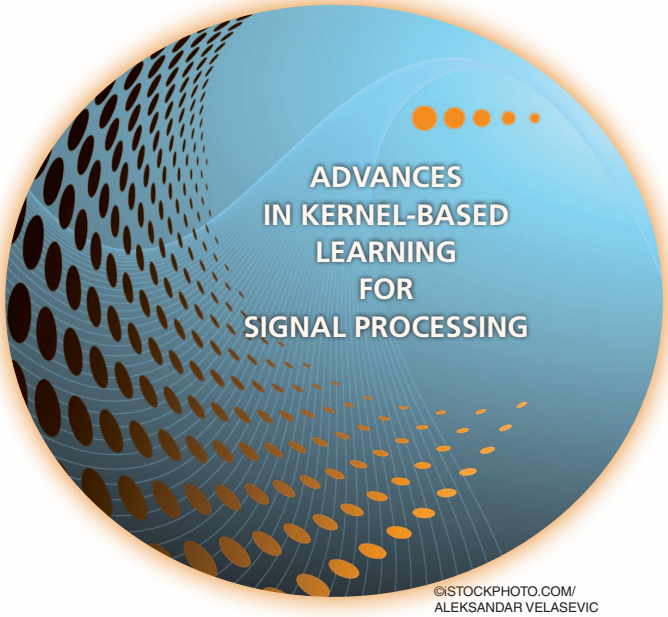
Guoru Ding, Qihui Wu, Yu-Dong Yao, Jinlong Wang, and Yingying Chen

Kernel-Based Learning for Statistical Signal Processing in Cognitive Radio Networks



Theoretical foundations, example applications, and future directions

Kernel-based learning (KBL) methods have recently become prevalent in many engineering applications, notably in signal processing and communications. The increased interest is mainly driven by the practical need of being able to develop efficient nonlinear algorithms, which can obtain significant performance improvements over their linear counterparts at the price of generally higher computational complexity. In this article, an overview of applying various KBL methods to statistical signal processing-related open issues in cognitive radio networks (CRNs) is presented. It is demonstrated that KBL methods provide a powerful set of tools for CRNs and enable rigorous formulation and effective solutions to both long-standing and emerging design problems.

INTRODUCTION AND MOTIVATION

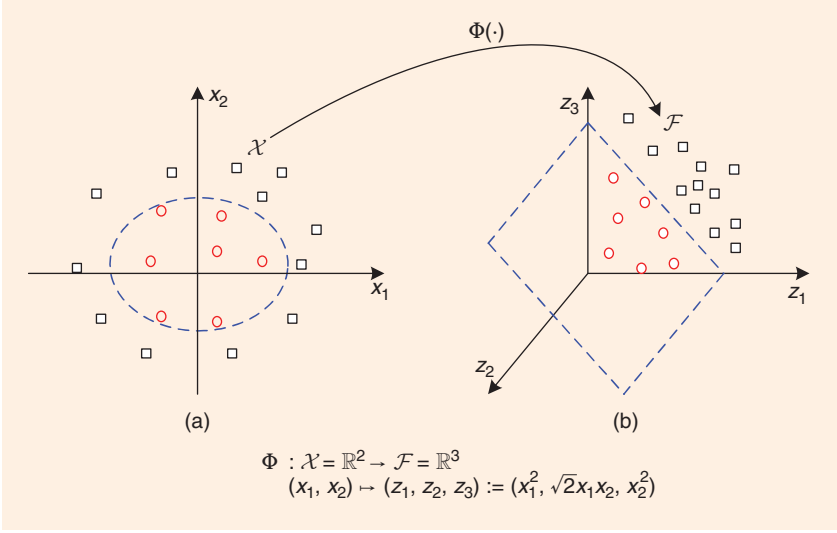
In the past decade, we have witnessed a dramatic growth in wireless communications due to the popularity of smartphones, mobile TVs, and many other wireless devices. The ever-increasing demand for high data rates in the face of limited spectral resources has motivated the introduction of cognitive radio (CR) [1]. The key idea behind CR is to allow secondary users (SUs) to exploit the spectral resources that have been licensed to primary users (PUs), but are underutilized, in a dynamic, opportunistic, and adaptive manner. An SU utilizes sensing and learning machines to be aware of his or her surrounding environment and adapts his or her internal states to statistical variations of the environment [2]. Although worldwide active efforts have been made for enabling CRNs in the past few years [3], many technical challenges still remain unsolved. A few related to statistical signal processing [4] include spectrum sensing, information fusion, irregular coverage boundary detection, robust signal classification, and spectrum occupancy online prediction.

KBL [5] exhibits the potential to provide effective solutions to many of these technical challenges. It has a number of attractive merits for statistical signal processing, e.g., nonlinear system designs, high-dimensional data processing, and superior computation efficiency [6]–[10]. Furthermore, there is a recent trend of applying machine learning to CRNs [11]–[14], especially an increasing interest in tailoring KBL methods to statistical signal processing-related issues in CRNs (e.g., [15] and [16]).

Digital Object Identifier 10.1109/MSP.2013.2251071
Date of publication: 12 June 2013

1053-5888/13/$31.00©2013IEEE

In light of these benefits, this article provides an overview of applying KBL methods to statistical signal processing in CRNs, which covers a wide range of topics related to both long-standing and recent-emerging design problems. The goal of this article is twofold. The first is to present theoretical foundations and main techniques in KBL, suitable for signal processing and communications communities. Special emphasis is placed on the nonlinear and high-dimensional signal processing capability of KBL methods. The second is to present approaches on how to apply the powerful set of KBL tools to resolve several challenging issues in CRNs. The focus is on the kernel part of various applications and, more importantly, we compare with the use of nonkernel methods to demonstrate both the advantages and the disadvantages of using KBL. We hope that this article with interdisciplinary perspectives will stimulate more interests in KBL theory and its applications in the signal processing and communications communities.



[FIG1] An introductory binary classification example [7]. By mapping data $\mathbf{x} = (x_1, x_2)$ in (a) two-dimensional (2-D) input space $\mathcal{X} = \mathbb{R}^2$ via nonlinear mapping $\Phi(\cdot)$ onto a (b) three-dimensional feature space $\mathcal{F} = \mathbb{R}^3$, the data become linearly separable.

NOTATION

Lowercase and uppercase boldface letters stand for column vectors and matrices, respectively. $(\cdot)^T$ and $(\cdot)^{-1}$ denote the transposition and inverse of a matrix, respectively. Diag $(x_1, \ldots, x_N)$ represents an $N \times N$ diagonal matrix with diagonal entries $x_1, \ldots, x_N$. $\mathbf{0}_N$ $(\mathbf{1}_N)$ denotes an $N \times 1$ vector of all zeros (ones) and $\mathbf{I}_N$ denotes an $N \times N$ identity matrix. $\|\mathbf{x}\|_2$ denotes the two-norm of a vector $\mathbf{x}$, $\|\mathbf{X}\|_F$ denotes the Frobenius norm of matrix $\mathbf{X}$, and $\langle \mathbf{x}, \mathbf{y} \rangle$ represents the inner product of $\mathbf{x}$ and $\mathbf{y}$.

KERNEL-BASED LEARNING: BASIC CONCEPTS, TOOLS, AND METHODS

In this section, we provide a brief review of KBL theory, showing its relevance in signal processing and communications applications. The basic concepts, tools, and kernel methods discussed in this section will be applied to research issues in CRNs as presented in the sections "Applications of Kernel-Based Learning in CRNs" and "Additional Applications and Future Directions."

CONCEPTS

There is a long history of constructing machines capable of learning from data within the statistical framework. Johann Carl Friedrich Gauss proposed the idea of least squares regression in the 18th century, while Ronald Aylmer Fisher's approach to classification in the 1930s provides the starting point for most analyses and methods [17]. Using hypotheses that form linear combinations of the input variables, linear learning machines for classification and regression problems dominated the research until the 1960s, when the limited computational power of them in dealing with many complex real-world problems was highlighted [18]. KBL, which has gained considerable popularity during the last 15 years, has offered a promising solution to increase the computational capability based on a breakthrough in the design of efficient nonlinear learning algorithms.

Specifically, in KBL theory, data $\mathbf{x}$ in the input space $\mathcal{X}$ is projected onto a potentially much higher dimensional feature space $\mathcal{F}$ via a nonlinear mapping Φ as follows:

$$\Phi : \mathcal{X} \to \mathcal{F}, \quad \mathbf{x} \mapsto \Phi(\mathbf{x}). \tag{1}$$

For a given learning problem, one now works with the mapped data $\Phi(\mathbf{x}) \in \mathcal{F}$ instead of $\mathbf{x} \in \mathcal{X}$ (see Figure 1 as an example). The data in the input space can be projected onto different feature spaces with different mappings. The diversity of feature spaces gives us more choices to gain better performance, while in practice the choice itself of a proper mapping for any given real-world problem may generally be nontrivial.

