathbf{y}$ received at the ear can be expressed as a linear combination of the dictionary elements in sensor space:

$$\begin{aligned}
\mathbf{y} &= \sum_{i=1}^{P} \mathbf{h}_i \bullet \mathbf{x}_i && \text{by (14.2’)}\\
&= \sum_{i=1}^{P} \mathbf{h}_i \bullet \left(\sum_{j=1}^{N_i} \mathbf{d}_j c_{ij} \right) && \text{by (14.3’)}\\
&= \sum_{i,j} \widetilde{\mathbf{d}}_{ij} c_{ij} && \text{by (14.15)}\\
&= \widetilde{\mathbf{D}}\mathbf{c}. && \text{(14.4’)}
\end{aligned}$$

As before, the BSS model in Eqs. (14.1’)–(14.4’) consists of two steps: first, a set of dictionary in source space $\mathbf{D}$ is learned from a training set of unmixed signals $\mathbf{x}_i$. Second, given a convolutional mixture $\mathbf{y}$ and position-dependent filters $\mathbf{h}_i$, appropriate coefficients c_{ij} are obtained for Eq. (14.4’) under a sparseness prior (*i.e.*, by computing $\mathbf{c} = \mathsf{L1opt}(\mathbf{y}, \widetilde{\mathbf{D}})$), and a given source

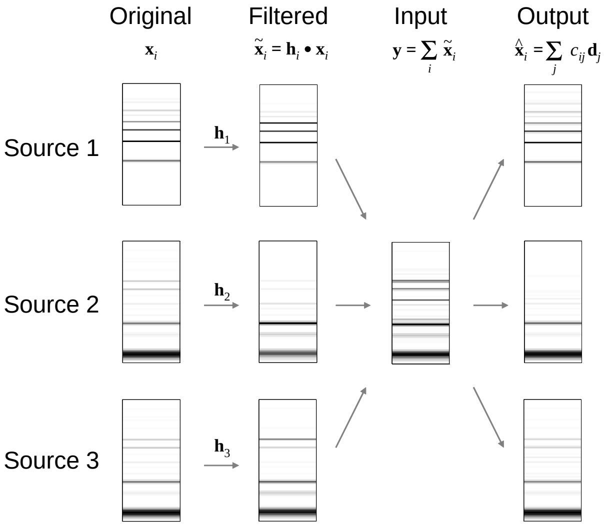
Original
Filtered
Input
Output
$\mathbf{x}_i$
$\tilde{\mathbf{x}}_i = \mathbf{h}_i \bullet \mathbf{x}_i$
$\mathbf{y} = \sum_i \tilde{\mathbf{x}}_i$
$\hat{\mathbf{x}}_i = \sum_j c_{ij}\mathbf{d}_j$
Source 1
Source 2
Source 3
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_3$

Fig. 14.3. Separation of three musical sources. Three musical instruments at three distinct spatial locations were filtered (by $\mathbf{h}_1, \ldots, \mathbf{h}_3$, corresponding to the HRTFs for azimuth -90°, 0° and 90° with zero elevation, respectively) and summed to produce the *input* $\mathbf{y}$, and then separated using a sparse overcomplete representation to produce the *output*. Note that two of the sources (a harp playing the note "D", *center* and *bottom*) were chosen to be identical; this example is thus particularly challenging, since the only cue for separating the sources is the filtering imposed by the HRTF. Nevertheless, separation was good as seen by comparing the left (*Original*) and right (*Output*) columns. [From 13, with permission.]

i can be reconstructed by summing over all dictionary elements associated with position i using Eq. (14.3'). Note that separation and deconvolution are simultaneously achieved here by estimating the coefficients by using a post-HRTF (sensor space) dictionary $\widetilde{\mathbf{D}}$ but reconstructing the signals by using a pre-HRTF (source space) dictionary $\mathbf{D}$ (Figure 14.3).

14.6 Separation of Convolutive Sources

Successful source separation for the BSS models described in section 14.5 requires that two conditions are satisfied. First, the sources must be sparsely representable, as is the case with natural auditory stimuli [4, 5, 18, 58]. Second, the sources must have spectral correlations matched to the HRTF. In the following, we will demonstrate that the model is able to separate acoustic sources consisting of mixtures of music, natural sounds, and speech.[3]

[3] Sound data were taken from commercially available audio CDs, and the HRTF data for a representative left human pinna were downloaded from http://www.itakura.nuee.nagoya-u.ac.jp/HRTF/ [59].

