An Online Kernel Change Detection Algorithm

Frédéric Desobry, Manuel Davy, and Christian Doncarli

***Abstract*—A number of abrupt change detection methods have been proposed in the past, among which are efficient model-based techniques such as the Generalized Likelihood Ratio (GLR) test. We consider the case where no accurate nor tractable model can be found, using a model-free approach, called *Kernel change detection* (KCD). KCD compares two sets of descriptors extracted online from the signal at each time instant: The immediate past set and the immediate future set. Based on the soft margin single-class Support Vector Machine (SVM), we build a dissimilarity measure in feature space between those sets, without estimating densities as an intermediary step. This dissimilarity measure is shown to be asymptotically equivalent to the Fisher ratio in the Gaussian case. Implementation issues are addressed; in particular, the dissimilarity measure can be computed online in input space. Simulation results on both synthetic signals and real music signals show the efficiency of KCD.**

***Index Terms*—Abrupt change detection, kernel method, music segmentation, online, single-class SVM.**

I. Introduction

DETECTING abrupt changes in signals or systems is a longstanding problem, and various approaches have been proposed in a number of papers. In particular, likelihood ratio based approaches, such as the Generalized Likelihood Ratio (GLR) test [1] or the marginal likelihood ratio test [2], are quite efficient whenever an accurate and tractable signal model exists and can be implemented. Online versions based on statistical filtering have good performance as well [1], [3]. Other model-based approaches perform efficient off-line Bayesian segmentation [4], [5]. Besides model-based techniques, a number of general and *ad hoc* model-free methods have been designed to detect abrupt changes in signals. Typical examples are time-frequency approaches [6], wavelet approaches [7], [8], and cepstral coefficients approaches [9].

Manuscript received November 26, 2003; revised August 3, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dominic K. C. Ho.

F. Desobry is with the Institut de Recherche en Communication et Cybernétique de Nantes, IRCCyN, UMR CNRS 6597, Nantes, France, and also with the Signal Processing Lab., Department of Engineering, University of Cambridge, CB2 1PZ Cambridge, U.K. (e-mail: fd238@eng.cam.ac.uk).

M. Davy is with the Laboratoire d'Automatique, Génie Informatique et Signal de Lille, LAGIS/CNRS, Ecole Centrale de Lille, Cite Scientifique, BP 48, 59651 Villeneuve d'Ascq Cedex, UMR CNRS 6597, France (e-mail: Manuel.Davy@ec-lille.fr).

C. Doncarli is with the Institut de Recherche en Communication et Cybernétique de Nantes, IRCCyN, UMR CNRS 6597, Nantes, France, and also with the IRCCyN/ECN, Ecole Centrale de Nantes, Nantes Cedex 03, France (e-mail: Christian.Doncarli@irccyn.ec-nantes.fr).

Digital Object Identifier 10.1109/TSP.2005.851098

In this paper, we present a general, model-free framework for on-line abrupt change detection.[1] Similar to other model-free techniques, the detection of abrupt changes is based on descriptors extracted from a signal of interest. The main subject of this paper is not about feature extraction; it is about the abrupt change detection algorithm that uses these descriptors. Our algorithm is quite general in the sense that it applies to one-dimensional (1-D) signals (e.g., music signals, speech signals, vibration signals) as well as to large dimensional signals (e.g., video, monitoring with multiple sensors). More precisely, the principle of this technique can be explained as follows. Assume that descriptors $x_t, t = 1, 2, \ldots$ in a space $\mathcal{X}$ are extracted on-line from a (possibly large-dimensional) signal $y_\tau, \tau = 1, 2, \ldots$, using a function $q(\cdot)$. (The time indexes for x and y do not necessarily coincide, as we may not extract descriptors at each sample time τ.) The problem consists now of detecting abrupt changes in the time series $x_t, t = 1, 2, \ldots$. Note that with typical descriptor extraction techniques, the dimension of x_t may be large; see, for example, [6] and [11].

A. General Framework for Nonparametric Online Abrupt Change Detection

The time series of descriptors may be used in many ways to design an abrupt change detector. Some techniques compute a distance measure between two successive descriptors in order to build a stationarity index (see, e.g., [6]). Other techniques implement the GLR test via Gaussian mixture modeling of the descriptors distribution (see, e.g., [9]). The latter technique, however, can hardly deal with large-dimensional inputs because the number of model parameters to be estimated increases quickly with the dimension (problem know as the *curse of dimensionality* [12]). Most of these techniques are special instances of the following generic framework: Consider time t and two descriptor subsets (the *immediate past* subset $\boldsymbol{x}_{t,1} = \{x_i\}_{i=t-m_1,\ldots,t-1}$ and the *immediate future* subset $\boldsymbol{x}_{t,2} = \{x_i\}_{i=t,\ldots,t+m_2-1}$), as depicted in Fig. 1. We can now state the on-line abrupt change detection problem as follows. Let t be some time instant, and assume that the samples in $\boldsymbol{x}_{t,1}$ (resp. in $\boldsymbol{x}_{t,2}$) are sampled independent and identically distributed (i.i.d.) according to some probability density function (pdf) p_1 (resp. p_2). Then, one of the two hypotheses holds:

$$\begin{cases} H_0 : p_2 = p_1, & \text{(No abrupt change occurs)} \\ H_1 : p_2 \neq p_1, & \text{(An abrupt change occurs).} \end{cases} \tag{1}$$

[1]We use the word 'on-line' with the same meaning as in [1] and [10] to indicate that we address the sequential framework. Real-time applications and implementation are not directly addressed in this paper.

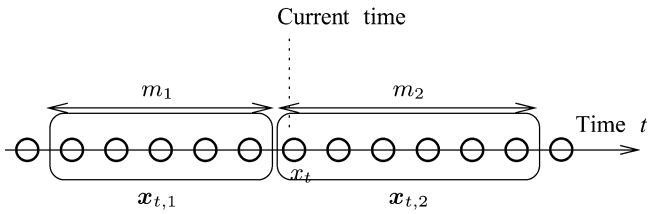

Fig. 1. General abrupt change detection framework based on the time series of descriptors $x_t, t = 1, 2, \ldots$, represented by circles.

This test cannot be applied in practice, however, since the pdfs p_1 and p_2 are not known. The standard practical approach uses some dissimilarity measure between p_1 and p_2, estimated from the sole knowledge of the sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$. Letting $\mathcal{D}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2})$ be such a dissimilarity measure, the previous problem can be written as follows:

$$\begin{cases} H_0 : \mathcal{D}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) \leq \eta, & \text{(No abrupt change occurs)} \\ H_1 : \mathcal{D}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) > \eta, & \text{(An abrupt change occurs)} \end{cases} \quad (2)$$

where η is a threshold that tunes the sensibility/robustness tradeoff, as in every detection framework. The detection performance is thus closely related to the dissimilarity measure $\mathcal{D}(\cdot, \cdot)$, as well as to the pdf estimation technique implemented. Possible choices are the following:

- Methods based on inferred descriptors distribution: The pdf of descriptors in $\boldsymbol{x}_{t,i} (i = 1,2)$ denoted $p(x \mid \theta_i, i) (i = 1,2)$ are supposed to have a given shape such as Gaussian, with an unknown parameter set denoted $\theta_i (i = 1,2)$. Parameter estimates, which are denoted $\hat{\theta}_i (i = 1,2)$, are computed on both sets, and the resulting pdfs $p(x \mid \hat{\theta}_i, i) (i = 1,2)$ are compared via a likelihood ratio [1] or using a densities dissimilarity measure, such as the Kullback–Leibler (KL) divergence [9]. This approach includes algorithms where statistics such as empirical means and covariances are estimated from descriptors and compared using a dissimilarity measure. This approach is not adapted to data with large dimension, due to the curse of dimensionality.
- Methods based on descriptors distribution, jointly with prior distributions for θ_1 and θ_2: These methods implement Bayes decision theory [12], [13] (Bayesian approach).
- Methods based on empirical descriptors density estimation: A density estimation algorithm is implemented (a typical example is the Parzen window estimator), and estimated densities $\hat{p}(x \mid 1), \hat{p}(x \mid 2)$ are compared using a dissimilarity measure. Examples can be found in [14] in the context of Independent Component Analysis (ICA). This approach is not adapted to large dimensional data, due to the difficulty to estimate accurately densities in such cases.
- Methods aimed at comparing the sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ without the intermediate density estimation step: This family of methods is the main subject of this paper.

B. Machine Learning Approach

The overall framework described above shows that abrupt change detection can be seen as a Machine Learning problem: Statistical behavior of the sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ are learned and compared. We propose a new approach inspired by recent Machine Learning theory [15]–[17], which is referred to as Kernel Change Detection (KCD) and based on the following remark: Whenever no abrupt change occurs, the location of the samples in $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ in $\mathcal{X}$ is approximately the same. On the other hand, when an abrupt change occurs, it happens that samples in $\boldsymbol{x}_{t,1}$ are located in different parts of the space than the samples in $\boldsymbol{x}_{t,2}$ (we show that a well-chosen descriptor extraction technique can yield such a situation). Following Vapnik's principle,[2] we argue that in such situations, it is more convenient to derive a dissimilarity measure that compares estimated *density supports* rather than estimated pdfs, where the support of a density $p(x)$ is roughly defined as the part of the space where $p(x)$ is "large". In order to implement this idea, it is necessary to derive a robust, accurate density support estimator together with a dissimilarity measure for regions comparison. Note that this dissimilarity measure needs be valid for regions with a complex, winding, and possibly nonconnected shape of large dimension.

C. Paper Organization

In Section II, we describe a recent density support estimation technique, namely, the ν-Support Vector (SV) approach to single-class problems [18]. In particular, we recall that this technique is adapted to large dimensional data and that it is robust to outliers. In Section III, we present our abrupt change detection algorithm, built on the single-class ν-Support Vector Machines (SVMs). This algorithm is discussed in Section IV. Comparisons with other techniques are presented. Section V is devoted to simulations on synthetic and real data. Finally, some conclusions and future research directions are proposed in Section VI.

II. ν-SV Approach to Single-Class Classification Problems

In this section, we briefly recall the elements of the ν-SV approach to single-class classification problems that are relevant to KCD. In the following, descriptors are referred to as vectors or descriptors wherever relevant. Let $\boldsymbol{x} = \{x_1, \ldots, x_m\}$ be a set of $m > 0$ so-called *training vectors* in $\mathcal{X}$. Although, practically, $\mathcal{X}$ is often an Euclidean space isomorphic to $\mathbb{R}^d$ with d finite, no stronger assumption than it being a set is needed. The set $\boldsymbol{x}$ is called the *training set*; $\mathcal{X}$ is the *input space*. We make the assumption that for any $i = 1, 2, \ldots, m$, the training vector x_i is distributed according to some unknown pdf $p(\cdot)$, independently of x_j (for $j = 1, \ldots, m$ with $j \neq i$).

The aim of single-class classification (which is also referred to as *novelty detection*) is the estimation of a region $\mathcal{R}^{\mathcal{X}}_{\boldsymbol{x}}$ in $\mathcal{X}$,

[2]Vapnik's principle is the following [15]: *If you possess a restricted amount of information for solving some problem, try to solve the problem directly, and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem.*

from the sole knowledge of the training set $\boldsymbol{x}$, such that vectors drawn according to $p(\cdot)$ are likely to fall in $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{X}}$ and such that vectors that are not in $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{X}}$ are not likely to be distributed according to $p(\cdot)$. Here, we adopt the equivalent representation of $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{X}}$ given by the real-valued decision function $f_{\boldsymbol{x}}(\cdot)$ such that

$$f_{\boldsymbol{x}}(\cdot) \geq 0 \text{ on } \mathcal{R}_{\boldsymbol{x}}^{\mathcal{X}} \quad \text{and} \quad f_{\boldsymbol{x}}(\cdot) < 0 \text{ elsewhere in } \mathcal{X}.$$

The estimation of $f_{\boldsymbol{x}}(\cdot)$ is realized via risk minimization. We define *errors* as vectors that are not in $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{X}}$, whereas they are actually distributed according to $p(\cdot)$ and the *loss function* $c(\cdot,\cdot)$ that defines the cost we assign to errors. Standard choices are the 0–1 loss function [12] or the *hinge loss* [16], which leads to ν-SV algorithms, as explained below. The *empirical risk* $R_{\text{emp}}(f_{\boldsymbol{x}})$ is defined as $R_{\text{emp}}(f_{\boldsymbol{x}}) = (1/m)\sum_{i=1}^{m} c(x_i, f_{\boldsymbol{x}}(x_i))$. Classically, estimating $f_{\boldsymbol{x}}(\cdot)$ by minimizing $R_{\text{emp}}(f_{\boldsymbol{x}})$ leads to overtraining, which can be avoided by instead minimizing the *regularized risk* $R_{\text{reg}}(f_{\boldsymbol{x}}) = R_{\text{emp}}(f_{\boldsymbol{x}}) + \lambda\|f_{\boldsymbol{x}}\|$, where $\|f_{\boldsymbol{x}}\|$ is some measure of the complexity of $f_{\boldsymbol{x}}$, and λ tunes the amount of regularization. In the SV approach, this is achieved by restricting $f_{\boldsymbol{x}}(\cdot)$ to elements of a class of simple functions with minimal complexity. This is further developed in the next subsection, which is dedicated to SV single-class classification.

