Co-SpOT: Cooperative Spectrum Opportunity Detection Using Bayesian Clustering in Spectrum-Heterogeneous Cognitive Radio Networks

Alireza Zaeemzadeh, *Student Member, IEEE*, Mohsen Joneidi, *Student Member, IEEE*, Nazanin Rahnavard, *Member, IEEE*, and Guo-Jun Qi

***Abstract*—A Bayesian data-mining approach is introduced for dynamic spectrum sensing in spectrum-heterogeneous cognitive radio networks. The goal is to find the spectrum opportunities in temporal, spectral, and spatial domains in the presence of unreliable sensing data and unavailable location information. In spectrum-heterogeneous networks, the availability of spectrum varies over the space and different sensors experience different spectrum opportunities. Thus, the measurements from sensors cannot be simply aggregated to detect the spectrum opportunities. Moreover, unreliable data will negatively impact the decision-making process. The task of inferring the spectrum status becomes even more challenging when the sensors are not equipped with location-finding technologies. In this paper, we propose a probabilistic model to cluster the sensors solely based on their observations, not requiring any prior knowledge of the network topology, location of the sensors, or the number of clusters. After receiving the sensing data, the base station updates the probability distributions of cluster membership, channel availability, and device reliability. All the update rules are derived mathematically by the variational inference. Then, the distributions are employed to find spectrum opportunities via multi-label graph cuts method. Experimental results demonstrate the effectiveness of the proposed approach with respect to existing algorithms.**

***Index Terms*—Cognitive radio ad hoc networks, dynamic spectrum access, consensus-based cooperative spectrum sensing, clustering methods, graph theory.**

I. Introduction

THE COGNITIVE radio (CR) paradigm is seen as the solution for the spectrum scarcity [1], [2]. The main idea is to let the unlicensed users, also known as secondary users (SUs), utilize the spectrum, provided that they will not interfere with the communication of licensed primary users (PUs) [2]. Thus, it is the responsibility of the SU devices to efficiently sense their surrounding environment and intelligently adapt to it.

The main goal of CR devices is to detect and utilize the *spectrum opportunities* in time and space, i.e., finding time intervals and areas in which the PU signal is absent [1]. *Spectrum heterogeneity* among SUs is due to the fact that SUs in different locations are experiencing different spectrum opportunities. An efficient CR device should be able to discover spectrum opportunities both in temporal and spatial domains.

However, integrating intelligence into CR devices is a challenging task. The cost of opportunistic usage of spectrum is often neglected by researchers. A CR framework is not practical unless the cost of dynamic spectrum usage is realistic [3]. Thus, a CR framework should be implementable with low-cost devices. In this work, it is assumed that:

- sensors are not capable of performing computationally intensive tasks or extracting complex features of environment,
- sensors do not have location-finding technologies,
- sensors might report *unreliable* measurements, either unintentionally or maliciously,
- each SU is able to sense at most one frequency band at each sensing time,[1] and
- missing entries will exist in the data stream, either because of energy or bandwidth constraints or because of network or device failures.

A spectrum opportunity detection algorithm is proposed to detect opportunities both in time and space. Our proposed method is referred to as *cooperative spectrum opportunity detection* (Co-SpOT). Here, since the SUs do not have processing power, they have to be able to communicate with a *base station* (BS). The BS has to aggregate data from different sensors and identify which frequency bands are more likely to be empty for each SU. The problem becomes more challenging when location information is not available, spectrum

Manuscript received March 13, 2017; revised August 7, 2017 and October 25, 2017; accepted December 14, 2017. Date of publication December 27, 2017; date of current version June 19, 2018. This material is based upon work supported by the National Science Foundation under Grant No. ECCS-1418710 and Grant No. CCF-1718195. The associate editor coordinating the review of this paper and approving it for publication was L. DaSilva. *(Corresponding author: Alireza Zaeemzadeh.)*

The authors are with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: zaeemzadeh@eecs.ucf.edu; joneidi@eecs.ucf.edu; nazanin@eecs.ucf.edu; guojun.qi@ucf.edu).

Digital Object Identifier 10.1109/TCCN.2017.2787710

[1]Our algorithm can be used for the cases when SUs are able to sense any arbitrary subset of channels.

opportunities vary over time and space, and SUs might report irrelevant measurements.

Most of the prior work focuses on discovering spectrum opportunities in either time [4]–[8] or space [9]–[12]. Recently, there has been an effort to address the problem of spatio-temporal spectrum sensing, though most studies are able to work in the presence of only a single PU or need to be aware of SU locations [13]–[21]. For instance, Debroy *et al.* [20] exploit interpolation techniques to build a spatio-temporal spectrum maps from the measurements of a network of SUs with known locations. A maximum likelihood technique is proposed in [19] to estimate the direction of transmission and power of a single directive source. In this work, it is assumed that the BS is aware of the locations of the SUs as well as the source. The problem of discovering the white spaces is discussed in [11]. In this approach, the fusion center aggregates the data from randomly distributed sensors with binary measurements and known locations to find the spectrum opportunities within a geographical area. Intelligent cooperative spectrum sensing is proposed in [12]. This method uses Bayesian inference to find the spectrum holes by considering the spatial correlation among the observations. However, the main focus of the work is on finding the spectrum opportunities, not allocating the resources. Moreover, temporal correlation of the observations is not taken into account. In [22], a cluster-based spectrum sensing is devised to discover and assign the spectrum opportunities in time and space without knowing the location of the devices. However, the update rules to predict the channel availability are mostly heuristic and the reliability of the measurements is not taken into the account.

A. Our Contributions

The main contribution of this work is addressing the spectrum opportunity detection problem in the spectrum-heterogeneous cognitive radio networks and in the absence of location information. The novelty of the proposed Co-SpOT framework is two-fold.

1) We propose a Bayesian data analysis technique that allows us to cluster the SUs solely based on their observations. Each cluster contains SUs that are experiencing similar spectrum opportunities. Simultaneously, the spectrum occupancy status and reliability of each SU are inferred.
2) We propose a graph-theory-based method to exploit the extracted information and to discover the spectrum opportunities for each SU.

To the best of out knowledge, this is the first time that a Bayesian inference algorithm is proposed to extract information in the absence of location information and a graph cut algorithm is used to find the spectrum opportunities. After receiving each set of new measurements, the probability distributions of the unknown variables are updated using the closed form update rules. It is also worthwhile to mention that it is assumed that the BS does not have any prior information on the number of clusters, number of PUs, and location of SUs.

The organization of the paper is as follows. In Section II, the system model is presented. Then, the generative model of the proposed Bayesian framework is introduced in Section III. The inference algorithm is discussed in Section IV. In Section V, the inferred information are used to discover the spectrum opportunities. Finally, Section VI presents the simulation results and Section VII draws conclusions.

II. System Model

The network consists of N stationary SUs trying to find the spectrum opportunities among M frequency channels. The SUs are able to sense one channel at each sensing time and report its status to the BS. Each SU measures the energy in the sensing channel and compares it with a threshold, coming from a standard [11], [23]. It is known that when there is no prior information concerning the PUs' signal, energy detector achieves the best performance [24]. If the measured energy in the m^{th} channel by the n^{th} SU is greater than the threshold, the SU will report the channel as unavailable ($y_{nm}(t) = 0$), otherwise it will report the channel as available ($y_{nm}(t) = 1$). However, due to fading, small-scale shadowing, device failure, or malicious attacks, a device might report faulty measurements on a subset of channels. Although, such binary measurements can be collected from a network of low-complexity inexpensive sensors, it is shown that this method is a robust and effective approach [11], [25].

We also assume that there exists a common control channel (CCC) between the SUs and the BS. The BS assigns the sensing channels using the CCC and synchronizes the SUs to operate on a frame-by-frame structure [26]. At each frame, there exist a quiet interval when the SUs stop their transmission, listen to the assigned channel, and send their measurements to the BS via CCC. Then, by aggregating the sensing data from all the devices, the BS infers the availability of channels and discovers the spectrum opportunities. The spectrum opportunity detection procedure is discussed in detail in Section V.

It is also assumed that an unknown number of licensed PUs are utilizing the bands. Each PU is transmitting on a subset of channels and the activity of the PUs is assumed to be independent of each other. To model the dynamic behavior of the PUs, a two-state Markov chain model is exploited [22]. The PUs alternate between active and inactive states. If the PU is active, it utilizes an unknown subset of channels.

III. Generative Model

Here, the generative model is presented, describing the observations using the hidden variables. A generative model specifies the probability distribution of all the variables, including the observed and hidden variables. In this work, the observed variables are the sensing data collected from the sensors. The hidden variables are parameters that are desired to be inferred from the observations, such as availability of the channels, reliability of devices, and the clusters. The goal of the inference algorithm is to infer the hidden variables given the observations. In Section IV, the details of inference and evolution of distributions over time will be discussed.

Specifically, in our model, the following hidden variables are defined:

1) Channel-specific reliability $u_{nm} \in \{0, 1\}$, which is either 0 or 1 and describes the reliability of measurements of n^{th} SU on m^{th} channel. A device might have different performances on different channels due to frequency selective fading or device failure. If $u_{nm} = 1$ ($u_{nm} = 0$), the device n is reliable (unreliable) on channel m.
2) Device reliability $r_n \in [0, 1]$, denoting the overall trustworthiness of the device. For small values of r_n, the device is more prone to reporting faulty data. A generally reliable device will report trustworthy measurements on most of the channels.
3) Cluster membership $g_n \in \{1, 2, 3, \ldots, L_{max}\}$, indicating the cluster that the n^{th} SU belongs to. Each cluster contains the devices experiencing similar spectrum opportunities. This prior captures the *spatial correlation* among the SUs. We will use l to index the clusters.
4) Channel availability for each cluster $c_{lm} \in [0, 1]$, describing the probability that channel m is available for SUs in cluster l.

Moreover, the observed variable is y_{nm}. The SU n will report $y_{nm} = 1$, if it senses the channel m as available and $y_{nm} = 0$ otherwise. In this model, clustering the SUs helps us to model correlation among the sensing outcomes. It is assumed that there exist clusters of SUs that agree on the availability of different frequency bands. Number of the clusters is assumed to be unknown a priori and will be inferred from the data.