14.6.1 Dictionary Learning

Here we used non-negative matrix factorization (NMF; see also section 14.2.1) to generate a set of complete dictionaries from spectrograms obtained from unmixed samples of solo instrumental music, natural sounds, and speech ($\mathbf{D}_{\mathrm{ms}}$, $\mathbf{D}_{\mathrm{ns}}$, and $\mathbf{D}_{\mathrm{sp}}$, respectively), and concatenated them to form an overcomplete source space dictionary: $\mathbf{D} = [\mathbf{D}_{\mathrm{ms}} \quad \mathbf{D}_{\mathrm{ns}} \quad \mathbf{D}_{\mathrm{sp}}]$.

The ability of the NMF dictionaries to represent sounds in a sparse model can be quantified in terms of the “sparseness index,” defined as $\|\mathbf{c}_i\|_0 / \dim \mathbf{x}_i$ in the presence of a single (unmixed) source $\mathbf{x}_i$ (see Eq. (14.3’)). The distribution of the index was 0.61±0.27, 0.64±0.17, 0.49±0.13 (mean±SD) for $\mathbf{D}_{\mathrm{ms}}$, $\mathbf{D}_{\mathrm{ns}}$, and $\mathbf{D}_{\mathrm{sp}}$, respectively, over 10,000 test samples. This suggests that the NMF dictionaries generally led to sparse representations of the ensembles, satisfying the first condition for the model to work.

When applied to music, NMF typically yielded elements suggestive of musical notes, each with a strong fundamental frequency and weaker harmonics at higher frequencies. In many cases, listeners could easily use timbre to identify the instrument from which a particular element was derived. When applied to sounds from other ensembles (natural sounds and speech), NMF yielded elements that had rich harmonic structure, but it was not in general easy to “interpret” the elements (*e.g.*, as vowels). Nonetheless these elements captured aspects of the statistical structure of the underlying ensemble of sounds, and thus satisfied the second condition as well.

It should be mentioned that the choice of NMF was merely a matter of convenience; we could have used any basis that satisfies the two conditions. Finding good overcomplete dictionaries from samples of a stimulus ensemble is a subject of ongoing research [26, see also section 14.3.3]. NMF is then not necessarily the best algorithm in this context, but is simply good enough for our monaural BSS model.

14.6.2 Separation with HRTF

To demonstrate the model’s ability to separate sources, we generated digital mixtures of three sources positioned at three distinct positions in space (Figure 14.3). On the *left column* are the spectrograms of the sources at their origin. Two of the sources (a harp playing the note “D”, *center* and *bottom*) were chosen to be identical; this example is thus particularly challenging, since the only cue for separating the sources is the filtering imposed by the HRTF.

Separation was nevertheless quite successful (compare *left* and *right* columns). These results were typical: whenever the underlying assumptions about the sparseness of the stimulus were satisfied, sources consisting of mixtures of music, natural sounds or speech were separated well (Figure 14.4A). Separation worked particularly well for mixtures of sparsely representable sources (*i.e.*, smaller sparseness index values), whereas it did not work for sources that were not sparsely represented (*i.e.*, larger sparseness index values.) Figure 14.4B shows that separation without differential pre-filtering by the HRTF was unsuccessful, as was separation using the Gaussian prior instead of the sparseness prior (dense representation: L_2-norm minimization).

The procedure for source separation in the BSS model conceptually consists of two distinct steps (although in practice the two steps occur simultaneously). In the first step, the stimuli are decomposed into the appropriate dictionary elements. In the second step, the dictionary elements are tagged and bundled together with other elements from the same source. It is for this bundling or "tagging" step that the HRTF along with the prior knowledge of source locations is essential.

The failure of the dense representation to separate sources (Figure 14.4) results from a failure of the first step. Instead of decomposing the sources into a small number of dictionary elements, the dense representation assumes that each element contributed about equally to the received signal, and so finds a representation in which a large fraction of dictionary elements are involved. That is, instead of "explaining" the sources in terms two harps and a trumpet, the dense representations also finds some clarinet, some cello, etc. at all positions. This is intrinsic to the dense solution, since it finds the "minimum power" solution in which neural activity is spread among the population (Figure 14.1B).

The failure of even the sparse approach when the spectral cues induced by the HRTF are absent (Figure 14.4B, leftmost point showing 0-degree separation) results from a failure at the second step. That is, the sparse approach finds a useful decomposition at the first step even without the HRTF, but in the absence of HRTF cues the active elements are not tagged, and so the dictionary elements cannot be assigned appropriately to distinct sources. Other psychophysical cues relevant for source separation, such as common onset time, might provide alternative or additional tags in this same framework. A more general formulation of source separation might allow tagging on longer time scales, so that a set of dictionaries active at one moment might be more (or less) likely to be active the next, reflecting the fact that sources tend to persist, but we do not pursue that approach further here.

