Cooperative Spectrum Sensing in the Presence of Correlated and Malicious Cognitive Radios

Mihir Laghate, Student Member, IEEE, and Danijela Cabric, Senior Member, IEEE

Abstract—While cooperative spectrum sensing improves sensing reliability when secondary users (SUs) have independent measurements, these gains are limited in the presence of correlated or malicious SUs. In this paper, we propose algorithms to improve spectrum sensing performance when the system contains both honest and malicious SUs all of whom may be experiencing correlated fading channels. We show that, when the SUs’ reports at each time slot are independent and identically distributed, it is impossible to distinguish between malicious collaboration, i.e., collusion, and correlations caused by the environment. Thus, the optimal test statistic for cooperative spectrum sensing depends on the correlations in the SU reports but not on the source of these correlations. We propose two algorithms: one to identify SUs whose reports are correlated and another to infer the spectrum occupancy by using the learned structure of correlations. Groups of SUs whose reports are correlated are identified by learning the structure of the underlying Bayesian network model. This structure is implemented as a factor graph to infer the spectrum occupancy using a loopy belief propagation algorithm. We derive an upper bound of the error probability of the proposed structure learning algorithm and prove convergence of our loopy belief propagation algorithm.

Index Terms—Cognitive radio, spectrum sensing, cooperation, security, graphical model, structure learning, belief propagation.

I. INTRODUCTION

COOPERATIVE spectrum sensing is a well-recognized method to identify if the primary user (PU) is transmitting. The presence of the PU is inferred by combining the estimates reported by multiple secondary users (SUs). Cooperative spectrum sensing exploits the spatial diversity inherent in the fading radio environment to achieve higher detection probability and lower false alarm probability than that achieved by an individual SU. However, if SUs experience correlated fading then cooperation amongst SUs does not significantly improve the performance of conventional cooperative spectrum sensing techniques [2]. Also, the distributed nature of cooperative spectrum sensing makes it vulnerable to misreporting by malicious SUs. In particular, statistical attacks have been shown to adversely affect both the detection and false alarm probabilities of cooperative spectrum sensing [3]–[6]. A detailed classification of various statistical data falsification attacks by individual SUs is provided in [7]. In this paper, we propose algorithms to improve spectrum sensing accuracy when SUs may be honest or malicious and all of whom may be experiencing correlated fading channels.

All SUs estimate the same quantity, viz. the PU spectrum occupancy. Therefore, their estimates are correlated. In fact, given the true value of the PU spectrum occupancy, the SU reports may or may not be conditionally independent. It can be shown that the measurements at two SUs will be conditionally independent given PU spectrum occupancy if they do not receive any common interference and the channels from the PU to the SUs are uncorrelated with each other.

SUs may be malicious and periodically report a falsified value to the fusion center. Statistical data falsification attacks by malicious attackers result in correlations in the reports of the malicious SUs if they are colluding. Reports sent by the SUs will be conditionally independent given PU spectrum occupancy only if their measurements are conditionally independent and the SUs do not collude [4]. Note that this does not imply that the SUs are honest because SUs may falsify their own reports without collaboration.

For the sake of readability, we abuse notation and call SU reports independent (resp. correlated) if they are conditionally independent (resp. dependent) given the true spectrum occupancy.

Cooperative spectrum sensing systems with correlated SU reports have been studied in literature. A computationally expensive reinforcement learning algorithm proposed in [8] estimates the Neyman-Pearson sufficient test statistic for spectrum sensing by learning the joint distribution of all SU reports. The spectrum sensing error probability was bounded below in [9] for SUs placed in a circle and experiencing correlated channel fading. Exact location and correlations in channel fading are used in [10] to identify malicious SUs. However, these methods require that correlations in SU reports be caused only by shadow fading as described in [11].

We will design a belief propagation (BP) based algorithm for cooperative spectrum sensing from these correlated reports. BP has been used for reducing the computational complexity of cooperative spectrum sensing. [12] uses BP on a tree to reduce the computational complexity of computing the likelihood function of the energy received in a single slot based on the sensor statistics. Loopy BP has been used in [3] to detect spectrum occupancy when the sensing results of all SUs are independent, and malicious SUs do not collude. Correlations in SU reports are modeled in [6] and [13] as Markov random fields by assuming that neighboring SUs have correlated reports. BP is used in [6] to estimate the prior parameter for reconstruction of the PU using compressive sensing. On the other hand, BP is used in [13] to estimate...
the received energy at each SU by using the energy received at its neighbors. In these cases, the correlations between the energies received at the SUs are estimated from the empirical model of [11]. Therefore, these methods are applicable to only shadow fading channels and cannot counter malicious attacks.

In this work, we wish to infer the spectrum occupancy in a computationally efficient manner irrespective of the cause of correlations in SU reports. We begin by recognizing that the joint distribution of the SU reports and the spectrum occupancy can be modeled by a Bayesian network. Studying this graphical model, we show that the cause of the correlations cannot be identified from SU reports. We propose sensing the spectrum in two stages. First, we identify the SUs that have correlated reports by learning the structure of the Bayesian network. Next, we use the resulting factorization of the joint distribution to develop a loopy BP algorithm to learn the posterior marginal distribution of the PU spectrum occupancy.

This paper is organized as follows. Section II describes the system model and represents it as a Bayesian network. Section III proves that correlation due to the environment cannot be distinguished from collusion. It also describes the algorithm to learn the structure of the graphical model and thus identify groups of correlated SUs. The proposed loopy BP algorithm for spectrum sensing is described in Section IV. Theoretical analysis is provided in Section V. Section VI evaluates the proposed algorithms by simulations. Section VII concludes the paper and describes some extensions.

II. SYSTEM MODEL

We consider a cognitive radio network where SUs cooperate to sense whether a PU is transmitting or not in a known frequency range. The fading channel experienced by the PU signals received at different SUs may be correlated. Multiple SUs may also receive correlated noise from common interferers. Each SU generates a binary estimate of the spectrum occupancy. The SUs then report these (or falsified) estimates of the spectrum occupancy to a fusion center.

Assume that a single PU is transmitting in the given frequency range. Let there be $K$ SUs, indexed by $1, 2, \ldots, K$, sensing this frequency range. At the $t$’th time slot, let the PU spectrum occupancy be denoted by $s(t) \in \{0, 1\}$, where $s(t) = 0$ if the PU is not transmitting and $s(t) = 1$ otherwise. We assume that we do not know PU traffic statistics. Hence, $s(t)$ is assumed to be i.i.d. At each time slot $t$, the $k$’th SU generates an estimate $u_k(t) \in \{0, 1\}$ of $s(t)$. It reports a value $y_k(t) \in \{0, 1\}$ to the fusion center. The fusion center combines these reports to form a binary estimate $\hat{s}(t)$, of $s(t)$.

We assume that the signal received by the SUs from the PU experiences a fading channel, such as that modeled in [11]. When two SUs are separated by distance $d$, the authors of [11] empirically show that the signals received by the SUs experience shadow fading and the receiving signal powers have correlation proportional to $e^{-\beta d}$ where $\beta$ depends on the environment. These propagation characteristics and the “local” detector used by the SUs to generate $\{u_k(t)\}_{k \in \{1, \ldots, K\}}$ decide the conditional joint distribution $P(u_1(t), \ldots, u_K(t)|s(t))$. We develop our algorithm starting from this conditional joint distribution in order to make our algorithm development agnostic to the channel propagation characteristics and the local detector statistics. In fact, our algorithm does not require that all SUs experience the same kind of channels or use the same kind of local detector. Our algorithm assumes that this conditional joint distribution is time invariant and that each SU’s reports have a greater than 0.5 probability of being correct, i.e., $P(y_k(t) = s(t)) > 0.5$. This is a reasonable assumption because outlier detection methods such as [5], [14] are capable of detecting and ignoring SUs that do not satisfy this requirement.

It is easy to show that the sensing results of two SUs are conditionally dependent given $s(t)$ if the signal received by them is conditionally dependent given $s(t)$.

We consider a statistical data falsification attack model for malicious SUs [3], [5]. The $k$’th SU is considered to be honest if $P(y_k(t) = u_k(t)) = 1$. Otherwise, it is considered to be malicious. SUs can be malicious on their own, or they can collude, i.e., modify their reports collaboratively. If the $k$’th SU is individually malicious, its attack strategy is described by the conditional distribution $P(y_k(t)|u_k(t))$. If a set $M_b$ of SUs is colluding, then their attack strategy is described by the conditional distribution $P(y_{M_b}(t)|u_{M_b}(t))$. Each SU is mapped to exactly one of $C$ non-empty sets $C_1, \ldots, C_C$ such that two SUs $k_1$ and $k_2$ are mapped to distinct sets iff $u_{k_1}(t)$ and $u_{k_2}(t)$ are conditionally independent given $s(t)$.