A. SV Single-Class Classification

In order to present the SV approach, we introduce the so-called *feature space* $\mathcal{H}$. Let $\phi : \mathcal{X} \to \mathcal{H}$ be a mapping defined over the input space $\mathcal{X}$ and taking values in feature space $\mathcal{H}$. Let $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ be a dot product defined in $\mathcal{H}$. We define the kernel $k(\cdot,\cdot)$ over $\mathcal{X}\times\mathcal{X}$ by

$$\forall (x_i, x_j) \in \mathcal{X}\times\mathcal{X}, \quad k(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle_{\mathcal{H}}. \tag{3}$$

Without loss of generality,[3] we assume that $k(\cdot,\cdot)$ is normalized such that for any x in $\mathcal{X}$, $k(x,x) = 1$. Using the notation $\mathbf{x} = \phi(x)$ for any x in $\mathcal{X}$, we have $\|\mathbf{x}\|_{\mathcal{H}} = 1$, where the norm in $\mathcal{H}$ is induced by the dot product, i.e., $\|\mathbf{x}\|_{\mathcal{H}}^2 = \langle \mathbf{x}, \mathbf{x}\rangle_{\mathcal{H}} = k(x,x)$. In other words, the mapped input space $\phi(\mathcal{X})$ is a subset of the hypersphere $\mathcal{S}$ with radius one centered at the origin of $\mathcal{H}$, which is denoted $\mathbf{O}$. Provided $k(\cdot,\cdot)$ is always positive, $\phi(\mathcal{X})$ is a subset of the positive orthant of that hypersphere. The training vectors are mapped in $\mathcal{H}$ and lie in $\mathcal{S}$, as depicted in Fig. 2. The SV approach to single-class classification consists of separating the training vectors in $\mathcal{H}$ from the center of the hypersphere $\mathcal{S}$ with a hyperplane $\mathcal{W}$; see Fig. 2. Any hyperplane in $\mathcal{H}$ can be written as a set[4] $\{\mathbf{x}\in\mathcal{H} \,|\, \langle \mathbf{w}, \mathbf{x}\rangle_{\mathcal{H}} - \rho = 0\}$ with parameters $\mathbf{w}\in\mathcal{H}$ and $\rho \geq 0$; thus, $\mathcal{W}$ verifies

$$\langle \mathbf{w}, \mathbf{x}\rangle_{\mathcal{H}} - \rho \geq 0 \text{ for } most \text{ mapped training vectors and}$$
$$\langle \mathbf{w}, \mathbf{x}\rangle_{\mathcal{H}} - \rho < 0 \text{ otherwise.} \tag{4}$$

[3]In fact, any kernel $k(\cdot,\cdot)$ satisfying (3) can be normalized into $k'(\cdot,\cdot)$ with $k'(x_i,x_j) = (k(x_i,x_j)/\sqrt{k(x_i,x_i)k(x_j,x_j)})$ when $\mathbf{x}_i, \mathbf{x}_j$ nonzero, and the resulting functional $k'(\cdot,\cdot)$ is also a kernel (see, e.g., [19]).

[4]This is true as long as the intersection of the hyperplane with the positive orthant of $\mathcal{S}$ is not empty.

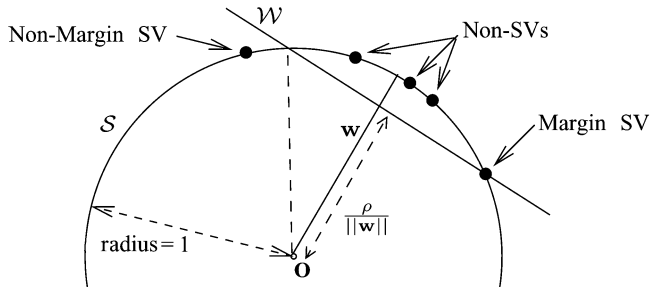

Fig. 2. In feature space $\mathcal{H}$, the mapped training vectors $\mathbf{x}_i = \phi(x_i), i = 1,\ldots,m$ are located on a hypersphere $\mathcal{S}$. The optimization problem of (4) yields $\mathbf{w}$ and ρ, which define the separating hyperplane $\mathcal{W}$. In feature space, the density support estimate $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{H}}$ is the segment of $\mathcal{S}$ limited by $\mathcal{W}$. The distance between the hyperplane and $\mathbf{O}$ is called the *margin*, and it equals $d(\mathbf{O},\mathcal{W})_H = \rho/\|\mathbf{w}\|_{\mathcal{H}}$.

In (4), it is not required that *all* the training points verify $\langle \mathbf{w}, \mathbf{x}\rangle_{\mathcal{H}} - \rho \geq 0$ because in many realistic situations, the training set $\boldsymbol{x}$ may contain outliers, that is, vectors (descriptors) that are not representative of the signal/system considered [16], [20]. Choosing $\mathcal{W}$ as in (4) is equivalent to choosing the decision function $f_{\boldsymbol{x}}(\cdot)$ such that $f_{\boldsymbol{x}}(x) = \langle \mathbf{w}, \mathbf{x}\rangle_{\mathcal{H}} - \rho, \mathbf{w}\in\mathcal{H}, \rho > 0$, where we recall that $\mathbf{x} = \phi(x)$. Similar to the input space settings, the decision function $f_{\boldsymbol{x}}(\cdot)$ defines the region $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{H}}$ as the segment of the hypersphere where $f_{\boldsymbol{x}}(\cdot)$ is positive. In the remainder of this paper, the region $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{X}}$ (resp. $\mathcal{R}_{\boldsymbol{x}}^{\mathcal{H}}$) will be referred to as the density support estimate of the unknown pdf $p(\cdot)$ in $\mathcal{X}$ (resp. in $\mathcal{H}$).

B. Selection of the Optimal Hyperplane

Among all possible hyperplanes $\mathcal{W}$, the ν-SV approach selects $\mathbf{w}$ and ρ with maximum *margin* (see Fig. 2), which results in [16], [20]

$$\max_{\mathbf{w},\boldsymbol{\xi},\rho} -\frac{1}{2}\|\mathbf{w}\|_{\mathcal{H}}^2 - \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i + \rho$$
$$\text{subject to } \langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{H}} \geq \rho - \xi_i \quad \text{and} \quad \xi_i \geq 0 \tag{5}$$

where ν is a positive parameter $(0 < \nu \leq 1)$ that tunes the amount of possible outliers [16], [20]. This optimization problem admits the following interpretation in $\mathcal{H}$: The margin is maximized under the constraint that most training vectors $\mathbf{x}_i, i = 1,\ldots,m$ are located in the half-space bounded by $\mathcal{W}$ that does not contain the center $\mathbf{O}$ of $\mathcal{S}$; see Fig. 2. The so-called *slack variables* $\xi_i, i = 1,\ldots,m$ implement the hinge loss by linearly penalizing outliers, that is, the training vectors that are located on the "wrong" side of $\mathcal{W}$. Margin maximization is the core principle of SV algorithms, as it can be shown that maximum margin hyperplanes have minimum regularized risk[5] $R_{\text{reg}}(f_{\boldsymbol{x}})$ [15]–[17]. Note that the complexity of $f_{\boldsymbol{x}}(\cdot)$ is penalized in (5) by $\|\mathbf{w}\|_{\mathcal{H}}^2$.

[5]Actually, maximum margin hyperplanes ensure a minimum Vapnik–Chervonenkis *upper bound* on the true risk [15]–[17].

This convex quadratic optimization problem is solved by introducing Lagrange multipliers $\alpha_i, i = 1, \ldots, m$, yielding the dual optimization problem (see, e.g., [16])

$$\min_{\boldsymbol{\alpha},\rho} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j k(x_i, x_j)$$

$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{\nu m} \text{ for } i = 1, \ldots, m \quad \text{and}$$

$$\sum_{i=1}^{m}\alpha_i = 1 \tag{6}$$

which is solved using a numerical procedure, such as the LOQO algorithm [21], [22]. $\mathbf{w}$ is given by

$$\mathbf{w} = \sum_{i=1}^{m}\alpha_i \mathbf{x}_i \tag{7}$$

which implies

$$f_{\boldsymbol{x}}(\cdot) = \sum_{i=1}^{m}\alpha_i k(x_i, \cdot) - \rho. \tag{8}$$

The solution is sparse: Most Lagrange multipliers (referred to as weights in the following) α_i are zero. Corresponding training vectors are called *Nonsupport Vectors* (NSVs) and are located inside $\mathcal{R}^{\mathcal{X}}_{\boldsymbol{x}}$ (or, equivalently, inside $\mathcal{R}^{\mathcal{H}}_{\boldsymbol{x}}$); see Fig. 2. Vectors such that $0 < \alpha_i < 1/\nu m$ are *margin support vectors* (MSVs) and are located on the boundary. Finally, *nonmargin support vectors* (NMSVs) are outliers; they verify $\alpha_i = 1/\nu m$. The parameter ν plays an important role: ν is an upper bound on the fraction of NMSVs and a lower bound on the fraction of SVs in $\boldsymbol{x}$ [20]. Moreover, under mild conditions, ν asymptotically equals both the fraction of NMSVs and SVs with probability 1.

C. Mercer Kernels

Results exposed above depend on the selection of a mapping $\phi(\cdot)$, which defines $k(\cdot,\cdot)$ via the dot product in $\mathcal{H}$. In practice, however, we observe that $\phi(\cdot)$ only appears in dot products, i.e., we only need to consider the kernel given in (3). With a reverse point of view, it is possible to instead specify a kernel $k(\cdot,\cdot)$ for which a mapping $\phi(\cdot)$ and a space $\mathcal{H}$ verify (3) [16], and [19]. Such kernels need to verify the necessary and sufficient condition given by Mercer [23]. From now on, unless explicitly stated otherwise, we consider that $\mathcal{X}$ is an Euclidean space and that the kernel is the Gaussian kernel noted $k_\sigma(\cdot,\cdot)$ with spread parameter σ (note, however, that the elements presented below remain true for any Mercer kernel such that $k(x,x) = 1, x \in \mathcal{X}$):

$$\forall (x, x') \in \mathcal{X}\times\mathcal{X}, \quad k_\sigma(x, x') = \exp\left(-\frac{\|x - x'\|^2_{\mathcal{X}}}{2\sigma^2}\right) \tag{9}$$

where $\|\cdot\|_{\mathcal{X}}$ is a norm in $\mathcal{X}$.

III. Kernel Change Detection Algorithm

As explained in Section I, our general framework for abrupt change detection requires a dissimilarity measure $\mathcal{D}(\cdot,\cdot)$ aimed at comparing the sets of descriptors $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$. More precisely, a relevant $\mathcal{D}(\cdot,\cdot)$ should output low values whenever $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ occupy the same region of the space $\mathcal{X}$ and large values whenever $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ occupy distinct regions. The density support estimation technique described in Section II can be used to build a dissimilarity measure, based on region comparison, as shown below.