To illustrate the idea, Figure 1 shows the clustered devices in a simple scenario. It is easy to notice that the devices with same cluster membership are more likely to be close to each other. In Section V, this knowledge will be used to avoid assigning same spectrum opportunities to neighboring SUs, to reduce the chance of interference. Also, channel-specific reliability and device reliability model the faulty measurements and devices. Without them, all the observations would be assumed to be reliable and the faulty measurements would easily reduce the accuracy of the inference algorithm. The random variable for channel-specific reliability, i.e., u_{nm}, is used to model the unreliable measurements caused by any frequency selective phenomena, such as fading. It is also worthwhile to mention that although the frequency selective fading differs from PU to PU, the reported measurement from the SU is either true or false for the channel m and this is what we need to draw conclusions on the occupancy status of the channels.

The random variable for device reliability is included in the model to avoid overfitting and to relate the performance of the device over different channels. Without this random variable, the performance of a device on each channel would be independent of other channels. However, it is desired for us to be able to use the inferred information on other channels to predict the performance of the device on a new unseen channel. For instance, if most of the device measurements have been tagged as reliable in previous time slots, it is intuitive to use this information as the prior information on a newly sensed channel. All of the reliability information over all the channels are summarized in the device reliability variable and its distribution.

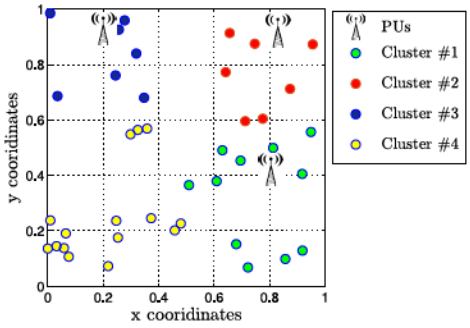


Fig. 1. Outcome of the clustering algorithm in the presence of 3 PUs.

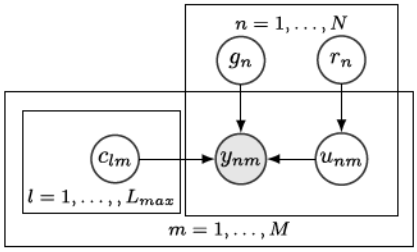


Fig. 2. Graphical representation of the generative model.

Figure 2 illustrates the graphical representation of the generative model, describing the dependencies among the hidden and observed variables. Each white circle represents a hidden variable in our model and the gray circle is the observed variable, which is y_{nm}. For simplicity, each plate groups the variables that repeat together in the model and the sequence inside the plate indicates the number of repetitions. This means that, for example, instead of having N nodes in the graph representing the devices' reliability, one for each device, there exist only one node in the graph denoted by r_n, which is repeated N times.

Since the exact number of clusters is not known, it is assumed that we know an upper bound for the number of clusters L_{max}. In theory, L_{max} can be infinity. However, for practical reasons, an upper bound is set for number of clusters.

The arrows in Figure 2 indicate the dependency among the variables. The observations made by each device on a specific channel depends on its cluster membership, channel availability for the cluster, and its reliability on the channel. Also, according to the model, the observations are independent of the device reliability, given the channel-specific reliability. This is intuitive since if we know the reliability of the device on a specific channel, we do not need the reliability of the device on the other channels or the overall reliability to describe the observations on that channel. The proposed model can be formulated as follows:

$$
\begin{aligned}
g_n &\sim \text{Discrete}(\pi_n) && n = 1, \ldots, N \\
r_n &\sim \text{Beta}\left(b_n^1, b_n^0\right) && n = 1, \ldots, N \\
u_{nm} &\sim \text{Bernoulli}(r_n) && n = 1, \ldots, N, m = 1, \ldots, M \\
c_{lm} &\sim \text{Beta}\left(a_{lm}^1, a_{lm}^0\right) && n = 1, \ldots, N, l = 1, \ldots, L_{max} \\
y_{nm} &\sim u_{nm}\, \text{Bernoulli}(c_{g_n,m}) \\
&\quad + (1 - u_{nm})\, \text{Bernoulli}(1 - c_{g_n,m}) \\
&n = 1, \ldots, N, m = 1, \ldots, M
\end{aligned} \tag{1}
$$

The hidden variable g_n is modeled with a discrete distribution and $\boldsymbol{\pi}_n = [\pi_{n,1}, \pi_{n,2}, \ldots, \pi_{n,L_{max}}]$ contains the cluster membership probabilities for SU n. This means that the distribution Discrete($\boldsymbol{\pi}_n$) generates the value $g_n = l$ with probability $\pi_{n,l}$, i.e., $\mathbb{P}\{g_n = l\} = \pi_{n,l}$ and $\sum_l \pi_{n,l} = 1$ for $n = 1, \ldots, N$. Number of the clusters will be determined by capturing the correlation among the SUs.

The variable for the device reliability, r_n, is modeled with a Beta distribution with parameters b_n^1 and b_n^0. This is the natural choice since r_n is used as the parameter of the Bernoulli distribution of the channel specific reliability and the conjugate prior for a Bernoulli distribution is Beta distribution.

The channel-specific reliability is modeled as a Bernoulli distribution with parameter r_n. This implies that if a device is generally reliable, i.e., r_n close to 1, it is more likely to be reliable on different channels. This prior links the performance of the device on different channels and reduces the chance of overfitting the channel-specific reliability. It is also worthwhile to mention that in the proposed model, we are modeling a symmetric channel. This means that the measurement reliabilities for $y_{nm} = 0$ and $y_{nm} = 1$ are equal. However, our numerical results show that the proposed method works well for non-symmetric channels.

The channel availability is also defined as a random variable sampled from a Beta distribution with parameters a_{lm}^1 and a_{lm}^0. This is also because c_{lm} is later used as the parameter of the Bernoulli distribution that describes the observations.

Finally, the observed variable, y_{nm}, which is the reported occupancy status of channel m by SU n, is defined as the summation of two Bernoulli distributions. In words, if the SU is reliable on a channel, $u_{nm} = 1$, the distribution would be Bernoulli($c_{g_n,m}$). This means that y_{nm} will be sampled from a Bernoulli distribution with true parameter of channel availability, i.e., $c_{g_n,m}$. Otherwise, for $u_{nm} = 0$, it will be sampled from Bernoulli($1 - c_{g_n,m}$).

As mentioned in Section II, activity of the PUs is assumed to follow a two-state Markov chain. It is worthwhile to mention that even though the hidden state sequence, i.e., PUs' status, is first-order Markov, the output process, i.e., true or measured occupancy status of channels for each SU, is not Markov of any order. The true occupancy status for SUs, which is a deterministic function of a Markov chain, does not satisfy the conditions to be Markov [27, Th. 1]. Moreover, the random variable describing the observations, y_{nm}, which is an stochastic function of a Markov chain, follows a Hidden Markov Model (HMM) and is not Markov of any order too [28, Sec. 13.1].

In Bayesian inference, we can marginalize the hidden variables to reduce the model size so that the inference becomes faster. In our work, we found the overall probability of channel availability a better measure to cluster devices than the transition probabilities. Moreover, we are not interested in the hidden states and the transition probabilities of the PUs. Thus, we do not need to include them in the model and we can integrate them out to make the model simpler. The proposed generative model ignores the hidden states and the transition probabilities, but makes the computations simpler and is good enough to infer the parameters of interest.

IV. Inference via Sequential Bayesian Updating

In the Bayesian approach, the goal is to infer the distribution of the hidden variables given the observations, i.e., $\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}|\boldsymbol{y}\}$, which is known as the posterior distribution. For compactness of notation, we set $\boldsymbol{y} = \{y_{nm}\}$, $\boldsymbol{g} = \{g_n\}$, $\boldsymbol{r} = \{r_n\}$, $\boldsymbol{u} = \{u_{nm}\}$, and $\boldsymbol{c} = \{g_{lm}\}$. Using the Bayes rule, the posterior distribution can be written as:

$$\begin{aligned}\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}|\boldsymbol{y}\} &\propto \mathbb{P}\{\boldsymbol{y}|\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}\}\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}\} \\ &\propto \mathbb{P}\{\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}\}.\end{aligned}$$

$\mathbb{P}\{\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}\}$ can be calculated using the model described in (1). Specifically, the last expression in (1), can be used to build $\mathbb{P}\{\boldsymbol{y}|\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}\}$, known as the likelihood of the observations, and the other terms represent our prior belief, $\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}\}$. The posterior distribution is the updated distribution of the hidden variables after receiving the observations.

In sequential Bayesian updating, the prior knowledge of the model is represented as the prior distribution, which is the distribution of the hidden variables before collecting data. After observing the first set of measurements, the posterior distribution is determined using the Bayes rule. Then, the posterior distribution can be used as the prior when the next set of observations becomes available. Thus, the problem boils down to updating the distribution of the hidden variables using the observations. In this approach, all the information is stored in the current distribution of the hidden variables. Hence, old observations and distributions can be ignored. This helps us obtain update rules that are not computationally burdensome. Moreover, since the distribution can be updated using a single measurement of a single SU or a complete set of measurements from all the channels and all the SUs, the BS can easily handle missing entries and different rates of data stream.

In the following, we will present how the distribution of hidden variables are initialized and how they are updated after receiving each set of measurements.

A. Initialization

As mentioned earlier, the initialization of the distribution of the hidden variables reflects our prior knowledge. Thus, there is no universal recipe to initialize the parameters. Also, it is clear that as the BS collects more measurements, the effect of the initialization becomes less and less significant.

The parameter sets a_{lm}^1 and a_{lm}^0 determine the distribution of channel availability for each cluster. In this work, no prior information is assumed on the channels availability and the truth will be inferred completely *a posteriori*. Thus, the parameters are initialized as $a_{lm}^1 = 1$ and $a_{lm}^0 = 1$, $\forall l, m$. This choice of parameters results in a uniform distribution for the probability of the channel availability.

To set the parameters for the device reliability, it is plausible to assume that on average at least half of the measurements are reliable. Also, since there is no information regarding which devices might report faulty data, the parameters of all the

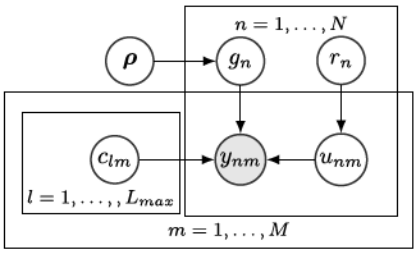


Fig. 3. Graphical model including the stick-breaking prior.

devices should be initialized in a similar manner. In our numerical experiments, we initialize $b_n^1 = 3$ and $b_n^0 = 1, \ \forall l, m$, which means on average 75% of the measurements are reliable. This is because the expected value of a random variable with distribution Beta(b_n^1, b_n^0) is $\frac{b_n^1}{b_n^1+b_n^0}$.