14.7 Conclusion

Sparse overcomplete representations can monaurally separate a mixture of sound sources into its constituent auditory streams. In our framework it is critical to use an appropriate overcomplete basis in order to achieve acceptable separation performance, and we described one way to exploit inherent properties of source signals for finding discriminative or "personalized" dictionaries that allow sparse representations of particular sound ensembles. We also modified the separation model to instead exploit one type of monaural separation cues that animals use, the HRTF, to "tag" dictionary elements so

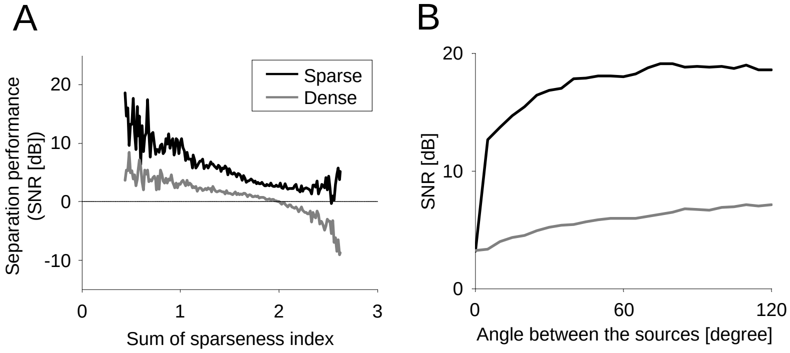

Fig. 14.4. Separation performance with three sources. (A) The separation performance (SNR in the TF domain averaged across the sources) is shown as a function of the sum of the "sparseness index" of the three sources (average over 20,000 sets of test sample mixtures). The sparse prior (*black*) always outperforms the dense prior (*gray*), and excellent separation was achieved especially when the sources are sparsely representable. The source locations were randomly chosen but 90° apart from each other with zero elevation. **(B)** Using a typical example of three novel stimuli (trumpet and two same harp), separation performance (y-axis) was examined with all the possible combinations of the three sources (from 0 to 120 degrees apart; x-axis). The average performance is shown here under either the sparse (*black*) or the dense (*gray*) prior. Note that separation was unsuccessful at angle zero since we cannot exploit *differential* filtering, whereas the performance gets better as the sources get further apart. [From 13, with permission.]

they can be assigned to the appropriate sources. We expect that other psychophysical cues important for acoustic stream segregation, such as common onset time, could be used in a similar way.

Recent advances in ICA have emphasized the utility of sparse overcomplete representations for source separation problems in acoustic, visual and other domains [25, 28, 35, 36, 60–62]. Our formulation of the source separation problem has been built on these ideas, generalizing the framework to allow cues other than differential attenuation. (It does not, however, sacrifice the ability to use binaural cues like differential attenuation, since such cues can also be incorporated by simply replacing the single-input-single-output HRTF filters by single-input-two-output filters, doubling the size of the dictionary elements by leaving the algorithm otherwise unchanged.) We have demonstrated the power and flexibility of the framework by applying it to two difficult monaural separation problems, one using as its sole cue differential low-level source models for the speakers, the other using as its sole cue the differential filtering of different sources by the HRTF.

Sparseness provides a powerful and useful constraint for choosing a unique representation in an overcomplete basis. We think that sparse representations can be a generic model for signal processing even in control theory or statistics as well as in neuroscience, and further advances in optimization and learning algorithms will find out its practical usages in many aspects, including the cocktail party problem in more general settings.

Bibliography

[1] S. B. Laughlin and T. J. Sejnowski, “Communication in neuronal networks,” *Science*, vol. 301, no. 5641, pp. 1870–4, 2003.

[2] W. B. Levy and R. A. Baxter, “Energy efficient neural codes,” *Neu. Comp.*, vol. 8, no. 3, pp. 531–43, 1996.

[3] E. P. Simoncelli and B. A. Olshausen, “Natural image statistics and neural representation,” *Annu. Rev. Neurosci.*, vol. 24, no. 1, pp. 1193–1216, 2001.

[4] M. S. Lewicki, “Efficient coding of natural sounds,” *Nat. Neurosci.*, vol. 5, no. 4, pp. 356–363, 2002.

[5] E. C. Smith and M. S. Lewicki, “Efficient auditory coding,” *Nature*, vol. 439, pp. 978–982, 2006.

[6] A. J. Bell and T. J. Sejnowski, “The ‘independent components’ of natural scenes are edge filters,” *Vision Research*, vol. 37, no. 23, pp. 3327–38, 1997.