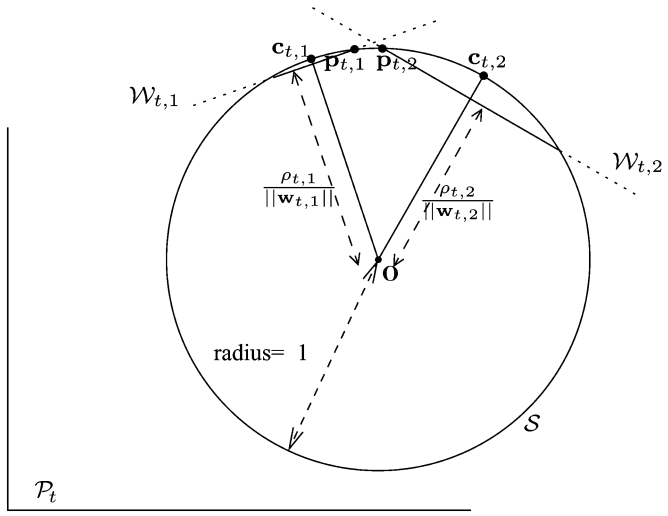

Fig. 3. SV single-class classifiers yield two regions $\mathcal{R}^{\mathcal{H}}_{\boldsymbol{x}_{t,1}}$ and $\mathcal{R}^{\mathcal{H}}_{\boldsymbol{x}_{t,2}}$, which are density support estimates in feature space. The circle represented corresponds to the intersection of the plane $\mathcal{P}_t$ (uniquely defined by $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,2}$) and $\mathcal{S}$. The intersection of the (prolongated) vector $\mathbf{w}_{t,1}$ (resp. $\mathbf{w}_{t,2}$) with $\mathcal{S}$ yields $\mathbf{c}_{t,1}$ (resp. $\mathbf{c}_{t,2}$), and the intersection of the hyperplane $\mathcal{W}_{t,1}$ (resp. $\mathcal{W}_{t,2}$) with $\mathcal{S}$ in the plane $\mathcal{P}_t$ yields two points, one of which is denoted $\mathbf{p}_{t,1}$ (resp. $\mathbf{p}_{t,2}$). The situation plotted corresponds to an abrupt change, as both regions do not strongly overlap.

A. Dissimilarity Measure in Feature Space

Consider analysis time t. Assume that two single-class classifiers are trained *independently* on the sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$, yielding two regions $\mathcal{R}^{\mathcal{X}}_{\boldsymbol{x}_{t,1}}$ and $\mathcal{R}^{\mathcal{X}}_{\boldsymbol{x}_{t,2}}$ or, equivalently, in feature space $\mathcal{H}$, two hyperplanes $\mathcal{W}_{t,1}$ and $\mathcal{W}_{t,2}$ parameterized by $(\mathbf{w}_{t,1}, \rho_{t,1})$ and $(\mathbf{w}_{t,2}, \rho_{t,2})$. In $\mathcal{H}$, the vectors $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,2}$ define a two-dimensional (2-D) plane, denoted $\mathcal{P}_t$, that intersects the hypersphere $\mathcal{S}$ along a circle with center $\mathbf{O}$ and radius 1, as depicted in Fig. 3. Actually, in the least probable case, where $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,2}$ are collinear (which is highly unlikely), there is an infinity of planes $\mathcal{P}_t$, and one can select any of them.

In feature space, the plane $\mathcal{W}_{t,1}$ (resp. $\mathcal{W}_{t,2}$) bounds the segment of $\mathcal{S}$, where most of the mapped points in $\boldsymbol{x}_{t,1}$ (resp. $\boldsymbol{x}_{t,2}$) lie. A good indication of the dissimilarity between $\phi(\boldsymbol{x}_{t,1})$ and $\phi(\boldsymbol{x}_{t,2})$ is given by the arc distance between the segment centers $\mathbf{c}_{t,1}$ and $\mathbf{c}_{t,2}$, which are denoted $\mathbf{c}_{t,1}\widehat{\mathbf{c}}_{t,2}$; see Fig. 3. However, this dissimilarity measure is not useful for abrupt change detection because it is not scaled by the spread of both $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$. We thus propose the following dissimilarity measure defined in feature space (see Fig. 3 for the definition of $\mathbf{p}_{t,1}$ and $\mathbf{p}_{t,2}$):

$$\mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) = \frac{\mathbf{c}_{t,1}\widehat{\mathbf{c}}_{t,2}}{\mathbf{c}_{t,1}\widehat{\mathbf{p}}_{t,1} + \mathbf{c}_{t,2}\widehat{\mathbf{p}}_{t,2}}. \tag{10}$$

Clearly, $\mathcal{D}_{\mathcal{H}}$ is an intra-regions/inter-regions ratio inspired by the Fisher ratio (see Section IV-A and [12]). Considering $\boldsymbol{x}_{t,1}$ only, we see that the arc distance $\mathbf{c}_{t,1}\widehat{\mathbf{p}}_{t,1}$ is a measure of the spread of samples in $\phi(\boldsymbol{x}_{t,1})$ in feature space. The more these samples are spread, the larger the distance $\mathbf{c}_{t,1}\widehat{\mathbf{p}}_{t,1}$ and the smaller the margin $\rho_{t,1}/\|\mathbf{w}_{t,1}\|$. The dissimilarity measure $\mathcal{D}_{\mathcal{H}}$ has thus the expected behavior in feature space, namely, it is large for well separated sets, and it is small for strongly overlapping sets. In Section IV, we present a deeper study of the behavior of $\mathcal{D}_{\mathcal{H}}$.

Remark 1—Least probable cases:

In the least probable case where $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,2}$ are collinear, i.e., $\mathbf{c}_{t,1}\widehat{\mathbf{c}}_{t,2}= 0$, it becomes impossible to detect differences of spread between the sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$. The problem can be tackled by adding a small $\varepsilon > 0$ to the numerator of (10). Similarly, the least probable case where the $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ have zero spread can be overcome by introducing $\varepsilon' > 0$ in the denominator of (10). In the following, we assume that these least probable situations do not occur as they do not correspond to realistic situations.

B. Computation in Input Space

The dissimilarity measure $\mathcal{D}_{\mathcal{H}}$ derived above is completely defined in feature space. However, an important question remains: $\mathcal{D}_{\mathcal{H}}$ must be computed directly in input space, without explicitly computing $\phi(\cdot)$.

The computation of $\mathcal{D}_{\mathcal{H}}$ in input space is only possible if we can express it as a function of the kernel $k(\cdot,\cdot)$ applied to vectors in input space $\mathcal{X}$. In feature space, the arc distance between two vectors $\mathbf{a}$ and $\mathbf{b}$ with norm one verifies

$$\widehat{\mathbf{ab}}= \widehat{\mathbf{aOb}} \tag{11}$$

where $\widehat{\mathbf{aOb}}$ is the angle between $\mathbf{a}$ and $\mathbf{b}$. Besides

$$\begin{aligned}\langle \mathbf{a}, \mathbf{b}\rangle_{\mathcal{H}} &= \|\mathbf{a}\|_{\mathcal{H}}\|\mathbf{b}\|_{\mathcal{H}} \cos(\widehat{\mathbf{aOb}}) \\ &= \cos(\widehat{\mathbf{aOb}}).\end{aligned} \tag{12}$$

Putting everything together, the arc distance $\widehat{\mathbf{ab}}$ is given by

$$\widehat{\mathbf{ab}}= \arccos(\langle \mathbf{a}, \mathbf{b}\rangle_{\mathcal{H}}). \tag{13}$$

In our case, the $\arccos$ function is used properly because the vectors we consider are all located in the same orthant of $\mathcal{S}$; in other words, $0 \leq \widehat{\mathbf{aOb}} \leq \pi/2$. The dot product in (13) can be evaluated in terms of the kernel $k(\cdot,\cdot)$ only if $\mathbf{a}$ and $\mathbf{b}$ are in the linear span of vectors in $\mathcal{X}$ [16]. In our case, the arc distance $\mathbf{c}_{t,1}\widehat{\mathbf{c}}_{t,2}$ can indeed be expressed in terms of $k(\cdot,\cdot)$: First, $\mathbf{c}_{t,1} = \mathbf{w}_{t,1}/\|\mathbf{w}_{t,1}\|_{\mathcal{H}}$ and $\mathbf{c}_{t,2} = \mathbf{w}_{t,2}/\|\mathbf{w}_{t,2}\|_{\mathcal{H}}$, and then

$$\mathbf{c}_{t,1}\widehat{\mathbf{c}}_{t,2}= \arccos\left(\frac{\langle \mathbf{w}_{t,1}, \mathbf{w}_{t,2}\rangle_{\mathcal{H}}}{\|\mathbf{w}_{t,1}\|_{\mathcal{H}}\|\mathbf{w}_{t,2}\|_{\mathcal{H}}}\right). \tag{14}$$

Second, using the fact that $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,2}$ are linear combinations of weighted kernels, as is shown in (7)

$$\frac{\langle \mathbf{w}_{t,1}, \mathbf{w}_{t,2}\rangle_{\mathcal{H}}}{\|\mathbf{w}_{t,1}\|_{\mathcal{H}}\|\mathbf{w}_{t,2}\|_{\mathcal{H}}} = \frac{\boldsymbol{\alpha}_{t,1}^{\mathrm{T}} K_{t,12}\boldsymbol{\alpha}_{t,2}}{\sqrt{\boldsymbol{\alpha}_{t,1}^{\mathrm{T}} K_{t,11}\boldsymbol{\alpha}_{t,1}}\sqrt{\boldsymbol{\alpha}_{t,2}^{\mathrm{T}} K_{t,22}\boldsymbol{\alpha}_{t,2}}} \tag{15}$$

where $\boldsymbol{\alpha}_{t,1}$ (resp. $\boldsymbol{\alpha}_{t,2}$) is the column vector whose entries are the parameters of $\mathbf{w}_{t,1}$ (resp. $\mathbf{w}_{t,2}$) that have been computed during training; see (7). The kernel matrix $K_{t,uv}, (u,v) \in \{1,2\} \times \{1,2\}$ has entries at row #i and column #j given by $k(x_{t,u}^i, x_{t,v}^j)$, where $x_{t,u}^i$ is the training vector #i in the set $\boldsymbol{x}_{t,u}$. Similar calculations can be applied to $\mathbf{c}_{t,1}\widehat{\mathbf{p}}_{t,1}$ and $\mathbf{c}_{t,2}\widehat{\mathbf{p}}_{t,2}$:

$$\mathbf{c}_{t,i}\widehat{\mathbf{p}}_{t,i}= \arccos\left(\frac{\rho_{t,i}}{\sqrt{\boldsymbol{\alpha}_{t,i}^{\mathrm{T}} K_{t,i}\boldsymbol{\alpha}_{t,i}}}\right), \quad i = \{1,2\} \tag{16}$$

which completes the derivation.

Remark 2—Equivalent definition of $\mathcal{D}_{\mathcal{H}}$:

$\mathcal{D}_{\mathcal{H}}$ can be equivalently defined by replacing $\mathbf{p}_{t,1}$ in (10) by any Margin SV (denoted $\mathbf{x}_{t,1}^{\mathrm{MSV}}$), which might not be in $\mathcal{P}_t$. The arc distance $\mathbf{c}_{t,1}\widehat{\mathbf{p}}_{t,1}$ equals $\mathbf{c}_{t,1}\widehat{\mathbf{x}}_{t,1}^{\mathrm{MSV}}$ because $\mathbf{p}_{t,1}$ and $\mathbf{x}_{t,1}^{\mathrm{MSV}}$ are both located on the intersection of $\mathcal{S}$ with $\mathcal{W}_{t,1}$ and have the same arc distance to $\mathbf{c}_{t,1}$. The same reasoning also holds for $\mathbf{p}_{t,2}$ and $\mathbf{x}_{t,2}^{\mathrm{MSV}}$.

C. General Framework for the KCD Algorithm

The elements derived in Section I and the dissimilarity measure introduced enable the presentation of the full abrupt change detection algorithm. Algorithm 1 below summarizes the whole procedure. As an intermediate step, we assume that the descriptors have already been extracted from the input time series y; therefore, we consider the series of descriptors $x_t, t = 1, 2, \ldots$.

