Kernel additive models for source separation

Antoine Liutkus*[1,2,3], Derry Fitzgerald[4], Zafar Rafii[5], Bryan Pardo[5], Laurent Daudet[6]

[1]Inria, Villers-lès-Nancy, F-54600, France
[2]Université de Lorraine, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
[3]CNRS, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
[4]NIMBUS Centre, Cork Institute of Technology, Ireland
[5]Northwestern University, Evanston, IL, USA
[6]Institut Langevin, Paris Diderot Univ., France

***Abstract*—Source separation consists of separating a signal into additive components. It is a topic of considerable interest with many applications that has gathered much attention recently. Here, we introduce a new framework for source separation called Kernel Additive Modelling, which is based on local regression and permits efficient separation of multidimensional and/or nonnegative and/or non-regularly sampled signals. The main idea of the method is to assume that a source at some location can be estimated using its values at other locations *nearby,* where nearness is defined through a source-specific *proximity kernel*. Such a kernel provides an efficient way to account for features like periodicity, continuity, smoothness, stability over time or frequency, self-similarity, etc. In many cases, such *local dynamics* are indeed much more natural to assess than any global model such as a tensor factorization. This framework permits one to use different proximity kernels for different sources and to separate them using the iterative *kernel backfitting* algorithm we describe. As we show, kernel additive modelling generalizes many recent and efficient techniques for source separation and opens the path to creating and combining source models in a principled way. Experimental results on the separation of synthetic and audio signals demonstrate the effectiveness of the approach.**

***Index Terms*—MLR-SSEP, SSP-SSEP, MLR-GRKN, MLR-MUSI**

I. Introduction

Source separation (see [1] for a review) is a research field that has gathered much attention during the last 30 years. Its objective is to recover several unknown signals called *sources* that were mixed into observable *mixtures*. It has applications in telecommunication, audio processing, latent components analysis, biological signals processing, etc.

The objective of *blind* source separation is to estimate the sources given the mixtures only. There are many ways to formulate this problem and many different approaches have been undertaken to address this challenging task. Among them, we can mention three different paradigms that have attracted much of the attention of researchers in the field.

LD is partly supported by the DReaM project of the French Agence Nationale de la Recherche (ANR-09-CORD-006, under CONTINT program). This work is partly supported by LABEX WIFI (Laboratory of Excellence within the French Program "Investments for the Future") under references ANR-10-LABX-24 and ANR-10-IDEX-0001-02 PSL.

One of the first efficient approaches to source separation is denoted Independent Components Analysis (ICA, see [2], [1]). It performs signal separation through the assumptions that the sources are all probabilistically independent and distributed with respect to a non-Gaussian distribution. Given these assumptions, contrast features representative of both non-Gaussianity and independence are maximized, leading to the recovery of separated signals. The main issue with these approaches is that they are hard to extend to *underdetermined* source separation, i.e. when less mixtures than sources are available. Furthermore, many signals of interest are usually poorly modelled as independent and identically distributed (i.i.d.), a common assumption in ICA.

A second set of techniques for source separation is grounded in state-space modelling. Indeed, it can be expressed in terms of an adaptive filtering problem where the hidden state of the system is composed of the sources and where the observation process leads to the resulting mixtures [3], [4], [5]. Source separation in this context can be performed through state-inference techniques. The main issue with this approach is that the sources can rarely be modelled as obeying linear dynamic models. Meanwhile, tractable nonlinear adaptive filtering is often restricted to very local dependencies. Furthermore, the computational cost of the methods has hindered their widespread use in practice. Still, some studies [6], [7], [8] have demonstrated that a state-space model is appropriate to account for the dynamics of audio spectra in many cases. Following from this, nonnegative dynamical systems were recently introduced [9] to perform efficient separation of nonnegative sources defined through a state-space model.

Finally, a third approach, which is currently the dominating paradigm for the underdetermined separation of waveforms, is the use of generalized Wiener filtering [10], [11], [12] under Gaussian assumptions [13]. In practice, it can be shown [14] that this approach reduces to decomposing the spectrograms of the mixtures into the spectrograms of the sources. The corresponding waveforms are then easily recovered. Most related methods rely on the idea that the sources are likely to exhibit some kind of spectral redundancy. This can be efficiently captured through dimension reduction methods like Nonnegative Matrix/Tensor Factorizations (NMF/NTF, [15], [16], [17], [18]).

In spite of their appealing tractability and their performance on many signals, the aforementioned ideas often have limitations. First, some sources like the human voice are hard to model with a few fixed spectral templates. Studies such as [19], [18] address this issue and introduce more sophisticated models, but they may require a careful tuning

in practice [20]. Second, these techniques typically assume that different sources are characterized by different sets of spectra, which may not be realistic in many cases, like in a string quartet for instance.

In this study, we do not attempt to decompose the sources as combinations of fixed patterns. Instead, we focus on their *regularities* to identify them from the mixtures. To motivate this, we can consider the case of musical signals. Audio sources can exhibit extremely complex spectrograms that sometimes cannot be modelled using block structures such as NTF. However, their *local dynamics* may be understood as obeying more simple rules. Auditory Scene Analysis [21] demonstrates on perceptual grounds that our ability to discriminate sources within a mixture largely depends on local features such as repetitivity, continuity or common fate. These dynamic features do not depend on any particular spectral template, but rather on local *regularities* concerning their evolution over time and frequency. If several studies have already addressed the problem of introducing regularities within the parameters of block models [22], [16], [23], [9], only a few focused on modelling the correlations within nonnegative sources [24]. In any case, these techniques can for now only account for a small number of correlations, thus strongly limiting their expressive power.

We focus on the modelling and separation of signals which are defined on arbitrary input spaces, meaning that the approach is applicable to both 1D signals (e.g. audio time-series) and multi-dimensional data. In order to model local dependencies within a source, we assume that it can locally be approximated by a parametric model such as a polynomial. If its values at some locations are not directly observable from the data, they can be estimated using its values at other locations through *local regression* [25], [26], [27], [28]. Usually, this local fitting is handled using a sliding window of adjacent locations, yielding smooth estimates. Instead, we introduce the general concept of a source-dependent *proximity kernel,* that gives the proximity of any two locations to use for local fitting. This direct generalization permits to account for signals that are not necessarily smooth, which is often the case in practice.

If we observe a mixture and want to estimate the value of one of the sources at some location, the method we describe assumes that the contribution of all other sources will average out during local regression and that separation can hence be performed in an iterative manner. In practice, we introduce a variant of the *backfitting* algorithm [29], which can use a different proximity kernel for each source. This approach is flexible enough to take prior knowledge about the dynamics of many kinds of signals into account. In the context of audio processing, we show that it encompasses a large number of recently proposed methods for source separation [30], [31], [32], [33], [34], [18] and provides an efficient way to devise new specific separation algorithms for sources that are characterized by local features, rather than by a global additive model such as NTF.

This text is organized as follows. First, we introduce kernel local parametric models for the sources in section II. Then, we consider the case of mixtures of such sources in section III and present an algorithm for their separation that we call *kernel backfitting* (KBF). In section IV, we illustrate the effectiveness of the approach for the separation of 1D mixtures contaminated by strong impulsive noise. Finally, we discuss the application of the framework to audio sources in section V and show that KAM is efficient for the separation of the vocal part in musical signals.

II. Kernel local parametric models

Throughout this paper, all signals are understood as functions $f(l)$ giving the value of the signal at any location l. For example, in the case of a time series, l will be a time position or a sample index, while $f(l) \in \mathbb{R}$ is the corresponding value for the waveform. In another setting, for image or field measurements for instance, l may be a spatial location and $f(l)$ the signal value at that position. Such a formulation permits one to handle both regularly and non-regularly sampled data in a principled way.

In this section, we present an approach to model the local dynamics of signals. Its principle is to locally approximate a signal through a parametric model. For this purpose, it is necessary to choose a parametric family of approximations but also to define the particular weighting function used for the local fitting. This weighting function, called a *proximity kernel*, may not be based on the standard Euclidean distance, but rather on some knowledge concerning the signal. This kernel local regression is an important building block of the separation procedure we present in section III.

A. Kernel local regression

Local regression [26], [28], [35] is a method that was initially introduced for smoothing scatter plots. Formally, let $\mathbb{L}$ denote an arbitrary space called input space and let us assume that our objective is to estimate a signal $f : \mathbb{L} \to \mathbb{R}$ based on N noisy observations (l_n, z_n), where $l_n \in \mathbb{L}$ is a location and $z_n \in \mathbb{R}$ is the observed value at that location. We write

$$\mathcal{D} = \{(l_n, z_n)\}_{n=1\cdots N}$$

as the set gathering the N available measurements.

The first step in local parametric modelling is to assess a relation between the signal we seek to estimate and the observations. Usually, each observation z_n is assumed to be the sum of $f(l_n)$ with some white additive noise ϵ_n:

$$z_n = f(l_n) + \epsilon_n. \tag{1}$$

More generally, we will consider that the negative log-likelihood $\mathcal{L}(z_n \mid f(l_n))$ of the observations z_n given $f(l_n)$, also called the *model cost function* in the following, is known:

$$\mathcal{L}(z_n \mid f(l_n)) = -\log p(z_n \mid f(l_n)). \tag{2}$$

This probabilistic formulation permits us to handle noise in a more flexible manner than (1). By selecting the appropriate

probability density function, noise may be modeled as additive, multiplicative, or some other relation. Throughout this paper, we suppose that all observations z_n are independent[1].