To initialize the parameters for cluster membership, stick-breaking construction is used [29]. The class of stick-breaking priors are widely used in different classification problems, where the complexity of the model (number of the clusters) is unknown. In general, the number of clusters can be either bounded or unbounded. However, it is reasonable and also more practical to assume an upper bound, L_{max}, for the number of candidate clusters. Here, the number of candidate clusters can be assumed to be no larger than the number of devices, i.e., $L_{max} = N$.

The weights of a finite dimensional stick-breaking prior with concentration parameter α can be formulated as $\eta_l = \rho_l \prod_{i=1}^{l-1}(1-\rho_i), \ \forall l$, where ρ_l are independent random variables drawn from Beta$(1, \alpha)$. Setting $\rho_{L_{max}} = 1$ will result in a finite dimensional prior [29]. Thus, the cluster membership distribution is set as $\boldsymbol{\pi}_n = \boldsymbol{\eta}$ for all the devices, where $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_{L_{max}}]$.

Parameter α represents the prior information on degree of correlation among the SUs. Specifically, according to the defined stick-breaking prior, it is known that the probability that two SUs belong to the same group is equal to $\frac{1}{1+\alpha}$ [30]. Thus, as we increase the value of α, less SUs may end up in the same group. For initialization, we set $\alpha = L_{max}$, which indicates a low degree of correlation among SUs.

To make the model more accurate, Figure 3 shows the graphical model including the stick-breaking prior. As a result, the Bayes rules should be rewritten as $\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}|\boldsymbol{y}\} \propto \mathbb{P}\{\boldsymbol{y}|\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}$, where $\boldsymbol{\rho} = \{\rho_l\}$ denotes the set of stick-breaking variables, which is the same for all the SUs. In this model, $\boldsymbol{\rho}$ links the information of cluster membership from different SUs, captures the level of correlation among SUs, and helps us cluster SUs without knowing the number of clusters. Without $\boldsymbol{\rho}$, the variables of cluster membership g_n for different SUs would be independent of each other, which is not the case in our application.

B. Sequential Updating

As the new observations become available, the BS has to update the posterior distributions. Using the model defined in (1) and depicted in Figure 3, the joint distribution of the hidden and observed variables can be presented as:

$$\begin{aligned}&\mathbb{P}\{\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}\\ &= \prod_{n,m} \mathbb{P}\{y_{nm}|g_n, u_{nm}, c_{g_n m}\} \prod_n \mathbb{P}\{g_n|\boldsymbol{\rho}\}\mathbb{P}\{\boldsymbol{\rho}\}\\ &\quad \prod_{n,m} \mathbb{P}\{u_{nm}|r_n\}\mathbb{P}\left\{r_n|b_n^1, b_n^0\right\} \prod_{l,m} \mathbb{P}\left\{c_{lm}|a_{lm}^1, a_{lm}^0\right\}, \quad (2)\end{aligned}$$

Now, given the observations, the goal is to infer the distribution of cluster membership, channel availability, channel-specific reliability, and device reliability, i.e., $\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}|\boldsymbol{y}\}$, in a timely manner. To handle the intractable integrals arising in the inference procedure, *variational inference* is often employed [30]–[35]. In variational inference, the posterior distribution is approximated by a family of distributions, for which the calculations are tractable. The approximate distribution is often assumed to be fully factorized over all the hidden variables. Specifically, the fully factorized variational distribution $\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}$ is defined as:

$$\begin{aligned}&\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}\\ &= \prod_n \mathbb{Q}\left\{g_n|\hat{\boldsymbol{\pi}}_n\right\} \prod_{n,m} \mathbb{Q}\{u_{nm}|\tau_{nm}\}\\ &\quad \prod_n \mathbb{Q}\left\{r_n|\hat{b}_n^1, \hat{b}_n^0\right\} \prod_{l,m} \mathbb{Q}\left\{c_{lm}|\hat{a}_{lm}^1, \hat{a}_{lm}^0\right\} \prod_l \mathbb{Q}\left\{\rho_l|\hat{\gamma}_l^1, \hat{\gamma}_l^0\right\},\end{aligned} \quad (3)$$

where $\hat{\boldsymbol{\pi}}_n$, $\hat{b}_n^1$, $\hat{b}_n^0$, $\hat{a}_{lm}^1$, $\hat{a}_{lm}^0$, τ_{nm}, $\hat{\gamma}_l^1$, and $\hat{\gamma}_l^0$ are the parameters of the factorized distributions. Our goal is to obtain $\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}$ such that it approximates the posterior distribution $\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}|\boldsymbol{y}\}$.

Specifically, in variational inference, the goal is to obtain a distribution $\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}$ that maximizes the likelihood of the observations. It is easy to show that the log-likelihood of the observations can be written as [28, Ch. 10]:

$$\begin{aligned}\ln(\mathbb{P}\{\boldsymbol{y}\}) &\geq \mathcal{L}(\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\})\\ &\quad + \mathrm{KL}(\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}\|\mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}|\boldsymbol{y}\}),\\ \text{where} \quad \mathcal{L}(\mathbb{Q}) &\triangleq \mathbb{E}\{\ln(\mathbb{P}\{\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\})\}\\ &\quad - \mathbb{E}\{\ln(\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\})\},\end{aligned}$$

and the expected value is with respect to variational distribution. KL(.||.) is the Kullback-Leibler (KL) divergence of two distributions. The equality occurs when $\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\} = \mathbb{P}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}|\boldsymbol{y}\}$, which means KL-divergence is zero. This means $\mathcal{L}(\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\})$ is the largest possible lower bound that can be attained. However, since the variational distribution is limited to the family of fully factorized distributions, the maximum of the lower bound cannot be achieved.

Thus, the problem boils down to maximizing the $\mathcal{L}(\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\})$ to find the best approximate posterior distribution. To do that, at each step, the lower bound is maximized with respect to only one of the factorized distributions, i.e., $\mathbb{Q}\{g_n|\hat{\boldsymbol{\pi}}_n\}$, $\mathbb{Q}\{u_{nm}|\tau_{nm}\}$, $\mathbb{Q}\{r_n|\hat{b}_n^1, \hat{b}_n^0\}$, $\mathbb{Q}\{c_{lm}|\hat{a}_{lm}^1, \hat{a}_{lm}^0\}$, or $\mathbb{Q}\{\rho_l|\hat{\gamma}_l^1, \hat{\gamma}_l^0\}$, keeping all the other distributions fixed. Each step results in the update rule for one of the variables. Since $\mathcal{L}(\mathbb{Q}\{\boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\})$ is concave with respect to each of the factorized distributions, convergence is guaranteed [28].

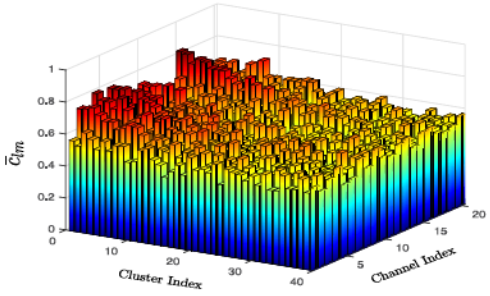


Fig. 4. Expected value of channel availability after 20 time frames.

The derivations of the updating rules are presented in the Appendix.

At each time frame, when the BS receives new observations, it employs the update rules and updates the distribution of the hidden variables. To illustrate the idea, Figure 4 shows the expected value of the channel availability for different clusters, i.e., $\bar{c}_{lm} = \mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{c_{lm}\}$, for a scenario similar to Figure 1. At the first time frame $\bar{c}_{lm}$, is equal to 0.5 for all the channels and the clusters, according to the initialization discussed in Section IV-A, which indicates unbiased estimate of channel availability in absence of further information. It is also worthwhile to mention that since the number of clusters the candidate clusters L_{max} is set to larger than the number of actual clusters, most of the clusters in Figure 4 are empty. Thus, the belief on channel availability of these clusters are not updated. On the hand, it is easy to see that, for the non-empty clusters, as the BS collects more measurements, uncertainty decreases and the spectrum opportunities are revealed.

Moreover, Figure 5 illustrates the expected value of channel availability for devices belonging to different clusters, i.e., $\bar{c}_{g_n m} = \mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\mathbb{E}_{\mathbb{Q}\{g_n\}}\{c_{g_n m}\}$, over time. Dashed lines show the true value of the channel availability for devices in that cluster. In some cases, there is a bias in the final estimation of the channel availability. This bias is partly due to the fact that the posterior distribution is being approximated using a factorized distribution. Also, it is clear that when the true value is closer to the prior belief, i.e., 0.5, the algorithm converges faster. On the other hand, starting from an incorrect prior, the BS needs to collect more measurements to estimate the true value of the channel availability. For instance, Figure 5(d) shows that devices in cluster #4, with true channel availability of 0, take longest to converge.

The sequential update rules allow us to propagate information without requiring to store the old observations or to repeat computations. In next section, the inferred distributions are exploited to assign channels for sensing and to detect the spectrum opportunities.

V. Spectrum Opportunity Discovery

As mentioned earlier, it is assumed that the network is performing on a frame-by-frame structure. At the beginning of each frame, the BS assigns a channel to each SU. Then, if the channel is sensed as empty, the SU utilizes the channel. Moreover, the SU reports the outcome of the sensing to the BS.

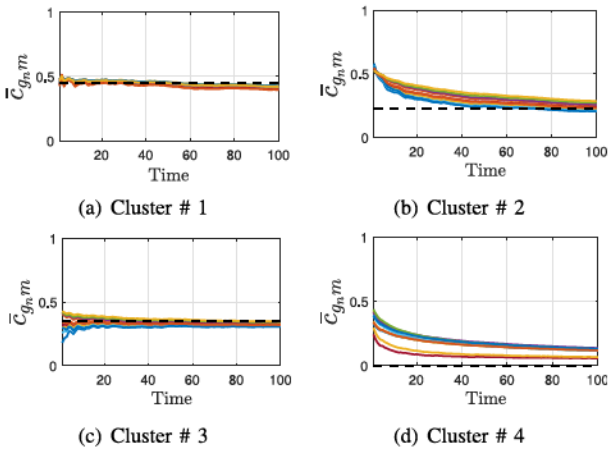


Fig. 5. Expected value of the channel availability for devices in Clusters 1-4 and $m = 1$. Solid lines shows the expected value of the inferred distribution for channel availability of devices in each cluster over time. Dashed line shows the true value of the channel availability for devices in that cluster.