Algorithm 1: Kernel Change Detection (KCD) Algorithm

Step 0: Initialization

- Select the training sets sizes m_1, m_2, the algorithm parameter ν and the threshold η.
- Set $k(\cdot,\cdot)$ to be, e.g., a Gaussian kernel with parameter σ.
- Set $t \leftarrow m_1 + 1$.

Step 1: online change detection

- Train an SV single-class classifier on the immediate past training set $\boldsymbol{x}_{t,1} = \{x_{t-m_1}, \ldots, x_{t-1}\}$ and train independently another SV single-class classifier on the immediate future training set $\boldsymbol{x}_{t,2} = \{x_t, \ldots, x_{t+m_2-1}\}$. The optimization process yields $(\mathbf{w}_{t,1}, \rho_{t,1})$ and $(\mathbf{w}_{t,2}, \rho_{t,2})$, or, equivalently, the parameters $(\boldsymbol{\alpha}_{t,1}, \rho_{t,1})$ and $(\boldsymbol{\alpha}_{t,2}, \rho_{t,2})$.
- Compute the decision index $I(t) = \mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2})$ defined in (10) using (14)–(16).
- Based on $I(t)$, decide:
  - If $I(t) \geq \eta$ then a change is detected at time instant t,
  - If $I(t) < \eta$ then no change is detected at time instant t.
- Set $t \leftarrow t + 1$ and go to Step 1.

In Algorithm 1, an abrupt change is detected whenever the time index $I(t)$ is over a threshold η. This approach is classical [1], and η tunes the false positive/false negative ratio. Testing time instant t requires the knowledge of the descriptors time-series $\boldsymbol{x}$ up to time instant $(t+m_2-1)$.

The SV single-class classifier is trained twice at each iteration, as novelty detection is performed over both the immediate past and the immediate future sets w.r.t. time instant t. A noncomputationally efficient procedure would consist of recomputing from scratch the parameters $(\boldsymbol{\alpha}_{t,i}, \rho_{t,1})(i=1,2)$ by solving the optimization problem of (6) at each iteration. Instead, it can be noticed that, e.g., $\boldsymbol{x}_{t+1,1}$ can be obtained from $\boldsymbol{x}_{t,1}$ by incrementing with x_t while decrementing with x_{t-m_1}. Efficient solutions for implementing this procedure include procedures based on the incremental/decremental technique derived in [24] and [25] or on stochastic gradient descent [26].

IV. Discussion

In Section III, we have described a dissimilarity measure $\mathcal{D}_{\mathcal{H}}$ based on the arc distance in feature space. In the simple case where pdfs p_1 and p_2 are Gaussian and radial, the Fisher ratio is a standard measure of dissimilarity [12], which is defined by

$$\mathcal{D}_F(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) = (\hat{\mu}_{t,1} - \hat{\mu}_{t,2})^{\mathrm{T}} \left(\hat{\boldsymbol{\Sigma}}_{t,1} + \hat{\boldsymbol{\Sigma}}_{t,2}\right)^{-1} (\hat{\mu}_{t,1} - \hat{\mu}_{t,2}) \quad (17)$$

where $\hat{\mu}_{t,i}$ and $\hat{\Sigma}_{t,i}$ are empirical estimates of the mean and covariance matrix of p_i, based on the set $\boldsymbol{x}_{t,i}$, for $i=1,2$. A key question is as follows: Does $\mathcal{D}_{\mathcal{H}}$ behave like the Fisher ratio in this simple case? A negative answer to this question would make $\mathcal{D}_{\mathcal{H}}$ inappropriate. In Section IV-A, we show that it actually behaves like the Fisher ratio in simple cases. In addition, we show that $\mathcal{D}_{\mathcal{H}}$ can deal with complicated situations where the Fisher ratio is not properly defined. Section IV-B is devoted to some discussion about the comparison with dissimilarity measures built on Parzen density estimates together with divergences. Section IV-C investigates other SV approaches to Abrupt Change Detection. Kernel selection and tuning of the algorithm are dealt with in Section IV-D.

A. Connection With the Fisher Ratio in Input Space

Consider $\boldsymbol{x}_{t,1}$ (the same reasoning holds for $\boldsymbol{x}_{t,2}$), and assume that the pdf $p_{t,1}$ is Gaussian with mean $\mu_{t,1}$ and covariance matrix $\Sigma_{t,1} = \sigma_{t,1}^2 \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Consider that the training set $\boldsymbol{x}_{t,1}$ is randomly generated from $p_{t,1}$. In this special case, the Fisher ratio becomes

$$\mathcal{D}_F(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) = \frac{\|\hat{\mu}_{t,1} - \hat{\mu}_{t,2}\|_{\mathcal{X}}^2}{\hat{\sigma}_{t,1}^2 + \hat{\sigma}_{t,2}^2}. \quad (18)$$

Besides, the following theorem holds (a proof can be found in [27]).

Theorem 3: Let $\boldsymbol{x}$ be a set of m vectors sampled i.i.d. from a pdf $p(\cdot)$ with mean μ and such that $p(\cdot)$ is a function of a distance $d_{\mathcal{X}}(\mu, \cdot)$ in $\mathcal{X}$. Let $k(\cdot,\cdot)$ be a normalized kernel such that

$$k(x,x') = \exp(-\lambda d_{\mathcal{X}}(x,x')) \text{ for all } (x,x') \in \mathcal{X} \times \mathcal{X} \quad (19)$$

with $\lambda > 0$. Then, with probability 1, the center $\mathbf{c}$ yielded by the ν-one class SVM converges to $\boldsymbol{\mu} = \phi(\mu)$ when $m \to \infty$.

Note that Theorem 3 is more general than the Gaussian case considered here. From Theorem 3, with probability one, for $m_1 \to \infty, \boldsymbol{\mu}_{t,1} = \phi(\mu_{t,1})$ is the limit of the center $\mathbf{c}_{t,1}$. In other words, the center $\mathbf{c}_{t,1}$ of the segment of the sphere $\mathcal{S}$ becomes infinitely close to the image $\boldsymbol{\mu}_{t,1}$ of $\mu_{t,1}$ in feature space. Using Remark 2, one can write

$$\mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) \underset{m_1, m_2 \to \infty}{\longrightarrow} \frac{\widehat{\boldsymbol{\mu}}_{t,1}\, \boldsymbol{\mu}_{t,2}}{\boldsymbol{\mu}_{t,1} \widehat{\mathbf{x}}_{t,1}^{\mathrm{MSV}} + \boldsymbol{\mu}_{t,2} \widehat{\mathbf{x}}_{t,2}^{\mathrm{MSV}}}. \quad (20)$$

Replacing arc distances by $\arccos(\langle \cdot, \cdot \rangle_{\mathcal{H}})$ and dot products in feature space by kernels, we have (21), shown at the bottom of the page. For the Gaussian kernel, it becomes (22), shown at the bottom of the page, where $g(u) = \arccos(\exp(-(1/2\sigma^2)u))$. It can also be shown that $\|\mu_i - x_i^{\mathrm{MSV}}\|_{\mathcal{X}}^2$ is asymptotically proportional to the variance σ_i^2 of $p_i, i=1,2$, and we finally have

$$\mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) \underset{m_1, m_2 \to \infty}{\longrightarrow} \frac{g\left(\|\mu_{t,1} - \mu_{t,2}\|_{\mathcal{X}}^2\right)}{g\left(\beta_{t,1}\sigma_{t,1}^2\right) + g\left(\beta_{t,2}\sigma_{t,2}^2\right)} \quad (23)$$

where we note that $g(\cdot)$ is an increasing function such that $g(0) = 0$ and $(\beta_{t,1}, \beta_{t,2}) > 0$ are constants. This result is quite important because it shows that for sets with radial Gaussian distributions, $\mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2})$ behaves like the standard Fisher ratio $\mathcal{D}_F(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2})$ in input space. This is assessed by the simulations presented in Fig. 4.

$$\mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) \underset{m_1, m_2 \to \infty}{\longrightarrow} \frac{\arccos(k(\mu_{t,1}, \mu_{t,2}))}{\arccos(k(\mu_{t,1}, x_{t,1}^{\mathrm{MSV}})) + \arccos(k(\mu_{t,2}, x_{t,2}^{\mathrm{MSV}}))}. \quad (21)$$

$$\mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}) \underset{m_1, m_2 \to \infty}{\longrightarrow} \frac{g\left(\|\mu_{t,1} - \mu_{t,2}\|_{\mathcal{X}}^2\right)}{g\left(\|\mu_{t,1} - x_{t,1}^{\mathrm{MSV}}\|_{\mathcal{X}}^2\right) + g\left(\|\mu_{t,2} - x_{t,2}^{\mathrm{MSV}}\|_{\mathcal{X}}^2\right)} \quad (22)$$

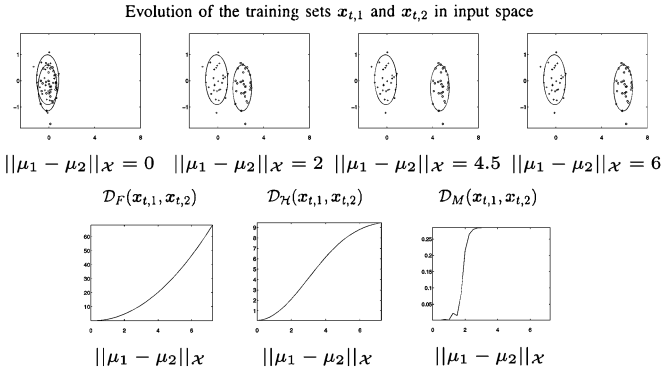

Fig. 4. Comparison of the dissimilarity measure $\mathcal{D}_{\mathcal{H}}$ with the Fisher ratio in input space on 2-D Gaussian training sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ with mean μ_1 and μ_2. (Top) Training sets in input space. From left to right, the distance $\|\mu_1 - \mu_2\|_{\mathcal{X}}$ increases, i.e., the sets are better and better separated. Solid lines represent regions estimated by the ν single-class SVM. (Bottom) The Fisher ratio in input space and $\mathcal{D}_{\mathcal{H}}$ are plotted w.r.t. the distance $\|\mu_1 - \mu_2\|_{\mathcal{X}}$. As can be seen, both dissimilarity measures increase as the training sets become better separated. The dissimilarity measure $\mathcal{D}_M$, which was introduced in Subsection IV.C, shows unsatisfactory fluctuations.

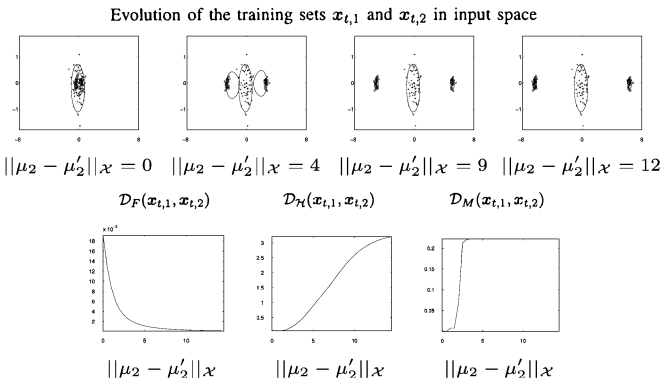

Fig. 5. Comparison of $\mathcal{D}_{\mathcal{H}}$ with the input-space Fisher ratio on 2-D training sets. The training set $\boldsymbol{x}_{t,1}$ is sampled according to a Gaussian pdf with mean μ_1 and is kept steady, whereas $\boldsymbol{x}_{t,2}$ is a mixture of two Gaussian pdfs with means μ_2 and μ_2'. (Top) Training sets in input space. From left to right, the distance $\|\mu_2 - \mu_2'\|_{\mathcal{X}}$ increases with $\mu_1 = (\mu_2 + \mu_2')/2$, i.e., the sets are better and better separated but keep the same means. (Bottom) The Fisher ratio in input space and $\mathcal{D}_{\mathcal{H}}$ are plotted w.r.t. the distance $\|\mu_2 - \mu_2'\|_{\mathcal{X}}$. As can be seen, $\mathcal{D}_{\mathcal{H}}$ increases as the training sets become better separated, but the Fisher ratio decreases. The dissimilarity measure $\mathcal{D}_M$ shows unsatisfactory fluctuations.