We use a graphical model to represent the PU spectrum occupancy, SU reports, and malicious behavior. This mathematical model can be represented as a Bayesian network as follows. Consider a single time slot. We drop the time index from the notation for ease of reading. The joint distribution of $s$, sensing results $u_k$, and reports $y_k$ ($k \in \{1, \ldots, K\}$) can be factorized as

$$P(s, u_1, \ldots, u_K; y_1, \ldots, y_K) = P(s)P(u_1, \ldots, u_K|s)P(y_1, \ldots, y_K|u_1, \ldots, u_K). \tag{1}$$
The correlations in the sensing results and our model of malicious SUs allow further factorization:

\[
P(s, u_1, \ldots, u_K, y_1, \ldots, y_K) = P(s) \prod_{c=1}^{C} P(u_{c|s}) \prod_{b=1}^{B} P(y_{M_b}|u_{M_b}).
\] (2)

This factorization is consistent with a Bayesian network with nodes for spectrum activity, correlations in sensing results, maliciousness, and SU reports. An example is shown in Fig. 1. Correlation for SUs in \(C_c\), due to correlated fading or common interferers is modeled by the nodes \(X_{c|s}\) for \(c \in \{1, \ldots, C\}\). For \(b \in \{1, \ldots, B\}\), \(A_{M_b}\) models the conditional distribution \(P(y_{M_b}|u_{M_b})\) which represents the (possibly malicious) behavior of SUs. Thus, the joint distribution underlying the spectrum sensing system is modeled as a Bayesian network.

### III. Structure Learning

Ideally, we would like to identify both partitionings \(\{C_c\}_{1 \leq c \leq C}\) and \(\{M_b\}_{1 \leq b \leq B}\) and use them to learn the conditional joint distribution of the SU reports given the spectrum occupancy. However, by Theorem 1 we show that it is impossible to identify all these partitionings.

**Theorem 1:** If the reports of a group of SUs are conditionally dependent given the true spectrum occupancy, then the SUs could either have correlated sensing results or they could be colluding. The joint distribution of the reports is not sufficient to identify the cause of the correlation.

**Proof:** If the SUs have correlated reports, then from (2), we note that identifying the reason for correlation between SU reports is equivalent to learning the factorization of the joint distribution underlying the SU reports. This factorization has a one-to-one correspondence with the structure of the Bayesian network [15]. From our system model, we know that collusion and environmental correlation occur at different heights in the Bayesian network. Hence, identifying the reason for correlation between SU reports is equivalent to learning the structure of the Bayesian network.

In our Bayesian network, the SU reports are observable variables, while all the rest are latent, or hidden, variables. When using the joint distribution to learn the structure of a Bayesian network which contains latent variables, latent variables with just two neighbors cannot be distinguished from their neighbors [15]. Therefore, nodes representing

1. the SUs’ sensing results \(u_s\),
2. the environmental correlation for SUs which have independent sensing results, and
3. the attack strategies for SUs which are either honest, or attack in an independent fashion

cannot be learned. Hence, we cannot distinguish between honest SUs which have correlated sensing results, and colluding SUs which have independent sensing results.

As an example, consider the SU network shown in Fig. 1. SUs 1 and 2 are honest and SUs 3 and 4 are colluding. Among the latent variables, nodes \(u_1, u_2, u_3, u_4, X_{\{3\}}\), and \(X_{\{4\}}\) have just 2 neighbors each. Therefore, their presence cannot be learned from the joint distribution of the leaf nodes. The structure that we would learn is shown in Fig. 2.

---

**Fig. 2.** The learnable structure of Bayesian network does not allow identification of the source of correlation

The learned structure is a tree. It can be rooted at the node representing \(s\), and has a maximum height of 3. SUs that have independent reports are children of the root node. SUs that have correlated reports are children of a node representing the correlations in their reports which, in turn, is a child of the node representing \(s\). Hence, in the learned structure, leaf nodes at height 2 represent SUs with independent reports, and leaf nodes at height 3 represent SUs which are either colluding or have correlated sensing results. Since this completely describes all SU behaviors, we conclude that learning the groups of SUs which are sibling nodes in the learned Bayesian tree identifies all the groups of correlated SUs and the one group of SUs with independent reports.

Learning the structure of a tree with latent variables by computing the maximum likelihood tree is known to be NP-Hard [16]. We approach it from a Bayesian perspective by studying the distribution of our chosen test statistic and setting a desired detection probability to check if a pair of SUs are sibling nodes. The Bayesian network structure is inferred from the \(\binom{K}{2}\) inferences obtained by checking each pair of SUs. We use a heuristic to account for errors in these inferences.

### A. Testing a Pair of SUs

Sibling relations between a pair of SUs can be tested by Lemma 4 of [17] which we reproduce below, without proof, as Theorem 2. It requires computing the information distance \(g_{ij}\) for each pair of nodes \(i\) and \(j\):

\[
g_{ij} \triangleq -\log \left( \frac{\det W_{ij}}{\sqrt{\det V_i \det V_j}} \right),
\] (3)

where \(W_{ij}\) is a \(2 \times 2\) matrix of the joint probability distribution of nodes \(i\) and \(j\), and \(V_i\) and \(V_j\) are diagonal matrices of the probability distributions of nodes \(i\) and \(j\), respectively.

**Theorem 2 (Lemma 4, [17]):** Leaf nodes \(i\) and \(j\) are siblings if and only if

\[
g_{ik} - g_{jk} = g_{ik'} - g_{jk'} \quad \forall \text{ distinct indices } k, k' \neq i, j.
\] (4)

In our system, each leaf node takes values in \(\{0, 1\}\), i.e., it is a Bernoulli random variable. We exploit this fact to derive an equivalent condition in terms of their Pearson’s correlation coefficient. In addition, we show that testing the leaf nodes is sufficient to identify siblings.

**Corollary 1:** Let an undirected tree graph have \(K\) leaf nodes indexed \(1, \ldots, K\). Assume that the \(k\)th leaf node
represents a Bernoulli random variable $y_k$. Let $\rho_{ij}$ be Pearson’s correlation coefficient of nodes $i$ and $j$ defined as [18]
\[
\rho_{ij} \triangleq \frac{\text{Cov}(y_i, y_j)}{\sqrt{\text{Var}(y_i)\text{Var}(y_j)}}.
\]
(5)

Then, leaf nodes $i$ and $j$ are siblings if and only if
\[
\frac{\rho_{ik}}{\rho_{jk}} = \frac{\rho_{ik'}}{\rho_{jk'}} \quad \forall k, k' \in \{1, \ldots, K\}\setminus\{i, j\}; k \neq k'.
\]
(6)

**Proof:** Note that if nodes $i$ and $j$ take binary values, then their information distance $g_{ij}$ relates to their Pearson’s correlation coefficient $\rho_{ij}$ as
\[
g_{ij} = -\log(|\rho_{ij}|).
\]
(7)

First, let’s prove that sibling nodes satisfy (6). Consider sibling nodes $i$ and $j$. By Theorem 2,
\[
g_{ik} - g_{jk} = g_{ik'} - g_{jk'}
\]
(8)
for all distinct nodes $k$ and $k'$ in the graph. Therefore, (8) also holds for all distinct leaf nodes $k$ and $k'$ different than both $i$ and $j$. Substitute (7) into (8) and use the bijective nature of the logarithm function to get
\[
-\log\left(\frac{\rho_{ik}}{\rho_{jk}}\right) = -\log\left(\frac{\rho_{ik'}}{\rho_{jk'}}\right) \Rightarrow \frac{\rho_{ik}}{\rho_{jk}} = \frac{\rho_{ik'}}{\rho_{jk'}}.
\]
(9)

Next, consider leaf nodes $i$ and $j$ such that (6) is true for all other leaf nodes $k \neq k'$. Since the leaf nodes take binary values, (7) implies that
\[
g_{ik} - g_{jk} = g_{ik'} - g_{jk'}
\]
(10)
for all distinct leaf nodes $k$ and $k'$. In order to prove that nodes $i$ and $j$ are sibling nodes, we need to prove that $g_{il} - g_{jl} = g_{il'} - g_{jl'}$ for all distinct nodes $l$ and $l'$ in the graph.

Now consider an interior node $l$ in the graph. Let $k \neq i, j$ be a leaf node in the subtree induced by the node $l$ and its descendants. Therefore, we can use the additivity of information distances [17, Lemma 3] to decompose $g_{ik}$ into two parts along the path from node $i$ to node $k$ in the tree: $g_{ik} = g_{il} + g_{lk}$. We use this to compute $g_{il} - g_{jl}$ as follows.

\[
g_{ik} - g_{jk} = (g_{il} + g_{lk}) - (g_{jl} + g_{lk}) = g_{il} - g_{jl}
\]
(11)

Equations (11) and (10) together show that (6) holds for all distinct nodes $k$ and $k'$ in the graph. Therefore, Theorem 2 implies that nodes $i$ and $j$ are siblings.

