Institut Mines-Télécom

Audio and music signal processing

Roland Badeau
`roland.badeau@telecom-paris.fr`

Contents

Acronyms **5**

Mathematical notation **7**

1 Spectral and temporal modifications **9**
1 Introduction . . . 9
2 Signal models to define temporal and spectral distortions . . . 10
2.1 McAuley-Quatieri model . . . 10
2.2 Serra-Smith model . . . 11
3 Definitions and equivalences . . . 11
3.1 Temporal distortion . . . 12
3.2 Pitch modification . . . 12
3.3 Reciprocity . . . 12
4 Short-term Fourier transform . . . 12
4.1 Theoretical reminders . . . 13
5 Modifications using the phase-vocoder . . . 15
5.1 Instantaneous frequency . . . 15
5.2 Temporal distortion . . . 16
5.3 Pitch modification . . . 17
6 Pitch synchronous temporal method . . . 18
6.1 Modification of the time scale. . . . 18
6.2 Modification of the frequency scale. . . . 18
6.3 The circular memory technique . . . 19
6.3.1 The analog origin . . . 20
6.3.2 Digital implementation . . . 21
6.3.3 Modification of the duration by the technique of circular memory . . . 22

2 Nonnegative matrix factorization **24**
1 Introduction . . . 24
2 NMF theory and algorithms . . . 25
2.1 Criteria for computing the NMF model parameters . . . 25
2.2 Probabilistic frameworks for NMF . . . 26
2.2.1 Gaussian noise model . . . 26
2.2.2 Probabilistic latent component analysis . . . 27
2.2.3 Poisson NMF model . . . 27
2.2.4 Gaussian composite model . . . 27
2.2.5 α-stable NMF models . . . 28
2.2.6 Choosing a particular NMF model . . . 29
2.3 Algorithms for NMF . . . 29
2.3.1 Multiplicative update rules . . . 29

2.3.2 The EM algorithm and its variants . . . 30
2.3.3 Application of the EM algorithm to PLCA . . . 31
2.3.4 Application of the space-alternating generalized EM algorithm to the Gaussian composite model . . . 31
3 Advanced NMF models . . . 32
3.1 Regularizations . . . 32
3.1.1 Sparsity . . . 32
3.1.2 Group sparsity . . . 33
3.1.3 Harmonicity and spectral smoothness . . . 33
3.1.4 Inharmonicity . . . 34
3.2 Nonstationarity . . . 34
3.2.1 Time-varying fundamental frequencies . . . 34
3.2.2 Time-varying spectral envelopes . . . 35
3.2.3 Both types of variations . . . 36
4 Summary . . . 37

3 Audio source separation **38**
1 Introduction . . . 38
1.1 Typology of the mixture models . . . 38
1.2 Instantaneous linear mixtures . . . 39
1.3 Anechoic linear mixtures . . . 39
1.4 Convolutive mixtures . . . 39
2 Mathematical reminders . . . 40
2.1 Real random vectors . . . 40
2.2 Real Gaussian random vectors . . . 41
2.3 WSS vector processes . . . 41
2.4 Information theory . . . 42
3 Linear instantaneous mixtures . . . 42
3.1 Blind source separation (BSS) model . . . 42
3.1.1 Identifiability . . . 43
3.1.2 Linear separation of sources . . . 43
3.2 Independent component analysis (ICA) . . . 44
3.2.1 Whitening . . . 44
3.2.2 Contrast functions . . . 46
3.3 Second order methods . . . 47
3.3.1 Temporal coherence of source signals . . . 47
3.3.2 Non-stationarity of source signals . . . 48
3.4 Time-frequency methods . . . 49
3.4.1 Time-frequency representations . . . 49
3.4.2 Time-frequency source model . . . 50
3.4.3 Separation method . . . 50
4 Convolutive mixtures . . . 51
4.1 Source images . . . 51
4.2 Convolutive mixture model . . . 51
4.3 Time-frequency approach . . . 52
4.4 Independent component analysis . . . 53
4.5 Indeterminacies . . . 53
5 Under-determined mixtures . . . 54
5.1 Under-determined convolutive mixtures . . . 54
5.2 Separation via non-stationary filtering . . . 55
5.3 Stereophonic mixtures: separation based on sparsity . . . 56
5.3.1 Temporal sparsity . . . 56

5.3.2 Sparsity in a transformed domain . . . 57
5.3.3 DUET method . . . 58
6 Conclusion . . . 59

Licence de droits d'usage **64**

Practical works **65**
Pitch and temporal scale modifications . . . 65
Practical work on Non-Negative Matrix Factorization . . . 70
Practical work on audio source separation . . . 73

List of Figures

1.1 Short-term Fourier transform . . . 13
1.2 Bandpass filtering equivalent to an STFT channel . . . 14
1.3 Canonical resampling chain of factor L/M . . . 17
1.4 Modification of the duration of the signal by the TD-PSOLA method. At the top, the original signal, in the middle three short-term signals generated from two short-term signals centered around the first two analysis marks. At the bottom, the modified signal. . . . 19
1.5 Original: "il s'est". . . . 20
1.6 Signal stretched by factor 2. . . . 20
1.7 Modification of the pitch of the signal by the TD-PSOLA method. At the top, the original signal, in the middle three short-term signals generated from the three first analysis marks. At the bottom, the modified signal. The spacing of the synthesis marks is not identical to that of the analysis marks. 21
1.8 The circular memory technique . . . 21
1.9 Digital implementation . . . 22

2.1 Decomposition of "Au clair de la Lune" spectrogram . . . 25
2.2 Gaussian composite model (IS-NMF) by Févotte et al. [2009] . . . 28
2.3 Harmonic NMF model by Vincent et al. [2010] and Bertin et al. [2010] . . . 33
2.4 Decomposition of an excerpt from the first Prelude by Johann Sebastian Bach . . . 35
2.5 Jew's harp sound decomposed with a time-frequency activation . . . 36

3.1 Instantaneous linear mixtures . . . 39
3.2 Anechoic linear mixtures . . . 40
3.3 Convolutive mixtures . . . 40
3.4 Identifiability theorem: signals $y_k(t)$ are independent if and only if matrix $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{A}$ is non-mixing . 44
3.5 Pre-whitening for independent component analysis: $\boldsymbol{B} = \boldsymbol{U}^\top \boldsymbol{W}$ where $\boldsymbol{U}$ is a rotation matrix . . . 46
3.6 Narrow-band approximation . . . 53
3.7 Beamforming mixture model . . . 55
3.8 Under-determined mixtures: there is no matrix $\boldsymbol{B}(f)$ such that $\boldsymbol{B}(f)\,\boldsymbol{A}(f) = \boldsymbol{I}_K$ . . . 55
3.9 Sparsity in time domain . . . 57
3.10 Sparsity in TF domain . . . 57

Acronyms

AR *Autoregressive*

ARMA *Autoregressive Moving Average*

BSS *Blind Source Separation*

DFT *Discrete Fourier Transform*

DNN *Deep Neural Networks*

DTFT *Discrete Time Fourier Transform*

DUET *Degenerate Unmixing Estimation Technique*

EM *Expectation-Maximization*

ERB *Equivalent Rectangular Bandwidth*

EUC *Euclidean*

FIR *Finite Impulse Response*

FT *Fourier transform*

ICA *Independent Component Analysis*

IID *Independent and Identically Distributed*

IS *Itakura-Saito*

JADE *Joint Approximate Diagonalization of Eigenmatrices*

KL *Kullback-Leibler*

LPC *Linear Predictive Coding*

MA *Moving Average*

MAP *Maximum a Posteriori*

MDCT *Modified Discrete Cosine Transform*

ML *Maxmimum Likelihood*

MM *Majorization-Minimization*

MMSE *Minimum Mean Square Error*

MSE *Mean Square Error*

NMF *Non-negative Matrix Factorization*

OLA *OverLap-Add*

PDF *Probability Density Function*

PLCA *Probabilistic Latent Component Analysis*

PSD *Power Spectral Density*

PSOLA *Pitch-Synchronous OverLap-Add*

SAGE *Space Alternating Generalized EM*

SOBI *Second Order Blind Identification*

STFT *Short Time Fourier Transform*

TF *Time-Frequency*

WSS *Wide Sense Stationary*

Mathematical notation

$\mathbb{N}$ set of natural numbers

$\mathbb{Z}$ set of integers

$\mathbb{R}$ set of real numbers

$\mathbb{C}$ set of complex numbers

$\mathcal{R}e(.)$ real part

$\mathcal{I}m(.)$ imaginary part

x (normal font, lower case) scalar

$\boldsymbol{x}$ (bold font, lower case) vector

$\boldsymbol{A}$ (bold font, upper case) matrix

$\|.\|_2$ Euclidean norm of a real vector, or Hermitian norm of a complex vector

$\|.\|_F$ Frobenius norm of a matrix

$\overline{(.)}$ conjugate of a matrix / vector / number

$.^\top$ transpose of a matrix

$.^H$ conjugate transpose of a matrix

$.^\dagger$ pseudo-inverse of a matrix (if $\boldsymbol{A} \in \mathbb{R}^{M\times K}$ with $M \geq K$, $\boldsymbol{A}^\dagger = (\boldsymbol{A}^\top \boldsymbol{A})^{-1}\boldsymbol{A}^\top$)

Span(.) range space of a matrix

Ker(.) kernel of a matrix

dim(.) dimension of a vector space

rank(.) rank of a matrix

trace(.) trace of a square matrix

det(.) determinant of a square matrix

$.^\dagger$ pseudo-inverse of a matrix (if $\boldsymbol{A} \in \mathbb{R}^{M\times K}$ with $M \geq K$, $\boldsymbol{A}^\dagger = (\boldsymbol{A}^\top \boldsymbol{A})^{-1}\boldsymbol{A}^\top$)

diag(.) diagonal matrix formed from a vector of diagonal coefficients, or from a matrix with same diagonal entries

$\boldsymbol{I}_K$ $K \times K$ identity matrix

$H(\nu) = \sum_{t\in\mathbb{Z}} h(t)e^{-2\imath\pi\nu t}$ discrete time Fourier transform

$L^\infty(\mathbb{R}^M)$ Lebesgue space of essentially bounded functions on $\mathbb{R}^M$

$*$ convolution product between two sequences (scalars, but also matrices and vectors of appropriate dimensions)

$\mathbf{1}_A$ is 1 if A is true, or 0 if A is false

$\mathbb{E}[.]$ expected value of a random variable or vector

$\mathbb{H}[.]$ entropy of a random variable or vector

$\mathbb{I}[.]$ mutual information of the entries of a random vector

$\widehat{(.)}$ estimator of a parameter

CRB{.} Cramér-Rao bound

Chapter 1

Spectral and temporal modifications

This chapter is mostly a translation in English of a course handout by Bertrand David. It draws from passages of various documents and mainly from a work specifically dedicated to audio signal processing Kahrs and Brandenbourg [1998] (Chap. 7). It develops more particularly the phase vocoder-based methods and the temporal methods.

1 Introduction

The objective of these modifications, which correspond to usual needs in various fields of sound and speech processing, is to *independently* control the temporal, spectral and possibly formantic (slow variations of the spectrum) evolutions of the signal:

- temporal dilation: we want to modify duration scales without altering the spectral content and especially the pitch in the case of a harmonic signal,
- pitch variation: in the case of harmonic signals, we want to change the pitch of the sound, while retaining its temporal evolution (for example the prosodic flow in the case of speech) and especially its duration,
- formantic control: in the case of a pitch modification, one can choose to either modify the spectral scale as a whole (and therefore to move the formants or spectral envelope) or to keep the spectral envelope constant while transposing the line spectrum.

Applications where these types of independent modifications are numerous:

- synthesis by sampling of a wave table (musical sounds or speech segments Allen [1991]),
- post-synchronization: to perform a synchronization of sound and image,
- data compression Makhoul and El-Jaroudi [1986]
- reading for blind people: our inner reading is much faster than our diction. By shrinking durations we can allow blind people to increase their speed of browsing documents,
- learning foreign languages: slowing the speech flow is helpful,
- musical post-production: to mix several recordings it may be useful to speed up or slightly reduce the tempo. It may also be interesting to locally correct the precision of a voice or instrument.

We can classify in three types the methods that carry out these modifications:

- methods inspired by the circular reader head or modified radiocassette (we add / subtract portions of the signal Fairbanks et al. [1954]). These methods are called *temporal*,

- phase vocoder-based methods (*spectral* methods using the *Short Time Fourier Transform* (STFT) Schroeder et al. [1967], Portnoff [1976]),
- methods based on signal models (*Linear Predictive Coding* (LPC) Makhoul [1975], Sinus+Noise Serra and Smith [1990], Audio grains Jones and Parks [1988],...).

Temporal methods have resulted in many developments in digital signal processing: Synchronized *OverLap-Add* (OLA) method Roucos and Wilgus [1985], *Pitch-Synchronous OverLap-Add* (PSOLA) method Moulines and Charpentier [1990]. This last one uses a synchronized copy/deletion technique on the glottal impulses. This achieves a very good quality modification of the time scale *without resampling the signal*.

The PSOLA method can be adapted to perform formantic modifications by modifying the duration of the segments without modifying the position of the glottal impulses. In this way, we can transpose the spectral envelope without modifying the pitch or the duration, and thus modify the timbre of a voice (transform a male voice into a more female voice by example). Other techniques of formantic modifications use cepstral representations Cappé et al. [1995].

2 Signal models to define temporal and spectral distortions

A simple replay at 16 kHz of an audio signal sampled at 8 kHz is enough to convince us that the temporal and spectral expansions or compressions are interdependent. This dependence can be interpreted as a simple theoretical result on the *Fourier transform* (FT): the Heisenberg uncertainty relation translates this dependence in terms of supports, and the high frequency decrease in the FT is related to the regularity of the time signal. Therefore the definition of *independent* temporal and spectral distortions can only be obtained for well-defined signal *models*.

2.1 McAuley-Quatieri model

Speech production model. The most common and widely used speech signal model is that of a time-varying linear filter, excited by a harmonic source (in the case of voiced sounds) or by a stationary random process with flat spectrum (in the case of unvoiced sounds). In the present case, we consider the voiced case, for which this source is a sum of sinusoidal components whose frequencies are multiple of a fundamental frequency $f_0(t)$. This representation is equivalent to writing the source as a Dirac comb whose period depends on time.

Let $g_t(\tau)$ be the impulse response of the system at time t. The signal is then simply written as a function of the excitation signal $e(t)$:

$$x(t) = \int_{-\infty}^{+\infty} g_t(\tau) e(t-\tau) d\tau. \tag{1.1}$$

This non time-invariant system can be represented by a time-dependent frequency response:

$$G(t, f) = M(t, f) \exp j\varphi(t, f).$$

The temporal variations of g_t are linked to the articulatory movements and are considered slow compared to the fundamental period of the signal. On the other hand, these variations are assumed to be weak over the duration of the filter memory. The system is *quasi-stationary*.

For voiced speech, that is to say involving a periodic vibration of the vocal cords, the excitation signal writes:

$$e(t) = \sum_{k=-\infty}^{+\infty} \exp j\xi_k(t) \tag{1.2}$$

with

$$\xi_k'(t) = 2\pi f_k(t).$$

The quasi-stationary nature of g_t leads to a practical limitation of the support of this function to a dimension of the order of the system memory. The integral in expression 1.1 is therefore well-defined in practice. In the same way, the frequency support of speech is limited in practice and the discrete sum of expression 1.2 is in fact a finite

sum of $L(t)$ complex exponential terms. Taking into account the fact that f_0 varies little over the memory duration of the filter we can expand

$$\xi_k(t-\tau) \approx \xi_k(t) - 2\pi\tau k f_0(t)$$

in the vicinity of t (*i.e.* for τ lower than the memory of the filter). Then we obtain:

$$x(t) = \sum_{k=1}^{L(t)} M(t, f_k(t)) \exp j[\xi_k(t) + \varphi(t, f_k(t))] \tag{1.3}$$

Mc-Auley and Quatieri model. This model was introduced by McAulay and Quatieri around 1985 McAulay and Quatieri [1986], mainly for low rate speech coding. So it is related to the expression obtained in 1.3. It is however a little more general since it does not assume a necessarily harmonic relationship between the instantaneous frequencies. The signal is represented as a sum of sines whose frequencies, amplitudes and phases are controlled over time:

$$x(t) = \sum_{k=1}^{L(t)} A_k(t) \exp j\Psi_k(t) \qquad \text{with} \qquad \Psi'_k(t) = \omega_k(t) = 2\pi f_k(t) \tag{1.4}$$

where $A_k(t)$ is the amplitude at time t of sine k, $\Psi_k(t)$ is the *instantaneous phase* of this sine at time t and $f_k(t)$ is its *instantaneous frequency*. This decomposition is not unequivocal and we generally consider that the functions $A_k(t)$ and $\omega_k(t)$ have slow variations compared to the functions $\exp(j\Psi_k(t))$.

2.2 Serra-Smith model

This model was developed in the early 90s Serra and Smith [1990] in order to meet the need for an analysis/synthesis system accounting for the noisy component of music or speech. This component is very expensive to represent as a sum of sines. The proposed model is therefore an extension of that of MacAuley-Quatieri:

$$x(t) = \sum_{k=1}^{L(t)} A_k(t) \exp j\Psi_k(t) + b(t) \tag{1.5}$$

where $b(t)$ is a stationary random process filtered by a time-varying filter, like filter g_t presented above. Let h_t be this filter, we will then write, taking into account the causality of signals,

$$b(t) = \int_0^t h_t(\tau) u(t-\tau) d\tau \tag{1.6}$$

where $u(t)$ is a white stationary random process.

The complete analysis/modification/synthesis system includes

- an estimation phase of the deterministic components,
- a phase of linear interpolation of the amplitudes and cubic interpolation of the phases from one frame to another of the signal for these components,
- a subtraction of this deterministic part to get $b(t)$ for each frame,
- the application of a possibly distinct transformation algorithm for each of the two components,
- resynthesis.

3 Definitions and equivalences

All the definitions given here relate to a model of signal with sinusoidal components. They therefore apply to the McAuley-Quatieri model or to the deterministic part of the Serra-Smith model. The phases at $t = 0$ will be assumed to be zero for the sake of simplification (this term can be incorporated into the definition of the amplitudes).

3.1 Temporal distortion

We define the time distortion function using the new time scale τ and the original time scale t by:

$$\tau = T(t). \tag{1.7}$$

This function is continuous and bijective from $\mathbb{R}^+$ to $\mathbb{R}^+$. The modification of the signal's time scale $x(t)$ is then defined by

$$y(\tau) = \sum_{k=1}^{L(T^{-1}(\tau))} A_k(T^{-1}(\tau)) \exp(j\phi_k(\tau)) \tag{1.8}$$

The conservation of the frequency content then requires to maintain the values of the instantaneous frequencies, hence the relation:

$$\phi_k(\tau) = \int_0^{\tau} \omega_k(T^{-1}(u))du \tag{1.9}$$

3.2 Pitch modification

To modify the pitch of the signal $x(t)$ we build the signal:

$$y(t) = \sum_{k=1}^{L(t)} A_k(t) \exp(j\Phi_k(t)) \tag{1.10}$$

The alteration of the frequency content is defined using a function $\alpha(t)$ called frequency compression rate, according to the expression:

$$\Phi_k(t) = \int_0^{t} \alpha(u)\omega_k(u)du \tag{1.11}$$

3.3 Reciprocity

By a quick calculation we show that the operating sequence: $x \rightarrow x_1$ by time distortion ($\tau = T(t)$) followed a simple replay at a different temporal speed (*i.e.* without maintaining the frequency characteristics) $x_1(\tau) = y(v)$ with $v = T^{-1}(\tau)$ is equivalent to a frequency modification governed by the function $\alpha(t) = T'(t)$, that is to say:

$$y(v) = \sum_{k=1}^{L(v)} A_k(v) \exp j\Phi_k(v) \tag{1.12}$$

with

$$\Phi_k(t) = \int_0^{v} T'(u)\omega_k(u)du \tag{1.13}$$

This relationship is particularly useful in cases where the corresponding time distortion is a multiplying factor, like for instance $T(t) = 2t$. Then $T'(t)$ is constant and the replay operation is a simple replay of the signal obtained at a different rate (for example, for sampled signals, $F'_e = 2Fe$ in the previous case).

4 Short-term Fourier transform

The methods of analysis/synthesis and modification of sounds based on the use of the STFT are very common. The corresponding tool is usually called *phase vocoder*. It refers to the polar representation (module & phase) of the STFT.

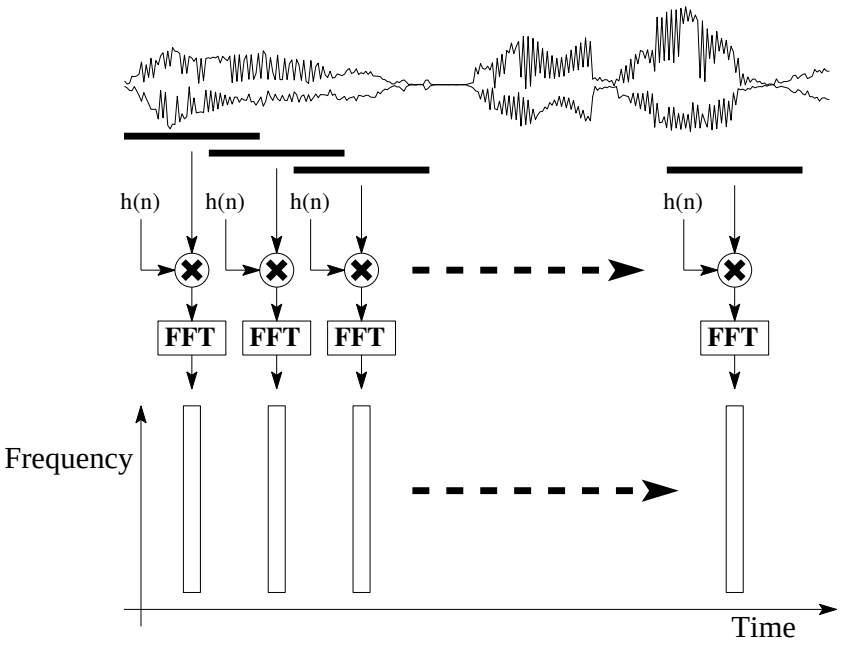


Figure 1.1: Short-term Fourier transform

4.1 Theoretical reminders

The block diagram of the STFT is represented in figure 1.1, as it is numerically computed. The principle is that of a sliding Fourier transform, performed on overlapping frames of the signal. Each frame is windowed by an analysis window. We will write the STFT of a digital signal $x(n)$ in the form

$$\tilde{X}(t_a, \nu) \triangleq \sum_{n\in\mathbb{Z}} x(n + t_a) w_a(n) e^{-j2\pi\nu n}. \tag{1.14}$$

w_a denotes the analysis window, most often of finished length, real and symmetrical. The analysis times are implicitly indexed by a natural integer u, that is $t_a = t_a(u), \quad u \in \mathbb{N}$. We preferred here a notation function of ν while a notation function of $e^{j2\pi\nu}$ would have been more consistent with the interpretation in terms of sliding Fourier transform, but this choice simplifies the expressions.

Interpretation. A quick calculation shows that, by defining $h(n) = w_a(-n)e^{j2\pi\nu_p n}$, the expression 1.14 can be written in the form of a convolution product:

$$\tilde{X}(t_a, \nu_p) = [x * h](t_a). \tag{1.15}$$

If $w_a(n)$ is a real and pair window of finite length, its FT $W_a(e^{j2\pi\nu})$ is real and even. The FT of h is then simply $H(e^{j2\pi\nu}) = W_a(e^{j2\pi(\nu-\nu_p)})$. An example of typical result is given in figure 1.2 for $\nu_p = 0.3$. This example shows that $\tilde{X}(t_a, \nu_p)$ performs a bandpass *Finite Impulse Response* (FIR) filtering around frequency ν_p. The characteristics of the filter are linked to that of the chosen analysis window. This interpretation is at the origin of the qualification of *bandpass convention* given to expression 1.14. There is another convention, called *low pass*, often used for its ease of handling calculations. We will, however, stick to the band pass convention because it corresponds to the practical realization.

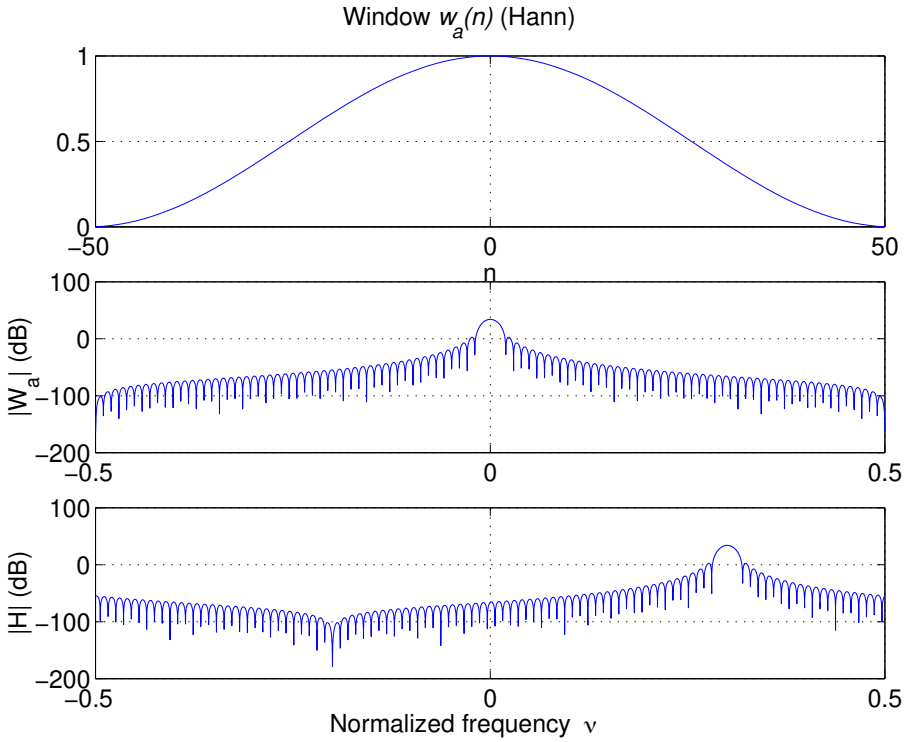


Figure 1.2: Bandpass filtering equivalent to an STFT channel

Discrete version of the STFT. In practice, the Fourier transform is evaluated using the *Discrete Fourier Transform* (DFT). This is equivalent to setting $\nu_p = p/N$ in the expression of $\tilde{X}(t_a, \nu_p)$. N is the order of the DFT. We thus obtain a discrete version of the STFT, i.e. sampled in frequency, i.e.

$$\tilde{X}(t_a, \nu_p) = \sum_{n=0}^{N-1} x(n + t_a) w_a(n) e^{-j2\pi \frac{pn}{N}}. \tag{1.16}$$

In order to avoid time aliasing, the length of the analysis windows will be less than or equal to N.

Modifications and problems posed. The modification of sounds by the phase vocoder involve obtaining a modified STFT from $\tilde{X}(t_a, \nu_p), \quad k = 0, \ldots, N-1$, then resynthesizing the signal. We denote by $t_s = t_s(u)$ the temporal synthesis marks. Hence the modification

$$\tilde{X}(t_a(u), \nu_p) \to Y(t_s(u), \nu_p).$$

The main difficulty encountered is that Y must satisfy strong conditions Portnoff [1976] to correspond to a well-defined original sequence. The solution to this problem is found in the least squares sense Moulines and Laroche [1995]. However, one can write the perfect reconstruction conditions in case no modification is performed (*i.e.* $t_s = t_a$ and $Y = \tilde{X}$).

Perfect reconstruction condition. The reverse operation of the analysis is carried out for the synthesis: from the stream of discrete spectra $\tilde{X}(t_a(u), \nu_p)$ we compute an inverse DFT and we reconstruct the signal by OLA. The result is given by

$$y(n) = \sum_u w_s(n - t_s(u)) y_w(n - t_s(u), t_s(u)) \tag{1.17}$$

with $t_s(u) = t_a(u)$ and

$$y_w(n, t_s(u)) = \frac{1}{N} \sum_{p=0}^{N-1} Y(t_s(u), \nu_p) e^{j2\pi\nu_p n}.$$

Taking $Y = \tilde{X}$ into account and substituting the expression 1.14 in 1.17, we show that $x(n) = y(n)$ is obtained by using the sufficient condition:

$$\sum_u w_a(n - t_a(u)) w_s(n - t_a(u)) = 1 \tag{1.18}$$

5 Modifications using the phase-vocoder

5.1 Instantaneous frequency

The transformations presented in section 3 require the calculation of the instantaneous frequencies $\omega_k(t)$ of each of the components of the sum 1.4. This calculation is carried out from two successive short term spectra $\tilde{X}(t_a(u), \nu_p)$ and $\tilde{X}(t_a(u+1), \nu_p)$ $p = 0, \ldots, N-1$, under certain conditions that ensure the existence of a solution.

Narrow-band condition. This first condition ensures the presence of *at most* one component per channel of the STFT. The substitution of the expression 1.4,

$$x(t) = \sum_{k=1}^{L(t)} A_k(t) \exp j\Psi_k(t),$$

into expression 1.14 of the STFT gives:

$$\tilde{X}(t_a(u), \nu_p) = \sum_{n=0}^{N-1} \sum_{k=1}^{L(n+t_a)} A_k(n + t_a) \exp(j\Psi_k(n + t_a)) w_a(n) e^{-j2\pi\nu_p n}$$

We then use the quasi-stationarity of the model, namely:

$$\begin{aligned} A_k(n + t_a) &\approx A_k(t_a) \\ \Psi_k(n + t_a) &\approx \Psi_k(t_a) + n\omega_k(t_a) \end{aligned}$$

and finally, by defining $\omega_k(t) = 2\pi f_k(t)$:

$$\tilde{X}(t_a(u), \nu_p) = \sum_{k=1}^{L(n+t_a)} A_k(t_a) \exp(j\Psi_k(t_a)) W_a(e^{j2\pi(\nu_p - f_k(t_a))}) \tag{1.19}$$

The narrow band condition leads to non-negligible values of $W_a(e^{j2\pi(\nu_p - f_k(t_a))})$ for at most one value of k. Let $k = l$ be this value if it exists. If we note f_c the cutoff frequency of the low-pass filter whose impulse response is $w_a(n)$, then the existence of l implies

$$|\nu_p - f_l(t_a)| \leq f_c \ ,$$

that is, the component number l is in the pass-band of the filter corresponding to the p-th channel of the STFT. The expression 1.19 is then reduced to the contribution of the l-th component alone:

$$\tilde{X}(t_a(u), \nu_p) = A_l(t_a) \exp(j\Psi_l(t_a)) W_a(e^{j2\pi(\nu_p - f_l(t_a))}) \tag{1.20}$$

If we assume that w_a is real and even, and therefore W_a is real and even, this expression is interpreted as follows: the phase of the STFT gives access to the instantaneous phases of the components of $x(t)$, up to an indeterminacy of a multiple of 2π, and the module of the STFT gives access to instantaneous amplitudes of $x(t)$, up to an amplitude factor due to filtering. We can therefore deduce the instantaneous frequencies of each component from the phase of the flow of short-term spectra, provided that the indeterminacy of 2π is removed.

