Cooperative Boundary Detection for Spectrum Sensing Using Dedicated Wireless Sensor Networks

Yanyan Yang, Yunhuai Liu, Qian Zhang and Lionel Ni
Hong Kong University of Science and Technology
Email: {yanyan, yunhuai, qianzh, ni}@cse.ust.hk

Abstract—Spectrum sensing is one of the key enabling technologies in Cognitive Radio Networks (CRNs). In CRNs, secondary users (SUs) are allowed to exploit the spectrum opportunities by sensing and accessing the spectrum, which exhibit many critical limitations in practical environments. In this paper, we propose a new sensing service model that uses dedicated wireless spectrum sensor networks (WSSN) for spectrum sensing. The major challenge in WSSN is the design of data fusion, for which the traditional fusion scheme will produce a large amount of errors. We formulate the problem as a boundary detection problem with notable unknown erroneous inputs. To solve the problem, we propose a novel cooperative boundary detection scheme that intelligently incorporates the cooperative spectrum sensing concept and the recent advances in support vector machine (SVM). Cooperative boundary detection consists of two major components, a declaration calibration algorithm and a boundary derivation algorithm. We prove that cooperative spectrum sensing can asymptotically approach the optimal solution. A prototype system as well as simulation experiments show that compared with the traditional approaches, cooperative boundary detection can reduce the errors by up to 95% with an average reduction about 85%.

I. INTRODUCTION

Cognitive Radio Networks (CRNs) bring new hope to alleviate the scarcity of spectrum resources for wireless communications [1]. In CRNs, Secondary Users (SUs) are allowed to use a licensed spectrum when they cause no harmful interference to spectrum license holders, the Primary Users (PUs). The spectrum and time period during which SUs can access the spectrum are referred to as spectrum opportunity.

In CRNs, a key enabling technology is spectrum sensing. Spectrums should be frequently sensed so that the appearance of the PUs can be instantly detected. When PUs return, it is a must that SUs immediately evacuate the occupied spectrum. In the draft of IEEE 802.22 standard [2], PUs should be detected within two seconds and the probability of misdetection $P_{MD}$ and false alarm $P_{FA}$ should be no more than 0.1 [3]. During spectrum sensing, these detectability requirements must be satisfied while exploiting spectrum opportunities.

Most existing CRNs use SUs for both spectrum sensing and spectrum accessing. This is, however, not an attractive approach due to many practical concerns. In spectrum sensing, individual sensing results are not always correct because of the hardware constraints. As the detectability requirements are mandatory, different sensing devices have to work cooperatively to increase the sensing accuracy. This requirement of cooperations brings a great deal of practical issues such as security, privacy protection and the difficulty of information sharing among different organizations. SU devices will be much more expensive when they have to carry both sensing and spectrum adaption functions together. The efficiency of utilizing the spectrum opportunities will also be degraded because SUs cannot sense and access spectrums simultaneously.

In this paper, we propose to use a dedicated wireless sensor network to conduct the spectrum sensing tasks for CRNs. A large number of low-cost, well-designed and carefully-controlled sensors are deployed to the application field, providing the spectrum sensing as a service. SUs subscribe the service and obtain the information on their demands. We refer such a systems to as wireless spectrum sensor networks (WSSNs), in which the legacy spectrums are the monitored resources and the appearance of the PUs are the event of interests. Volunteer SUs can also participate in the sensing if they agree to function as a simple sensor.

In WSSNs, besides the information about the presence of PUs at individual locations, we are in particular interested in which areas are covered by PUs and which are not. Fig. 1 gives an illustration example of WSSNs. In this example, there is a PU (primary transmitter) and a real boundary (blue curve) between the PU covered and uncovered areas. Sensors can sense the spectrum and make declarations based on their sensing results. The solid dots represent “on” sensors (i.e., PUs exist), and circle dots are “off”. As the sensing can be erroneous due to the hardware constraints, both circles on the left and solids on the right of the real boundary are erroneous. We, however, have no knowledge which sensors are correct.

![Fig. 1. An illustration of the system model and the motivation example. The blue curve is the ground truth boundary; the green one is by traditional boundary detections and the red one is by cooperative boundary detection.](image-url)
The basic goal of spectrum sensing in WSSNs is to accurately detect this real boundary based on the erroneous inputs.

The boundary detection in WSSNs is inherently different from the traditional boundary detection (e.g., [4]) as the latter one is mostly based on correct inputs. For this problem, traditional cooperative sensing scheme cannot be directly applied either. The traditional approach works in an all-or-nothing manner that based on the OR-rule for data fusion [5]. In the example of Fig. 1, the best that traditional approach can do is to draw a convex envelop (red curve) to enclose all the “on” nodes. As the erroneous declarations are notable, the detection error between the detected boundary and the real boundary will be large. This big gap is the main motivation of this work.