Thus, we have proven that we do not need to test (4) for all nodes in the graph. Instead, checking equality amongst all the leaf nodes is sufficient.

The original equality test of (4) tests the equality of the logarithm of the ratio of correlation coefficients. As we show later, the ratio of sample correlation coefficients is Gaussian distributed with variance inversely proportional to the number of samples. The small gradient of the logarithm function reduces the probability of correctly distinguishing between unequal ratios. Our condition (6) avoids this problem. Hence, rewriting condition (4) in terms of the correlation coefficient improves the accuracy.

Based on Corollary 1, we can now set up a binary hypothesis testing problem for checking whether nodes $i$ and $j$ are siblings. Let $\mathcal{H}_0(i, j)$ denote the hypothesis that nodes $i$ and $j$ are siblings and let $\mathcal{H}_1(i, j)$ denote the hypothesis that they are not siblings. However, empirical estimates of the correlation coefficients will be noisy. In order to test (6) with noisy estimates we need to know the distribution of the ratio of sample correlation coefficients.

**B. Distribution of Ratio of Correlation Coefficients**

We start by deriving the distribution of the sample correlation coefficient of two Bernoulli random variables $y_i$ and $y_j$. Assume that $T_i$ samples of $y_i$ and $y_j$ have been collected. The sample correlation coefficient $\hat{\rho}_{ij}$ of $y_i$ and $y_j$ is [19]
\[
\hat{\rho}_{ij} = \frac{\sum_{t=1}^{T_i}(y_i(t) - \hat{m}_i)(y_j(t) - \hat{m}_j)}{\sqrt{\sum_{t=1}^{T_i}(y_i(t) - \hat{m}_i)^2}(\sum_{t=1}^{T_i}(y_j(t) - \hat{m}_j)^2)}.
\]
(12)

The next step is to compute the distribution of the ratio of sample correlation coefficients. We show that, given a sufficient number of samples, the ratio of sample correlation coefficients is also Gaussian distributed.

**Theorem 3:** Consider distinct Bernoulli random variables $y_i$, $y_j$, and $y_k$. If $T_i > \frac{1}{0.0081 \frac{\sigma_{ij}^2}{\rho_{ij}^2}}$, then the ratio $\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}$ of the sample correlation coefficients is Gaussian distributed with
\[
\mathbb{E}\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right] = \frac{\rho_{ik}}{\rho_{jk}} \left(1 + \frac{1}{T_i} \frac{\sigma_{ik}^2}{\rho_{ik}^2}\right) + O(T_i^{-2})
\]
and
\[
\text{Var}\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right] = \frac{1}{T_i} \frac{1}{\rho_{jk}^2} \left(\frac{\rho_{ik}}{\rho_{jk}}\right)^2 \frac{\sigma_{ik}^2 + \sigma_{jk}^2}{\left(\frac{\sigma_{ik}}{\rho_{ik}}\right)^2} + O(T_i^{-2})
\]
where $O(\cdot)$ is the big-O notation and $\sigma_{ij}^2 \triangleq T_i \text{Var}[\hat{\rho}_{ij}]$ for all $i, j \in \{1, \ldots, K\}$.

**Proof:** The proof is provided in Appendix A.

We wish to test a pair of SUs for sibling relationship by setting a threshold based on a desired detection probability. Dynamically setting a threshold computed from the variance (14) would be difficult because $\sigma_{ik}^2$ and $\sigma_{jk}^2$ are estimated from the same samples as those used for testing. Instead, we propose using an upper bound of the variance to decide the thresholds because this upper bound depends on the estimated variance of individual SU’s reports rather than the estimated variance of the correlation coefficients. Since we have just one sample of correlation coefficient, we expect the estimated variance of this correlation coefficient to have a higher error than the estimated variance of the individual SU’s reports of which we have $T_i$ samples.

**Corollary 2:** Consider distinct Bernoulli random variables $y_i$, $y_j$, and $y_k$. If $T_i > \frac{1}{\max_{i,j,k} \frac{\sigma_{ij}^2}{\rho_{ij}^2}}$, then the variance of the ratio of the sample correlation coefficients $\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}$ is bounded above as
\[
\text{Var}\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right] \leq \frac{1}{16T_i \rho_{jk}^2 T_k} \left[ \left(\frac{\rho_{ik}}{\rho_{jk}}\right)^2 \frac{\sigma_{ij}^2 + 1}{\sigma_{ij}^2} \right] \leq \frac{\sigma_{ij}^2}{T_i}
\]
(15)
Proof: Begin by analyzing the variance $\varsigma^2_{ik}$ of the sample correlation coefficient $\hat{\rho}_{ik}$:

$$\varsigma^2_{ik} = \frac{(p_{ik}(1,0) + p_{ik}(0,1))(p_{ik}(1,1) + p_{ik}(0,0))}{4\sigma^2_i \sigma^2_k} = (1 - p_{ik}(0,0) - p_{ik}(1,1))(p_{ik}(1,1) + p_{ik}(0,0))$$

Now, note that $0 \leq p_{ik}(0,0) + p_{ik}(1,1) \leq 1$ and $x(1-x) \leq 0.25$ if $x \in [0,1]$. Hence,

$$\varsigma^2_{ik} \leq \frac{1}{16\sigma^2_i \sigma^2_k}. \quad (16)$$

Similarly, we have

$$\varsigma^2_{jk} \leq \frac{1}{16\sigma^2_i \sigma^2_k}. \quad (17)$$

If $T_i > 0.0081 \frac{\varsigma_{ik}^2}{\sigma_{ik}^2}$, then Theorem 3 proves that $\hat{\rho}_{jk}$ is approximately Gaussian with variance given by (14). Substituting (16) and (17) in (14) gives the required bound (15).

C. Identifying Pairs of Siblings

Pairs of sibling nodes in a graph can be identified by Corollary 1. However, since error-free estimates of the correlation coefficients are not available, we propose testing (6) by using Theorem 3 and Corollary 2, i.e., for each pair of nodes $(i,j)$, we test if $K - 2$ Gaussian distributions have the same mean.

Consider two distinct SUs indexed $i$ and $j$. We propose testing whether each triplet of distinct nodes $(i,j,k)$ supports $\mathcal{H}_0(i,j)$. If each such triplet supports $\mathcal{H}_0(i,j)$, then we infer that the nodes $i$ and $j$ are siblings. Otherwise, we infer that the nodes $i$ and $j$ are not siblings.

If $\mathcal{H}_0(i,j)$ is true, i.e., nodes $i$ and $j$ are siblings, then

$$\hat{\rho}_{ij} \triangleq \frac{1}{K-2} \sum_{k \neq \{i,j\}} \hat{\rho}_{jk}$$

is the maximum likelihood estimate of $\rho_{jk}$. Using the estimated mean $\hat{\mu}_{ij}$ and the upper bound of the variance $\sigma^2_{ij}$, we define the test statistic

$$z(i,j,k) = \frac{\hat{\rho}_{ik} \hat{\rho}_{jk} - \hat{\mu}_{ij}}{1/\sqrt{16T_i \rho_{jk}^2 \sigma^2_k} \left[ \rho_{ik}^2 \sigma^2_k + 1/\sigma^2_i \right]} \quad (19)$$

to test whether the triplet $(i,j,k)$ supports the hypothesis $\mathcal{H}_0(i,j)$ or $\mathcal{H}_1(i,j)$. Therefore, $z(i,j,k)$ is Gaussian distributed with mean 0 and variance less than 1. This is visualized in Fig. 3 for an example system. A two-sided test with detection probability $P_d$ as a parameter can be set up using the cumulative distribution function $\Phi$ of the standard normal distribution as follows: If for all $k \in \{1, \ldots, K\} \setminus \{i,j\}$

$$1 - P_d < 1 - \Phi(z(i,j,k)) < 1 + P_d/2, \quad (20)$$

then declare hypothesis $\mathcal{H}_0(i,j)$ to be true, i.e., declare nodes $i$ and $j$ to be siblings. These test results can be arranged into a binary matrix $A$ of size $K \times K$. It is defined element-wise as $A(i,j) = 1$ if nodes $i$, $j$ are inferred to be siblings and $A(i,j) = 0$ otherwise.

D. Identifying Groups of Siblings

Testing each pair $(i,j)$ of SU nodes may not give consistent results in a global sense. If nodes $i$ and $j$ are siblings, and nodes $j$ and $k$ are siblings, then we know that nodes $k$ and $i$ have to be siblings. However, this may not be reflected in the pair-wise test results $A$ because one (or more) of these tests may be in error.

