Abstract—In cognitive radio (CR) technology, the trend of sensing is no longer to only detect the presence of active primary users. A large number of applications demand for the primary user behavior in spatial, temporal, and frequency domains. To satisfy such requirements, we study the statistical relationship of primary users by introducing a Bayesian network (BN) based framework. How to learn such a BN structure is a long standing issue, not fully understood even in the statistical learning community. Besides, another key problem in the learning scenario is that the CR has to identify how many variables in the BN, which is usually considered as prior knowledge in statistical learning applications. To solve such two issues simultaneously, this paper proposes a BN structure learning scheme consisting of an efficient structure learning algorithm and a blind variable identification algorithm. The proposed scheme incurs significantly lower computational complexity compared with previous ones, and is capable of determining the structure without assuming any prior knowledge about variables. With this scheme, cognitive users could efficiently understand the statistical pattern of primary networks, such that more efficient cognitive protocols could be designed across different network layers.

I. INTRODUCTION

Since the terminology was coined in 1999 [1], cognitive radio (CR) has been developed for more than fifteen years, which draws attention from both academic and industrial communities since it is intended to enable smart use of the scarce spectrum resource with the initial objective of maximizing spectrum utilization. Recently, cognitive network design goes beyond spectrum utilization and target at broader network objectives such as higher quality of service, lower energy cos, etc. To achieve such new objectives, the statistical knowledge on the primary network status becomes necessary [2] for resource management and system control, which gets us closer to the ideal CR operation that integrates spectrum sensing, environment learning, statistical reasoning, and predictive acting. This will go beyond most of the existing CR sensing literature, which usually focus on detecting the presence of primary users [3]–[5].

In practice, the network behavior is impacted by many factors which may change dynamically, such that the uncertainty of a network cannot be priory represented by a certain statistical distribution. To cope with such an issue, the statistical machine learning methodology becomes a feasible solution for understanding the network activity pattern. In this paper, we introduce the Bayesian network (BN) [6] structure learning method to obtain the statistical primary networking pattern, via observing the on/off status of primary base stations. The Bayesian model has been well known in the field of artificial intelligence (AI) [7]. When considering the probability and uncertainty, BN is a distinct technique for modeling the complex interaction among real world facts [8]. In particular, the BN structure learning is an effective new modeling tool in both spatial and temporal domains. However, the associated computational complexity is high since it needs to evaluate the dependence between each pair of variables in a target system and compute the corresponding conditional probability table1 [7], which becomes the major drawback in applying BN structure learning. In this paper, we focus on reducing the computational load for efficiently learning the statistical behavior pattern of primary base users.

For BN structure learning, the related algorithms could be sorted into two categories. One is to use heuristic searching to construct a probable model and then evaluate it by a scoring function [10]. The structure with the highest score is preferred as the learning outcome. The score-based approach of learning BNs has been proven as a NP-hard problem [11]. The other one is to use conditional independence test to measure the every possible dependence relationship one by one, and then determine the structure based on the evaluated dependence. In [12], the authors show a much more efficient approach to learn the ordered BN by using the mutual information to check the dependence of any possible pair of nodes. However, this approach cannot be adaptively adjusted when the number of variable changes. To overcome the drawbacks of the current learning methods, we propose a structure learning algorithm based on a completely connected graph and the conditional mutual information, which could efficiently learn both the structure and the corresponding conditional probability table. In particular, each pair of variables in the BN has a generalized relationship where the independence case is unified as the weakest dependence case. As a result, the BN network structure becomes very regular, such that the conditional mutual information based learning could be formulated as a sequence of closed-form function evaluations. With this, our learning algorithm not only has the same computational

1 In statistics, the conditional probability table is defined for a set of discrete random variables to quantify the marginal probability of a single variable with respect to the others [9].
and an exponential service law, respectively. Hence, the on/off of primary data traffic at each user follow a Poisson process. Access (MAC) scheme is adopted, and the arrival and departure are located geographically; 3) an ideal TDMA-based multiple access a primary base station depending on which cell they the cells); 2) the primary users share a single channel and multiple secondary sensors over an observation area, of which the detailed specifications are given as follows. Additionally, the statistical period of the BN structure, which reflects the temporal pattern, is not known a priori. In essence, such issues belong to the unsupervised classification problem, with a new challenge to consider the missing time period. To address this issue, we propose a blind variable identification algorithm, working with the proposed structure learning algorithm, to learn the period via the fast Fourier transformation (FFT). In conclusion, the proposed structure learning and period learning algorithms constitute a complete BN structure learning scheme, by which CR could understand the statistical pattern of primary base stations in both spatial and temporal domains.