Fortunately, originally proposed in [18], the kernel trick provides an elegant mathematical means to construct powerful nonlinear variants of most well-known statistical linear techniques, without knowing the mapping Φ explicitly. Indeed, one only needs to replace the inner product operator of a linear technique with an appropriate kernel k (i.e., a positive semidefinite symmetric function), which arises as a similarity measure that can be thought of as an inner product between pairs of data in the feature space

$$k(\mathbf{x}_i, \mathbf{x}_j) := \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{F}}, \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}. \tag{2}$$

Table 1 lists the most widely used kernels, which can be divided into two categories: projective kernels (functions of inner product) and radial kernels (functions of distance). These kernels implicitly map the data onto high-dimensional spaces, even infinite-dimensional spaces.

[TABLE 1] COMMONLY USED KERNELS WITH PARAMETERS $c > 0$, $p \in \mathbb{N}+$, AND $\sigma > 0$.

KERNELS (PROJECTIVE)	EXPRESSIONS	KERNELS (RADIAL)	EXPRESSIONS
MONOMIAL	$(\langle \mathbf{x}_i, \mathbf{x}_j \rangle)^d$	GAUSSIAN	$\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/2\sigma^2)$
POLYNOMIAL	$(\langle \mathbf{x}_i, \mathbf{x}_j \rangle + c)^d$	LAPLACIAN	$\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2/2\sigma^2)$
EXPONENTIAL	$\exp(\langle \mathbf{x}_i, \mathbf{x}_j \rangle/2\sigma^2)$	MULTIQUADRATIC	$\sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + c}$
SIGMOID (PERCEPTRON)	$\tanh(\langle \mathbf{x}_i, \mathbf{x}_j \rangle/\sigma + c)$	INVERSE MULTIQUADRATIC	$1/\sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + c}$

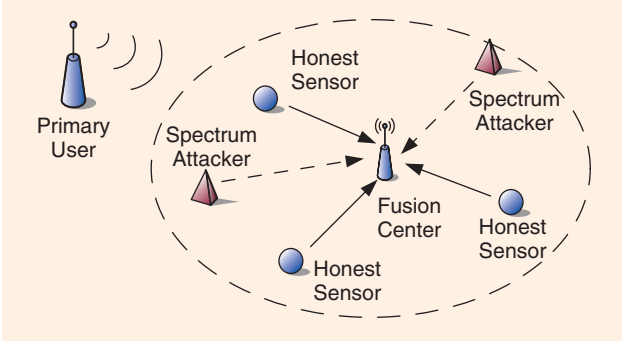
TOOLS

The introduction of kernels greatly increases the computational capability by constructing nonlinear learning machines for data in the input space, while retaining the underlying linearity in the feature space that will ensure that learning remains tractable. The increased computational capability, however, leads to new technical challenges, such as numerical instabilities and inferior generalization performance. The literature provides a few powerful mathematical tools (e.g., regularization, generalization, and optimization) to tackle these challenges, and we refer interested readers to [5]–[7] as entry points, and [8]–[10] for more advanced results.

Briefly, the regularization theory serves as a tool to control the complexity of the model used for explaining the training data, by mainly utilizing objective functions that involve both an empirical loss term and a regularization term [10]. The generalization theory provides theoretical insights on learning machines from a statistical point of view by introducing several complexity measures, such as the Vapnik–Chervonenkis (VC) dimension, and illustrating how to derive bounds on the generalization error [6]. The optimization theory, notably the convex optimization theory, can be seen as a mathematical tool concerned with characterizing practical solutions of various KBL problems and developing effective algorithms for finding the solutions [7].

KERNEL METHODS

Kernel methods are counterparts of linear methods that are implemented in feature spaces. Pioneered by support vector machines (SVMs) [6], kernel methods have recently gained wide popularity, mainly due to the theoretical guarantees regarding performance and powerful nonlinear algorithms. Besides SVMs, the most well-known kernel methods include kernel Fisher discriminant analysis (FDA) [19], kernel K-means clustering [20], kernel principal component analysis (PCA) [5], and kernel online learning [9]. These methods have shown practical relevance for many signal processing-related engineering applications. We will discuss some of these KBL methods along with presentations of example applications in statistical signal processing-related open issues in CRNs.



[FIG2] An illustrative example of CSS in the presence of spectrum attackers.

APPLICATIONS OF KERNEL-BASED LEARNING IN CRNs

APPLICATION 1: KERNEL CLUSTERING FOR ATTACKER DETECTION IN COLLABORATIVE SPECTRUM SENSING

Spectrum sensing is a fundamental issue in CRNs, which detects the presence ($\mathcal{H}_1$) or absence ($\mathcal{H}_0$) of a licensed/primary user signal over a radio frequency band [21]. To combat the impacts of multipath fading, shadowing, and receiver uncertainty, collaborative spectrum sensing (CSS) among multiple spectrum sensors has been proposed as an effective method by exploiting spatial diversity [22], [23]. However, due to the openness of low-layer protocol stacks in CR devices, many security threats in CSS have been raised (see, e.g., [24]–[26] and the recent reviews [27] and [28]). As a case study, we propose to apply robust clustering algorithms to distinguish spectrum attackers from honest sensors in CSS and show the advantages and the disadvantages of a kernel clustering algorithm over its nonkernel counterpart.

PROBLEM STATEMENT

As shown in Figure 2, we consider a CRN with N spectrum sensors and one fusion center (FC) collaboratively detecting the presence of a primary signal over a given channel, which is modeled as an ON/OFF renewal process. An ON period represents a time duration in which the primary signal is present and an OFF period is a time duration in which the primary signal is absent and the CRN is allowed to utilize the channel for its data transmission. Each spectrum sensor captures the signal periodically, and reports its "original" (from an honest sensor) or "false" (from a spectrum attacker) observation to the FC. The FC fuses the collected sensing reports and makes a decision on the presence ($\mathcal{H}_1$) or absence ($\mathcal{H}_0$) of the primary signal. The goal of a spectrum attacker, which injects attack data into its original observation, is to mislead the FC to make a wrong decision.

Specifically, if an energy detector is used, the sensing report of spectrum sensor n in the tth sensing period can be given as [29]

$$x_{nt} = \underbrace{P_{nt} \cdot \mathbf{1}_{\{\mathcal{H}_1\}} + N_0 + E_{nt}}_{\text{energy detector output } (T_{nt})} + o_{nt}, \quad n = 1, 2, \ldots, N; t = 1, 2, \ldots, p, \tag{3}$$

where $\mathbf{1}_{\{\cdot\}}$ is an indicator function, T_{nt} is the output of the energy detector including the received primary signal power P_{nt}, the noise power N_0, and the Gaussian measurement error E_{nt} with zero mean and variance $\sigma_{E_{nt}^2} = (P_{nt} \cdot \mathbf{1}_{\{\mathcal{H}_1\}} + N_0)^2 / N_{sam}$. N_{sam} is the number of samples in each sensing period. Notably, o_{nt} represents the attack data, which is zero if spectrum sensor n is a honest sensor and nonzero if it is a spectrum attacker. There are two types of harmful attacks:

1) In Type-1, an attacker injects positive attack data ($o_{nt} > 0$), which aims to increase the false-positive rate (classifying the absence of a primary signal as present).
2) In Type-2, an attacker injects negative attack data ($o_{nt} < 0$), which aims to increase the false-negative rate (classifying the presence of a primary signal as absent).

Generally, a Type-1 attack results in lower spectrum utilization for SUs, and a Type-2 attack results in more interference to the PU. Therefore, to mitigate wrong decisions, the FC must be able to correctly identify any spectrum attackers before fusing the sensing reports to make a final decision.

Comprehensive reviews of related studies on this topic have been recently given in [27] and [28]. Different from the existing studies, two powerful clustering algorithms in the field of data mining will be applied at the FC to detect multiple attackers simultaneously, without a priori information about the attackers' strategy. The basic idea is as follows: After collecting the sensing reports from all spectrum sensors in p successive sensing periods, the FC seeks a partition of a set of p-dimensional vectors $\mathfrak{X} := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with the aid of analyzing the similarity among the vectors, such that the honest sensors and the spectrum attackers are grouped into different clusters. Note that $\mathbf{x}_n := (x_{n1}, x_{n2}, \ldots, x_{np})^T$ is a sensing report vector of sensor n in p successive sensing periods.