Resolving this inconsistency optimally requires knowing the correlations between the test results. These are hard to estimate online because we have just one realization of data. Instead, we propose a heuristic as follows. Consider $A$, the matrix of test results, as an adjacency matrix of an undirected unweighted graph with $K$ vertices. Each leaf node in the original graph now corresponds to a vertex of this graph. A group of sibling nodes corresponds to a maximal clique in the new graph. We need to find all the maximal cliques in this graph while correcting for possible erroneous test results.

Note that we designed our test for pair-wise sibling relationships by constraining the probability of missing a sibling relationship. Accordingly, our new graph is more likely to have extra edges than missing edges. Hence, we propose resolving the inconsistency in test results by finding the minimum set of non-intersecting maximal cliques that cover the entire vertex set. These maximal cliques are declared as groups of sibling nodes. The maximal cliques can be identified by iterating through all the vertices of the graph. Therefore, it has $O(K^2)$ worst case complexity.

Thus, we have identified the groups of SU nodes that have correlated reports. For use in the next section, we denote the identified groups of correlated SU $\mathcal{G}_1, \ldots, \mathcal{G}_M$ where $M$ is the total number of groups identified. Two SUs $i$ and $j$ belong to different sets of this partition iff their reports $y_i(t)$ and $y_j(t)$ are conditionally independent given $s(t)$. 

Fig. 3. Empirical distribution of test statistic $z(i,j,k)$ for a 6 SU system where the SUs 1, 2, and 3 have correlated reports. Reports collected for 400 time slots.
IV. SENSING USING BELIEF PROPAGATION

The next step is to use this newly learned information to improve the spectrum sensing performance. Referring back to the discussion in Section II, it is impossible to identify the reason why a particular group of SUs has correlated reports. Therefore, it is only logical that the sufficient test statistic for spectrum sensing depends only on the correlations in the reports and not on the reason for the correlation. Hence, we now propose a Bayesian inference method to fuse the SU reports in a way that is agnostic to the cause of the correlations.

For the purpose of developing the algorithm, we assume that all groups of correlated SU reports have been identified correctly. This is a reasonable assumption since the algorithm we proposed in Section III will identify these groups with very high probability as is verified by simulations in Fig. 5. The effect of incorrectly learned structure is evaluated through simulations in Fig. 13 of Section VI.

A. Detector Formulation

The MAP detector of \( s(t) \) can be computed from the SU reports of the same time slot:

\[
\hat{s}_{\text{MAP}}(t) = \arg \max_{s \in \{0,1\}} p(s) \prod_{m=1}^{M} P(y_{3,m}(t) | s(t) = s)
\]

where \( p(\cdot) \) denotes the prior distribution of unknown quantities. However, since these conditional distributions aren’t known, we are unable to compute the MAP estimate directly. Instead, we propose detecting \( s(t) \) by using the SU reports \( y_k(t) \) received from SUs \( k \in \{1, \ldots, K\} \) over \( T_s \) time slots \( t \in \{1, \ldots, T_s\} \) to learn the conditional distribution of the SU reports given the spectrum occupancy. Factorizing the joint distribution of SU reports and spectrum occupancy as per the Bayesian tree model described in Section II reduces the computational complexity of the learning process. In particular, the belief propagation (BP) algorithm [20] [21], also known as the sum-product algorithm, has been shown to be computationally efficient at learning underlying joint distribution and estimating hidden variables. In our case, the hidden variables are the spectrum occupancy \( s(t) \) and the conditional distributions \( r_m(s,y) \propto P(y_{3,m}(t) = y | s(t) = s) \). The proposed BP algorithm attempts to compute the detector

\[
\{\hat{s}_{\text{BP}}(t)\}_{t \in \{1, \ldots, T_s\}} = \arg \max_{s \in \{0,1\}^{T_s}} \prod_{t=1}^{T_s} p(s(t)) \prod_{m=1}^{M} p(r_m) \times \prod_{t=1}^{T_s} \prod_{m=1}^{M} P(y_{3,m}(t) | s(t), r_m) (22)
\]

where \( D_m \) is the domain of \( r_m \) and a parameter of our algorithm.

B. Factor Graph and Belief Propagation

The BP algorithm estimates the posterior marginal distributions of hidden variables by passing messages along the edges in a factor graph. A factor graph is a graph consisting of two types of nodes – factor nodes and variable nodes. The factor nodes compute the factors of the joint distribution (2). The variable nodes represent the variables in the factor computed by the adjacent factor node(s).

The factor graph for our proposed BP algorithm is constructed by combining \( T_s \) copies of the Bayesian network model learned by the algorithm proposed in Section III. An example factor graph is depicted in Fig. 4. There are two types of variable nodes in our factor graph: the spectrum occupancy \( s(t) \) for \( t \in \{1, \ldots, T_s\} \) and the conditional distribution \( r_m(s,y_{3,m}) \) for \( m \in \{1, \ldots, M\} \). Factor nodes are represented by \( \Psi_{m,t} \) for \( m \in \{1, \ldots, M\} \) and \( t \in \{1, \ldots, T_s\} \). They take the received reports \( y_{3,m}(t) \) as deterministic input and compute the probability of observing these reports conditioned on the values \( r \in D_m \) and \( s \in \{0,1\} \) of the variable nodes \( r_m \) and \( s(t) \) respectively:

\[
\Psi_{m,t}(s,r) \propto P(y_{3,m}(t) | s(t) = s, r_m = r). (23)
\]

The BP algorithm is set up as a parallel message passing scheme, i.e., all nodes transmit messages simultaneously. At the \( n^{th} \) iteration, the messages sent from the variable nodes to the factor nodes are:

\[
\eta_{s(t) \rightarrow \Psi_{m,t}}^{(n)}(s) \propto p(s) \prod_{m' \neq m} \eta_{\Psi_{m',t} \rightarrow s(t)}^{(n-1)}(s) (24)
\]

\[
\eta_{r_m \rightarrow \Psi_{m,t}}^{(n)}(r) \propto p(r_m = r) \prod_{t' \neq t} \eta_{\Psi_{m',t'} \rightarrow r_m}^{(n-1)}(r) (25)
\]

The messages sent from the factor nodes to the variable nodes summarize the messages received.

\[
\eta_{\Psi_{m,t} \rightarrow r_m}^{(n)}(r) \propto \sum_{s \in \{0,1\}} \Psi_{m,t}(s,r) \eta_{s(t) \rightarrow \Psi_{m,t}}^{(n-1)}(s) (26)
\]
The conditional distribution of the perfect honest SU. For the
assume the prior probability of \( D \) to distributions that we expect from the SU reports. Here we
denote distributions. The number of dimensions of these
fading. Thus, \( D \) is independent but the received signal experiences correlated
example, consider the case of SUs whose measurement noise
that in general, the distribution of the reports
J strategies of the SUs of the set
attack strategy.
The assumptions about the environment and the malicious SUs’
potential choices for these parameters and how these reflect
D. Sequential Implementation
The parameters for our algorithm are the priors for
r and \( D \) and the domain \( D_m \) of \( r_m \). Here we describe the
fading channel and, if malicious, the attack
strategies of the SUs of the set \( \beta_m \). While designing \( D_m \), note
that in general, the distribution of the reports \( y_{r_m} \) conditioned on \( s(t) = 0 \) is independent of that conditioned on \( s(t) = 1 \). For example, consider the case of SUs whose measurement noise is independent but the received signal experiences correlated fading. Thus, \( D_m \) should be the Cartesian product of two sets. Furthermore, each of these sets should be a simplex since they denote distributions. The number of dimensions of these simplexes can be used to broaden or restrict the algorithm to distributions that we expect from the SU reports. Here we present two examples of how \( D_m \) can be chosen.
- Consider a case where the SUs within the same group are
statistically equivalent. Such an attack strategy reduces the effectiveness of outlier detection algorithms [14]. In such a case, the fusion algorithm uses an aggregate of the reports of each group of correlated SUs. For example, if the sum of the reports of the group of correlated SUs is used, then \( r_m (y_{r_m}, s) = P \left( \sum_{j \in \beta_m} y_j \bigg| s \right) \) and \( D_m \) is a product of two simplexes each of \( |\beta_m| \) dimensions.
- In the most general case, \( D_m \) is a product of two simplexes each with \( (|\beta_m| - 1) \) dimensions and can model any conditional distribution of the reports.
The prior probability for spectrum occupancy is assumed to be uniform on \( \{0, 1\} \), i.e., \( p(s = 0) = 0.5 = p(s = 1) \). We assume the prior probability of \( r_m \) to be the \( l^1 \) distance from the conditional distribution of the perfect honest SU. For the first example described above, \( p(r_m) \) can be written as:
\[
p(r_m = r) \propto 4 - \left[ 1 - \begin{bmatrix} 0 \mid \beta_m \end{bmatrix} - \begin{bmatrix} r(0, \beta_m) \\ r(1, \beta_m) \end{bmatrix} \right]_1
\]
We choose the \( l^1 \) norm because it has the slowest decay among the \( l^p \) norms. This choice of prior probability of \( r_m \) requires that \( P(y_k(t) = s(t)) > 0.5 \) for at least half the SUs in each \( \beta_m \). In other words, at least half the SUs should report the correct spectrum occupancy. If this does not hold, then it is likely that the fusion decision will be almost always flipped and the honest SUs will appear to be malicious or malfunctioning.