Here, we present the procedure for discovering the spectrum opportunities, using the distribution of the hidden variables.

In order to increase the number of *unique* spectrum opportunities in space and to reduce the risk of interference, same channel should not be appointed to neighboring SUs. Thus, the BS has to assign spectrum opportunities to SUs such that the channels are more likely to be available; and the same channel is not assigned to physically adjacent SUs.

Since the location information of the SUs is not available, cluster membership information is exploited as a measure of adjacency. This is based on the assumption that measurements from neighboring SUs are correlated. Thus, SUs with correlated measurements will end up in the same cluster. Figure 1 shows the outcome of the proposed clustering for a sample environment. In the figure, the n^{th} SU is assigned a cluster l^*, where $l^* = \max_l \mathbb{Q}\{g_n = l\}$. It is easy to notice that the neighboring nodes are more likely to be in the same cluster.

In this section, we present how the spectrum opportunities can be assigned to SUs by mapping the problem to an energy minimization problem [36]. First, we need to quantify the probability that a channel is available for each of the SUs. Let t_{nm} be the true occupancy status of the channel m for device n, which clearly might be different from the observed value y_{nm}. If channel m is available for device n, t_{nm} will be equal to 1, otherwise it will be 0. Then, $\mathbb{Q}\{t_{nm}\}$, which is the inferred variational distribution for t_{nm}, can be calculated as the marginal distribution of $\mathbb{Q}\{t_{nm}, g_n, c_{lm}\} = \mathbb{Q}\{t_{nm}|g_n, c_{lm}\}\mathbb{Q}\{g_n\}\mathbb{Q}\{c_{lm}\}$, given by:

$$\begin{aligned}\mathbb{Q}\{t_{nm}=1\} &= \sum_{l=1}^{L_{max}} \int_{c_{lm}} \mathbb{Q}\{t_{nm}=1|g_n=l, c_{lm}\} \\ &\quad \times \mathbb{Q}\{g_n=l\}\mathbb{Q}\{c_{lm}\}dc_{lm} \\ &\overset{(a)}{=} \sum_{l=1}^{L_{max}} \mathbb{Q}\{g_n=l\} \int_{c_{lm}} c_{lm}\mathbb{Q}\{c_{lm}\}dc_{lm} \\ &\overset{(b)}{=} \sum_{l=1}^{L_{max}} \hat{\pi}_{n,l} \frac{\hat{a}^1_{lm}}{\hat{a}^1_{lm}+\hat{a}^0_{lm}}. \end{aligned} \tag{4}$$

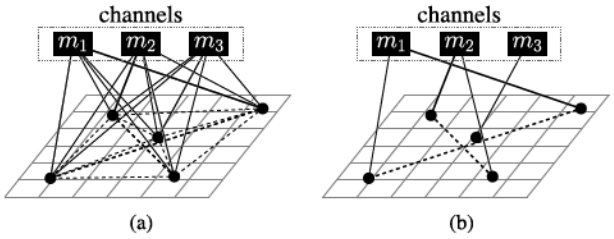


Fig. 6. (a) An example of spectrum opportunity assignment using multi label energy minimization. The thicker lines represent the edges with larger weights. (b) A multiway cut on the graph consisting of the channels and the SUs. In (a), each SU is connected to all the other SUs and all the channels. A multiway cut divides the SUs into disjoint groups and assigns each group a unique channel.

Here, (a) uses the fact that $\mathbb{P}\{t_{nm}=1|g_n=l, c_{lm}\}=c_{lm}$. This comes from the definition of the c_{lm}, which is the probability of channel m being available for devices in cluster l. Moreover, (b) exploits the fact that $\mathbb{Q}\{c_{lm}\}$ is a Beta distribution with parameters $\hat{a}^1_{lm}$ and $\hat{a}^0_{lm}$, which is shown in the Appendix.

Now, let $h_n \in \{1,\ldots,M\}$ be the channel detected as the spectrum opportunity for device n and $\boldsymbol{h}=\{h_1,\ldots,h_N\}$ be the set of spectrum opportunities assigned to all of the SUs. The goal of the BS is to find the best set of channels $\boldsymbol{h}^*=\{h^*_1,\ldots,h^*_N\}$, which are most likely to be *unique spectrum opportunities*. For that, the BS has to ensure that the h^*_n is available for device n and is not assigned to its neighbors.

Accordingly, the following objective function can be defined:

$$\boldsymbol{h}^*=\arg\max_{\boldsymbol{h}}\prod_{n=1}^{N}\mathbb{P}\{t_{nh_n}=1\}\prod_{\substack{\forall n_i,n_j\\ h_{n_j}\neq h_{n_i}}}\mathbb{P}\{g_{n_i}=g_{n_j}\}. \quad (5)$$

The first term is the probability that all the assigned channels are available. The second term indicates the probability that the SUs in the same cluster are assigned to different channels. This way, the BS makes sure that the SUs that are spatially close to each other will be assigned different channels. Using the negative log-likelihood, the objective can be further rewritten as:

$$\boldsymbol{h}^*=\arg\min\sum_{n=1}^{N}-\log\big(\mathbb{P}\{t_{nh_n}=1\}\big)+\sum_{n_i=1}^{N}\sum_{n_j=1}^{N}-\log\big(\mathbb{P}\{g_{n_i}=g_{n_j}\}\big)\delta\big(h_{n_j}\neq h_{n_i}\big). \quad (6)$$

$\delta(h_{n_j}\neq h_{n_i})$ is equal to 1, when $h_{n_j}\neq h_{n_i}$ and 0 other wise. It is known that the optimization problems with similar cost function as the problem formulated in (6) can be solved via graph cuts [36]–[39].

Graph cuts are often used in computer vision to assign each pixel of an image a label, while ensuring that the similar pixels are assigned the same label [37], [38]. Likewise, in our problem, the BS wants to assign each SU a channel, while assigning the nonadjacent SUs the same channel.

A cut divides a graph into disjoint sub-graphs by removing the edges of the graph. A minimum graph cut $\mathcal{C}$ is a cut that minimizes the cost of the cut $|\mathcal{C}|$, which is defined as the sum of weights of edges removed by the cut.

As an example, consider the simple network depicted in Figure 6(a). This network consists of 5 SUs, shown by circles, and 3 channels, represented by squares. This network can be represented as a graph of 8 vertices, containing one vertex for each SU and one vertex for each channel. The edge between each pair of SUs, denotes the probability that this pair of SUs does not belong to the same cluster. Thus, if a pair of SUs are likely to be in the same cluster, the weight of the edge will be small and it will be more likely to be removed by the graph cut algorithm. Moreover, the weight of the edge that links SU n to channel m represents the probability that channel m is not available for device n. Hence, if a channel is available for an SU, it will not easily be removed from the graph.

In Figure 6(b), an example of channel assignment is depicted. The graph is partitioned into three disjoint subsets, one for each channel. Such partitioning is achieved by removing edges of the original graph via multi-label graph cut.

Specifically, to minimize (6), each channel-SU edge is weighted by $K+\log(\mathbb{P}\{t_{nh_n}=1\})$, where K is a constant greater than $\max\{\log(\mathbb{P}\{t_{nh_n}=1\})\}, \forall n$, to make the weights positive. This is because removing an edge with negative weight decreases the cost of the cut. Moreover, each SU-SU edge is weighted by $-\log(\mathbb{P}\{g_{n_i}=g_{n_j}\})$.

Let h^*_n be the channel assigned to SU n. Using the defined weights, the cost of removing all the channel-SU edges to SU n, except the edge to h^*_n, would be $\sum_{m\neq h^*_n}K+\log(\mathbb{P}\{t_{nm}=1\})$. Similarly, the cost of removing edges between SU n and all the SUs that are not assigned the same channel is: $\sum_{\substack{\forall n'\\ h^*_n\neq h^*_{n'}}}-\log(\mathbb{P}\{g_n=g_{n'}\})$. Hence, the total cost of the cut can be written as:

$$|\mathcal{C}|=\sum_{n=1}^{N}\sum_{m\neq h^*_n}K+\log(\mathbb{P}\{t_{nm}=1\})-\sum_{n_i=1}^{N}\sum_{n_j=1}^{N}\log\big(\mathbb{P}\{g_{n_i}=g_{n_j}\}\big)\delta\Big(h^*_{n_j}\neq h^*_{n_i}\Big). \quad (7)$$

This cost can be further rewritten as:

$$|\mathcal{C}|=N(M-1)K+\sum_{n=1}^{N}\sum_{m=1}^{M}\log(\mathbb{P}\{t_{nm}=1\})-\sum_{n=1}^{N}\log\big(\mathbb{P}\{t_{nh^*_n}=1\}\big)-\sum_{n_i=1}^{N}\sum_{n_j=1}^{N}\log\big(\mathbb{P}\{g_{n_i}=g_{n_j}\}\big)\delta\Big(h^*_{n_j}\neq h^*_{n_i}\Big). \quad (8)$$

Since, the first two terms do not depend on the cut, a set $\boldsymbol{h}^*$ that minimizes the cost of the cut $|\mathcal{C}|$ will also minimize (6), which is equivalent to maximizing the objective function in (5). Thus, after inferring the distributions of the hidden variables, the BS can perform a graph cut algorithm, using the defined graph and weights, to find the spectrum opportunities.

VI. Numerical Results

In this section, the simulation results are presented to quantify the performance of Co-SpOT. The simulation parameters are set as follows, unless otherwise is stated. By default, we consider a square area with size 50 (distance unit)^2, with $N = 15$ stationary SUs and 4 stationary PUs uniformly at random distributed over the space. For simplicity, it is assumed that a channel will be unavailable if the SU is in the circular transmission range of a PU that is utilizing the channel. Each SU device is in the interference range of a PU or SU if they are closer than 2 distance units. Using such parameters, each SU has to share the spectrum with about 2.7 other SUs and 0.8 PUs on average.