The remainder of the paper is organized as follows. In section II, the system model is presented in details. Section III briefly introduces the relationship between the BN method and the wireless communication problem in CR. In section IV, an efficient algorithm is proposed for jointly learning the structure and the corresponding conditional probability table. In Section V, an algorithm is proposed to obtain the statistical period of BN structure. In Section VI, the simulation and learning results are presented to validate the proposed scheme. Finally, we conclude the paper in Section VII.

II. SYSTEM MODEL

Our system model consists of a primary cellular network and multiple secondary sensors over an observation area, of which the detailed specifications are given as follows.

The primary cellular network consists of several primary base stations and multiple mobile primary users. The observation area consists of the cells and a road as illustrated in Fig. 1, where we consider the mobile primary users only moving along a one-way (from right to left) road for our case of study. The arrival of the users at the entrance of the road follows a Poisson process. The arrived mobile users pass along the road at a speed generated by a uniform distribution. Once the mobile users move out the road, they are no longer observed. Additionally, the primary network obeys the following setup: 1) the primary base stations are located based on a pre-designed network deployment plan (i.e., at the centers of the cells); 2) the primary users share a single channel and access a primary base station depending on which cell they are located geographically; 3) an ideal TDMA-based multiple access (MAC) scheme is adopted, and the arrival and departure of primary data traffic at each user follow a Poisson process and an exponential service law, respectively. Hence, the on/off status of a primary base station is determined by the overall data traffic generated from the mobile primary users in its cell (here we only consider the uplink transmissions).

A cognitive secondary sensor is installed very close to each primary base station, which implies that the accuracy of sensing could be assumed perfect, such that no sensing errors are considered in this paper. The secondary sensors sense the on/off status of primary base stations periodically in a synchronous fashion. Let set \( M = \{1, 2, \ldots, M\} \) denote the observed primary base stations, and set \( T = \{1, 2, \ldots, T\} \) denote the sequence of time epoch \( t, t \in T \). In addition, let \( f_{i,t} \) be a variable denoting the state of the \( i \)-th primary base station at time \( t \), and \( f_{i,t} \in \mathcal{O} = \{0, 1\} \) where 0 and 1 represents the off and on status respectively.

III. COGNITIVE BAYESIAN NETWORK

In CR, as we argued before, it is valuable to know the statistical behavior of the primary network. By exploring such statistical behavior, the secondary network could exploit the idle spectrum resource more efficiently and more broadly. In our setup, each secondary sensor could obtain a number of observations (or samples) about the on/off status of the observed primary base station, which is the key element of the primary network. The existing works with respect to CR sensing have been mainly focused on the busy/idle status of a particular spectrum hole by observing a specific primary base station. Contrastingly, our objective is to learn the statistical pattern of the spatial and temporal behavior of multiple networked primary base stations by mining the obtained observations over the whole network across different time epochs. To achieve our objective, BN structure learning is deployed as the key methodology. The BN framework has been known in the field of artificial intelligence and exploited in different expert systems to model complex interactions among causes and consequences. While, the BN structure learning aims to derive and quantify the complex interaction from data.

A. Bayesian Network Approach

With BN, the spatial and temporal interactions of primary base stations are expressed by a directed graph \( G = (V, E) \), where \( V \) is a finite, nonempty set whose elements are called the nodes denoting the variable \( f_{i,t} \), and \( E \) is a set of directed
B. BN Structure Learning in CR

As explained above, for generating a BN and the related graph from observations, we need to determine both the variables (nodes) and the dependence of each pair of variables (edge). In many applications, the number of variables is priorly known. Hence, most of learning results mainly focus on how to learn the edges efficiently. In our CR scenario, since the observations are collected from the deployed sensors, it is easy to identify \( M \) for the range of \( i \) in \( f_{j,t} \). However, due to the randomness of primary user number, user speed, and user traffic, the temporal information about \( T \), which defines the range for \( t \) of \( f_{j,t} \), is not directly known. Thus, the BN learning in CR not only needs to efficiently mine the relationship and interaction among the variables, but also has to identify the temporal scale \( T \) of BN. Since \( T \) is not known, we cannot simply sort the observations into the corresponding nodes. Hence, the unknown time scale \( T \) becomes a critical issue in our CR BN learning. In machine learning language, such an issue belongs to the unsupervised classification problem, which is widely considered difficult.