D. Sequential Implementation
Our proposed algorithm makes inferences about spectrum occupancy \( s(t) \) at \( T_s \) consecutive time slots simultaneously. If the algorithm is implemented such that it waits for reports from \( T_s \) time slots to be received before outputting a decision on the spectrum occupancy of the first time slot, then the spectrum sensing is not real time and incurs a delay \( T_s \) time slots. Instead, similar to the “Sequential-BP” method proposed in [3], we propose implementing the algorithm on a sliding window of length \( T_s \). Therefore, when operated at time \( t \), the algorithm uses reports from time indices \( t - T_s + 1, t - T_s + 2, \ldots, t - 1, t \) but the inferred spectrum occupancy for time indices \( t - 1, t - 2, \ldots, t - T_s + 1 \) is ignored. The inferred spectrum occupancy for time index \( t \) is used as the current spectrum sensing result. Thus, it operates in real time. Since the algorithm does not treat any of the time indices differently, the performance of the algorithm is not affected by this change in implementation.

V. Theoretical Results
In this section, we study the two algorithms proposed in the previous sections.

Studying the structure learning algorithm, we analyze the probability density function of the proposed test statistic and provide upper bounds on the probability of erroneously inferring correlation (or independence) between two SUs.

The loopy BP algorithm is studied for its convergence properties. The existence of fixed points and its convergence to a fixed point is proven under restrictions on the domains \( D_m \).

A. Structure Learning
The structure learning algorithm consists of a set of pair-wise hypotheses tests followed by a method to combine the test results in order to infer the groups of correlated SUs. We analyze the performance of the pair-wise hypotheses tests by deriving the probability density function (PDF) of the test statistic conditioned on each hypothesis. Next, we analyze the probability of failure of multiple tests simultaneously. The
resulting observation justifies our choice of sub-optimal combination algorithm. We complete our analysis of the proposed structure learning algorithm by deriving upper bounds on the probability of erroneously including an SU in a group and the probability of erroneously excluding an SU from a group.

1) PDF of Test Statistic for Pair-Wise Tests: Consider distinct SUs indexed $i$, $j$, and $k$ and the test statistic $z(i,j,k)$ proposed in (19) to test whether SUs $i$ and $j$ are siblings, i.e., have conditionally correlated reports given the true spectrum occupancy. If we assume that $\hat{\rho}_{ik}/\hat{\rho}_{jk}$ is independent of $\hat{\rho}_{ik}/\hat{\rho}_{jk'}$ for $k \neq k'$, then we can approximate $z(i,j,k)$ as a Gaussian random variable – details are provided in Appendix B. If $y_i$ and $y_j$ are not siblings, i.e., if $\mathcal{H}_1(i,j)$ is true, then the mean and variance are as below. Here, we have ignored $O(1/\sqrt{T_i})$ terms.

$$E[z(i,j,k) | \mathcal{H}_1(i,j)] \approx \frac{K - 3}{K - 2} \frac{4\rho_{jk} \sigma \sqrt{T_i}}{\sqrt{\sum_{k' \neq i,j} \rho_{ik}^2 1/ \sigma_i^2 + 1/ \sigma_i^2}} \times \left[ \frac{\rho_{ik}}{\rho_{jk}} - \frac{1}{K - 3} \sum_{k' \neq i,j} \frac{\rho_{ik}}{\rho_{jk'}} \right]$$

$$\text{Var}[z(i,j,k) | \mathcal{H}_1(i,j)] \approx \frac{K - 3}{K - 2} \frac{16\sigma^2}{\sqrt{\sum_{k' \neq i,j} \rho_{ik}^2 1/ \sigma_i^2 + 1/ \sigma_i^2}} \times \left\{ \left( \frac{\rho_{ik}}{\rho_{jk}} \right)^2 \frac{\rho_{jk}^2}{\rho_{jk'}^2} \right\}$$

As with $\mathcal{H}_1(i,j)$, the variance of $z(i,j,k)$ is constant with time due to our chosen normalization. However, the mean approaches zero asymptotically because $\mu(i,j)$ is the maximum likelihood estimate of $E[\hat{\rho}_{ik}/\hat{\rho}_{jk} | \mathcal{H}_1(i,j)]$.

Given a fixed threshold $\tau$, we can now use the above derived mean and variance of the test statistic to estimate the probability of error for each of these tests.

2) Correlation between Tests: The test statistics $z(i,j,k)$ and $z(i,j,k')$ are computed from the same set of data. Therefore, the two test statistics are correlated.

**Theorem 4:** Consider 4 distinct SUs indexed $i$, $j$, and $k'$. The test statistics $z(i,j,k)$ and $z(i,j,k')$ are positively correlated.

**Proof:** Consider the test statistic $z(i,j,k')$. We can rewrite it in terms of $z(i,j,k)$ as follows.

$$z(i,j,k') = \frac{\hat{\rho}_{ik}/\hat{\rho}_{jk} - \hat{\mu}(i,j)}{\hat{\rho}_{jk}/\hat{\rho}_{jk'}} \left( \sqrt{\frac{\mu(i,j)^2}{\sigma_i^2}} + \frac{1}{\sigma_i^2} \right)$$

$$= \frac{\hat{\rho}_{ik}/\hat{\rho}_{jk} - \hat{\mu}(i,j)}{\hat{\rho}_{jk}/\hat{\rho}_{jk'}} \left( \sqrt{\frac{\mu(i,j)^2}{\sigma_i^2}} + \frac{1}{\sigma_i^2} \right) + \frac{\hat{\rho}_{jk}/\hat{\rho}_{jk'}}{\hat{\rho}_{jk}/\hat{\rho}_{jk'}} z(i,j,k)$$

Now, assume that it has been observed that $z(i,j,k) \leq \tau$. For the sake of readability, we omit the indices and use $z \triangleq z(i,j,k)$ and $z' \triangleq z(i,j,k')$ in the next few equations. Then,

$$P(z' \leq \tau | z \leq \tau) = P \left[ z \leq \left( \frac{\hat{\rho}_{ik}/\hat{\rho}_{jk} - \hat{\mu}(i,j)}{\hat{\rho}_{jk}/\hat{\rho}_{jk'}} \left( \sqrt{\frac{\mu(i,j)^2}{\sigma_i^2}} + \frac{1}{\sigma_i^2} \right) \right) \leq \tau \right]$$

$$\geq P \left[ z \leq \left( \frac{\hat{\rho}_{ik}/\hat{\rho}_{jk} - \hat{\mu}(i,j)}{\hat{\rho}_{jk}/\hat{\rho}_{jk'}} \left( \sqrt{\frac{\mu(i,j)^2}{\sigma_i^2}} + \frac{1}{\sigma_i^2} \right) \right) \right]$$

The above inequality is true because both $z(i,j,k)$ and $z(i,j,k')$ are random variables with infinite support. Hence, $z(i,j,k)$ and $z(i,j,k')$ are positively correlated.

This theorem suggests that two SUs should be declared as siblings only if $z(i,j,k)$ satisfies the two-sided test (20) for all $k \in \{1, \ldots, K\} \setminus \{i,j\}$, i.e., an AND rule.

3) Bounding Mislabeling Errors: We define mislabeling errors as an erroneous labeling of pairs of SUs as either siblings or non-siblings. A pair $y_i$ and $y_j$ are labeled as siblings if $|z(i,j,k)| \leq \tau$ for all $k \in \{1, \ldots, K\} \setminus \{i,j\}$ where $\tau = \Phi^{-1}(1-2\alpha/\beta)$. As proven in Theorem 4, these test statistics are positively correlated. The probability of erroneously labeling a pair $y_i$ and $y_j$ as siblings can be bounded as

$$P(y_i \text{ and } y_j \text{ erroneously labeled as siblings}) \geq P(\bigwedge_{k \neq i,j} |z(i,j,k)| \leq \tau | \mathcal{H}_1(i,j,k))$$

$$\leq \min_{k \neq i,j} P(|z(i,j,k)| \leq \tau | \mathcal{H}_1(i,j,k)) \geq 1 - \alpha/2$$

(36)
The bound can be computed using (32) and (33) for a given system. Due to the large size of the joint distribution of the reports, it is not possible to provide a bound applicable to all systems.

The probability of erroneously labeling a pair $y_i$ and $y_j$ as non-siblings can also be similarly bounded. However, the threshold in our test is chosen in order to bound this particular error probability. The bound and the actual error are both, therefore, flat as is illustrated in Fig. 7(b).