The main idea of local parametric modelling, reminiscent of Taylor series, is to consider that for a particular position $l^\star \in \mathbb{L}$, the signal f can be *locally* approximated using a member $\mathcal{F}_{\theta(l^\star)}$ of some given parametric set of functions $\mathcal{F}$. In that case, $\theta(l^\star)$ denotes the set of parameters defining the function $\mathcal{F}_{\theta(l^\star)}$. For example, $\mathcal{F}$ can be the set of all polynomials of a given order and $\theta(l^\star)$ is then a particular set of coefficients. Finally, $f(l^\star)$ is estimated as:

$$\hat{f}(l^\star) = \mathcal{F}_{\theta(l^\star)}(l^\star), \tag{3}$$

with local parameters $\theta(l^\star)$ chosen as:

$$\theta(l^\star) = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_{\mathcal{D}}(l_n, l^\star)\, \mathcal{L}(z_n \mid \mathcal{F}_\theta(l_n)), \tag{4}$$

where $w_{\mathcal{D}}(l_n, l^\star)$ is a *known* nonnegative weight giving the importance of having a good fit $\mathcal{L}(z_n \mid \mathcal{F}_\theta(l_n))$ at l_n for estimating $f(l^\star)$. The rationale behind the weight function in (4) is that a good choice for $\theta(l^\star)$ is only required to be good locally. In the literature, having $\mathbb{L} = \mathbb{R}^d$ with some $d \in \mathbb{N}$ is common, $\mathcal{F}$ is mostly chosen as the set of linear functions $\mathcal{F}_{\alpha,\beta} = \left\{l \mapsto \alpha + \beta^\top l\right\}_{\alpha,\beta}$ and $\mathcal{L}$ as the squared error, leading to the following cost function:

$$(\alpha^\star, \beta^\star) = \underset{(\alpha,\beta)}{\operatorname{argmin}} \sum_{n=1}^{N} w_{\mathcal{D}}(l_n, l^\star)\left(z_n - \alpha - \beta^\top l_n\right)^2. \tag{5}$$

This is easily solved and leads to the estimate $\hat{f}(l^\star) = \alpha^\star + \beta^{\star\top} l^\star$. When $\mathcal{F}$ is chosen as the class of constant functions (having $\beta = 0$), and $\mathcal{L}$ as the squared error, (5) is solved by the Nadaraya-Watson [36], [25] estimate:

$$\hat{f}(l^\star) = \frac{\sum_{n=1}^{N} w_{\mathcal{D}}(l_n, l^\star)\, z_n}{\sum_{n=1}^{N} w_{\mathcal{D}}(l_n, l^\star)}, \tag{6}$$

which is essentially a weighted average of the observations around $l^\star$. The parametric space $\mathcal{F}$ and the penalty function $\mathcal{L}$ to use can strongly depend on the application.

The presentation above is slightly more general than what is common in the literature. First, $\mathbb{L}$ is often taken as Euclidean equipped with a norm $l \in \mathbb{L} \mapsto \|l\| \in \mathbb{R}_+$. Second, the weight function $w_{\mathcal{D}}(l_n, l^\star)$ that gives the proximity of l_n to $l^\star$ in (4) is typically given in terms of their *distance* $\|l_n - l^\star\|$, justifying the name *local regression* for the approach. Here, we purposefully did not make these assumptions, because we allow the value $f(l^\star)$ at some location $l^\star$ to depend not necessarily on its neighbours under the initial input metric, but rather on neighbours under some arbitrary metric defined by a proximity kernel $w_{\mathcal{D}}(l, l^\star)$. This further flexibility adds improved expressive power.

[1]Assuming the observations to be independent does not mean that no relationship is to be expected between them. In the additive formulation (1) for instance, it only means that the additive noises ϵ_n are independent.

B. Proximity kernels

We refer to the weight function $w_{\mathcal{D}}(l, l^\star)$ as a proximity kernel. Its output must be non-negative and should increase as the importance of using $f(l)$ to estimate $f(l^\star)$ increases. It may be implemented using a distance metric based on the location of l and $l^\star$. This leads to the kind of kernel typical in the local regression literature. We begin with an example of such kernels and then show how they can be generalized to significantly increase the power of the approach.

1) An example proximity kernel for local regression: Most existing proximity kernels $w_{\mathcal{D}}(l, l')$ found in the local regression literature are *stationary*, i.e. functions of the *distance* $\|l - l'\|$ between their operands. This can for instance be written as:

$$w_{\mathcal{D}}(l, l') = \delta\left(\frac{\|l - l'\|}{h}\right), \tag{7}$$

where h is called the *bandwidth* in this context and is chosen using the measurements $\mathcal{D}$. δ is usually taken as a smoothly decreasing function[2] of its argument like the tricube function [28]:

$$\delta(\tau) = \begin{cases} \left(1 - |\tau|^3\right)^3 & \text{for } |\tau| \leq 1 \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

This kind of choice for $w_{\mathcal{D}}(l, l')$ is motivated by its intuitive connection with the notion of proximity of l from l' when $\mathbb{L}$ has an Euclidean topology.

2) Generalized proximity kernels: In the present study, we show how more general proximity kernels can be used to significantly reduce the size of $\mathcal{F}$, and thus the computational complexity of the optimization problem (4). The main idea is to choose a kernel $w_{\mathcal{D}}(l, l')$ that is not necessarily related to the Euclidean distance $\|l - l'\|$ between l and l' in $\mathbb{L}$, but rather to how much we expect the parametric approximations of f at l and l' to share the same parameters. This way, estimation of the parametric model $\mathcal{F}_{\theta(l^\star)}$ to estimate $f(l^\star)$ in (4) may depend on points l that are "far" from $l^\star$ in the Euclidean distance, but for which $w_{\mathcal{D}}(l, l^\star)$ is high.

Furthermore, even if a proximity kernel $w_{\mathcal{D}}(l, l^\star)$ is a function of the locations l and $l^\star$, it must be emphasized that it may also depend on the data $\mathcal{D}$ as is notably the case in robust local regression [26]. In short, $w_{\mathcal{D}}(l, l^\star)$ is a positive quantity, which is high whenever we expect f to share the same parameters at l and $l^\star$, *in light of the data*.

3) Example: pseudo-periodic signals: In order to illustrate these ideas, consider the example given in figure 1. We assume $\mathbb{L} = \mathbb{R}$ and we suppose that f is a function from $\mathbb{R}$ to $\mathbb{R}_+$, which is known to be periodic with period T, but not necessarily smooth. To model f, one can either pick $\mathcal{F}$ as the set of all positive trigonometric polynomials of period T and choose a global fitting strategy, or simply choose $\mathcal{F}$ as the set of constant functions and, for example, define $w_{\mathcal{D}}(l, l') = 1$ if and only if $l - l' = pT$ ($p \in \mathbb{Z}$) and 0

[2]A function is usually called smooth if it is derivable. The more derivable, the smoother it is.

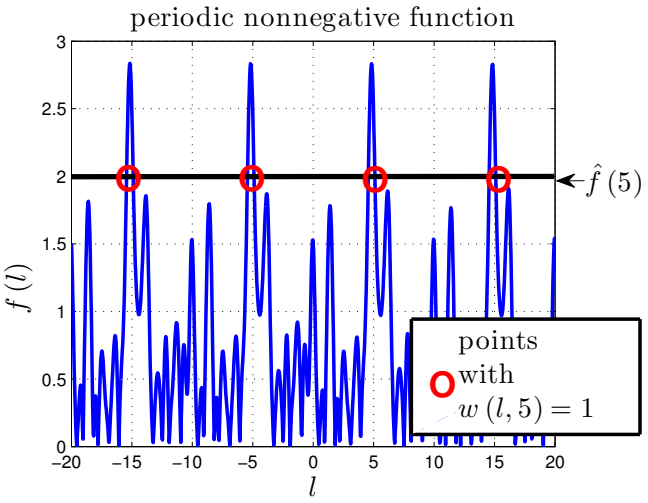

Figure 1. Nonnegative periodic function of period 10. Since f is periodic, $f(l)$ is identical to $f(l+10p)$ with $p \in \mathbb{Z}$ for any l, justifying the use of a locally constant model with a periodic proximity kernel.

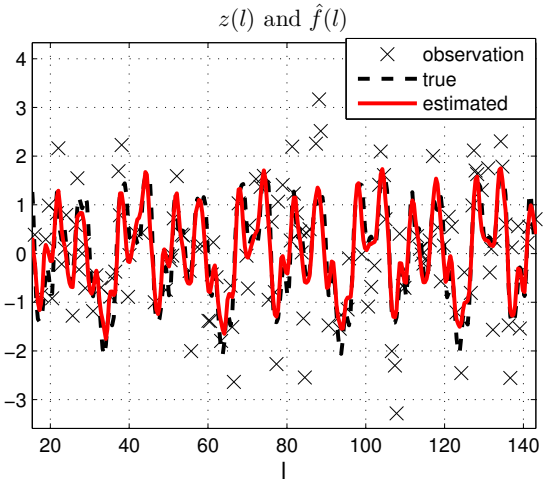

Figure 2. Kernel local regression of a non-regularly sampled and pseudo-periodic time-series mixed with white Gaussian noise of variance $\sigma^2 = 1$, using a constant model with a pseudo-periodic proximity kernel (9). Estimation is done using the Nadaraya-Watson estimate (6).

otherwise. This accounts for the fact that for any l, $f(l)$ is identical to $f(l+pT)$ with $p \in \mathbb{Z}$, because f is periodic. Further if f is assumed to be smooth, this can easily be expressed with a proximity kernel that additionally includes some proximity within each period [13]:

$$w_{\mathcal{D}}(l,l') = \exp\left(-\frac{2}{\lambda^2}\sin^2\left(\pi\frac{l-l'}{T}\right) - \frac{|l-l'|^2}{2P^2T^2}\right), \quad (9)$$

where P is a parameter indicating for how many periods the signal is known to be self-similar, while λ denotes the phase distance $\sin^2\left(\pi\frac{l-l'}{T}\right)$ required for two samples $f(l)$ and $f(l')$ to become independent.

On figure 2, we show an example of kernel local regression where the observations z_n are the sum of a non-regularly sampled locally-periodic signal f with additive white Gaussian noise of variance $\sigma^2 = 1$. f is modelled as locally constant with proximity kernel (9). Estimation is hence performed using the classical Nadaraya-Watson estimate (6) using this non-conventional proximity kernel.

The above example is representative of the expressive power of the proposed method. Instead of focusing on complex parametric spaces to globally model the signals, kernel local parametric modelling fits simpler models, but adapts the notion of proximity for the estimation of f based on some prior knowledge about its dynamics. Put otherwise, when the proximity kernel $w_{\mathcal{D}}(l,l')$ is high whenever the values of $f(l)$ and $f(l')$ are similar and negligible otherwise, simple spaces $\mathcal{F}$ of *smooth* functions such as low order polynomials may be used in (4), even if f is not smooth with respect to the canonical topology of $\mathbb{L}$. Above, $\mathcal{F}$ has been simplified from the heavily parameterized set of nonnegative periodic functions to the trivial set of constant functions.