Example: for a Hann analysis window of length L, the narrow-band condition applied to a line spectrum of harmonics (case of a voiced speech segment for example) leads to a spacing of spectral peaks at least equal to the bandwidth of the Fourier transform of the window, that is $4/L$. That results in $f_0 < 4/L$, i.e. a window length at least equal to 4 times the fundamental period.

Overlap condition. We will see here that the removing of the indeterminacy leads to a condition of minimal recovery of the analysis windows. Indeed, the phase difference between two successive analysis times, for the p-th channel of STFT is written, by defining $\Phi(t_a(u), \nu_p) = \arg \tilde{X}(t_a(u), \nu_p)$,

$$\begin{aligned}\Delta\Phi_p &= \Phi(t_a(u+1), \nu_p) - \Phi(t_a(u), \nu_p) = \Psi(t_a(u+1)) - \Psi(t_a(u)) \ [2\pi] \\ &= 2\pi f_l \Delta t_a(u) + 2n\pi \\ &= 2\pi(f_l - \nu_p)\Delta t_a(u) + 2\pi\nu_p \Delta t_a(u) + 2n\pi\end{aligned}$$

where n is a relative integer and $\Delta t_a(u) = t_a(u+1) - t_a(u)$. By taking $|\nu_p - f_l(t_a)| \leq f_c$ into account, the previous equation leads, if the condition 1.21 below is verified

$$f_c \Delta t_a(u) < 1/2, \tag{1.21}$$

to the inequality

$$|\Delta\Phi_p - 2\pi\nu_p \Delta t_a(u) - 2n\pi| < \pi \ ,$$

however there is one and only one value of n that verifies this property. This has removed the indeterminacy. In summary, we can thus get the value of the instantaneous frequency *in each STFT channel* by the following algorithm:

1. Calculation of the STFT at two successive times of analysis, which gives $\Delta\Phi_p$ for each channel ($p = 0, \ldots, N-1$),
2. For each channel, we look for the value $Q(n_0)$ of $Q(n) = \Delta\Phi_p - 2\pi\nu_p \Delta t_a - 2n\pi$ such that $|Q(n_0)| < \pi$,
3. we deduce the instantaneous frequencies by $f_l = \nu_p + \dfrac{Q(n_0)}{2\pi\Delta t_a}$.

Interpretation: inequality 1.21 leads to a minimum overlap condition between the analysis windows. Indeed, if we take for example a Hann window, for which we can estimate $f_c = 2/L$ where L is the window length, it becomes

$$\Delta t_a < \frac{L}{4}$$

which corresponds to a minimum recovery of 75% in analysis.

5.2 Temporal distortion

Once the instantaneous frequencies in each channel are deduced[1], the temporal signal distortion may be considered. In particular, instantaneous phases can be "unwound" so as to synchronize the modified STFT on the synthesis times. We then obtain the following modification algorithm, assuming that the analysis STFT $\tilde{X}(t_a(u), \nu_p)$ and the synthesis STFT $\tilde{Y}(t_s(u), \nu_p)$ are calculated for the index u, and given the time distortion law $T(t)$:

1. calculation of the STFT at time $t_a(u+1)$ and deduction of the instantaneous frequency $f_k(t_a(u))$ in each channel,
2. calculation of the new synthesis time $t_s(u) = T(t_a(u))$; in practice we take the whole part of this new instant
3. iteration of the synthesis instantaneous phase

$$\Phi_s(t_s(u+1), \nu_p) = \Phi_s(t_s(u), \nu_p) + 2\pi f_p(t_a(u))(t_s(u+1) - t_s(u))$$

4. calculation of the synthesis STFT for the index $u+1$ according to

$$\tilde{Y}(t_s(u+1), \nu_p) = A_p(t_a(u+1)) \exp j\Phi_s(t_s(u+1), \nu_p)$$

[1] in doing so, we assume that there is one and only one component by channel and therefore we can identify the indexes p (STFT channel) and k (components).

5.3 Pitch modification

The modification of pitch, or more generally of the frequency scale, is obtained either by temporal resampling or by spectral resampling.

Temporal resampling. This method is based on the reciprocity properties as seen in paragraph 3.3.

In the case of a constant frequency compression ratio $\alpha(t) = \alpha_0$, we obtain the desired modification by

1. a time stretch of factor α_0,
2. a replay at sampling frequency $\alpha_0 F_e$

where F_e is the original sampling frequency. This technique is equivalent to performing a resampling of factor $\alpha_0 = F'_e/Fe$ and playing at F_e. In this last case, it should however be noted that the time support is divided by α_0.

An extension of this resampling technique can be applied to obtain compression ratios $\alpha(t)$ variable over time. We use the canonical resampling method of digital signals by approaching α by a rational fraction at each analysis time $\alpha(t_a) = L(t_a)/M(t_a)$ and by performing the processing chain of figure 1.3 where $H(z)$ is a low-pass filter of cutoff frequency $\nu_c = \min(1/2L, 1/2M)$. We can therefore apply this processing to each frame of the signal and

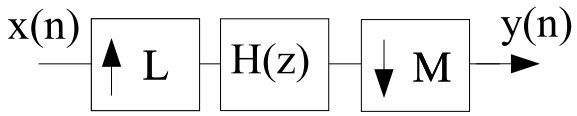


Figure 1.3: Canonical resampling chain of factor L/M

use synchronized analysis and synthesis times $t_a(u) = t_s(u)$. It should be noted that in this case, the phase vocoder is not used. This method can be quite demanding because it requires the calculation of a new interpolation filter H at each analysis step.

Spectral resampling. The phase vocoder allows a less greedy solution than the resampling in the case of variable compression ratios, by performing resampling in the frequency domain. This frequency resampling is performed by linear interpolation of the analysis short-term spectrum, that is

$$\begin{aligned} q &= \lfloor p/\alpha(t_a(u)) \rfloor \\ \mu_p &= p/\alpha(t_a(u)) - q \\ \tilde{Y}(t_s(u), \nu_q) &= (1-\mu_p)\tilde{X}(t_a(u), \nu_p) + \mu_p \tilde{X}(t_a(u), \nu_{p+1}) \end{aligned}$$

p and q are natural integers that index the channels of the STFT, they therefore vary from 0 to $N-1$. We note that this interpolation, if it presents no difficulty for compression rates greater than unity (pitch is increased), however requires completion of the high frequency spectrum for rates lower than unity (the synthesized sound is lower-pitched). One way to achieve this completion was suggested in Seneff [1982] and simply consists of copying the low frequency part of the spectrum into the missing part. This spectral copy gives good rendering for sampling frequencies of at least 16 kHz. In this case, the completion occurs at high frequencies where the sound has mainly unvoiced characteristics.

Finally, to carry out the modification, it is necessary to take into account the local modification of the time scale caused by the frequency modification. Indeed the phases of the synthesis STFT $\Phi_s(t_s(u), \nu_p) = \Phi_a(t_a(u), \nu_p)$ are now synchronized on synthesis times different from the analysis times:

$$\begin{aligned} \Phi_s(t_s(u+1), \nu_p) &= \Phi_s(t_s(u), \nu_p) + 2\pi f_p(t_a(u))\Delta t_a(u) \\ &= \Phi_s(t_s(u), \nu_p) + 2\pi\alpha(t_a(u)) f_p(t_a(u))\Delta t_s(u) \end{aligned}$$

We therefore see that the local analysis duration $\Delta t_a(u)$ has been divided by α in the synthesis. Let $\Delta t_s(u) = \Delta t_a(u)/\alpha(t_a(u))$. This corresponds to a virtual time distortion

$$T(t) = \int_0^t \alpha(w)^{-1} dw$$

To carry out the time scale modification, it is therefore necessary to finally apply a compensatory temporal distortion $D(t) = T^{-1}(t)$.

Note regarding the processing of spoken or singing voice. In the case of pitch modifications in spoken or singing voice, a direct transposition of the signal leads to the "Donald Duck" effect. Indeed, the transposition of the overall spectrum leads to a transposition of its envelope and therefore of the formants. The timbre is then severely modified and the voice acquires a nasal characteristic evoking the sound of the duck. This effect is also produced by the modification of the characteristic impedance of the medium caused by the mixed gaz breathed in by divers. A solution to overcome this defect consists in estimating the spectrum envelope before the processing (by LPC or direct modeling El-Jaroudi and Makhoul [1991]). The processing is then applied to the source signal (LPC residual for instance) then the obtained result is filtered to find the original spectral envelope, unchanged.

6 Pitch synchronous temporal method

This method, called Time Domain PSOLA, assumes that we process a speech signal whose period is known.

The idea Moulines and Charpentier [1990] is based on the assumption that the speech signal is made up of glottal pulses filtered by the vocal tract. We thus observe a succession of impulse responses, positioned at multiple times of the period (assumption of the time comb convoluted with the impulse response of the vocal tract).

We then define "analysis marks" synchronous with the fundamental frequency for the voiced parts, positioned on the waveform at each period. Scale modifications are then carried out as follows:

6.1 Modification of the time scale.

In order to modify the signal duration without altering the fundamental frequency, we will simply duplicate (for time stretching) or eliminate (for time compression) periods of the waveform, depending on the desired modification rate. So we are led to define synthesis marks also synchronous with the fundamental period, associated with the analysis marks (in a non-bijective way since some marks are duplicated or eliminated).

Short-term signals around each analysis mark are then extracted (by the use of a time window, by example a Hann window, of duration equal to two periods and centered on the analysis mark) and 'copied' around the corresponding synthesis marks, and the modified signal is obtained by a simple OLA method. Figure 1.4 illustrates the principle of this method for a local time stretch rate of 1.5.

We see that two periods of the original signal gave birth to three periods in the modified signal, which corresponds well to a time stretch but the duration of the period is not not modified (the spacing of the synthesis marks is the same as that of the analysis marks), the fundamental frequency of the signal is preserved. Figure 1.6 gives an application example to the sentence "il s'est" whose original is given in figure 1.5. We notice the unvoiced part in the center of the window (the sound 's'), separating the two voiced parts /i/ and /e/.

6.2 Modification of the frequency scale.

If we are able to position the analysis marks in the signal exactly at the start of each glottal wave (impulse response of the vocal tract occurring at each glottal closure), we can see that decreasing (resp. increasing) the time interval separating two consecutive analysis marks will increase (resp. decrease) the fundamental frequency, without the formants being modified (the impulse response is not modified, in particular its temporal decay and its resonance frequencies - the formants).

We are thus led to define synthesis marks corresponding to the modified value of the fundamental, and to associate them with the analysis marks, as previously. Since the synthesis marks are closer (elevation of the fundamental) or

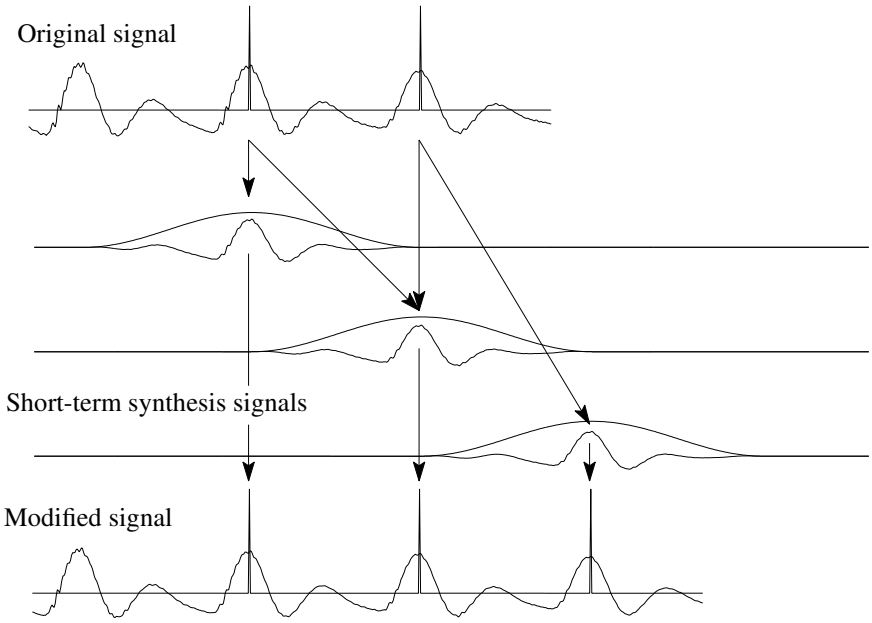


Figure 1.4: Modification of the duration of the signal by the TD-PSOLA method. At the top, the original signal, in the middle three short-term signals generated from two short-term signals centered around the first two analysis marks. At the bottom, the modified signal.

farther (lowering of the fundamental) than in the original signal, we have to duplicate or eliminate some marks in order to keep the duration of the signal. Figure 1.7 illustrates the principle of this method.

It can be seen that the synthesis marks being more spaced out than the analysis marks, the signal period is lengthened. In order to avoid an elongation of the signal, it is necessary to periodically eliminate some short-term signals.

When the signal no longer has a precise fundamental frequency (case of consonants for instance), the modification is carried out non-synchronously, until we find a region with a sharper fundamental.

The method described above is mainly applied to speech, and makes very good quality modifications. By its simplicity, it can be subject to a real-time implementation. However, its application to more complex sounds, or sounds devoid of "pitch" (case of music in general) poses serious problems.
The modifications of fundamental frequency are however very sensitive to the position of the analysis marks. To make the method more robust, the modifications of frequency scale can be performed in the frequency domain (FD-PSOLA method) Moulines and Charpentier [1990], Moulines and Laroche [1995].

For other methods based on very similar ideas, we can refer to Scott and Gerber [1972], Wayman and Wilson [1988], Malah [1979], Hardam [1990].

6.3 The circular memory technique

The circular memory technique is the simplest and most ancient time and frequency scale modification technique Benson [1988]. It is also a method operating in the time domain.

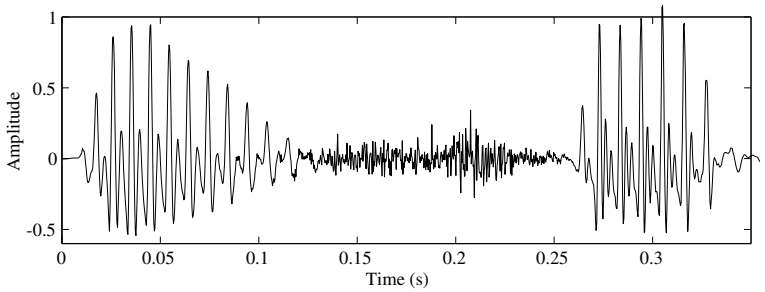


Figure 1.5: Original: "il s'est".

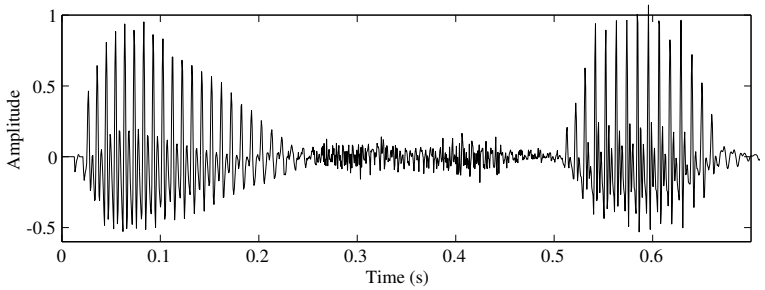


Figure 1.6: Signal stretched by factor 2.

6.3.1 The analog origin

This technique is derived from an analog system proposed in the 1950s Fairbanks et al. [1954]. It consists in using a tape recorder equipped with a rotating head. The closed loop tape wraps around half of the cylinder (as for the VCR and the DAT) and scrolls at constant speed. The cylinder is provided with two diametrically opposed reader heads whose signals are mixed with an identical gain. It is possible to control the direction of rotation and the speed of the cylinder.

When the cylinder is motionless, the tape scrolls in an identical manner in front of the recording head and in front of one of the reader heads. The signal read is therefore identical to the signal recorded (up to recording errors).

When the cylinder rotates in the opposite direction of the scrolling of the tape, the relative speed V_r of the scrolling tape with respect to the reader head is faster than its absolute scroll speed V_a. During the period of contact between the reader head and the tape, the signal is thus read faster than it has been recorded, which corresponds to a dilation of the frequency axis. The presence of two heads ensures the continuity thanks to a natural cross-fade (when a head leaves the band, the other approaches it, so that the total signal does not decrease). Note that some portions of the signal can be read *two or more times*, depending on the speed of rotation of the head. It is this rereading that keeps the signal duration.

Conversely, when the cylinder rotates in the scrolling direction of the band, the frequency content of the signal is contracted towards the origin since the tape is read at a slower speed than it is recorded. In this case, some portions of the signal may not be read at all.

The ratio of frequency homothety is expressed as:

$$\alpha = \frac{V_r}{V_a} = \frac{V_a + R\,\Omega_{cylinder}}{V_a}$$

where V_a is the tape scrolling speed in front of the recording head, V_r is the relative speed of the tape with respect to the reader head, $\Omega_{cylinder}$ is the speed of rotation of the cylinder in radians s^{-1}, and R is the radius of the cylinder.

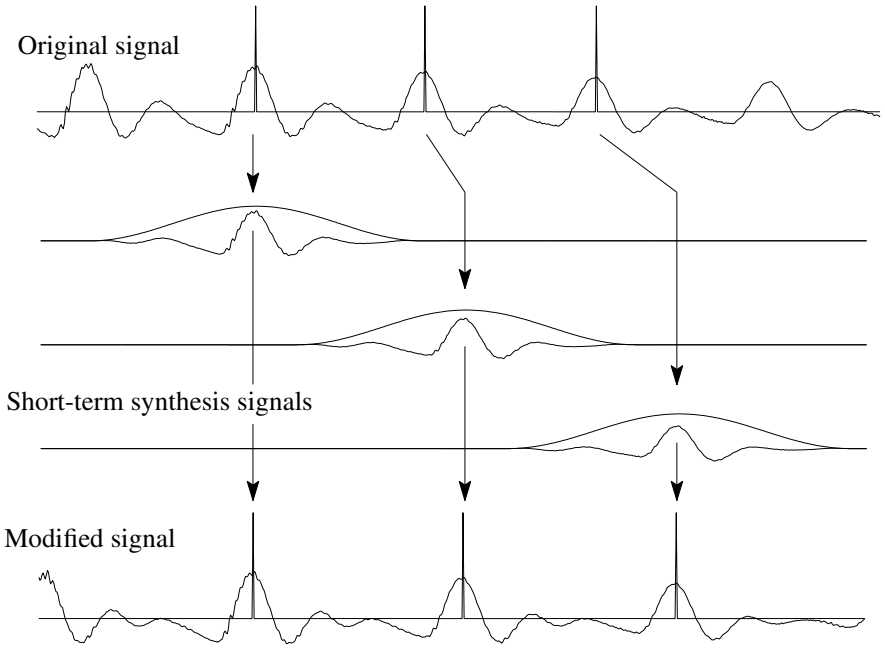


Figure 1.7: Modification of the pitch of the signal by the TD-PSOLA method. At the top, the original signal, in the middle three short-term signals generated from the three first analysis marks. At the bottom, the modified signal. The spacing of the synthesis marks is not identical to that of the analysis marks.

In all cases, the regular alternation of the two heads induces a periodic "noise" of frequency $\Omega_{cylinder}/\pi$.

Modifications in the signal time scale are obtained for instance by recording the signal a first time on the tape, then by replaying it with a tape scrolling speed multiplied by factor α. In the absence of rotation of the reader head, the signal pitch is of course multiplied by factor α, which we try to avoid. We therefore compensate for the pitch change by a proper rotation of the reader head.

6.3.2 Digital implementation

Most commercially available pitch modifiers are based on a digital realization of the system described above. The magnetic tape is replaced by a circular memory in which the input signal samples are placed. This circular memory is read by two diametrically opposite pointers.

For each sample written in the memory (every ΔT seconds), we advance the reading pointers by $\alpha\Delta T$ seconds, where α is the rate of change, and then we read a sample in memory. In general (for non-integer values of α), we

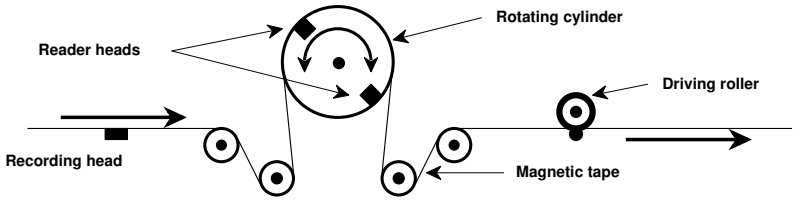


Figure 1.8: The circular memory technique

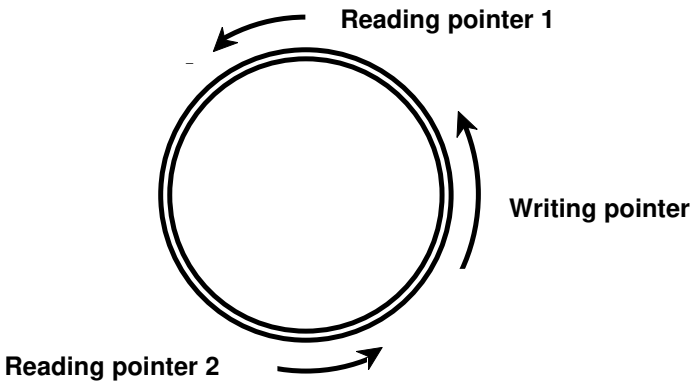


Figure 1.9: Digital implementation

find ourselves between two samples, and as in the case of the "flanger", it is necessary to calculate the signal value at this time. Here too, a simple linear interpolation is suitable.
Thus, the signal is read with a sampling frequency different from the one it was recorded with, which causes a modification of the frequency scale of rate α. A problem arises when the reading pointer catches up (when $\alpha > 1$) or is caught (when $\alpha < 1$) by the writing pointer. As in the analog equivalent, continuity is ensured by a mixture of the two pointers at the time when the encounter occurs ("cross-fade"): the sample read by the current reader pointer (e.g. pointer 1) is lowered while the one read by the other reader pointer (pointer 2) is increased. Finally, the second pointer becomes the current pointer, and keeps its maximum weighting, until the writing pointer gets close to it.

Implemented in this way, the pitch shifter has a behavior substantially equivalent to its analog counterpart (except that it is more easily configurable). Its implementation in real-time is not a particular problem, since it requires very few calculations.
Unfortunately, it produces artificial noise which comes from the periodic mixing of the two reader pointers. To try to improve the obtained quality, we seek to better combine the signals read by the two reader pointers, somewhat similarly to what is done in the synchronous methods. We can for example use the signal autocorrelation function to determine the most suitable location for the "cross-fading" Dattorro [1987], Laroche [1993].

6.3.3 Modification of the duration by the technique of circular memory

Like its analog counterpart, the circular buffer technique can also be used for temporal scale modification: if you have a pitch shifter with circular memory, and want to perform a "time scaling" of parameter α, just change the signal sampling frequency by a rate α, then process it with the pitch shifter. So, in order to slow down the signal twice, it suffices to oversample it twice. If we listen to the signal obtained at the original frequency, it will be twice as much long, but also at a lower octave. So just listen to it at the original frequency by inserting a pitch shifter at rate $\alpha = 2$.

We quickly realize that it is easier to do both operations jointly: the technique then consists in repeating or periodically eliminating signal portions so as to increase (or decrease) the duration. Viewed from this angle, this technique (which is called "splicing method") approximates a TD-PSOLA technique in which we would not know the value of the fundamental frequency. The artifacts inherent in this method, which come from the breaks in the periodicity of the signal during the repetitions or eliminations, can be considerably reduced by the use of methods based on the autocorrelation of the signal in order to optimize the length and location of signal portions to be duplicated or destroyed Lee [1972], Dattorro [1987], Laroche [1993], Roucos and Wilgus [1985], Sylvestre and

Kabal [1992], Verhelst and Roelands [1993].

Chapter 2

Nonnegative matrix factorization

Non-negative Matrix Factorization (NMF) refers to a set of techniques that have been used to model the spectra of sound sources in various audio applications, including source separation. Sound sources have a structure in time and frequency: music consists of basic units like notes and chords played by different instruments, speech consists of elementary units such as phonemes, syllables or words, and environmental sounds consist of sound events produced by various sound sources. NMF models this structure by representing the spectra of sounds as a sum of components with fixed spectrum and time-varying gain, so that each component in the model represents these elementary units in the sound.

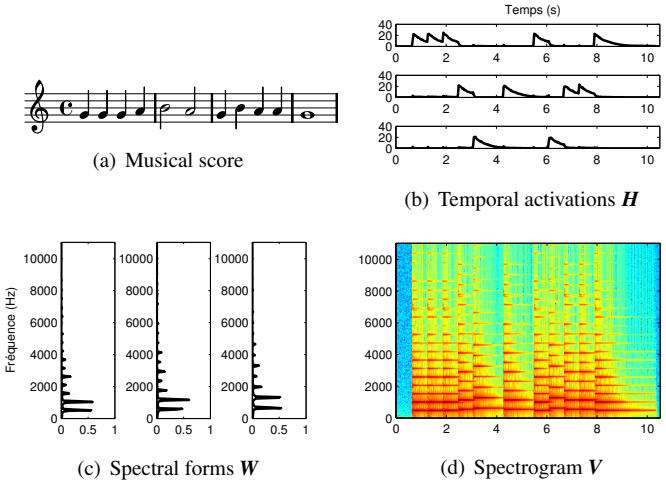
Modeling this structure is beneficial in source separation, since inferring the structure makes it possible to use contextual information for source separation. NMF is typically used to model the magnitude or power spectrogram of audio signals, and its ability to represent the structure of audio sources makes separation possible even in single-channel scenarios.

This chapter presents the use of NMF-based single-channel techniques. In Section 2, several deterministic and probabilistic frameworks for NMF are presented, along with various NMF algorithms. In Section 3, some advanced NMF models are introduced, including regularizations and nonstationary models. Finally, Section 4 summarizes the key concepts introduced in this chapter.

1 Introduction

The NMF was introduced by Lee and Sung to decompose non-negative two-dimensional data into a linear combination of elements in a dictionary [Lee and Seung, 1999].

Given a data matrix $\boldsymbol{V}$ of dimensions $F \times N$ whose coefficients are non-negative, the NMF problem consists in calculating an approximation $\widehat{\boldsymbol{V}}$ of matrix $\boldsymbol{V}$ truncated at rank $K < \min(F, N)$, expressed as a product $\widehat{\boldsymbol{V}} = \boldsymbol{W}\boldsymbol{H}$, where the two matrices $\boldsymbol{W}$ of dimensions $F \times K$ and $\boldsymbol{H}$ of dimensions $K \times N$ have non-negative entries. The columns of the matrix $\boldsymbol{W}$ form the elements of the dictionary and the rows of $\boldsymbol{H}$ contain the coefficients of the decomposition. The dimension K is generally chosen such that $FK + KN \ll FN$, so as to reduce the dimension of the data. The NMF can be considered as a supervised or unsupervised learning technique. In the case of supervised learning, the dictionary $\boldsymbol{W}$ is previously estimated from training data and matrix $\boldsymbol{H}$ only must be calculated from matrix $\boldsymbol{V}$. In the case of unsupervised learning, the two matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ must be computed jointly from $\boldsymbol{V}$. In audio applications, $\boldsymbol{V}$ is often the amplitude or power spectrogram, f denotes the frequency channel and n the time window. Figure 2.1 represents the musical score, spectrogram and unsupervised NMF of the melody of "Au clair de la lune". This figure clearly shows the interest of such a decomposition: it shows the spectra of musical notes in matrix $\boldsymbol{W}$ and their temporal activations in matrix $\boldsymbol{H}$, which makes it possible to consider both transcription and musical note separation applications.



(a) Musical score

(b) Temporal activations $\boldsymbol{H}$

(c) Spectral forms $\boldsymbol{W}$

(d) Spectrogram $\boldsymbol{V}$

Figure 2.1: Decomposition of "Au clair de la Lune" spectrogram (from [Bertin, 2009, pp. 40–41])

2 NMF theory and algorithms

Let $\boldsymbol{V} = [v(n,f)]_{fn}$ denote the $F \times N$ nonnegative time-frequency representation of a signal $x(t)$, where $n \in \{0,\dots,N-1\}$ is the time frame index, and $f \in \{0,\dots,F-1\}$ is the frequency index. For instance, if $\boldsymbol{X}$ is the $F \times N$ complex-valued STFT of x, then $\boldsymbol{V}$ can be the magnitude spectrogram $|\boldsymbol{X}|$ or the power spectrogram $|\boldsymbol{X}|^2$ [Smaragdis and Brown, 2003]. Other choices may include perceptual frequency scales, such as constant-Q [Fuentes et al., 2013] or *Equivalent Rectangular Bandwidth* (ERB) [Vincent et al., 2010] representations.

NMF [Lee and Seung, 1999] approximates the nonnegative matrix $\boldsymbol{V}$ with another nonnegative matrix $\widehat{\boldsymbol{V}} = [\widehat{v}(n,f)]_{fn}$ with entries $v(n,f) = \sigma_x^2(n,f)$, defined as the product

$$\widehat{\boldsymbol{V}} = \boldsymbol{W}\boldsymbol{H} \tag{2.1}$$

of a $F \times K$ nonnegative matrix $\boldsymbol{W}$ and a $K \times N$ nonnegative matrix $\boldsymbol{H}$ of lower rank $K < \min(F,N)$. This factorization can also be written $\widehat{\boldsymbol{V}} = \sum_k \widehat{\boldsymbol{V}}_k$, where $\widehat{\boldsymbol{V}}_k = [\widehat{v}_k(n,f)]_{fn} = \boldsymbol{w}_k \boldsymbol{h}_k^\top$, for all $k \in \{1,\dots,K\}$, is the k-th rank-1 matrix component. The k-th column vector $\boldsymbol{w}_k = [w_k(f)]_f$ can be interpreted as its spectrum, and the k-th row vector $\boldsymbol{h}_k^\top = [h_k(n)]_n$ comprises its *activation coefficients* over time. We also write $\widehat{v}_k(n,f) = w_k(f)h_k(n)$. All the parameters of the model, as well as the observed magnitude or power spectra, are elementwise nonnegative.