B. Sensing using Belief Propagation

The BP algorithm is known to be hard to analyze. We analyze the convergence of the BP with the best results available in literature. Our proposed algorithm is usually run on digital computers which requires that the domains of $r_m$ be discretized. For the purpose of the following proofs, we assume that each $r_m$ is discretized by a suitably chosen step size $\epsilon$ along each dimension.

First, we prove a sufficient condition for the existence of a fixed point.

Theorem 5: If the domain $\mathcal{D}_m$ of the variable nodes $r_m$ is strictly positive in each dimension for all $m \in \{1, \ldots, M\}$, then the proposed loopy BP algorithm has at least one fixed point.

Proof: The messages in the factor graph were initialized to uniform distributions to reflect our ignorance of their true value. Therefore, the initial messages are all non-negative.

Our factor graph contains $T_s \times M$ factor nodes each defined by (23) and bounded by the domain of the $r_m$ variable nodes. Hence, $\Psi_{m,t}(\cdot, \cdot) > 0$ iff the domain of $r_m$ is strictly positive.

By [22, Thm 4], there exists at least one fixed point of the BP algorithm if all the initial messages are non-negative and all factor nodes are strictly positive.

The existence of fixed point(s) does not imply that the algorithm will converge to a fixed point. Convergence is determined by the domain of the variable nodes, as described next.

Theorem 6: Let $r_{\min}$ (resp. $r_{\max}$) be the minimum (resp. maximum) value along any dimension across all $\mathcal{D}_m$ for each $m \in \{1, \ldots, M\}$. Assume that $r_{\min} > 0$ in order to satisfy Theorem 5. Then the proposed loopy BP will converge to a fixed point if

$$r_{\max} < \exp \left( 2 \tanh^{-1} \left[ \frac{1}{L - 1} \right] \right),$$

(37)

where $L \triangleq \max\{M, T_s\}$. If $r_{\max}$ and $r_{\min}$ are complementary, i.e., $r_{\max} + r_{\min} = 1$, then the proposed loopy BP will converge to a fixed point if

$$r_{\max} < \frac{\exp \left( 2 \tanh^{-1} \left[ \frac{1}{L - 1} \right] \right)}{1 + \exp \left( 2 \tanh^{-1} \left[ \frac{1}{L - 1} \right] \right)},$$

(38)

Proof: Define a function $Q : \{1, \ldots, T_s\} \times \{1, \ldots, M\} \rightarrow [-1, 1]$ as

$$Q(t, m) \triangleq \sup_{s \in \{0, 1\}} \tanh \left( \frac{1}{4} \log \frac{\Psi_{m,t}(s, r) \Psi_{m,t}(1 - s, r')}{\Psi_{m,t}(1 - s, r) \Psi_{m,t}(s, r')} \right).$$

(39)

We can evaluate $Q(t, m)$ using the maximum and minimum values along the dimensions of the simplex corresponding to the observed reports $y_{\bar{m}}(t)$. Further, it can be bounded by our definition of $r_{\max}$ and $r_{\min}$ as:

$$Q(t, m) \leq \tanh \left( \frac{1}{4} \log \frac{r_{\max}^2}{r_{\min}^2} \right) = \tanh \left( \frac{1}{2} \log \frac{r_{\max}}{r_{\min}} \right).$$

(40)

Rewriting Theorem 3 from [23] for factor nodes with just two neighboring variable nodes, we see that the BP algorithm will converge if

$$\max_{m,t} \sum_{t' \neq m} Q(t', m) < 1 \text{ and } \max_{m,t} \sum_{m' \neq m} Q(t, m') < 1.$$  

(41)

(42)

Substituting the bound (40) gives the requirements

$$(T_s - 1) \tanh \left( \frac{1}{2} \log \frac{r_{\max}}{r_{\min}} \right) < 1 \text{ and } (M - 1) \tanh \left( \frac{1}{2} \log \frac{r_{\max}}{r_{\min}} \right) < 1.$$ 

(43)

(44)

which can be combined as

$$(\max\{M, T_s\} - 1) \tanh \left( \frac{1}{2} \log \frac{r_{\max}}{r_{\min}} \right) < 1.$$  

(45)

Rearranging the terms results in the required condition (37).

Substituting $r_{\min} = 1 - r_{\max}$ into (37) gives (38).

Theorem 6 provides a very conservative bound on the domain of $r_m$. However, our simulations show that the BP
algorithm converges even if the domain is chosen to be larger than that allowed by Theorem 6. Specifically, our simulations are carried out by discretizing each dimension of $\mathcal{D}_m$ into 5 levels between $r_{\min} = 0$ and $r_{\max} = 1$.

VI. SIMULATION RESULTS

In this section, we provide simulation results to demonstrate the performance of our proposed structure learning and BP algorithms. We simulate cooperative spectrum sensing systems with K SUs each with identical sensing statistics of detection probability 0.95 and false alarm probability 0.1 unless specified otherwise. To ease discussion, we consider only pairs of colluding SUs, unless specified otherwise. Note that the proposed algorithm and theory is not similarly restricted.

We assume that all SUs in a group with correlated reports are statistically equivalent. This is a model for a particularly difficult attack strategy described by [14]. We assume that the BP algorithm is restricted to use only the sum of reports from correlated SUs. Thus, as described in Section IV, $r_m$ is a function of $\sum_{j \in J_m} y_j$ and $s$. $\mathcal{D}_m$ is a product of two simplexes each of $|J_m|$ dimensions.

The attack strategy of a group $J_m$ of SUs is defined by parameters $\alpha_0, \alpha_1, \ldots, \alpha_{|J_m|}$ as

$$P \left( \sum_{j \in J_m} y_j = \tau \mid \sum_{j \in J_m} u_j = \tau \right) \triangleq \begin{cases} \frac{\alpha_\tau}{1 - \alpha_\tau}, & \text{if } \tau = \tau' \\ \frac{\alpha_\tau}{|J_m|}, & \text{otherwise.} \end{cases}$$

At each Monte Carlo run, each $\alpha_\tau$ is chosen randomly from $[0.5, 1.0]$. We report the performance of our proposed algorithms averaged over all these attack strategies. Monte Carlo integration is used to integrate messages over $\mathcal{D}_m$.

A. Structure Learning

Our proposed structure learning algorithm learns the factorization of the joint distribution of the SU reports by learning the sibling nature of the nodes representing the SU reports. Performance of learning the factorization depends on the number of terms, i.e., number of groups of correlated SUs, as well as the conditional distributions, i.e., the attack strategies. We focus on the sample complexity of our algorithm which is defined as the number of samples $T$, required to achieve a specified probability of inferring the correct structure.

We compare our method to that proposed by Choi et al in [17] because both methods are based on the same principle of additive information distances over trees. Their algorithm has significantly increased sample complexity with increasing number of SUs as well as increasing number of colluding pairs. We believe that the primary reason for this, as described in the discussion after Corollary 1, is the use of $\log(\hat{\rho}_{ik}/\hat{\rho}_{jk})$ as a metric for clustering as opposed to our use of $\hat{\rho}_{ik}/\hat{\rho}_{jk}$. Further, they use a heuristic distance metric to identify the number and labels of clusters. As shown in [1], their probability of inferring the correct structure does not go to unity for most configurations.

The total number of SUs in the system does not affect the sample complexity of our proposed algorithm significantly, as shown in Fig. 5(a), because our algorithm is based on testing SUs pair-wise with bounded error probability. However, increasing the number of pairs of correlated SUs increases the sample complexity of our algorithm, as shown in Fig. 5(b). Increasing the number of pairs of correlated SUs reduces the distance between the ratio of correlation coefficients for different groups of SUs and, hence, distinguishing them requires a larger number of samples.

Our proposed method uses the Gaussianity of the ratio of correlation coefficients to derive the threshold for each pair wise test (20). As derived in Theorem 3, this requires the PU’s average spectrum occupancy to be approximately 0.5. However, Fig. 6 shows that this is not a strict requirement. The performance of our proposed algorithm is constant for a wide range of spectrum occupancy. When the spectrum is almost always occupied (or empty), then our proposed algorithm fails to learn the correct structure because it does not have enough samples to observe the joint distribution of the reports for both spectrum occupancy states.

Finally, we verify our derived upper bound on mislabeling errors. The bound (36) is specific to the particular system and attack strategy. Fig. 7(a) shows that the bound (36) is not valid for small number of samples. The reason is that the distribution of the sample correlation coefficients and its ratios is not Gaussian in that regime. Fig. 7(b) shows the probability of not inferring that a pair of correlated SUs are sibling SUs. Both the theoretical bound and simulation result are almost constant because our hypothesis test (20) is designed to maintain a constant detection probability. The observed value is lower than the target (0.05) because our test statistic uses an upper bound (16) of $\text{Var}(\hat{\rho}_{ik}/\hat{\rho}_{jk})$ instead of the exact variance.
B. Spectrum Sensing

We present simulation results to discuss two topics: convergence with increasing number of message passing iterations and the improvement in spectrum sensing accuracy by learning the correlations between SU reports in different systems.