4) Some examples of proximity kernels: Many studies in the literature [28], [35], [37], [38], [39] focus on the case of an Euclidean input space $\mathbb{L} = \mathbb{R}^d$ with $d \in \mathbb{N}$ and the proximity kernel is chosen [35], [39] as:

$$w_{\mathcal{D}}(l,l') = |H_{\mathcal{D}}|^{-1}\,\delta\left(H_{\mathcal{D}}^{-\frac{1}{2}}(l-l')\right)$$

where $H_{\mathcal{D}}$ is a $d\times d$ symmetric positive definite matrix called the *bandwidth matrix*, because it is the direct multivariate extension of the *bandwidth parameter* h of (7). δ is a smoothly decreasing function of the norm of its argument like the tricube function (8). The choice of $H_{\mathcal{D}}$ is generally data dependent and a local choice of $H_{\mathcal{D}}$ has provided good edge-preserving estimates in image processing [38], [39].

Other proximity kernels of interest include k-nearest neighbours ($k-$NN) kernels [27]. For a location $l^\star \in \mathbb{L}$, such kernels are defined as assigning a non-zero proximity value $w_{\mathcal{D}}(l,l^\star) > 0$ to at most $k \in \mathbb{N}$ locations l, called the the *nearest neighbours* of $l^\star$ and denoted $\mathcal{I}(l^\star) \in \mathbb{L}^k$. Several cases of $k-$NN kernels can be found, such as the uniform $k-$NN, that assigns the same proximity to all k nearest neighbours of $l^\star$. Some examples of $k-$NN kernels will be given in sections IV and V.

Finally, we also mention kernels that are obtained through the *embedding* $\phi_{\mathcal{D}}$ of $\mathbb{L}$ into a *feature space* Φ of arbitrary dimension through:

$$\phi_{\mathcal{D}} : l \in \mathbb{L} \mapsto \phi(l) \in \Phi. \quad (10)$$

The feature space Φ is assumed to be equipped with a dot product $\langle \phi(l), \phi(l') \rangle_\Phi$ and the proximity kernel $w_{\mathcal{D}}(l,l')$ to use can be chosen as:

$$w_{\mathcal{D}}(l,l') = \langle \phi(l), \phi(l') \rangle_\Phi. \quad (11)$$

Alternatively, the proximity kernel $w_{\mathcal{D}}(l,l')$ can be a k-NN kernel based on the distance in the feature space. This is notably the case for the *nonlocal means* method [40] for image denoising that computes the similarity $w_{\mathcal{D}}(l,l')$ of two locations based on the similarity of the observations in

their respective neighbourhoods. When the embedding (10) does not depend on $\mathcal{D}$, proximity kernels (11) have the noticeable property of always being positive definite [41], [42], which is a key element in many kernel methods. Conversely, any positive definite kernel can be shown [42] to be the dot product of some feature space. Thus, the embedding (10) may either involve the effective computation of a set of features for each l_n, or one may simply choose any positive definite function as a proximity kernel, a method which is known as the *kernel trick* in the specialized literature.

C. Comparison with Bayesian nonparametric methods

The local regression framework can be seen as a particular instance of the kernel method [42]. In our context, it has several advantages compared to other regression frameworks such as Gaussian Processes (GP, [41]). First, it allows the proximity kernel $w_{\mathcal{D}}$ to be a function of the observations $\mathcal{D}$, which is not possible through a consistent Bayesian treatment based on GP [43]. Second, it can easily permit non-negativity of the estimates given nonnegativity of the observations. Indeed, provided $w_{\mathcal{D}}(l, l') \geq 0$ and $\forall n, z_n \geq 0$, the simple Nadaraya-Watson estimate (6) is for instance also nonnegative. This feature may be important in some applications like audio processing as we show in section V. Third, contrary to the GP case, noise distributions that are not Gaussian can easily be taken into account in the local regression case. Finally, kernel local regression has the important advantage of being computationally very efficient for some choices of $w_{\mathcal{D}}$ and $\mathcal{L}$. For example, computations involved in the example displayed in figure 2 involve $\mathcal{O}\left(N^2\right)$ operations whereas consistent regression using GP would have involved the inversion of a $N \times N$ covariance matrix, requiring $\mathcal{O}\left(N^3\right)$ operations in the general case. Whenever $w_{\mathcal{D}}$ has limited support, meaning that $w_{\mathcal{D}}(\cdot, l')$ is nonzero for at most $k \ll N$ locations, complexity of local regression can drop down to $\mathcal{O}(kN)$.

Of course, this approach is not always the most appropriate for signal modelling, because its performance strongly depends on the assumption that the true underlying function can locally be approximated as lying in some given —and known— parametric set, which may not be the case. On the contrary, fully nonparametric Bayesian methods such as GP do not require such an assumption. Still, there are many cases of practical interest where a parametric model may locally be a very good fit, which is for example demonstrated by the ubiquitous use of Taylor series in science.

III. Kernel additive models

In this section, we assume that the measured signal, called the *mixture* and written x, is a noisy observation which depends on $J > 1$ functions s_j called the *sources*. A common example is the case of a sum $x_n = \sum_j s_j(l_n)$ of the sources. Our objective becomes the estimation of all J sources s_j and thus to achieve *source separation*.

The particularity of the approach we propose is that each source s_j is modelled locally using a kernel local parametric model as defined above, with its own parametric family $\mathcal{F}^{(j)}$ and proximity kernel[3] w_j. As we will illustrate in sections IV and V, this feature is central and permits the combination of different source separation methods in a principled way.

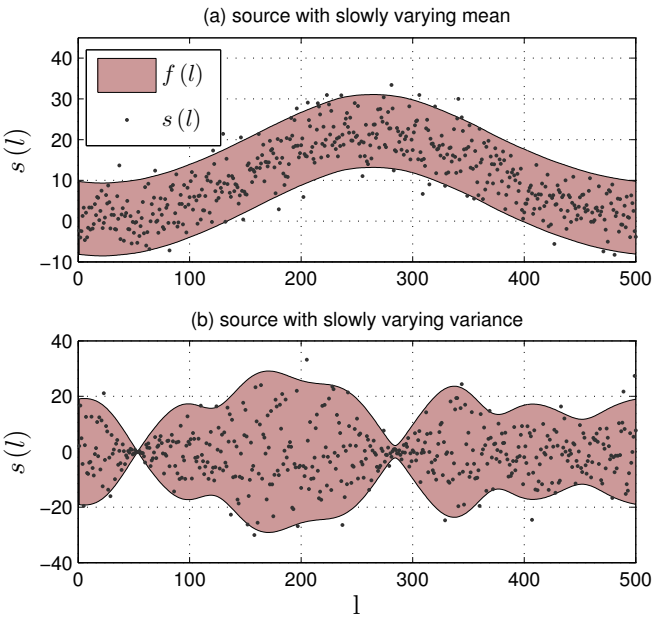

Figure 3. Two examples of sources modelled through local kernel models. (a) Independent Gaussian samples with a varying mean. (b) Independent Gaussian samples with a varying variance.

A. Formalization

Let x be a signal called the *mixture*, defined on an arbitrary input space $\mathbb{L}$ and taking values in $\mathbb{C}^I$, which means that for all $l \in \mathbb{L}$, $x(l)$ is a $I \times 1$ complex vector. The mixture depends on J underlying signals s_j called the *sources*. Each source s_j is also a function from $\mathbb{L}$ to $\mathbb{C}^I$. We assume that x is observed at N locations, yielding the observed data $\mathcal{D} = \{(l_n, x_n)\}_{n=1,\ldots,N}$.

The first step in Kernel Additive Modelling (KAM) is to model the source signals. Formally, for a source j, all samples $s_j(l)$ are assumed independent and their distribution is driven by some location-dependent hyperparameters. To illustrate this, consider the examples given in figure 3. In figure 3(a), the source samples have a slowly varying mean. In figure 3(b), they have a slowly varying variance. In both cases, their distribution depends on a slowly varying latent variable. Other parameters may also be provided, such as the variance σ^2 of all samples in figure 3(a).

In the general case, all samples $s_j(l)$ are assumed independent and their likelihood is known and given by the

[3]For conciseness, we drop the $\mathcal{D}$ subscript for the proximity kernels, but it must be emphasized that every proximity kernel considered may depend on the data.

source cost function[4]:

$$\mathcal{L}_{s_j|\Gamma_j}\left(s_j\left(l\right)\right) = -\log p\left(s_j\left(l\right) \mid \Gamma_j\left(l\right)\right), \tag{12}$$

where $\Gamma_j\left(l\right)$ is a set of source parameters that identifies which distribution to use. It is split in two parts:

$$\Gamma_j\left(l\right) = \left\{f_j\left(l\right), R_j\left(l\right)\right\}. \tag{13}$$

Among all parameters, $f_j\left(l\right)$ is called a *latent explanatory function* and is the one that undergoes local modelling, while $R_j\left(l\right)$ gathers all other parameters. For instance, the source cost function for figure 3(a) corresponds to a Gaussian distribution with mean $f\left(l\right)$ and variance $\sigma^2 \in R_j\left(l\right)$ and is thus $\mathcal{L}_{s|f}\left(s\left(l\right)\right) = \left|s\left(l\right) - f\left(l\right)\right|^2/\sigma^2$. As such, model (12) is classical and simply assesses the relation between source signals s_j and some parameters of their distributions.