In this section, we first present the standard criteria for computing the NMF model parameters (Section 2.1), then we introduce probabilistic frameworks for NMF (Section 2.2), and we describe several algorithms designed for computing an NMF (Section 2.3).

2.1 Criteria for computing the NMF model parameters

Since NMF is a rank reduction technique, it involves an approximation: $\widehat{\boldsymbol{V}} \approx \boldsymbol{V}$. Computing the NMF can thus be formalized as an optimization problem: we want to minimize a measure $C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}})$ of divergence between matrices $\boldsymbol{V}$ and $\widehat{\boldsymbol{V}}$. The most popular measures in the NMF literature include the squared *Euclidean* (EUC) distance [Lee and Seung, 1999], the *Kullback-Leibler* (KL) divergence [Lee and Seung, 2001], and the *Itakura-Saito* (IS) divergence [Févotte et al., 2009]. The various NMFs computed by minimizing each of these three measures are named accordingly: EUC-NMF, KL-NMF, and IS-NMF. Actually, these three measures fall under the umbrella of *β-divergences* [Nakano et al., 2010, Févotte and Idier, 2011]. Formally, they are defined for any real-valued β as

$$C^\beta(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) = \sum_{nf} d^\beta(v(n,f) \mid \widehat{v}(n,f)), \tag{2.2}$$

where

- $\forall \beta \notin \{0,1\},\ d^{\beta}(x \mid y) = \frac{1}{\beta(\beta-1)}\left(x^{\beta} + (\beta-1)y^{\beta} - \beta x y^{\beta-1}\right)$,
- $\beta = 2$ corresponds to the squared EUC distance: $d^{\mathrm{EUC}}(x \mid y) = \frac{1}{2}|x-y|^2$,
- $\beta = 1$ corresponds to the KL divergence: $d^{\mathrm{KL}}(x \mid y) = x\log(\frac{x}{y}) - x + y$,
- $\beta = 0$ corresponds to the IS divergence: $d^{\mathrm{IS}}(x \mid y) = \frac{x}{y} - \log(\frac{x}{y}) - 1$.

It can be easily proved that $\forall x > 0$, the function $y \mapsto d^{\beta}(x \mid y)$ is convex with respect to y if and only if $\beta \in [1,2]$ [Févotte and Idier, 2011]. This means that minimizing $C^{\beta}(\boldsymbol{V} \mid \boldsymbol{WH})$ with respect to $\boldsymbol{H}$ with $\boldsymbol{W}$ fixed, or conversely with respect to $\boldsymbol{W}$ with $\boldsymbol{H}$ fixed, is a convex optimization problem if and only if $\beta \in [1,2]$. This convexity property is particularly convenient in a pretrained framework, where matrix $\boldsymbol{W}$ is fixed, and only matrix $\boldsymbol{H}$ is estimated from the observed data. Optimization algorithms are then insensitive to initialization, which might explain the better success of KL-NMF and EUC-NMF compared with IS-NMF in the NMF literature.

However, in the context of learning-free separation, whatever the value of β, minimizing $C^{\beta}(\boldsymbol{V} \mid \boldsymbol{WH})$ jointly with respect to $\boldsymbol{W}$ and $\boldsymbol{H}$ is not a convex optimization problem. Indeed, this factorization is not unique, since we also have $\widehat{\boldsymbol{V}} = \boldsymbol{W}'\boldsymbol{H}'$ with $\boldsymbol{W}' = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{\Pi}$ and $\boldsymbol{H}' = \boldsymbol{\Pi}^{\top}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}$, where $\boldsymbol{\Lambda}$ can be any $K \times K$ diagonal matrix with positive diagonal entries, and $\boldsymbol{\Pi}$ can be any $K \times K$ permutation matrix. Note that the nonuniqueness of the model is actually ubiquitous in source separation and is generally not considered as a problem: sources are recovered up to a scale factor and a permutation. In the case of NMF however, other kinds of indeterminacies may also exist [Laurberg et al., 2008]. Due to the existence of local minima, optimization algorithms become sensitive to initialization (cf. Section 2.3). In practice, with a random initialization, there is no longer any guarantee to converge to a solution that is helpful for source separation. For this reason, advanced NMF models have later been proposed to enforce some specific desired properties in the decomposition (cf. Section 3).

2.2 Probabilistic frameworks for NMF

Computing an NMF can also be formalized as a parametric estimation problem based on a probabilistic model, that involves both observed and latent (hidden) random variables. Typically, observed variables are related to matrix $\boldsymbol{V}$, whereas latent variables are related to matrices $\boldsymbol{W}$ and $\boldsymbol{H}$. The main advantages of using a probabilistic framework are the facility of exploiting some a priori knowledge that we may have about matrices $\boldsymbol{W}$ and $\boldsymbol{H}$, and the existence of well-known statistical inference techniques, such as the *Expectation-Maximization* (EM) algorithm.

Popular probabilistic models of nonnegative time-frequency representations include Gaussian models that are equivalent to EUC-NMF [Schmidt and Laurberg, 2008] (Section 2.2.1), and count models, such as the celebrated *Probabilistic Latent Component Analysis* (PLCA) [Shashanka et al., 2008] (Section 2.2.2) and the Poisson NMF model based on the Poisson distribution [Virtanen et al., 2008] (Section 2.2.3), that are both related to KL-NMF. However these probabilistic models do not account for the fact that matrix $\boldsymbol{V}$ has been generated from a time-domain signal $x(t)$. As a result, they can be used to estimate a nonnegative time-frequency representation, but they are not able to account for the phase, that is necessary to reconstruct a time-domain signal.

Other probabilistic frameworks focus on power or magnitude spectrograms, and intend to directly model the STFT $\boldsymbol{X}$ instead of the nonnegative time-frequency representation $\boldsymbol{V}$, in order to permit the resynthesis of time-domain signals. The main advantage of this approach is the ability to account for the phase and, in source separation applications, to provide a theoretical ground for using time-frequency masking techniques. Such models include Gaussian models that are equivalent to IS-NMF [Févotte et al., 2009] (Section 2.2.4) and the Cauchy NMF model based on the Cauchy distribution [Liutkus et al., 2015], that both fall under the umbrella of α-stable models [Liutkus and Badeau, 2015] (Section 2.2.5).

2.2.1 Gaussian noise model

A simple probabilistic model for EUC-NMF was presented by Schmidt and Laurberg [2008]: $\boldsymbol{V} = \boldsymbol{WH} + \boldsymbol{U}$, where matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ are seen as deterministic parameters, and the entries of matrix $\boldsymbol{U}$ are Gaussian, independent and identically distributed (i.i.d.): $u(n,f) \sim \mathcal{N}(u(n,f) \mid 0, \sigma_u^2)$. Then the log-likelihood of matrix $\boldsymbol{V}$ is $\log p(\boldsymbol{V} \mid$

$\boldsymbol{W},\boldsymbol{H}) = -\frac{1}{2\sigma_u^2}\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_F^2 + \text{cst} = -\frac{1}{\sigma_u^2}C^2(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) + \text{cst}$ where cst denotes a constant additive term, as defined in (2.2) with $\beta = 2$. Therefore *Maxmimum Likelihood* (ML) estimation of $(\boldsymbol{W},\boldsymbol{H})$ is equivalent to EUC-NMF. The main drawback of this generative model is that it does not enforce the nonnegativity of $\boldsymbol{V}$, whose entries might take negative values.

2.2.2 Probabilistic latent component analysis

PLCA [Shashanka et al., 2008] is a count model that views matrix $\widehat{\boldsymbol{V}}$ as a probability distribution (normalized so that $\sum_{nf}\widehat{v}(n,f) = 1$). The observation model is the following one: the probability distribution $P(n,f) = \widehat{v}(n,f)$ is sampled M times to produce M independent time-frequency pairs (n_m, f_m), $m \in \{1,\dots,M\}$. Then matrix $\boldsymbol{V}$ is generated as a histogram: $v(n,f) = \frac{1}{M}\sum_m \delta_{(n_m,f_m)}(n,f)$, that also satisfies $\sum_{nf}\widehat{z}(n,f) = 1$. The connection with NMF is established by introducing a latent variable k that is also sampled M times to produce k_m, $m \in \{1,\dots,M\}$. More precisely, it is assumed that (k_m, n_m) are first sampled together according to distribution $P(k,n) = h_k(n)$, and that f_m is then sampled given k_m according to distribution $P(f \mid k) = w_k(f)$, resulting in the joint distribution $P(n,f,k) = P(k,n)P(f \mid k) = \widehat{v}_k(n,f)$. Then $P(n,f)$ is the marginal distribution resulting from the joint distribution $P(n,f,k)$: $P(n,f) = \sum_k P(n,f,k) = \widehat{v}(n,f)$. Finally, note that another convenient formulation of PLCA is to simply state that $\boldsymbol{v}(n) \sim \mathcal{M}(\boldsymbol{v}(n) \mid \|\boldsymbol{v}(n)\|_1, \boldsymbol{W}\boldsymbol{h}(n))$ where $\boldsymbol{v}(n)$ and $\boldsymbol{h}(n)$ are the n-th columns of matrices $\boldsymbol{V}$ and $\boldsymbol{H}$, respectively, $\mathcal{M}$ denotes the multinomial distribution, and $\boldsymbol{h}(n)$ and the columns of matrix $\boldsymbol{W}$ are vectors that sum to 1.

In Section 2.3, it will be shown that this probabilistic model is closely related to KL-NMF. Indeed, the update rules obtained by applying the EM algorithm are formally equivalent to KL-NMF multiplicative update rules (cf. Section 2.3.3).

2.2.3 Poisson NMF model

The Poisson NMF model [Virtanen et al., 2008] is another count model, that assumes that the observed nonnegative matrix $\boldsymbol{V}$ is generated as the sum of K independent, nonnegative latent components $\boldsymbol{V}_k$. The entries $v_k(n,f)$ of matrix $\boldsymbol{V}_k$ are assumed independent and *Poisson*-distributed: $v_k(n,f) \sim \mathcal{P}(v_k(n,f) \mid \widehat{v}_k(n,f))$. The Poisson distribution is defined for any positive integer $\widehat{v}$ as $\mathcal{P}(\widehat{v} \mid \lambda) = \frac{e^{-\lambda}\lambda^{\widehat{v}}}{\widehat{v}!}$, where λ is the intensity parameter and $\widehat{v}!$ is the factorial of $\widehat{v}$. A nice feature of the Poisson distribution is that the sum of K independent Poisson random variables with intensity parameters λ_k is a Poisson random variable with intensity parameter $\lambda = \sum_k \lambda_k$. Consequently, $v(n,f) \sim \mathcal{P}(v(n,f) \mid \widehat{v}(n,f))$. The NMF model $\widehat{\boldsymbol{V}} = \boldsymbol{W}\boldsymbol{H}$ can thus be computed by maximizing $P(\boldsymbol{V} \mid \boldsymbol{W},\boldsymbol{H}) = \prod_{nf}\mathcal{P}(v(n,f) \mid \widehat{v}(n,f))$. It can be noticed that $\log P(\boldsymbol{V} \mid \boldsymbol{W},\boldsymbol{H}) = -C^1(\boldsymbol{V} \mid \widehat{\boldsymbol{V}})$, as defined in (2.2) with $\beta = 1$. Therefore ML estimation of $(\boldsymbol{W},\boldsymbol{H})$ is equivalent to KL-NMF.

2.2.4 Gaussian composite model

The Gaussian *composite model*introduced by Févotte et al. [2009] exploits a feature of the Gaussian distribution that is similar to that of the Poisson distribution: a sum of K independent Gaussian random variables of means μ_k and variances σ_k^2 is a Gaussian random variable of mean $\mu = \sum_k \mu_k$ and variance $\sigma^2 = \sum_k \sigma_k^2$. The main difference with the Poisson NMF model is that, instead of modeling the nonnegative time-frequency representation $\boldsymbol{V}$, IS-NMF aims to model the complex STFT $\boldsymbol{X}$, such that $\boldsymbol{V} = |\boldsymbol{X}|^2$. The observed complex matrix $\boldsymbol{X}$ is thus generated as the sum of K independent complex latent components $\boldsymbol{X}_k$ (cf. Fig 2.2). The entries of matrix $\boldsymbol{X}_k$ are assumed independent and complex Gaussian distributed: $x_k(n,f) \sim \mathcal{N}_c(x_k(n,f) \mid 0, \widehat{v}_k(n,f))$. Here the complex Gaussian distribution is defined as $\mathcal{N}_c(x \mid \mu,\sigma^2) = \frac{1}{\pi\sigma^2}\exp(-\frac{|x-\mu|^2}{\sigma^2})$, where μ and σ^2 are the mean and variance parameters. Consequently, $x(n,f) = \sum_k x_k(n,f) \sim \mathcal{N}_c(x(n,f) \mid 0, \widehat{v}(n,f))$. The NMF model $\widehat{\boldsymbol{V}} = \boldsymbol{W}\boldsymbol{H}$ can thus be computed by maximizing $p(\boldsymbol{X} \mid \boldsymbol{W},\boldsymbol{H}) = \prod_{nf}\mathcal{N}_c(x(n,f) \mid 0, \widehat{v}(n,f))$. It can be noticed that $\log p(\boldsymbol{X} \mid \boldsymbol{W},\boldsymbol{H}) = -C^0(\boldsymbol{V} \mid \widehat{\boldsymbol{V}})$, as defined in (2.2) with $\beta = 0$. Therefore ML estimation of $(\boldsymbol{W},\boldsymbol{H})$ is equivalent to IS-NMF.

In a source separation application, the main practical advantage of this Gaussian composite model is that the *Minimum Mean Square Error* (MMSE) estimates of the sources are obtained by time-frequency masking, in a way that is closely related to Wiener filtering. Indeed, suppose now that the observed signal $x(t)$ is the sum of J unknown source signals $s_j(t)$, so that $x(n,f) = \sum_j s_j(n,f)$, and that each source follows an IS-NMF model:

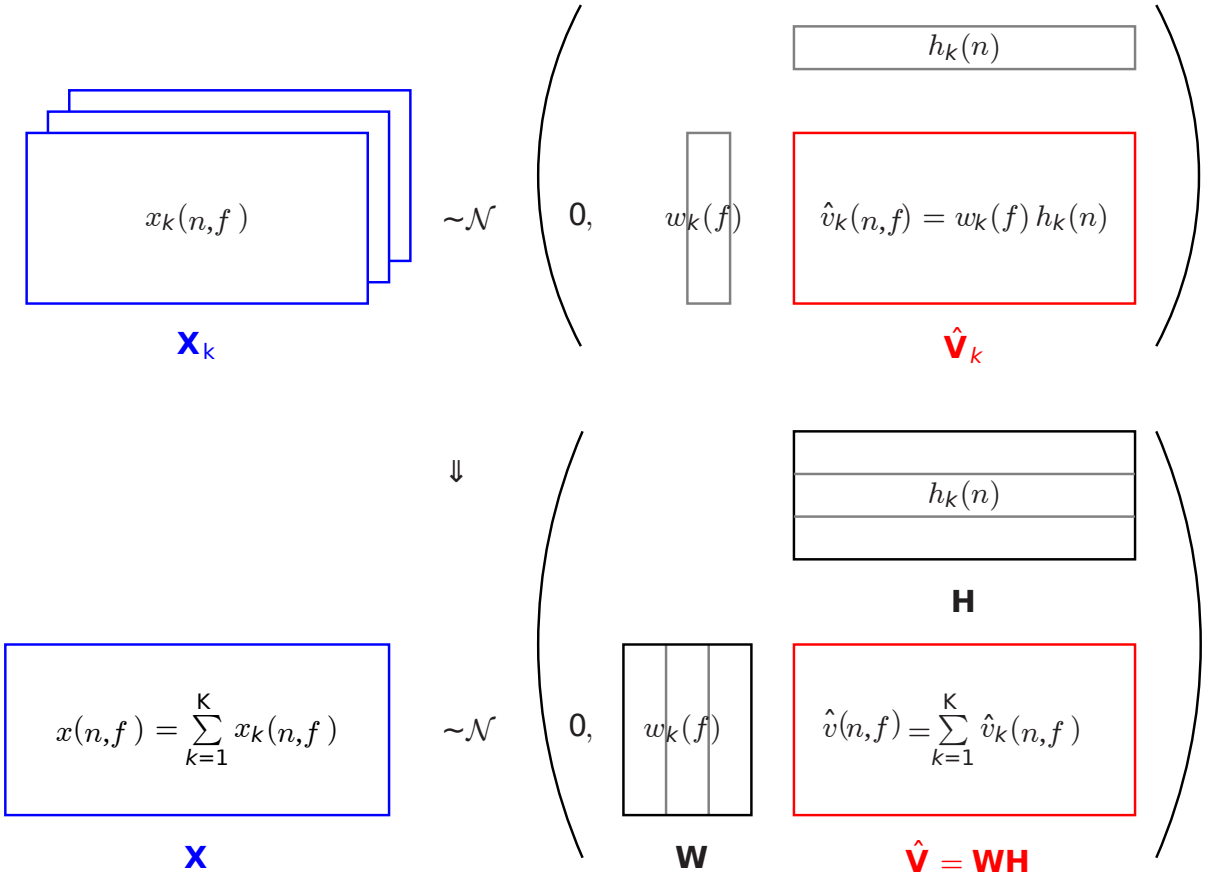


Figure 2.2: Gaussian composite model (IS-NMF) by Févotte et al. [2009]

$s_j(n,f) \sim \mathcal{N}_c(s_j(n,f) \mid 0, \widehat{v}_j(n,f))$ where $\widehat{v}_j(n,f) = \sigma^2_{s_j}(n,f)$ denotes the entries of matrix $\widehat{\boldsymbol{V}}_j = \boldsymbol{W}_j\boldsymbol{H}_j$. Then the minimum of the mean square error (MSE) criterion $\sum_{nf} \mathbb{E}\{|s_j(n,f) - \widehat{s}_j(n,f)|^2 \mid x(n,f)\}$ is reached when

$$\forall n,f,\ \widehat{s}_j(n,f) = \mathbb{E}\{s_j(n,f) \mid x(n,f)\} = m_j(n,f)x(n,f), \tag{2.3}$$

where the time-frequency mask $m_j(n,f)$ is defined as

$$m_j(n,f) = \frac{\widehat{v}_j(n,f)}{\sum_{j'} \widehat{v}_{j'}(n,f)}. \tag{2.4}$$

2.2.5 α-stable NMF models

Despite its nice features, the Gaussian model introduced in the previous section presents two drawbacks. Firstly, it amounts to assuming the additivity of the source power spectrograms, whereas several experimental studies have shown that the additivity of magnitude spectrograms is a better fit (see Liutkus and Badeau [2015] and references therein). Secondly, the IS divergence is not convex, which leads to increased optimization issues due to the existence of local minima. In order to circumvent these problems, a generalization of this model was introduced by Liutkus and Badeau [2015], based on isotropic complex α-*stable* distributions denoted $S\alpha S_c$, which stands for complex symmetric α-stable. This is a family of heavy-tailed probability distributions defined for any $\alpha \in]0,2]$, which do not have a closed-form expression, except in the particular cases $\alpha = 2$, which corresponds to the complex

Gaussian distribution, and $\alpha = 1$, which corresponds to the isotropic complex *Cauchy* distribution. In the general case, the distribution is defined by its characteristic function: $x \sim S\alpha S_c(x \mid \sigma) \Leftrightarrow \phi_z(\theta) = \mathbb{E}\{e^{\jmath \mathrm{Re}(\bar{\theta}x)}\} = e^{-\sigma^\alpha |\theta|^\alpha}$ for any complex-valued θ, where $\sigma > 0$ is the scale parameter (which corresponds to the standard deviation in the Gaussian case). These probability distributions enjoy the same nice feature shared by the Poisson and Gaussian distributions: a sum of K independent isotropic complex α-stable random variables of scale parameters σ_k is an isotropic complex α-stable random variable of scale parameter $\sigma^\alpha = \sum_k \sigma_k^\alpha$.

In this context, the observed STFT matrix $\boldsymbol{X}$ is again modeled as the sum of K independent latent components $\boldsymbol{X}_k$. The entries of matrix $\boldsymbol{X}_k$ are independent and isotropic complex α-stable: $x_k(n,f) \sim S\alpha S_c(x_k(n,f) \mid \sigma_k(n,f))$, where $\widehat{v}_k(n,f) = \sigma_k^\alpha(n,f)$ is called an α-spectrogram. Thus $x(n,f) = \sum_k x_k(n,f) \sim S\alpha S_c(x(n,f) \mid \sigma(n,f))$, with

$$\widehat{v}(n,f) = \sigma^\alpha(n,f) = \sum_k \sigma_k^\alpha(n,f) = \sum_k \widehat{v}_k(n,f). \tag{2.5}$$

When the distribution has a closed-form expression (i.e., $\alpha = 1$ or 2), the NMF model $\widehat{\boldsymbol{V}} = \boldsymbol{WH}$ can still be estimated in the ML sense, otherwise different inference methods are required. In the Cauchy case [Liutkus et al., 2015], it has been experimentally observed that Cauchy NMF is much less sensitive to initialization than IS-NMF and produces meaningful basis spectra for source separation.

In a source separation application, we again suppose that the observed signal $x(t)$ is the sum of J unknown source signals $s_j(t)$, so that $x(n,f) = \sum_j s_j(n,f)$, and that each source follows an isotropic complex α-stable NMF model: $s_j(n,f) \sim S\alpha S_c(s_j(n,f) \mid \widehat{v}_j^{1/\alpha}(n,f))$ where $\widehat{v}_j(n,f)$ denotes the entries of matrix $\widehat{\boldsymbol{V}}_j = \boldsymbol{W}_j\boldsymbol{H}_j$. Then the MSE criterion is no longer defined for all $\alpha \in (0,2]$, but in any case, the posterior mean $\widehat{s}_j(n,f) = \mathbb{E}\{s_j(n,f) \mid x(n,f)\}$ is still well-defined, and admits the same expression as in (2.3) and (2.4).

2.2.6 Choosing a particular NMF model

When choosing a particular NMF model for a given source separation application, several criteria may be considered, including the following ones:

Robustness to initialization: Cauchy NMF has proved to be more robust to initialization than all other probabilistic NMF models. Besides, in the context of pretrained source separation, the Gaussian noise model (related to EUC-NMF) and the PLCA/Poisson NMF models (related to KL-NMF) lead to a convex optimization problem with a unique minimum, which is not the case of the Gaussian composite model (related to IS-NMF).

Source reconstruction: only the α-stable NMF models, including IS-NMF and Cauchy NMF, provide a theoretical ground for using Wiener filtering in order to reconstruct time-domain signals.

Existence of closed-form update rules: ML estimation of the model parameters is tractable for all NMF models except α-stable models with $\alpha \neq 1, 2$.

2.3 Algorithms for NMF

In the literature, various algorithms have been designed for computing an NMF, including the famous multiplicative update rules [Lee and Seung, 2001], the alternated least squares method [Finesso and Spreij, 2004], and the projected gradient method [Lin, 2007]. In Section 2.3.1, we present the multiplicative update rules, that form the most celebrated NMF algorithm, and we summarize their convergence properties. Then in the following sections, we present some algorithms dedicated to the probabilistic frameworks introduced in Section 2.2.

2.3.1 Multiplicative update rules

The basic idea of *multiplicative update* rules is that the nonnegativity constraint can be easily enforced by updating the previous values of the model parameters by multiplication with a nonnegative scale factor. A heuristic way of deriving these updates consists in decomposing the gradient of the cost function $C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}})$, e.g., the β-divergence introduced in (2.2), as the difference of two nonnegative terms: $\nabla_{\boldsymbol{W}} C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) = \nabla^+_{\boldsymbol{W}} C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) - \nabla^-_{\boldsymbol{W}} C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}})$, where

$\nabla^+_{\boldsymbol{W}} C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) \geq 0$ and $\nabla^-_{\boldsymbol{W}} C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) \geq 0$, meaning that all the entries of these two matrices are nonnegative. Then matrix $\boldsymbol{W}$ can be updated as $\boldsymbol{W} \leftarrow \boldsymbol{W} \circ (\nabla^-_{\boldsymbol{W}} C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) / \nabla^+_{\boldsymbol{W}} C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}))^\eta$, where $\circ$ denotes elementwise matrix product, / denotes elementwise matrix division, the matrix exponentiation must be understood elementwise, and $\eta > 0$ is a stepsize similar to that involved in a gradient descent [Badeau et al., 2010]. The same update can be derived for matrix $\boldsymbol{H}$, and then matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ can be updated in turn, until convergence[1]. Note that the decomposition of the gradient as a difference of two nonnegative terms is not unique, and different choices can be made, leading to different multiplicative update rules. In the case of the β-divergence, the standard multiplicative update rules are expressed as follows [Févotte and Idier, 2011]:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \circ \left(\frac{(\boldsymbol{V} \circ (\boldsymbol{W}\boldsymbol{H})^{\beta-2})\boldsymbol{H}^\top}{(\boldsymbol{W}\boldsymbol{H})^{\beta-1}\boldsymbol{H}^\top} \right)^\eta \tag{2.6}$$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \circ \left(\frac{\boldsymbol{W}^\top(\boldsymbol{V} \circ (\boldsymbol{W}\boldsymbol{H})^{\beta-2})}{\boldsymbol{W}^\top(\boldsymbol{W}\boldsymbol{H})^{\beta-1}} \right)^\eta \tag{2.7}$$

where matrix division and exponentiation must be understood elementwise. By using the auxiliary function approach, Nakano et al. [2010] proved that the cost function $C^\beta(\boldsymbol{V} \mid \widehat{\boldsymbol{V}})$ is nonincreasing under these updates when the stepsize η is given by $\eta = \frac{1}{2-\beta}$ for $\beta < 1$, $\eta = 1$ for $1 \leq \beta \leq 2$, and $\eta = \frac{1}{\beta-1}$ for $\beta > 2$. In addition, Févotte and Idier [2011] proved that the same cost function is nonincreasing under (2.6)–(2.7) for $\eta = 1$ and for all $\beta \in [0, 2]$ (which includes the popular EUC, KL and IS-NMF). They also proved that these updates correspond to a *Majorization-Minimization* (MM) algorithm when the stepsize η is expressed as a given function of β, which is equal to 1 for all $\beta \in [1, 2]$, and they correspond to a majorization-equalization algorithm for $\eta = 1$ and $\beta = 0$. However, contrary to a widespread belief [Lee and Seung, 2001], the decrease of the cost function is not sufficient to prove the convergence of the algorithm to a local or global minimum. Badeau et al. [2010] analyzed the convergence of multiplicative update rules by means of Lyapunov's stability theory. In particular, it was proved that:

- There is $\eta^{\max} > 0$ such that these rules are exponentially or asymptotically stable for all $\eta \in (0, \eta^{\max})$. Moreover, $\forall \beta$, the upper bound $\eta^{\max}$ is such that $\eta^{\max} \in (0, 2]$, and if $\beta \in [1, 2]$, $\eta^{\max} = 2$.
- These rules are unstable if $\eta \notin [0, 2]$, $\forall \beta$.

In practice, the step size η permits us to control the convergence rate of the algorithm.

Note that, due to the nonuniqueness of NMF, there is a scaling and permutation ambiguity between matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ (cf. Section 2.1). Therefore, when $\boldsymbol{W}$ and $\boldsymbol{H}$ are to be updated in turn, numerical stability can be improved by renormalizing the columns of $\boldsymbol{W}$ (resp. the rows of $\boldsymbol{H}$), and scaling the rows of $\boldsymbol{H}$ (resp. the columns of $\boldsymbol{W}$) accordingly, so as to keep the product $\boldsymbol{W}\boldsymbol{H}$ unchanged.

Finally, a well-known drawback of most NMF algorithms is the sensitivity to initialization, that is due to the multiplicity of local minima of the cost function (cf. Section 2.1). Many initialization strategies were thus proposed in the literature [Cichocki et al., 2009]. In the case of IS-NMF multiplicative update rules, a *tempering* approach was proposed by Bertin et al. [2009]. The basic idea is the following one: since the β-divergence is convex for all $\beta \in [1, 2]$, but not for $\beta = 0$, the number of local minima is expected to increase when β goes from 2 to 0. Therefore a simple solution for improving the robustness to initialization consists in making parameter β vary from 2 to 0 over the iterations of the algorithm. Nevertheless, the best way of improving the robustness to initialization in general is to select a robust NMF criterion, such as that involved in Cauchy NMF (cf. Section 2.2.5).

2.3.2 The EM algorithm and its variants

As mentioned in Section 2.2, one advantage of using a probabilistic framework for NMF is the availability of classical inference techniques, whose convergence properties are well-known. Classical algorithms used in the NMF literature include the EM algorithm [Shashanka et al., 2008], the space-alternating generalized EM algorithm [Févotte et al., 2009], variational Bayesian (VB) inference [Badeau and Drémeau, 2013], and *Markov chain Monte Carlo* [Simsekli and Cemgil, 2012].

[1] This iterative algorithm can stop, e.g., when the decrease of the β-divergence, or when the distance between the successive iterates of matrices $\boldsymbol{W}$ and $\boldsymbol{H}$, goes below a given threshold.

Below, we introduce the basic principles of the space-alternating generalized EM algorithm [Fessler and Hero, 1994], which includes the regular EM algorithm as a particular case. We then apply the EM algorithm to the PLCA framework described in Section 2.2.2, and the space-alternating generalized EM algorithm to the Gaussian composite model described in Section 2.2.4.