As defined in feature space where the points $\mathbf{c}_{t,i}$ and $\mathbf{p}_{t,i}, i = 1, 2$ are always defined, $\mathcal{D}_{\mathcal{H}}$ is adapted to situations where the vectors are located in regions of $\mathcal{X}$ with complicated shapes. In particular, if the support of density, say, p_2 is nonconnected, then $\mathcal{D}_{\mathcal{H}}$ is still defined. In Fig. 5, simulations on a toy example show that $\mathcal{D}_{\mathcal{H}}$ still has the expected behavior, namely, it is small for similar training sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$, and large for training sets with different shapes, even though the mean of p_1 remains close to the mean of p_2. This illustrates the interest of $\mathcal{D}_{\mathcal{H}}$ in general situations.

Remark 4—Kernel Fisher Discriminant: $\mathcal{D}_{\mathcal{H}}$ is built according to the same principle as the Rayleigh coefficient used in Fisher analysis [12]. Yet it is completely different from a Kernel Fisher Discriminant analysis (KFD, see e.g., [28]) as, in our case, the direction vectors $\mathbf{w}_{t,i} \in \mathcal{H} (i = 1, 2)$ are computed using two independent single-class SV machines. In KFD, it is optimized to obtain the projection directions that maximize the variance of the projected vectors.

This asymptotic result is a consistency check about the dissimilarity measure $\mathcal{D}_{\mathcal{H}}$, because it has the same behavior as a known optimal measure under Gaussian assumption, namely, the Fisher ratio. In terms of computational complexity, however, the Fisher ratio is generally cheaper. It must be noticed, however, that $\mathcal{D}_{\mathcal{H}}$ addresses much more general classes of problems including those with limited amount of training samples, large dimensional descriptors, or unknown pdfs with possibly complicate shapes and nonconnected supports: In these situations, using the Fisher ratio is inadequate. It also has little sense to compute the Fisher ratio when the size m of the training set is small compared with the dimension of the input space.

B. Comparison With Parzen Density Estimation Techniques

For a given kernel $k(\cdot, \cdot) \geq 0$ such that $\int_{\mathcal{X}} k(x, x')dx' = 1$ for all x in $\mathcal{X}$, the *Parzen windows* density estimation technique provides the following estimate $\hat{p}_m(\cdot)$ of the pdf $p(\cdot)$:

$$\hat{p}_m(\cdot) = \frac{1}{m} \sum_{i=1}^{m} k(x_i, \cdot). \tag{24}$$

Abrupt change detection can be achieved by computing $\hat{p}_m(\cdot \,|\, 1)$ and $\hat{p}_m(\cdot \,|\, 2)$ (see Section I-A), respectively, on $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ and then comparing the two estimated densities using a dissimilarity measure, such as the KL divergence. Its numerical computation can, however, be difficult, as it involves an integral in possibly large dimension. When $\hat{p}_m(\cdot \,|\, i), i = 1, 2$ dies off at a strong rate with monotonic tails, a Monte Carlo approximation of the KL divergence based on a sampling done according to $\hat{p}_m(x \,|\, 1)$ is numerically unstable. The approach to tackle this problem can be found in the ICA literature; see, e.g., [14]. Note that when $\nu = 1$, the ν-SV one-class decision function is build on the Parzen window estimate [16].

Another possible solution consists of computing the Fisher ratio directly in feature space. This approach, however, cannot deal with outliers. Moreover, the empirical estimate of the covariance matrix is poor, since it is computed using only m_j training vectors in dimension $m_j, j = 1, 2$.

C. Other Possible SV Approaches to Abrupt Change Detection?

In this subsection, we investigate other possible SV-based approaches to abrupt change detection. A first alternative SV approach considers the composite two-class training set at time t written as $\boldsymbol{x}_t = \{\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}\}$ and applies a mere two-class ν-SVM (see, e.g., [16] for a detailed presentation of the method). The sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ are attributed different labels (for instance, $+1$ for the $\boldsymbol{x}_{t,1}$ and -1 for $\boldsymbol{x}_{t,2}$) and therefore define two classes of vectors. The 2-class ν-SVM yields a margin $1/\|\mathbf{w}_t\|$, which is denoted $\mathcal{D}_M(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,1})$ and indicates the distance in feature space between the separating hyperplane and the closest mapped training vectors (called margin SVs). Of course, the larger the margin, the more separated the sets $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$. However, this approach is not satisfactory in the

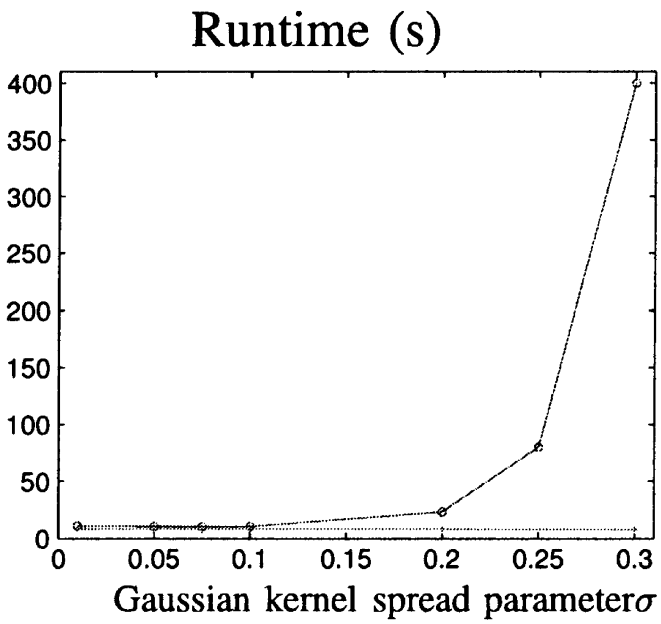

Fig. 6. Runtime of the two-class classification approach (computation of $\mathcal{D}_{\mathcal{H}}$, circles) and KCD approach (computation of $\mathcal{D}_M$, crosses) as functions of the Gaussian kernel spread parameter σ for the toy example of Fig. 4. The parameters of both algorithms are $\nu = 0.2, m_1 = m_2 = 40$.

context of abrupt change detection for three major reasons. First, when no abrupt change occurs, vectors from $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ are located in the same region of $\mathcal{X}$, and defining a margin between them is meaningless. Second, outliers are defined in different ways in our KCD approach and in 2-class ν-SVM: In the 2-class case, vectors are considered to be outliers (i.e., as NMSVs) if they are close to the hyperplane that best separates $\boldsymbol{x}_{t,1}$ from $\boldsymbol{x}_{t,2}$. In other words, outliers in the set $\boldsymbol{x}_{t,1}$ are defined w.r.t. the other set $\boldsymbol{x}_{t,2}$ (and conversely) and not w.r.t. the underlying process that generates the time series of descriptors; see Figs. 4 and 5. Third, the computational load is much heavier with the 2-class ν-SVM approach, as training is performed over $m_1 + m_2$ vectors instead of m_1 and m_2 vectors (training classical SVM scales roughly with $O(m^3)$ or even $O(m^4)$, and optimized solver complexity reaches $O(m^2)$); see Fig. 6. When no change occurs, training the 2-class ν SVM takes a very long time compared with our approach.[6] This, of course, also occurs with large values for the kernel spread parameter σ (see Fig. 6); the greater the σ, the greater the runtime of the 2-class-based change detection method. On the other hand, for a fixed number $m_1 = m_2$ of training vectors, the KCD algorithm runtime is roughly constant w.r.t. σ.

D. Tuning the Algorithm—Kernel Selection

For the sake of clarity, we presented the KCD algorithm as a decision layer using previously extracted descriptors. We can, however, have another interpretation of the procedure, based on the following property [19]:

[6]The simulations were conducted using the Spider toolbox [22] default optimizer; the switch to other optimization routines did not change the observed behavior.

Property 5: Let $k(\cdot,\cdot)$ be a Mercer kernel on $\mathcal{X} \times \mathcal{X}$. Then, $k(q(\cdot), q(\cdot))$, where $q(\cdot)$ is a $\mathcal{X}$-valued function on $\mathcal{X}$, is also a Mercer kernel.

In other words, the preprocessing function $q(\cdot)$ that maps the y_τ's into descriptors x_t's, together with the Gaussian kernel applied x_t's, can also be interpreted as the direct application of a more general Mercer kernel (that includes the preprocessing function $q(\cdot)$) to original time series y_τ. The parameters of the KCD algorithm are thus ν, m_1, m_2, the parameters of $k(\cdot,\cdot)$ (i.e., σ in the Gaussian kernel case), the parameters of $q(\cdot)$, and the detection threshold parameter η.

The influence of $q(\cdot)$ clearly depends on the descriptors extraction technique selected. However, KCD efficiency is clearly related to the ability of $q(\cdot)$ to characterize abrupt changes as shifts in the descriptors space $\mathcal{X}$. Typical descriptor extractors for audio are time-frequency representations (a discussion about time-frequency preprocessing parameters tuning can be found in [25] and [29]).

More generally, the KCD algorithm parameters can be tuned as follows in applications. When dealing with complicated applications, such as speech/music processing or industrial applications where no underlying physical model is available, the use of high-level descriptors proves to be particularly efficient. These include time-frequency representations, mel-cepstral coefficients, wavelet coefficients, etc. As these are redundant transforms with known good properties for reflecting transient behavior, they can be expected to provide descriptors adapted to the change detection framework we address. The descriptors they yield are large dimensional, but this is not a problem due to the nice properties of SV novelty detection; see Section II and [16]. Most of these techniques can be further adapted to the signal considered. In the case of time-frequency representations,[7] this adaptivity includes the choice of the time-frequency kernel and windows [31], [32]. Of course, if the change is expected to happen in some known domain, projection on relevant subspace should also be considered.

The kernel $k(\cdot,\cdot)$ defines the map $\phi : \mathcal{X} \rightarrow \mathcal{H}$. In the Gaussian kernel case, σ influences the location of vectors $\phi(x_i)$'s on the hypersphere $\mathcal{S}$. If σ is much smaller than the distances $\|x - x'\|_{\mathcal{X}}, (x, x') \in \mathcal{X}^2$, then in feature space, the angle between any $\mathbf{x}$ and $\mathbf{x}'$ is close to $\pi/2$. In other words, $\phi(\mathcal{X})$ occupies a large portion of the positive orthant of the hypersphere $\mathcal{S}$. On the contrary, when $\sigma \gg \|x - x'\|_{\mathcal{X}}$, for any $(x, x') \in \mathcal{X}^2$, mapped training vectors are close one to another in feature space $\mathcal{H}$; the angle between vectors from $\phi(\mathcal{X})$ is close to 0. None of these situations is satisfactory, and choosing σ one order of magnitude smaller than the average distance $\|x - x'\|_{\mathcal{X}}, (x, x') \in \mathcal{X}^2$ is sensible and easy to implement; see Fig. 7.

Tuning of m_1 and m_2 is generally imposed by the dynamics of the signal/system y: Small m_1 and m_2 make the KCD algorithm detect frequent, small changes. On the contrary, large m_1 and m_2 enable the detection of long-term changes and neglect small changes. External constraints may also be considered: If

[7]More information about time-frequency representations can be found in [30].

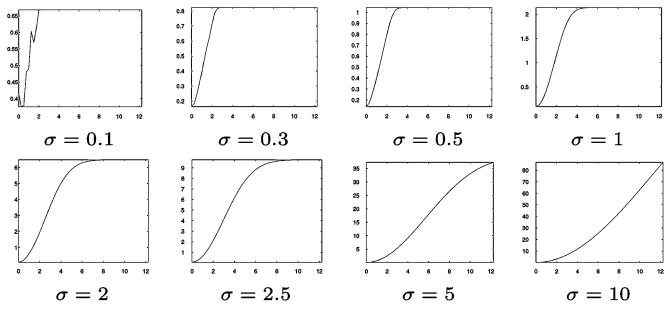

Fig. 7. Comparison of the dissimilarity measure $\mathcal{D}_{\mathcal{H}}$ for different values of the kernel width σ for $\sigma = 0.1, 0.3, 0.5, 1, 2, 2.5, 5, 10$. The training sets are the same 2-D Gaussian sets as in Fig. 4. As can be seen, the KCD index has a sensible behavior for a large range of values for σ.

a small detection lag is required, then m_2 is kept small. Note, however, that like in any Machine Learning approach, the accuracy of the training phase (that is, of the computation of $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,2}$) increases together with m_1 and m_2.