The second and most noticeable step in KAM is to model the latent explanatory functions f_j. In the literature, it is common to assume that they are well described as a member $\mathcal{F}^{(j)}_{\theta_j}$ of some parametric family $\mathcal{F}^{(j)}$ of functions, such as in variance modelling with NTF [44], [18]. Here, we drop this global assumption and rather focus on a *local* model. Each latent explanatory function f_j is approximated in the vicinity of location $l^\star$ as:

$$f_j\left(l^\star\right) \approx \mathcal{F}^{(j)}_{\theta_j(l^\star)}\left(l^\star\right), \tag{14}$$

where $\mathcal{F}^{(j)}$ is a parametric family of functions and $\theta_j\left(l^\star\right)$ some location-dependent parameters. If we assume as in section II that some noisy observations z_j of f_j are available, the local parameters $\theta_j\left(l^\star\right)$ in (14) are chosen as:

$$\theta_j\left(l^\star\right) = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_j\left(l_n, l^\star\right) \mathcal{L}_j\left(z_n \mid \mathcal{F}^{(j)}_{\theta}\left(l_n\right)\right), \tag{15}$$

where w_j is the proximity kernel of source j as defined above in section II-B and $\mathcal{L}_j\left(z_n \mid u\right)$ is a known *model cost function* for source j. It is the penalty of choosing $f_j\left(l_n\right) = u$ when its noisy observation is z_n.

The final step is to assess the relation between the mixture x, the sources $s_1, \ldots, s_J$ and their parameters $\Gamma_1, \ldots, \Gamma_J$. This is done by specifying the *separation cost function*, $\mathcal{L}_{s|x,\Gamma}$, which describes our knowledge on how to perform a good separation given some set of parameters. If we adopt a probabilistic perspective, it may be understood as the negative log-likelihood of the sources given all their parameters and the mixture:

$$\begin{aligned}&\mathcal{L}_{s|x,\Gamma}\left(s_1\left(l_n\right), \ldots, s_J\left(l_n\right)\right)\\ &\quad = -\log p\left(s_1\left(l_n\right), \ldots, s_J\left(l_n\right) \mid x_n, \Gamma_1 \ldots \Gamma_J\right).\end{aligned} \tag{16}$$

In that case, it may be derived by combining the source cost function (12) with a known mixing model $p\left(x_n \mid s_1\left(l_n\right), \ldots, s_J\left(l_n\right)\right)$ through Bayes' theorem. However, some studies have demonstrated that sticking to that probabilistic interpretation may not always be advantageous and that user-defined separation cost functions may yield very good results [45]. For that reason, we retain an optimization perspective and simply assume $\mathcal{L}_{s|x,\Gamma}$ is given. Two different examples are given in sections IV and V. Furthermore, this broad definition (16) permits handling more complex mixing scenarios than simple sums, like a product of sources or non-linear mixing.

With the source, model and separation cost functions in hand, source separation amounts to computing the estimates $\hat{s}_j$, $\hat{f}_j$and $\hat{R}_j$ that jointly minimize all cost functions (12), (15) and (16). Whereas this may seem daunting to solve in full generality, we now adapt the ideas of backfitting in order to perform the estimation iteratively.

B. Kernel backfitting algorithm

The problem above has been extensively studied. In particular, if $\mathbb{L} = \mathbb{R}^J$ and in the additive case $x = \sum_j s_j$, if we assume that the sources coincide with the latent explanatory variables[5], having $s_j = f_j$ and that each source $s_j\left(l\right)$ only depends on the j^{th} coordinate of l, this problem has been extensively studied under the name of Generalized Additive Models (GAM, [29], [46]). The more general case where each function $s_j\left(l\right) = f_j\left(l\right)$ depends on a particular projection $a_j^\top l$ of the input has been considered by FRIEDMAN et al. as *projection pursuit regression* (PPR, [47]).

In this work, we propose to adapt the GAM and PPR models so that they can be used for source separation without their original assumptions that $s_j = f_j$ with each $f_j\left(l\right)$ depending on the projection of l into the real line. We instead only assume that the mixing, source and model cost functions as defined above are given, along with the parametric families $\mathcal{F}^{(j)}$ of functions and proximity kernels w_j. Even if this is a generalization of both approaches, the separation algorithm we present is strongly inspired by the original *backfitting* procedure described in [47] and further studied in [29], [46] for the estimation of GAMs. Logically, we propose the term *kernel backfitting* for this algorithm.

Intuitively, the algorithm goes as follows. For a set of source estimates $\hat{s}_j$ and for each location l_n, we compute the parameters $\hat{\Gamma}_j\left(l_n\right)$ that minimize the sources cost functions $\mathcal{L}_{s_j|\Gamma_j}$without taking the model cost function into account. This leads to a set of parameters $\hat{\Gamma}_j = \left\{z_j, \hat{R}_j\right\}$, where z_j is a noisy observation of the true latent explanatory function f_j. Then, new estimates $\hat{f}_j$ are computed by kernel-smoothing z_j through kernel local regression as in section II-A, using the model cost function (15). Finally, with this new set of parameters $\left\{\hat{f}_j, \hat{R}_j\right\}$, new sources estimates are computed by minimizing the separation cost function (16). The process then repeats using those new source estimates, until a stopping criterion is reached, such as iteration number or the difference between new and old estimates. The different steps are outlined in algorithm 1.

Apart from its similarity with the backfitting procedure, this algorithm also coincides in some cases with the

[4]In the following, each notation $\mathcal{L}_\star\left(\cdot\right)$ denotes a cost function which depends on the location l considered. For ease of notation, this dependence on l is not made explicit.

[5]For instance, we have $s_j = f_j$ when $\sigma^2 = 0$ in figure 3(a).

Algorithm 1 Kernel backfitting (KBF). General formulation.

1) **Input**:
   - Mixture data $\mathcal{D} = \{(l_n, x_n)\}_{n=1,\dots,N}$
   - Number of source J
   - Sources (12), model (15) and separation (16) cost functions
   - Kernel models $\left\{w_j, \mathcal{F}^{(j)}\right\}_{j=1,\dots,J}$
   - Stopping criterion
2) **Initialization**
   a) $\forall (j,n), \hat{s}_j(l_n) \leftarrow x_n/J$
3) **Parameters update step**
   a) $\forall j, \left\{z_j, \hat{R}_j\right\} \leftarrow \underset{\Gamma}{\text{argmin}} \sum_n \mathcal{L}_{s_j|\Gamma_j}(\hat{s}_j(l_n))$
   b) $\forall (j,n), \hat{f}_j(l_n) \leftarrow \mathcal{F}^{(j)}_{\theta_j(l_n)}(l_n)$, where $\theta_j(l_n)$ is estimated as in (15)
   c) $\forall (j,n), \hat{\Gamma}_j(l_n) \leftarrow \left\{\hat{f}_j(l_n), \hat{R}_j(l_n)\right\}$
4) **Separation step**
   a) $\{\hat{s}_1, \dots, \hat{s}_J\} \leftarrow \underset{s}{\text{argmin}} \mathcal{L}_{s|x,\hat{\Gamma}}$
5) If stopping criterion is not met, go to step 3
6) **Output**: sources and parameter estimates $\left\{\hat{s}_j, \hat{\Gamma}_j\right\}_{j=1\cdots J}$

Expectation-Maximization approach (EM [48]) undertaken for underdetermined source separation, e.g. in [17], [12], [18]. This happens when the proximity kernels w_j for the sources are uniformly one $\forall (l,l'), w_j(l,l') = 1$, leading to a global fitting of θ_j in step 3b, and when the model cost function is chosen in a probabilistically coherent way with the source and separation models. For instance, it has been argued in [49], [44], [12], [14], [13], [18] that the Itakura-Saito (IS) divergence should be chosen as the model cost function (15) if the parameters θ_j are to be used for variance modelling of Gaussian random variables. The corresponding source and separation distributions are derived consequently. However, many studies have demonstrated that the use of other model cost functions such as the Kullback-Leibler (KL) divergence may provide better performance in the same context [50], [51]. This is our motivation in using user definable cost functions for the sources, models and separation steps. As we show later, this can be important in the case of strong impulsive noise in the measurements.

Finally, the KBF algorithm is close in spirit to the Denoising Source Separation framework (DSS [52]), which is limited to overdetermined source separation. Indeed, it includes a local smoothing of the latent functions z_j in step 3b to yield the updated estimates $\hat{f}_j$. Even if this smoothing actually includes arbitrary proximity kernels as defined in section II-A, the main idea remains the same: prior knowledge is used within the separation algorithm to improve estimates through some kind of *procedural* denoising operation, which permits to model sources. In a sense, the KBF algorithm 1 may be considered as a counterpart for DSS in the case of underdetermined mixtures.

IV. Toy example : robust source separation of locally constant sources

In this section, we study the separation of synthetic signals mixed with impulsive noise and show that KAM gives good performance in this context, unlike linear methods such as GP [13]. MATLAB code for these toy examples is available at the web page dedicated to this paper[6]

A. KAM formulation

To illustrate the use of KAM for source separation, assume that the observation is composed of N real measurements $x_1, \dots, x_N$ of the mixture at locations $l_1, \dots, l_N$. They are a simple sum of J sources $s_j(l_n)$:

$$x_n = \sum_{j=1}^{J} s_j(l_n). \tag{17}$$

The first step in KAM is to model each source. We will assume for now that all samples $s_j(l_n)$ from each source s_j are independent and are Gaussian distributed with mean $f_j(l_n)$ and variance σ^2 as in figure 3(a):

$$s_j(l_n) \sim \mathcal{N}\left(f_j(l_n), \sigma^2\right). \tag{18}$$

Based on the observation of a single sample $\hat{s}_j(l_n)$, the maximum likelihood estimate $z_j(l_n)$ of the mean $f_j(l_n)$, which minimizes the source cost function (12) is the trivial

$$z_j(l_n) = \hat{s}_j(l_n), \tag{19}$$

to be used in KBF at step 3a.

The second element required for KBF is to set a kernel local parametric model to each latent mean function f_j. This is achieved by specifying a parametric family $\mathcal{F}^{(j)}$ of functions, a proximity kernel w_j and a model cost function $\mathcal{L}_j$. First, f_j is simply assumed locally constant, so that (14) collapses to $f_j(l_n) = \theta_j(l_n)$. Second, we choose a nearest neighbours proximity kernel w_j as described in section II-B. Its particular shape depends on the source and is described below. Finally, the model cost function is arbitrarily chosen as the absolute deviation:

$$\mathcal{L}_j(z_j(l_n) \mid f_j(l_n)) = |z_j(l_n) - f_j(l_n)|. \tag{20}$$

This choice is motivated by the fact that $z_j(l_n)$ is likely to be a very poor estimate of $f_j(l_n)$, because it is based on a single observation. The absolute deviation is widely known to be more robust to the presence of outliers in the data. It is readily shown [53] that minimization of the binary-weighted model cost function (20) is achieved by the median value of $\{z_j(l) \mid l \in \mathcal{I}_j(l_n)\}$, denoted:

$$\hat{f}_j(l_n) = \text{median}\{z_j(l) \mid l \in \mathcal{I}_j(l_n)\}, \tag{21}$$

to be used in KBF at step 3b.