SPECTRUM ATTACKER DETECTION USING K-MEANS CLUSTERING

Among the algorithms that cluster data represented by vectors, K-means clustering (KMC) is one of the most popular schemes with well-documented merits [30], [31]. Under the framework of KMC, the data model in (3) can be rewritten in a vector form as

$$\mathbf{x}_n = \mathbf{m}_c + \mathbf{v}_n + \mathbf{o}_n, \quad n = 1, 2, .., N, \tag{4}$$

where $\mathbf{m}_c$ denotes the cluster centroid of the honest sensors, $\mathbf{v}_n := (v_{n1}, v_{n2}, \ldots, v_{np})^T$ is a Gaussian vector capturing the deviation of $\mathbf{x}_n$ from the $p \times 1$ centroid vector $\mathbf{m}_c$, and $\mathbf{o}_n := (o_{n1}, o_{n2}, \ldots, o_{np})^T$ denotes the attack data vector. One practical and common assumption in the literature is that the presence of spectrum attackers is sparse [28] and most of the $\mathbf{o}_n$s in (4) are zero. Consequently, the unknowns $\{\mathbf{m}_c, \mathbf{o}_n\}$ can now be estimated using the least squares (LS) approach, which can be formulated as

$$\underset{\{\mathbf{m}_c, \mathbf{o}_n\}}{\text{minimize}} \sum_{n=1}^{N} \| \mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n \|_2^2, \quad \text{subject to} \sum_{n=1}^{N} \mathbf{1}_{\{\|\mathbf{o}_n\|_2 > 0\}} \le M, \tag{5}$$

where the constraint in (5) means that the number of attackers is no larger than M. This optimization problem is well known as NP-hard [30]. To efficiently resolve the problem, a suboptimal yet practical algorithm along the lines of [20] will be developed. Consider first the Lagrangian form of (5)

$$\underset{\{\mathbf{m}_c, \mathbf{o}_n\}}{\text{minimize}} \sum_{n=1}^{N} \| \mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n \|_2^2 + \lambda \sum_{n=1}^{N} \| \mathbf{o}_n \|_2, \tag{6}$$

where λ is an attacker-controlling parameter with respect to M. The pseudo-l_0-norm $\| \mathbf{o} \|_0 := \sum_{n=1}^{N} \mathbf{1}_{\{\|\mathbf{o}_n\|_2 > 0\}}$ in (5) was surrogated by its convex l_1-norm $\| \mathbf{o} \|_1 := \sum_{n=1}^{N} \| \mathbf{o}_n \|_2$ in (6), following the successful compressive sensing paradigm [32]. Now the problem in (6) is convex in both $\{\mathbf{m}_c\}$ and $\{\mathbf{o}_n\}$, but jointly nonconvex. The per-variable convexity motivates a KMC solver as shown next in Algorithm 1:

Algorithm 1: Attacker Detection Using KMC

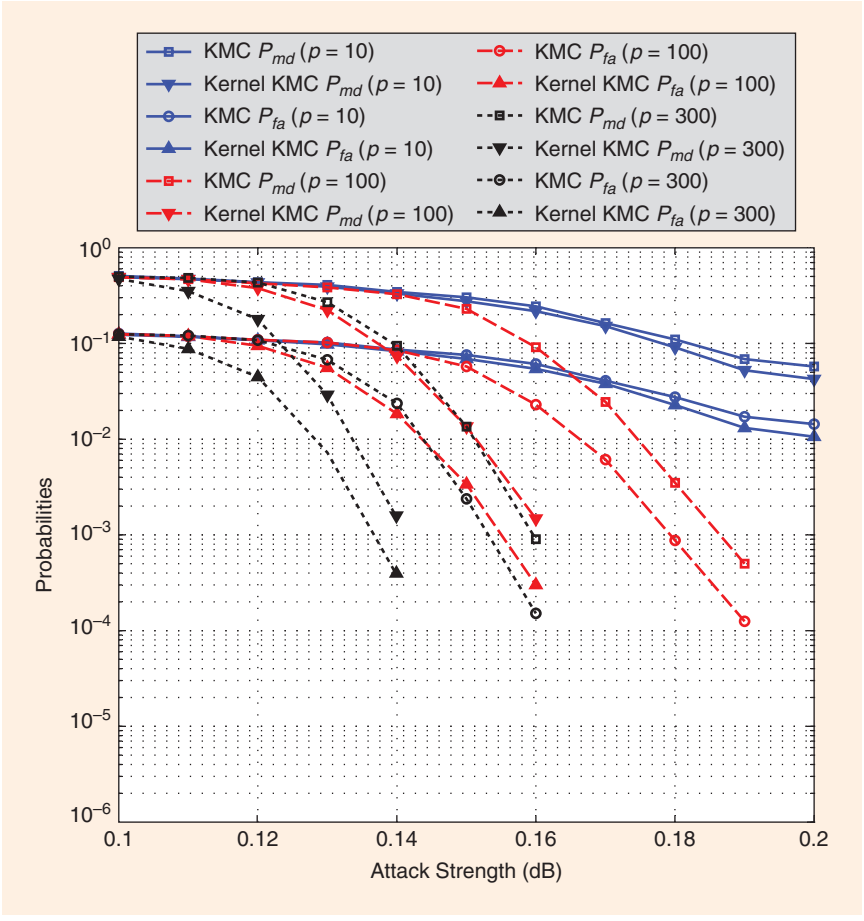
1) **Initialization:** Input sensing data matrix $\mathbf{X} := [\mathbf{x}_1\ \mathbf{x}_2 \ldots \mathbf{x}_N]$, select λ using a grid search technique as described in [33] to satisfy $\sum_{n=1}^{N} \mathbf{1}_{\{\|\mathbf{o}_n^{(t)}\|_2 > 0\}} \le M$, and set $\mathbf{O}^{(0)} := [\mathbf{o}_1^{(0)}\ \mathbf{o}_2^{(0)} \ldots \mathbf{o}_N^{(0)}]$ to zero.
2) **for** $t = 1, 2, \ldots$ **do**
3) Update the $p \times 1$ centroid matrix $\mathbf{M}^{(t)} = (\mathbf{X} - \mathbf{O}^{(t-1)})\mathbf{1}_N / N = \mathbf{m}_c^{(t)}$.
4) Update the $p \times N$ attack data matrix $\mathbf{O}^{(t)}$ via $\mathbf{o}_n^{(t)} = \max\{\mathbf{r}_n^{(t)}[1 - (\lambda/2\|\mathbf{r}_n^{(t)}\|_2)], \mathbf{0}_p\}$, where $\mathbf{r}_n^{(t)} := \mathbf{x}_n - \mathbf{m}_c^{(t)}, \forall n = 1, 2, \ldots N$.
5) **end for**

The for-loop is terminated when the tth iteration satisfies $\|\mathbf{M}^{(t)} - \mathbf{M}^{(t-1)}\|_F / \|\mathbf{M}^{(t)}\|_F \le \epsilon_s$, where ϵ_s is a small positive threshold (e.g., 10^{-6}).

SPECTRUM ATTACKER DETECTION USING KERNEL KMC

As shown in (5) and (6), with the Euclidean distance as a similarity measure, the KMC algorithm favors clustering the underlying data vectors that are of spherical shape and linearly separable. However, in the considered problem, the set of p-dimensional sensing report vectors $\mathfrak{X} := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is neither of standard spherical shape nor linearly separable, since the data distributions for the presence ($\mathcal{H}_1$) and absence ($\mathcal{H}_0$) of the primary signal are heterogeneous as shown in (3). To bypass this hurdle, an effective approach is to design a kernelized version of the KMC algorithm, where the sensing report vector $\mathbf{x}_n \in \mathbb{R}^p$ is mapped to a higher (even infinite) dimensional feature space $\mathcal{F}$ via a nonlinear function $\mathbf{\Phi} : \mathbb{R}^p \mapsto \mathcal{F}$. Thus, linearly separable partition in feature space $\mathcal{F}$ enables nonlinearly separable partition in the original data space $\mathbb{R}^p$.