The rate of outliers ν is tuned according to detection requirements: For values about 0.2 to 0.8, the influence of outliers is limited, which reduces the rate of false alarms. No significant difference is observed in detection performance for values in the range $0.2 \leq \nu \leq 0.8$. On the contrary, $\nu = 0$ is the so-called *hard margin* case, which yields more false alarms. The specific case $\nu = 1$ has been examined in Section IV-B.

Tuning the threshold automatically is still open research for the KCD algorithm, as for most change detection techniques. In the audio framework, supervised tuning on a short signal sample can be considered; it is effectively the method we employed for the music segmentation simulations reported in Section V-B.

V. Simulation Results

This section is dedicated to two simulation situations. We first compare the KCD algorithm to the GLR algorithm in the case of (synthetic) noisy sum-of-sines signals. KCD is then applied to music signal segmentation (organ and saxophone), where defining a convenient model suited to the GLR algorithm is almost intractable.

A. Noisy Sum-of-Sines Signals

In this subsection, we consider noisy sum-of-sines signals given by $y_t = a_1 \sin(2\pi f_1 t) + \cdots + a_n \sin(2\pi f_n t) + \epsilon_t, (t = 1, \ldots, N)$, where f_i are frequencies, a_i are amplitudes $(i = 1, \ldots, n)$, and ϵ is a Gaussian white noise with variance σ_ϵ^2. We implement a modified on-line version of the original GLR approach presented in [1]; then, we apply the KCD algorithm with time-frequency based descriptors.[8] Through the simulations we propose, we intend to compare the behavior of KCD and the GLR in a realistic situation.

We first describe the modified online GLR used in this simulation. GLR applies directly to the signal y and requires the choice of a model (with parameters $\boldsymbol{\theta}$) and a threshold η. For each t in $\{m_1' + 1, \ldots, N - m_2' + 1\}$, $\boldsymbol{\theta}$ is first estimated on the signal $\boldsymbol{y}_{t,0} = \{y_i\}, i = t - m_1', \ldots, t + m_2' - 1$, producing prediction errors $\boldsymbol{e}_0$. This corresponds to the hypothesis that no change occurs *at all* in this interval (hypothesis H_0). Then, for the same t, the model parameters $\boldsymbol{\theta}$ are estimated on $\boldsymbol{y}_{t,1} = \{y_i\}_{i=t-m_1', \ldots, t-1}$ (resp. $\boldsymbol{y}_{t,2} = \{y_i\}_{i=t, \ldots, t+m_2'-1}$) with prediction errors $\boldsymbol{e}_{t,1}$ (resp. $\boldsymbol{e}_{t,2}$). A change is detected in y whenever the likelihood ratio $(1 - (\|\boldsymbol{e}_{t,1}\|_2^2 + \|\boldsymbol{e}_{t,2}\|_2^2 / \|\boldsymbol{e}_0\|_2^2))$ exceeds the threshold η. The *immediate past* set size m_1' (resp. *immediate future* set size m_2') may differ from the size m_1 (resp. m_2) used in KCD, which corresponds to a set of descriptors. When tuning KCD and the GLR parameters, m_i and $m_i'(i = 1, 2)$ are chosen such that both approaches use the same set of signal samples $\{y_\tau\}, \tau = m_1' + 1, \ldots, N - m_2' + 1$ at each time instant. The framework for the modified GLR is presented in Fig. 8.

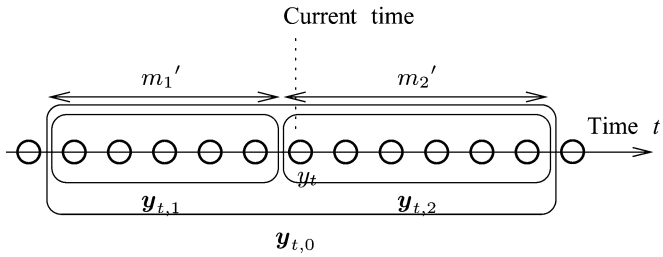

Fig. 8. Framework for the modified online GLR used in this subsection; the approach applies directly to the signal y, which is represented by circles.

Remark 6—Classical GLR approach:

The classical, batch approach for GLR applies the above procedure, with the sets $y_{t,1}$ and $y_{t,2}$ now defined as $y_{t,1} = \{y_i\}_{1,\ldots,t-1}$ and $y_{t,2} = \{y_i\}_{i=t,\ldots,N}$. The prediction errors $\boldsymbol{e}_0$ are computed on the whole signal $y_\tau, \tau = 1, \ldots, N$, and the likelihood ratio $(\|\boldsymbol{e}_{t,1}\|_2^2 + \|\boldsymbol{e}_{t,2}\|_2^2 / \|\boldsymbol{e}_0\|_2^2)$ is computed at each time instant t, as before. Hence, the information contained in the whole signal is used to test each time instant t as a possible change time. KCD can be adapted to this context by defining accordingly the immediate past of time instant t as $\boldsymbol{x}_{t,1} = \{x_i\}_{i=1,\ldots,t-1}$ and its immediate future as $\boldsymbol{x}_{t,1} = \{x_i\}_{i=t,\ldots,N}$. The size of $\boldsymbol{x}_{t,1}$ is $m_1 = t - 1$, and the size of $\boldsymbol{x}_{t,2}$ is $m_2 = N - t + 1$. Change times are estimated as in classical KCD whenever $\mathcal{D}_{\mathcal{H}}(\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2})$ peaks.

Synthetic signals are $N = 2048$ points long, with $n = 2$ sines of constant amplitudes $a_1 = a_2 = 1$ and frequencies $f_1 = 0.075, f_2 = 0.2$, which may jump abruptly at time $t = 1024$ to $f_1 = 0.1$ and $f_2 = 0.125$. For different SNRs $\{0, 1, 2, 3, 4, 5, 10, 25\}$ dB, 400 signal realizations are generated, half of which have an abrupt change at time $t = 1024$ and half of which have none. The spectrograms of two realizations of y are plotted in Fig. 9.

In engineering problems such as music/speech processing, a perfectly suitable and relevant parametric model can rarely be obtained; one has to choose the model that best fits the data and, in the same time, that takes into account constraints such as

[8]Applying a Gaussian kernel to time-frequency descriptors results in fact in a composite Gaussian/time-frequency kernel, as used in [11] and [25]

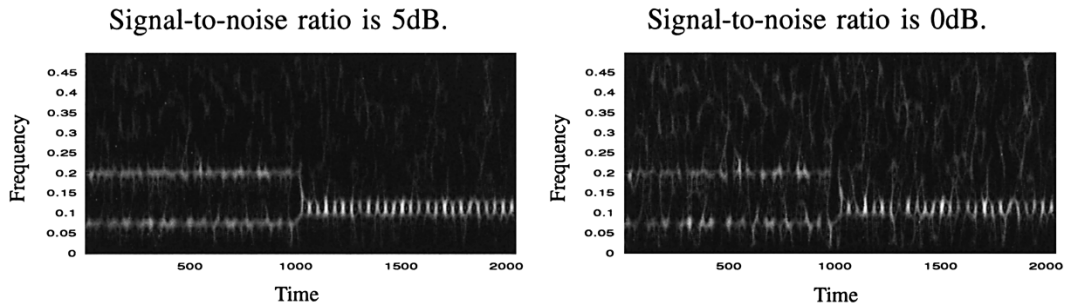

Fig. 9. Spectrograms of two realizations of the sum-of-sines signals used in the simulation. (Left) The SNR is 5 dB. (Right) The SNR is 0 dB. The spectrogram settings are described in Section V-A. On both realizations, an abrupt change occurs at time $t = 1024$.

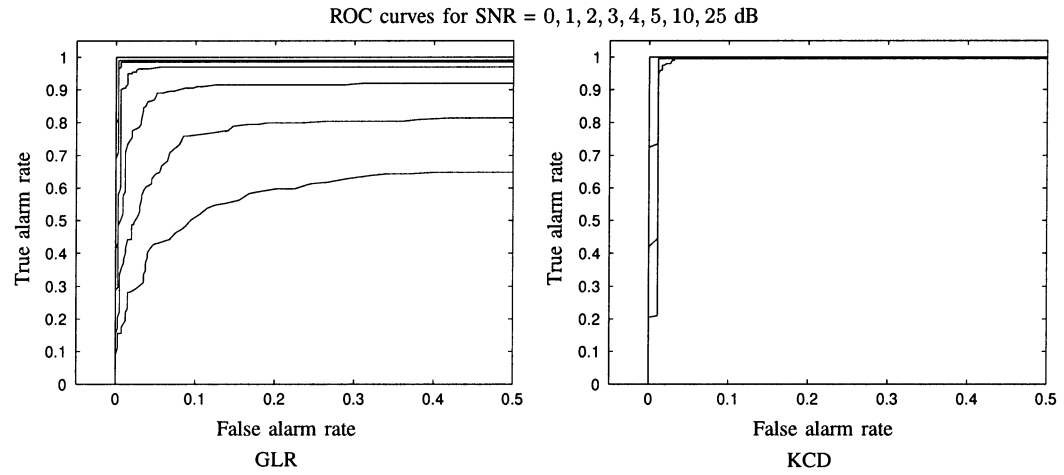

Fig. 10. Comparison of the performance of GLR and KCD algorithms on noisy sum-of-sines signals. The ROC curves display the true alarm rate as a function of the false alarm rate for (left) GLR and (right) KCD. The curves are plotted for different values of the SNR (from bottom to top on each figure): SNR = 0, 1, 2, 3, 4, 5, 10, and 25 dB. When the SNR is above 5 dB, both methods have excellent performance; for smaller SNRs, KCD outperforms the GLR approach: The modeling errors in the GLR increase.

complexity, computational load, etc.[9] A standard approach to music/speech signal segmentation consists of implementing the GLR test jointly with an autoregressive (AR) dynamic model; see [34] and [35]. Therefore, it is sensible to compare KCD to the state-of-the-art GLR + AR model with order 4. However, for fair comparison, the two methods are provided the same amount of data when testing a given possible change time instant. It should be noted that the modeling error made by implementing an order-4 AR model is mild as it still assumes the presence of two frequencies in the considered signal but does not correctly deal with the additive noise. This illustrates realistic situations where modeling errors cannot be avoided. Here, modeling errors increase when the SNR decreases. In KCD, this issue is not met directly as the approach is model-free; however, signal-dependence issues are embedded in the choice of the descriptors x.

The sizes of the sets $y_{t,1}$ and $y_{t,2}$ used in the GLR are $m'_1 = m'_2 = 171$. The KCD algorithm is parameterized as follows. The descriptors used are time-frequency subimages extracted from the smoothed-pseudo Wigner–Ville time-frequency representation (TFR) of y (see [6], [11], and [25] for other examples of time-frequency descriptors). The TFR smoothing windows are Gaussian with length 25 points (time smoothing window) and 61 points (frequency smoothing window). Each extracted descriptor is a subimage extracted from the whole TFR, made of 12 consecutive TFR columns that do not overlap; the training sets sizes are $m_1 = m_2 = 12$. This preprocessing uses 171 points of the signal y to define the immediate past and future sets for time instant t: Both methods use the same set of samples of the signal y to test each time instant t. The SV parameters are $\nu = 0.2$ (i.e., less than 20% of outliers are allowed in the training set) and $\sigma = 1.5$ for the Gaussian kernel.