To address the problem, in this paper we propose a novel approach called cooperative boundary detection. Cooperative boundary detection is a two-stage scheme. It first runs a calibration algorithm, trying to reduce the input errors. After the calibration, the erroneous declarations can be effectively reduced and the left erroneous ones, if there are any, will be likely to be deployed near the real boundary. Taking the calibrated declarations as the new input, in the second stage a error-tolerant boundary derivation algorithm is invoked to depict the boundary. The derivation algorithm is based on recent advances in support vector machines (SVM). It offers an asymptotical approach to the optimal solution. Fig. 2 illustrates the effectiveness of the cooperative boundary detection against traditional approaches. The x-axis is the number of deployed sensors in the field and the y-axis represents the detection errors. By cooperative boundary detection we are striking a better tradeoff between the sensing accuracy and the number of sensors needed. For a given sensing requirement, we are investigating the minimal number of sensors to satisfy the requirement. The performance upper bound is obtained when every sensor makes a correct declaration. Notice that under this scenario there are still detection errors as a number of real boundaries may produce the same input.

The main contributions of this paper are highlighted as follows. First, we propose a new spectrum sensing service model using dedicated WSSNs and formulate the problem as the boundary detection with notable erroneous inputs. Second, to deal with the problem we propose a cooperative boundary detection scheme, taking cooperative sensing as an advantage. This is, to the best of our knowledge, the first attempt in literature to solve this problem. Third, as there is no optimal solution, we prove that the cooperative boundary detection can asymptotically approach the optimal solution. Fourth, we implement the cooperative spectrum sensing using off-the-shelf wireless sensor nodes in a prototype system. We also conduct comprehensive simulations to evaluate the performance. Experimental results show that compared with the traditional boundary detection algorithms, cooperative boundary detection reduces the detection errors by up to 95% with an average reduction of 85%.

The rest of the paper is organized as follows. In Sec. 2 we give a review of the related work. In Sec. 3 we present the system model and provide the formal statement of the problem. In Sec. 4 we details the calibration algorithm and the error-tolerant boundary derivation algorithm. Some analytic results will also be presented. In Sec. 5 we evaluate the performance of different approaches by simulations and field studies. In the last section we conclude this work and point out some possible future work directions.

II. RELATED WORK

Spectrum sensing at low signal power levels has gained crucial importance in CRNs. The current research efforts can be mainly classified into two categories. One focuses on how to increase the individual sensing accuracy and reduce the probability of erroneous declarations. By path-loss propagation model, the Received Signal Strength (RSS) is a random variable with the mean related to the distance between the primary transmitter and the receiver [6]. Harmonic downmixing effect [7] was also studied which may confuse the energy detector and should be taken into consideration. Besides the shadowing/multipath diversity, interference diversity was also explored [8]. For the detection methods, various methods have been proposed. In additional to the simplest energy detector, feature detections [9] allowed users to distinguish the primary transmitter from SUs. Cyclostationary signatures [10] embed a watermark in a transmitted signal by which the received signals can be uniquely detected. This category of works focuses on the accuracy of individual readings and thus is orthogonal to our problem.

The other category of works are looking for better cooperation schemes so as to produce better data fusion results. Ghasemi et.al [5] analyzed the limitation of individual sensing and shows the importance of cooperation among different sensing devices. They proposed the OR-rule data fusion to counter the shadowing/multipath effects. Ganesan et.al [11] studied the cooperations of two nodes and extended the work to a small scale of networked sensors. Kim et.al [1] studies the scheduling of feature detection and energy detection, as well as forming sensor cluster to yields the best performance. In the work [12] a consensus scheme was proposed so that no common data collector was needed to aggregate the data. Chen et.al [13] investigated the Byzantine failure problem in spectrum sensing which may be caused by malfunction sensing devices or data falsification attacks. All this kind of works are
based on the traditional sensing devices of SUs of small scales. They do not consider the scenarios of large fields in which the boundary between the boundary of the PU covered areas.

Boundary detection has also been extensively studied. In existing boundary detections, the key issue is to derive a fit boundary based on correct inputs. The widely used tool is the computational geometry. Some research efforts were devoted to recognize boundaries between sensor’s covered and uncovered areas based on node degree [14], connectivity information [15], or solely topology information [4]. In classifier based approaches, it was assumed that sensor readings in different subareas were different. Accordingly, the data similarity could be a reliable indicator for the boundary [16]. In particular, fault-tolerant detection was studied [17] and a distributed Bayesian algorithm was proposed. It allowed sensors to have error readings but focused on the off-boundary areas where the erroneous sensors are always surrounded by corrected ones. Multiple events were studied in follow-up works (e.g., [18]). To conclude, none of the exiting work is suitable for our application scenarios where the erroneous inputs are notable and mixed together.

III. Problem Formulation

In this section, we give a formal description of the problem we want to address in this paper. We first present the system model and the sensing model which are the key of the spectrum sensing apart from the traditional sensing task. In the last we formulate the problem based on these models and state the key challenges when addressing the problem.