We set the number of spectrum bands to 20 ($M = 20$) and each PU is transmitting on 6 channels, randomly selected at the beginning of the simulation. To model the PU's behavior, a two-state Markov chain is adapted. The PUs utilize the channels $\lambda = 0.5$ of the time and switch from inactive state to active state with probability 0.1. These two parameters determine the remaining transition probability.

At each time frame, SUs can sense and report the occupancy status of only one channel, which is specified by the BS. Furthermore, probabilities of false-alarm and miss-detection are set to be 0.2, i.e., $P_f = 0.2$ and $P_m = 0.2$. Digham *et al.* [40] derived closed form expressions for probabilities of false-alarm and miss-detection of an energy detector over AWGN, Rayleigh, Rician, and Nakagami-m channels. For simplicity, and without loss of generality, we use P_m and P_f to model the adverse effect of different channel models. The BS collects the measurements, updates the posterior distributions, discovers the spectrum opportunities, and assigns channels for sensing in the next frame.

The performance of Co-SpOT is compared with some existing non-cooperative and cooperative methods such as *greedy adaptive learning* [24], *exponential-weight algorithm for exploration and exploitation* (Exp3) [41], and *cluster-based coordinated multiband spectrum sensing* (Cluster-CMSS) [22]. The code for the Cluster-CMSS method was provided by the authors. The performances are compared according to a defined time-averaged normalized success rate (TNSR). Success rate indicates what ratio of discovered channels is in fact available and unique in a neighborhood of SUs. By availability, we mean that at the time of utilization the SU will not interfere with the communication of any of the PUs. Moreover, only *unique* available channels are considered to be spectrum opportunity. Thus, if an available channel is detected as a spectrum opportunity for several SUs, which are in interference range of each other, this spectrum opportunity is considered as spectrum opportunity only for one of the SUs. So, both primary user protection and secondary user protection are considered in the performance metric.

The success rate is averaged over $T = 100$ time slots and normalized by the maximum achievable success rate. Color-sensitive graph coloring (CSGC), introduced in [13], is employed to find the maximum achievable success rate. Location of all the SUs and the true occupancy status of channels are provided as the inputs of CSGC. Moreover, the algorithm introduced in [36] is exploited to find the minimum graph cut and discover the spectrum opportunities.[2]

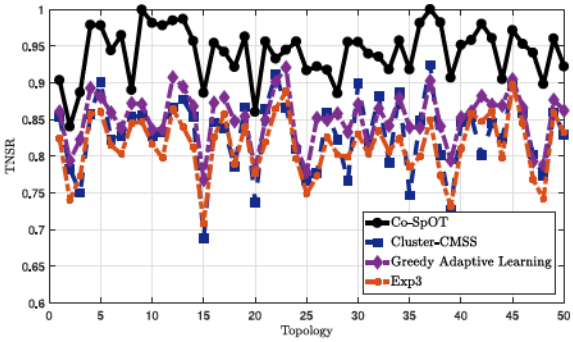


Fig. 7. Performance of different frameworks for various topologies.

TABLE I
Performance of Different Spectrum Opportunity Detection Algorithms

Method	Mean	SD	RSD (%)
Co-SpOT	**0.94**	**0.036**	**3.83**
Cluster CMSS [22]	0.83	0.050	6.15
Greedy Adaptive Learning [24]	0.85	0.035	4.13
Exp3 [41]	0.81	0.042	5.18

Choosing success rate as the performance metric makes the results as general as possible and is a common practice [13], [22], [24]. A higher TNSR leads to improved throughput of individual SUs, overall throughput of the network, packet loss rate, and transmission delay. It is also worthwhile to mention that the energy cost of Co-SpOT is same as the centralized competitors. Moreover, for a fair comparison, number of observations collected by the BS at each time slot is the same for all the algorithms, i.e., one channel per SU. This means that the cost of sensing is the same for all the approaches.

In our first numerical experiment, we start by examining the performance of the mentioned methods for different network topologies. Using the default setting defined above, Figure 7 shows the performance of different algorithms for 50 different topologies generated randomly. It is easy to notice that Co-SpOT outperforms the competitors for all the cases, which illustrates the fact that Co-SpOT is not sensitive to the network topology and is able to perform well for different topologies. To quantify the performance, Table I shows the mean, standard deviation, and relative standard deviation of different methods. The table illustrates that Co-SpOT has a low variation, while having a high average TNSR.

Figure 8 illustrates the performance of different methods for different values for probability of false-alarm P_f and probability of miss-detection P_m. As mentioned earlier, parameters of different channel models, such as AWGN, Rayleigh, Rician, and Nakagami-m, can be translated into P_m and P_f, using closed form expressions [40]. Thus, Figure 8 can be used to evaluate different methods for different channel conditions. It is clear that Co-SpOT is less sensitive to P_f than P_m. As P_m

[2] The pseudo-code for the graph cut algorithm can be found in [36] and the code is available at http://vision.ucla.edu/ brian/gcmex.html.

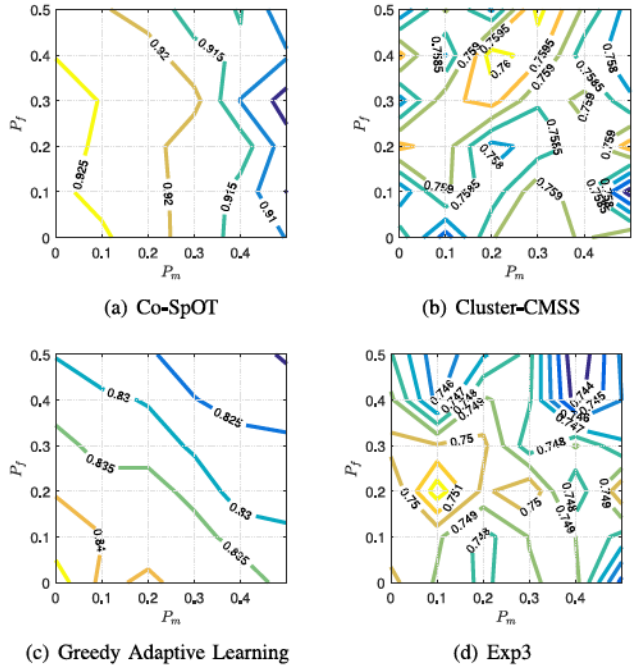


Fig. 8. Spectrum opportunity detection performance for different false-alarm and miss-detection probabilities.

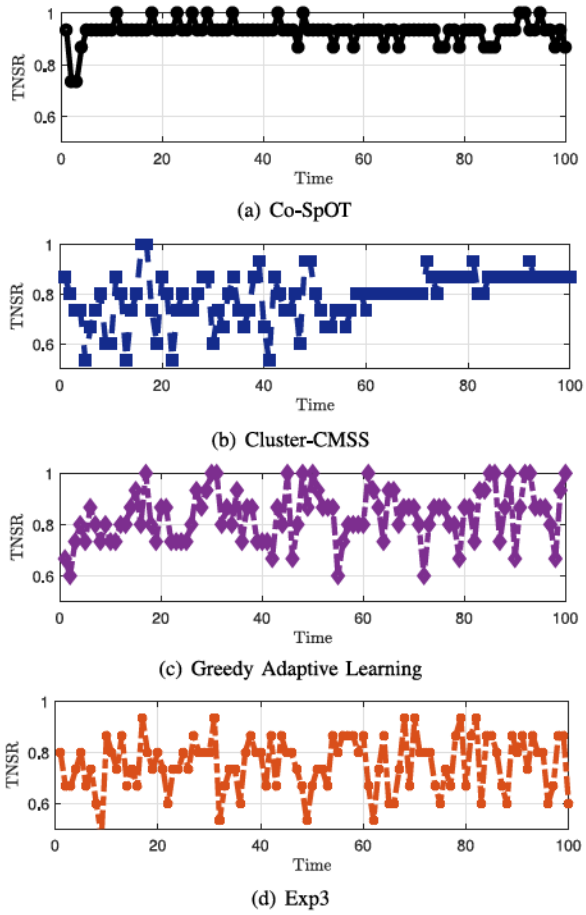


Fig. 9. TNSR over time for (a) Co-SpOT, (b) Cluster-CMSS, (c) greedy adaptive learning, and (d) Exp3.

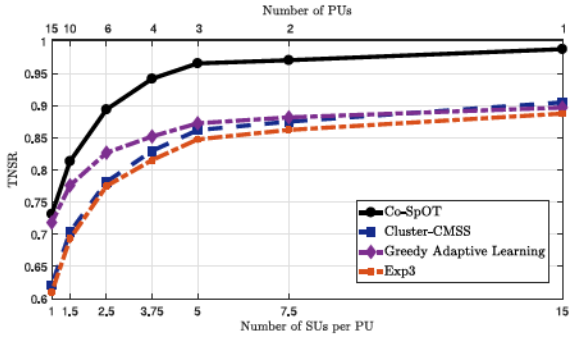


Fig. 10. Spectrum opportunity detection performance for different levels of spatial correlation for $N = 15$. Performance gain is achieved when the number SUs is greater than the number of PUs, i.e., there exist some level of correlation among observations.

increases, the probability of accessing an unavailable channel increases, which leads to poor TSNR. Moreover, although the model is developed for symmetric channels, i.e., with same measurement reliability for $y_{nm} = 0$ and $y_{nm} = 1$, Figure 8 shows that Co-SpOT works well for non-symmetric channels with $P_f \neq P_m$.

Figure 9 shows the performance of different methods over time for topology #7 in Figure 7. After only a few observations, Co-SpOT is able to achieve a high success rate. This indicates that the learning period is short. Comparing with Figure 5, it is easy to notice that the success rate of Co-SpOT converges much faster than the inference algorithm. This means that after receiving a few measurements, we can find the spectrum opportunities with good accuracy, although the probability distributions have not yet converged to their final states. Figure 9(b) shows that Cluster-CMSS also converges to a decision after about 60 time slots.