Finally, the last step in KAM is to specify the separation cost function. Provided all latent mean functions f_j are known, the posterior distribution $p(s_1, \dots, s_J \mid x, \Gamma)$ of the

[6] www.loria.fr/~aliutkus/kam/

sources given the mixture is Gaussian. Their a posteriori mean thus minimizes the separation cost function and may hence be used during KBF at step 4a:

$$\hat{s}_j(l_n) = f_j(l_n) + \frac{x_n - \sum_{j'=1}^{J} f_{j'}(l_n)}{J}. \quad (22)$$

Using expressions (19), (21) and (22) in the corresponding steps of the KBF algorithm, separation of all sources and estimation of the latent mean functions f_j can be achieved. If all proximity kernels have limited support $k \ll N$, complexity of the KBF algorithm is $\mathcal{O}(kN)$.

B. GP formulation

The same problem can be handled using Gaussian Processes (GP) for source separation [13]. Combining (17) and (18), we get:

$$x_n = \sum_{j=1}^{J} f_j(l_n) + \epsilon_n, \quad (23)$$

where all ϵ_n are independent and identically distributed (i.i.d.) with respect to a Gaussian distribution. For reasons that will become clear soon, their common variance $J\sigma^2$ is rewritten as $2\gamma^2$, where γ^2 is called the noise *power*. Provided each source is modelled as a GP with known mean and covariance functions (see [41], [13]), their separation is readily achieved as a particular case of GP regression:

$$\left[\hat{f}_j(l_1)\dots\hat{f}_j(l_n)\right]^\top = K_j\left[\sum_{j'=1}^{J} K_{j'} + 2\gamma^2\mathbf{I}_N\right]^{-1}[x_1,\dots,x_N]^\top, \quad (24)$$

where $\cdot^\top$ denotes conjugation, K_j is a known $N\times N$ covariance matrix of $\left[\hat{f}_j(l_1)\dots\hat{f}_j(l_n)\right]^\top$ and $\mathbf{I}_N$ is the $N\times N$ identity matrix. In the GP framework, prior information about each source comes as a particular choice for the covariance function, which encodes our knowledge about the regularities of f_j and permits the building of K_j. As demonstrated for instance in [13], in the case of regularly sampled signals and stationary covariance functions, separation (24) may be achieved in $\mathcal{O}\left(N^2\log N\right)$ operations. Many techniques for underdetermined source separation can be understood as such GP regression [13].

C. Results and discussion

In order to compare the KAM and GP frameworks for source separation, we synthesize two latent explanatory functions f_1 and f_2 as realizations of two GP whose covariance functions are known. In other words, the covariance matrices K_1 and K_2 in (24) are assumed known, which is the ideal case for the GP approach. x is then built as in (23), and separation is performed with both KAM and GP. The metric considered for performance evaluation is the signal to error ratio SER_j:

$$SER_j = 10\log_{10}\frac{\|f_j\|_2}{\left\|f_j - \hat{f}_j\right\|_2},$$

which is higher for better separation. 50 independent trials of this experiment are performed and results are reported as the median and interquartile range of all SER_j.

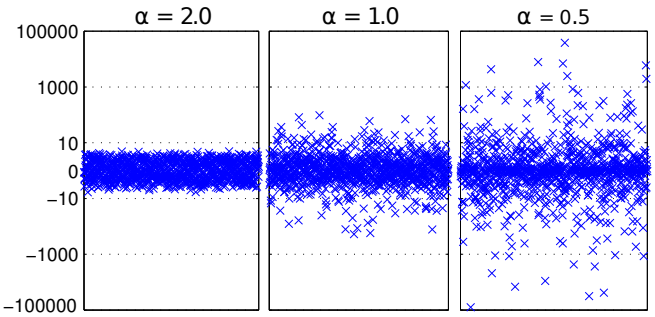

Figure 4. Independent and identically distributed symmetric and centered α-stable noise with power 1 and different α. Plots are semi-logarithmic. The case $\alpha = 2$ is Gaussian and $\alpha \to 0$ features strong outliers.

Our objective in this toy-study is to test the robustness of KAM and GP to violations of the Gaussian assumption for the additive noise ϵ_n. More precisely, we check for their performance when some outliers are present among the ϵ_n. In practice, instead of taking all ϵ_n as i.i.d. Gaussian, they are drawn from a symmetric α-stable distribution of power γ^2. The family of α-stable distributions includes Gaussian ($\alpha = 2$), Cauchy ($\alpha = 1$) and Levy ($\alpha = 1/2$) distributions as special cases. Their main characteristic is that a sum of α-stable random variables remains an α-stable random variable. Their *stability* parameter $\alpha \in [0\,2]$ controls the tail of the distribution, ($\alpha \to 0$ leads to heavy tails) and its *power* γ^2 controls its spread. In figure 4, we show independent and identically distributed samples from such symmetric α-stable distributions. They have been largely studied in the field of nonlinear signal processing because they are good models for impulsive data, yielding estimates that are robust to outliers (see [53] for a review).

For many different values of α between $\alpha = 0.5$ and $\alpha = 2$, we perform separation with both KAM and GP as described above. Of course, the assumptions underlying GP separation do not hold except for $\alpha = 2$. Still, estimation can nonetheless be performed as in (24), where $2\gamma^2$ is used instead of the variance of noise. As can be noticed, none of the KAM updates (19), (21) and (22) involve the noise variance and all can hence be used as is. Periodic kernels are chosen for the fitting of f_j, with the true periods assumed known as in figure 1. For GP, the true covariance matrices K_j are used for separation, which is a stronger prior information than the periods only.

Results are displayed in figure 5 and clearly show that while KAM provides good performance for all $\alpha \in [0.5\,;\,2]$, the scores obtained by GP rapidly drop below $\alpha = 1.6$.

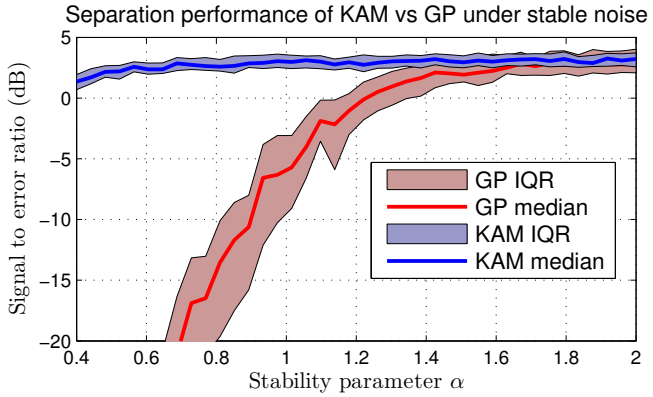

Figure 5. Performance of separation under impulsive α-stable noise of unit power as a function of α for both GP and KAM separation algorithms. 50 independent trials are considered for each α. Whereas KAM is robust to impulsive noise, the performance of GP separation is good for $\alpha \geq 1.6$ only, i.e. for non-impulsive noise. IQR stands for InterQuartile Range.

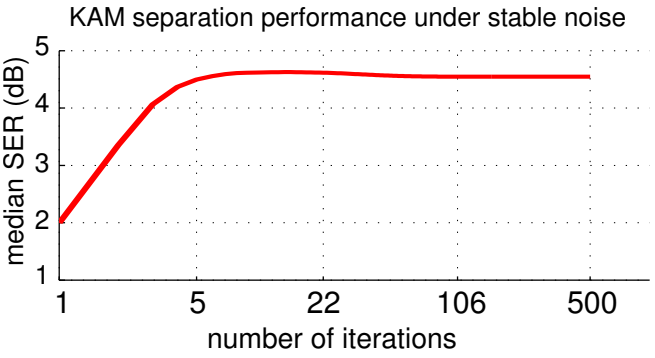

Figure 6. Median signal to error ratio for KAM separation of periodic signals from α-stable noise with $\alpha = 1$, as a function of the number of iterations, for 10 independent trials. Performance plateaus after 5 iterations.

Remarkably, even for the Gaussian case $\alpha = 2$, GP separation is not better than KAM. We can conclude that GP cannot handle outliers as well as KAM. This is an expected result, since (24) boils down to a linear combination of observations. On the contrary, separation using the KBF algorithm involves a robust estimation of f_j at step 3b, which permits excellent performance even in case of $\alpha-$stable noise. On figure 6, we show the performance of KAM as a function of the number of iterations for this example. As can be seen, performance plateaus in about 5 iterations.

Finally, we tested KAM for the separation of step-like signals from periodic oscillations under stable noise. Some illustrative results are given in figure 7. Remarkably, it is impossible to use a GP with a stationary covariance function to model such step-like signals. In KAM, the only difference from the scenario above is the use of a classical proximity kernel $\mathcal{I}_j(l) = [l-p, \dots l+p]$. This leads to a median filtering of z_j in the corresponding KBF step 3b. Separation with KAM is still of linear complexity and done in a few seconds using a standard laptop computer for $N = 5000$ observations and 10 iterations of KBF.