A. System model and assumptions

We assume that a set of static spectrum sensors $V = \{v_0, v_1, \ldots, v_n\}$ has been deployed to a 2-D square application field $A$. Nodes can sense a spectrum and declare whether PUs are present or not. The sensing should be carried out frequently enough to satisfy the requirements in IEEE 802.22 standard. We assume sensors are using energy detection method, while our proposed algorithms can be easily extended when feature detection [9] is applied.

We model the spectrum as an “on/off” resource. It is “on” when the PUs are present at the location and “off” when otherwise. We assume in each field there is at most one PU. All the spectrum sensors are in cooperation to determine whether this single PU is present or not. The assumption of single PU in a given field is a reasonable assumption in practice. This is because PUs have the exclusion priority to use the spectrum. Different PUs will be far away enough to avoid the interference in between. For instance, the coverage range of a Digital TV transmitter is 150.3 km [19]. To well protect the DTV users, different DTV transmitters, when they are using the same spectrum, must be away several times of this range. Compared with this distance, the size of the field $A$ in practice is fairly small. Accordingly, it will be very rare that one field covers two different PUs.

We assume $A$ is a rectangle for the ease of presentation. The left-down corner of $A$ is set as the coordinate origin. Sensors are aware of their precise locations by carrying GPS or applying localization algorithms (e.g., [20]). Their locations are denoted as $L = \{(x_i, y_i), i = 1, \ldots, |V|\}$. Sensors will report their declarations to a central server through a control channel reliably (e.g., using a 2.4G ISM band). This server will then make calculations and provide the sensing information to end users.

B. Sensing model

Energy detection is the most popular method for signal detection due to its simplicity, small energy consumption and short sensing time [1]. The sensing results may be error, while these errors are not arbitrary but closely related to the context information (e.g., the distance between the PU and sensor). In this subsection, we will derive this relationship.

Ideally, by the path-loss model, the received signal strength (RSS) $P(d)$ in logarithm scale is $P(d) = P_0 + 10k\log(d/d_0)$ where $k$ is the path-loss exponent, $d_0$ is a distance of reference location, $d$ is the distance between the primary transmitter and the receiver and $P_0$ is a referencing transmitting power. Given a desired threshold $\lambda_{th}$ of the Signal-Noise-Ratio, the contour of $P(d) = \lambda_{th}$ is the real boundary of PU’s covered areas. In the ideal case, it is a circular arc.

In real environments, RSS is a random variable $x$ due to the shadowing effects. Its probability density function can be modeled by Rayleigh fading, i.e.,

$$\Pr(x) = \begin{cases} \frac{x^2}{\sigma^2} \exp(-\frac{x^2}{2\sigma^2}), & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Eq. 1 implies that a lower $P(d)$ will have a smaller detection variance and a higher $P(d)$ results in a larger variance. For instance, suppose $\lambda_{th} = 15dB$. Sensors of $P(d) = 15dB$ should declare “off” with $P_{FA} = 0.24$, sensors of $P(d) = 15dB$ are “on” but have $P_{MD} = 0.38$, and sensors of $P(d) = 20dB$ have $P_{MD} = 0.23$.

More measurements can reduce the error rate by the cost of more sensing time. When $m$ samples are used to make a single declaration, we have

$$x = \sum_{i=1}^{m} x_i^2/m$$

that follows non chi-square distribution where $x_i$ denotes RSS of $i$-th measurement. By this scheme we have

$$P_{MD} = 1 - e^{-\frac{\lambda}{m}} \sum_{k=0}^{m-2} \frac{1}{k!} \left(\frac{\lambda}{m}\right)^k \left(\frac{1 + P(d)}{P(d)}\right)^{m-1} \times \left(e^\frac{\lambda}{m} - e^{-\frac{\lambda}{m}} \sum_{k=0}^{m-2} \frac{1}{k!} \left(\frac{\lambda P(d)}{2(1 + P(d))}\right)^k\right)$$

Definition Let $h(l)$ denote a declaration of a sensor at location $l = (x, y)$ where $h(l) = 0$ when the average of $m$ sensing samples is above a given threshold, implying that no PUs exist, and $h(l) = 1$ when otherwise. The declaration $h(l)$ is also written as $h(v_i)$, $v_i \in V$ if $v_i$ is at $l$.

In this paper we will use the real boundary and the ground truth boundary interchangeable.
Definition We define declaration set \( H(L) \) as the set of declarations at all locations in \( L \), \( H(L) = \{ l(l), l \in L \} \). We also write \( H(L) \) as \( H(V) \) if \( V \)'s location set is \( L \).

We use \( \mathcal{H}(L) = \{ \tilde{h}(l), l \in L \} \) to denote the ground truth of the presence of PUs.