To cluster the SUs and to draw conclusions about the availability of the channels, Co-SpOT exploits spatial correlation among the observations. This means that, to achieve a performance gain, there should exist some level of spatial correlation in the network. Since the activity of the PUs are assumed to be uncorrelated, the SUs sensing the same PUs will report similar observations. Thus, as the number of SUs per each PU increases, the observations will be more correlated. Figure 10 illustrates the performance gain for different levels of spatial correlation, averaged over sufficiently large Monte Carlo trials. In this experiment, for a fixed number of SUs, i.e., $N = 15$, number of PUs is decreased from 15 to 1.

The effect of decreasing the number of PUs is two-fold. First, by decreasing the number of PUs, more channels become available and it is easier to discover and assign the spectrum opportunities. This explains the improvement in the performance of all the methods. Also, for fewer number of PUs, the observations of the SUs are more correlated. Hence, the performance gap between Co-SpOT and other methods becomes increasingly larger as the number of PUs decreases. When the number of PUs is equal to the number of SUs, meaning that on average we have a PU for each SU, the observations are not correlated. Therefore, no performance gain is achieved by considering the correlation among the SU. However, as soon as the number of SUs for each PU becomes greater than

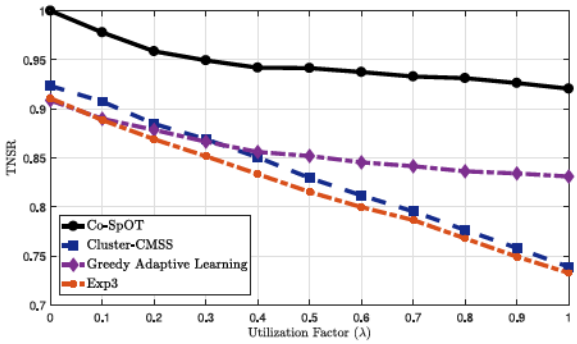


Fig. 11. Spectrum opportunity detection performance for different utilization factors, i.e., λ, in presence of 4 PUs for environment size of 50 ($\sqrt{50} \times \sqrt{50}$) and interference range of 2.

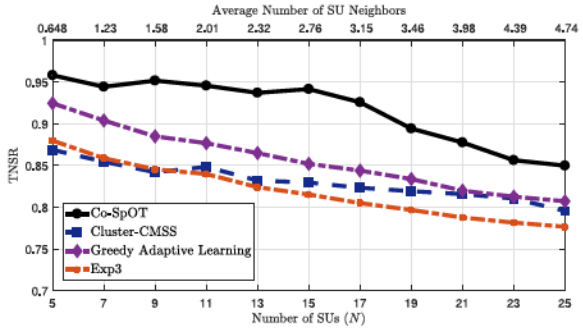
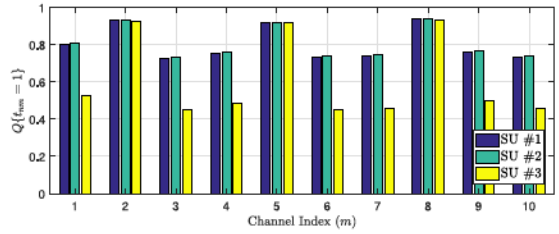
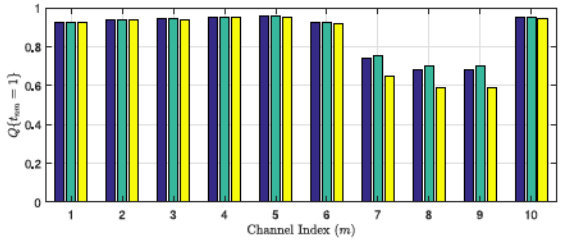


Fig. 12. Spectrum opportunity detection performance with varying number of secondary users. There are 4 PUs, each transmitting on a subset of 6 channels out of $M = 20$ channels.



(a) Channels #2, #5, and #8 are available. TNSR = 0.90



(b) Channels #7, #8, and #9 are unavailable. TNSR = 0.99.

Fig. 13. Expected value of channel availability for all the devices after $T = 100$ time slots. SUs #1 and #2 are shadowed and sense all the channels as available.

1, Co-SpOT is able to employ the correlation to improve the spectrum opportunity detection performance, without knowing the location of SUs.

To study the influence of the utilization factor on the performance of the algorithm, Figure 11 compares the success rate of different algorithms for different utilization factors, i.e., λ. As it was expected, due to spectrum opportunity scarcity, the success rate of all the methods deteriorates for higher utilization factors. It is also evident that Co-SpOT is only method that achieves an optimal performance for the best-case scenario, i.e., $TNSR = 1$, for $\lambda = 0$. This is because Co-SpOT is able to exploit the spatial correlation to avoid interference among the SUs. In such scenario, all the SUs are in the same cluster, due to similarity of the observations. Thus, a unique spectrum channel is assigned to each of them.

Density of the SU devices is also an important factor in the performance of any spectrum opportunity detection framework. For larger number of SUs, the fusion center receives more data, which leads to better decision making performance. However, as the number of SUs increases, for a fixed size of environment and a fixed transmission range, the scarcity of unique spectrum opportunities becomes more considerable, degrading the spectrum utilization.

Figure 12 shows the effect of these two opposing forces. For $N \leq 15$, incrementing the number of SUs does not decrease the performance of Co-SpOT significantly. In fact, the performance gap keeps increasing. However, for $N > 15$, the unique spectrum opportunities become too scarce and more measurements does not help the BS to find unique spectrum opportunities. As an example, for $N = 25$, using the default environment size of $\sqrt{50}$ distance unit ($\approx 3.5\times$ interference range), each SU is sharing the spectrum channels with about 4.74 other SUs and 0.8 PUs, which are transmitting on 6 channels. Figure 12 shows that for large number of SUs, the success rate of different spectrum opportunity detection methods decreases.

Unavailable location information puts certain limits on the performance of any cooperative spectrum opportunity detection algorithm. Two SUs might be far away from each other, while experiencing the exact same spectrum opportunities. Thus, in the absence of additional side information, there is no way to handle such scenario and have a perfect spectrum reuse. This issue is particularly of more importance in large-scale networks with many SUs. For such networks, exploiting a single BS to aggregate the data from the entire network reduces the success rate. Hence, data form each region of the network should be processed by a local BS, also referred to as cluster head.

Finally, to understand the performance of Co-SpOT under shadowing, we investigate a simple, but important, scenario. This setting gives us insight to understand the behavior of Co-SpOT for complicated scenarios. We consider an environment consisting of 3 SUs and 1 PU, transmitting on a subset of 10 channels. All the SUs are in the interference range of the PU and each other. However, due to shadowing, 2 of the SUs (SUs #1 and #2) cannot hear the PU and sense all the channels as available. As before, $P_m = P_f = 0.2$.

Figure 13 illustrates the expected value of channel availability for the SUs, i.e., $\mathbb{Q}\{t_{nm} = 1\}$, after $T = 100$ time slots. In this simulation, to make the probability distributions more interpretable, we set $L_{max} = 2$, $\lambda = 1$, and observations are collected on all the channels. Figure 13(a) shows $\mathbb{Q}\{t_{nm} = 1\}$ for the case when the PU is transmitting on 7 channels.

TABLE II
CLUSTER MEMBERSHIP DISTRIBUTIONS ($\hat{\pi}_{n,l} = \mathbb{Q}\{g_n = l\}$).

	SU #1	SU #2	SU #3
Cluster #1	0.91	0.93	0.02
Cluster #2	0.09	0.07	0.98

As shown, the expected channel availabilities for busy channels of the shadowed SUs are less than those of available channels, although all the channels are sensed as available by the shadowed users. This means that the shadowed users can benefit from the measurements of the non-shadowed user. This is due to the fact that, in the proposed method, SUs are assigned to the clusters in a probabilistic manner. Table II shows the inferred cluster membership probabilities in the same scenario as Figure 13(a).

In such scenario, when the SUs disagree on most of the channels, they belong to different clusters with very high probability, but not with probability 1. This probabilistic cluster membership leads to information sharing in two ways. First, the expression for $\mathbb{Q}\{t_{nm} = 1\}$, Equation (4), can be seen as a weighted average of the beliefs of different clusters, where the weights are the cluster membership probabilities, i.e., $\hat{\pi}_{n,l}$. Thus, in this example, the belief of SU #2 is updated using the beliefs of both of the clusters, with different weights. On the other hand, to update the beliefs of each cluster, the measurements from all the devices are exploited, with different weights (Appendix A). This means that the measurements of SU #2 are used to update the channel availability of both clusters.

Thus, the information is shared among all the devices. This is why shadowed SUs have different beliefs on different channels, although they sense all of them as available. Moreover, this is the reason that expected channel availabilities for SU #3 have large values ($\mathbb{Q}\{t_{nm} = 1\} > 0.4, \forall m, n = 3$), even for the channels it senses as unavailable. This information sharing helps the spectrum opportunity detection algorithm to find the available channels and to achieve TNSR of 0.90.

Figure 13(b) shows $\mathbb{Q}\{t_{nm} = 1\}$ for a scenario where the PU transmits on only 3 channels. In this scenario, since the devices agree on the status of most of the channels, they will be assigned to the same cluster with a higher probability, compared to Figure 13(a), and will end up having similar beliefs on the channels. Specifically, $\mathbb{Q}\{g_n = 1\}$ is equal to 0.75, 0.82, and 0.27 for SU #1, #2, and #3, respectively. Again, the spectrum opportunity detection algorithm is able to find the available channels for all the SUs and achieves TNSR of 0.99.

VII. CONCLUSION

In this paper, we considered the problem of discovering the spectrum opportunities in spectrum-heterogeneous networks. We proposed a centralized, cooperative, Bayesian inference framework, referred to as Co-SpOT, to extract information from the measurements, assuming the presence of faulty data and correlation among the measurements of different SUs. A sequential updating formulation is utilized and the update rules are derived mathematically. Then, it is shown that multi-label graph cut can be employed to discover the unique spectrum opportunities. The simulation results suggest that Co-SpOT outperforms the existing cooperative and non-cooperative methods, whenever the measurements of different SUs are spatially correlated.

APPENDIX

In this section, the derivations of the update rules for the inference algorithm are presented. As discussed in Section IV-B, the posterior distribution is approximated by a family of distributions for which the calculations are tractable. We restrict the distribution by fully factorizing it over all the hidden variables. This method is referred to as *naive mean field* approach [31].