Convergence: Our simulations use domains with $r_{\min} = 0$ and $r_{\max} = 1$ which does not satisfy the conditions for convergence as specified by Theorem 6. However, simulations show that the proposed loopy BP algorithm converges for all the systems considered. Fig. 8(a) shows that in a system with a pair of colluding SUs, the average beliefs converge by the 9th message passing iteration. Convergence in multiple systems can be verified by convergence of the error probability $P(\hat{s}(n)(t) \neq s(t))$ shown in Fig. 8(b). All the following performance results are reported at the 9th message passing iteration since the error probabilities always converge by then.

The naïve BP algorithm of [3] assumes that all SUs are independent and that their sensor’s statistics, i.e., detection and false alarm probabilities are known. In comparison, our proposed algorithm allows SUs to be correlated and assumes that their sensor statistics are unknown. The two differences between these systems are the correlations in SU reports and the assumption of sensor statistics. We elaborate on each of these differences. We observed that, in general, our algorithm improves the detection probability while the false alarm probability is approximately the same, as can be observed in Figs. 9(b) and 10(b). Hence, we focus on the detection probability in the following discussion.

Impact of the number of pairs of colluding SUs: Increasing the number of pairs of colluding SUs reduces the diversity

Fig. 9. Impact of increasing the number of pairs of colluding users. Reports collected for $T_s = 12$ time slots from $K = 8$ SUs.

Fig. 10. Impact of increasing the number of independent users. Reports collected for $T_s$ time slots, one pair of SUs collude while the rest are independent.
of the information received and, therefore, the probability of
detection reduces for all sensing algorithms. Also, the joint
distribution of the SU reports moves farther away from the
domain searched by the naïve algorithm but remains within the
domain $D_1 \times \ldots \times D_M$ searched by our proposed algorithm.
Therefore, the detection probability of our proposed algorithm
suffers less than the naïve algorithm, as shown in Fig. 9(a).

**Impact of the total number of SUs:** On the other hand,
increasing the total number of SUs increases the diversity in
the received reports and, hence, increases detection probability.
As independent SUs increases, the joint distribution of SU
reports approaches the domain searched by the naïve BP algo-
rithm. Therefore, though our proposed algorithm improves the
detection probability for all $K$, the gain reduces as independent
SUs increase, as shown in Fig. 10(a).

**Impact of Individual SU’s Detection Probability:** Increasing
individual SUs’ detection probability reduces the distance
between the true distribution of the SU reports and the prior
probability used by our algorithm. Hence, the error proba-
bility, shown in Fig. 11(a), is better than that of the naïve
algorithm when the individual SU’s detection probability is
sufficiently high, here 0.8. Figs. 11(b) and 11(c) show that
the improvement in error probability is caused more by the
improvement in the detection probability than that of the false
alarm probability. The detection probability converges to that
of the MAP algorithm while the false alarm probability does
not because the individual SU’s false alarm probability is held
constant.

**Effect of Increasing Number of Correlated SUs:** Next,
we consider the effect of increasing the size of a group
of correlated SUs while keeping the total number of SUs
constant. Analogous to the effect of increasing number of
pairs of colluding SUs, the detection probability reduces and
the false alarm probability increases for all sensing algorithms
because the diversity of information received reduces. This can
be seen in Fig. 12. Further, as the number of colluding SUs
increases, the detection probability of our proposed algorithm
suffers less than that of the naïve algorithm because the joint
distribution of the SU reports moves farther away from the
domain searched by the naïve algorithm but remains within the
domain searched by our proposed algorithm.

**Effect of Misidentification of Correlated SUs:** We have pro-
posed two algorithms such that the second algorithm (spectrum
sensing) uses the results from the first algorithm (structure
learning). It is possible that if the wrong structure is learned
then the spectrum sensing performance may be affected ad-
versely as is observed in [24]. In Fig. 13, we evaluate the
spectrum sensing performance when using insufficient number
of time slots to learn the structure. It is observed that learning
the wrong structure affects the spectrum sensing more when
the group of correlated SUs is large. However, note that the
spectrum sensing converges by about 1000 time slots while
there is only a 0.9 probability of learning the correct structure
from those reports.

**Effect of Varying Average Spectrum Occupancy:** In the
spectrum sensing algorithm, we have used a prior on the
improves the spectrum sensing accuracy when there is at least one group of correlated SUs. The improvement increases with increasing number of groups of correlated SUs. An upper bound for the probability of making an error in the learned structure is derived. The BP algorithm is shown to have fixed points and its convergence is proven under restrictions on the domains of the variable nodes.

Our proposed algorithms are designed specifically for binary reports. Future work would include similar algorithms for real-valued reports such as sensed energy or other parameters. As seen in [24], the potential attack strategies for real-valued reports are different than those for binary reports. In case of SUs reporting the received energy, a challenging situation in the presence of malicious users would be that the distribution of the reports need not be Gaussian and may be unknown. This requires a complete redesign of the structure learning method and the prior for $r_m$ for the sensing algorithm.

VII. CONCLUSION

In this work, a complete system is proposed to sense primary user spectrum occupancy from correlated reports by identifying the correlation structure present in the SU reports. Distinguishing between correlations caused by malicious behavior and that caused by environmental factors is proven to be impossible. Groups of correlated SUs are identified by learning the structure of the Bayesian network underlying the joint distribution of the spectrum occupancy and SU reports. The learned structure is used to construct a loopy BP based algorithm to sense spectrum occupancy without a priori assuming any model for the cause of the correlations. Simulations show that learning the identities of correlated SUs improves the spectrum sensing accuracy when there is at least one group of correlated SUs. The improvement increases with increasing number of groups of correlated SUs. An upper bound for the probability of making an error in the learned structure is derived. The BP algorithm is shown to have fixed points and its convergence is proven under restrictions on the domains of the variable nodes.

Our proposed algorithms are designed specifically for binary reports. Future work would include similar algorithms for real-valued reports such as sensed energy or other parameters. As seen in [24], the potential attack strategies for real-valued reports are different than those for binary reports. In case of SUs reporting the received energy, a challenging situation in the presence of malicious users would be that the distribution of the reports need not be Gaussian and may be unknown. This requires a complete redesign of the structure learning method and the prior for $r_m$ for the sensing algorithm.

APPENDIX A

PROOF OF THEOREM 3

Proof: Start by rewriting the sample correlation coefficient $\hat{\rho}_{ij}$ from (12) as

$$\hat{\rho}_{ij} = \frac{\hat{\rho}_{ij}(1, 1) - \hat{m}_i \hat{m}_j}{\sqrt{\hat{m}_i(1 - \hat{m}_i)\hat{m}_j(1 - \hat{m}_j)}}$$  (46)

using the estimators $\hat{\rho}_{ij}(1, 1) = \frac{1}{T} \sum_{t=1}^{T} y_i(t)y_j(t)$ and $\hat{m}_i = \frac{1}{T} \sum_{t=1}^{T} y_i(t) = \frac{1}{T} \sum_{t=1}^{T} y_i^2(t)$ which rely on $y_i(t) \in \{0, 1\}$ for all $i \in \{1, \ldots, K\}$. Algebraic manipulations give us a form in which we may use the Central Limit theorem to describe
the distribution of $\hat{\rho}_{ij}$. We use the facts that $\hat{m}_i = \hat{p}_{ij}(1,0) + \hat{p}_{ij}(1,1)$ and $\hat{m}_j = \hat{p}_{ij}(0,1) + \hat{p}_{ij}(1,1)$.

$$\hat{\rho}_{ij} = \frac{\hat{p}_{ij}(1,1) - (\hat{p}_{ij}(1,0) + \hat{p}_{ij}(1,1))(\hat{p}_{ij}(0,1) + \hat{p}_{ij}(1,1))}{\sqrt{\hat{m}_i(1-\hat{m}_i)\hat{m}_j(1-\hat{m}_j)}}$$

$$= \frac{\hat{p}_{ij}(1,1)\hat{p}_{ij}(0,0) - \hat{p}_{ij}(1,0)\hat{p}_{ij}(0,1)}{\hat{\sigma}_i\hat{\sigma}_j}$$

$$= \left(\frac{\hat{p}_{ij}(1,1)\hat{p}_{ij}(0,0) - (\hat{p}_{ij}(1,1) - \hat{p}_{ij}(0,0))^2}{4\hat{\sigma}_i\hat{\sigma}_j}\right)$$

$$- \left(\frac{\hat{p}_{ij}(1,0) + \hat{p}_{ij}(0,1)}{4\hat{\sigma}_i\hat{\sigma}_j}\right)^2$$