Definition Boundary in a field \( A \) is a real-valued function \( f : R^2 \rightarrow R \). A location \( l = \{ x, y \} \in A \) is assigned to the “off” class (no PU presence) if \( f(l) \geq 0 \), and “on” class otherwise. The declaration set \( H_f(A) \) of \( f \) can be computed by

\[
H_f(A) = \{ h_f(l) | l \in A, \ (f(l) \geq 0) \cap (h_f(l) = 0), \ (f(l) < 0) \cap (h_f(l) = 1) \}
\]

C. Problem statement and key challenges

The main objective of cooperative spectrum sensing is to maximize the utilization of the spectrum opportunity while guaranteeing the detectability requirements of the PUs. It can be formulated as follows.

Sensing Error Minimization (SEM): Given a set of spectrum sensors \( V \) and their locations \( L \) in \( A \) and a declaration set \( H(L) \). The problem is to find a boundary function \( f \) with the minimal detection errors, i.e.,

\[
\text{minimize : } \hat{H}(A) \oplus H_f(A) = \sum_{l \in A} | \tilde{h}(l) \oplus h_f(l) |
\]

where \( H_f(A) \) is \( f \)'s declaration set and \( \oplus \) is XOR binary operation.

Indeed, because of the nature of the erroneous input, no algorithm can produce the ground truth boundary with guarantee for SEM problem. Consider a simple case that all the declarations are the same, say “on”. In such a case, no boundary can be drawn even we are given the knowledge that there is a boundary. In general, there are several key challenges when we explore the solutions.

The first challenge is that the sensing errors are not sensitive to the distance. A node far from the real boundary also has a high probability of errors. Thus the correct and erroneous declarations will be mixed together. To tackle this challenge, we intelligently leverage the cooperative spectrum sensing concept. The basic idea is that a sensor is very likely to have an erroneous declaration if all its neighbor sensors in proximity have the opposite declaration. The key issue is how to define the neighbors of a sensor, how to deal with the multiple errors, and what is its effectiveness.

The second challenge is the lack of extra tools for verification. In other words, after the declaration calibration, there are still some erroneous declarations. For this challenges we notice that such erroneous declarations, if there are any, will be very likely to be distributed near the real boundary. Accordingly, we design an error-tolerate boundary derivation algorithm and carefully set the penalty for an erroneous declaration. As such the erroneous declaration can be effectively compensated.

The third major challenge is that the ground truth boundary can be any arbitrary shape due to the fading effect. For this challenge we have the following observations. In the ideal case of path-loss model, the read boundary is a circular arc. It becomes irregular because of the shadow fading and multi-path effects. According to the IEEE 802.22 standard, however, the spectrum opportunities coming from shadow fading and multi-path effects are not allowed to access. The corresponding area will be considered as “PU presence”. From the user’s point of view, it makes no difference that whether the spectrum exists but not allowed to use, or the opportunity does not exist at all. This nature promisingly makes the design of the spectrum sensing easier. We can simply assume that the target boundary has the regular shape of a circular arc.

According to these design principles, we develop the cooperative boundary detection, which will be detailed in the next section.

IV. COOPERATIVE BOUNDARY DETECTION

Cooperative boundary detection is mainly composed of two components, a reciprocal declaration calibration algorithm and an error-tolerate boundary derivation algorithm. Fig. 3 illustrates the general architecture of cooperative boundary detection. Initially the server should collects the declarations from all sensors in the field. It then constructs declaration set \( H(V) \) as the original input (Fig. 3.a). Based on \( H(V) \) the server executes the reciprocal declaration calibration algorithm and obtain a calibrated declaration set \( H_c(V) \) (Fig. 3.b with a transfer label). The calibrated declaration set \( H_c(V) \) will be used as the input of the error-tolerant boundary derivation to compute the boundary function (Fig.3.c). It is then ready for SUs to query whether a particular position has PU in present or not.

In the remainder of this section, we present the two major components respectively.

A. Reciprocal declaration calibration

In this subsection, we first introduce the basic idea of reciprocal declaration calibration, and then detail the design. In the last we show some analytic results.

1) Design principles: Declaration calibration is based on the cooperative sensing concept. A pair of nearby sensors can mutually “support” each other when they two share the same declaration. Otherwise, they two oppose each other, implying that either the ground truth boundary is in between, or one of the two declarations is incorrect and should be flipped. Indeed, it is still possible that two same nearby declarations are both errors, while their confidence of the correctness is dramatically increased. The key issue is the definition of neighborhood.

A naive approach to define neighborhood is range-based. It is also referred to as disk model that each node has a deterministic range. All other nodes within the range are considered as the neighbors of this node. Disk model is, however, not an appropriate one because in spectrum sensing, there is no sensing range and declarations only represent the sensing results of individual locations. Moreover, directions are more important than the distance. In the example of Fig. 3, sensors \( u \) and \( w \) are very close to each other and have the same declarations. However, they two should provide no mutual
support as \( u \) has been surrounded by sensors of opposite declarations.

Inspired by these observations, we propose a Voronoi-based declaration calibration algorithm. Voronoi diagram [21] and its dual form Delaunay triangulation (DT) are powerful tools that present many important and useful properties. Due to space limitation, we do not introduce too much about them and interested readers may refer to the reference [21] for more details.