Consider a random observed dataset $\mathcal{X}$, whose probability distribution is parameterized by a parameter set $\boldsymbol{\theta}$, that is partitioned as $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_k$. The space-alternating generalized EM algorithm aims to estimate parameters $\boldsymbol{\theta}_k$ iteratively, while guaranteeing that the likelihood $p(\mathcal{X} \mid \boldsymbol{\theta})$ is nondecreasing over the iterations. It requires choosing for each subset $\boldsymbol{\theta}_k$ a *hidden data* space which is complete for this particular subset, i.e., a latent dataset $\mathcal{X}_k$ such that $p(\mathcal{X}, \mathcal{X}_k \mid \boldsymbol{\theta}) = p(\mathcal{X} \mid \mathcal{X}_k, \{\boldsymbol{\theta}_{k'}\}_{k' \neq k}) p(\mathcal{X}_k \mid \boldsymbol{\theta})$. The algorithm iterates over both the iteration index and over k. For each iteration and each k, it is composed of an expectation step (*E-step*) and a maximization step (*M-step*):

- E-step: evaluate $Q_k(\boldsymbol{\theta}_k) = \mathbb{E}\{\log p(\mathcal{X}_k \mid \boldsymbol{\theta}_k, \{\boldsymbol{\theta}_{k'}\}_{k' \neq k}) \mid \mathcal{X}, \boldsymbol{\theta}\}$;
- M-step: compute $\boldsymbol{\theta}_k = \operatorname{argmax}_{\boldsymbol{\theta}_k} Q_k(\boldsymbol{\theta}_k)$.

The regular EM algorithm corresponds to the particular case $K = 1$, where $\mathcal{X}$ is a deterministic function of the complete data space.

2.3.3 Application of the EM algorithm to PLCA

Shashanka et al. [2008] applied the EM algorithm to the PLCA model described in Section 2.2.2. The observed dataset is $\mathcal{X} = \{n_m, f_m\}_m$, the parameter set is $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{H}\}$, and the complete data space is $\{n_m, f_m, k_m\}_m$. Then:

- The E-step consists in computing $P(k \mid n, f) = \frac{P(n,f,k)}{\sum_{k'} P(n,f,k')} = \frac{\widehat{v}_k(n,f)}{\widehat{v}(n,f)}$, that appears in the expression $Q(\boldsymbol{\theta}) = \sum_{nf} v(n,f) \sum_k P(k \mid n, f) \log\left(w_k(f) h_k(n)\right)$.
- The M-step consists in maximizing $Q(\boldsymbol{\theta})$ with respect to $w_k(f)$ and $h_k(n)$, subject to $\forall k$, $\sum_f w_k(f) = 1$ and $\sum_{kn} h_k(n) = 1$. Given that $\sum_{nf} v(n,f) = 1$, we get:

$$h_k(n) \leftarrow \frac{\sum_f v(n,f) P(k \mid n, f)}{\sum_{k'n'f} v(n',f) P(k' \mid n', f)} = h_k(n) \sum_f w_k(f) \frac{v(n,f)}{\widehat{v}(n,f)}, \tag{2.8}$$

$$w_k(f) \leftarrow \frac{\sum_n v(n,f) P(k \mid n, f)}{\sum_{nf'} v(n,f') P(k \mid n, f')} = \frac{\tilde{w}_k(f)}{\sum_{f'} \tilde{w}_k(f')}, \tag{2.9}$$

  where $\tilde{w}_k(f) = w_k(f) \sum_n h_k(n) \frac{v(n,f)}{\widehat{v}(n,f)}$.

It is easy to check that this algorithm is identical to the multiplicative update rules for KL divergence, as described in (2.6)–(2.7) with $\eta = \beta = 1$, up to a scaling factor in $\boldsymbol{H}$ due to the normalization of matrix $\boldsymbol{V}$ [Shashanka et al., 2008].

2.3.4 Application of the space-alternating generalized EM algorithm to the Gaussian composite model

Févotte et al. [2009] applied the *Space Alternating Generalized EM* (SAGE) algorithm to the Gaussian composite model described in Section 2.2.4. The observed dataset is $\mathcal{X} = \boldsymbol{X}$, the k-th parameter set is $\boldsymbol{\theta}_k = \{\boldsymbol{w}_k, \boldsymbol{h}_k\}$, and the k-th complete latent dataset is $\mathcal{X}_k = \boldsymbol{X}_k$. Then:

- The E-step consists in computing $\boldsymbol{V}_k = \frac{\widehat{\boldsymbol{V}}_k^2}{\widehat{\boldsymbol{V}}^2} \circ \boldsymbol{V} + \frac{\widehat{\boldsymbol{V}}_k \circ (\widehat{\boldsymbol{V}} - \widehat{\boldsymbol{V}}_k)}{\widehat{\boldsymbol{V}}}$, that appears in the expression $Q_k(\boldsymbol{\theta}_k) = -C^0(\boldsymbol{V}_k \mid \widehat{\boldsymbol{V}}_k)$, where criterion C^0 was defined in (2.2) with $\beta = 0$.
- The M-step computes $h_k(n) \leftarrow \frac{1}{F} \sum_f \frac{v_k(n,f)}{w_k(f)}$ and $w_k(f) \leftarrow \frac{1}{N} \sum_n \frac{v_k(n,f)}{h_k(n)}$.

Note that it has been experimentally observed by Févotte et al. [2009] that this space-alternating generalized EM algorithm converges more slowly than the IS-NMF multiplicative update rules described in (2.6)–(2.7) for $\eta = 1$ and $\beta = 0$.

3 Advanced NMF models

The basic NMF model presented in Section 1 has proved successful for addressing a variety of audio source separation problems. Nevertheless, the source separation performance can still be improved by exploiting prior knowledge that we may have about the source signals. For instance, we know that musical notes and voiced sounds have a harmonic spectrum (or, more generally, an inharmonic or a sparse spectrum), and that both their spectral envelope and their temporal power profile have smooth variations. On the opposite, percussive sounds rather have a smooth spectrum, and a sparse temporal power profile. It may thus be desirable to impose properties such as *harmonicity*, *smoothness*, and *sparsity* on either the spectral matrix $\boldsymbol{W}$ or the activation matrix $\boldsymbol{H}$ in the NMF $\widehat{\boldsymbol{V}} = \boldsymbol{W}\boldsymbol{H}$. For that purpose, it is possible to apply either hard constraints, e.g., by parameterizing matrix $\boldsymbol{W}$ or $\boldsymbol{H}$, or soft constraints, e.g., by adding a regularization term to the criterion (2.2), or by introducing the prior distributions of $\boldsymbol{W}$ or $\boldsymbol{H}$ in the probabilistic frameworks introduced in Section 2.2 (Bayesian approach). Examples of such regularizations are described in Section 3.1. Note that another possible way of exploiting prior information is to use a predefined dictionary $\boldsymbol{W}$ trained on a training dataset.

In other respects, audio signals are known to be nonstationary, therefore it is useful to consider that some characteristics such as the fundamental frequency or the spectral envelope may vary over time. Such nonstationary models will be presented in Section 3.2.

3.1 Regularizations

In this section, we present a few examples of NMF regularizations, including sparsity (Section 3.1.1), group-sparsity (Section 3.1.2), harmonicity and spectral smoothness (Section 3.1.3), and inharmonicity (Section 3.1.4).

3.1.1 Sparsity

Since NMF is well suited to the problem of separating audio signals formed of a few repeated audio events, it is often desirable to enforce the sparsity of matrix $\boldsymbol{H}$.

The most straightforward way of doing so is to add to the NMF criterion a sparsity-promoting regularization term. Ideally, sparsity is measured by the ℓ_0 norm, which counts the number of nonzero entries in a vector. However, optimizing a criterion involving the ℓ_0 norm raises intractable combinatorial issues. In the optimization literature, the ℓ_1 norm is often preferred, because it is the tightest convex relaxation of the ℓ_0 norm. Therefore the criterion $C^\beta(\boldsymbol{V} \mid \widehat{\boldsymbol{V}})$ in (2.2) may be replaced with

$$C(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) = C^\beta(\boldsymbol{V} \mid \widehat{\boldsymbol{V}}) + \lambda \sum_k \|\boldsymbol{h}_k\|_1, \tag{2.10}$$

where $\lambda > 0$ is a tradeoff parameter to be tuned manually, as suggested, e.g., by Hurmalainen et al. [2015].

However, if the NMF is embedded in a probabilistic framework such as those introduced in Section 2.2, sparsity is rather enforced by introducing an appropriate prior distribution of matrix $\boldsymbol{H}$. In this case, $\boldsymbol{H}$ is estimated by maximizing its posterior probability given $\boldsymbol{V}$, or equivalently the *Maximum a Posteriori* (MAP) criterion $\log p(\boldsymbol{V} \mid \boldsymbol{W}, \boldsymbol{H}) + \log p(\boldsymbol{H})$, instead of the log-likelihood $\log p(\boldsymbol{V} \mid \boldsymbol{W}, \boldsymbol{H})$. For instance, Kameoka et al. [2009] consider a generative model similar to the Gaussian noise model presented in Section 2.2.1, where the sparsity of matrix $\boldsymbol{H}$ is enforced by means of a generalized Gaussian prior:

$$p(\boldsymbol{H}) = \prod_{kn} \frac{1}{2\Gamma\left(1 + \frac{1}{p}\right)\sigma} e^{-\frac{|h_k(n)|^p}{\sigma^p}}, \tag{2.11}$$

where $\Gamma(\cdot)$ denotes the gamma function, parameter p promotes sparsity if $0 < p < 2$, and the case $p = 2$ corresponds to the standard Gaussian distribution.

In the PLCA framework described in Section 2.2.2, the entries of $\boldsymbol{H}$ are the discrete probabilities $P(k, n)$. By noticing that the entropy $\mathbb{H}\{\boldsymbol{H}\}$ of this discrete probability distribution is related to the sparsity of matrix $\boldsymbol{H}$ (the lower $\mathbb{H}\{\boldsymbol{H}\}$, the sparser $\boldsymbol{H}$), a suitable sparsity-promoting prior is the so-called *entropic prior* [Shashanka et al., 2008], defined as $p(\boldsymbol{H}) \propto e^{-\beta\mathbb{H}\{\boldsymbol{H}\}}$, where $\beta > 0$.

3.1.2 Group sparsity

Now suppose that the observed signal $x(t)$ is the sum of J unknown source signals $s_j(t)$ for $j \in \{1, \ldots, J\}$, whose spectrograms $\boldsymbol{V}_j$ are approximated as $\widehat{\boldsymbol{V}}_j = \boldsymbol{W}_j \boldsymbol{H}_j$, as in Section 2.2.4. Then the spectrogram $\boldsymbol{V}$ of $x(t)$ is approximated with the NMF $\sum_j \widehat{\boldsymbol{V}}_j = \boldsymbol{W}\boldsymbol{H}$, where $\boldsymbol{W} = [\boldsymbol{W}_1, \ldots, \boldsymbol{W}_J]$ and $\boldsymbol{H} = [\boldsymbol{H}_1^\top, \ldots, \boldsymbol{H}_J^\top]^\top$. In this context, it is natural to expect that if a given source j is inactive at time n, then all the entries in the n-th column of $\boldsymbol{H}_j$ are zero. Such a property can be enforced by using *group sparsity*. A well-known group sparsity regularization term is the mixed ℓ_2-ℓ_1 norm: $\|\boldsymbol{H}\|_{2,1} = \sum_{jn} \|\boldsymbol{h}_j(n)\|_2$ where $\boldsymbol{h}_j(n)$ is the n-th column of matrix $\boldsymbol{H}_j$, as suggested, e.g., by Hurmalainen et al. [2015]. Indeed, the minimization of the criterion (2.10) involving this regularization term tends to enforce sparsity over both n and j, while ensuring that the whole vector $\boldsymbol{h}_j(n)$ gets close to zero for most values of n and j.

Lefevre et al. [2011] proposed a group sparsity prior for the IS-NMF probabilistic framework described in Section 2.2.4. The idea is to consider a prior distribution of matrix $\boldsymbol{H}$ such that all vectors $\boldsymbol{h}_j(n)$ are independent: $p(\boldsymbol{H}) = \prod_{jn} p(\boldsymbol{h}_j(n))$. Each $p(\boldsymbol{h}_j(n))$ is chosen so as to promote near-zero vectors. Then, as in Section 3.1.1, the NMF parameters are estimated in the MAP sense: $(\boldsymbol{W}, \boldsymbol{H}) = \operatorname{argmax}_{\boldsymbol{W},\boldsymbol{H}} \log p(\boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{H}) + \sum_{jn} \log p(\boldsymbol{h}_j(n))$, where $p(\boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{H})$ was defined in Section 2.2.4.

3.1.3 Harmonicity and spectral smoothness

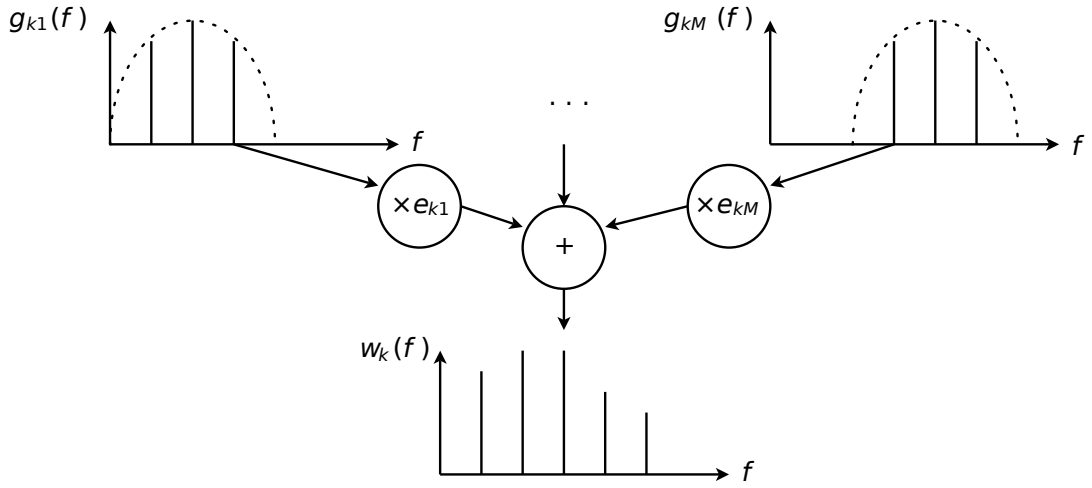


Figure 2.3: Harmonic NMF model by Vincent et al. [2010] and Bertin et al. [2010]

Contrary to sparsity, harmonicity in matrix $\boldsymbol{W}$ is generally enforced as a hard constraint, by using parametric models, whose parameter set includes the fundamental frequency. For instance, Vincent et al. [2010] and Bertin et al. [2010] parameterized the spectrum vector $\boldsymbol{w}_k$ as a nonnegative linear combination of M narrowband, harmonic spectral patterns (cf. Fig. 2.3): $w_k(f) = \sum_m e_{km} g_{km}(f)$, where all spectral patterns $\{g_{km}(f)\}_m$ share the same fundamental frequency $\nu_k^0 > 0$, have smooth spectral envelopes and different spectral centroids, so as to form a filterbank-like decomposition of the whole spectrum, and $\{e_{km}\}_m$ are the nonnegative coefficients of this decomposition. In this way, it is guaranteed that $w_k(f)$ is a harmonic spectrum of fundamental frequency ν_k^0, with a smooth spectral envelope. If the signal of interest is a music signal, then the order K and the fundamental frequencies ν_k^0 can typically be preset according to the semitone scale; otherwise they have to be estimated along with the other parameters. Two methods were proposed for estimating the coefficients e_{km} and the activations in matrix $\boldsymbol{H}$ from the observed spectrogram: a space-alternating generalized EM algorithm based on a Gaussian model [Bertin et al., 2010] (cf. Section 2.3.2) and multiplicative update rules (with a faster convergence speed) based either on the IS divergence [Bertin et al., 2010], or more generally on the β-divergence [Vincent et al., 2010].

Hennequin et al. [2010] proposed a similar parameterization of the spectrum vector $\boldsymbol{w}_k$, considering that harmonic spectra are formed of a number M of distinct partials:

$$w_k(f) = \sum_m a_k^m \, g_{km}(f), \tag{2.12}$$

where $a_k^m \geq 0$, $g_{km}(f) = g\left(\nu_f - \nu_k^m\right)$, $\nu_f = \frac{f}{F} f_s$ and $\nu_k^m = m\,\nu_k^0$, and $g(\cdot)$ is the spectrum of the analysis window used for computing the spectrogram. Multiplicative update rules based on the β-divergence were proposed for estimating this model. Since this parametric model does not explicitly enforce the smoothness of the spectral envelope, a regularization term promoting this smoothness was added to the β-divergence, resulting in a better decomposition of music spectrograms [Hennequin et al., 2010].

3.1.4 Inharmonicity

When modeling some string musical instruments such as the piano or the guitar, the harmonicity assumption has to be relaxed. Indeed, because of the bending stiffness, the partial frequencies no longer follow an exact harmonic progression, but rather a so-called *inharmonic* progression:

$$\nu_k^m = m\nu_k^0 \sqrt{1 + Bm^2}, \tag{2.13}$$

where m is the partial index, $B > 0$ is the inharmonicity coefficient, and $\nu_k^0 > 0$ is the fundamental frequency of vibration of an ideal flexible string [Rigaud et al., 2013]. Then the spectrum vector $\boldsymbol{w}_k$ can be parameterized as in (2.12), and all parameters, including the inharmonicity coefficient B, can be estimated by minimizing the β-divergence criterion by means of multiplicative update rules. However, it was observed that the resulting algorithm is very sensitive to initialization (cf. Section 2.3.1). In order to improve the robustness to initialization, the exact parameterization of frequencies ν_k^m in (2.13) was relaxed by considering these frequencies as free parameters, and by adding the following regularization term to the β-divergence criterion: $\sum_{km} |\nu_k^m - m\nu_k^0 \sqrt{1 + Bm^2}|^2$.

3.2 Nonstationarity

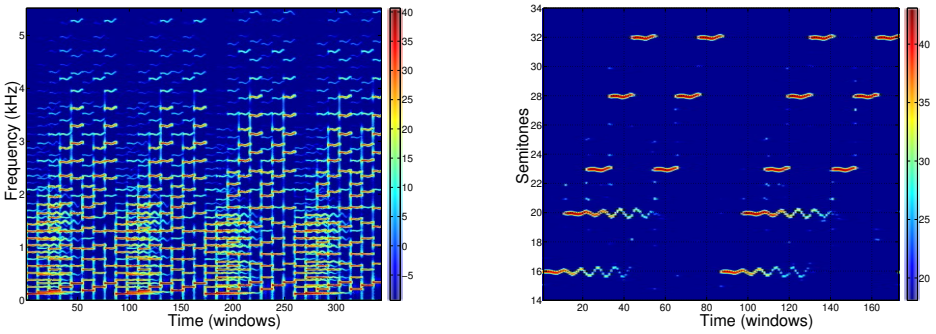
In the previous Section 3.1, several methods have been presented for enforcing the harmonicity and the spectral smoothness of vectors $\boldsymbol{w}_k$ in matrix $\boldsymbol{W}$, by means of either hard or soft constraints. All these methods assumed that the spectra of the audio events forming the observed spectrogram are stationary. However, many real audio signals are known to be nonstationary: the fundamental frequency, as well as the spectral envelope, may vary over time. In this section, we present some models that aim to represent such nonstationary signals, by allowing the fundamental frequency and spectral envelope parameters to vary over time.

3.2.1 Time-varying fundamental frequencies

In Section 3.1.3, a harmonic parameterization of vector $\boldsymbol{w}_k$ was described in (2.12). Hennequin et al. [2010] proposed a straightforward generalization of this model by making the spectral coefficient w_k also depend of time n: $w_k(n, f) = \sum_m a_k^m \, g\left(\nu_f - \nu_k^m(n)\right)$, resulting in a a spectrogram model that is a generalization of NMF: $\widehat{v}(n, f) = \sum_k \widehat{v}_k(n, f)$ with $\widehat{v}_k(n, f) = w_k(n, f)\, h_k(n)$.

Multiplicative update rules based on the β-divergence were proposed for estimating this extended model, along with several regularization terms designed to better fit music spectrograms [Hennequin et al., 2010].

We used this model to decompose the spectrogram of an excerpt (first 4 bars) of the first prelude of Johann Sebastian Bach played by a synthesizer (figure 2.4(a)). A slight vibrato has been added in the played notes to emphasize the variable fundamental frequency estimation. The decomposition uses 72 spectral shapes distributed every semitone. The figure 2.4(b) represents the activations and the fundamental frequencies ν_{kn}^0 obtained. The notes of the prelude appear very clearly, with the vibrato effect.



(a) Original spectrogram

(b) Representation of the activations including the fundamental frequencies (first two measurements of the extract). The color scale is in dB.

Figure 2.4: Decomposition of an excerpt from the first Prelude by Johann Sebastian Bach (figure extracted from Hennequin et al. [2010])

3.2.2 Time-varying spectral envelopes

Beyond the fundamental frequency, the spectral envelope of freely vibrating harmonic tones (such as those produced by a piano or a guitar) is not constant over time: generally, the upper partials decrease faster than the lower ones. Besides, some sounds such as those produced by a didgeridoo are characterized by a strong resonance in the spectrum that varies over time. Similarly, every time fingerings change on a wind instrument, the shape of the resonating body changes and the resonance pattern is different.

In order to properly model such sounds involving time-varying spectra, Hennequin et al. [2011] proposed to make the activations in vector $\boldsymbol{h}_k$ not only depend on time, but also on frequency, in order to account for the temporal variations of the spectral envelope of vector $\boldsymbol{w}_k$. More precisely, the activation coefficient $h_k(n,f)$ is parameterized according to an *Autoregressive Moving Average* (ARMA) model:

$$h_k(n,f) = \sigma_k^2(n) \left| \frac{1 + \sum_{n'} \beta_k(n,n') e^{-2j\pi n' f/F}}{1 - \sum_{n'} \alpha_k(n,n') e^{-2j\pi n' f/F}} \right|^2 \tag{2.14}$$

where $\sigma_k^2(n)$ is the variance parameter at time n, $\alpha_k(n,n')$ denotes the *Autoregressive* (AR) coefficients, and $\beta_k(n,n')$ the *Moving Average* (MA) coefficients, at time n. Then the NMF model is generalized in the following way: $\widehat{v}(n,f) = \sum_k \widehat{v}_k(n,f)$ with $\widehat{v}_k(n,f) = w_k(f)\, h_k(n,f)$. This model is also estimated by minimizing a β-divergence criterion. All parameters, including the ARMA coefficients, are computed by means of multiplicative update rules, without any training. Note that even though the ARMA model of $h_k(n,f)$ in (2.14) is nonnegative, the model coefficients $\alpha_k(n,n')$ and $\beta_k(n,n')$ are not necessarily nonnegative, which means that the multiplicative update rules introduced in Section 2.3.1 were generalized so as to handle these coefficients appropriately [Hennequin et al., 2011].

This algorithm allowed to efficiently represent non-stationary sounds with strong spectral variations, such as the Jew's harp sounds. The Jew's harp is an instrument made of a vibrating metal rod. This rod is placed in the mouth of the instrumentalist who modulates the sound with his mouth. It is thus a harmonic sound (with a fixed fundamental frequency) with a strong resonance varying with time (see the spectrogram in figure 2.5(a)). The decomposition obtained with our algorithm (using a single component) is shown in figures 2.5(b) and 2.5(c) : it shows well the harmonic shape of the spectrum on the one hand and the temporal variations of the resonance on the other hand.

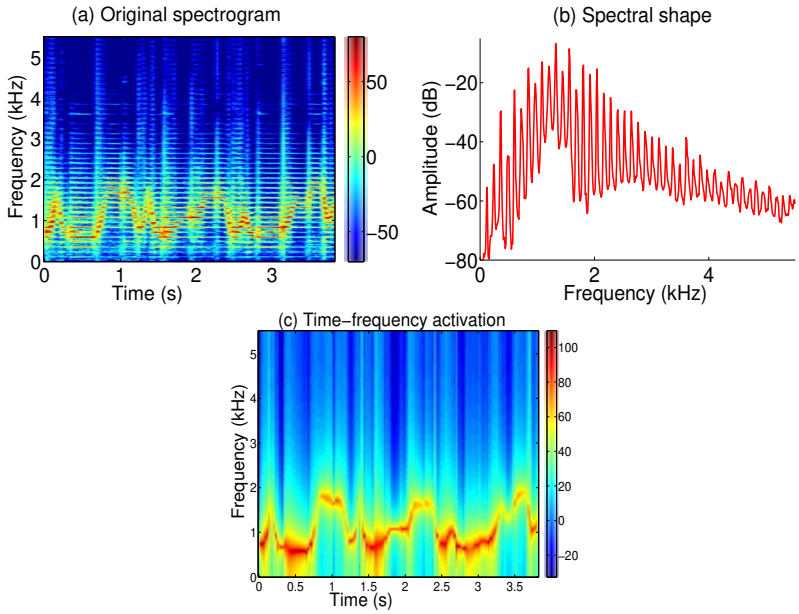


Figure 2.5: Jew's harp sound decomposed with a time-frequency activation parameterized by an ARMA filter of order (1,1) (figure extracted from Hennequin [2010])

3.2.3 Both types of variations

In order to account for the temporal variations of the fundamental frequency and the spectral envelope jointly, Fuentes et al. [2013] proposed a model called harmonic adaptive latent component analysis. This models falls within the scope of the PLCA framework described in Section 2.2.2: matrix $\widehat{\boldsymbol{V}}$ is viewed as a discrete probability distribution $P(n,f) = \widehat{v}(n,f)$, and the spectrogram $\boldsymbol{V}$ is modeled as a histogram: $v(n,f) = \frac{1}{M}\sum_m \delta_{(n_m,f_m)}(n,f)$, where $\{(n_m,f_m)\}_m$ are i.i.d. random vectors distributed according to $P(n,f)$.

In practice, the time-frequency transform used to compute $\boldsymbol{V}$ is a constant-Q transform. Because this transform involves a log-frequency scale, pitch shifting can be approximated as a translation of the spectrum along this log-frequency axis. Therefore all the notes produced by source j at time n are approximately characterized by a unique template spectrum modeled by a probability distribution $P(\mu \mid j,n)$ (where μ is a frequency parameter), which does not depend on the fundamental frequency. However this distribution depends on time n in order to account for possible temporal variations of the spectral envelope. Besides, the variation of the pitch f_0 of source j over time n is modeled by a probability distribution $P(f_0 \mid j,n)$. Hence the resulting distribution of the shifted frequency $f = \mu + f_0$ is $P(f \mid j,n) = \sum_{f_0} P(f - f_0 \mid j,n)P(f_0 \mid j,n)$. Finally, the presence of source j at time n is characterized by a distribution $P(j,n)$. Therefore the resulting spectrogram corresponds to the probability distribution $P(n,f) = \sum_{jf_0} P(f - f_0 \mid j,n)P(f_0 \mid j,n)P(j,n)$.

In order to enforce both the harmonicity and the smoothness of the spectral envelope, the template spectrum $P(\mu \mid j,n)$ is modeled in the same way as in the first paragraph of Section 3.1.3, as a nonnegative linear combination of K narrowband, harmonic spectral patterns $P(\mu \mid k)$: $P(\mu \mid j,n) = \sum_k P(\mu \mid k)P(k \mid j,n)$, where $P(k \mid j,n)$ is the nonnegative weight of pattern k at time n for source j. Finally, the resulting harmonic adaptive latent component analysis model is expressed as

$$P(n,f) = \sum_{f_0 k j} P(f - f_0 \mid k)P(k \mid j,n)P(f_0 \mid j,n)P(j,n). \tag{2.15}$$

4 Summary

In this chapter, we have shown that NMF is a very powerful model for representing speech and music data. We have presented the mathematical foundations, and described several probabilistic frameworks and various algorithms for computing an NMF. We have also presented some advanced NMF models that are able to more accurately represent audio signals, by enforcing properties such as sparsity, harmonicity and spectral smoothness, and by taking the nonstationarity of the data into account. We have shown that coupled factorizations make it possible to exploit some extra information we may have about the observed signal, such as the musical score. Finally, we have presented several methods that perform dictionary learning for NMF.

The benefits of NMF in comparison with other separation approaches are the capability of performing unsupervised source separation, learning source models from a relatively small amount of material (especially in comparison with *Deep Neural Networks* (DNN)), and easily implementing and adapting the source models and the algorithms. The main downside is the complexity of iterative NMF algorithms. Note that beyond source separation, NMF models have also proved successful in a broad range of audio applications, including automatic music transcription [Smaragdis and Brown, 2003], multipitch estimation [Vincent et al., 2010, Bertin et al., 2010, Fuentes et al., 2013, Benetos et al., 2014], and audio inpainting [Smaragdis et al., 2011].

Chapter 3

Audio source separation

1 Introduction

Source separation is the art of estimating *source* signals, which are assumed statistically independent, from the observation of one or several *mixtures* of these signals. It is useful in many audio signal processing tasks, including *denoising* applications:

- separation of the instruments in polyphonic music;
- karaoke: remove the singer voice in music recordings;
- cocktail party problem: isolate the voice of the person you are speaking to from many other voices;
- suppression of vuvuzela in TV broadcasting of football matches during the 2010 FIFA world cup.

Besides, the separated audio tracks can be used for remixing purposes, possibly including transformations (e.g. pitch shifting, time scaling, etc.) or re-spatialization of the separated audio sources.

1.1 Typology of the mixture models

Formally, the observed data is made of M mixture signals $x_m(t)$, concatenated in a vector $\boldsymbol{x}(t)$. The unknowns are the K (possibly different from M) source signals $s_k(t)$, concatenated in a vector $\boldsymbol{s}(t)$.

The mixture is modeled as a function $\mathcal{A}$ which transforms the source signals $\boldsymbol{s}(t)$ into the mixture signals $\boldsymbol{x}(t)$. Generally, some simplifying assumptions are introduced regarding the mixture model [Vincent et al., 2018, chap. 1]:

- *Stationarity*: function $\mathcal{A}$ is translation invariant.
- *Linearity*: function $\mathcal{A}$ is a linear map.
- *Memory*:
  - Transformations that are both stationary and linear can be modeled with convolution products in the time domain (linear filtering).
  - The *memory* of such transformations corresponds to the length of the impulse response.
  - If there is no memory (i.e. the length is zero), the mixture is called *instantaneous* and $\mathcal{A}$ is characterized by a *mixing matrix* $\boldsymbol{A}$ (of dimension $M \times K$): $\boldsymbol{x}(t) = \boldsymbol{A}\,\boldsymbol{s}(t)$. Instantaneous mixture models are suitable e.g. for some biomedical applications (electroencephalography (EEG) or magnetoencephalography (MEG)), but generally not for audio applications, because of reverberation.