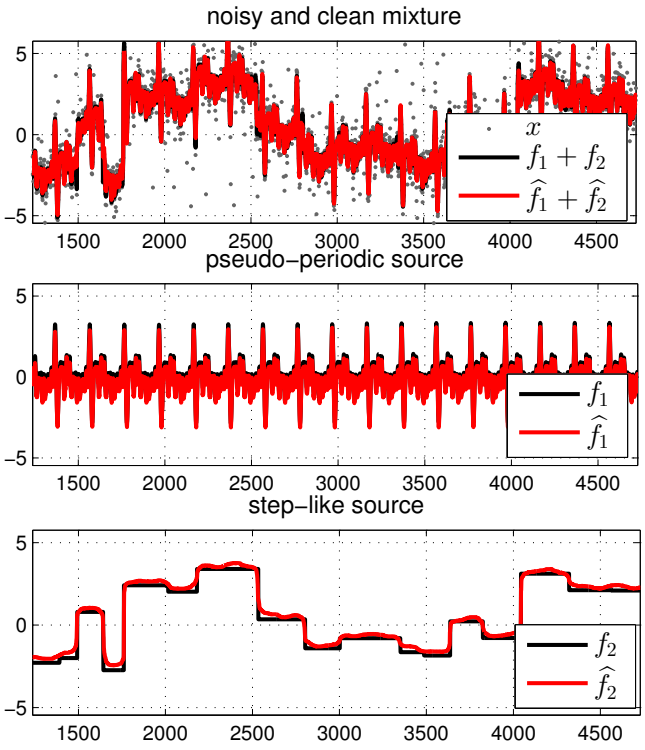

Figure 7. Example of kernel additive modelling. The noisy observed signal (top) is the sum of two sources (middle, bottom) and very adverse Cauchy noise ($\alpha = 1$). KAM permits the separation of step-like signals, for which no stationary GP model is available.

V. Application : audio source separation

In this section, we illustrate how to use KAM for a particular real-world scenario: the separation of music recordings. After some theoretical background on audio source separation, we show how to instantiate the KAM framework to devise efficient audio separation methods.

A. Separation of Gaussian processes : principles

The observed mixture consists of I audio waveforms denoted $\tilde{x}$. Each one of them is called a *channel* of the mixture. In music processing, the case $I = 2$ of stereophonic mixtures $\tilde{x}_n$ is common. The mixture $\tilde{x}$ is assumed to be the sum of J unknown signals $\{\tilde{s}_j\}_{j=1,\dots,J}$ called *sources*, that are also multichannel waveforms:

$$\tilde{x} = \sum_{j=1}^{J} \tilde{s}_j. \tag{25}$$

The Short Term Fourier Transforms (STFTs) of the J sources and of the mixture are written $\{s_j\}_{j=1\dots J}$ and x, respectively. They all are $N_\omega \times N_t \times I$ tensors, where N_ω is the number of frequency bins and N_t the number of frames. $N = N_\omega N_t$ is the total number of Time-Frequency (TF) bins. $x(\omega, t)$ and $s_j(\omega, t)$ are $I \times 1$ vectors that gather the value of the STFT of all channels (e.g. left and right) of x and s_j at TF bin (ω, t). We denote $\mathbb{L}$ as a set of all TF bins (ω, t): $\mathbb{L} = [1 \cdots N_\omega] \times [1 \cdots N_t]$.

In the monophonic case $I = 1$, it can be shown that under local stationarity assumptions [13], all TF bins $s_j(\omega, t)$ of the STFT are independent and normally distributed. In the multichannel case, a popular related model is the Local Gaussian Model [12]. It assumes that all vectors $s_j(\omega, t)$ are independent, each one of them being distributed with respect to a multivariate centered complex Gaussian distribution:

$$\forall (\omega, t),\ s_j(\omega, t) \sim \mathcal{N}_c\left(0, f_j(\omega, t) R_j(\omega)\right), \tag{26}$$

where $f_j(\omega, t) \geq 0$ is the *power spectral density* (PSD) of source j at TF bin (ω, t). It is a nonnegative scalar that corresponds to the energy of source j at TF bin (ω, t). The *spatial covariance matrix* $R_j(f)$ is a $I \times I$ positive semidefinite matrix that encodes the covariance between the different channels of s_j at frequency ω. As shown in [12], this model generalizes the popular linear instantaneous and convolutive cases [1] and permits more flexibility in the modelling of the spatial dispersion of a source. As can be seen, the source model (26) is a multichannel extension of the heteroscedastic model depicted in figure 3(b), which includes both a latent explanatory function f_j and other parameters R_j, gathered in $\Gamma_j(\omega, t) = \{f_j(\omega, t), R_j(\omega)\}$. Being the sum of J independent random Gaussian vectors $s_j(\omega, t)$, the mixture $x(\omega, t)$ is itself distributed as:

$$\forall (\omega, t),\ x(\omega, t) \sim \mathcal{N}_c\left(0, \sum_{j=1}^{J} f_j(\omega, t) R_j(\omega)\right). \tag{27}$$

If the parameters $\Gamma_j = \{f_j, R_j\}$ are known or estimated as $\hat{f}_j$ and $\hat{R}_j$, the Minimum Mean-Squared Error (MMSE) estimates $\hat{s}_j$ of the STFTs of the sources are obtained via generalized spatial Wiener filtering [10], [11], [13], [12]:

$$\hat{s}_j(\omega, t) = \hat{f}_j(\omega, t) \hat{R}_j(\omega) \left[\sum_{j'=1}^{J} \hat{f}_{j'}(\omega, t) \hat{R}_{j'}(\omega)\right]^{-1} x(\omega, t). \tag{28}$$

The waveforms $\tilde{s}_j$ of the sources in the time domain are easily recovered by inverse STFTs.

B. Locally constant models for audio sources

Setting this in the KAM methodology, we see that (26) readily provides a source cost function while (28) permits minimization of the separation cost function. We now choose a kernel local parametric model for the PSD f_j of the sources, to be used in the KBF algorithm at step 3b. We model all PSDs f_j as locally constant and use k-NN proximity kernels as presented in section II-B. In other words, for each TF bin $(\omega, t) \in \mathbb{L}$, we specify a set of k neighbours $\mathcal{I}_j(\omega, t) \in \mathbb{L}^k$, for which the PSD has a value close to $f_j(\omega, t)$:

$$\forall l \in \mathcal{I}_j(\omega, t),\ f_j(l) \approx f_j(\omega, t).$$

Some examples of such binary proximity kernels are given below in section V-C.

With the proximity kernels in hand, the only missing part for the use of KAM is the definition of the model cost function $\mathcal{L}_j$. Just like in the toy example above in section IV, we choose the absolute deviation (20), because it is known to be less sensitive to outliers in the estimates z_j, which are numerous during convergence. Indeed, $z_j(l_n)$ is computed using I observations only and is likely to be contaminated with interferences from other sources. This leads to the following cost function to be minimized at KBF step 3b:

$$\hat{f}_j(\omega, t) = \operatorname*{argmin}_f \sum_{l \in \mathcal{I}_j(\omega, t)} |z_j(l) - f|, \tag{29}$$

which is achieved by:

$$\hat{f}_j(\omega, t) = \operatorname{median}\left(z_j(l) \mid l \in \mathcal{I}_j(\omega, t)\right). \tag{30}$$

The application of the general KBF algorithm 1 to this audio setup is summarized in algorithm 2, where $\cdot^\star$ denotes conjugate transpose and $\operatorname{tr}(\cdot)$ is the trace operator. Steps 4b and 4c of this algorithm correspond to maximum likelihood estimation of z_j and $\hat{R}_j$ given $\hat{s}_j$. The interested reader is referred to [12], [18] for further details . A noticeable feature of this algorithm is that all sources can be handled in parallel during both steps 3 and 4, permitting computationally efficient implementations. On a current desktop computer, typical total computing time is about 5 times slower than real time and the computational complexity of KBF scales linearly with track length and number of iterations.

Algorithm 2 Kernel backfitting for multichannel audio source separation with locally constant spectrogram models and k-NN proximity kernels.

1) **Input**:
   - Mixture STFT $x(\omega, t)$
   - Neighbourhoods $\mathcal{I}_j(\omega, t)$ as in figure 8.
   - Number of iterations
2) **Initialization**
   - $\forall j, \hat{f}_j(\omega, t) \leftarrow x(\omega, t)^\star x(\omega, t) / IJ$
   - $R_j(\omega) \leftarrow I \times I$ identity matrix
3) Compute estimates $\hat{s}_j$ of all sources using (28)
4) For each source j:
   a) $C_j(\omega, t) \leftarrow \hat{s}_j(\omega, t) \hat{s}_j(\omega, t)^\star$
   b) $\hat{R}_j(\omega) \leftarrow \frac{I}{N_t} \sum_t \frac{C_j(\omega, t)}{\operatorname{tr}(C_j(\omega, t))}$
   c) $z_j(\omega, t) \leftarrow \frac{1}{I} \sum_t \operatorname{tr}\left(\hat{R}_j(\omega)^{-1} C_j(\omega, t)\right)$
   d) $\hat{f}_j(\omega, t) \leftarrow \operatorname{median}\{z_j(l) \mid l \in \mathcal{I}_j(\omega, t)\}$
5) For another iteration, go to step 3
6) **Output**:
   sources PSDs $\hat{f}_j$ and spatial covariance matrices $\hat{R}_j(\omega)$ to use for filtering (28).

C. Examples of kernels for audio sources

Many methods for audio source separation can be understood as instances of the framework presented above, including the many variants of REPET [31], [32], [33], [54], [34] or the median filtering approach presented in [30]. From

the point of view of KAM, those methods simply correspond to different choices for the proximity kernels of the sources.

As highlighted in [32], most of those studies rely on ad-hoc filtering approaches and are suboptimal in light of the developments above. In particular, when several local source models are provided as in [30], the estimation is performed independently for each source and no special care is taken in correctly modelling the observation as the *mixture* of the sources. As such, these techniques can be understood as performing only one iteration of the kernel backfitting procedure described in algorithm 2.

In this section, we illustrate the capacity of KAM to combine completely different approaches to source separation which use different assumptions. To this end, we present four families of proximity kernels to use with algorithm 2.

1) Models for percussive and harmonic sounds: In musical signals, percussive elements are known to be self-similar along the frequency axis, while harmonic steady sounds are self-similar along time [30]. This prior knowledge can be exploited in the KAM framework by choosing k-NN proximity kernels that are either vertical or horizontal, as depicted in figure 8(a) and 8(b) respectively. Using them in algorithm 2 leads to a generalization of the procedure presented in [30] that allows for multichannel mixtures.