**Definition** Define \( \Upsilon(v_i) \), the Voronoi cell for a sensor \( v_i \), to be the set of points in \( A \) that are closer to \( v_i \) than any other sensor, i.e.,

\[
\Upsilon(v_i) = \{ p | d_{p,v_i} \leq d_{p,v_j}, p \in A, i \neq j \}
\]

where \( d_{p,v} \) is the Euclidean distance between \( p \) and \( v \).

The set of all Voronoi cells of all sensors is a tessellation corresponding to the sensor set \( V \). This tessellation is referred to as Voronoi diagram in a 2-D space.

**Definition** Two sensors \( u \) and \( v \) are Voronoi neighbors if \( \Upsilon(u) \) and \( \Upsilon(v) \) share a common edge in the Voronoi diagram. The set of all Voronoi neighbors of a given sensor \( v_i \) is denoted as \( \mathcal{N}(v_i) \). Voronoi neighbors are also the neighbors in DT.

**Definition** Given a sensor \( v \) and a declaration set \( H(V) \), another sensor \( u \) is said as “supporter” of \( v \) if \( h(u) = h(v) \); and “objector” if otherwise. Let \( \mathcal{N}_s^H(v) \) denote the supporter set and \( \mathcal{N}_o^H(v) \) the objector set, i.e.,

\[
\mathcal{N}_s^H(v) = \{ u | h(u) \odot h(v_i) = 0, u \in V \}
\]

\[
\mathcal{N}_o^H(v) = \{ u | h(u) \odot h(v_i) = 1, u \in V \}
\]

Noticing that a neighbor is either a supporter or an objector, we have

\[
\mathcal{N}_s^H(v) \cup \mathcal{N}_o^H(v) = V
\]

\[
\mathcal{N}_o^H(v) \cap \mathcal{N}_s^H(v) = \emptyset
\]

**Definition** A sensor \( v_i \) is safe if at least one of the three following conditions is satisfied:

1) All \( v_i \)'s Voronoi neighbors are \( v_i \)'s support, or;
2) \( v_i \) is along the perimeters of the field, or;
3) There exist a path in the DT from \( v_i \) to a \( v_i \)'s safe supporter such that all nodes along the path are also

In our reciprocal declaration calibration, all the safe nodes are kept unchanged, while the rest un-safe sensors should flip their declarations until safe. In later we will prove that after the calibration, all sensors will be safe with respect to the calibrated declaration set. In the next we present the algorithm in details.

2) Detailed design: Figure 4 gives the pseudo-code of the reciprocal declaration calibration algorithm. Initially, the data server should obtain \( H(V) \) from sensors. In line (2), the server computes Voronoi diagram based on \( L \) and \( H(V) \). With the Voronoi diagram and \( H(V) \) available, \( \forall v \in V, \mathcal{N}(v), \mathcal{N}_s^H(v), \mathcal{N}_o^H(v) \) are all available. The server then scans all the sensors to check whether there are safe nodes satisfying the first condition in the safe node definition, and put all these safe nodes in a queue \( Q \). As long as all such safe nodes are found, the server runs a simple Breath-First-Search from each safe node to find other safe nodes. Notice that during the search, only links between supporters are considered (line 6.a). The declaration set is obtained when declarations of all unsafe nodes are flipped.

The algorithm complexity is easy to analyze. We omit the process of analysis but give the results directly. The time complexity of the calibration is \( O(n \log n) \). The most important thing is that DT has at most \( 3n - 6 \) links as DT is a planar graph. In DT, the average node degree is a constant 3.
3) Analytic results: In this part we show some important features of the calibrated declaration set. Due to space limitation, proofs of theorems are omitted.

**Theorem 4.1:** All sensors are safe with the respect to the calibrated declaration set.

**Theorem 4.2:** Given a set of sensors \( V \) and the calibrated declaration set \( H_c \), assume each sensor has at least \( \rho \) Voronoi neighbors in DT. Then the sensing performance of a node with \( N \) connected supporters is \( P_{MD}(h_c) = P_{MD}^N \) and \( P_{FA}(h_c) = 1 - (1 - P_{FA})^N \) where \( h_c \in H_c \) is the node’s calibrated declaration, under the assumption that every sensor has the same \( P_{MD} \) and \( P_{FA} \) for a given signal strength. Notice that \( N \geq 4 \) in practice.

Figure 5 shows some sensing performance improvement after calibration. Assuming the RSS threshold is 15 dB, for sensors of \( P(d) = 15 dB \), \( P_{MD} \) is reduced from 0.38 to 0.02, and for sensors of \( P(d) = 20 dB \), \( P_{MD} \) is reduced from 0.23 to 0.002. Note that we set \( N = 4 \) for conservative and in practice \( N \) is much larger.