Following the kernel method in [18], the KMC algorithm can be kernelized without knowing the mapping $\mathbf{\Phi}$ explicitly. The key idea is to replace the inner product operations, on the input sensing report vectors, with an appropriate kernel function (see Table 1). Suppose that there exists an $N \times N$ matrix $\mathbf{A}$



[FIG3] Attacker detection performance of KMC and kernel KMC algorithms under different attack strengths. In this simulation, $N = 10$ and $M = 2$. The probability P_{md} characterizes the cases that an attacker is incorrectly classified as an honest sensor and the P_{fa} measures the cases that an honest sensor is incorrectly classified as an attacker.

such that the $p \times N$ attack data matrix satisfies $\mathbf{O} = \mathbf{XA}$, $\mathbf{X} := [\mathbf{x}_1\ \mathbf{x}_2 \ldots \mathbf{x}_N]$ is the $p \times N$ sensing data matrix. Then, we can rewrite the $p \times 1$ centroid matrix as $\mathbf{M} = (\mathbf{X} - \mathbf{O})\mathbf{1}_N/N = \mathbf{XB}$, where the $N \times N$ matrix $\mathbf{B} := (\mathbf{I}_N - \mathbf{A})\mathbf{1}_N/N$. Neglecting the mathematical derivations, the kernel KMC solver is summarized next in Algorithm 2:

Algorithm 2: Attacker Detection Using Kernel KMC

1) **Initialization:** Input sensing data matrix $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \ldots \mathbf{x}_N]$, select λ using a grid search technique as described in [33] to satisfy $\sum_{n=1}^{N} \mathbf{1}_{\{\|\mathbf{o}_n^{(t)}\|_2 > 0\}} \le M$ and calculate the $N \times N$ kernel matrix $\mathbf{K}$ with entries $\mathbf{K}_{i,j} := \mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)$. Initialize $\mathbf{A}^{(0)}$ to zero.
2) **for** $t = 1, 2, \ldots$ **do**
3) Update $\mathbf{B}^{(t)} = (\mathbf{I}_N - \mathbf{A}^{(t-1)})\mathbf{1}_N/N$.
4) Update $\Delta^{(t)} = \mathbf{I}_N - \mathbf{B}^{(t)}(\mathbf{1}_N)^T$, where the nth column of $\Delta^{(t)}$ is denoted as $\mathbf{\Gamma}_n^{(t)}$.
5) Update $\mathbf{A}^{(t)}$, where the nth column of $\mathbf{A}^{(t)}$ is given by $\mathbf{a}_n^{(t)} = \max\{\mathbf{\Gamma}_n^{(t)}[1 - (\lambda/2\|\mathbf{\Gamma}_n^{(t)}\|_{\mathrm{K}})], 0_p\}$.
6) **end for**
7) Calculate $\mathbf{M}^{(t)} = \mathbf{XB}^{(t)}$.
8) Calculate $\mathbf{O}^{(t)} = \mathbf{XA}^{(t)}$.

Note that the key kernel operation of the kernel KMC solver lies in the fifth line, i.e., $\|\mathbf{\Gamma}_n^{(t)}\|_{\mathrm{K}} := \sqrt{(\mathbf{\Gamma}_n^{(t)})^T \mathbf{K} \mathbf{\Gamma}_n^{(t)}}$. Furthermore, to enable a fair comparison of the attacker detection performance, the convergence condition of the kernel KMC iterations is set to be the same as the KMC algorithm.

MAIN RESULTS AND INSIGHTS

1) *Kernel parameter selection*: The selection of the optimal kernel parameters is very important in KBL methods, which is often stated as an open problem [5]–[7]. In this article, we focus on the scenario that the CRN environment is stationary or quasistationary, which is a case commonly considered in the CRN research community. For this case, kernel parameters are determined by exhaustive search or cross-validation in the training period. Specifically, in this application, the Gaussian kernel is used and the kernel parameter σ^2 is set as the variance estimate of the entire data set $\mathbf{X}$ as described in [34].

2) *Detection performance comparison*: In this application, both KMC and kernel KMC are, in essence, batch or offline algorithms. The algorithms are implemented after p successive sensing periods. It is observed in Figure 3 that, for both clustering algorithms, an increasing dimensionality p of the sensing report vectors generally yields better detection performance; and the kernel KMC algorithm yields better detection performance than the KMC algorithm, which mainly benefits from its capability of identifying nonlinearly separable clusters.

3) *Computational complexity analysis*: After careful evaluation, it is found that

- For the KMC algorithm, it performs $\mathcal{O}(Np)$ scalar operations per iteration and requires storing $\mathcal{O}(Np)$ scalar variables.
- For the kernel KMC algorithm, it is noted that the $p \times N$ sensing data matrix $\mathbf{X}$ is used when calculating the kernel matrix $\mathbf{K}$, the final centroid matrix $\mathbf{M}$, and the final attack data matrix $\mathbf{O}$, which are all performed outside the for-loop. Thus, it requires $\mathcal{O}(N^3)$ operations per-iteration and $\mathcal{O}(N^2)$ spaces. Consequently, in the high-dimensional data regime (e.g., $p = 300 > N^2 = 10^2$ in Figure 3), kernel KMC not only improves the attacker detection performance, but also offers processing and memory savings.

APPLICATION 2: KERNEL FISHER DISCRIMINANT ANALYSIS FOR RANDOM PU NETWORK DETECTION

In general, Fisher discriminant analysis (FDA) addresses the following question: Given a data set, say, with two classes, which is the best feature or feature set to discriminate the two classes [17]? We show its application in the multiuser information fusion problem. We will first discuss the use of the well-known linear FDA in the issue of random primary user network detection in CRNs, and then show the performance improvement from using the kernel FDA.

PROBLEM STATEMENT

The problem of random PU network detection was first addressed in [35]. As shown in Figure 4, the primary transmitters (PTxs) are randomly distributed in a 2-D space, following a Poisson point process. Each primary receiver (PRx) is uniformly distributed around its PTx in a small disc area. An SU located at the origin tries to detect the presence of any PRx within a given disc detection area $\mathcal{A}$, with the help of N_{sc} secondary cooperators (SCs). If there is no PRx within $\mathcal{A}$, the SU can transmit its signal safely. However, if there is at least one PRx, the SU must cease its transmission to avoid inflicting interference to the primary reception. Let Y be a random variable that indicates whether there is any PU receiver within $\mathcal{A}$. The goal is to correctly distinguish the PRx-present case ($Y = 1$) from the PRx-absent case ($Y = 0$).

Due to the lack of cooperation from the PU network, the SU has no prior knowledge of the location information of the PRxs. Therefore, the SU has to determine Y by analyzing the sensing results of itself and its SCs, i.e., the vector $\mathbf{x} := (x_0, x_1, \ldots, x_{N_{sc}})^T$, where the entry x_0 and x_i, $i = 1, 2, \ldots, N_{sc}$ denote the total energy received by the SU and the ith SCs, respectively. The problem is in essence a multiuser information fusion problem—that is, how do you efficiently fuse the sensing results from multiple users to detect the presence of any PRx within a given detection area as accurately as possible?

LIKELIHOOD-RATIO TEST DETECTOR

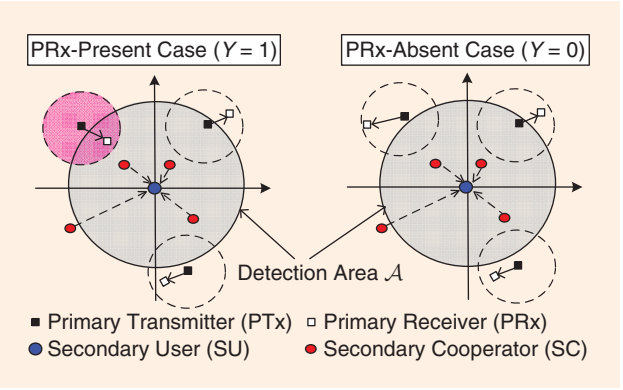
The likelihood-ratio test (LRT) has been shown to be the optimal fusion rule according to the Neyman–Pearson criterion, which is obtained by performing the following likelihood ratio testing:

$$\frac{f_{\mathbf{x}|Y=1}(\mathbf{x})}{f_{\mathbf{x}|Y=0}(\mathbf{x})} \underset{Y=0}{\overset{Y=1}{\gtrless}} T_{lrt}, \tag{7}$$

where $f_{\mathbf{x}|Y=1}$ ($f_{\mathbf{x}|Y=0}$) is the probability distribution function (pdf) of $\mathbf{x}$ given that $Y = 1$ ($Y = 0$), T_{lrt} is the decision threshold. For the LRT detector, we require the calculation of the pdf of $\mathbf{x}$, given that $Y = 1$ and $Y = 0$, which is mathematically intractable. Therefore, the design of a detector with low complexity and good performance is needed in practice.

LINEAR FUSION-BASED COOPERATIVE DETECTION USING LINEAR FDA

To design a simple and efficient detector, a linear fusion rule is usually adopted, which can be given as



[FIG4] An illustration of random primary user network detection. The PTxs are randomly distributed in a 2-D space. Each PRx is uniformly distributed around its PTx in a small disc area. An SU located at the origin tries to detect the presence of any PRx within a given disc detection area $\mathcal{A}$, with the help of several SCs.

$$D = \mathbf{w}^T\mathbf{x} = \sum_{n=0}^{N_{sc}} w_n x_n \underset{Y=0}{\overset{Y=1}{\gtrless}} T_{\text{linear}}, \tag{8}$$

where $\mathbf{w} := (w_1, w_2, \ldots, w_{N_{sc}})^T$ is the vector of the linear coefficients. Now, the problem is how to determine the value of the linear coefficient vector $\mathbf{w}$. The optimal $\mathbf{w}$ is the one that maximizes the detection probability $\Pr\{\mathbf{w}^T\mathbf{x} \geq T_{\text{linear}} \,|\, Y = 1\}$, while minimizing the false alarm probability $\Pr\{\mathbf{w}^T\mathbf{x} \geq T_{\text{linear}} \,|\, Y = 0\}$. However, considering the randomness in the PU network model, a closed-form expression of the optimal linear coefficients is not easy to obtain. To circumvent this problem, [35] uses the linear FDA to determine a set of suboptimal linear coefficients.