[7] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, pp. 607–609, 1996.

[8] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler, “Automatic music transcription and audio source separation,” *Cybernetics and Systems*, vol. 33, no. 6, pp. 603–627, 2002.

[9] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, “Non negative sparse representation for wiener based source separation with a single sensor,” in *Acoustics, Speech, and Signal Processing*, 2003, pp. 613–616.

[10] T. Virtanen, “Sound source separation using sparse coding with temporal continuity objective,” in *ICMC*, 2003.

[11] G. J. Jang and T. W. Lee, “A maximum likelihood approach to single channel source separation,” *Journal of Machine Learning Research*, vol. 4, pp. 1365–1392, 2003.

[12] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, Massachusetts: MIT Press, 1990.

[13] H. Asari, B. A. Pearlmutter, and A. M. Zador, “Sparse representations for the cocktail party problem,” *J. Neurosci.*, vol. 26, no. 28, pp. 7477–90, 2006. [Online]. Available: http://www.jneurosci.org/cgi/content/abstract/26/28/7477

[14] B. A. Pearlmutter and R. K. Olsson, “Linear program differentiation for single-channel speech separation,” in *International Workshop on Machine Learning for Signal Processing*. Maynooth, Ireland: IEEE Press, Sept. 6–8 2006.

[15] S. W. Hainsworth, “Techniques for the automated analysis of musical audio,” Ph.D. dissertation, Department of Engineering, University of Cambridge, 2004.

[16] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by V1?” *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[17] O. Schwartz and E. P. Simoncelli, “Natural signal statistics and sensory gain control,” *Nat. Neurosci.*, vol. 4, no. 8, pp. 819–825, 2001.

[18] D. Klein, P. Konig, and K. P. Kording, “Sparse spectrotemporal coding of sounds,” *Journal on Applied Signal Processing*, vol. 7, pp. 659–667, 2003.

[19] S. S. Stevens, “On the phychophysical law,” *Psychol. Rev.*, vol. 64, pp. 153–181, 1957.

[20] D. D. Lee and H. S. Seung, “Learning the parts of objects with nonnegative matric factorization,” *Nature*, vol. 401, pp. 788–91, 1999.

[21] P. Smaragdis, “Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs,” in *Fifth International Conference on Independent Component Analysis*, ser. LNCS 3195. Granada, Spain: Springer-Verlag, Sept. 22–24 2004, pp. 494–9.

[22] ——, “Convolutive speech bases and their application to supervised speech separation,” *IEEE Transaction on Audio, Speech and Language Processing - to appear*, 2007.

[23] T. Virtanen, “Techniques for the automated analysis of musical audio,” Ph.D. dissertation, Institute of Signal Processing, Tampere University of Technology, 2006.

[24] H. B. Barlow, “Possible principles underlying the transformations of sensory messages,” in *Sensory Communication*, W. A. Rosenblith, Ed. MIT Press, 1961, pp. 217–234.

[25] M. S. Lewicki and T. J. Sejnowski, “Learning overcomplete representations,” *Neu. Comp.*, vol. 12, no. 2, pp. 337–65, 2000.

[26] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, “Dictionary learning algorithms for sparse representation,” *Neu. Comp.*, vol. 15, no. 2, pp. 349–96, 2003.

[27] D. L. Donoho and M. Elad, “Maximal sparsity representation via $l1$ minimization,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 100, pp. 2197–202, Mar. 2003.

[28] Y. Li, A. Cichocki, and S. Amari, “Analysis of sparse representation and blind source separation,” *Neu. Comp.*, vol. 16, no. 6, pp. 1193–234, 2004.

[29] R. Fletcher, “Semidefinite matrix constraints in optimization,” *SIAM J. Control and Opt.*, vol. 23, pp. 493–513, 1985.

[30] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, “Breast cancer diagnosis and prognosis via linear programming,” *Operations Research*, vol. 43, no. 4, pp. 570–7, July-Aug. 1995.

[31] P. S. Bradley, O. L. Mangasarian, and W. N. Street, “Clustering via concave minimization,” in *Adv. in Neu. Info. Proc. Sys. 9.* MIT Press, 1997, pp. 368–74.

[32] I. F. Gorodnitsky and B. D. Rao, “Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm,” *IEEE Trans. Signal Proccessing*, vol. 45, no. 3, pp. 600–16, 1997.

[33] M. Lewicki and B. A. Olshausen, "Inferring sparse, overcomplete image codes using an efficient coding framework," in *Advances in Neural Information Processing Systems 10.* MIT Press, 1998, pp. 815–821.