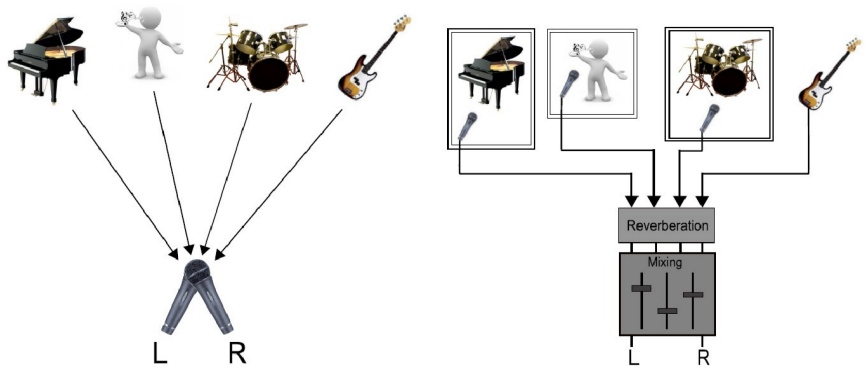
Depending on the respective values of M and K, the mixture may or may not be invertible:

- If $M = K$, the mixture is called *determined*: it is generally invertible.
- If $M > K$, the mixture is called *over-determined*: a unique solution can be found in the least squares sense.
- If $M < K$, the mixture is called *under-determined*: there are infinitely many solutions. Without additional information about the mixture or the source signals, it is impossible to retrieve the original sources from the mixture signals.

1.2 Instantaneous linear mixtures

Examples of instantaneous linear mixtures are given in Figure 3.1:

- In a real audio environment, an approximately instantaneous linear mixture can be obtained with the X-Y stereo recording technique, by putting two directional microphones at the same place, typically oriented at 90 degrees or more from each other (Figure 3.1-(a)). However the audio mixture obtained in this way is never perfectly instantaneous.
- Otherwise, truly instantaneous linear mixtures can of course be created artificially by using a mixing deck or a computer (Figure 3.1-(b)).



(a) XY Stereo configuration (b) Direct injection to the mixer

Figure 3.1: Instantaneous linear mixtures

1.3 Anechoic linear mixtures

Anechoic linear mixtures are a particular case of convolutive mixtures that can be recorded in an *anechoic chamber*: because the sound reflections on the room walls are greatly attenuated, every impulse response is formed only of a single pulse, characterized by its delay and its magnitude, which corresponds to the direct propagation path from every source to every microphone (Figure 3.2).

1.4 Convolutive mixtures

In the general case, audio mixtures are convolutive: in a room, the sound waves are reflected on the walls, so the impulse response is formed of infinitely many pulses, which correspond to the direct propagation path and the various reflections, whose density grows quadratically with time. This phenomenon is called *reverberation* (Figure 3.3-(a)). Convolutive mixtures can also be created artificially, e.g. to simulate a 3-D stereo sound sensation for the listener using headphones (*binaural* mixture, illustrated in Figure 3.3-(b)).

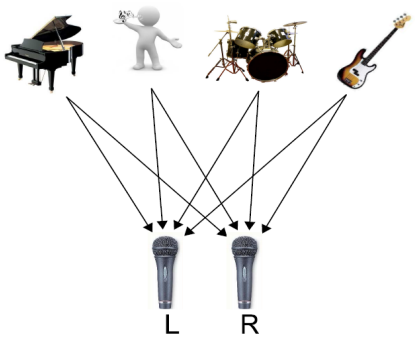
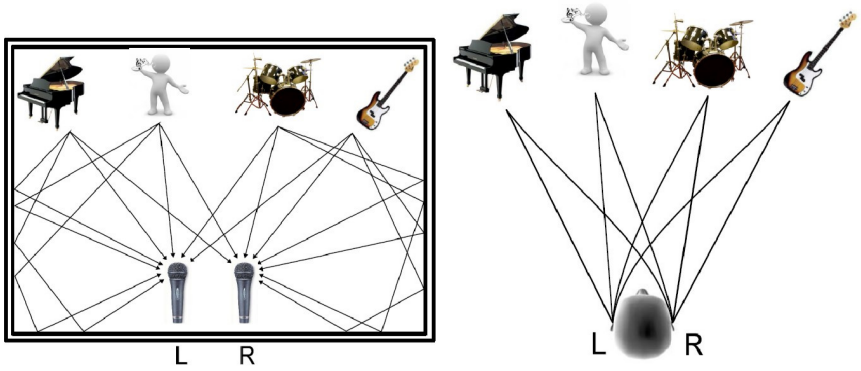


Figure 3.2: Anechoic linear mixtures



(a) Convolutive mixture (b) Binaural mixture

Figure 3.3: Convolutive mixtures

2 Mathematical reminders

Because most source separation techniques involve probabilistic models, we first start with some mathematical reminders from probability theory and statistical signal processing.

2.1 Real random vectors

Let $\boldsymbol{x} \in \mathbb{R}^M$ denote a real random vector. In the rest of this document, we will use the following notation: $\phi[\boldsymbol{x}]$ (with square brackets) denotes a function of the distribution of the random vector $\boldsymbol{x}$, whereas a random variable defined as a function of $\boldsymbol{x}$ would be denoted $\psi(\boldsymbol{x})$ (with parentheses). In particular, we will consider:

- the mean vector: $\boldsymbol{\mu}_x = \mathbb{E}[\boldsymbol{x}] \in \mathbb{R}^M$ (where $\mathbb{E}$ denotes the *mathematical expectation*, a.k.a the *expected value*);
- the covariance matrix: $\boldsymbol{\Sigma}_{xx} = \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^\top] \in \mathbb{R}^{M \times M}$, which is always symmetric (i.e. $\boldsymbol{\Sigma}_{xx}^\top = \boldsymbol{\Sigma}_{xx}$ where $.^\top$ denotes the transpose of a matrix) and positive semi-definite (i.e. $\forall \boldsymbol{v} \in \mathbb{R}^M, \boldsymbol{v}^\top \boldsymbol{\Sigma}_{xx} \boldsymbol{v} \geq 0$);
- the characteristic function: $\phi_x(\boldsymbol{f}) = \mathbb{E}[e^{-2i\pi \boldsymbol{f}^\top \boldsymbol{x}}] \in L^\infty(\mathbb{R}^M)$ (where $L^\infty(\mathbb{R}^M)$ denotes the Lebesgue space of essentially bounded functions on $\mathbb{R}^M$);
- when the inverse Fourier transform of ϕ_x is a measurable function on $\mathbb{R}^M$, $p(\boldsymbol{x}) = \int_{\mathbb{R}} \phi_x(\boldsymbol{f}) e^{+2i\pi \boldsymbol{f}^\top \boldsymbol{x}} d\boldsymbol{f}$ is called the *Probability Density Function* (PDF) of the random vector $\boldsymbol{x}$.

Some of the oldest source separation methods are based on the notion of *cumulants*. The cumulants of the random vector $\boldsymbol{x}$ will be denoted $\kappa^n_{k_1\ldots k_n}[\boldsymbol{x}] \in \mathbb{R}$ for all orders $n \in \mathbb{N}$ and entries $k_i \in \{1 \ldots M\}$, and they are defined as the coefficients of the Taylor expansion of the *cumulant generating function*, which is the natural logarithm of the characteristic function: when ϕ_x is an analytic function, we can write

$$\ln(\phi_x(\boldsymbol{f})) = \sum_{n=1}^{+\infty} \frac{(-2\imath\pi)^n}{n!} \sum_{k_1=1}^{M} \sum_{k_n=1}^{M} \kappa^n_{k_1\ldots k_n}[\boldsymbol{x}] f_{k_1} \ldots f_{k_n}.$$

The cumulants satisfy the following properties:

- $\forall n \in \mathbb{N}^*$, $\kappa^n[\boldsymbol{x}]$ is an n-th order tensor of coefficients $\kappa^n_{k_1\ldots k_n}[\boldsymbol{x}]$;
- $\kappa^1[\boldsymbol{x}]$ is the mean vector $\boldsymbol{\mu}_x$ and $\kappa^2[\boldsymbol{x}]$ is the covariance matrix $\boldsymbol{\Sigma}_{xx}$;
- If the PDF $p(\boldsymbol{x})$ is symmetric ($p(-\boldsymbol{x}) = p(\boldsymbol{x})$), then $\kappa^n[\boldsymbol{x}] = 0$ for any odd value n;
- The ratio between the fourth order cumulant $\kappa^4_{k,k,k,k}[\boldsymbol{x}]$ and the squared variance $(\kappa^2_{k,k}[\boldsymbol{x}])^2$ plays a special role in independent component analysis (*cf.* Section 3.2). It is called the *excess kurtosis*.

2.2 Real Gaussian random vectors

Among all probability distributions with well-defined cumulants of all orders, the Gaussian distribution is the one such that all cumulants of order $n > 2$ are zero. The characteristic function of a Gaussian random vector $\boldsymbol{x} \in \mathbb{R}^M$ can thus be expressed as

$$\phi_x(\boldsymbol{f}) = \exp\left(-2\imath\pi \boldsymbol{f}^\top \boldsymbol{\mu}_x - 2\pi^2 \boldsymbol{f}^\top \boldsymbol{\Sigma}_{xx} \boldsymbol{f}\right).$$

When the covariance matrix $\boldsymbol{\Sigma}_{xx}$ is invertible, then the PDF is defined as:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{M}{2}} \det(\boldsymbol{\Sigma}_{xx})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_x)^\top \boldsymbol{\Sigma}_{xx}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_x)\right).$$

2.3 WSS vector processes

A discrete vector process is a sequence of random vectors $\boldsymbol{x}(t) \in \mathbb{R}^M$ indexed by time $t \in \mathbb{Z}$. A second order vector process is a discrete vector process with well-defined second order moments. Finally, a *Wide Sense Stationary* (WSS) vector process $\boldsymbol{x}(t)$ is a second order vector process whose cumulants of orders 1 and 2 are invariant under any translation of time:

- $\mathbb{E}[\boldsymbol{x}(t)] = \boldsymbol{\mu}_x \ \forall t \in \mathbb{Z}$ where $\boldsymbol{\mu}_x \in \mathbb{R}^M$ is the *mean vector* of the vector process $\boldsymbol{x}(t)$;
- $\forall t \in \mathbb{Z}$, $\mathbb{E}\left[(\boldsymbol{x}(t+\tau) - \boldsymbol{\mu}_x)(\boldsymbol{x}(t) - \boldsymbol{\mu}_x)^\top\right] = \boldsymbol{R}_{xx}(\tau)$, where $\forall \tau \in \mathbb{Z}$ $\boldsymbol{R}_{xx}(\tau) \in \mathbb{R}^{M \times M}$ defines the *autocovariance function* of the vector process $\boldsymbol{x}(t)$. When $\tau = 0$, $\boldsymbol{R}_{xx}(0) = \boldsymbol{\Sigma}_{xx}$ is the covariance matrix of the random vector $\boldsymbol{x}(t)$ $\forall t \in \mathbb{Z}$, and as such it is symmetric and positive semi-definite.

Finally, given two jointly WSS vector processes $\boldsymbol{x}(t) \in \mathbb{R}^M$ and $\boldsymbol{y}(t) \in \mathbb{R}^N$ of mean zero, we define their *intercovariance function* $\boldsymbol{R}_{xy}(\tau) \in \mathbb{R}^{M \times N}$:

$$\forall \tau \in \mathbb{Z},\ \boldsymbol{R}_{xy}(\tau) = \mathbb{E}\left[\boldsymbol{x}(t+\tau)\boldsymbol{y}(t)^\top\right].$$

When the *Discrete Time Fourier Transform* (DTFT) of the autocovariance function $\boldsymbol{R}_{xx}(\tau)$ of a WSS vector process $\boldsymbol{x}(t)$ is a measurable function $\boldsymbol{S}_{xx}(\nu) \in \mathbb{C}^{M \times M}$, this function is called the *Power Spectral Density* (PSD) of $\boldsymbol{x}(t)$:

$$\forall \nu \in \mathbb{R},\ \boldsymbol{S}_{xx}(\nu) = \sum_{\tau \in \mathbb{Z}} \boldsymbol{R}_{xx}(\tau) e^{-2\imath\pi\nu\tau}.$$

The PSD is always periodic of period 1, and $\forall \nu \in \mathbb{R}$, matrix $\boldsymbol{S}_{xx}(\nu)$ is always Hermitian symmetric (i.e. $\boldsymbol{S}_{xx}(\nu)^H = \boldsymbol{S}_{xx}(\nu)$ where $.^H$ denotes the conjugate transpose of a matrix) and positive semi-definite (i.e. $\forall \boldsymbol{v} \in \mathbb{C}^M$, $\boldsymbol{v}^H \boldsymbol{S}_{xx}(\nu) \boldsymbol{v} \geq 0$).

2.4 Information theory

Information theory is a fundamental tool in *blind source separation* (*cf.* Section 3.1), because it makes it possible to measure the amount of information shared between several random variables.

We first consider the notion of *entropy*, which measures the degree of uncertainty in a probability distribution. For a discrete random variable x with probability distribution p, the *Shannon entropy* is defined as $\mathbb{H}[x] = -\mathbb{E}[\ln(p(x))]$. It is always a non-negative real number. The higher this number, the more "uncertain" the outcome of x is. For a continuous random vector $\boldsymbol{x} \in \mathbb{R}^M$ with PDF $p(\boldsymbol{x})$, the *differential entropy* is defined in the same way: $\mathbb{H}[\boldsymbol{x}] = -\mathbb{E}[\ln(p(\boldsymbol{x}))]$. However the differential entropy $\mathbb{H}[\boldsymbol{x}]$ is not necessarily non-negative.

For continuous random vectors $\boldsymbol{x} \in \mathbb{R}^M$, the *Kullback-Leibler divergence* measures the degree of dissimilarity between two probability distributions characterized by their PDFs p and q:

$$D_{KL}(p\|q) = \int_{\boldsymbol{x}\in\mathbb{R}^M} p(\boldsymbol{x}) \ln\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right) d\boldsymbol{x}.$$

As a *divergence*, it is always nonnegative, and $D_{KL}(p\|q) = 0$ if and only if $p = q$. However, the Kullback-Leibler divergence is not a *distance*, because it is not symmetric (in general $D_{KL}(p\|q) \neq D_{KL}(q\|p)$), and it does not satisfy the triangle inequality.

Finally, the *mutual information* measures the mutual dependence between several random variables. For instance if $\boldsymbol{x} \in \mathbb{R}^M$ is a continuous random vector, then the mutual information between the entries of $\boldsymbol{x}$ is defined as:

$$\mathbb{I}[\boldsymbol{x}] = \mathbb{E}\left[\ln\left(\frac{p(\boldsymbol{x})}{p(x_1)\dots p(x_M)}\right)\right] = D_{KL}(p(\boldsymbol{x})\|p(x_1)\dots p(x_M)).$$

Since D_{KL} is a divergence, $\mathbb{I}[\boldsymbol{x}]$ is always nonnegative, and $\mathbb{I}[\boldsymbol{x}] = 0$ if and only if $p(\boldsymbol{x}) = p(x_1)\dots p(x_M)$, i.e. if and only if the random variables $x_1 \dots x_M$ are mutually independent. The mutual information is related to the differential entropy through the equality

$$\mathbb{I}[\boldsymbol{x}] = \left(\sum_{m=1}^{M} \mathbb{H}[x_m]\right) - \mathbb{H}[\boldsymbol{x}]. \tag{3.1}$$

In *independent component analysis*, the mutual information is an objective function to be minimized, in order to make several random variables as independent as possible (*cf.* Section 3.2.2). Equation (3.1) shows that minimizing $\mathbb{I}[\boldsymbol{x}]$ is equivalent to minimizing the sum of the individual entropies $\mathbb{H}[x_m]$ when the joint entropy $\mathbb{H}[\boldsymbol{x}]$ is fixed.

3 Linear instantaneous mixtures

Even though we have seen in Section 1.2 page 39 that the linear instantaneous mixture model cannot accurately represent real acoustic mixtures, the oldest separation techniques which paved the way for modern audio source separation methods are based on this model. We thus first address this over-simplified mixture model, which will permit us to introduce several useful concepts and methods, that will then be extended to the more realistic convolutive mixture model in Section 4.

3.1 Blind source separation (BSS) model

Blind Source Separation (BSS) techniques Cardoso [1998] assume that we know very little about the source signals: they are only assumed to be statistically independent. This is the case for instance in many denoising applications, where the signal of interest (e.g. speech) is independent from the source of noise (e.g. background environmental noise). This hypothesis is the funding principle of most multichannel source separation methods.

Actually, BSS methods also rely on a generative source model, but this model is chosen as little informative as possible: the samples of each source signal are assumed *Independent and Identically Distributed* (IID). The IID source model thus ignores any temporal dynamics (i.e. power variations over time), or spectral dynamics (i.e. temporal correlations), that might be present in the source signals:

Definition 1 (IID source model)**.** *We consider K independent source signals $s_k(t) \in \mathbb{R}$ with $t \in \mathbb{Z}$. For all $k \in \{1 \dots K\}$, s_k is modeled as an IID random process: the samples $s_k(t)$ are independent random variables, of same probability distribution p_k (which depends on source k).*

The possibility of performing source separation in such a blind way may seem to be an incredible feat. Actually, the trick is that, contrary to the source model, the mixture model is *very* constraining: at first we will only consider linear instantaneous mixtures, characterized by a mixing matrix $\boldsymbol{A}$:

Definition 2 (Linear instantaneous mixture model)**.** *We consider K source signals $s_k(t) \in \mathbb{R}$ with $t \in \mathbb{Z}$. Then the samples of the M mixture signals $x_m(t) \in \mathbb{R}$ for $m \in \{1 \dots M\}$ are defined as the entries of the M-dimensional vector*

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t), \tag{3.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{M \times K}$ is called the mixing matrix*, and $\boldsymbol{s}(t)$ is the K-dimensional vector of entries $s_k(t)$ for $k \in \{1 \dots K\}$.*

Given the source model in Definition 1 and the mixture model in Definition 2, the purpose of BSS is to estimate the source signals $s_k(t)$ given the observed mixture signals $x_m(t)$, *without knowing* the mixing matrix $\boldsymbol{A}$. When the mixture is *determined* ($M = K$) and matrix $\boldsymbol{A}$ is invertible, we will show that this is generally feasible.

3.1.1 Identifiability

Suppose that the mixture is determined ($M = K$) . Before investigating how source separation can be performed, we first need to study the *identifiability* of the linear instantaneous BSS model: is it really possible to retrieve both the source signals $s_k(t)$ and the mixing matrix $\boldsymbol{A}$ from only the observed mixture signals $x_m(t)$?

Clearly, if $\boldsymbol{P}$ is a permutation matrix (i.e. it has a unique 1 entry in each row and each column, all other entries being 0), then matrix $\tilde{\boldsymbol{A}} = \boldsymbol{A}\,\boldsymbol{P}^{-1}$ and vector $\tilde{\boldsymbol{s}}(t) = \boldsymbol{P}\boldsymbol{s}(t)$ lead to the same observations $\boldsymbol{x}(t)$, while satisfying all the properties of the linear instantaneous BSS model in Definitions 1 and 2. So the source signals can only be retrieved up to a permutation: at best we can retrieve the source signals, but we cannot *identify* them.

In the same way, if $\boldsymbol{D}$ is an invertible diagonal matrix, then matrix $\tilde{\boldsymbol{A}} = \boldsymbol{A}\,\boldsymbol{D}^{-1}$ and vector $\tilde{\boldsymbol{s}}(t) = \boldsymbol{D}\boldsymbol{s}(t)$ lead to the same observations $\boldsymbol{x}(t)$, while satisfying all the properties of the linear instantaneous BSS model in Definitions 1 and 2. Therefore the source signals can only be retrieved up to a multiplicative factor (which in most applications is not a problem: e.g. audio signals are generally scaled during playback).

So the linear instantaneous BSS model in Definitions 1 and 2 has at least *permutation* and *scale* indeterminacies. Actually, it can be proved that there is no other one. These two indeterminacies are summarized by the concept of *non-mixing matrices*:

Definition 3 (Non-mixing matrix)**.** *A matrix $\boldsymbol{C} \in \mathbb{R}^{K \times K}$ is* non-mixing *if and only if it has a unique non-zero entry in each row and each column.*

A non-mixing matrix can always be decomposed as the product of a permutation matrix and an invertible diagonal matrix.

3.1.2 Linear separation of sources

When the mixture is linear instantaneous, it may seem natural to estimate the source signals as linear instantaneous combinations of the mixture signals:

$$\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{x}(t), \tag{3.3}$$

where the entries of vector $\boldsymbol{y}(t)$ are the source signal estimates, and $\boldsymbol{B} \in \mathbb{R}^{K \times M}$ is referred to as the *separation matrix*. Then the source separation problem amounts to finding an optimal separation matrix.

Linear source separation is generally feasible in the case of determined and over-determined mixtures:

- if $M = K$ and if matrix $\boldsymbol{A}$ is invertible, then the separation matrix $\boldsymbol{B} = \boldsymbol{A}^{-1}$ leads to $\boldsymbol{y}(t) = \boldsymbol{s}(t)$;
- more generally, if $M \geq K$ and if matrix $\boldsymbol{A}$ has full rank, then the separation matrix $\boldsymbol{B} = \boldsymbol{A}^{\dagger}$ leads to $\boldsymbol{y}(t) = \boldsymbol{s}(t)$, where $.^{\dagger}$ denotes the matrix the pseudo-inverse: $\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top}$, which is such that $\boldsymbol{A}^{\dagger}\boldsymbol{A} = \boldsymbol{I}_K$.

However, in the under-determined case ($M < K$), linear source separation is generally not feasible (*cf.* Section 5).

3.2 Independent component analysis (ICA)

Independent Component Analysis (ICA) Comon and Jutten [2010] is a linear source separation technique which consists in looking for a separation matrix $\boldsymbol{B}$ that makes the signals $y_k(t)$ independent.

Since $\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t)$ (equation (3.2)) and $\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{x}(t)$ (equation (3.3)), we have $\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{s}(t)$ with $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{A}$. According to the identifiability analysis in Section 3.1.1, the BSS problem is solved if and only if matrix $\boldsymbol{C}$ is non-mixing (*cf.* Figure 3.4).

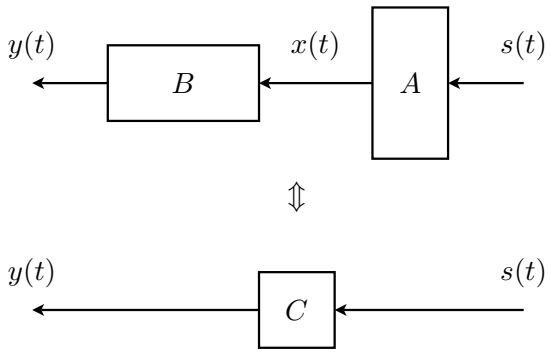


Figure 3.4: Identifiability theorem: signals $y_k(t)$ are independent if and only if matrix $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{A}$ is non-mixing

The following identifiability theorem due to P. Comon Comon [1994] proves the feasibility of ICA under mild conditions about the source signals:

Theorem 1 (Identifiability theorem)**.** *Consider the linear instantaneous BSS model in Definitions 1 and 2 in the determined case ($M = K$). Among the K IID sources s_k, suppose that at most one is Gaussian-distributed. Let $\boldsymbol{C} \in \mathbb{R}^{K\times K}$ and $\forall t \in \mathbb{Z}$, $\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{s}(t)$. Then the random processes $y_k(t)$ for $k \in \{1 \dots K\}$ are independent if and only if matrix $\boldsymbol{C}$ is non-mixing.*

Theorem 1 proves that finding a separation matrix $\boldsymbol{B}$ that makes signals $y_k(t)$ independent solves the BSS problem: the estimated signals $y_k(t)$ are equal to the source signals $s_k(t)$ up to permutation and scale indeterminacies.

Here, pay attention to the non-Gaussianity assumption in Theorem 1: in Section 3.2.1, we will show that indeed, if two (or more) sources are Gaussian, then the BSS problem cannot be solved.

3.2.1 Whitening

Independent component analysis can be performed in two steps; the first one consists in *whitening*[1], i.e. *decorrelating* the observed mixture signals. Remember that independence implies decorrelation, but decorrelation does generally not imply independence. Therefore the second step will consist in making the whitened signals independent.

To simplify the problem, we will address the case of determined mixtures $M = K$ (even though whitening could also be performed in the over-determined case $M > K$), and we will assume that matrix $\boldsymbol{A}$ is invertible and that the source signals are centered: $\mathbb{E}[\boldsymbol{s}(t)] = \boldsymbol{0}$ (which is always the case of audio signals).

To further simplify, we will focus on the *canonical BSS problem*: without loss of generality, we will assume that the random vectors $\boldsymbol{s}(t)$ are spatially white, i.e. their covariance matrix is $\boldsymbol{\Sigma}_{ss} = \mathbb{E}[\boldsymbol{s}(t)\boldsymbol{s}(t)^\top] = \boldsymbol{I}_K$. Indeed, since the source signals are independent, we already know that matrix $\boldsymbol{\Sigma}_{ss}$ is diagonal; since in addition the source

[1] Whitening is performed in the spatial domain, i.e. over channels, not in the time domain, i.e. over time samples.

signals can only be retrieved up to a multiplicative factor, we can also assume without loss of generality that the diagonal entries of matrix $\boldsymbol{\Sigma}_{ss}$ are 1.

Since $\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t)$ (equation (3.2)), the covariance matrix of the mixture vectors $\boldsymbol{x}(t)$ is $\boldsymbol{\Sigma}_{xx} = \boldsymbol{A}\boldsymbol{\Sigma}_{ss}\boldsymbol{A}^\top = \boldsymbol{A}\,\boldsymbol{A}^\top$: we say that $\boldsymbol{A}$ is a *matrix square root* of $\boldsymbol{\Sigma}_{xx}$. This property is interesting because $\boldsymbol{\Sigma}_{xx}$ can be estimated from the observed data, and it carries information about the mixing matrix $\boldsymbol{A}$. Unfortunately, we will see that this property is not sufficient to fully characterize $\boldsymbol{A}$. Nevertheless, it allows us to make a first step towards the estimation of $\boldsymbol{A}$. For the moment, just note that since matrix $\boldsymbol{A}$ is invertible, matrix $\boldsymbol{\Sigma}_{xx}$ is also invertible, thus positive definite.

The whitening of the mixture signals can then be performed as follows Cardoso and Souloumiac [1993]:

- Since matrix $\boldsymbol{\Sigma}_{xx}$ is positive definite, the spectral theorem in matrix theory shows us that it is diagonalizable in an orthonormal basis: there is an orthonormal matrix $\boldsymbol{Q} \in \mathbb{R}^{K\times K}$ (i.e. such that $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^\top$), and a diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{K\times K}$ with positive diagonal entries, such that

$$\boldsymbol{\Sigma}_{xx} = \boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top. \tag{3.4}$$

- Then let $\boldsymbol{S} = \boldsymbol{Q}\,\boldsymbol{\Lambda} \in \mathbb{R}^{K\times K}$; matrix $\boldsymbol{S}$ is also a matrix square root of $\boldsymbol{\Sigma}_{xx}$, since $\boldsymbol{S}\,\boldsymbol{S}^\top = \boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top = \boldsymbol{\Sigma}_{xx}$.

- Finally, let

$$\boldsymbol{W} = \boldsymbol{S}^{-1} \tag{3.5}$$

and

$$\forall t \in \mathbb{Z},\ \boldsymbol{z}(t) = \boldsymbol{W}\boldsymbol{x}(t). \tag{3.6}$$

Then the random vector process $\boldsymbol{z}(t)$ is spatially white, in the sense that on the one hand it is centered: $\mathbb{E}[\boldsymbol{z}(t)] = \boldsymbol{0}$ (since $\boldsymbol{z}(t) = \boldsymbol{W}\boldsymbol{A}\boldsymbol{s}(t)$ and $\mathbb{E}[\boldsymbol{s}(t)] = \boldsymbol{0}$), and on the other hand its covariance matrix is $\boldsymbol{\Sigma}_{zz} = \boldsymbol{W}\boldsymbol{\Sigma}_{xx}\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{W}^\top = \boldsymbol{I}_K$). Matrix $\boldsymbol{W}$ will thus be referred to as the *whitening matrix* and $\boldsymbol{z}(t)$ is the *whitened data*.

Then let us define matrix $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{A}$. We have $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{\Sigma}_{xx}\boldsymbol{W}^\top = \boldsymbol{I}_K$, therefore $\boldsymbol{U}$ is an orthonormal matrix. In particular, $|\det(\boldsymbol{U})| = 1$: if $\det(\boldsymbol{U}) = 1$, $\boldsymbol{U}$ is a *rotation matrix*, otherwise if $\det(\boldsymbol{U}) = -1$, $\boldsymbol{U}$ is a *reflection matrix*. However, remember that the source signals can only be retrieved up to a multiplicative factor, which might as well be negative. Therefore, by changing the sign of the k-th source signal $s_k(t)$, the product $\boldsymbol{A}\boldsymbol{s}(t)$ is left unchanged by changing the sign of the k-th column of matrix $\boldsymbol{A}$, which changes the sign of $\det(\boldsymbol{U})$. Therefore, without loss of generality, we can assume that $\boldsymbol{U}$ *is a rotation matrix* ($\det(\boldsymbol{U}) = 1$).

Finally, let

$$\boldsymbol{y}(t) = \boldsymbol{U}^\top\boldsymbol{z}(t). \tag{3.7}$$

Then $\boldsymbol{y}(t) = \boldsymbol{U}^\top\boldsymbol{W}\boldsymbol{x}(t) = (\boldsymbol{W}\boldsymbol{A})^{-1}\boldsymbol{W}(\boldsymbol{A}\boldsymbol{s}(t)) = \boldsymbol{s}(t)$. Therefore matrix $\boldsymbol{B} = \boldsymbol{U}^\top\boldsymbol{W}$ is a separation matrix. In practice of course, matrix $\boldsymbol{A}$ is unknown, thus so is matrix $\boldsymbol{U}$. But it remains that ICA can be performed in two steps (*cf.* Figure 3.5):

Step 1 : compute the whitening matrix $\boldsymbol{W}$ and the whitened data $\boldsymbol{z}(t)$ from an estimate of the covariance matrix $\boldsymbol{\Sigma}_{xx}$;

Step 2 : look for a rotation matrix $\boldsymbol{U}$ such that the entries of vector $\boldsymbol{y}(t) = \boldsymbol{U}^\top\boldsymbol{z}(t)$ are independent.