**B. Error-tolerant boundary derivation**

Calibration can effectively reduce the erroneous declarations but provide no error-free guarantee. To draw an accurate boundary, we are facing a number of other challenges.

The first challenge is the infinite number of candidate boundary functions. Given the input declaration set, assuming all declarations are correct, there will an infinite number of boundaries that can produce the same declaration set as the input. In other words, any of them can be the solution. We need to choose the most adequate one with the minimal errors. The intuition is that the boundary with the maximal margin width between the two classes of declaration will be the one.

The second challenge is that the input declaration set is not error-free. Nevertheless, we are given that the erroneous declarations should be distributed near the real boundary. In other words, the boundary based on which the calibrated declaration set has the minimal errors will be the one closest to the real.

Based on these two observations, we select SVM as the fundamental tool to derive the boundary function. SVM is a learning system that uses a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory, to obtain the results. It is a supervised learning that the learning machine is given a training set of examples with the associated labels as the input (the declarations of sensors in this work). As long as the learning system is training by the training set, a number of sets of hypotheses can be chosen for the problem. Due to space limitation, we provide no details for SVM and more information can be found in the reference [22].

SVM can only derive linear boundary functions. As the target boundary function is a circular arc but not linear, we need to change the representation of the sample data from the 2-D input space to a higher dimensional space. As such, the target boundary function becomes linear i.e.,

\[
 l = (x, y) \in A \mapsto \varphi(l) = (\varphi_1(l), \varphi_2(l), ..., \varphi_K(l)) \in \mathbb{R}^K
\]

This step is equivalent to mapping the input space \( A \) into a new space \( \mathbb{R}^K \) which is referred to as the feature space. To derive the feature function \( \varphi \), we write the target boundary function as

\[
 f : (x - x_0)^2 + (y - y_0)^2 = R_{th}^2
\]

where \((x_0, y_0)\) is the location of the primary transmitter and \( R_{th} \) is the keep-out radius of the primary transmitter. The key issue in boundary detection then becomes to find \((x_0, y_0)\) and \( R_{th} \) of minimal detection errors. Eq. (2) can be rewritten as

\[
 f : <w, \varphi(l)> + b = 0
\]

where

\[
 w = [1, -2x_0, -2y_0] \\
 \varphi(l) = [x^2 + y^2, x, y] \\
 b = x_0^2 + y_0^2 - R_{th}^2
\]

In other words, in a 3-D feature space the target boundary function is a linear function. The mapping function for the feature space is

\[
 \forall l = (x, y) \in \mathbb{R}^2, \varphi(l) = [\varphi_1(l), ..., \varphi_3(l)] \in \mathbb{R}^3
\]

where \( \varphi_1(l) = x^2 + y^2, \varphi_2(l) = x \), and \( \varphi_3(l) = y \).

A last question when applying SVM is the erroneous inputs. This can be addressed by setting an appropriate cost parameter to the erroneous declarations. Combining all these, we design SVM with the target boundary function \( f \) such that it can minimize

\[
 \min \quad : <w, \varphi(l)> + C \cdot \sum_{i=1}^n \varepsilon_i \quad \text{subject to} \quad : \langle w \cdot l_i \rangle + b \geq 0, i \in [1, n], h_i = 0 \\
 \langle w \cdot l_i \rangle + b < 0, i \in [1, n], h_i = 1
\]

where \( <w \cdot \varphi(l)>^2 \) is the margin width between different classes (sensors with different declarations), and the cost parameter \( C \) that strikes the tradeoff between the maximization of margin width and the penalty to the erroneous declarations. When all labels (declarations) are correct, there is no erroneous input

^2Indeed, the margin width should be \( \frac{2}{\sqrt{<w \cdot \varphi(l)>}} \). It is equivalent to \( <w \cdot \varphi(l)> \) by an appropriate scalar

Fig. 5. The sensing performance before and after the reciprocal declaration calibration
and the cost parameter $C$ is 0. When the erroneous declarations are many, a higher $C$ is needed. In practice, we study the impact of the parameter $C$ through experiments.

The optimal solution function $f$ in Eq. 3 can be found by quadratic programming [23].

Fig. 6 give the computation steps of the boundary function and shows how to use the boundary to determine whether a given position has the PU presence or not. We omit explanations due to the simplicity. The complexity of the computation algorithm is straightforward. The mapping from 2-D input space to the 3-D feature space requires $O(N)$ time. The complexity of computing the linear classifier $f$ depends on the efficiency of the quadratic programming which can be solved in polynomial time by the state-of-art algorithms [24]. Therefore the overall algorithm complexity is polynomial.

**Theorem 4.3:** The boundary computed by the error-tolerant boundary derivation is asymptotically the real boundary provided all the sensors have the correct declarations and the real boundary is a circular arc.

We skeleton the proof of this theorem due to its great importance. To prove it we need to show two things. First, the output linear function $f$ in the feature space defines a circular arc in 2-D input space. This can be done by deducing the inverse of the feature function $\phi$. Second, the detection error between the $f$’s defined circular arc and the real boundary approaches 0 as the number of sensors goes to infinity and the distances between Voronoi neighbors approaches 0.