The linear FDA finds a linear coefficient vector $\mathbf{w}^{L-FDA}$ that most clearly separates the different cases by assigning a higher linear coefficient to a more significant sensing result. Specifically, linear FDA maximizes the so-called *Rayleigh coefficient* [17]

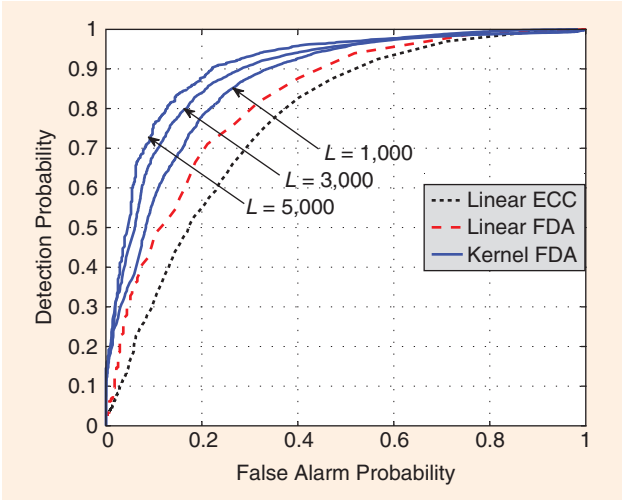
$$J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} \tag{9}$$

with respect to $\mathbf{w}$ to obtain $\mathbf{w}^{L-FDA}$, i.e., $\mathbf{w}^{L-FDA} = \arg\max_{\mathbf{w}} J(\mathbf{w})$, where $\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_0)(\mathbf{m}_1 - \mathbf{m}_0)^T$ and $\mathbf{S}_W = \sum_{Y=0,1}\sum_{x\in\mathcal{J}_Y} (\mathbf{x} - \mathbf{m}_Y)(\mathbf{x} - \mathbf{m}_Y)^T$ are the between- and within-class scatter matrices respectively, $\mathbf{m}_1$ ($\mathbf{m}_0$) and $\mathcal{J}_1$ ($\mathcal{J}_0$) denote the mean and the sensing result set given that $Y = 1$ ($Y = 0$). According to [17], the optimal linear FDA coefficient vector can be derived as

$$\mathbf{w}^{L-FDA} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_0). \tag{10}$$

NONLINEAR FUSION-BASED COOPERATIVE DETECTION USING KERNEL FDA

Linear FDA is a linear technique in nature, and it is often limited and inadequate to derive more efficient and general information fusion algorithms. To better express the discriminant and thus improve the detection performance, we introduce a kernelized version of linear FDA to reformulate the above problem.



[FIG5] The receiver operating characteristic (ROC) curves for the linear ECC [29], the linear FDA [35], and the proposed kernel FDA-based cooperative detection algorithms. The Gaussian kernel is used and the kernel parameter σ^2 is set as the variance estimate of the training sample data set $[\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_L]$ as described in [34]. The radius of the SU detection area is 300 m, and the transmission range of the PTx is 150 m. The SU network consists of one SU at the origin and $N_{sc} = 4$ SCs located 100 m, 200 m, 300 m, and 400 m away from the origin, respectively. The other experiment configurations are the same as in [35].

The key idea of kernel FDA is to solve a linear fusion problem in a kernel feature space $\mathcal{F}$, thereby yielding a nonlinear fusion rule in the original input space. Let Φ be the nonlinear mapping as defined in (1), the discriminant coefficient in the feature space $\mathcal{F}$ can be obtained by maximizing

$$J(\mathbf{w}^{\Phi}) = \frac{(\mathbf{w}^{\Phi})^T \mathbf{S}_B^{\Phi} \mathbf{w}^{\Phi}}{(\mathbf{w}^{\Phi})^T \mathbf{S}_W^{\Phi} \mathbf{w}^{\Phi}}, \tag{11}$$

where $\mathbf{w}^{\Phi} \in \mathcal{F}$, $\mathbf{S}_B^{\Phi} = (\mathbf{m}_1^{\Phi} - \mathbf{m}_0^{\Phi})(\mathbf{m}_1^{\Phi} - \mathbf{m}_0^{\Phi})^T$ and $\mathbf{S}_W^{\Phi} = \sum_{Y=0,1} \sum_{\mathbf{x} \in \mathcal{J}_Y} (\Phi(\mathbf{x}) - \mathbf{m}_Y^{\Phi})(\Phi(\mathbf{x}) - \mathbf{m}_Y^{\Phi})^T$. Note that $\mathbf{m}_1^{\Phi}$ ($\mathbf{m}_0^{\Phi}$) is the mean vector of the sensing results in feature space given that $Y = 1$ ($Y = 0$).

To build the kernel FDA [19] in the feature space $\mathcal{F}$, we first need to reformulate (11) in terms of inner products of input data, which can then be replaced by kernels. From [19], we know that any solution $\mathbf{w}^{\Phi}$ must lie in the span of all training samples $\{\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_L)\}$ in $\mathcal{F}$. Therefore, we express $\mathbf{w}^{\Phi}$ as

$$\mathbf{w}^{\Phi} = \sum_{i=1}^{L} v_i \Phi(\mathbf{x}_i), \tag{12}$$

and we further have

$$\begin{aligned}(\mathbf{w}^{\Phi})^T \mathbf{m}_Y^{\Phi} &= \frac{1}{L_Y} \sum_{j=1}^{L} \sum_{i=1}^{L_Y} v_j \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_i^Y) \rangle \\ &= \frac{1}{L_Y} \sum_{j=1}^{L} \sum_{i=1}^{L_Y} v_j \mathbf{k}(\mathbf{x}_j, \mathbf{x}_i^Y) = \mathbf{v}^T \mathbf{k}_Y, \forall Y \in \{0,1\},\end{aligned} \tag{13}$$

where $\mathbf{x}_i^Y$ denotes the ith training sample vector of sensing results in the original data space given that $Y \in \{0,1\}$, $\mathbf{v} := (v_1, v_2, \ldots, v_L)^T$, and $(\mathbf{k}_Y)_j := (1/L_Y) \sum_{i=1}^{L_Y} \mathbf{k}(\mathbf{x}_j, \mathbf{x}_i^Y)$.

Substituting (12) and (13) into (11), the optimization problem can be rewritten as

$$\mathbf{v}^{K-FDA} = \arg\max_{\mathbf{v}} J(\mathbf{v}) := \frac{\mathbf{v}^T \tilde{\mathbf{S}}_B \mathbf{v}}{\mathbf{v}^T \tilde{\mathbf{S}}_W \mathbf{v}}, \tag{14}$$

where $\tilde{\mathbf{S}}_B = (\mathbf{k}_1 - \mathbf{k}_0)(\mathbf{k}_1 - \mathbf{k}_0)^T$ and $\tilde{\mathbf{S}}_W = \sum_{Y=0,1} \mathbf{K}_Y (\mathbf{I}_{L_Y} - \mathbf{1}_{L_Y} \mathbf{1}_{L_Y}^T / L_Y) \mathbf{K}_Y^T$. $\mathbf{K}_Y$ is an $L \times L_Y$ kernel matrix with $(\mathbf{K}_Y)_{nm} := \mathbf{k}(\mathbf{x}_n, \mathbf{x}_m^Y)$.

To maximize $J(\mathbf{v})$ with respect to $\mathbf{v}$, one could obtain $\mathbf{v}^{K-FDA}$ by finding the leading eigenvector of $\tilde{\mathbf{S}}_W^{-1} \tilde{\mathbf{S}}_B$ [19]. Although there exist many efficient eigenvalue problem solvers, one problem remains: for a large training sample size L, the matrices $\tilde{\mathbf{S}}_B$ and $\tilde{\mathbf{S}}_W$ become large and the solutions $\mathbf{v}^{K-FDA}$ are nonsparse. One way of dealing with this problem is to reformulate kernel FDA as a convex quadratic programming problem (refer to [7] for details).

Finally, for the problem of random PU network detection, a nonlinear fusion-based detector, using kernel FDA, can be expressed as

$$\langle \mathbf{w}^{\Phi}, \Phi(\mathbf{x}) \rangle = \sum_{i=1}^{L} v_i^{K-FDA} \mathbf{k}(\mathbf{x}_i, \mathbf{x}) \underset{Y=0}{\overset{Y=1}{\gtrless}} T_{kfda}, \tag{15}$$

where T_{kfda} is an adjustable sensing threshold.