Both KCD and GLR algorithms yield a stationarity index whose peaks are supposed to indicate abrupt changes. Performance is assessed via ROC curves, where the false alarm rate and the true alarm rate are defined as follows: A true alarm is decided if both 1) a true change is detected, and 2) the estimated time instant is less than 80 points beside the true change time instant (this corresponds to $\pm 2\%$ of the length of y). A false alarm is decided whenever an abrupt change is detected and matches neither point 1) nor point 2) above. The ROC curves obtained for different values for the SNR are plotted in Fig. 10. When the SNR is 5 dB or above, both methods yield excellent results. However, when the level of noise increases, KCD exhibits superior performance as, for a given SNR, the KCD ROC curve is well above the GLR ROC curve (see Fig. 10). This behavior is due to modeling errors in the use of the GLR: The model is an

[9]An example of full parametric model for music transcription may be found in [33].

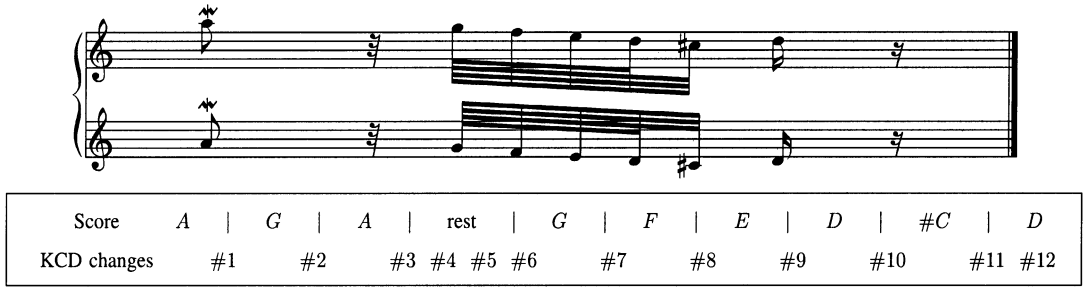

Fig. 11. (Top) Score of the training organ extract considered in this subsection. The music signal is 3.27 sec long and is half the first measure of the Toccata in D minor by J. S. Bach; it is played on the pipe organ in a church, which makes its segmentation a very tough task. (Bottom) KCD results: Notes are represented in upper row, with theoretical changes noted | ; the lower row corresponds to the (numbered) change detected by KCD. Changes #4, #5, and #12 are due to oversegmentation.

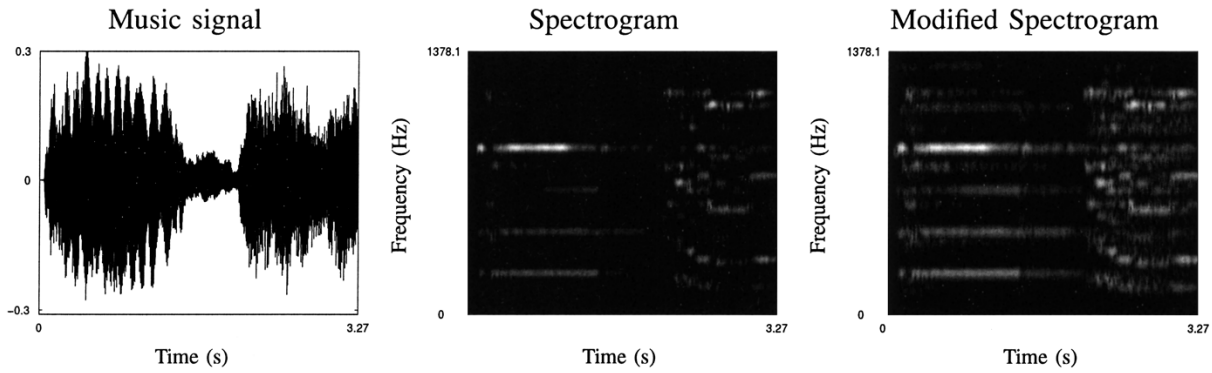

Fig. 12. (Left) As the music signal is mainly composed of notes that alternate very quickly, it is impossible to distinguish between them in the time domain. (Middle) The spectrogram of the considered signal is computed. (Right) The spectrogram is scaled to the power 0.05; descriptors are extracted from this modified TFR.

order-4 autoregressive model, whereas the signal y is a sum of two sines with noise. In that sense, KCD is not subject to modeling errors; this simulation also emphasizes that our approach proves to be more robust to noise in this example.

B. Applications to Music Segmentation

In this subsection, we apply the KCD algorithm to abrupt change detection in music to perform music segmentation. The first two signals processed here are recorded from a church pipe organ. A third signal is a short saxophone extract. Segmenting organ-in-church signals is considered an especially difficult task because sound reflects on church walls and considerably garbles the original sound, as the reverberation dies away quite slowly. Further difficulties arise from the tremulant or from the score when notes alternate very quickly. The music signal we consider includes examples of these two issues. Our purpose is to accurately detect all abrupt frequency changes in the music signal, that is, all the changes of notes in the signal. The choice of a well-known piece of music makes it easier to validate heuristically the change detected by KCD either by simply listening to the piece of music or by reading the corresponding score. The music signal is an extract of the Toccata in D minor by J. S. Bach [36]. The KCD parameters are tuned on a first 3.27-sec extract with sampling frequency $F_s = 44.1$ kHz. The corresponding score is presented in Fig. 11. A second extract is then used for validation; see Fig. 14.

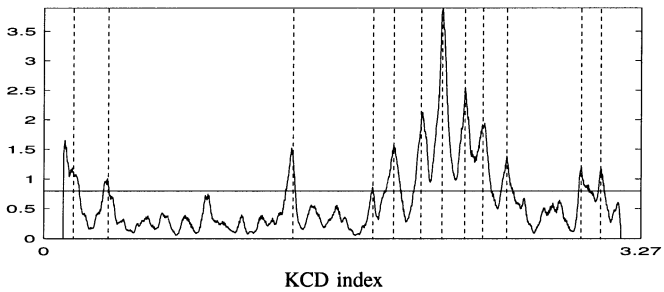

Fig. 13. First extract: The KCD index peaks over a certain threshold (horizontal solid line) whenever changes are detected in the music signal. Such changes are represented with a dotted vertical line. All changes are correctly detected; limited oversegmentation can be observed, with two false alarms. Listening to the piece of music confirms the correctness of the segmentation.

1) First Organ Extract: In the first extract, the mordent A—G—A is a difficult segmentation task: The first two notes alternate very quickly, and the final A is played with vibrato: The oscillation peaks may be detected as abrupt changes by the segmentation algorithm. Then, the four notes following the 32nd rest are 64th notes; hence, they are played very quickly, and the difficulty is even raised as the frequency gap between some of them (E and F, D and C sharp, and C sharp and D) is only a half step. The musician plays the same melodic line with both hands at the distance of one octave.

The KCD algorithm is tuned to obtain accurate segmentation results on the first extract. First, the music signal is downsampled by a factor 16, yielding a frequency bandwidth of

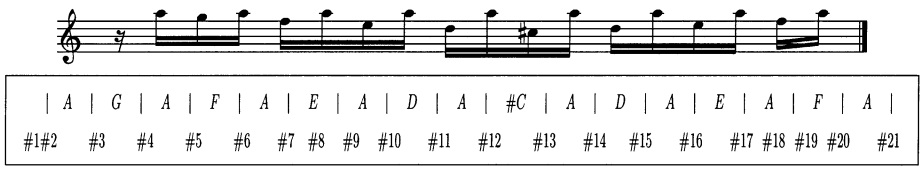

Fig. 14. (Top) Score for the second extract considered in this subsection. The music signal is 3.69 sec long and is part of the first measure of the Fugue in D minor by J. S. Bach. (Bottom) KCD results: Notes are represented in the upper row, with theoretical changes noted | ; the lower row corresponds to the (numbered) change detected by KCD.

[0, 1378.1] Hz; although cutting many partials, this resizes the frequency bandwidth to a smaller domain, which contains sufficient information for segmentation. The instantaneous energy of the downsampled signal is then set to one. Descriptors are extracted from the spectrogram of the preprocessed signal (represented in Fig. 12). The frequency smoothing window is Gaussian with a length of 161 points (58.4 ms). Each extracted descriptor is a subimage of the TFR made of 20 consecutive TFR columns. The TFR is computed over 512 frequency bins. Each vector of the training set is then scaled to the power 0.05: In the optimization problem of (5), large values are more likely to be preferred as they require a small weight in $\mathbf{w}$; rescaling the training vectors avoids this effect. The training set sizes are $m_1 = m_2 = 15$; the training set durations are thus 108.84 ms each, and training vector's durations are 7.26 ms. The SV kernel parameter is $\sigma = 75$ for the Gaussian kernel, and $\nu = 0.8$. We expect the TFR to be very perturbed by echoes or harmonics, which is why we make the algorithm more robust by specifying (with $\nu = 0.8$) that only 20% of each training set really is significant if one wishes to estimate $\mathcal{R}^{\mathcal{X}}_{\boldsymbol{x}_{t,1}}$ and $\mathcal{R}^{\mathcal{X}}_{\boldsymbol{x}_{t,2}}$.

Fig. 13 displays the KCD index with a threshold chosen heuristically: $\eta = 0.76$. All changes are correctly identified; see Fig. 11. Provided oversegmenting remains limited, it is not a bothersome issue: If we consider the segmentation as a preprocessing step that comes before such analysis as speech recognition or harmonic modeling [33], limited oversegmentation leads to small computational overload, whereas undersegmentation has far more undesirable effects. Change #5, in fact, detects the end of the echo of the second A. All the other changes are changes expected from the score presented in Fig. 11. The correctness of the detection is further confirmed by listening to the music signal.

2) Second Organ Extract: We now apply KCD to the second extract with the parameters tuned on the first extract to check the generalization ability of the algorithm. The second extract comes from the same recording, and it is a 3.69-sec extract of the first measure of the Fugue in D minor [36]. The corresponding score is presented in Fig. 14 together with KCD segmentation results. The KCD index is plotted in Fig. 15. Overall, changes are correctly detected. Change #13 is detected with some lag. Changes #8 and #9 detected by the KCD algorithm actually correspond to one change (similarly for changes #18–#19), the player actually ends one note after having started to play the following one. One may notice that in such a situation, it is hard to conclude even by listening to the extract, due to the fast alternation of notes and their

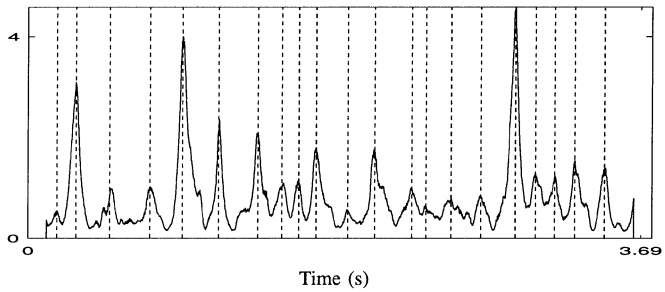

Fig. 15. KCD index for the test music sample. Overall, changes are correctly detected.

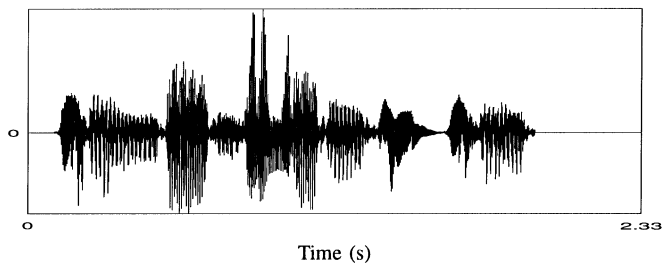

Fig. 16. Downsampled sax signal in the time domain; length of the signal is 2.33 sec.

trailing echos. Hence, except for one change detected with lag, all changes are detected properly.

We also implemented a GLR approach based on an AR model with order up to 40. In spite of many attempts to tune the GLR parameters, we did not not obtain convincing results, mainly because of 1) the AR model order needs to be very high to start obtaining some results, but this requires very long windows (larger than the duration of shortest notes) to enable accurate AR parameters estimation, and 2) computations are very long.

In this subsection, we applied KCD to the problem of music segmentation. Although dealing with a difficult music signal recorded in an echoic environment, the approach yields good results; even better results are obtained with easier signals, such as a classical piano solo or choruses played by the reed section of a jazz quintet.

