Abstract—Consider spectrum sensing systems where the presence of the primary user of the spectrum is detected by secondary users (SUs) in a centralized cooperative fashion. The sensing results can be correlated due to environmental reasons. SUs may or may not be honest. If dishonest, they could be colluding. Existing methods to fuse the SU reports either ignore the correlation in the SU reports, or they need to know the source of correlation. In this paper, we propose a belief propagation based fusion algorithm to exploit the correlations in reports of groups of SUs irrespective of cause. We show that identifying the groups of SUs having correlated reports reduces the probability of error of spectrum sensing. Our method is based on modeling the probability distribution underlying the SU reports as a Bayesian network. The process of learning the Bayesian network also shows that it is theoretically impossible to identify collusion.

I. INTRODUCTION

Cognitive radios fuse their sensing results in order to improve spectrum sensing accuracy in the presence of fading channels [1]. We consider centralized cooperative spectrum sensing systems where secondary users (SUs) report their estimates of spectrum occupancy to a fusion center. The fusion center attempts to increase the accuracy of estimating the spectrum occupancy by combining the reports of all the SUs. The accuracy of the fusion algorithm is affected by various environmental factors and system parameters. In this paper, we focus on the effects of correlations in the SU reports.

Since all SUs measure the same quantity – primary user (PU) spectrum occupancy – the SU reports at each time slot are correlated. However, given the true value of the PU spectrum occupancy, the SU reports may or may not be conditionally independent. They could be conditionally dependent given the spectrum occupancy due to either environmental reasons, or malicious behavior. In this scenario, we are interested in estimating the value of the sensed quantity and identifying malicious behavior by the SUs.

The fading channels from the primary user to the SUs have a correlation which increases exponentially as the distance between the SUs reduces [2]. Statistical data falsification attacks by malicious attackers also result in correlations in the reports of the malicious SUs if they are colluding [3]. For the sake of readability, we abuse notation slightly and call SU reports independent (correlated) if they are conditionally independent (dependent) given the true spectrum occupancy.

Various methods have been proposed in literature to detect the PU spectrum occupancy despite correlations in SU reports, or to use these correlations to identify malicious SUs. An adaptive reinforcement learning based algorithm is proposed in [4] to estimate the Neyman-Pearson sufficient test statistic for detecting the PU spectrum occupancy. A lower bound on the probability of error for detecting the PU spectrum occupancy when the SUs are placed in a particular configuration and experience correlated channel fading was derived in [5]. Location information and the corresponding correlations in channel fading are used in [6] to detect malicious SUs in the cooperative spectrum sensing system. However, these methods are either computationally expensive because they attempt to learn the joint distribution of all the SU reports without factorizing it, or they depend on a particular cause of the correlations in SU reports.

In this work, we propose sensing the spectrum without assuming the cause for the correlations in the SU reports. To perform such sensing in a computationally efficient manner, we use loopy belief propagation (BP) to learn a factorized form of the joint distribution of the SU reports. This factorization is based on a Bayesian network model for the joint distribution. Loopy BP has been used in the past to detect spectrum occupancy when the sensing results of all SUs are independent, and malicious SUs do not collude [7]. We show, through simulations, that our proposed BP algorithm has higher accuracy than the afore mentioned BP algorithm which ignores correlations in SU reports. We also show that in an environment where SUs’ sensing results could be correlated, it is impossible to identify colluding SUs.

This paper is organized as follows. Section II describes the system model and introduces the Bayesian network representation. Section III proves that correlation due to environment cannot be distinguished from collusion. The proposed loopy belief propagation algorithm for estimating the PU spectrum occupancy is described in Section IV. The graph structure learning algorithm used to identify groups of correlated SUs is described in Section V. Section VI evaluates the proposed method by simulations. Section VII concludes the paper.

This work has been supported by the National Science Foundation under CNS grant 1117600.
II. System Model and Notation

We consider a cognitive radio network where secondary users sense the presence of one PU and report their binary estimates to a fusion center. Let there be $K$ SUs in the network, indexed by $1, 2, \ldots, K$. At the $i$'th time slot, let the PU spectrum occupancy be denoted by $s(i) \in \{0, 1\}$, where $s(i) = 0$ if the PU is not transmitting and $s(i) = 1$ otherwise. We assume that we do not have any knowledge of the PU transmission activity statistics. Hence, $s(i)$ is assumed to be i.i.d. at each time slot. At each time slot $i$, the $k$'th SU generates an estimate $u_k(i) \in \{0, 1\}$ of $s(i)$. It reports a value $y_k(i) \in \{0, 1\}$ to the fusion center. The fusion center combines these reports to form an estimate $\hat{s}(i)$ of $s(i)$.