To summarize, the whiteness property (based on second order cumulants) determines matrix $\boldsymbol{W}$ and leaves the rotation matrix $\boldsymbol{U}$ unknown.

Note that in the Gaussian case, decorrelation implies independence. Therefore if the source signals are Gaussian-distributed, then the whitened signals $z_k(t)$ are independent, and $\boldsymbol{U}$ cannot be determined. This explains the assumption made in Theorem 1 page 44 that at most one source can be Gaussian-distributed (if they are two of them, they cannot be separated).

Therefore if we want to determine rotation $\boldsymbol{U}$, we will need to explicitly exploit the non-Gaussianity of the source signals. To do so, we will characterize the independence property by using cumulants of order greater than 2 Cardoso and Souloumiac [1993].

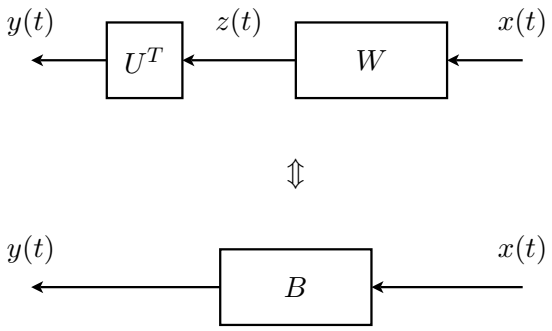


Figure 3.5: Pre-whitening for independent component analysis: $\boldsymbol{B} = \boldsymbol{U}^\top \boldsymbol{W}$ where $\boldsymbol{U}$ is a rotation matrix

3.2.2 Contrast functions

In Section 2.4, we have introduced the concept of *mutual information*: the mutual information between several random variables is always non-negative, and it is zero if and only if these random variables are independent. Therefore the mutual information between the entries of vector $\boldsymbol{y}(t)$ can be used as an objective function to be minimized in order to perform ICA.

More generally, the concept of *contrast functions* has been introduced in order to formulate ICA as an optimization problem, the mutual information being only one example of such functions. Formally, still in the determined case ($M = K$), Theorem 1 leads to the following definition of *contrast functions* Cardoso [1998]:

Definition 4 (Contrast function)**.** *For all $k \in \{1 \dots K\}$, we consider source signals $s_k(t)$ as defined in Definition 1. Then a function $\phi : \mathbb{R}^K \to \mathbb{R}$ is a* contrast function *when $\phi[\boldsymbol{C}\boldsymbol{s}(t)] \geq \phi[\boldsymbol{s}(t)]$ for any matrix $\boldsymbol{C} \in \mathbb{R}^{K \times K}$, and $\phi[\boldsymbol{C}\boldsymbol{s}(t)] = \phi[\boldsymbol{s}(t)]$ if and only if matrix $\boldsymbol{C}$ is non-mixing.*

Following Definition 4 and considering linear instantaneous mixtures $\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t)$ (equation (3.2)) as in Definition 2 page 43, linear source separation $\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{x}(t)$ (equation (3.3) page 43), and matrix $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{A}$, the BSS problem is solved by minimizing the contrast function $\phi[\boldsymbol{y}(t)]$ with respect to (w.r.t.) the separation matrix $\boldsymbol{B}$, or w.r.t. the rotation matrix $\boldsymbol{U}$ if the data has been whitened. Among all possible contrast functions, the mutual information $\phi_{IM}[\boldsymbol{y}(t)] = \mathbb{I}[\boldsymbol{y}(t)]$ is considered as the *canonical* contrast function.

If the observed data has already been whitened, it is possible to consider only *orthogonal contrast functions*, which are such that ICA is performed by minimizing an orthogonal contrast function subject to the constraint $\mathbb{E}[\boldsymbol{y}(t)\boldsymbol{y}(t)^\top] = \boldsymbol{I}_K$. For instance, by using equation (3.1) page 42, it can be shown that an orthogonal contrast function associated to the mutual information is $\phi^\circ_{IM}[\boldsymbol{y}(t)] = \sum_{k=1}^K H(y_k(t))$, because $H(\boldsymbol{y}(t))$ is left unchanged under the constraint $\mathbb{E}[\boldsymbol{y}(t)\boldsymbol{y}(t)^\top] = \boldsymbol{I}_K$.

In practice, the orthogonal contrast function ϕ°_{IM} can be expressed as a function of the cumulants of $\boldsymbol{y}(t)$, and approximated by considering only the cumulants up to order 4 Cardoso and Souloumiac [1993]:

$$\phi^\circ_{ICA}[\boldsymbol{y}(t)] = \sum_{ijkl \neq iiii} (\kappa^4_{ijkl}[\boldsymbol{y}(t)])^2. \tag{3.8}$$

Then the minimization of ϕ°_{ICA} with respect to the rotation matrix $\boldsymbol{U}$ can e.g. be performed by factorizing $\boldsymbol{U}$ as a product of Givens rotations (i.e. 2-dimensional rotation matrices parameterized by a single angle in $[0, 2\pi]$, which are applied iteratively to every pair of entries of vector $\boldsymbol{y}(t)$), and by performing a coordinate descent, also known as (a.k.a.) Jacobi technique, w.r.t. the angles of these Givens rotations Comon and Jutten [2010].

Compared to equation (3.8), the independence can also be tested on a smaller subset of cumulants, as

$$\phi^{\circ}_{JADE}[\boldsymbol{y}(t)] = \sum_{ijkl \neq ijkk} (\kappa^4_{ijkl}[\boldsymbol{y}(t)])^2. \tag{3.9}$$

The motivation for using this specific subset is that $\phi^{\circ}_{JADE}[\boldsymbol{y}(t)]$ can also be seen as a joint diagonalization criterion[2]. This approach leads to the celebrated *Joint Approximate Diagonalization of Eigenmatrices* (JADE) method Cardoso and Souloumiac [1993] summarized in Algorithm 1.

Algorithm 1 JADE method

Estimation of the covariance matrix $\boldsymbol{\Sigma}_{xx}$ and diagonalization: $\boldsymbol{\Sigma}_{xx} = \boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top$ (equation (3.4))
Computation of $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}$ and of the whitening matrix $\boldsymbol{W} = \boldsymbol{S}^{-1}$ (equation (3.5))
Data whitening: $\boldsymbol{z}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ (equation (3.6))
Estimation of $\boldsymbol{U}$ by minimizing the contrast function ϕ°_{JADE} in equation (3.9)
Estimation of source signals via $\boldsymbol{y}(t) = \boldsymbol{U}^\top \boldsymbol{z}(t)$ (equation (3.7))

3.3 Second order methods

The JADE method is dedicated to the linear instantaneous BSS model in Definitions 1 and 2 page 43. It makes use of higher order statistics (i.e. of order greater than 2), because the identifiability of the model requires that at most one source signal be Gaussian distributed (*cf.* Theorem 1 page 44). However, it is well known that the estimating higher order statistics is more sensitive (e.g. in terms of mean square error) than estimating second order statistics. Therefore it would be interesting to develop source separation methods that only make use of second order statistics. That will require to relax the source model in Definition 1, so that the model become identifiable from its second order statistics only. In Sections 3.3.1 and 3.3.2, we will show two different ways of relaxing the source model.

3.3.1 Temporal coherence of source signals

In this section, we keep the same mixture model as in Definition 2, and we consider the source model in Definition 1 in the determined case ($M = K$), except that each source signal $s_k(t)$ is no longer assumed to be IID, but rather WSS, with a non-flat power spectral density. Therefore, as in Definition 1, the samples $s_k(t)$ for $t \in \mathbb{Z}$ can still follow the same distribution p_k, but now they are assumed to be mutually dependent (the source model is relaxed by removing the first "I" of "IID"):

Definition 5 (WSS source model)**.** *We consider K independent source signals $s_k(t) \in \mathbb{R}$ with $t \in \mathbb{Z}$, concatenated in a vector $\boldsymbol{s}(t)$. For all $k \in \{1 \dots K\}$, s_k is modeled as a centered WSS random process. So $\boldsymbol{s}(t)$ is a WSS vector process of mean $\mathbb{E}[\boldsymbol{s}(t)] = \boldsymbol{0}$ and of autocovariance function $\boldsymbol{R}_{ss}(\tau) = \mathbb{E}\left[\boldsymbol{s}(t+\tau)\boldsymbol{s}(t)^\top\right]$.*

Since the source signals are independent, $\forall \tau \in \mathbb{Z}$ the covariance matrix $\boldsymbol{R}_{ss}(\tau)$ is diagonal: $\boldsymbol{R}_{ss}(\tau) = \mathrm{diag}(r_{s_k}(\tau))$, where $r_{s_k}(\tau) \in \mathbb{R}$ is the autocovariance function of the scalar WSS process $s_k(t)$, and diag(.) denotes a diagonal matrix formed from a vector of diagonal coefficients.

As in Section 3.2.1 page 44, we can still consider the canonical BSS problem and assume that $\boldsymbol{\Sigma}_{ss} = \boldsymbol{R}_{ss}(0) = \boldsymbol{I}_K$. Then, still as in Section 3.2.1, we can *spatially* whiten the mixture signals:

- compute a matrix square root $\boldsymbol{S}$ of $\boldsymbol{\Sigma}_{xx}$;
- compute $\boldsymbol{W} = \boldsymbol{S}^{-1}$ and the whitened data $\boldsymbol{z}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ $\forall t \in \mathbb{Z}$.

Again, since $\boldsymbol{\Sigma}_{xx} = \boldsymbol{A}\,\boldsymbol{A}^\top$, matrix $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{A}$ is a rotation matrix. The novelty, compared with the mathematical developments in Section 3.2.1, is that we can now consider matrices $\boldsymbol{R}_{zz}(\tau)$ $\forall \tau \in \mathbb{Z}$, since they are no longer

[2] Joint diagonalization of matrices will be addressed in Section 3.3.1.

assumed to be zero. On the contrary, we have $\forall \tau \in \mathbb{Z}$, $\boldsymbol{R}_{zz}(\tau) = \boldsymbol{W}\boldsymbol{R}_{xx}(\tau)\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{A}\boldsymbol{R}_{ss}(\tau)\boldsymbol{A}^\top\boldsymbol{W}^\top = \boldsymbol{U}\boldsymbol{R}_{ss}(\tau)\boldsymbol{U}^\top$. This equation shows that matrices $\boldsymbol{R}_{zz}(\tau)$ are jointly diagonalized by the same set of eigenvectors, which are the columns of matrix $\boldsymbol{U}$, the eigenvalues being the diagonal entries of matrices $\boldsymbol{R}_{ss}(\tau)$.

Therefore the estimation of matrix $\boldsymbol{U}$ will no longer require the use of higher order statistics: $\boldsymbol{U}$ can be uniquely determined as the only rotation matrix (up to a non-mixing matrix) that jointly diagonalizes matrices $\boldsymbol{R}_{zz}(\tau)$ for different values of τ, as shown by the following theorem:

Theorem 2 (Unicity theorem)**.** *Let us consider a set of matrices $\boldsymbol{R}_{zz}(\tau) \in \mathbb{R}^{K\times K}$ indexed by $\tau \in \mathbb{Z}$, of the form $\boldsymbol{R}_{zz}(\tau) = \boldsymbol{U}\boldsymbol{R}_{ss}(\tau)\boldsymbol{U}^\top$, where matrix $\boldsymbol{U} \in \mathbb{R}^{K\times K}$ is orthonormal and matrices $\boldsymbol{R}_{ss}(\tau) \in \mathbb{R}^{K\times K}$ are diagonal: $\boldsymbol{R}_{ss}(\tau) = \mathrm{diag}(r_{s_k}(\tau))$. Then $\boldsymbol{U}$ is unique (up to a non-mixing matrix) if and only if $\forall k \neq l \in \{1 \dots K\}$, there is $\tau \in \mathbb{Z}$ such that $r_{s_k}(\tau) \neq r_{s_l}(\tau)$.*

In order to compute matrix $\boldsymbol{U}$, we can thus use any joint diagonalization method Cardoso and Souloumiac [1996]. For instance, we can numerically minimize the following objective function:

$$J(\boldsymbol{U}) = \sum_\tau \|\boldsymbol{U}^\top \boldsymbol{R}_{zz}(\tau)\boldsymbol{U} - \mathrm{diag}(\boldsymbol{U}^\top \boldsymbol{R}_{zz}(\tau)\boldsymbol{U})\|_F^2 \tag{3.10}$$

where $\|.\|_F$ denotes the Frobenius norm of a matrix (i.e. the Euclidean norm of a vector made of all its entries) and diag(.) denotes a diagonal matrix formed from a matrix with same diagonal entries. The criterion $J(\boldsymbol{U})$ is zero if and only if all matrices $\boldsymbol{U}^\top \boldsymbol{R}_{zz}(\tau)\boldsymbol{U}$ are diagonal.

As in Section 3.2.2 page 46, this minimization can be performed by factorizing $\boldsymbol{U}$ as a product of Givens rotations, and by performing a coordinate descent w.r.t. the angles of these Givens rotations Comon and Jutten [2010]. The resulting BSS method is known as the *Second Order Blind Identification* (SOBI) technique Belouchrani et al. [1997], and summarized in Algorithm 2.

Algorithm 2 SOBI method for WSS sources

Estimation of the covariance matrix $\boldsymbol{\Sigma}_{xx}$ and diagonalization: $\boldsymbol{\Sigma}_{xx} = \boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top$ (equation (3.4))
Computation of $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}$ and of the whitening matrix $\boldsymbol{W} = \boldsymbol{S}^{-1}$ (equation (3.5))
Data whitening: $\boldsymbol{z}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ (equation (3.6))
Estimation of covariance matrices $\boldsymbol{R}_{zz}(\tau)$ for various delays τ
Approximate joint diagonalization of matrices $\boldsymbol{R}_{zz}(\tau)$ in a common basis $\boldsymbol{U}$ by minimizing (3.10)
Estimation of source signals via $\boldsymbol{y}(t) = \boldsymbol{U}^\top \boldsymbol{z}(t)$ (equation (3.7))

3.3.2 Non-stationarity of source signals

In this section, we keep the same mixture model as in Definition 2, and we consider the source model in Definition 1 in the determined case ($M = K$), except that each source signal $s_k(t)$ is no longer assumed to be IID, but rather non-stationary. More precisely, as in Definition 1, the samples $s_k(t)$ for $t \in \mathbb{Z}$ can still be assumed independent, but now they no longer follow the same distribution p_k (the source model is relaxed by removing the last letters "ID" of "IID"):

Definition 6 (Non-stationary source model)**.** *We consider K independent source signals $s_k(t) \in \mathbb{R}$ with $t \in \mathbb{Z}$, concatenated in a vector $\boldsymbol{s}(t)$. For all $k \in \{1 \dots K\}$, s_k is modeled as a centered random process with uncorrelated samples $s_k(t)$ for $t \in \mathbb{Z}$, of time-varying variance $\sigma_k^2(t)$. So $\boldsymbol{s}(t)$ is a random vector process of mean $\mathbb{E}[\boldsymbol{s}(t)] = \boldsymbol{0}$ and of time-varying covariance matrix $\boldsymbol{\Sigma}_{ss}(t) = \mathbb{E}\left[\boldsymbol{s}(t)\boldsymbol{s}(t)^\top\right]$.*

Since the source signals are independent, $\forall t \in \mathbb{Z}$ matrix $\boldsymbol{\Sigma}_{ss}(t)$ is diagonal: $\boldsymbol{\Sigma}_{ss}(t) = \mathrm{diag}(\sigma_k^2(t))$.
Then, still as in Section 3.2.1 page 44, we can *spatially* whiten the mixture signals:

- compute a matrix square root $\boldsymbol{S}$ of $\boldsymbol{\Sigma}_{xx} = \sum_t \boldsymbol{\Sigma}_{xx}(t)$;
- compute $\boldsymbol{W} = \boldsymbol{S}^{-1}$ and the whitened data $\boldsymbol{z}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ $\forall t \in \mathbb{Z}$.

Again, as in Section 3.2.1 we can consider a canonical BSS problem and assume $\boldsymbol{\Sigma}_{xx} = \boldsymbol{A}\,\boldsymbol{A}^\top$, therefore matrix $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{A}$ is a rotation matrix.

Then if we consider the covariance matrices of the whitened data: $\forall t \in \mathbb{Z}$, $\boldsymbol{\Sigma}_{zz}(t) = \mathbb{E}\left[\boldsymbol{z}(t)\boldsymbol{z}(t)^\top\right]$, we get $\forall t \in \mathbb{Z}$, $\boldsymbol{\Sigma}_{zz}(t) = \boldsymbol{W}\boldsymbol{\Sigma}_{xx}(t)\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{A}\boldsymbol{\Sigma}_{ss}(t)\boldsymbol{A}^\top\boldsymbol{W}^\top = \boldsymbol{U}\boldsymbol{\Sigma}_{ss}(t)\boldsymbol{U}^\top$. Therefore, as in Section 3.3.1, matrix $\boldsymbol{U}$ can be determined by solving a joint diagonalization problem Cardoso and Souloumiac [1996], e.g. by minimizing the following objective function:

$$J(\boldsymbol{U}) = \sum_t \|\boldsymbol{U}\boldsymbol{\Sigma}_{zz}(t)\boldsymbol{U}^\top - \mathrm{diag}(\boldsymbol{U}\boldsymbol{\Sigma}_{zz}(t)\boldsymbol{U}^\top)\|_F^2. \tag{3.11}$$

We thus get a variant of the SOBI algorithm Belouchrani et al. [1997], summarized in Algorithm 3.

Algorithm 3 SOBI method for non-stationary sources

Estimation of the covariance matrix $\boldsymbol{\Sigma}_{xx}$ and diagonalization: $\boldsymbol{\Sigma}_{xx} = \boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top$ (equation (3.4))
Computation of $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}$ and of the whitening matrix $\boldsymbol{W} = \boldsymbol{S}^{-1}$ (equation (3.5))
Data whitening: $\boldsymbol{z}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ (equation (3.6))
Segmentation of whitened data and estimation of covariance matrices $\boldsymbol{\Sigma}_{zz}(t)$ on the different time frames
Approximate joint diagonalization of matrices $\boldsymbol{\Sigma}_{zz}(t)$ in a common basis $\boldsymbol{U}$ by minimizing (3.11)
Estimation of source signals via $\boldsymbol{y}(t) = \boldsymbol{U}^\top\boldsymbol{z}(t)$ (equation (3.7))

3.4 Time-frequency methods

So far, we have seen that:

- the use of higher order cumulants is only necessary for the non-Gaussian IID source model in Definition 1;
- second order statistics are sufficient for separating source signals that are:
  - either WSS but not IID, as in Definition 5, which amounts to exploit their *spectral* dynamics (through the autocovariance function $\boldsymbol{R}_{ss}(\tau)$);
  - or uncorrelated but not stationary, as in Definition 6, which amounts to exploit their *temporal* dynamics (through the time-varying covariance matrices $\boldsymbol{\Sigma}_{ss}(t)$).

The take-home message is that classical signal processing tools based on second order statistics are appropriate for performing blind separation of independent (and possibly Gaussian) sources, provided that the spectral and/or temporal source dynamics are taken into account.

However, a very simple way of highlighting the spectral and temporal dynamics of a signal is to use a *Time-Frequency* (TF) representation. In this section, we will show how TF representations allow us to easily perform source separation of determined linear instantaneous mixtures. Then in Sections 4 and 5, we will see that TF representations reveal their full potential when processing convolutive and/or under-determined mixtures.

3.4.1 Time-frequency representations

Here we use the expression *time-frequency representation* to refer to complex or real-valued linear time-frequency transforms that can be implemented by means of perfect-reconstruction filterbanks Vaidyanathan [1993]. Classical examples of perfect reconstruction filterbanks include the STFT (which is complex-valued) and the *Modified Discrete Cosine Transform* (MDCT) (which is real-valued) [Vincent et al., 2018, chap. 2].

Every mixture signal $x_m(t)$ is thus filtered by F *analysis filters* h_f corresponding to each frequency channel $f \in \{1 \dots F\}$. The output signals are decimated by a factor $T \leq F$ to produce the F sub-band signals:

$$x_m(f, n) = (h_f * x_m)(nT), \tag{3.12}$$

where $n \in \mathbb{Z}$ is the *time frame index* and T is the *hop-size*.

Since we consider perfect-reconstruction filterbanks, we assume that there exist F *synthesis filters* g_f so that every signal $x_m(t)$ can be perfectly reconstructed from the sub-band signals: $x_m(t) = \sum_{f=1}^{F} g_f(t - nT)x_m(f,n)$.

In the same way, the source signals are decomposed in F sub-band signals $s_k(f,n) = (h_f * s_k)(nT)$ and reconstructed as

$$s_k(t) = \sum_{f=1}^{F} \sum_{n\in\mathbb{Z}} g_f(t - nT)s_k(f,n). \tag{3.13}$$

An interesting property of such a time-frequency representation is that it leaves the linear instantaneous mixture model in Definition 2 page 43 unchanged: if $\boldsymbol{x}(f,n) \in \mathbb{C}^M$ (resp. $\boldsymbol{s}(f,n) \in \mathbb{C}^K$) denotes the vector of coefficients $x_m(f,n)$ (resp. $s_k(f,n)$), then the equality $\boldsymbol{x}(t) = \boldsymbol{A}\,\boldsymbol{s}(t)\ \forall t \in \mathbb{Z}$ is equivalent to

$$\forall f \in \{1 \dots F\}, \forall n \in \mathbb{Z},\ \boldsymbol{x}(f,n) = \boldsymbol{A}\,\boldsymbol{s}(f,n). \tag{3.14}$$

Indeed, if $a_{m,k}$ denotes the entries of the mixing matrix $\boldsymbol{A}$, we have $\forall m \in \{1 \dots M\}$,

$$x_m(f,n) = (h_f * x_m)(nT) = \left(h_f * \sum_{k=1}^{K} a_{m,k}s_k\right)(nT) = \sum_{k=1}^{K} a_{m,k}(h_f * s_k)(nT) = \sum_{k=1}^{K} a_{m,k}s_k(f,n).$$

3.4.2 Time-frequency source model

Let us now introduce a general non-stationary source model. As usual we assume that the K sub-band source signals $s_k(f,n)$ are centered and independent.

In Section 3.3.1 page 47, we have presented the SOBI BSS method, that exploits the *spectral dynamics* of the source signals, by modeling them as WSS processes. Remember that the well-known *spectral representation theorem* of WSS processes Brockwell and Davis [1987] shows that the Fourier transforms of WSS processes are formed of uncorrelated random elements whose variances vary over frequency.

In a similar way, in Section 3.3.2 page 48, we have presented a variant of the SOBI method, that exploits the *temporal dynamics* of the source signals, by modeling them as sequences of uncorrelated random variables whose variances vary over time.

Now, the use of a time-frequency representation allows use to jointly exploit the spectral and the temporal dynamics, just by modeling the samples of the sub-band source signals $s_k(f,n)$ for $f \in \{1 \dots F\}$ and $n \in \mathbb{Z}$ as uncorrelated random variables whose variance $\sigma_k^2(f,n)$ depends both on the frequency channel f and the time frame n:

Definition 7 (Non-stationary TF source model)**.** *We consider K independent source signals $s_k(t) \in \mathbb{R}$ with $t \in \mathbb{Z}$ and their TF representations $s_k(f,n)$ as defined in Section 3.4.1, concatenated in a vector $\boldsymbol{s}(f,n)$. For all $k \in \{1 \dots K\}$, the sub-band source signals $s_k(f,n)$ for $f \in \{1 \dots F\}$ and $n \in \mathbb{Z}$ are modeled as uncorrelated random variables of mean 0 and whose variance $\sigma_k^2(f,n)$ depends both on f and n. So $\boldsymbol{s}(f,n)$ is a random vector process of mean $\mathbb{E}[\boldsymbol{s}(f,n)] = \boldsymbol{0}$ and of TF-varying covariance matrix $\boldsymbol{\Sigma}_{ss}(f,n) = \mathbb{E}\left[\boldsymbol{s}(f,n)\boldsymbol{s}(f,n)^H\right]$.*

3.4.3 Separation method

This leads us to a new variant of the SOBI method in the determined case ($M = K$), based on the TF source model in Definition 7. Let us define the mixture covariance matrices $\boldsymbol{\Sigma}_{xx}(f,n) = \mathbb{E}[\boldsymbol{x}(f,n)\boldsymbol{x}(f,n)^H]$. Since $\boldsymbol{x}(f,n) = \boldsymbol{A}\,\boldsymbol{s}(f,n)$ (equation (3.14)), we have $\boldsymbol{\Sigma}_{xx}(f,n) = \boldsymbol{A}\boldsymbol{\Sigma}_{ss}(f,n)\boldsymbol{A}^\top$. Moreover, since the source signals are independent, matrix $\boldsymbol{\Sigma}_{ss}(f,n)$ is diagonal: $\boldsymbol{\Sigma}_{ss}(f,n) = \mathrm{diag}(\sigma_k^2(f,n))$.

Then, as in Section 3.2.1 page 44, we can *spatially* whiten the mixture signals:

- compute a matrix square root $\boldsymbol{S}$ of $\boldsymbol{\Sigma}_{xx} = \sum_{f,n} \boldsymbol{\Sigma}_{xx}(f,n)$;
- compute $\boldsymbol{W} = \boldsymbol{S}^{-1}$ and the whitened data

$$\forall f \in \{1 \dots F\}, \forall n \in \mathbb{Z},\ \boldsymbol{z}(f,n) = \boldsymbol{W}\boldsymbol{x}(f,n). \tag{3.15}$$

Again, as in Section 3.2.1 we can consider a canonical BSS problem and assume $\boldsymbol{\Sigma}_{xx} = \boldsymbol{A}\,\boldsymbol{A}^\top$, therefore matrix $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{A}$ is a rotation matrix.

Then if we consider the covariance matrices of the whitened data: $\forall f, n,\ \boldsymbol{\Sigma}_{zz}(f,n) = \mathbb{E}\left[\boldsymbol{z}(f,n)\boldsymbol{z}(f,n)^H\right]$, we get $\boldsymbol{\Sigma}_{zz}(f,n) = \boldsymbol{W}\boldsymbol{\Sigma}_{xx}(f,n)\boldsymbol{W}^H = \boldsymbol{W}\boldsymbol{A}\boldsymbol{\Sigma}_{ss}(f,n)\boldsymbol{A}^H\boldsymbol{W}^H = \boldsymbol{U}\boldsymbol{\Sigma}_{ss}(f,n)\boldsymbol{U}^H$. Therefore, as in Section 3.3.1 page 47, matrix $\boldsymbol{U}$ can be determined by solving a joint diagonalization problem Cardoso and Souloumiac [1996], e.g. by minimizing the following objective function:

$$J(\boldsymbol{U}) = \sum_{f,n} \|\boldsymbol{U}\boldsymbol{\Sigma}_{zz}(f,n)\boldsymbol{U}^H - \mathrm{diag}(\boldsymbol{U}\boldsymbol{\Sigma}_{zz}(f,n)\boldsymbol{U}^H)\|_F^2. \tag{3.16}$$

Finally, the source sub-band signals can be estimated as

$$\boldsymbol{y}(f,n) = \boldsymbol{U}^\top \boldsymbol{z}(f,n). \tag{3.17}$$

We thus get a variant of the SOBI algorithm Belouchrani et al. [1997], summarized in Algorithm 4.

Algorithm 4 SOBI method in the TF domain

TF analysis of mixture signals: $x_m(f,n) = (h_f * x_m)(nT)$ (equation (3.12))
Estimation of the covariance matrix $\boldsymbol{\Sigma}_{xx}$ and diagonalization: $\boldsymbol{\Sigma}_{xx} = \boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^H$ (equation (3.4))
Computation of $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}$ and of the whitening matrix $\boldsymbol{W} = \boldsymbol{S}^{-1}$ (equation (3.5))
Data whitening: $\boldsymbol{z}(f,n) = \boldsymbol{W}\boldsymbol{x}(f,n)$ (equation (3.15))
Estimation of covariance matrices $\boldsymbol{\Sigma}_{zz}(f,n)$ on all time-frequency bins
Approximate joint diagonalization of matrices $\boldsymbol{\Sigma}_{zz}(f,n)$ in a common basis $\boldsymbol{U}$ by minimizing (3.16)
Estimation of source signals via $\boldsymbol{y}(f,n) = \boldsymbol{U}^H\boldsymbol{z}(f,n)$ (equation (3.17))
TF synthesis of source signals: $y_k(t) = \sum_{f=1}^{F}\sum_{n\in\mathbb{Z}} g_f(t-nT)y_k(f,n)$ (equation (3.13))

4 Convolutive mixtures

As already mentioned in Section 1.2 page 39, linear instantaneous mixtures cannot accurately model real acoustic mixtures, since reverberation in a room involves convolutive effects. For this reason, we now address the extension of the BSS methods presented in Section 3 page 42 to convolutive mixtures.

4.1 Source images

First, suppose that K source signals $s_k(t)$ are simultaneously emitted in a room, and that M microphones receive the observed data vector $\boldsymbol{x}(t) \in \mathbb{R}^M$. The raw source separation problem would consist in estimating the *image* of each source k, i.e. the data vector $\boldsymbol{x}_k(t) \in \mathbb{R}^M$ that would be received by the M microphones if only source k was active. These images are such that $\boldsymbol{x}(f,n) = \sum_{k=1}^{K} \boldsymbol{x}_k(f,n)$. Then the task that consists in estimating the scalar source signals $s_k(t)$ from each vector image $\boldsymbol{x}_k(t)$ is called *deconvolution* or *dereverberation*.

In this way, the source separation problem is decomposed in two steps:

- **separation**: estimate the image $\boldsymbol{x}_k(f,n)$ from the mixture $\boldsymbol{x}(f,n)$
- **deconvolution**: estimate the source signal $s_k(f,n)$ from $\boldsymbol{x}_k(f,n)$

4.2 Convolutive mixture model

Let us now introduce the convolutive mixture model in the time domain:

Definition 8 (Convolutive mixture model)**.** *We consider K source signals $s_k(t) \in \mathbb{R}$ with $t \in \mathbb{Z}$, concatenated in a vector $\boldsymbol{s}(t)$. The samples of the M mixture signals $x_m(t) \in \mathbb{R}$ for $m \in \{1 \dots M\}$ are then defined as*

$$x_m(t) = \sum_{k=1}^{K} (a_{mk} * s_k)(t).$$

where $\forall k \in \{1 \dots K\}$, $\forall m \in \{1 \dots M\}$, a_{mk} is the impulse response of a stable[3] filter. In vector form, we will write $\boldsymbol{x}(t) = \boldsymbol{A} * \boldsymbol{s}(t)$ where $\boldsymbol{A}$ is an $M \times K$ matrix of stable impulse responses a_{mk}, and $*$ denotes the convolution product between a sequence of matrices and a sequence of vectors.