**V. PERFORMANCE EVALUATIONS**

In this section we study the impact of the control parameters and evaluate the performance. We design and implement a prototype spectrum sensing system using 2.4G ISM band to emulate the licensed band. We also conduct comprehensive simulations to evaluate the performance under different settings. Recall the error can be calculated as

$$\tilde{H}(A) \oplus H_c(A) = \sum_{l \in A} [\tilde{h}(l) \oplus h_c(l)]$$

which will be used as our evaluation metric. As the shape of the real boundary is not regular and can be arbitrary, we apply Monte-Carlo method to measure this error. The studied parameters include the field size, the density of sensor deployment as well as the distribution. Each measurement is obtained by averaging 100 independent runs.

We compare cooperative boundary detection with three alternatives. The first one is a simple combination of traditional boundary detection and the cooperative spectrum sensing. OR-rule [5] is applied for data fusion and a minimal convex envelop [4] is computed to enclose all such sensors. The boundary derived by such a approach is illustrated in Fig. 1. We term this one as “Traditional”. The second alternative is the simple SVM on the raw data termed as ‘Simple SVM’. The last one is again SVM but assume all sensors have the correct declarations. This is the best we can do to draw the boundary and is depicted as “Upper Bound”. Our proposed scheme is termed as “Cooperative”.

**A. Simulation results**

In this subsection, we present the simulation experiment results. We first study the impact of two system parameters and then show how different algorithms perform in different environmental settings.

We adopt the Rayleigh model and follows IEEE 802.22 standard [25] to simulate the sensing behavior of individual sensors. Two distributions of sensors are investigated. One is a random deployment following Poisson distribution. The other is a uniformly random distribution in which the field is partitioned into fixed size grids. In each grid we deploy one sensor at a random location. The number of deployed sensors varies from 25 to 900. We also evaluate the performance of different algorithms under different field sizes. According to the experimental setting, 15 km corresponding to 1 dB radio attenuation of PUs. Due to space limitation, only some representative results are presented.

1) Impact of number of samples $m$: Figure 7 shows the impact of $m$ the number of samples when individual sensors signal the spectrums. The field size is set to be $15 \times 15km^2$. Two numbers of sensors are deployed to the field, namely 225 and 900. We are using uniformly random deployment and thus in each side there are about 15 nodes. From this figure we find many interesting results.

First, traditional boundary detections can get few benefits from more signaling samples. The error is steadily around 50% regardless the number of deployed sensors and $m$. This is because the accuracy of the traditional approach heavily relies on the sensor with “on” declarations. As the OR-rule is applied in data fusion, three “on” sensors are sufficient to draw the boundary. As a result, a smaller number of error declarations does not help much. Compared with this naive approach, simple SVM gets obvious benefits from a smaller number of error declarations. In simple SVM, one more declaration may move the boundary towards the correct direction and thus fewer errors produce a higher accuracy. In our experiments, $m = 6$ can reduce the error from 42% to 20% compared with $m = 2$. Cooperative boundary detection is able to obtain a satisfactory result even when $m$ is relatively low. This is mainly due to the power of the reciprocal declaration calibrations, which can greatly reduce the error declarations. As many errors can be calibrated, cooperative boundary detection provides a similar performance of a small $m$ as the simple SVM with a large $m$. 

---

**Error-tolerant boundary detection**

1. Collect the calibrated declaration set
2. Map the 2-D input space to 4-D feature space based on the feature function $\phi(l)$
3. Execute the quadratic programming to find a linear function $f$
4. Determination of PU’s presence
   1. For a given queried position $p$, map it to the feature space using $\phi(p)$
   2. Compute $f(\phi(p))$
   3. Report “PU presence” if $f(\phi(p)) \leq 0$; and “No PU presence” if otherwise

Fig. 6. Pseudocode of the error-tolerant boundary derivation and the determination of PU presence for a location $p$
2) Impact of cost parameter $C$: In this set of experiments we study the cost parameter $C$ in Eq. 4. The cost parameter $C$ strikes the tradeoff between the detection error and the number of error declarations. Ideally, a high density leads to a smaller detection error but more declaration errors, and therefore a larger $C$ is suitable. Fig. 8 depicts the error of cooperative boundary detection against the real boundary. We adopt the uniform randomly deployment and the numbers of sensors $n$ varies from 25 to 900. The field size is fixed as $15km^2$. In general, a larger cost parameter $C$ can really improve the accuracy as the error declarations will obtain more penalties. As $C$ is more than 100, the penalties are sufficient to compensate the effect of error declarations and thus the improvement of further increasing $C$ becomes minor. In practice, we suggest $C > 100$. As more nodes are deployed, a larger $C$ is needed.