SU sensing results $u_k(i)$ could be correlated due to correlated fading environments [2]. Each SU is mapped to exactly one of $C$ non-empty sets $\mathcal{H}_1, \ldots, \mathcal{H}_C$ such that two SUs $k_1$ and $k_2$ are mapped to different sets if the sensing results $u_{k_1}(i)$ and $u_{k_2}(i)$ are conditionally independent given $s(i)$.

The $k$'th SU is considered to be honest if $P(y_k(i) = u_k(i)) = 1$. Otherwise it is considered to be malicious. SUs can be malicious on their own, or they can collude, i.e., modify their reports collaboratively. Each SU is mapped to exactly one of $N$ non-empty sets $\mathcal{G}_1, \ldots, \mathcal{G}_N$ such that two SUs $k_1$ and $k_2$ are mapped to different sets if the reports $y_{k_1}(i)$ and $y_{k_2}(i)$ are conditionally independent given $u_{k_1}(i)$ and $u_{k_2}(i)$. SUs that behave in an independent fashion form a singleton set in this set partition. For $n \in \{1, \ldots, N\}$, the reports $y_{\mathcal{G}_n}(i)$ are assumed to be generated from the conditional distribution $P(y_{\mathcal{G}_n}(i)|u_{\mathcal{G}_n}(i))$.

We assume that the joint distribution of the PU spectrum occupancy, the SU results, and malicious behavior is time invariant.

A. Bayesian Network

The joint distribution of $s$, sensing results $u_k$, and reports $y_k$ ($k \in \{1, \ldots, K\}$) can be factorized as

$$P(s, u_1, \ldots, u_K, y_1, \ldots, y_K) = P(s)P(u_1, \ldots, u_K|s) \times P(y_1, \ldots, y_K|u_1, \ldots, u_K)$$

$$= P(s) \prod_{c=1}^{C} P(u_{\mathcal{H}_c}|s) \prod_{n=1}^{N} P(y_{\mathcal{G}_n}|u_{\mathcal{G}_n}).$$

(1)

This factorization is consistent with a Bayesian network, as shown in Figure 1. Correlation due to environmental reasons is modeled by the nodes $X_{H_c}$ for $c \in \{1, \ldots, C\}$. Possible malicious behavior is modeled by the nodes $A_{G_n}$ for $n \in \{1, \ldots, N\}$. Thus, the joint distribution underlying the spectrum sensing system can be modeled as a Bayesian network.

III. Identifying the Source of Correlation

Correlations between SU reports could be caused by correlations in the sensing, or by collusion. By the following theorem, we show that the joint distribution of the SU reports is not sufficient to distinguish between these two phenomena.

Theorem. If the reports of a group of SUs are conditionally dependent given the true spectrum occupancy, then the joint distribution of the reports is not sufficient to identify whether the SUs are colluding, or not.

Proof. If the SUs have correlated reports, then the SUs could either have correlated sensing results, or they could be colluding. From (1), 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 [8]. In effect, 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 [8]. Therefore, nodes representing

1) the SUs’ sensing results,
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 Figure 1. SUs 1 and 2 are honest, but their sensing results are not
conditionally independent given \( s \). SUs 3 and 4 are colluding, and their sensing results are conditionally independent given \( s \). Nodes \( u_1, u_2, u_3, u_4 \), \( X_{[3]} \) and \( X_{[4]} \) represent latent variables, and have just 2 neighbors each. Therefore, their presence cannot be learned from the joint distribution of the leaf nodes. In other words, we cannot identify which SUs are colluding and which SUs have correlated sensing results. The structure that we would learn is shown in Figure 2.

Note that the structure learned is a tree. The tree 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.

IV. Estimation by Belief Propagation

For this section, we assume that the structure of the Bayesian network has been learned perfectly. This implies that we know the set partition \( \mathcal{F}_1, \ldots, \mathcal{F}_M \) such that \( y_k(t) \) and \( y_{k'}(t) \) are conditionally independent given \( s(t) \) iff SUs indexed \( k \) and \( k' \) are in separate partitions. Learning the structure is difficult because of its high dimensionality. The Bayesian network helps us factorize the joint distribution based on the groups of correlated reports, thus reducing the learning complexity. The sum-product algorithm, also known as belief propagation, has been shown to be a computationally efficient method of estimating the posterior distribution [9] when such a factorization is possible. For running the belief propagation, a factor graph is constructed from the learned Bayesian network, similar to the method proposed in [7] for independent, possibly malicious, SUs.