The following identifiability theorem Nguyen Thi and Jutten [1995] generalizes Theorem 1 page 44 to the convolutive case: it proves the feasibility of ICA under mild conditions about the source signals:

Theorem 3 (Identifiability theorem)**.** *Consider the source model in Definition 1, with the convolutive mixture model in Definition 8 in the determined case ($M = K$). Among the K IID sources s_k, suppose that at most one is Gaussian-distributed. Let $\boldsymbol{C}$ be a $K \times K$ matrix of stable impulse responses, and $\forall t \in \mathbb{Z}$, $\boldsymbol{y}(t) = \boldsymbol{C} * \boldsymbol{s}(t)$. Then the random processes $y_k(t)$ for $k \in \{1 \dots K\}$ are independent if and only if matrix $\boldsymbol{C}$ is non-mixing.*

Theorem 3 shows that the source signals can be retrieved up to an unknown permutation and an unknown scale factor.

4.3 Time-frequency approach

In order to simplify the problem and to make it possible to reuse the source separation methods introduced in Section 3, let us now rewrite the mixture model in Definition 8 in the time-frequency domain. More precisely, signals will be represented by their STFT, using the filterbank notation introduced in Section 3.4.1 page 49.

Moreover, we will consider the *narrow-band* approximation: we will assume that the impulse response of each mixing filter a_{mk} is short w.r.t. the time frame length of the STFT [4]. As a consequence, the spectral variations of the frequency responses $A_{mk}(\nu)$ are slow compared to those of $H_f(\nu)$ $\forall f \in \{1 \dots F\}$. Since h_f is a very narrow band-pass filter, we will even assume that $A_{mk}(\nu)$ is approximately constant in the pass-band of $H_f(\nu)$ (see Figure 3.6). Consequently, we can make the approximation $H_f(\nu)\, A_{mk}(\nu) \approx H_f(\nu)\, a_{mk}(f)$, where $a_{mk}(f)$ is the average value of $A_{mk}(\nu)$ in frequency channel f. Back in the time domain, this approximation can be rewritten $(h_f * a_{mk})(t) \approx a_{mk}(f)\, h_f(t)$.

If we now consider the STFT of the m-th mixture signal, we get

$$\begin{aligned}
x_m(f,n) &= (h_f * x_m)(nT) \\
&= \left(h_f * \left(\textstyle\sum_{k=1}^{K} a_{mk} * s_k\right)\right)(nT) \\
&= \left(\textstyle\sum_{k=1}^{K} \left(h_f * a_{mk}\right) * s_k\right)(nT) \\
&\approx \textstyle\sum_{k=1}^{K} a_{mk}(f) \left(h_f * s_k\right)(nT) \\
&= \textstyle\sum_{k=1}^{K} a_{mk}(f) s_k(f,n).
\end{aligned}$$

Hence the following approximate convolutive mixture model in the time-frequency domain:

Definition 9 (TF mixture model)**.** *We consider K source signals $s_k(t) \in \mathbb{R}$ with $t \in \mathbb{Z}$ and their TF representations $s_k(f,n)$ as defined in Section 3.4.1, concatenated in a vector $\boldsymbol{s}(f,n)$. For all $m \in \{1 \dots M\}$, the sub-band mixture signals $x_m(f,n)$ for $f \in \{1 \dots F\}$ and $n \in \mathbb{Z}$ are then defined as*

$$x_m(f,n) = \sum_{k=1}^{K} a_{mk}(f) s_k(f,n),$$

[3] Stability is defined in the *bounded-input, bounded-output* (BIBO) sense: if the input signal is bounded, then the output signal is also bounded.

[4] Note that this assumption is not realistic: the length of the impulse response a_{mk} corresponds to the reverberation time, which is usually several hundreds of miliseconds, while the typical time frame length in an STFT is a few tens of miliseconds. Nevertheless, this approximation is often used in audio source separation methods because it leads to a very simple mixture model in the TF domain (*cf.* Definition 9), which proves to perform well in various applications Ozerov and Févotte [2010].

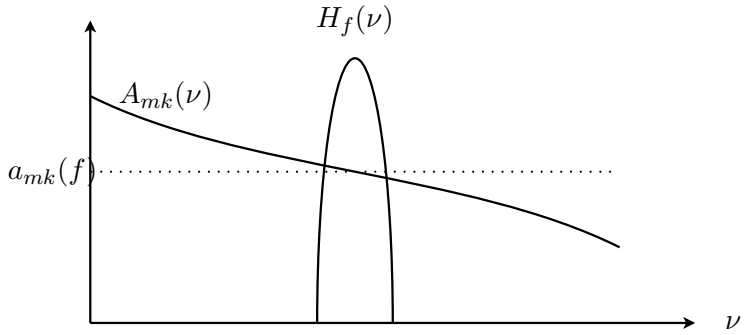


Figure 3.6: Narrow-band approximation

or in matrix form,

$$\boldsymbol{x}(f,n) = \boldsymbol{A}(f)\boldsymbol{s}(f,n), \tag{3.18}$$

where $\boldsymbol{A}(f) \in \mathbb{C}^{M\times K}$ *is the matrix of entries* a_{mk}.

It can be noted that in each frequency channel f, (3.18) is a linear instantaneous mixture model (to be compared to equation (3.2) page 43), parameterized by the mixing matrix $\boldsymbol{A}(f)$. Therefore we are tempted to apply any ICA method designed for the linear instantaneous mixture model, such as those described in Section 3, in every frequency channel of the STFT, in order to estimate the sub-band signals $s_k(f,n)$.

4.4 Independent component analysis

As in Section 3.1.2 page 43, it may seem natural to estimate the sub-band source signals as linear instantaneous combinations of the sub-band mixture signals: $\boldsymbol{y}(f,n) = \boldsymbol{B}(f)\,\boldsymbol{x}(f,n)$, where the entries of vector $\boldsymbol{y}(f,n)$ are the sub-band source signal estimates, and $\boldsymbol{B}(f) \in \mathbb{C}^{K\times M}$ is referred to as the *separation matrix*. Then the source separation problem amounts to finding an optimal separation matrix.

Linear source separation is generally feasible in the case of determined and over-determined mixtures:

- if $M = K$ and if matrix $\boldsymbol{A}(f)$ is invertible, then the separation matrix $\boldsymbol{B}(f) = \boldsymbol{A}(f)^{-1}$ leads to $\boldsymbol{y}(f,n) = \boldsymbol{s}(f,n)$;
- more generally, if $M \geq K$ and if matrix $\boldsymbol{A}(f)$ has full rank, then the separation matrix $\boldsymbol{B}(f) = \boldsymbol{A}(f)^{\dagger}$ leads to $\boldsymbol{y}(f,n) = \boldsymbol{s}(f,n)$.

However, in the under-determined case ($M < K$), linear source separation is generally not feasible (*cf.* Section 5).

In the determined case ($M = K$), independent component analysis Comon and Jutten [2010] can be applied in each frequency channel f. It aims to find a *separation matrix* $\boldsymbol{B}(f)$ that makes the K sub-band signals $y_k(f,n)$ independent. Then, as in Section 3.2 page 44, we get $\boldsymbol{y}(f,n) = \boldsymbol{C}(f)\boldsymbol{s}(f,n)$, where $\boldsymbol{C}(f) = \boldsymbol{B}(f)\boldsymbol{A}(f)$ is a non-mixing matrix (*cf.* Definition 3 page 43).

4.5 Indeterminacies

While the indeterminacies induced by the non-mixing matrix $\boldsymbol{C}$ were acceptable when we were considering linear instantaneous mixtures in Section 3, we now encounter an unexpected issue, because with the TF mixture model in Definition 9, there are F possibly different non-mixing matrices $\boldsymbol{C}(f)$. The problem is that, for instance, the permutations can be different in two different frequency channels f. If we choose to ignore that, and to just

reconstruct the source signals $y_k(t)$ from the separated sub-band signals $y_k(f,n)$ with the synthesis filters g_f, it is very likely that the resulting signals $y_k(t)$ will be formed of different sources s_k in different frequency channels. In other words, reconstructing the source signals from the separated sub-band signals would amount to remix the estimated sources!

In order to avoid this problem, we need to solve the permutation indeterminacy in the frequency channels of the STFT. Note that this multiple permutation issue is inherent to the time-frequency approach and to the narrow-band approximation introduced in Section 4.3: if BSS was performed in the time domain instead, Theorem 3 page 52 proves that all sources can theoretically be retrieved up to a unique permutation.

Even though, assuming that we do have a method that allows us to solve the multiple permutation indeterminacy, the other indeterminacy remains: there is an unknown multiplicative factor associated to each source in each frequency channel f.

Actually, both kinds of indeterminacies can be solved jointly, by introducing additional assumptions about the mixing filters a_{mk} and/or the source signals s_k:

- regarding the source signals, we can assume that the temporal dynamics (over n) of $\sigma_k^2(f,n)$ are similar between different frequency channels f for a same source k (nonparametric approach), or we can also exploit a parametric model, such as the *Nonnegative Matrix Factorization (NMF)* [Vincent et al., 2018, chap. 8] Ozerov and Févotte [2010].

- regarding the mixing filters, we can assume that their frequency responses $a_{mk}(f)$ are slowly varying w.r.t. f (nonparametric approach), or we can also exploit a parametric mixture model, such as the beamforming model or the anechoic model. The beamforming model [Vincent et al., 2018, chap. 10] relies on the plane wave and far field hypotheses (no reverberation) and assumes that the microphone antenna is linear. In this case, we get $a_{mk}(f) = e^{-2\imath\pi f\tau_{mk}}$ where $\tau_{mk} = \frac{d_m}{c}\sin(\theta_k)$, where parameters d_m denote the positions of the sensors on the linear antenna and parameters θ_k denote the angles of the sources (see Figure 3.7). The anechoic model is a bit more general: it assumes that the sources are punctual and that there is no reverberation. In this case, we get $a_{mk}(f) = \alpha_{mk}e^{-2\imath\pi f\tau_{mk}}$ where $\alpha_{mk} = \frac{1}{\sqrt{4\pi}r_{mk}}$, $\tau_{mk} = \frac{r_{mk}}{c}$, and parameters r_{mk} denote the distances between the sensors and sources. In practice, none of these two mixture models is able to accurately represent real acoustic mixtures; nevertheless they can be helpful to solve the multiple permutation problem.

5 Under-determined mixtures

As mentioned in Section 1.1 page 38, linear source separation is generally not feasible in the under-determined case, because there are infinitely many solutions. Without additional information about the mixture or the source signals, it is impossible to retrieve the original sources from the mixture signals.

Unfortunately, the under-determined case is often encountered in audio signal processing: indeed, many audio signals are either monophonic ($M = 1$) or stereophonic ($M = 2$), whereas the number of sources K is generally greater than 2. In this section, we will see how additional information can be taken into account to perform source separation in such a challenging scenario.

5.1 Under-determined convolutive mixtures

We still consider the TF mixture model in Definition 9 page 52 : $\boldsymbol{x}(f,n) = \boldsymbol{A}(f)\boldsymbol{s}(f,n)$ (equation (3.18)), and the TF source model in Definition 7 page 50: the samples of the sub-band source signals $s_k(f,n)$ for $f \in \{1 \dots F\}$ and $n \in \mathbb{Z}$ are uncorrelated random variables whose variance $\sigma_k^2(f,n)$ depends both on the frequency channel f and the time frame n, so that the covariance matrix of $\boldsymbol{s}(f,n)$ is $\boldsymbol{\Sigma}_{ss}(f,n) = \mathrm{diag}(\sigma_k^2(f,n))$.

As in Section 4.4 page 53, it may seem natural to estimate the sub-band source signals as linear instantaneous combinations of the sub-band mixture signals: $\boldsymbol{y}(f,n) = \boldsymbol{B}(f)\,\boldsymbol{x}(f,n)$, where the entries of vector $\boldsymbol{y}(f,n)$ are the sub-band source signal estimates, and $\boldsymbol{B}(f) \in \mathbb{C}^{K\times M}$ is referred to as the *separation matrix*. Then the source separation problem amounts to finding an optimal separation matrix.

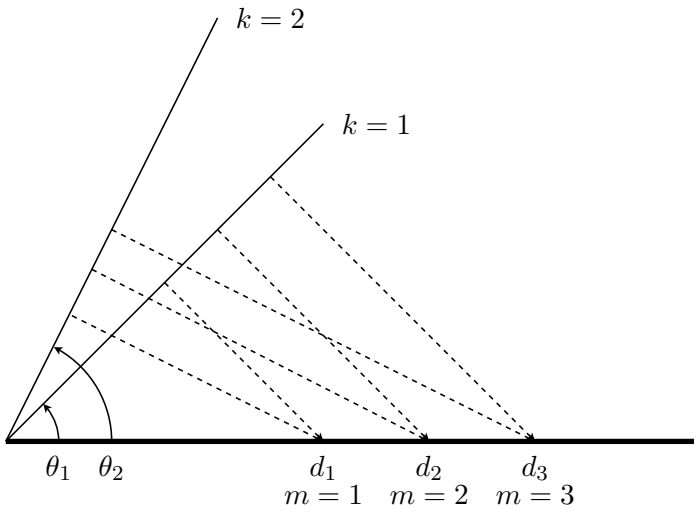


Figure 3.7: Beamforming mixture model

However if $M < K$, even if the mixing matrix $\boldsymbol{A}(f)$ and the source model $\boldsymbol{\Sigma}_{ss}(f,n)$ were known, the exact separation would not be feasible: there is no matrix $\boldsymbol{B}(f)$ such that $\boldsymbol{B}(f)\,\boldsymbol{A}(f) = \boldsymbol{I}_K$, because the maximum rank of matrix $\boldsymbol{B}(f)\,\boldsymbol{A}(f)$ is $M < K$ (*cf.* Figure 3.8).

However, we can still try to find an approximate solution $\boldsymbol{y}(f,n)$ in the least squares sense.

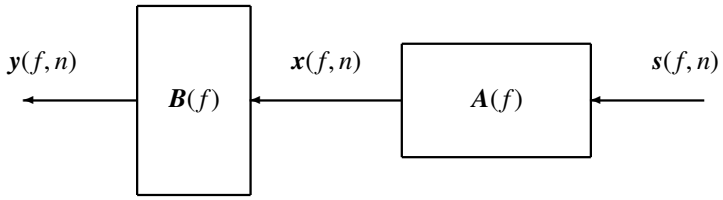


Figure 3.8: Under-determined mixtures: there is no matrix $\boldsymbol{B}(f)$ such that $\boldsymbol{B}(f)\,\boldsymbol{A}(f) = \boldsymbol{I}_K$

5.2 Separation via non-stationary filtering

First, we suppose that the mixing matrix $\boldsymbol{A}(f)$ and the source model $\boldsymbol{\Sigma}_{ss}(f,n)$ are known. Even though only an approximate solution $\boldsymbol{y}(f,n)$ can be obtained, this solution can be improved by adding a degree of freedom to the separation matrix $\boldsymbol{B}$: we will now make it depend on time n, so that the estimate $\boldsymbol{y}(f,n)$ is obtained by *non-stationary filtering*:

$$\boldsymbol{y}(f,n) = \boldsymbol{B}(f,n)\,\boldsymbol{x}(f,n) \tag{3.19}$$

where $\boldsymbol{B}(f,n) \in \mathbb{C}^{K\times M}$. The approximate solution will be found in the least squares sense, by considering the MMSE estimator Souden et al. [2013]: we will look for the separation matrix $\boldsymbol{B}(f,n)$ which minimizes the *Mean Square Error* (MSE) $\mathbb{E}[\|\boldsymbol{y}(f,n) - \boldsymbol{s}(f,n)\|_2^2]$. This MSE is such that

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{y}(f,n) - \boldsymbol{s}(f,n)\|_2^2] &= \mathbb{E}\left[(\boldsymbol{B}(f,n)\,\boldsymbol{x}(f,n) - \boldsymbol{s}(f,n))^H\,(\boldsymbol{B}(f,n)\,\boldsymbol{x}(f,n) - \boldsymbol{s}(f,n))\right] \\
&= \text{trace}\left(\mathbb{E}\left[(\boldsymbol{B}(f,n)\,\boldsymbol{x}(f,n) - \boldsymbol{s}(f,n))\,(\boldsymbol{B}(f,n)\,\boldsymbol{x}(f,n) - \boldsymbol{s}(f,n))^H\right]\right) \\
&= \text{trace}\left(\boldsymbol{B}(f,n)\boldsymbol{\Sigma}_{xx}(f,n)\boldsymbol{B}(f,n)^H - \boldsymbol{B}(f,n)\boldsymbol{\Sigma}_{xs}(f,n) - \boldsymbol{\Sigma}_{sx}(f,n)\boldsymbol{B}(f,n)^H + \boldsymbol{\Sigma}_{ss}(f,n)\right).
\end{aligned}$$

The MSE is minimized when the Wirtinger matrix gradient Gunning and Rossi [1965] w.r.t. $\boldsymbol{B}(f,n)$ is zero:

$$\boldsymbol{B}(f,n)\boldsymbol{\Sigma}_{xx}(f,n) - \boldsymbol{\Sigma}_{sx}(f,n) = \boldsymbol{0}.$$

Therefore the solution is given by $\boldsymbol{B}(f,n) = \boldsymbol{\Sigma}_{sx}(f,n)\,\boldsymbol{\Sigma}_{xx}(f,n)^{-1}$, where $\boldsymbol{\Sigma}_{xx}(f,n) = \boldsymbol{A}(f)\,\boldsymbol{\Sigma}_{ss}(f,n)\,\boldsymbol{A}(f)^H$ and $\boldsymbol{\Sigma}_{sx}(f,n) = \boldsymbol{\Sigma}_{ss}(f,n)\,\boldsymbol{A}(f)^H$. We can finally express the MMSE estimator as (3.19), with

$$\boldsymbol{B}(f,n) = \boldsymbol{\Sigma}_{ss}(f,n)\,\boldsymbol{A}(f)^H \left(\boldsymbol{A}(f)\,\boldsymbol{\Sigma}_{ss}(f,n)\,\boldsymbol{A}(f)^H\right)^{-1}. \tag{3.20}$$

We remark that the MMSE estimator guarantees the perfect reconstruction of the mixture signals from the estimated source signals: $\boldsymbol{A}(f)\,\boldsymbol{y}(f,n) = \boldsymbol{x}(f,n)$.

This MMSE estimator is also known as the *generalized* or *multichannel* Wiener filter, because in the particular case of monophonic mixtures ($M = 1$), it boils down to the well-known Wiener filter. Indeed, because of the scale indeterminacy of the model, we can assume without loss of generality that $\boldsymbol{A}(f) = [1,\ldots,1]$. Then the MMSE estimator defined by (3.19) and (3.20) can be rewritten as $y_k(f,n) = \frac{\sigma_k^2(f,n)}{\sum_{l=1}^K \sigma_l^2(f,n)}\, x(f,n)$, which is the usual form of the Wiener filter.

In practice of course, the mixing matrix $\boldsymbol{A}(f)$ and the source model $\boldsymbol{\Sigma}_{ss}(f,n)$ are unknown; they thus have to be estimated from the observed data. For instance, $\boldsymbol{A}(f)$ can be assumed slowly varying over f (nonparametric approach), or parameterized according to the beamforming or the anechoic model introduced in Section 4.5 page 53, and $\boldsymbol{\Sigma}_{ss}(f,n)$ can be assumed sparse in the TF domain as in Section 5.3.1 (nonparametric approach), or parameterized according to an NMF model [Vincent et al., 2018, chap. 8] Ozerov and Févotte [2010].

The resulting algorithm is sketched in Algorithm 5.

Algorithm 5 Under-determined source separation in the TF domain

TF analysis of mixture signals: $x_k(f,n) = (h_f * x_k)(nT)$ (equation (3.12))
Estimation of $\boldsymbol{A}(f)$ and $\sigma_k^2(f,n)$
Computation of $\boldsymbol{B}(f,n) = \boldsymbol{\Sigma}_{ss}(f,n)\,\boldsymbol{A}(f)^H \left(\boldsymbol{A}(f)\,\boldsymbol{\Sigma}_{ss}(f,n)\,\boldsymbol{A}(f)^H\right)^{-1}$ (equation (3.20))
Estimation of source sub-band signals as $\boldsymbol{y}(f,n) = \boldsymbol{B}(f,n)\,\boldsymbol{x}(f,n)$ (equation (3.19))
TF synthesis of source signals: $y_k(t) = \sum_{f=1}^F \sum_{n\in\mathbb{Z}} g_f(t-nT)y_k(f,n)$ (equation (3.13))

5.3 Stereophonic mixtures: separation based on sparsity

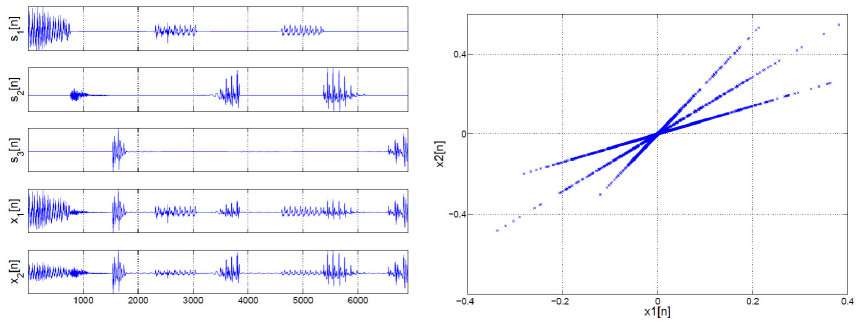
5.3.1 Temporal sparsity

We now consider the particular case of stereophonic ($M = 2$) linear instantaneous mixtures as in Definition 2 page 43 (defined by a unique mixing matrix $\boldsymbol{A}$ in order to simplify, but this approach would also work with the TF mixture model in Definition 9 page 52), so that the mixture model in the time domain is $\boldsymbol{x}(t) = \boldsymbol{A}\,\boldsymbol{s}(t)$.

We consider the example in Figure 3.9-(a): the $K = 3$ source signals represented in the three top lines of the figure are never active at the same time. We say that they are *sparse* in the time domain, in the sense that most of their temporal samples are zero. The $M = 2$ mixture signals are represented in the two bottom lines. Clearly, in this particular case of non-overlapping source signals, the source separation problem is a simple *classification* problem: it amounts to *segment* the mixture signals, and label the successive segments as "source 1", "source 2", etc.

In this simple scenario, the classification of the time samples can be easily performed by plotting the *dispersion diagram*, represented in Figure 3.9-(b): for every time t, the point of coordinates $(x_1(t), x_2(t))$ is drawn in the plane. This diagram clearly makes appear three straight lines, which correspond to the three sources. Indeed, when only source k is active, the linear instantaneous mixture model in Definition 2 yields $\boldsymbol{x}(t) = \boldsymbol{a}_k\, s_k(t)$, where $\boldsymbol{a}_k$ is the k-th column of matrix $\boldsymbol{A}$ (remember that because of the scale indeterminacy, we can assume without loss of generality that $\boldsymbol{a}_k$ is a unit vector). Therefore all points of coordinates $\boldsymbol{x}(t)$ in the plane that are generated by source k belong to the straight line passing through the origin and defined by the direction vector $\boldsymbol{a}_k$, and their position on this straight line corresponds to the value of the time sample $s_k(t)$.

Therefore in this simple case, source separation can be very easily performed by detecting the lines in the dispersion diagram (e.g. by using the Hough transform Stockman and Shapiro [2001]), which makes it possible to jointly estimate the column vectors $\boldsymbol{a}_k$ and the number of sources K. Then the images of the sources (defined in Section 4.1 page 51) can be retrieved by selecting the points $\boldsymbol{x}(t)$ that are the closest to each line, and finally the source signals $s_k(t)$ can be estimated by calculating the positions of these points on the line.



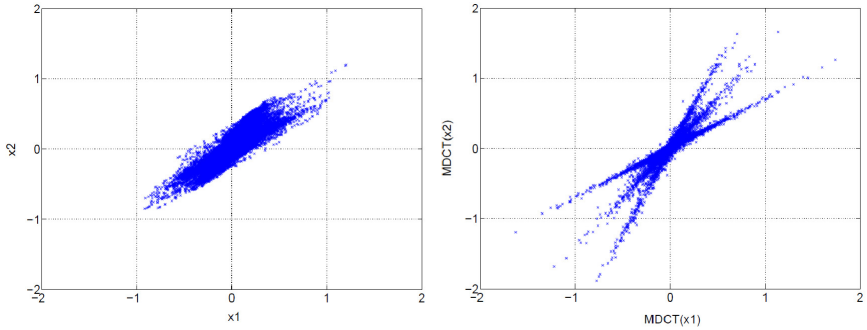
(a) Temporal source signals and corresponding stereo mixture (b) Dispersion diagrams (x_1, x_2) over time

Figure 3.9: Sparsity in time domain

5.3.2 Sparsity in a transformed domain

Unfortunately, the example in Figure 3.9 is not very realistic: it rarely happens, especially in music, that the different source signals do not overlap in the time domain. Figure 3.10-(a) shows an example of dispersion diagram obtained with a mixture of overlapping music sources: no straight line emerges from the cloud of points.

However, even when they do overlap in time, audio signals are often sparse in the TF domain: the spectrum of various sounds (especially in music) is made of a discrete set of frequencies. Figure 3.10-(b) shows another dispersion diagram obtained from the same mixture of music sources as in Figure 3.10-(a), except that the coordinates of the points are not obtained from the time samples $\boldsymbol{x}(t)$, but from the MDCT TF transform[5] $\boldsymbol{x}(f, n)$. The resulting dispersion diagram is not as clean as that of Figure 3.9-(b), but again three straight lines clearly emerge from the cloud of points, which shows that the mixture is made of $K = 3$ sources, which can be separated in the same way as in Section 5.3.1, but in the TF domain.



(a) Time samples (b) Time-frequency coefficients after MDCT decomposition

Figure 3.10: Sparsity in TF domain

[5]The MDCT is known to produce very sparse TF representations, which is why it is widely used in lossy audio data compression Luo [2009].

5.3.3 DUET method

We can now introduce the celebrated *Degenerate Unmixing Estimation Technique* (DUET) method Jourjine et al. [2000], Rickard [2007], which is dedicated to stereophonic ($M = 2$) mixtures, in the linear instantaneous mixture case: $\boldsymbol{x}(f,n) = \boldsymbol{A}\,\boldsymbol{s}(f,n)$ (equation 3.14 page 50). Without loss of generality, the column vectors of the mixing matrix $\boldsymbol{A}$ are parameterized as $\boldsymbol{a}_k = \begin{bmatrix} \cos(\theta_k) \\ \sin(\theta_k) \end{bmatrix}$, where $\theta_k \in \mathbb{R}$. Regarding the source signals, we consider a *sparse* source model in the TF domain:

Definition 10 (Sparse TF source model)**.** *We consider the same source model as in Definition 7. In addition, we assume that* $\forall f,n$*, there is a unique* $k_{(f,n)} \in \{1 \dots K\}$ *such that* $\sigma^2_{k_{(f,n)}}(f,n) > 0$*, and* $\forall l \neq k_{(f,n)}$*,* $\sigma^2_l(f,n) = 0$*.*

This additional assumption means that only one source can be active at any TF bin (f,n) (sources do not overlap in the TF domain). Therefore $\forall f,n$, $\boldsymbol{x}(f,n) = \boldsymbol{a}_{k_{(f,n)}} s_{k_{(f,n)}}(f,n)$.

The DUET method then consists of two steps: parameter estimation and source separation.

In the first step, the TF representation of the mixture signals is first computed by using the analysis filters h_f as in equation (3.12) page 49. Then, in order to estimate the mixture parameters θ_k, the histogram of the angles of vectors $\boldsymbol{x}(f,n)$ is first computed. The peaks of this histogram theoretically correspond to the angles θ_k, which can thus be estimated by performing peak detection. Then the active source $k_{(f,n)}$ at frequency bin (f,n) is estimated by selecting the angle θ_k which is the closest to the angle of the observed vector $\boldsymbol{x}(f,n)$.

In the second step, the source images (defined in Section 4.1 page 51) are estimated via binary masking Yilmaz and Rickard [2004]:

$$\forall k \in \{1 \dots K\},\ \boldsymbol{y}_k(f,n) = \begin{cases} \boldsymbol{x}(f,n)\ \forall (f,n) \text{ such that } k_{(f,n)} = k, \\ \boldsymbol{0} \text{ for the other time-frequency bins } (f,n). \end{cases} \tag{3.21}$$

Then the sub-band source signals are estimated with the MMSE estimator introduced in equation (3.20) page 56, which here boils down to a zero separation matrix $\boldsymbol{B}(f,n)$, except its k-th row which is[6]

$$\boldsymbol{a}_k(f)^\dagger = \frac{\boldsymbol{a}_k(f)^H}{\|\boldsymbol{a}_k(f)\|_2^2}. \tag{3.22}$$

Therefore the estimate of the k-th sub-band source signal as defined in (3.19) page 55 is

$$y_k(f,n) = \boldsymbol{a}_k(f)^\dagger \boldsymbol{y}_k(f,n). \tag{3.23}$$

Finally, the source signals are reconstructed in the time domain by using the synthesis filters g_f, as in equation (3.13) page 50. The DUET method is summarized in Algorithm 6.

Algorithm 6 DUET method

- TF analysis of mixture signals: $x_k(f,n) = (h_f * x_k)(nT)$ (equation (3.12))
- Estimation of parameters θ_k and of the active source $k_{(f,n)}$
  - Computation of the histogram of the angles of vectors $\boldsymbol{x}(f,n)$
  - Peak detection in order to estimate parameters θ_k
  - Determination of the active source at (f,n) by proximity with θ_k
- Source separation:
  - Estimation of source images $\boldsymbol{y}_k(f,n)$ via binary masking (equation (3.21))
  - MMSE estimation of sub-band source signals: $y_k(f,n) = \boldsymbol{a}_k(f)^\dagger \boldsymbol{y}_k(f,n)$ (equation (3.23))
- TF synthesis of source signals: $y_k(t) = \sum_{f=1}^{F} \sum_{n\in\mathbb{Z}} g_f(t - nT) y_k(f,n)$ (equation (3.13))

[6] In equation (3.20), matrix $\boldsymbol{\Sigma}_{ss}$ is singular, so the matrix inverse is replaced by the matrix pseudo-inverse, leading to equation (3.22).