3) Saxophone Extract: We now apply KCD to a saxophone extract plotted in Fig. 16, which was previously used to assess the quality of standard model-based approaches in [37] and cited in [38]. Segmentation results using both frequency-domain and time-domain approaches, including GLR, can be found in the reference provided.

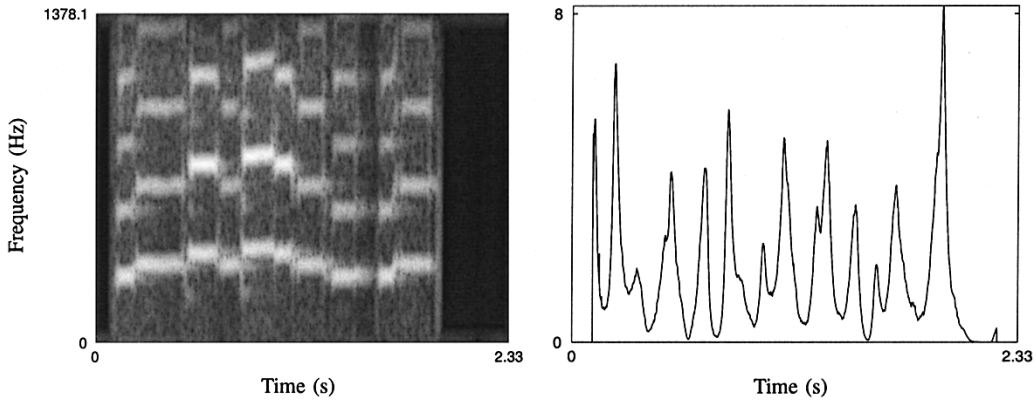

Fig. 17. Spectrogram of the sax signal is computed and (left) scaled to the power 0.05. Descriptors are extracted from this modified TFR. In the KCD index (right), all peaks correspond to a change in the sax signal. These results are obtained with the same parameters as for the organ signal in Section V-B1 and 2.

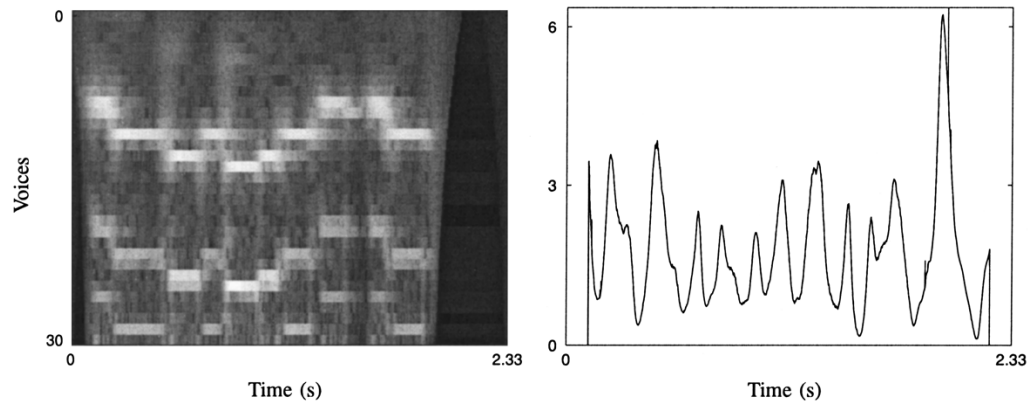

Fig. 18. Scalogram of the sax signal is computed and (left) scaled to the power 0.05. Descriptors are extracted from this modified Time-Scale Representation. In the KCD index (right), all peaks correspond to a change in the sax signal. Peaks are the same as with time-frequency descriptors.

We first apply KCD to this signal using time-frequency descriptors and wavelet-based descriptors. The latter are known to represent well transients.

Time-Frequency Descriptors: Descriptors are extracted from the spectrogram using exactly the same parameter tuning as for the organ signal presented in Sections V-B1 and 2 (this illustrates the good generalization ability of KCD). The parameters for the ν-SV single-class classifier are also kept unchanged. The change detection index is plotted in Fig. 17. True changes are correctly detected.

Wavelet Descriptors: Descriptors are extracted from a scalogram [30] (computed with a 50 points-18.1 ms-Morlet wavelet, over 64 voices) of the downsampled signal represented in Fig. 16. As above, it is then scaled to the power 0.05. Each training vector is made of 20 consecutive Scalogram columns, which correspond to 7.26 ms. The training set sizes are $m_1 = m_2 = 20$; the training set lengths thus are 108.84 ms. The SV kernel parameter is $\sigma = 75$ for the Gaussian kernel, and $\nu = 0.8$. The KCD index is plotted in Fig. 18. Again, true changes are correctly detected.

VI. Conclusion

In this paper, we proposed a novel Machine Learning approach to the abrupt change detection problem: The detection is achieved by means of a new dissimilarity measure defined in feature space yet computed in input space $\mathcal{X}$ using the kernel trick. This approach is robust to outliers and need not assume a statistical modeling of underlying distributions. Simulations showed that our KCD algorithm overperforms the standard Generalized Likelihood Ratio (GLR) approach. The application of KCD to music segmentation also showed good results. The perspectives for this work include further research on the application of KCD to the segmentation of music and speech signals; in particular, focusing on the design of descriptors specifically dedicated to these tasks should improve the results already obtained in this paper.

References

[1] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes—Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall, Apr. 1993.

[2] F. Gustafsson, "The marginalized likelihood ratio test for detecting abrupt changes," *IEEE Trans. Autom. Control*, vol. 41, no. 1, pp. 66–78, Jan. 1996.

[3] V. Kadirkamanathan, P. Li, M. Jaward, and S. Fabri, "Particle filtering-based fault detection in nonlinear stochastic systems," *Int. J. Syst. Sci.*, vol. 33, no. 4, pp. 259–265, Mar. 2002.

[4] E. Punskaya, C. Andrieu, A. Doucet, and W. J. Fitzgerald, "Bayesian curve fitting with applications to signal segmentation," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 747–758, Mar. 2002.

[5] J. Y. Tourneret, M. Doisy, and M. Lavielle, "Bayesian retrospective detection of multiple changepoints corrupted by multiplicative noise. Application to SAR image edge detection," *Signal Process.*, vol. 83, no. 9, pp. 1871–1887, Sept. 2003.

[6] H. Laurent and C. Doncarli, "Stationarity index for abrupt changes detection in the time-frequency plane," *IEEE Signal Process. Lett.*, vol. 5, no. 2, pp. 43–45, Feb. 1998.
[7] M. Grouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 886–902, Apr. 1998.
[8] E. Hitti and M.-F. Lucas, "Wavelet-packet basis selection for abrupt changes detection in multicomponent signals," in *Proc. EUSIPCO*, Rhodes, Greece, Sep. 1998.
[9] M. Seek, I. Magrin-Chagnolleau, and F. Bimbot, "Experiments on speech tracking in audio documents using Gaussian mixture modeling," in *Proc. IEEE ICASSP*, Salt Lake City, UT, 2001.
[10] F. Gustafsson, *Adaptive Filtering and Change Detection*. New York: Wiley, 2000.
[11] M. Davy and S. Godsill, "Detection of abrupt spectral changes using support vector machines, an application to audio signal segmentation," in *Proc. IEEE ICASSP*, Orlando, FL, May 2002.
[12] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[13] M. Davy, C. Doncarli, and J. Y. Tourneret, "Classification of chirp signals using hierarchical bayesian learning and MCMC methods," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 377–388, Feb. 2002.
[14] D.-T. Pham, "Fast algorithms for mutual information based independent componenent analysis," *IEEE Trans. on Signal Processing*, vol. 52, no. 10, pp. 2690–2700, Oct. 2002.
[15] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[16] A. Smola and B. Schölkopf, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.
[17] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[18] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[19] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, pp. 337–404, 1950.
[20] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, pp. 1207–1245, 2000.
[21] R. J. Vanderbei, "Loqo: An Interior Point Code for Quadratic Programming," Dept. Civil Eng. Oper. Res., Princeton Univ., Princeton, NJ, Tech. Rep. TR SOR-94-15, 1995.
[22] J. Weston, A. Elisseeff, G. Bakir, and F. Sinz, Version 1.3 of the Spider for Matlab, Sep. 2003.
[23] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philos. Trans. R. Soc.*, vol. A, no. 209, pp. 415–146, 1909.
[24] A. Gretton and F. Desobry, "On-line one-class ν-support vector machines, an application to signal segmentation," in *Proc. IEEE ICASSP*, Hong Kong, Apr. 2003.
[25] M. Davy, F. Desobry, and A. Gretton, "An online support vector machine for abnormal events detection," *Signal Process.*, submitted for publication.
[26] Online learning with kernels, 2003, to be published.
[27] F. Desobry, M. Davy, and C. Doncarli, An Online Kernel Change Detection Algorithm, IRCCyN, 2003. Internal Rep. 10.
[28] B. Schölkopf, S. Mika, C. Surges, P. Knirsch, K.-R. Miiller, G. Ratsch, and A. Smola, "Input space vs. feature space in kernel-based methods," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1000–1017, Sep. 1999.
[29] M. Davy, A. Gretton, A. Doucet, and P. Rayner, "Optimized support vector machines for nonstationary signal classification," *IEEE Signal Process. Lett.*, vol. 9, no. 12, pp. 442–445, Dec. 2002.
[30] P. Flandrin, *Time-Frequency/Time-Scale Analysis*. San Diego, CA: Academic, 1999.
[31] M. Davy, C. Doncarli, and G. Boudreaux-Bartels, "Improved optimization of time-frequency based signal classifiers," *IEEE Signal Process. Lett.*, vol. 8, no. 2, pp. 52–57, Feb. 2001.
[32] M. Davy, A. Gretton, A. Doucet, and P. Rayner, "Optimized support vector machines for nonstationary signal classification," *IEEE Signal Process. Lett.*, vol. 9, no. 12, pp. 442–445, Dec. 2002.
[33] M. Davy and S. Godsill, "Bayesian harmonic models for musical signal analysis," in *Proc. Seventh Valencia Int. Meet. Bayesian Statist.*, vol. 7, Tenerife, Spain, Jun. 2002.
[34] T. Jehan, "Musical Signal Parameter Estimation," M.S. Thesis, Elect. Eng. Comput. Sci. IFSIC, Univ. Rennes 1, Rennes, France, 1997. Center for New Music and Audio Technologies.
[35] R. Andre-Obrecht, "A new statistical approach for the automatic segmentation of continuous speech signals," in *IEEE Trans. Acoust., Speech Signal Process.*, vol. 36, Jan. 1988, pp. 29–40.
[36] J. S. Bach, "Toccata & Fuge BWV 565 in D minor," *Helmut Walcha, at the Organ of the St. Laurenskerk*, 1963.
[37] [Online]. Available: http://www.cnmat.berkeley.edu/tristan/Thesis/time-domain.html
[38] [Online]. Available: http://www.irisa.fr/sigma2/michele/segmentation.html

Frédéric Desobry was born in Algiers in 1979. He received the M.Eng. degree in 2000 and the Ph.D. degree in 2004, both from Ecole Centrale de Nantes, Nantes, France.

He is now a research associate in signal processing with the Department of Engineering, University of Cambridge, Cambridge, U.K. His research interests in machine learning include unsupervised learning, kernel methods, and pattern recognition.

Manuel Davy was born in Caen, France, in 1972. He received the ingénieur degree in electrical engineering in 1996 from Ecole Centrale de Nantes, Nantes, France, and the Ph.D. degree from the University of Nantes in 2000.

He is currently a chargé de recherches CNRS, LAGIS, Lille, France, where his research activity is centered around Kernel algorithms and Bayesian/Monte Carlo Methods for audio processing.

Christian Doncarli was born in Marseille, France. He received the Ph.D. degree in automatic control in 1977.

He is First Class Professor of Signal Procesing at Ecole Centrale de Nantes, Nantes, France. His current research interest is the area of nonstationary classification, using time-frequency/time scale representations as well as parametric methods. His preferred application field is audio (he plays Hammond Organ in an R&B band). He is in charge of the signal division of the National Group of Research on Information, Signal, Image, and viSion (ISIS).