Consider a factor graph with two types of variable nodes and a single type of factor nodes, as shown in Figure 3. The variable nodes represent the unknowns in our system: \( s(t) \) for \( t \in \{1, \ldots, T\} \), and the conditional distribution \( P(y_t|s) \) for \( m \in \{1, \ldots, M\} \), \( s \in \{0, 1\} \). The factor node \( \Psi_{m,t} \) takes \( y_{m,t} \) as a deterministic input and computes a function of \( s(t) \) and \( r_m \). We define it as

\[
\Psi_{m,t}(s, r) \triangleq P(y_{m,t}|s(t)) = r(s, y_{m,t}(t)),
\]

where the second equality is a result of the definition of \( r_m \).

Belief propagation approximates the posterior distribution of \( s(t) \) by an iterative exchange of messages along the edges of the factor graph. We consider the parallel update scheme of message passing, i.e., at each iteration, messages are sent out from all nodes simultaneously.

At the \( l \)th iteration, the messages sent from the variable nodes to the factor nodes are as follows.

\[
\mu_{s(t)\rightarrow\Psi_m}^{(l)}(s) \propto P(s) \prod_{m' \neq m} \mu_{s(t)':\rightarrow\Psi_m}^{(l-1)}(s)
\]

\[
\mu_{r_m\rightarrow\Psi_m}^{(l)}(r) \propto P(r_m = r) \prod_{r_m' \neq r_m} \mu_{s(t)':\rightarrow\Psi_m}^{(l-1)}(r)
\]

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

\[
\mu_{\Psi_m\rightarrow r_m}^{(l)}(r) \propto \sum_{s \in \{0, 1\}} \Psi_{m,t}(s, r) \mu_{s(t)'\rightarrow\Psi_m}^{(l-1)}(s)
\]

\[
\mu_{\Psi_m\rightarrow \mathcal{D}_m}^{(l)}(s) \propto \int_{\mathcal{D}_m} \Psi_{m,t}(s, r) \mu_{r_m\rightarrow\Psi_m}^{(l-1)}(r) dr
\]

Here, the prior probabilities for \( r_m \) and \( s \) are denoted by the notation \( p(\cdot) \). We assume that \( r_m \) can take values from the set \( \mathcal{D}_m \). The choice of \( \mathcal{D}_m \) depends on the correlation structure possible due to the environment, and the attack strategies that we wish to consider.

Note that \( r_m \) is defined as \( P(y_{m,t}(t)|s(t)) \). It cannot be factorized further without prior knowledge of the structure of the correlations between SU reports. In the most general case, \( r_m(0, y_{m,t}) \) is independent of \( r_m(1, y_{m,t}') \), e.g., consider the case of independent noise and correlated fading. Also, \( \sum_s r_m(s, y) = 1 \) for \( s \in \{0, 1\} \). Therefore, \( r_m \) takes values from the product of two identical simplexes each of \( (2^{|\mathcal{F}_m|} - 1) \) dimensions. Later in simulations we will use a simpler model in order to reduce the computational complexity.

The prior probability of \( s(t) \) is assumed to be uniform on \( \{0, 1\} \). For \( r_m \), we assume the prior probability to correspond to the conditional distribution of perfect honest SUs.
V. Learning Structure of Graphical Model

In this section, we adapt a structure learning algorithm to identify the set partitioning \( \{F_m\}_{m=1}^M \) by learning the structure of the Bayesian network. There is a wide variety of algorithms in literature to learn the structure of the latent Bayesian network. Quartet-based algorithms have been shown to be very effective in learning latent trees [10] [11] [12].

Consider the learnable structure of the tree, as discussed in Section III. The reports of a group of SUs which are colluding, or have correlated sensing results are children of the same node. They are sibling nodes. Therefore, we do not need to learn the entire structure of the latent tree in order to identify the groups of SUs having correlated reports. Only the sibling relations between the SU reports need to be learned to reconstruct the entire structure of the tree. We use this fact to reduce the search space of possible structures as compared to learning the whole structure.

The problem with this approach is that SUs with conditionally independent reports are also sibling nodes. If we know the identity of at least one independent SU, we can identify the group of sibling nodes that are independent. For the purpose of this work, we assume that we do know the identity of at least one independent SU.