MAIN RESULTS AND INSIGHTS

1) *Detection performance comparison*: It is shown in Figure 5 that
   - The detection performance of the proposed nonlinear kernel FDA detector generally increases with the training sample size L and, as described in [35], linear equal coefficient combining (ECC) and linear FDA are not affected by L because no training period is needed.
   - For a large L, the proposed kernel FDA detector yields much better detection performance than the linear FDA developed in [35] and the commonly used linear ECC algorithm.
   - The optimal LRT detector will certainly yield better performance than the proposed kernel FDA. However, considering that the optimal detector is infeasible to be practically implemented, the proposed kernel FDA algorithm can be a good alternative.
2) *Computational complexity analysis*:
   - From (8) and (10), it is known that the time-complexity of linear FDA is $\mathcal{O}(N_{sc}^3)$, and the memory requirements are $\mathcal{O}(N_{sc}^2)$, where N_{sc} is the number of SCs.
   - From (14) and (15), it is seen that the time-complexity of kernel FDA is $\mathcal{O}(L^3)$, and the memory requirements are $\mathcal{O}(L^2)$. In practice, to obtain superior detection performance, the number of training samples should satisfy $L \gg N_{sc}$ in the proposed kernel FDA algorithm. In summary, compared to linear FDA, nonlinear kernel FDA can obtain significant detection performance improvement at the expense of a much higher computational complexity.

APPLICATION 3: NONLINEAR SVM FOR COVERAGE BOUNDARY DETECTION

In CRNs, besides the information about the presence of PRxs at individual locations, we are also particularly interested in which areas are covered by PTxs and which are not. The information on the coverage boundary of PTxs is crucial for spatial spectrum reuse between the primary network and the secondary network. In this case study, the adaptivity and effectiveness of SVM for the problem of coverage boundary detection will be demonstrated.

PROBLEM STATEMENT

Figure 6 presents an illustrative example of the problem. In this example, there is a PTx and a ground-truth coverage boundary between its covered and uncovered areas. Here we generalize the problem proposed in [15] by considering a practical irregular radio coverage model. The irregular shape of the ground-truth boundary results from signal attenuation due to obstructions such as buildings. Suppose that N spectrum sensors are uniformly distributed in a 2-D area $\tilde{\mathcal{A}} \subset \mathbb{R}^2$. Equipped with an energy detector, each spectrum sensor at location $l_i := (x_i, y_i) \in \tilde{\mathcal{A}}, i \in \{1, 2, \ldots, N\}$ makes a binary declaration $h(l_i) \in \{-1, 1\}$ on whether the location l_i is covered (say $h(l_i) = -1$) by the PTx or not (say $h(l_i) = 1$). Due to the hardware constraints of the spectrum sensors and radio channel randomness, the declarations are generally not error free. We, however, have no knowledge which declarations are correct. Based on the declarations with potential errors, the objective is to find a boundary function f with the minimal detection errors, i.e.,

$$\underset{f}{\text{minimize}} \sum_{l_i \in \tilde{\mathcal{A}}} \bar{h}_f(l_i) \oplus h(l_i), \tag{16}$$

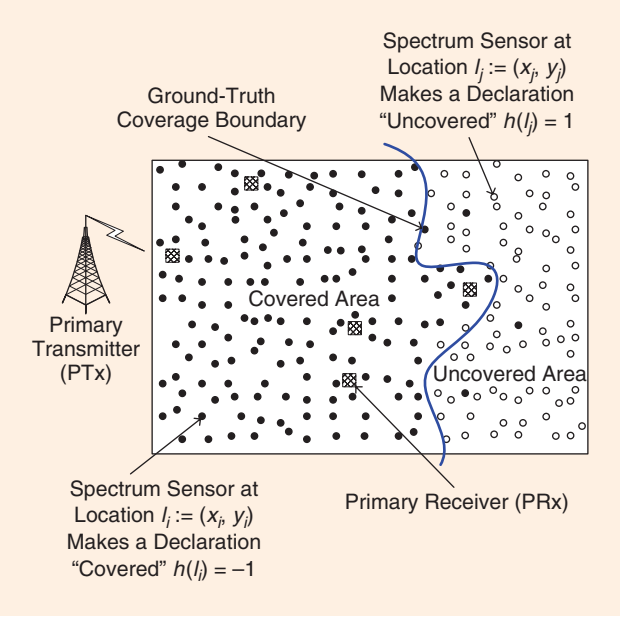
where $\bar{h}_f(l_i) \in \{-1, 1\}$ denotes the coverage state at location l_i determined by the boundary function (i.e., if $f(l_i) \geq 1, \bar{h}_f(l_i) = 1$; else if $f(l_i) \leq -1, \bar{h}_f(l_i) = -1$) and $\oplus$ is a binary operator defined as if $\bar{h}_f(l_i) = h(l_i)$, then $\bar{h}_f(l_i) \oplus h(l_i) = 0$; else if $\bar{h}_f(l_i) \neq h(l_i)$, then $\bar{h}_f(l_i) \oplus h(l_i) = 1$.

SVMs serve as a promising theoretical tool to provide an efficient solution to (16). We start with the formulation of the problem using simple linear SVMs, and then extend to the design of nonlinear classifiers by effectively kernelizing the linear SVMs. The performance comparison between linear SVMs and kernel SVMs will be provided later.

LINEAR SVM FORMULATION

Linear SVMs work for data that are linearly separable and hence, may not work well for the problem considered here. Nonetheless, it is the easiest algorithm to understand, and it serves as a performance benchmark and forms the main building block for the more complex SVMs. It exhibits the key features that characterize this kind of learning machine, and its description is therefore crucial for understanding the more advanced systems introduced later.

Linear SVMs attempt to find a separating hyperplane (corresponding to a linear coverage boundary in this problem) $\langle \mathbf{w}, l \rangle + b = 0, l \in \tilde{\mathcal{A}}$ with the largest margin satisfying constraints



[FIG6] A generalized illustrative example of the problem of coverage boundary detection proposed in [15]. The PRxs can be located anywhere inside the PTx's coverage area. The solid dots represent that the sensors make a declaration as "covered," and circle dots represent those making a declaration as "uncovered" by the PTx. Circles inside the ground-truth coverage and solids outside the ground-truth coverage are erroneous declarations.

$$\langle \mathbf{w}, l_i \rangle + b \geq 1, \text{ for } h(l_i) = 1; \langle \mathbf{w}, l_i \rangle + b \leq -1, \text{ for } h(l_i) = -1. \tag{17}$$

The optimal separating hyperplane can be derived by solving the following optimization problem [6]:

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{subject to} \quad h(l_i)(\langle \mathbf{w}, l_i \rangle + b) \geq 1, l_i \in \tilde{\mathcal{A}}. \tag{18}$$

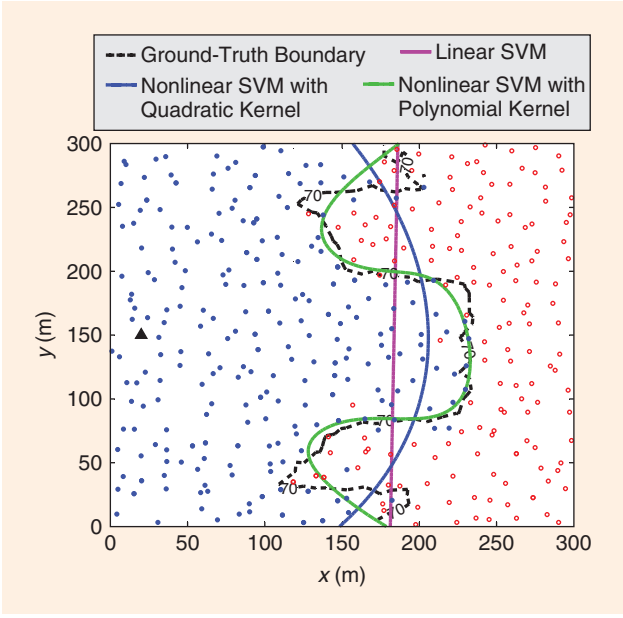
However, considering the erroneous input (declarations), an appropriate regularization parameter C should be introduced to consider the tradeoff between the maximization of margin width and the penalty to the erroneous declarations [7]

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\text{minimize}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \\ &\text{subject to} \quad h(l_i)(\langle \mathbf{w}, l_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, l_i \in \tilde{\mathcal{A}}, \end{aligned} \tag{19}$$

where ξ_i is a slack variable to consider the possibility of erroneous declarations.

By introducing Lagrange multipliers $\alpha_i \geq 0, i \in \{1, 2, \ldots, N\}$, the dual form of the optimization problem in (19) can be expressed as

$$\begin{aligned} &\underset{\alpha_i \in \mathbb{R}^N}{\text{maximize}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j h(l_i) h(l_j) \langle l_i, l_j \rangle \\ &\text{subject to } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{N} \alpha_i h(l_i) = 0, \forall i \in \{1, 2, \ldots, N\}. \end{aligned} \tag{20}$$



[FIG7] An implementation instance for coverage boundary detection. The PTx is located at (20,150). The received signal strength at the ground-truth boundary is assumed to be -70 dB. The solid dots represent the sensors making a declaration as "covered" by the PTx, and the circle dots represent those making a declaration as "uncovered."