6 Conclusion

In this chapter, we have reviewed several source separation models and methods, dedicated to determined linear instantaneous mixtures, determined convolutive mixtures, and under-determined mixtures. All these methods exploit the spatial diversity of the observed mixture signals (they require that $M > 1$), and their funding principle is that all source signals are statistically independent.

Source separation requires to make assumptions about the mixture and about the source signals, which are generally expressed in terms of probability distributions. For (over-)determined linear mixtures, we have seen that assuming independent sources is generally sufficient to make the separation possible (under mild conditions on the source probability distributions). When the mixture is under-determined however, it is necessary to make additional assumptions about the mixture and about the source signals, that can be formulated either in a non-parametric way (via regularization), or by exploiting parametric models.

This chapter forms an introduction to audio source separation, with a selection of models and methods; several topics that have been investigated in the literature could not been addressed here, for instance:

- The separation of non-stationary mixtures requires to develop adaptive separation algorithms Cardoso and Laheld [1996];
- *Informed source separation* techniques exploit some possibly available extra information about the mixture or the sources, such as the spatial positions of the sources and microphones (e.g. via *beamforming*), or the transcription of the source signals (speech or music) Ewert and Müller [2012];
- Deep learning techniques are able to automatically learn how to perform separation from a large database of source and mixture signals Vincent et al. [2018].
- Criteria for the objective assessment of audio source separation are required in order to compare the performance of various separation methods Vincent et al. [2006].

Bibliography

J. Allen. Overwiew of text-to-speech systems. In S. Furui and M. Sondhi, editors, *Advances in Speech Signal Processing*, chapter 23, pages 741–790. Marcel Dekker, 1991.

R. Badeau and A. Drémeau. Variational Bayesian EM algorithm for modeling mixtures of non-stationary signals in the time-frequency domain (HR-NMF). In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, pages 6171–6175, May 2013.

R. Badeau, N. Bertin, and E. Vincent. Stability analysis of multiplicative update algorithms and application to non-negative matrix factorization. *IEEE Transactions on Neural Networks*, 21(12):1869–1881, Dec. 2010.

A. Belouchrani, K. Abed-Meraim, J.-F. o. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

E. Benetos, G. Richard, and R. Badeau. Template adaptation for improving automatic music transcription. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 175–180, Oct. 2014.

J. Benson. *Audio Engineering Handbook*. mcGraw-Hill, New York, 1988.

N. Bertin. *Les factorisations en matrices non-négatives. Approches contraintes et probabilistes, application à la transcription automatique de musique polyphonique.* PhD thesis, École Nationale Supérieure des Télécommunications, Paris, France, Oct. 2009.

N. Bertin, C. Févotte, and R. Badeau. A tempering approach for Itakura-Saito non-negative matrix factorization. With application to music transcription. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, pages 1545–1548, Apr. 2009.

N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 18(3):538–549, Mar. 2010.

P. J. Brockwell and R. A. Davis. *The Spectral Representation of a Stationary Process*, pages 112–158. Springer New York, New York, NY, 1987.

O. Cappé, J. Laroche, and E. Moulines. Regularized estimation of cepstrum envelope from discrete frequency points. *IEEE ASSP Workshop on app. of sig. proc. to audio and acous.*, 1995.

J.-F. Cardoso and B. H. Laheld. Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, 44(12):3017–3030, 1996.

J.-F. o. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, 1998.

J.-F. o. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6): 362–370, 1993.

J.-F. o. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.

A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, Sept. 2009.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, April 1994. Special issue on Higher-Order Statistics.

P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, Inc. (Elsevier), USA, 1st edition, 2010.

J. Dattorro. Using digital signal processor chips in a stereo audio time compressor/expander. *Proc. 83rd AES Convention, New York*, Oct 1987. preprint 2500 (M-6).

A. El-Jaroudi and J. Makhoul. Discrete all pole modeling. *IEEE Trans. Acoust., Speech, Signal Processing*, 39(2): 411–423, Fev 1991.

S. Ewert and M. Müller. *Multimodal Music Processing*, volume 3, chapter Score-Informed Source Separation for Music Signals, pages 73–94. January 2012.

G. Fairbanks, W. Everitt, and R. Jaeger. Method for time or frequency compression-expansion of speech. *IEEE Trans. Audio Electroacoust.*, AU-2:7–12, Jan 1954.

J. A. Fessler and A. O. Hero. Space-alternating generalized expectation-maximization algorithm. *IEEE Transactions on Signal Processing*, 42(10):2664–2677, Oct. 1994.

C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural Computation*, 23(9):2421–2456, Sep. 2011.

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis. *Neural Computation*, 21(3):793–830, Mar. 2009.

L. Finesso and P. Spreij. Approximate nonnegative matrix factorization via alternating minimization. In *Proceedings of International Symposium on Mathematical Theory of Networks and Systems*, July 2004.

B. Fuentes, R. Badeau, and G. Richard. Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 21(9):1854–1866, Sept. 2013.

R. C. Gunning and H. Rossi. *Analytic Functions of Several Complex Variables*. AMS Chelsea Publishing. Prentice-Hall, Englewood Cliffs, N.J., USA, 1965.

E. Hardam. High quality time scale modification of speech signals using fast synchronized overlap add algorithms. *Proc. IEEE ICASSP-90*, pages 409–412, 1990.

R. Hennequin. Rapport à mi-parcours de travaux de thèse. Télécom ParisTech, Apr. 2010.

R. Hennequin, R. Badeau, and B. David. Time-dependent parametric and harmonic templates in non-negative matrix factorization. In *Proceedings of International Conference on Digital Audio Effects*, Sept. 2010.

R. Hennequin, R. Badeau, and B. David. NMF with time-frequency activations to model non-stationary audio events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 19(4):744–753, May 2011.

A. Hurmalainen, R. Saeidi, and T. Virtanen. Similarity induced group sparsity for non-negative matrix factorisation. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, pages 4425–4429, Apr. 2015.

D. Jones and T. Parks. On the generation and combination of grains for music synthesis. *Computer Music J.*, 12 (2):27–34, Summer 1988.

A. Jourjine, S. Rickard, and O. Yilmaz. Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures. In *Proc. of 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2985–2988, 2000.

M. Kahrs and K. Brandenbourg. *Applications of Digital Signal Processing to Audio and Acoustics*. Kluwer Academic Press, Dortrecht, Netherland, 1998.

H. Kameoka, N. Ono, K. Kashino, and S. Sagayama. Complex NMF: A new sparse representation for acoustic signals. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, pages 3437–3440, April 2009.

J. Laroche. Autocorrelation method for high quality time/pitch scaling. *IEEE ASSP Workshop on app. of sig. proc. to audio and acous.*, 1993.

H. Laurberg, M. Christensen, M. D. Plumbley, L. K. Hansen, and S. H. Jensen. Theorems on positive data: On the uniqueness of NMF. *Computational Intelligence and Neuroscience*, 2008, 2008. Article ID 764206, 9 pages.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791, Oct. 1999.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of Neural Information Processing Systems*, pages 556–562, Dec. 2001.

F. Lee. Time compression and expansion of speech by the sampling method. *J. Audio Eng. Soc.*, 20(9):738–742, 1972.

A. Lefevre, F. Bach, and C. Févotte. Itakura-Saito nonnegative matrix factorization with group sparsity. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, pages 21–24, May 2011.

C.-J. Lin. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19(10):2756–2779, Oct. 2007.

A. Liutkus and R. Badeau. Generalized Wiener filtering with fractional power spectrograms. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, pages 266–270, Apr. 2015.

A. Liutkus, D. Fitzgerald, and R. Badeau. Cauchy nonnegative matrix factorization. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 2015.

F.-L. Luo. *Mobile Multimedia Broadcasting Standards: Technology and Practice*. Springer Science & Business Media, January 2009.

J. Makhoul. Linear prediction: A tutorial review. *Proc. IEEE*, 63(11):1380–1418, Nov 1975.

J. Makhoul and A. El-Jaroudi. Time scale modification in medium to low rate speech coding. *Proc. IEEE ICASSP-86*, pages 1705–1708, 1986.

D. Malah. Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals. *IEEE Trans. Acoust., Speech, Signal Processing*, 27(2):121–133, 1979.

R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-34(4):744–754, Aug 1986.

E. Moulines and F. Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9(5/6):453–467, Dec 1990.

E. Moulines and J. Laroche. Non parametric techniques for pitch-scale and time-scale modification of speech. *Speech Communication*, 16:175–205, Feb 1995.

M. Nakano, H. Kameoka, J. Le Roux, Y. Kitano, N. Ono, and S. Sagayama. Convergence-guaranteed multiplicative algorithms for non-negative matrix factorization with beta-divergence. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, pages 283–288, 2010.

H.-L. Nguyen Thi and C. Jutten. Blind source separation for convolutive mixtures. *Signal Processing*, 45(2):209 – 229, 1995.

A. Ozerov and C. Févotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, 2010.

M. R. Portnoff. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24(3):243–248, Jun 1976.

S. Rickard. *The DUET Blind Source Separation Algorithm*, pages 217–241. Springer, Dordrecht, Netherlands, 2007.

F. Rigaud, B. David, and L. Daudet. A parametric model and estimation techniques for the inharmonicity and tuning of the piano. *Journal of the Acoustical Society of America*, 133(5):3107–3118, May 2013.

S. Roucos and A. M. Wilgus. High quality time-scale modification of speech. *Proc. IEEE ICASSP-85, Tampa*, pages 493–496, Apr 1985.

M. N. Schmidt and H. Laurberg. Non-negative matrix factorization with Gaussian process priors. *Computational Intelligence and Neuroscience*, 2008:1–10, 2008. Article ID 361705.

M. Schroeder, J. Flanagan, and E. Lundry. Bandwidth compression of speech by analytic-signal rooting. *Proc. IEEE*, 55:396–401, Mar 1967.

R. Scott and S. Gerber. Pitch-synchronous time-compression of speech. *Proceedings of the Conference for Speech Communication Processing*, pages 63–65, Apr 1972.

S. Seneff. System to independently modify excitation and/or spectrum of speech waveform without explicit pitch extraction. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24:358–365, 1982.

X. Serra and J. Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music J.*, 14(4):12–24, Winter 1990.

M. Shashanka, B. Raj, and P. Smaragdis. Probabilistic latent variable models as nonnegative factorizations. *Computational Intelligence and Neuroscience*, 2008:1–8, 2008. Article ID 947438.

U. Simsekli and A. T. Cemgil. Markov chain Monte Carlo inference for probabilistic latent tensor factorization. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2012.

P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, Oct. 2003.

P. Smaragdis, B. Raj, and M. Shashanka. Missing data imputation for time-frequency representations of audio signals. *Journal of Signal Processing Systems*, 65:361–370, Aug 2011.

M. Souden, S. Araki, K. Kinoshita, T. Nakatani, and H. Sawada. A multichannel MMSE-based framework for speech source separation and noise reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1913–1928, 2013.

G. Stockman and L. G. Shapiro. *Computer Vision*. Prentice Hall PTR, USA, 1st edition, 2001.

B. Sylvestre and P. Kabal. Time-scale modification of speech using an incremental time-frequency approach with waveform structure compensation. *Proc. IEEE ICASSP-92*, pages 81–84, 1992.

P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Inc., USA, 1993.

W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. *Proc. IEEE ICASSP-93, Minneapolis*, pages 554–557, Apr 1993.

E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.

E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 18(3):528–537, Mar. 2010.

E. Vincent, T. Virtanen, and S. Gannot. *Audio Source Separation and Speech Enhancement*. Wiley Publishing, 1st edition, 2018.

T. Virtanen, A. T. Cemgil, and S. Godsill. Bayesian extensions to non-negative matrix factorisation for audio signal modelling. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, pages 1825–1828, Apr. 2008.

J. Wayman and D. Wilson. Some improvements on the sychronized-overlap-add method of time scale modification for use in real-time speech compression and noise filtering. *IEEE Trans. Acoust., Speech, Signal Processing*, 36 (1):139–140, Jan 1988.

O. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, 2004.

Institut Mines-Télécom

Spectral and temporal modifications

Roland Badeau

In this practical work, you will implement the PSOLA method for the analysis/synthesis of speech signals. This method will be tested on signals that can be downloaded on the TSIA 206 Moodle. These signals are sampled at `Fs`. You can load them with Matlab e.g. by typing up `load aeiou`; they will then be stocked in variable `s`. To listen to them, you can type up `soundsc(s,Fs)`. In Python, you can use the provided notebook template `template-TP-modifications.ipynb` that you can download on the website.

1 Extraction of the analysis marks

Firstly, you will program the following function

`function A = AnalysisPitchMarks(s,Fs)`

which extracts the analysis marks. The arguments `s` and `Fs` respectively are the signal to be analyzed and the sampling frequency. The returned matrix `A` will contain the times and pitches corresponding to each analysis mark. More precisely, `A` will be formed of three rows, such that `A(1,n)` $= t_a(n)$ is the time corresponding to the n^{th} analysis mark ($t_a(n) \in \mathbb{N}$ is expressed in number of samples), `A(2,n)` = voiced(n) is a Boolean which indicates whether the signal is voiced or unvoiced in the neighborhood of this mark, and `A(3,n)` $= P_a(n) \in \mathbb{N}$ describes the pitch corresponding to the same mark (i.e. the period expressed in number of samples) in the voiced case, or equals 10ms $\times$ `Fs` in the unvoiced case.

To do so, you will need a pitch estimator. In order to spare time, you can use function `period.m`, whose Matlab code is provided with the example signals, and whose Python code is included in the notebook template `template-TP-HR.ipynb`. This function requires two arguments: a short term signal `x` extracted from `s`, and the sampling frequency `Fs` (the other arguments are optional), and returns a couple `[P, voiced]` where `voiced` is a Boolean which indicates whether `x` is voiced or non, and `P` $\in \mathbb{N}$ is the period expressed in number of samples in the voiced case, or equals 10ms $\times$ `Fs` in the unvoiced case.

Let us now detail how to determine the analysis marks. For the sake of simplicity, we will not try to align the mark $t_a(n)$ on the beginning of a glottal pulse. To compute $P_a(n)$ and $t_a(n)$, we proceed by recursion on $n \geq 1$:

- extraction of a sequence `x` that starts at time $t_a(n-1)$, and whose duration is equal to 2.5 $P_a(n-1)$;
- computation of $P_a(n)$ and voiced(n) by means of function `period`;
- computation of $t_a(n) = t_a(n-1) + P_a(n)$.

The algorithm will be initialized by setting $t_a(0) = 1$ (in Matlab) or $t_a(0) = 0$ (in Python) and $P_a(0) =$ 10ms $\times$ `Fs`.

2 Synthesis and modification of the temporal and spectral scales

To perform the synthesis of the signal, we must start by defining the synthesis marks. They will be stocked in a matrix `B` formed of two rows, such that `B(1,k)` $= t_s(k)$ is the time corresponding to the k^{th} synthesis mark, and `B(2,k)` $= n(k)$ is the index of the analysis mark corresponding to this same synthesis mark. To start, you can perform a synthesis without modification, by setting:

- `B(1,:)` = `A(1,:)`;
- `B(2,:)` $= [1, 2, 3, \ldots]$.

2.1 Signal synthesis

You will now program the following function

```
function y = Synthesis(s,Fs,A,B)
```

which computes the synthesis signal `y` from the original signal `s`, the sampling frequency `Fs`, the analysis marks stocked in matrix `A` and the synthesis marks stocked in matrix `B`. The synthesis is very simply performed by recursion on $k \geq 1$ (vector `y` being initialized to the zero vector of dimension $t_s(k_{\text{end}}) + P_a(n(k_{\text{end}}))$):

- extraction of a sequence `x` centered at $t_a(n(k))$ and of length $2\,P_a(n(k)) + 1$;
- windowing of `x` by a Hann window (Matlab function `hann` or Python function `scipy.signal.hanning`);
- overlap-add of the sequence `x` windowed on $y(t_s(k) - P_a(n(k)) : t_s(k) + P_a(n(k)))$.

2.2 Modification of the temporal scale

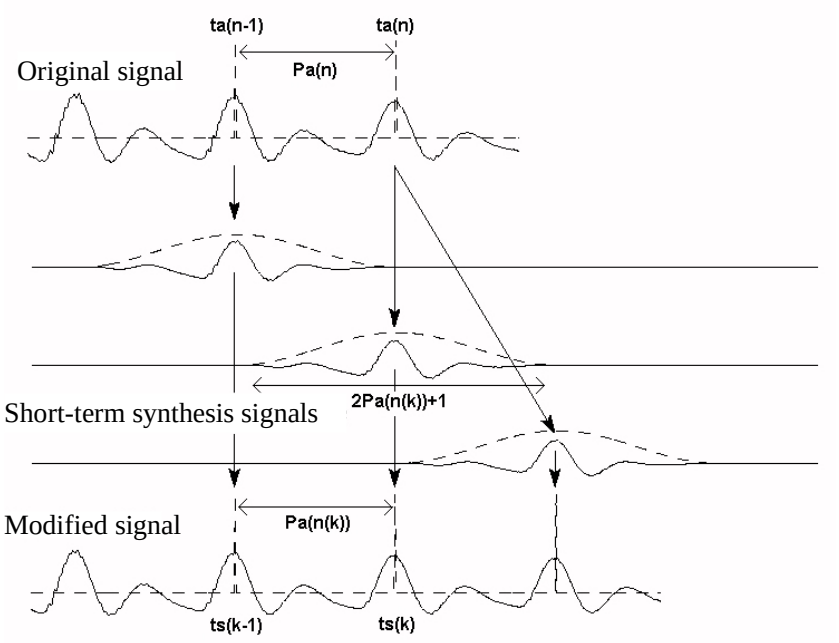


Figure 1: Modification of the temporal scale

We now want to determine the synthesis marks that will modify the temporal scale by a factor α, i.e. to determine a matrix `B` such that the duration of the signal synthesized by function `Synthesis` is equal to that of the original signal `s` multiplied by α. This operation will be performed by function

```
function B = ChangeTimeScale(alpha,A,Fs)
```

which computes matrix `B` from the factor α, the analysis marks stocked in `A`, and the sampling frequency `Fs`. You can proceed by recursion on $k \geq 1$, by using a non-integer index $n(k)$:

- $t_s(k) = t_s(k-1) + P_a(\lfloor n(k) \rfloor)$;
- $n(k+1) = n(k) + \frac{1}{\alpha}$.

The algorithm will be initialized by setting $t_s(0) = 1$ and $n(1) = 1$. You will take care of only stocking integer values in matrix `B`.

2.3 Modification of the spectral scale

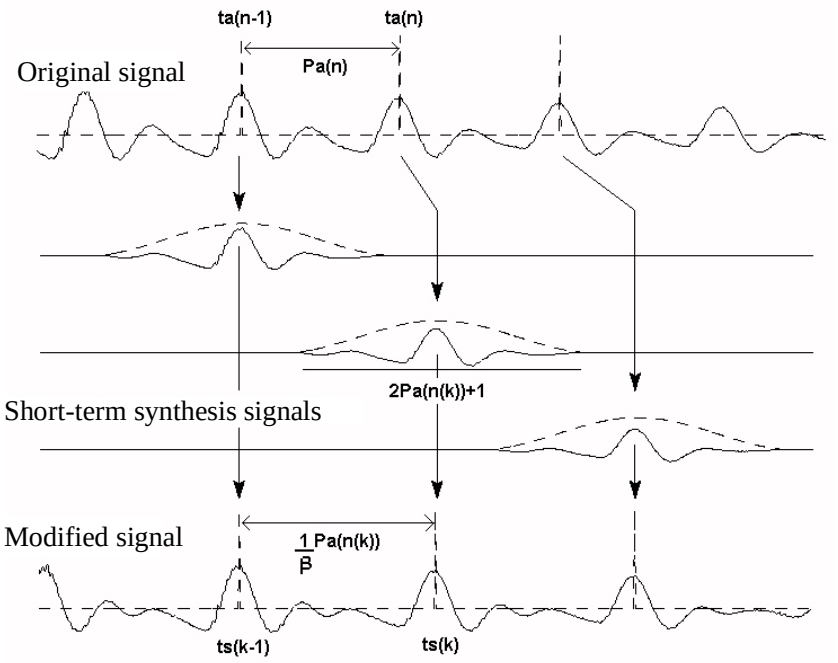


Figure 2: Modification of the spectral scale

You will now perform the dual operation of the previous one: determine the synthesis marks that will modify the spectral scale by a factor β, i.e. determine a matrix `B` such that the fundamental frequency of the signal synthesized by function `Synthesis` is equal to that of the original signal `s` multiplied by β. This operation will be performed by function

```
function B = ChangePitchScale(beta,A,Fs)
```

which computes matrix `B` from the factor β, the analysis marks stocked in `A`, and the sampling frequency `Fs`. As in the previous case, you can proceed by recursion on $k \geq 1$, by using a non-integer index $n(k)$ and non-integer synthesis times $t_s(k)$, and by making the difference between the voiced and unvoiced cases:

- if the analysis mark of index $\lfloor n(k) \rfloor$ is voiced, $\text{scale}(k) = \frac{1}{\beta}$, otherwise $\text{scale}(k) = 1$;
- $t_s(k) = t_s(k-1) + \text{scale}(k) \times P_a(\lfloor n(k) \rfloor)$;
- $n(k+1) = n(k) + \text{scale}(k)$.

Again, you will take care of only stocking integer values in matrix `B`.

2.4 Joint modification of the temporal and spectral scales

To finish, you will program a function that jointly modifies the two scales:

```
function B = ChangeBothScales(alpha,beta,A,Fs)
```

where the arguments are defined as previously. The content of this function will be almost identical to that of `ChangePitchScale`; you will just need to modify it properly.

Institut Mines-Télécom

Practical Work on Non-Negative Matrix Factorization

Roland Badeau
from the subject written by U. Simsekli

The files related to this practical work are available on the TSIA 206 Moodle.

Non-Negative Matrix Factorization with β-divergence

In this practical work, we will deal with non-negative matrix factorization (NMF) with the β-divergence. The problem that we aim to solve is given as follows:

$$(W^\star, H^\star) = \underset{W\geq 0, H\geq 0}{\arg\min} \sum_{i=1}^{I}\sum_{j=1}^{J} d_\beta(x_{ij}||\hat{x}_{ij}), \tag{1}$$

where x_{ij} is an element of $X \in \mathbb{R}_+^{I\times J}$, that is the *non-negative* data matrix, and $W \in \mathbb{R}_+^{I\times K}$ and $H \in \mathbb{R}_+^{K\times J}$ are the unknown *non-negative* factor matrices. We also define $\hat{x}_{ij} = \sum_k w_{ik}h_{kj}$. The cost function that we are minimizing is called the β-divergence, which is defined as follows:

$$d_\beta(x||\hat{x}) = \frac{x^\beta}{\beta(\beta-1)} - \frac{x\hat{x}^{\beta-1}}{\beta-1} + \frac{\hat{x}^\beta}{\beta}. \tag{2}$$

When $\beta = 1$, we obtain the Kullback-Leibler (KL) divergence, when $\beta = 0$ we obtain the Itakura-Saito (IS) divergence.

One of the most popular algorithms for NMF is called the multiplicative update rules (MUR). The MUR algorithm has the following update rules:

$$W \leftarrow W \circ \frac{(X \circ \hat{X}^{\beta-2})H^\top}{\hat{X}^{\beta-1}H^\top} \tag{3}$$

$$H \leftarrow H \circ \frac{W^\top(X \circ \hat{X}^{\beta-2})}{W^\top\hat{X}^{\beta-1}}, \tag{4}$$

where $\circ$ denotes element-wise multiplication and $/$ and $\div$ denote element-wise division.

Questions:

1. By following the technique that we used in the lecture, derive the MUR algorithm by yourselves.
2. Fill in the template provided in the notebook which will require you to implement the MUR update rules.
3. Experiment with the algorithm parameters, such as β, number of columns in W, STFT window size, STFT hop size etc. What do you observe?

Roland Badeau
from the subject written by U. Simsekli

Contexte académique } sans modifications

Roland Badeau
from the subject written by U. Simsekli

Institut Mines-Télécom

Practical work on audio source separation

Roland Badeau

In this practical work, you will program a simple implementation of a variant of the DUET (*Degenerate Unmixing Estimation Technique*) method, which aims to separate sound sources in a stereophonic mixture. You will thus address the case of an under-determined ($M = 2$ sensors and $K > 2$ sources) instantaneous linear mixture. The separation is achieved by exploiting the spatial information, and by assuming that the sources are sparse in the time-frequency plane (a single source is active at each time-frequency bin). The transformation that you will use is the MDCT (*Modified Discrete Cosine Transform*), which presents the double advantage of having real values, and of producing a sparser representation than the STFT (short time Fourier transform). In order to compute it, you will use the *Linear Time/Frequency Toolbox* (`ltfat`) toolbox in Matlab, that you can download on the TSIA 206 Moodle, or you can use the provided Python notebook template `template-TP-separation.ipynb` which is based on the `mdct` toolkit. You will also find on this website the stereophonic sound file to be processed, named `mix.wav`. In order to use the functions of the `ltfat` toolbox in Matlab, you will first have to load it by calling function `ltfatstart`.

1 Mixture model and principle of the DUET method

The DUET method relies on the following mixture model: at every time-frequency bin (f, n),

$$\boldsymbol{X}(f,n) = \boldsymbol{S}(f,n)\,\boldsymbol{A}$$

where

- the row vectors $\boldsymbol{X}(f,n) = [X(f,n,1), X(f,n,2)]$ of dimension $M = 2$ contain the MDCT of the two stereophonic channels $x(t,m)$ of the observed mixture;
- the $K \times M$ matrix $\boldsymbol{A} = [\cos(\boldsymbol{\theta}), \sin(\boldsymbol{\theta})]$ is the mixing matrix;
- the K-dimensional column vector $\boldsymbol{\theta}$ contains the angles $\theta(k)$ of sources k;
- the K-dimensional row vectors $\boldsymbol{S}(f,n) = [S(f,n,1), \ldots, S(f,n,K)]$ contain the MDCT of the K unknown source signals.

If only source k is active at (f,n), the point of affix $Z(f,n) = X(f,n,1) + iX(f,n,2) \in \mathbb{C}$ is such that $Z(f,n) = S(f,n,k)\,e^{i\theta(k)}$, where $S(f,n,k) \in \mathbb{R}$. We remark that its argument permits us both to identify the active source k at (f,n) and its angle $\theta(k)$. The magnitude of $Z(f,n)$ permits us to determine the value of $S(f,n,k)$, up to its sign. In order to remove the sign ambiguity of $S(f,n,k)$, we can assume that $\theta(k) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Once source k is identified at every time-frequency bin, a binary mask $B(f,n,k)$ can be applied to $X(f,n,m)$, in order to obtain an estimation $Y(f,n,m,k)$ of the stereophonic image of source k. The source signal is finally reconstructed by means of the MMSE (*Minimum Mean Square Error*) estimator, which is such that $S(f,n,k) = Y(f,n,1,k)\cos(\theta(k)) + Y(f,n,2,k)\sin(\theta(k))$.

2 Work to do

1. Open file `mix.wav` and load it in a $T \times M$ matrix $x(t,m)$, where $M = 2$ and T is the number of samples. Use your headphones to listen to the mixture. What is the number K of instruments that you can hear? From which direction do you perceive them?

2. Plot the temporal dispersion diagram, defined as the set of points in the plane of coordinates $(x(t,1), x(t,2))$ for all t (in order to plot a set of points, you can use the Matlab function `plot` or the Python function `matplotlib.pyplot.plot` with parameter `'x'`, and you can normalize the axes with the Matlab instruction `axis equal` or the Python Matplotlib function `axis('equal')`). Can you distinguish the directions of the sources?

3. Compute the MDCT $X(f,n,m)$ of the two stereophonic channels $x(t,m)$ (you can use the Matlab function `wmdct`, with $F = 512$ frequency bands and the window `'sqrthann'`, or the Python function `mdct`). Plot the corresponding time-frequency representations $|X(f,n,m)|^2$ (you can use the Matlab function `plotwmdct` or the Python function `matplotlib.pyplot.imshow`).

4. Plot the time-frequency dispersion diagram, defined as the set of points in the plane of affix $Z(f,n)$ for all f and n. Can you distinguish the directions of the sources? How do you explain it?

5. Plot the histogram of the arguments of the points of affix $Z(f,n)$ for all f and n (you can use the Matlab function `atan` or the Python function `numpy.arctan` to compute the arguments modulo π, between $-\frac{\pi}{2}$ and $+\frac{\pi}{2}$, and the Matlab function `hist` or the Python function `matplotlib.pyplot.hist` to compute the histogram, whose number of classes has to be tuned so as to make the directions of the sources clearly visible). Estimate the angles $\theta(k)$ (you can determine these values graphically from the histogram).

6. In order to estimate the active source at every time-frequency bin (f,n), you can look for the source k whose angle $\theta(k)$ is closest to the argument of $Z(f,n)$, modulo π (you can use a deviation measure invariant modulo π, for instance $|\sin(\theta(k) - \angle Z(f,n))|$). Then generate the binary masks $B \in \{0,1\}$, such that $B(f,n,k)$ is equal to 1 if source k is active at (f,n), or 0 otherwise.

7. Apply masks B to the MDCT $X(f,n,m)$ in order to estimate the MDCT of the stereophonic images $Y(f,n,m,k)$. Then reconstruct the images $y(t,m,k)$ of the source signals by applying the inverse MDCT (you can use the Matlab function `iwmdct` or the Python function `imdct`).

8. Listen to the K reconstructed stereophonic images $y(:,:,k)$. What defects can you perceive?

9. Compute the MMSE estimator $S(f,n,k)$ of source k. Reconstruct the source signals $s(t,k)$ by applying the inverse MDCT to $S(f,n,k)$. Listen to the result.

10. We now wish to respatialize the sources, i.e. to resynthesize the mixture $x(t,m)$ by modifying the angles $\theta(k)$ (remark that it is not needed to switch back to the MDCT domain). For instance, try to permute the directions of the sources. Listen to the result. What audible defects can you notice?