The algorithm proposed in [13] learns the structure of latent trees by using the additivity of information distances. We adapt a part of their algorithm to learn the sibling relations between SU reports. Information distance between nodes \( k \) and \( n \) in a graph is defined as

\[
g_{kn} = \frac{|\text{det} W_{kn}|}{\sqrt{\text{det} V_k \text{det} V_n}},
\]

where \( W_{kn} \) is a matrix of the joint probability mass function of nodes \( k \) and \( n \), and \( V_k \) is the diagonal matrix of the marginal probability mass function of node \( k \). When both nodes take binary values, \( g_{kn} \) is the logarithm of the Pearson’s correlation coefficient of nodes \( k \) and \( n \). Information distances are additive tree metrics, i.e., for any two nodes \( k \) and \( n \),

\[
g_{kn} = \sum_{i,j \in \text{Path}(k,n)} g_{ij},
\]

where \( \text{Path}(k,n) \) is a path from node \( k \) to node \( n \) in the tree.

Nodes \( k \) and \( n \) are siblings if and only if \( g_{kj} - g_{nj} = g_{kj'} - g_{nj'} \) for all \( j, j' \in \{1, 2, \ldots, K\} \setminus \{k,n\} \). If the system has 4 or more SUs, then for an approximate result, we can use only SU reports to check sibling relations.

The empirically computed information distances are noisy. Let \( \hat{g}_{ij} \) denote the noisy estimate of the information distance of nodes \( i \) and \( j \). Instead of the equality test described above, the authors of [13] propose using a clustering algorithm on \( \max(\hat{g}_{kj} - \hat{g}_{nj}) - \min(\hat{g}_{kj} - \hat{g}_{nj}) \) to increase robustness to noise.

For simulation purposes, we use a modified K-Means algorithm to cluster the nodes. The optimal number of clusters is decided at run-time by the Silhouette method [14]. The number of clusters is limited by the fact that there can be at most one cluster with a single node. All other clusters have at least two nodes. Therefore, the number of clusters is at most \( \lceil K/2 \rceil \).

The Silhouette method can compute weights for the clusters found by the clustering algorithm only if more than one cluster has been identified. For identifying all independent SUs, we use the fact that the variance of the Monte Carlo estimate is approximately the inverse of the number of samples \( T \) used for estimating the distribution. We claim that all the SU reports are independent if \( \max(\hat{g}_{kj} - \hat{g}_{nj}) - \min(\hat{g}_{kj} - \hat{g}_{nj}) < \frac{1}{T} \) for all nodes \( k \) and \( n \), where \( T \) is an appropriately chosen threshold.

VI. Simulation Results

Graph structure learning has been shown to be a NP-hard problem [8], and the convergence of the belief propagation algorithm is an open problem [9] for most classes of graphs. Keeping these difficulties in mind, we present simulation results to evaluate the performance of our proposed algorithm.

A. Simulation System

The system parameters are the number of SUs, the sensing detection probability, the configuration in terms of groups of correlated SUs, and the attack strategies for the SUs.

As noted earlier, \( r_m \) belongs to a product space of two simplexes, each of \( (2^{|F_m|} - 1) \) dimensions. Therefore, the domain of the distribution of \( r_m \) for a pair of correlated SUs has \( 2^2 = 4 \) dimensions, and that for a group of 3 correlated SUs has \( 2^3 = 8 \) dimensions. Computing \( \mu_{\alpha,y} \) using (7) requires integration over this product space. Numerically integrating over such a large space is computationally very expensive. We reduce the computational complexity by assuming that within a group of correlated SUs, all SUs are statistically equivalent. Such an attack strategy reduces the effectiveness of outlier detection algorithms [15]. The belief propagation algorithm is restricted to consider only the sum of the reports from each group. In this case, \( r_m \) is a function of \( \Sigma y_j \in J \) and \( s \). Hence, the domain of the distribution of \( r_m \) is a product of two simplexes each of \( |F_m| \) dimensions. This assumption reduces the number of dimensions to \( 2^2 = 4 \) for a pair of correlated SUs, and \( 2^3 = 9 \) for a group of three correlated SUs.

For a group \( F_m \) of malicious SUs, we define the attack strategy as follows. Define parameters \( \alpha_0, \alpha_1, \ldots, \alpha_{|F_m|} \) as \( \alpha_t \triangleq P(\Sigma y_j = t | \Sigma u_{j \in J} = \tau) \), and \( P(\Sigma y_j \neq t | \Sigma u_{j \in J} = \tau) \triangleq (1 - \alpha_t)/|F_m| \), where \( t \in \{0, 1, \ldots, |F_m|\} \). In our simulations, we choose each \( \alpha_t \) randomly from \([0.5,1.0]\) at each Monte Carlo run. Our results, therefore, show the performance averaged over all these attack strategies.