The optimal solution $\alpha_i^*, i \in \{1, 2, \ldots, N\}$ to (20) can be found by a quadratic programming (QP) slover [7] and for any location $l \in \tilde{\mathcal{A}}$, the decision function for the linear boundary can be expressed as

$$f(l) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i^* h(l_i)\langle l, l_i\rangle + b^*\right), \tag{21}$$

where the threshold b^* can be obtained by averaging $\tilde{b}_j = h(l_j) - \sum_{i=1}^{N} \alpha_i h(l_i)\langle l_j, l_i\rangle$ over all points with $0 < \alpha_j < C, j \in \{1, 2, \ldots, N\}$.

NONLINEAR SVM FORMULATION

In practice, the ground-truth coverage boundary of the PTx is nonlinear, and it can be in arbitrary irregular shape due to wireless shadowing fading [37]. To effectively improve the results obtained in the last subsection, a nonlinear function Φ is first used to map the original data in the input space $\mathcal{X} = \mathbb{R}^2$ onto a higher-dimensional feature space $\mathcal{F} = \mathbb{R}^K$, i.e.,

$$\Phi : l = (x, y) \in \mathbb{R}^2 \quad \longmapsto \quad \Phi(l) = (\Phi_1(l), \Phi_2(l), \ldots, \Phi_K(l)) \in \mathbb{R}^K. \tag{22}$$

Specifically, in [15], the nonlinear coverage boundary in area $\tilde{\mathcal{A}}$ is simplified as a circular arc, which is written as

$$f : (x - x_0)^2 + (y - y_0)^2 = R_{th}^2, \tag{23}$$

where (x_0, y_0) is the location of the PTx, (x, y) represents any location lying on the coverage boundary, and R_{th} is the coverage radius. The key issue in boundary detection then becomes to find (x_0, y_0) and R_{th} of minimal detection errors. The simplified boundary function in (23) can be rewritten, in the form of the separating hyperplane in the feature space, as [15]

$$f : \langle \mathrm{w}, \Phi(l)\rangle + b = 0, \tag{24}$$

where $\mathrm{w} = (1, -2x_0, -2y_0)$, $b = x_0^2 + y_0^2 + R_{th}^2$, and a nonlinear function is explicitly given as $\Phi(l) = (x^2 + y^2, x, y)$, which is used to map the data from the 2-D input space into a three-dimensional feature space.

Thanks to the kernel trick, we can derive a more general nonlinear and irregular boundary function without knowing the nonlinear mapping $\Phi(l)$ explicitly. From a practical point of view, to perform nonlinear SVM for coverage boundary detection, the key idea is to replace the inner product operator in the linear SVM with proper kernels

$$\langle l_i, l_j\rangle \quad \longmapsto \quad k(l_i, l_j) := \langle \Phi(l_i), \Phi(l_j)\rangle, \quad \forall l_i, l_j \in \tilde{\mathcal{A}}. \tag{25}$$

Consequently, the linear decision boundary function in (21) can be extended to a more general nonlinear boundary function as

$$f(l) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i^* h(l_i)\, \mathrm{k}(l, l_i) + b^*\right), \tag{26}$$

where α_i^* and b^* can be obtained in a way similar as that introduced in the last subsection.

MAIN RESULTS AND INSIGHTS:

Figure 7 illustrates the simulation scenario for coverage boundary detection using linear SVM, nonlinear SVM with quadratic kernel [15], and nonlinear SVM with polynomial kernel ($d = 5$), respectively. All algorithms are implemented on the basis of a powerful SVM MATLAB toolbox [36], where the functions of kernel parameter selection are well integrated. It appears that the boundary obtained from nonlinear SVM with polynomial kernel performs the best in matching to the ground-truth boundary, while the boundary obtained from linear SVM performs the worst.

1) *Detection performance comparison*: Figure 8 plots the mean and standard deviation of error probabilities of boundary detection for various SVM algorithms. It is shown that, for different sensor densities, i) nonlinear SVMs outperform linear SVM (in terms of lower average error probabilities), and ii) the proposed nonlinear SVM algorithm with polynomial kernel performs much better than the nonlinear SVM with quadratic kernel developed in [15], which has been designed for circular coverage boundary.

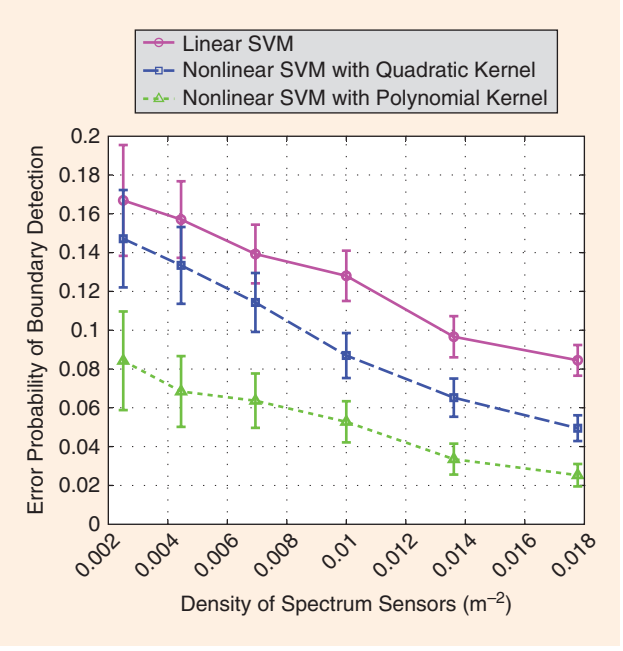
2) *Computational complexity analysis*: The core of an SVM is a quadratic programming (QP) problem as shown in (20). The computational complexity of QP solvers is data-set dependent, scaling between $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$ [38], where N denotes the number of training samples.

ADDITIONAL APPLICATIONS AND FUTURE DIRECTIONS

In this article, we have provided an overview of the state-of-the art methods and recent advances in applications of kernel-based learning for statistical signal processing in CRNs. We have covered a set of representative works to present in this article. The following topics under active research are also worth mentioning.

- Kernel supervised/unsupervised learning for robust signal classification is a major research field. In our recent work [16], the most common kernel supervised learning method, SVMs, with nonlinear radial basis function (RBF) kernel is used to identify legacy radios in a CRN by exploiting multidimensional radio frequency fingerprinting. Interestingly, in [39] unsupervised learning methods (both KMC and self-organizing maps) are used to distinguish the primary signal from the secondary signal in the absence of training data, with one basic assumption that the signal statistics used for classification are linearly separable. Based on the recent theoretical advances in the kernel KMC [20] and kernel self-organizing maps [40], this assumption can be removed and improved classification performance can be expected.
- Kernel online learning for spectrum occupancy prediction is an interesting and important application. Spectrum measurement studies (e.g., [41]) have shown that there exist significant correlations of spectrum occupancies in time, frequency, and space, which facilitate the exploitation of benefits from spectrum occupancy prediction [42]. Different from most existing spectrum prediction algorithms (see the introduction of [41]), "online learning with kernels" (e.g., [9] and [43]–[45]) provides sound theoretical tools for nonlinear, nonstationary, high-dimensional, online time-series prediction.

Finally, we envision that the topic of this article is a fruitful research direction, and we hope that this article, with interdisciplinary perspectives, will stimulate more interests in KBL theory and its applications in the signal processing and communications communities.



[FIG8] Error probabilities of boundary detection in settings of different sensor densities. Each measurement is obtained by averaging 1,000 independent runs. In each run, the sensors are uniformly distributed in a 300 m $\times$ 300 m area.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their constructive comments and helpful suggestions and Dr. Nansai Hu and Dr. Junnan Yao for their valuable input. This work is supported in part by the National Natural Science Foundation of China under grants 61172062 and 60932002, the National Basic Research Program of China under grant 2009CB320400, and the Jiangsu Province Natural Science Foundation under grant BK2011116.

AUTHORS

Guoru Ding (dingguoru@gmail.com) received his B.S. degree (with honors) in electrical engineering from Xidian University, Xi'an, China, in 2008 and his M.S. degree from the Institute of Communications Engineering, PLA University of Science and Technology, Nanjing, China, in 2011. He is currently pursuing his Ph.D. degree in communications and information systems at the College of Communications Engineering, PLA University of Science and Technology. His research interests include CRNs, machine learning, and statistical signal processing.