For simplicity of notation, let us denote the whole set of hidden variables with $\boldsymbol{Z} = \{\{g_n\}, \{r_n\}, \{u_{nm}\}, \{c_{lm}\}, \{\rho_l\}\}$. In (3), $\boldsymbol{Z}$ is divided into disjoint groups $Z_i, i = 1, \ldots,$ where each Z_i is representing one of the hidden variables in $\boldsymbol{Z}$. By maximizing the lower bound $\mathcal{L}(\mathbb{Q}\{\boldsymbol{Z}\})$, the variational distribution of each partition $\mathbb{Q}\{\boldsymbol{Z}_i\}$ is given by [28, Ch. 10]:

$$\ln(\mathbb{Q}\{\boldsymbol{Z}_i\}) = \mathbb{E}_{j\neq i}\{\ln(\mathbb{P}\{\boldsymbol{y}, \boldsymbol{Z}\})\} + const, \tag{9}$$

where $\mathbb{E}_{j\neq i}\{.\}$ is the expectation with respect to distributions $\mathbb{Q}\{\boldsymbol{Z}_j\}, j \neq i$. Then by plugging in $\mathbb{P}\{\boldsymbol{y}, \boldsymbol{Z}\} = \mathbb{P}\{\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\}$ from (2) and employing the exponential form of the distributions, the variational distributions can be obtained. The constant value is determined by normalizing the distribution.

It is worthwhile to state that if $x \sim \text{Bernoulli}(p)$, then

$$\ln(\mathbb{P}\{x\}) = \ln\left(\frac{p}{1-p}\right)x + \ln(1-p) \tag{10}$$

and if $x \sim \text{Beta}(b^1, b^0)$, we have

$$\begin{aligned}
\ln(\mathbb{P}\{x\}) &= \left(b^1 - 1\right)\ln(x) + \left(b^0 - 1\right)\ln(1-x) + const \\
\mathbb{E}\{\ln(x)\} &= \psi\left(b^1\right) - \psi\left(b^1 + b^0\right), \\
\mathbb{E}\{\ln(1-x)\} &= \psi\left(b^0\right) - \psi\left(b^1 + b^0\right),
\end{aligned} \tag{11}$$

where $\psi(.)$ is digamma functions. Here, using the observations $\boldsymbol{y}$ and the prior distribution, the update rules to obtain the approximate posterior distributions are presented.

A. Channel Availability

Using (9), to update the channel availability, we have:

$$\ln(\mathbb{Q}\{c_{lm}\}) = \mathbb{E}\{\ln(\mathbb{P}\{\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\rho}\})\} + const.$$

Employing (10) and (11) and integrating out all the variables except c_{lm}, we will have:

$$\begin{aligned}
&\ln(\mathbb{Q}\{c_{lm}\}) \\
&= const + \ln\left(\mathbb{P}\left\{c_{lm} | a^1_{lm}, a^0_{lm}\right\}\right), \\
&\quad + \sum_n \mathbb{E}_{\mathbb{Q}\{g_n\}}\mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{\ln(\mathbb{P}\{y_{nm} | u_{nm}, g_n, c_{lm}\})\} \\
&= const + \left(a^1_{lm} - 1\right)\ln(c_{lm}) + \left(a^0_{lm} - 1\right)\ln(1 - c_{lm}) \\
&\quad + \sum_n \hat{\pi}_{n,l}\Bigg[\mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}\left(\ln\left(\frac{c_{lm}}{1-c_{lm}}\right)y_{nm} + \ln(1-c_{lm})\right) \\
&\qquad + \left(1 - \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}\right)\left(\ln\left(\frac{1-c_{lm}}{c_{lm}}\right)y_{nm} + \ln(c_{lm})\right)\Bigg].
\end{aligned}$$

In this expression, the summations are taken over the devices that have reported a measurement on m^{th} device. The first two terms come from the prior knowledge on the channel status and are written using (11). Moreover, the summation is aggregating the information from observations, considering the reliability of each measurement and the cluster membership of each device. The terms inside the summation are written using the exponential form of Bernoulli distribution, as described in (10).

This expression can be further written in the form of $(\hat{a}^1_{lm} - 1)\ln(c_{lm}) + (\hat{a}^0_{lm} - 1)\ln(1 - c_{lm}) + const$, which is a Beta distribution with parameters:

$$\begin{aligned}\hat{a}^1_{lm} &= a^1_{lm} + \sum_n \hat{\pi}_{n,l}\big[\mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}y_{nm} \\ &\quad + \big(1 - \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}\big)(1 - y_{nm})\big] \\ \hat{a}^0_{lm} &= a^0_{lm} + \sum_n \hat{\pi}_{n,l}\big[\mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}(1 - y_{nm}) \\ &\quad + \big(1 - \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}\big)y_{nm}\big].\end{aligned}$$

Here, $\mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{.\}$ is expectation with respect to $\mathbb{Q}\{u_{nm}\}$ and $\mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}$ can be calculated using $\mathbb{Q}\{u_{nm}\}$, which will be discussed shortly.

B. Device Reliability

Similarly, to update the device reliability, for all n, we have:

$$\begin{aligned}\ln(\mathbb{Q}\{r_n\}) &= const + \left(b^1_n - 1\right)\ln(r_n) + \left(b^0_n - 1\right)\ln(1 - r_n) \\ &\quad + \sum_m \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\left\{\ln\left(\frac{r_n}{1 - r_n}\right)u_{nm} + \ln(1 - r_n)\right\} \\ &= const + \ln(r_n)\left(b^1_n + \sum_m \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\} - 1\right) \\ &\quad + \ln(1 - r_n)\left(b^0_n + \sum_m\big[1 - \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}\big] - 1\right),\end{aligned}$$

By comparing this expression to the exponential form of the Beta distribution, it is easy to see that $\mathbb{Q}\{r_n\}$ is a Beta distribution with parameters $\hat{b}^1_n = b^1_n + \sum_m \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}$ and $\hat{b}^0_n = b^0_n + M - \sum_m \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\}$.

C. Channel-Specific Reliability

Again, by integrating out all the variables except u_{nm}:

$$\begin{aligned}\ln(\mathbb{Q}\{u_{nm}\}) &= const + \mathbb{E}_{\mathbb{Q}\{r_n\}}\{\ln(\mathbb{P}\{u_{nm}|r_n\})\} \\ &\quad + \mathbb{E}_{\mathbb{Q}\{g_n\}}\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(\mathbb{P}\{y_{nm}|u_{nm}, g_n, c_{lm}\})\}\end{aligned}$$

By employing (10) and (11), we have:

$$\begin{aligned}\ln(\mathbb{Q}\{u_{nm}\}) &= const + u_{nm}\mathbb{E}_{\mathbb{Q}\{r_n\}}\{\ln(r_n)\} \\ &\quad + (1 - u_{nm})\mathbb{E}_{\mathbb{Q}\{r_n\}}\{\ln(1 - r_n)\} \\ &\quad + u_{nm}\Bigg[y_{nm}\sum_l \hat{\pi}_{n,l}\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(c_{lm})\} \\ &\quad + (1 - y_{nm})\sum_l \hat{\pi}_{n,l}\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(1 - c_{lm})\}\Bigg] \\ &\quad + (1 - u_{nm})\Bigg[y_{nm}\sum_l \hat{\pi}_{n,l}\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(1 - c_{lm})\} \\ &\quad + (1 - y_{nm})\sum_l \hat{\pi}_{n,l}\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(c_{lm})\}\Bigg].\end{aligned}$$

This update rule, like the other ones, boils down to simple expressions, as the observations are either 0 or 1. It is intuitive that, at any time frame, the BS cannot update u_{nm}, if device n has not reported a measurement on channel m. Thus, the update rule is employed for each pair of SUs and channels that the BS has received a new observation.

To update the distribution, the expression is evaluated for $u_{nm} = 0$ and $u_{nm} = 1$. Since it is shown that $\mathbb{Q}\{c_{lm}\}$ and $\mathbb{Q}\{r_n\}$ are Beta distributions, $\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(c_{lm})\}$, $\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(1 - c_{lm})\}$, $\mathbb{E}_{\mathbb{Q}\{r_n\}}\{\ln(r_n)\}$, and $\mathbb{E}_{\mathbb{Q}\{r_n\}}\{\ln(1 - r_n)\}$ can be calculated using (11). The summations are comparing the observations with the expected value of y_{nm}, coming from the model, and soft counting the agreements and disagreements.

After normalizing the probabilities to have a valid Bernoulli distribution, the parameter of the distribution can be updated as $\tau_{nm} = \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\{u_{nm}\} = \mathbb{Q}\{u_{nm} = 1\}$.

D. Stick-Breaking Variables

Using (9), the update rule can be derived as:

$$\begin{aligned}\ln(\mathbb{Q}\{\rho_l\}) &= const + \ln\Big(\mathbb{P}\Big\{\rho_l|\gamma^1_l, \gamma^0_l\Big\}\Big) \\ &\quad + \sum_n \mathbb{E}_{\mathbb{Q}\{g_n\}}\{\ln(\mathbb{P}\{g_n|\boldsymbol{\rho}\})\}.\end{aligned}$$

Notice that, as mentioned in Section **IV-A**, $\mathbb{P}\{\rho_l|\gamma^1_l, \gamma^0_l\}$ is initialized with Beta$(1, \alpha)$. $\mathbb{P}\{g_n|\boldsymbol{\rho}\}$ can be written in terms of $\boldsymbol{\rho}$, by utilizing the stick-breaking construction rules. Also, $\mathbb{P}\{\rho_l|\gamma^1_l, \gamma^0_l\}$ can be rewritten using (11). Thus, we have:

$$\begin{aligned}\ln(\mathbb{Q}\{\rho_l\}) &= const + \left(\gamma^1_l - 1\right)\ln(\rho_l) + \left(\gamma^0_l - 1\right)\ln(1 - \rho_l) \\ &\quad + \sum_n \sum_{i=1}^{L_{max}} \hat{\pi}_{n,i}\left[\ln(\rho_i) + \sum_{j=1}^{i-1}\ln(1 - \rho_j)\right],\end{aligned}$$

which can be written in form of $(\hat{\gamma}^1_l - 1)\ln(\rho_l) + (\hat{\gamma}^0_l - 1)\ln(1 - \rho_l) + const$. This is also a Beta distribution with parameters:

$$\begin{aligned}\hat{\gamma}^1_l &= \gamma^1_l + \sum_n \hat{\pi}_{n,l}, \\ \hat{\gamma}^0_l &= \gamma^0_l + \sum_n \sum_{i=l+1}^{L_{max}} \hat{\pi}_{n,i}.\end{aligned}$$