B. Estimation of Spectrum Occupancy

First, we show that our proposed loopy belief propagation algorithm converges with increasing number of message passing iterations. Next, the value of knowing the correlation structure is studied by comparing our algorithm with a “naïve” BP-based algorithm [7] which ignores correlations in SU reports. Paradoxically, our proposed algorithm has lower error probability than the existing algorithm irrespective of whether SUs colluding or not. Finally, we evaluate our algorithm as the detection probability of individual SUs is varied.

From simulations, we observe that 9 message passing iterations are sufficient for the beliefs to converge. In Figure 4, we show that the probability of error in estimating \( s(t) \) converges
by the 9th message passing iteration for systems with a pair of colluding SUs. For the remaining results described next, we report the probability of error observed at iteration 9.

Increasing the number of SUs increases the number of reports available to the fusion center. But this also increases the number of variables $r_m$ to be estimated. Keeping the total number of SUs constant, increasing the number of colluding SUs reduces the diversity of the reports received. Hence, we can expect that an increase in the number of independent SUs reduces the probability of error significantly. On the other hand, increasing the number of colluding SUs increases the probability of error. Collecting reports for more time slots improves the estimate of the underlying distribution. Therefore, an increase in the number of time slots improves the probability of error. However, the improvement is not large.

These properties can be seen in Figure 5 which shows simulation results of systems with an increasing number of independent SUs, and Figure 6 which shows the performance with increasing number of colluding SUs. From simulations, we also note that, compared to the naïve BP which ignores correlations in SU reports, our proposed algorithm reduces the error probability by having a higher detection probability. The false alarm probability is not significantly different. This is a result of choosing a prior probability for $r_m$ which favors high detection probability.

Finally, we consider the effect of sensor quality on the algorithm performance. Our proposed algorithm assumes no knowledge of the sensing statistics of individual SUs. In particular, it a priori assumes that the SU is honest and has perfect detection and false alarm probabilities, i.e., $P_d = 1$ and $P_{fa} = 0$. Therefore, we would expect that the proposed algorithm performs best when the SUs’ sensors are perfect. In Figure 7, we consider a system of 8 SUs reporting for 12 time slots and compare the performance of our algorithm with the naïve BP algorithm. As expected, the error probability of our proposed algorithm approaches that for the MAP algorithm as the individual SU’s detection probability increases. In comparison with the naïve BP algorithm, our algorithm has higher accuracy when the individual SU’s detection probability is above 0.6. In spectrum sensing, the false alarm probability is a function of the noise power and the detection probability is a function of the SNR. Therefore, studying the performance of our proposed algorithm with different detection probabilities is equivalent to studying its performance in environments with different SNRs. We can conclude that using the correlations in the SU reports is useful when the SNR is high enough for an individual SU to achieve 0.6 detection probability.

C. Identification of Structure of Bayesian Network

The groups of SUs with correlated reports are identified by learning the structure of the Bayesian network underlying the joint distribution of the SU reports and spectrum occupancy, as described in Section V. We test this algorithm by simulating systems whose corresponding Bayesian networks have different structures. Since we test only the sibling relationships to reconstruct the entire tree, this requires that we simulate systems with different groups of correlated SUs.

The performance of the algorithm depends on the joint distribution of the reports of each of the groups of these SUs. The empirical estimate of this joint distribution also depends on the size of the group of correlated SUs. In order to keep the simulations manageable, we restrict our study to pairs of colluding SUs.
In this paper, we have proposed a belief propagation based algorithm for fusing correlated spectrum measurement reports in a cooperative spectrum sensing system. The algorithm recognizes that the joint distribution of the SU reports and spectrum occupancy can be factorized as a Bayesian network. Through simulations, we show that our proposed loopy BP algorithm reduces the probability of error by increasing the probability of detection as compared to a loopy BP algorithm which does not account for the correlated nature of reports.

By analyzing the Bayesian network model, we show that it is impossible to identify colluding SUs if the sensing environment could be causing correlations in the SU reports. We also adapt a latent tree structure learning algorithm in order to group SUs based on correlations in reports.

In future, we will be analyzing the performance of our proposed loopy belief propagation algorithm. Also, we will be proposing improvements to the structure learning algorithm in order to improve the sample complexity.

REFERENCES