2) Models for repetitive patterns: The musical accompaniment of a song may often be considered as locally repetitive. For instance, it may contain drum loops or guitar riffs. This has already been exploited for audio source separation in the REPET approach [31], [32] and is reminiscent of pioneering work by CLEVELAND et. al [55] on the separation of seasonal and trend components in time series. Here, we show that REPET fits well within the KAM framework and can be extended to account for superpositions of different repetitive patterns at different time scales.

Formally, the PSD f_j of a repeating source j is assumed to be locally periodic along time with period T_j, which means that $f_j(\omega, t)$ ought to be similar to $f_j(\omega, t + pT_j)$ with $p = -P, \ldots, P$. Following the discussion in section II-B3, this can be accounted for by choosing $\mathcal{I}_j(\omega, t) = \{(\omega, t + pT_j)\}$, as depicted in figure 8(c) for $P = 2$.

Then, the repeating part of a song can be modelled as the sum of J_a such spectrally pseudo-periodic signals. This formulation encompasses the REPET model discussed in [31], [32], [33] that is limited to 1 repeating pattern only.

For the purpose of estimating the periods T_j of all repeating sources, we use a peak detection of the average autocorrelation for all frequency bands of the spectrogram of the mixture. More sophisticated approaches may be considered to allow for non-integer periods.

3) Weak models for natural sounds: When devising models for the PSD of a voice signal, we are faced with the extraordinary diversity of sounds it may produce. In the past, many studies exploited the fact that sung melodies are often composed of harmonic parts obeying the classical source-filter model for phonation, including the renowned IMM model [19], [20]. Even if it often obtains good performance, this approach has issues with the separation of consonants and breathy voices, that do not fit well the harmonic model.

In this study, *natural sounds* such as the human voice are simply assumed to have smooth variations in their PSD, along time or along frequency, e.g. during voiced or voiceless parts, respectively. Since this assumption is rather loose and is valid for a large variety of signals, we call it a *weak* model for natural sounds. Formally, such a model considers that $f_j(l)$ and $f_j(l')$ are close whenever l and l' are close either along time or frequency. This is achieved by choosing the cross-like kernel depicted in figure 8(d).

4) NMF model within KAM: Even if the KAM approach encompasses a large number of recent methods for source separation [30], [31], [32], [33], [34], it can also be used to combine such approaches with a more classical NMF model. Some audio sources are indeed well explained as the activation over time of fixed and global patterns. To this purpose, the PSD f_j of source j may be modelled as:

$$f_j(\omega, t) = \sum_{k=1}^{K} W_j(\omega, k) H_j(k, t), \tag{31}$$

where W_j and H_j are parameters to be fitted globally. This is readily achieved in the KAM framework by setting $w_j = 1$. During step 4d of the KBF algorithm 2, median filtering is then simply replaced for such a source by a global fitting of z_j by the NMF model (31) through standard procedures. The model cost function $\mathcal{L}_j$ to use may be any divergence seen fit, such as IS or KL. Remarkably, if all sources are modelled this way and if the IS divergence is chosen, algorithm 2 coincides with the EM procedure described, e.g. in [18], [12].

D. Voice extraction performance

In our experiments, we processed 50 full-length stereo tracks from the ccMixter[7] database, featuring many different musical genres. For each track, the accompaniment was modelled as a sum of $J_a = 6$ repeating patterns along with a 2-seconds steady harmonic source. Vocals were modelled using a cross-like kernel of height 15Hz and width 20ms. Framelength is set to 90ms , with 80% overlap.

Kernel backfitting as described in algorithm 2 was applied for 6 iterations. A MATLAB implementation of KAM may be found in the companion webpage of this paper[8], along with the audio database and separation examples. We also performed vocal separation on these 50 full-length tracks with 3 techniques from the state of the art: IMM [19], RPCA [56] and REPETsim [54], [34]. Since RPCA and REPETsim do not handle stereo signals explicitly, they were applied on left and right channels independently. Once the tracks have been separated, they are split into 30s excerpts and performance is evaluated on the 350 resulting excerpts. The metric considered is the Source to Distortion Ratio (SDR) computed with the BSSeval toolkit [57], which is

[7] www.ccmixter.org

[8] www.loria.fr/~aliutkus/kam/

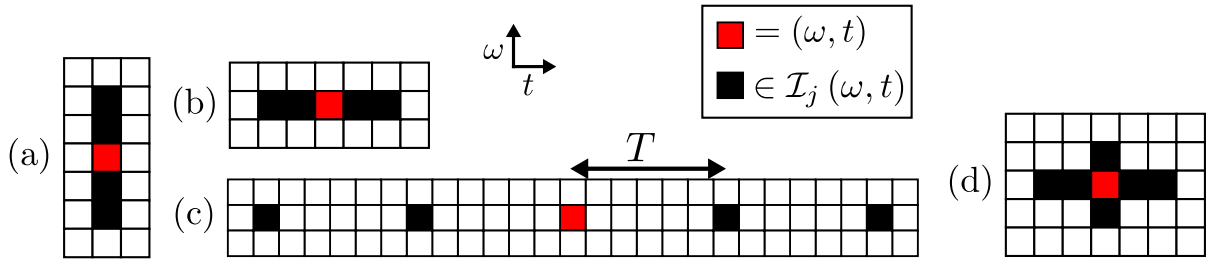

Figure 8. Some examples of k-Nearest Neigbours proximity kernels for modelling audio sources. (a) *vertical*, for percussive elements, (b) *horizontal*, for stable harmonic elements, (c) *periodic*, for repetitive elements, (d) *cross-like*, for smoothly varying power spectral densities such as vocals.

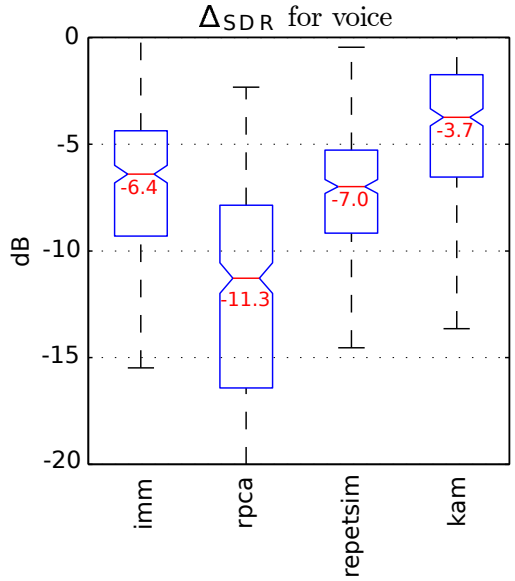

Figure 9. Δ_{SDR} scores for the separation of vocals over 50 full-length tracks, including those of the proposed KAM setup. Higher is better.

given in dB. In order to normalize separation results along the different tracks, Δ_{SDR} is given instead of SDR and indicates the loss in performance as compared to the soft-mask oracle [58]. In other words, $\Delta_{SDR} = 0$dB indicates that separation is as good as oracle Wiener filtering and the higher Δ_{SDR} is, the better the separation. Boxplots of the results are displayed on figure 9. As can be noticed, performance of the proposed KAM setup for vocal separation beats other competing methods by approximately 3dB. A multiple comparison test using a non-parametric Kruskal-Wallis analysis of variance, at the 5% confidence interval level, shows that KAM is significantly better in terms of Δ_{SDR} than all other methods. In any case, these scores only hold for the choice of proximity kernels we made in this voice/music separation task. Indeed, KAM may be used in many other settings or yield improved performance with more adequate proximity kernels and careful tuning.

VI. Conclusion

In this paper, we have proposed a new framework for source separation, where each source is modelled both *locally* and *parametrically*. Each source taken at some location is assumed to be correctly predicted using its values at other locations *nearby*. In this case, estimation can be performed using local regression.

However, not all sources are well understood by stating that neighbouring locations necessarily induce close values. This would only be true for smooth signals, which are not a good fit to the data in many cases. Instead, there may be a more sophisticated way to decide whether two locations give similar values. More generally, we introduced *proximity kernels*, which give the proximity of two points from the perspective of a source model. There are several ways of building such kernels and many methods from the literature come as special cases of this framework.

Separation of a mixture in this context can be performed using a variant of the *backfitting* algorithm, termed *kernel backfitting*, for which topological distance is replaced by source-specific proximity kernels. We showed how this Kernel Additive Modelling approach permits separation of sources that are defined through different proximity kernels.

A first feature of this method is that it is flexible enough to account for the dynamics of many kinds of signals and we indeed showed that it comes as a unifying framework for many state-of-the-art methods for source separation. Second, it yields an easy and principled way to create and combine kernel models in order to build sophisticated mixture models. Finally, the corresponding algorithms are very easy to implement in some cases and provide good performance, as demonstrated in our evaluations.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] P. Comon and C. Jutten, eds., *Handbook of Blind Source Separation: Independent Component Analysis and Blind Deconvolution*. Academic Press, 2010.

[2] A. Hyvärinen, J. Karhunen, and E. Oja, eds., *Independent Component Analysis*. Wiley and Sons, 2001.

[3] A. Cichocki, L. Zhang, and T. Rutkowski, "Blind separation and filtering using state space models," in *Proc. IEEE Int. Symp. Circuits and Systems ISCAS '99*, vol. 5, pp. 78–81, 1999.

[4] L.-Q. Zhang, A. Cichocki, and S. Amari, "Kalman filter and state-space approach to blind deconvolution," in *Proc. IEEE Signal Processing Society Workshop Neural Networks for Signal Processing X*, vol. 1, pp. 425–434, 2000.

[5] H. Valpola, A. Honkela, and J. Karhunen, "An ensemble learning approach to nonlinear dynamic blind source separation using state-space models," in *Proc. Int. Joint Conf. Neural Networks IJCNN '02*, vol. 1, pp. 460–465, 2002.

[6] L. Barrington, A. Chan, and G. Lanckriet, "Modeling music as a dynamic texture," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, pp. 602–612, Mar. 2010.

[7] R. Badeau, "Gaussian modeling of mixtures of non-stationary signals in the time-frequency domain (HR-NMF)," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, (New Paltz, NY, USA), pp. 253–256, Oct. 2011.

[8] E. Schmidt, R. Migneco, J. Scott, and Y. Kim, "Modeling instrument tones as dynamic textures," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, (New Paltz, NY, USA), pp. 253–256, Oct. 2011.