Rearranging the terms, we get

$$\hat{\rho}_{ij} = \frac{(\hat{p}_{ij}(1,1) + \hat{p}_{ij}(0,0))^2 - (\hat{p}_{ij}(1,0) + \hat{p}_{ij}(0,1))^2}{4\hat{\sigma}_i\hat{\sigma}_j}$$

$$- \frac{\hat{p}_{ij}(1,1) + \hat{p}_{ij}(0,0) - \hat{p}_{ij}(1,0) - \hat{p}_{ij}(0,1)}{4\hat{\sigma}_i\hat{\sigma}_j}$$

$$= \frac{1 - (2\hat{m}_i - 1)(2\hat{m}_j - 1) - \hat{p}_{ij}(1,0) + \hat{p}_{ij}(0,1)}{2\hat{\sigma}_i\hat{\sigma}_j}. \quad (47)$$

Note that $m_i$ and $m_j$ are close to 0. Hence, $\hat{m}_i$, $\hat{m}_j$, $\hat{\sigma}_i$, and $\hat{\sigma}_j$ have comparatively smaller error than $\hat{\rho}_{ij}$ and can be assumed to be perfectly estimated. By the Central Limit Theorem, for sufficiently large $T_i$, $\hat{\rho}_{ij}(1,0) + \hat{\rho}_{ij}(0,1)$ is Gaussian distributed. Since $\hat{\rho}_{ij}(1,0) + \hat{\rho}_{ij}(0,1) \overset{\Delta}{=} \delta$ is the mean of a Bernoulli random variable $1_{y_i = y_j = 1}$, we have

$$\hat{\rho}_{ij}(1,0) + \hat{\rho}_{ij}(0,1) \sim N(\delta, \delta([1 - \delta])) \quad (48)$$

Using the fact that $1 - \delta = \hat{p}_{ij}(0,0) + \hat{p}_{ij}(1,1)$ and that the variance of a constant is 0, we get the mean and variance of $\hat{\rho}_{ij}$ as (49) and (50) respectively.

$$E[\hat{\rho}_{ij}] = \hat{\rho}_{ij}$$

$$Var[\hat{\rho}_{ij}] = \frac{(\hat{p}_{ij}(1,0) + \hat{p}_{ij}(0,1))(\hat{p}_{ij}(1,1) + \hat{p}_{ij}(0,0))}{4T\hat{\sigma}_i^2\hat{\sigma}_j^2}. \quad (50)$$

Therefore, $\hat{\rho}_{ik}/\hat{\rho}_{jk}$ is the ratio of two Gaussian random variables — $\hat{\rho}_{ik}$ and $\hat{\rho}_{jk}$. The ratio of two Gaussian random variables can be approximated by a Gaussian distribution if the denominator has coefficient of variation (defined as the ratio of the standard deviation to the mean) less than 0.09 [25]. The coefficient of variation of the numerator, $\hat{\rho}_{jk}$, is $\frac{\hat{\rho}_{jk}}{\sqrt{T_i}\hat{\rho}_{jk}}$. Hence, for $T_i > \frac{0.0081}{\hat{\rho}_{jk}^2}$, its coefficient of variation will be less than 0.09 and the ratio $\hat{\rho}_{ik}/\hat{\rho}_{jk}$ will be Gaussian distributed.

By using a second order Taylor series, $E\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right]$ can be approximated as

$$E\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right] \approx \frac{E[\hat{\rho}_{ik}]}{E[\hat{\rho}_{jk}]} + \frac{Var[\hat{\rho}_{ik}]E[\hat{\rho}_{jk}]}{E[\hat{\rho}_{jk}]^3}. \quad (51)$$

The approximation residue is of the order of $T_i^{-2}$. We use the fact that $\hat{\rho}_{ik}$ and $\hat{\rho}_{jk}$ estimate distinct quantities and their correlation reduces with increasing $T_i$, i.e., $\hat{\rho}_{ik}$ and $\hat{\rho}_{jk}$ are independent when $T_i$ is sufficiently large. To achieve the same order of magnitude for the approximation residue of $Var\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right]$, a first order Taylor series expansion suffices and results in

$$Var\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right] \approx Var[\hat{\rho}_{jk}]E[\hat{\rho}_{ik}]^2 + Var[\hat{\rho}_{ik}]E[\hat{\rho}_{jk}]^2. \quad (52)$$

Substituting (49) and (50) into (51) and (52) gives the required mean (13) and variance (14).

**APPENDIX B**

**DISTRIBUTION OF $z(i,j,k)$**

The numerator of the test statistic (19) can be rewritten as:

$$\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}} - \hat{\mu}(i,j) = \hat{\rho}_{ik} - \frac{1}{K-2} \sum_{k' \neq i,j}^{K} \frac{\hat{\rho}_{ik'}}{\hat{\rho}_{jk'}}$$

$$= \frac{K - 3}{K - 2} \left(\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right) - \frac{1}{K - 2} \sum_{k' \neq i,j,k}^{K} \left(\frac{\hat{\rho}_{ik'}}{\hat{\rho}_{jk'}}\right). \quad (53)$$

Conditioning on nodes $y_i$ and $y_j$ being siblings, i.e., hypothesis $H_{ij}(i,j)$ is true, its conditional mean can be computed by Theorem 3.

$$E\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}} - \hat{\mu}(i,j) \mid H_{ij}(i,j)\right]$$

$$= \frac{K - 3}{K - 2} \left(\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right) \left[1 + \frac{1}{T_i} \hat{\rho}_{jk}\right]$$

$$- \frac{1}{K - 2} \sum_{k' \neq i,j,k}^{K} \hat{\rho}_{ik'} \left[1 + \frac{1}{T_i} \hat{\rho}_{jk'}\right]$$

$$= \frac{1}{T_i(K - 2)} \left(\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right) \left[(K - 3) + \sum_{k' \neq i,j,k}^{K} \frac{\hat{\rho}_{ik'}}{\hat{\rho}_{jk'}}\right]. \quad (54)$$

Conditioning on $H_{ij}(i,j)$ being true does not allow such simplification. The resulting mean is

$$E\left[\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}} - \hat{\mu}(i,j) \mid H_{ij}(i,j)\right]$$

$$= \frac{K - 3}{K - 2} \left(\frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}}\right) \left[1 + \frac{1}{T_i} \hat{\rho}_{jk}\right]$$

$$- \frac{1}{K - 2} \sum_{k' \neq i,j,k}^{K} \frac{\hat{\rho}_{ik'}}{\hat{\rho}_{jk'}} \left[1 + \frac{1}{T_i} \hat{\rho}_{jk'}^2\right]. \quad (55)$$

To compute the variance of the numerator, we assume that the estimates of $\hat{\rho}_{ik}$ are independent for all $k$. Though not true for small $T_i$, the correlation between the estimates reduces when they reach their true values. Unfortunately, conditioning on the hypothesis $H(i,j)$ does not simplify the variance expression.
Using (53) and substituting from (14), we get
\[
\begin{align*}
\text{Var} \left[ \frac{\hat{\rho}_{ik}}{\hat{\rho}_{jk}} - \hat{\mu}(i, j) \right] & = \left( \frac{K - 3}{K - 2} \right)^2 \frac{1}{T_{ip_{jk}^2}} \left[ \sum_{k \neq i, j, k} \frac{1}{T_{ip_{jk}^2}} \left( \frac{\hat{\rho}_{ik}^2}{\hat{\rho}_{jk}} \right) \right. \\
& + \left. \left( \frac{K - 3}{K - 2} \right)^2 \sum_{k' = 1}^{K} \frac{1}{T_{ip_{jk}^2}} \left( \frac{\hat{\rho}_{ik}^2}{\hat{\rho}_{jk}} \right) \right] \left( \hat{\rho}_{ik}^2 + \hat{\rho}_{jk}^2 \right).
\end{align*}
\]

The test statistic \( z(i, j, k) \) is normalized by the upper bound of \( \text{Var}(\hat{\rho}_{ik}/\hat{\rho}_{jk}) \) because the estimation error in the upper bound \( \hat{\sigma}_{up}(i, j, k)/\sqrt{T_i} \) is expected to be less than that for estimating \( \text{Var}(\hat{\rho}_{ik}/\hat{\rho}_{jk}) \). Hence, we assume that \( \hat{\sigma}_{up}(i, j, k) \) is a noiseless estimate. By the transformations \( E[x/c] = E[x]/c \) and \( \text{Var}[x/c] = \text{Var}[x]/(c^2) \) for random variable \( x \) and constant scalar \( c \), we can compute the mean and variance of \( z(i, j, k) \) conditioned on the appropriate hypothesis \( \mathcal{H}(i, j) \) to get (32)–(35) from (15) and (54)–(56).

REFERENCES