3) Scalability of different approaches: In this set of experiment we study the scalability of different approaches on the performance. The field size is fixed as $1km^2$ and the number of samples $m$ is set to 1. Four alternatives are studied, namely the traditional boundary detection, the simple SVM, our proposed cooperative boundary detection, and the upper bound under ideal cases. Fig. 9 plots the errors of these four approaches with $n$ from 25 to 900. The traditional approach has a slight performance degradation as $n$ scales up. From $n = 25$ to $n = 900$, its error increases from 42% to 49%. Regarding the the upper bound error 4%, the relative error is about 45% ($n = 900$). For the simple SVM, as $n$ scales up, the error declarations are more likely to be overcome by correct ones, and thus the error steadily decreases. From $n = 100$ to $n = 900$, it decreases from 38% to 18%. We, however, are not clear why its performance becomes worse when the network scales from 25 to 100, which will need further investigations. For cooperative boundary detection, by the power of reciprocal declaration calibration, there is a greater improvement over the traditional approach than the simple SVM. When $n$ is small, the calibration is limited as the correct declarations are not sufficient to dominate incorrect ones. As more sensors are deployed, the corrected ones become the majority. As a result, cooperatively boundary detection significantly reduces the error from 28% of $n = 25$ to 7% of $n = 900$. Compared with the upper bound of 4%, the relative error is only 3%. In other words, the detection error is reduced by 95%, from 45% to 3%, and the average reduction is around 85%, from 45% to 7%.

4) Impact of field size: Figure 10 shows the errors of different approaches under different field sizes. We set $n = 400$, while the length of the field ranges from 15 km to 90 km. In general, as the field size increases, sensors in different areas will have more differences of the probabilities of erroneous declarations. The traditional boundary detection can hardly leverage this features as its performance only depends on three false alarm nodes. Both simple SVM and cooperative boundary detection can take this feature as an advantage, while cooperative boundary takes more.

5) Impact of different distributions: Figure 11 plots the performance under different distributions. Compared with those in Fig. 9 of uniform distribution, simple SVM and cooperative boundary detection have the similar results, implying that these two algorithms are robust to the non-uniform distributions. Traditional approach has a slight performance degradation. For the upper bound, notice that under non-uniform distribution
there are more variances between nearby sensor distances. As upper bound is closely related to the nearby sensor distance, in low network scale setup the error of the upper bound is larger than that in Fig. 9.

B. Prototype measurements

In simulations we assume that the Rayleigh model can adequately describe the sensing behavior of individual sensors. To justify this, we conduct a number of field studies and compare the real measurements with the theoretical models. Since there is no off-the-shelf spectrum sensors available, we use the 2.4G ISM band as the licensed band. We use a 802.11g wireless router to emulate the behavior of PUs. Backlog traffics are injected so that the spectrum is very likely to be fully utilized. We use 10 Berkeley TelosB sensor nodes [26], [27] to function as the spectrum sensors. They are configured in the same spectrum as that in router and report the RSS to a laptop. Each sensor measures 1000 samples of the spectrum and the CDF of RSS at three locations are depicted in Fig. 12. Since we have no knowledge about the transmission power of the transmitter, we select a scaler such that the plots are comparable. Results show that Rayleigh model can quite accurately characterize the sensing behavior. Though there is a gap between the field measurement and the ideal measuring, there is a consistent trend and the gap is acceptable.

VI. CONCLUSION AND FUTURE WORK

In this paper we proposed a new service model for cooperative spectrum sensing in CRNs. We formulated the problem as the boundary detection with notable sensing errors and proposed cooperative boundary detection scheme to solve the problem. We proved that cooperative boundary detection was asymptotically the optimal. Experimental results showed that cooperative boundary detection could accurately derive the boundary in practice. Compared with the traditional boundary detection, cooperative reduced the detection errors by up to 95% with an average reduction of 85%. Results also showed that cooperative boundary detection was more robust to erroneous input, making it in particular suitable for high efficiency utilization of spectrum opportunities.

The new service model of WSSNs for spectrum sensing provides many new yet challenging issues. First, we only considered the problem of using a single spectrum. The ultimate goal of spectrum sensing is to maximize the utilization spectrum opportunities. Towards this end, we need to find a better tradeoff between the detection accuracy and the number of sensors. Second, we assume that there is only one primary user in present. The assumption may not always be valid. We need smarter algorithms to handle more complex cases. We also plan to carry out more field studies based on our test-bed but of a larger scale. The current implementation scalability is still limited.

ACKNOWLEDGMENTS

This research was supported in part by Hong Kong RGC Grant HKUST617908, China NSFC Grant 60933011, the National Basic Research Program of China (973 Program) under Grant No. 2006CB303000, the National Hi-Tech R&D Program of China (863 Program) under Grant No. 2008AA01A324, the National Science and Technology Major Project of China under Grant No. 2009ZX03006-001, and the Science and Technology Planning Project of Guangdong Province, China under Grant No. 2009A080207002.

REFERENCES