[9] C. Févotte, J. L. Roux, and J. R. Hershey, "Non-negative dynamical system with application to speech and audio," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Vancouver, Canada), May 2013.

[10] L. Benaroya, F. Bimbot, and R. Gribonval, "Audio source separation with a single sensor," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, pp. 191–199, Jan. 2006.

[11] A. Cemgil, P. Peeling, O. Dikmen, and S. Godsill, "Prior structures for Time-Frequency energy distributions," in *Proc. of the 2007 IEEE Workshop on. App. of Signal Proc. to Audio and Acoust. (WASPAA'07)*, (NY, USA), pp. 151–154, Oct. 2007.

[12] Q. K. N. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 1830–1840, July 2010.

[13] A. Liutkus, R. Badeau, and G. Richard, "Gaussian processes for underdetermined source separation," *IEEE Transactions on Signal Processing*, vol. 59, pp. 3155 –3167, July 2011.

[14] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, vol. 23, pp. 2421–2456, Sep. 2011.

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

[16] O. Dikmen and A. Cemgil, "Unsupervised single-channel source separation using Bayesian NMF," in *Proc. of the 2009 IEEE Workshop on. App. of Signal Proc. to Audio and Acoust. (WASPAA'09)*, (NY, USA), pp. 93–96, Oct. 2009.

[17] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 18, pp. 550–563, Mar. 2010.

[18] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[19] J. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, pp. 1180 –1191, oct. 2011.

[20] J. Durrieu and J. Thiran, "Musical audio source separation based on user-selected F0 track," in *Proc. of International Conference on Latent Variable Analysis and Signal Separation*, (Tel-Aviv, Israel), March 12-15 2012.

[21] A. Bregman, *Auditory Scene Analysis, The perceptual Organization of Sound*. MIT Press, 1994.

[22] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE trans. on Audio, Speech and Language Proc. (TASLP)*, vol. 18(3), pp. 528–537, Mar. 2010.

[23] M. N. Schmidt and H. Laurberg, "Non-negative matrix factorization with Gaussian process priors," *Computational Intelligence and Neuroscience*, vol. 2008, ID 361705.

[24] O. Dikmen and A. T. Cemgil, "Gamma markov random fields for audio source modelling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 589–601, Mar. 2010.

[25] G. Watson, "Smooth regression analysis," *Sankhya Ser.*, vol. A 26, pp. 359–372, 1964.

[26] W. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American Statistical Association*, vol. 74, pp. 829–836, 1979.

[27] C. Stone, "Consistent nonparametric regression," *The Annals of Statistics*, vol. 5, pp. 595–620, 1977.

[28] W. Cleveland and S. Devlin, "Locally weighted regression: An approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, pp. 596–610, 1988.

[29] T. Hastie and R. Tibshirani, "Generalized additive models," *Statistical Science*, vol. 1, pp. 297–310, 1986.

[30] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, (Graz, Austria), Sept. 2010.

[31] Z. Rafii and B. Pardo, "A simple music/voice separation method based on the extraction of the repeating musical structure," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 221 –224, may 2011.

[32] A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard, "Adaptive filtering for music/voice separation exploiting the repeating musical structure," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 53–56, mars 2012.

[33] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (REPET): A simple method for music/voice separation," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 21, no. 1, pp. 71–82, 2013.

[34] Z. Rafii and B. Pardo, "Music/voice separation using the similarity matrix," in *ISMIR* (F. Gouyon, P. Herrera, L. G. Martins, and M. Müller, eds.), pp. 583–588, FEUP Edições, 2012.

[35] D. Ruppert and M. P. Wand, "Multivariate locally weighted least squares regression," *The Annals of Statistics*, vol. 22, pp. 1346–1370, 1994.

[36] E. Nadaraya, "On estimating regression," *Theor. Probab. Appl.*, vol. 9, pp. 141–142, 1964.

[37] L. Yang and R. Tschernig, "Multivariate bandwidth selection for local linear regression," *Journal Of The Royal Statistical Society Series B*, vol. 61, no. 4, pp. 793–815, 1999.

[38] I. Gijbels, R. Lambert, and P. Qiu, "Edge-preserving image denoising and estimation of discontinuous surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1075–1087, 2006.

[39] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, pp. 349–366, 2007.

[40] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, (CVPR 2005). IEEE Conference on*, vol. 2, pp. 60–65, 2005.

[41] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[42] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[43] J. Quiñonero-Candela, C. E. Rasmussen, and R. Herbrich, "A unifying view of sparse approximate Gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, Dec. 2005.

[44] D. FitzGerald, M. Cranitch, and E. Coyle, "On the use of the beta divergence for musical source separation," in *Proc. of Irish Sig. and Systems Conf. (ISSC'08)*, 2008.

[45] D. Fitzgerald and R. Jaiswal, "On the use of masking filters in sound source separation," in *Proc. of 15th International Conference on Digital Audio Effects, (DAFX12)*, 2012.

[46] T. J. Hastie and R. J. Tibshirani, *Generalized additive models*. London: Chapman & Hall, 1990.

[47] J. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, pp. 817–823, 1981.

[48] A. Dempster, N. Laird, and B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[49] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis," *Neural Computation*, vol. 21, pp. 793–830, Mar. 2009.

[50] M. Shashanka, B. Raj, and P. Smaragdis, "Probabilistic latent variable models as non-negative factorizations," *Computational Intelligence and Neuroscience special issue on Advances in Non-negative Matrix and Tensor Factorization*, vol. May, pp. 12–20, 2008.

[51] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *ICA '04: Proc. of the 8th Int. Conf. on Independent Component Analysis and Signal Separation*, 2004.
[52] J. Särelä, H. Valpola, and M. Jordan, "Denoising source separation," *Journal of Machine Learning Research*, vol. 6, no. 3, 2005.
[53] G. Arce, *Nonlinear signal processing: a statistical approach*. John Wiley & Sons, 2005.
[54] D. FitzGerald, "Vocal separation using nearest neighbours and median filtering," in *23rd IET Irish Signals and Systems Conference (ISSC2012)*, (NUI Maynooth, Ireland), June 2012.
[55] R. Cleveland, W. Cleveland, J. Mcrae, and I. Terpenning, "STL: A Seasonal-Trend Decomposition Procedure Based on Loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.
[56] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.
[57] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1462 –1469, July 2006.
[58] E. Vincent, R. Gribonval, and M. Plumbley, "Oracle estimators for the benchmarking of source separation algorithms," *Signal Processing*, vol. 87, pp. 1933 – 1950, Aug. 2007.

Antoine Liutkus received the State Engineering degree from Telecom ParisTech, France, in 2005, and the M.Sc. degree in acoustics, computer science and signal processing applied to music (ATIAM) from the Université Pierre et Marie Curie (Paris VI), Paris, in 2005. He worked as a research engineer on source separation at Audionamix from 2007 to 2010 and obtained his PhD in electrical engineering at Telecom ParisTech in 2012. He is currently researcher at Inria Nancy Grand Est in the speech processing team. His research interests include audio source separation and machine learning.

Derry Fitzgerald (PhD, M.A. B.Eng.) is a senior Researcher in the NIMBUS Centre at Cork Institute of Technology. He was a Stokes Lecturer in Sound Source Separation algorithms at the Audio Research Group in DIT from 2008-2013. Previous to this he worked as a post-doctoral researcher in the Dept. of Electronic Engineering at Cork Institute of Technology, having previously completed a Ph.D. and an M.A. at Dublin Institute of Technology. He has also worked as a Chemical Engineer in the pharmaceutical industry for some years. In the field of music and audio, he has also worked as a sound engineer and has written scores for theatre. He has recently utilised his sound source separation technologies to create the first ever officially released stereo mixes of several songs for the Beach Boys, including "Good Vibrations" and "I get around". His research interests are in the areas of sound source separation, tensor factorizations, and music information retrieval systems.

Zafar Rafii is a Ph.D. candidate in the department of Electrical Engineering and Computer Science at Northwestern University. He received a Master of Science in Electrical Engineering, Computer Science and Telecommunications from Ecole Nationale Sup´erieure de l'Electronique et de ses Applications (ENSEA) in France and a Master of Science in Electrical Engineering from Illinois Institute of Technology (IIT) in the US. He also worked as a research engineer at Audionamix in France and as a research intern at Gracenote in the US. His research interests are centered on audio analysis, at the intersection of signal processing, machine learning, and cognitive science.

Bryan Pardo, head of the Northwestern University Interactive Audio Lab, is an associate professor in the Northwestern University Department of Electrical Engineering and Computer Science with additional appointments in the Music Theory and Cognition Department, the Segal Design Institute and is an associate of the Northwestern Cognitive Science program. Prof. Pardo received a M. Mus. in Jazz Studies in 2001 and a Ph.D. in Computer Science in 2005, both from the University of Michigan. He has authored over 70 peer-reviewed publications. He is a past associate editor for IEEE Transactions on Audio Speech and Language Processing. He has developed speech analysis software for the Speech and Hearing department of the Ohio State University, statistical software for SPSS and worked as a machine learning researcher for General Dynamics. While finishing his doctorate, he taught in the Music Department of Madonna University. When he's not programming, writing or teaching, he performs throughout the United States on saxophone and clarinet at venues such as Albion College, the Chicago Cultural Center, the Detroit Concert of Colors, Bloomington Indiana's Lotus Festival and Tucson's Rialto Theatre.

Laurent Daudet (M'04–SM'10) studied at the Ecole Normale Supérieure in Paris, where he graduated in statistical and non-linear physics. In 2000, he received a PhD in mathematical modeling from the Université de Provence, Marseille, France. After a Marie Curie post-doctoral fellowship at the C4DM, Queen Mary University of London, UK, he worked as associate professor at UPMC (Paris 6 University) in the Musical Acoustics Lab. He is now Professor at Paris Diderot University – Paris 7, with research at the Langevin Institute for Waves and Images, where he currently holds a joint position with the Institut Universitaire de France. Laurent Daudet is author or co-author of over 150 publications (journal papers or conference proceedings) on various aspects of acoustics and audio signal processing, in particular using sparse representations.