[34] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[35] T.-W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representations," *IEEE Signal Processing Letters*, vol. 4, no. 5, pp. 87–90, 1999.

[36] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neu. Comp.*, vol. 13, no. 4, pp. 863–82, Apr. 2001.

[37] B. A. Pearlmutter and A. M. Zador, "Monaural source separation using spectral cues," in *Fifth International Conference on Independent Component Analysis*, ser. LNCS 3195. Granada, Spain: Springer-Verlag, Sept. 22–24 2004, pp. 478–85.

[38] G. B. Dantzig, "Programming in a linear structure," 1948, uSAF, Washington D.C.

[39] S. I. Gass, *An Illustrated Guide to Linear Programming.* McGraw-Hill, 1970.

[40] R. Dorfman, "The discovery of linear programming," *Annals of the History of Computing*, vol. 6, no. 3, pp. 283–95, July–Sept. 1984.

[41] A. Griewank, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, ser. Frontiers in Appl. Math. Philadelphia, PA: SIAM, 2000, no. 19.

[42] R. E. Wengert, "A simple automatic derivative evaluation program," *Communications of the ACM*, vol. 7, no. 8, pp. 463–4, 1964.

[43] B. Speelpenning, "Compiling fast partial derivatives of functions given by algorithms," Ph.D. dissertation, Department of Computer Science, University of Illinois, Urbana-Champaign, Jan. 1980.

[44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back–propagating errors," *Nature*, vol. 323, pp. 533–6, 1986.

[45] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neu. Comp.*, vol. 7, no. 6, pp. 1129–59, 1995.

[46] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Mat. Stats.*, vol. 22, pp. 400–7, 1951.

[47] S. T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems*, 2001, pp. 793–799.

[48] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, "Super-human multi-talker speech recognition," in *ICSLP*, 2006.

[49] F. Bach and M. I. Jordan, "Blind one-microphone speech separation: A spectral learning approach," in *Advances in Neural Information Processing Systems 17*, 2005, pp. 65–72.

[50] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 120, pp. 2421–2424, 2006.

[51] D. P. W. Ellis and R. J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *ICASSP*, 2006.

[52] W. A. Yost, R. H. Dye, Jr., and S. Sheft, "A simulated "cocktail party" with up to three sound sources," *Percept Psychophys*, vol. 58, no. 7, pp. 1026–1036, 1996.

[53] F. L. Wightman and D. J. Kistler, "Headphone simulation of free-field listening. II: Psychophysical validation," *J Acoust Soc Am*, vol. 85, no. 2, pp. 868–878, 1989.

[54] A. Kulkarni and H. S. Colburn, "Role of spectral detail in sound-source localization," *Nature*, vol. 396, no. 6713, pp. 747–749, 1998.

[55] E. I. Knudsen and M. Konishi, "Mechanisms of sound localization in the barn owl," *Journal of Comparative Physiology*, vol. 133, pp. 13–21, 1979.

[56] E. M. Wenzel, M. Arruda, D. J. Kistler, and F. L. Wightman, "Localization using nonindividualized head-related transfer functions," *J Acoust Soc Am*, vol. 94, no. 1, pp. 111–123, 1993.

[57] P. M. Hofman and A. J. V. Opstal, "Bayesian reconstruction of sound localization cues from responses to random spectra," *Biol Cybern*, vol. 86, no. 4, pp. 305–316, 2002.

[58] H. Attias and C. Schreiner, "Temporal low-order statistics of natural sounds," in *Advances in Neural Information Processing Systems*, 1997.

[59] T. Nishino, Y. Nakai, K. Takeda, and F. Itakura, "Estimating head related transfer function using multiple regression analysis," *IEICE Trans. A*, vol. J84-A, no. 3, pp. 260–268, 2001, in Japanese.

[60] H. Farid and E. H. Adelson, "Separating reflections from images by use of independent components analysis," *J. Optical Society of America*, vol. 16, no. 9, pp. 2136–45, 1999.

[61] S. T. Rickard and F. Dietrich, "DOA estimation of many W-disjoint orthogonal sources from two mixtures using DUET," in *Proceedings of the 10th IEEE Workshop on Statistical Signal and Array Processing (SSAP2000)*, Pocono Manor, PA, Aug. 2000, pp. 311–4.

[62] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," in *Proc. of the European Conference on Computer Vision (ECCV)*, Prague, May 2004.

[63] *Fifth International Conference on Independent Component Analysis*, ser. LNCS 3195. Granada, Spain: Springer-Verlag, Sept. 22–24 2004.

Index