Qihui Wu (wqhtxdk@yahoo.cn) received his Ph.D. degree in communications and information systems from the Institute of Communications Engineering, Nanjing, China, in 2000. From 2003 to 2005, he was a postdoctoral research associate at Southeast University, Nanjing, China. From 2005 to 2007, he was an associate professor with PLA University of Science and Technology, Nanjing, China, where he is currently a full professor. In 2011, he was an advanced visiting scholar at Stevens Institute of Technology, Hoboken, New Jersey. His research interests include software-defined radio, CR, and smart radio.

Yu-Dong Yao (yyao@stevens.edu) has been with Stevens Institute of Technology, Hoboken, New Jersey, since 2000, and is currently a professor and department director of electrical and computer engineering. Previously, he was a research associate at Carleton University, Ottawa, Canada, working on mobile radio communications. From 1990 to 1994, he was with Spar Aerospace Ltd., Montreal, Canada, where he was involved in research on satellite communications. From 1994 to 2000, he was with Qualcomm Inc., San Diego, California, where he participated in research and development in wireless CDMA systems.

Jinlong Wang (wjl543@sina.com) received his B.S. degree in wireless communications and his M.S. degree and Ph.D. degree in communications and electronic systems from the Institute of Communications Engineering, Nanjing, China, in 1983, 1986, and 1992, respectively. He is currently a chair professor at PLA University of Science and Technology, Nanjing,

China. He is also the cochair of the IEEE Nanjing Section. He has published widely in the areas of signal processing for wireless communications and networking. His current research interests include software-defined radio, CR, and green wireless communication systems.

Yingying Chen (yingying.chen@stevens.edu) received her Ph.D. degree in computer science from Rutgers University. She is currently an associate professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. Prior to joining Stevens Institute of Technology, she was with Alcatel-Lucent in Holmdel, New Jersey, and Murray Hill, New Jersey. Her work has involved a combination of research and development of new technologies and real systems. Her research interests include wireless security and privacy, wireless networking, and distributed systems. Her research has been reported in numerous media outlets including *The Wall Street Journal, MIT Technology Review,* Inside Science, NPR, and CNET.

REFERENCES

[1] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, KTH Royal Inst. Technol., Stockholm, Sweden, 2000.

[2] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Select. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[3] B. Wang and K. J. Liu, "Advances in cognitive radio networks: A survey," *IEEE J. Sel. Topics Signal Processing*, vol. 5, no. 1, pp. 5–23, Feb. 2011.

[4] Y.-C. Liang, X. Wang, Y. Zeng, J. Choi, R. Zhang, and M. Luise, "Advanced signal processing for cognitive radio networks," *EURASIP J. Adv. Signal Process.*, vol. 2010, 715987, Feb. 2010. [Online]. Available: http://asp.eurasipjournals.com/series/CRN

[5] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[6] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[7] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[8] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," *J. Mach. Learn. Res.*, vol. 5, no. 1, pp. 73–99, 2004.

[9] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*. Hoboken, NJ: Wiley, 2010.

[10] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Mag.*, vol. 28, no. 1, pp. 97–123, 2011.

[11] C. Clancy, J. Hecker, E. Stuntebeck, and T. O. Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Commun.*, vol. 14, no. 4, pp. 47–52, Aug. 2007.

[12] T. Adali and S. Haykin, *Adaptive Signal Processing: Next Generation Solutions*. Wiley-IEEE Press, 2010.

[13] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1380–1391, Apr. 2012.

[14] G. Ding, Q. Wu, F. Song, and J. Wang, "Decentralized sensor selection for cooperative spectrum sensing using unsupervised learning," in *Proc. IEEE Int. Conf. Communications (ICC)*, Ottawa, ON, Canada, 2012, pp. 1601–1605.

[15] Y. Yang, Y. Liu, Q. Zhang, and L. Ni, "Cooperative boundary detection for spectrum sensing using dedicated wireless sensor networks," in *Proc. IEEE INFOCOM'10,* pp. 1–9.

[16] N. Hu and Y.-D. Yao, "Identification of legacy radios in a cognitive radio network using a radio frequency fingerprinting based method," in *Proc. IEEE Int. Conf. Communications (ICC)*, Ottawa, ON, Canada, 2012, pp. 1597–1602.

[17] R. Fisher, *Contributions to Mathematical Statistics*. New York: Wiley, 1952.

[18] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoér, "Theoretical foundations of the potential function method in pattern recognition learning," *Autom. Remote Control*, vol. 25, pp. 821–837, 1964.

[19] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Muller, "Fisher discriminant analysis with kernels," in *Proc. 1999 IEEE Signal Processing Society Workshop Neural Networks Signal Processing IX*, Madison, WI, pp. 41–48.

[20] P. A. Forero, V. Kekatos, and G. B. Giannakis, "Robust clustering using outlier-sparsity regularization," *IEEE Trans. Signal Processing*, vol. 60, no. 8, pp. 4163–4177, Aug. 2012.

[21] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio: State-of-the-art and recent advances," *IEEE Signal Processing Mag.*, vol. 29, no. 3, pp. 101–116, May 2012.

[22] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Phys. Commun.*, vol. 4, pp. 40–62, Apr. 2011.

[23] Q. Wu, G. Ding, J. Wang, Y.-D. Yao, "Spatial-temporal opportunity detection in spectrum-heterogeneous cognitive radio networks: Two-dimensional sensing," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 516–526, Feb. 2012

[24] J. Yao, Q. Wu, and J. Wang, "Attacker detection based on dissimilarity of local reports in collaborative spectrum sensing," *IEICE Trans. Commun. (Japan)*, vol. E95-B, no. 9, pp. 3024–3027, Sept. 2012.

[25] C. Chen, H. Cheng, and Y.-D. Yao, "Cooperative spectrum sensing in cognitive radio networks in the presence of the primary user emulation attack," *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, pp. 2135–2141, July 2011.

[26] N. Hu, Y.-D. Yao, and J. Mitola, "Most active band (MAB) attack and countermeasures in a cognitive radio network," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 898–902, Mar. 2012.

[27] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *Commun. Surveys Tuts.,* vol. 15, no. 1, pp. 428–445, 2013

[28] R.-P. Helena, J. B. Mercedes, and G. Carles, "Review of robust cooperative spectrum sensing techniques for cognitive radio networks," *Wireless Personal Commun.,* vol. 67, no. 2, pp. 175–198, Nov. 2012.

[29] A. W. Min, K. G. Shin, and X. Hu, "Secure cooperative sensing in IEEE 802.22 WRANs using shadow fading correlation," *IEEE Trans. Mobile Comput.,* vol. 10, no. 10, pp. 1434–1447, Oct. 2011.

[30] H.-H. Bock, "Clustering methods: A history of K-means algorithms," in *Selected Contributions in Data Analysis and Classification* (Studies in Classification, Data Analysis, and Knowledge Organization Series), P. Brito, G. Cucumel, P. Bertrand, and F. Carvalho, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 161–172.

[31] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. San Diego, CA: Academic, 2008.

[32] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory,* vol. 52, no. 2, pp. 489–509, Feb. 2006.

[33] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pairwise coordinate optimization," *Annals Appl. Stat.*, vol. 1, no. 2, pp. 302–332, 2007.

[34] Y. Chen, X. Dang, H. Peng, and H. L. Bart, "Outlier detection with the kernelized spatial depth function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 288–305, Feb. 2009.

[35] K. W. Choi, E. Hossain, and D. I. Kim, "Cooperative spectrum sensing under a random geometric primary user network model," *IEEE Trans. Wireless Commun.*, vol. 10, no. 6, pp. 1932–1944, June 2011.

[36] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–39, May. 2011.

[37] M. Malmirchegimi and Y. Mostofi, "On the predictability of communication channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 964–978, Mar. 2012.

[38] F. Pedregosa, G. Varoquaux, and A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.,* vol. 12, pp. 2825–2830, Oct. 2011.

[39] T. Clancy, A. Khawar, and T. Newman, "Robust signal classification using unsupervised learning," *IEEE Trans. Wireless Commun.*, vol. 10, no. 4, pp. 1289–1299, Apr. 2011.

[40] K. W. Lau, H. Yin, and S. Hubbard, "Kernel self-organizing maps for classification," *Neurocomputing*, vol. 69, no. 16–18, pp. 2033–2040, 2006.

[41] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Trans. Mobile Comput.,* vol. 11, no. 6, pp. 1033–1046, June 2012.

[42] M. Timmers, S. Pollin, and F. Catthoor, "On the value of prediction in opportunistic radio systems," in *Proc. IEEE Int. Conf. Communications (ICC)*, Kyoto, Japan, 2011, pp. 1–6.

[43] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Processing,* vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[44] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Processing,* vol. 56, no. 7, pp. 2781–2796, July 2008.

[45] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS," *IEEE Trans. Signal Processing,* vol. 59, no. 3, pp. 964–978, Mar. 2011.

[SP]