E. Cluster Membership

The probability of device n belonging to cluster l, i.e., $\hat{\pi}_{n,l}$, is updated using this expression:

$$\begin{aligned}\ln(\mathbb{Q}\{g_n = l\}) &= const + \mathbb{E}_{\mathbb{Q}\{\boldsymbol{\rho}\}}\{\ln(\mathbb{P}\{g_n = l|\boldsymbol{\rho}\})\}, \\ &\quad + \sum_m \mathbb{E}_{\mathbb{Q}\{u_{nm}\}}\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(\mathbb{P}\{y_{nm}|u_{nm}, g_n = l, c_{lm}\})\}.\end{aligned}$$

Again, by using stick-breaking construction rules and (10):

$$\begin{aligned}\ln(\mathbb{Q}\{g_n = l\}) = const &+ \mathbb{E}_{\mathbb{Q}\{\rho_l\}}\{\ln(\rho_l)\} \\ &+ \sum_{i=1}^{l-1} \mathbb{E}_{\mathbb{Q}\{\rho_l\}}\{\ln(1-\rho_l)\} \\ &+ \sum_{m} \big[\tau_{nm}\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(c_{lm})y_{nm} \\ &+ \ln(1-c_{lm})(1-y_{nm})\} \\ &+ (1-\tau_{nm})\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(1-c_{lm})y_{nm} \\ &+ \ln(c_{lm})(1-y_{nm})\}\big],\end{aligned}$$

where, the summation is over the channels that device n has reported a measurement. $\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(c_{lm})\}$, $\mathbb{E}_{\mathbb{Q}\{c_{lm}\}}\{\ln(1-c_{lm})\}$, $\mathbb{E}_{\mathbb{Q}\{\rho_l\}}\{\ln(\rho_l)\}$, and $\mathbb{E}_{\mathbb{Q}\{\rho_l\}}\{\ln(1-\rho_l)\}$ can be evaluated using (11) and the distributions $\mathbb{Q}\{g_n\}$ are normalized to represent a valid probability distribution.

The derived update rules are employed alternatively to update the parameters of the variational distributions. Generally, the per iteration complexity of the update rules is $\mathcal{O}(NML_{max})$. However, considering the fact that there are at most N measurements at each time frame, the complexity of the inference algorithm becomes $\mathcal{O}(N\max\{M, L_{max}\})$.

REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, 2006.

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

[3] M. B. H. Weiss, M. Altamimi, and L. Cui, "Spatio-temporal spectrum modeling: Taxonomy and economic evaluation of context acquisition," *Telecommun. Policy*, vol. 36, no. 4, pp. 335–348, 2012.

[4] Z. Khan, "Opportunistic channel selection by cognitive wireless nodes under imperfect observations and limited memory: A repeated game model," *IEEE Trans. Mobile Comput.*, vol. 15, no. 1, pp. 173–187, Jan. 2016.

[5] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Optimal multiband joint detection for spectrum sensing in cognitive radio networks," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1128–1140, Mar. 2009.

[6] C. S. Hyder, B. Grebur, L. Xiao, and M. Ellison, "ARC: Adaptive reputation based clustering against spectrum sensing data falsification attacks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1707–1719, Aug. 2014.

[7] A. Zaeemzadeh, M. Joneidi, B. Shahrasbi, and N. Rahnavard, "Missing spectrum-data recovery in cognitive radio networks using piecewise constant nonnegative matrix factorization," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Tampa, FL, USA, 2015, pp. 238–243.

[8] S. H. A. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, "Optimality of myopic sensing in multichannel opportunistic access," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4040–4050, Sep. 2009.

[9] B. L. Mark and A. O. Nasif, "Estimation of maximum interference-free power level for opportunistic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2505–2513, May 2009.

[10] H. Li, "Cooperative spectrum sensing via belief propagation in spectrum-heterogeneous cognitive radio systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Sydney, NSW, Australia, 2010, pp. 1–6.

[11] R. Vaze and C. R. Murthy, "Multiple transmitter localization and whitespace identification using randomly deployed binary sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 2, no. 4, pp. 358–369, Dec. 2016.

[12] X.-L. Huang *et al.*, "Intelligent cooperative spectrum sensing via hierarchical Dirichlet process in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 5, pp. 771–787, May 2015.

[13] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and fairness in spectrum assignment for opportunistic spectrum access," *Mobile Netw. Appl.*, vol. 11, no. 4, pp. 555–576, Aug. 2006.

[14] Z. Zhao, Z. Peng, S. Zheng, and J. Shang, "Cognitive radio spectrum allocation using evolutionary algorithms," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4421–4425, Sep. 2009.

[15] T. Do and B. L. Mark, "Joint spatial–temporal spectrum sensing for cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 7, pp. 3480–3490, Sep. 2010.

[16] Q. Wu, G. Ding, J. Wang, and Y.-D. Yao, "Spatial-temporal opportunity detection for spectrum-heterogeneous cognitive radio networks: Two-dimensional sensing," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 516–526, Feb. 2013.

[17] Z. Wei, Z. Feng, Q. Zhang, and W. Li, "Three regions for space–time spectrum sensing and access in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2448–2462, Jun. 2015.

[18] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.

[19] S. Maleki, P. Ciblat, S. Chatzinotas, B. S. M. R., and B. Ottersten, "Cooperative estimation of power and direction of transmission for a directive source," *IEEE Trans. Cogn. Commun. Netw.*, vol. 2, no. 4, pp. 343–357, Dec. 2016.

[20] S. Debroy, S. Bhattacharjee, and M. Chatterjee, "Spectrum map and its application in resource management in cognitive radio networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 1, no. 4, pp. 406–419, Dec. 2015.

[21] X.-L. Huang *et al.*, "Historical spectrum sensing data mining for cognitive radio enabled vehicular ad-hoc networks," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 1, pp. 59–70, Jan./Feb. 2016.

[22] B. Shahrasbi, N. Rahnavard, and A. Vosoughi, "Cluster-CMSS: A cluster-based coordinated spectrum sensing in geographically dispersed mobile cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6378–6387, Jul. 2017.

[23] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar, "IEEE 802.22: The first worldwide wireless standard based on cognitive radios," in *Proc. 1st IEEE Int. Symp. New Frontiers Dyn. Spectr. Access Netw. (DySPAN)*, Baltimore, MD, USA, Nov. 2005, pp. 328–337.

[24] C. Tekin, S. Hong, and W. Stark, "Enhancing cognitive radio dynamic spectrum sensing through adaptive learning," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Boston, MA, USA, 2009, pp. 1–7.

[25] A. Arora *et al.*, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Comput. Netw.*, vol. 46, no. 5, pp. 605–634, 2004.

[26] A. De Domenico, E. C. Strinati, and M.-G. Di Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 21–44, 1st Quart., 2012.

[27] C. J. Burke and M. Rosenblatt, "A Markovian function of a Markov chain," *Ann. Math. Stat.*, vol. 29, no. 4, pp. 1112–1122, Dec. 1958.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006. [Online]. Available: http://www.springer.com/us/book/9780387310732

[29] H. Ishwaran and L. F. James, "Some further developments for stick-breaking priors: Finite and infinite clustering and classification," *Sankhya Indian J. Stat.*, vol. 65, no. 3, pp. 577–592, Aug. 2003. [Online]. Available: http://www.jstor.org/stable/25053289

[30] G.-J. Qi, C. C. Aggarwal, J. Han, and T. Huang, "Mining collective intelligence in diverse groups," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, Geneva, Switzerland, 2013, pp. 1041–1052.

[31] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, nos. 1–2, pp. 1–305, 2008.

[32] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, 1999.

[33] A. Zaeemzadeh, M. Joneidi, and N. Rahnavard, "Adaptive non-uniform compressive sampling for time-varying signals," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, 2017, pp. 1–6.

[34] M. Bahrami and B. Vasic, "Generalized belief propagation based deliberate bit flipping modulation coding," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[35] R. Kazemi, M. Boloursaz, S. M. Etemadi, and F. Behnia, "Capacity bounds and detection schemes for data over voice," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 8964–8977, Nov. 2016.

[36] Y. Boykov, O. Veksler, and R. Zabih, "Efficient approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1222–1239, Nov. 2001.

[37] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2009, pp. 670–677.

[38] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[39] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.
[40] F. Digham, M.-S. Alouini, and M. Simon, "On the energy detection of unknown signals over fading channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 5. Anchorage, AK, USA, pp. 3575–3579.
[41] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.

Alireza Zaeemzadeh (S'11) received the B.S. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2014. He is currently pursuing the M.Sc. and Ph.D. degrees in electrical engineering with the University of Central Florida. His current research interests lie in the areas of statistical signal processing and Bayesian data analysis. He was a recipient of the University of Central Florida Multidisciplinary Doctoral Fellowship and the Graduate Dean's Fellowship.

Mohsen Joneidi (S'14) received the B.S. degree from the Ferdowsi University of Mashhad in 2009 and the M.S. degree from the Sharif University of Technology in 2012. He joined CWNLAB with the Electrical and Computer Engineering Department, University of Central Florida, Orlando, FL, USA. His research interests include statistical signal processing and compressed sensing.

Nazanin Rahnavard (S'97–M'10) received the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2007. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA. She has interest and expertise in a variety of research topics in the communications, networking, and signal processing areas. She was a recipient of NSF CAREER Award in 2011. She serves on the editorial board of the *Journal on Computer Networks* (Elsevier) and on the Technical Program Committee of several prestigious international conferences.

Guo-Jun Qi received the Ph.D. degree from the University of Illinois at Urbana–Champaign in 2013. He is currently a Faculty Member with the Department of Computer Science, University of Central Florida. His research interests include pattern recognition, machine learning, computer vision, multimedia, and data mining. He was a recipient of twice IBM Ph.D. Fellowships, the Microsoft Fellowship, and the Best Paper Award at the 15th ACM International Conference on Multimedia, Augsburg, Germany, 2007. He has served as a program committee member and a reviewer for many academic conferences and journals in the fields of pattern recognition, machine learning, data mining, computer vision, and multimedia.