For edge learning, the traditional learning methods are based on scoring or dependence checking functions. Given a scoring function, the computational complexity of determining the BN structure increases exponentially when the number of variables increases. The score-based approach for learning Bayesian networks has been shown NP-hard [11], which is a key challenge in the learning community. Recently, a relatively efficient way to learn an ordered BN is given in [12] by using the conditional mutual information to check the dependence between any possible pair of nodes. Formally, the conditional mutual information is defined as

\[
I(X; Y|Z) = \sum_{z\in Z} \sum_{y\in Y} \sum_{x\in X} P_X(x, y, z) \times \log \frac{P_{Z}(z)P_{X,Y,Z}(x, y, z)}{P_{X,Z}(x, z)P_{Y,Z}(y, z)},
\]

which is a measure of the mutual dependence between variables \( X \) and \( Y \) given variable \( Z \), where the marginal, joint, and/or conditional probability mass functions are denoted by \( P \) with appropriate subscripts. Evidently, this method demands for multiple nested for-loops for implementation, and the number of for-loops is determined by the number of variables in the BN. Hence, this method is not applicable for the online learning case where the number of variables may be time-varying.

C. Cognitive Bayesian Network Learning

Considering the unknown \( T \) and the high computational complexity issues, we propose a BN model for our CR sensing case, and term it as cognitive BN (CBN). The CBN has four characteristics: 1) it is a first-order BN and its nodes are ordered in the temporal domain; 2) its structure is completely connected which means \( \forall i \in M, \forall t \in T \backslash \{1\} \), there exists an edge between \( f_{i,t} \) and \( f_{i,t-1} \); 3) the observation of variables is binary; 4) it has an unknown operation period. Here, “ordered in the temporal domain” means that for a given \( i \), \( f_{i,t}, f_{i,t+1}, \ldots \) are ordered in \( t, t+1, \ldots \). Characteristics 3 and 4 are two direct outcomes from the system model. Thus we only explain characteristics 1 and 2 below.

In our system model, the user movement and data service behaviors of primary users at the current time epoch \( t \) could be solely determined by the system states in the former time epoch \( t-1 \). In other words, our system model has a first-order Markov property which has been adopted previously. In [13], it is shown that a wireless communication network could be represented by a Markov state transition system, where the observations of sensors are faithful with a BN. Hence, we have characteristic 1 in our model, which leads to huge complexity reduction in checking the multiple ordered nodes. Next, we explain characteristic 2 in detail, which is unique and critical in further reducing the computational burden.

When learning the edges, the conventional approaches usually consider the edges either existing or absent, by analyzing the directed dependence that is based on the empirical probabilities generated from observations, where the edge weight could be considered as a bi-level quantization of the directed correlation [14]. In contrast, this paper considers every edge existing, and quantifies the existence with an analog value to reflect the dependent level between any two nodes. In other words, we consider the independence measure is a special case of the dependence measure (with edge value = 0), which implies that each pair of variables in the CBN has a generalized relationship. Based on such an approach, the CBN has a completely connected structure that is highly regular, as shown in Fig. 2. In the next section, we show that, by exploring the regularity of the CBN structure, both the analog-valued edges and the conditional probability table could be learned efficiently.
IV. EFFICIENT LEARNING IN CBN

In this section, we propose an efficient learning algorithm which could correctly work under arbitrary period $T$. While, the unknown $T$ problem is studied in next section. As explained above, the high computational complexity is a critical issue in BN structure learning. It has been shown [12] that the mutual information check based approach leads to the desired efficiency; but it cannot handle the dynamical number of variables. Here, we proposed an efficient learning algorithm of the same complexity as the $I(X;Y|Z)$ based method, and can cope with the varying number of variables.

When employing the conditional mutual information, every possible edge will be checked in turn. In other words, the number of possible edges affects the learning overhead. Recall that our CBN is a first-order BN and ordered in temporal domain. It means that, in CBN, the direction of edges is known and the edge only connects the adjacent nodes for a given $i$ in $f_{i,t}$ as shown in Fig. 2. Hence, the computational complexity of learning such a CBN is proportional to learning the subgraph enclosed by the bold-line rectangle in Fig. 2. If we do not consider the direction of edges, this subgraph is called as a clique in a Markov network. For the clarity of expression, we here call the targeted subgraph as a C-clique. According to characteristic 1, the number of edges in a CBN is $T$ times as it in a C-clique. Apparently, the computational complexity of learning CBN is linearly proportional to the overhead of learning a C-clique. Hence, in the following study, we focus on how to efficiently learn a C-clique.

A. Learning a C-clique by Conventional Methods

Before introducing our idea, we need to show how the current approaches learn a CBN by using the conditional mutual information check, which is helpful to understand the computational complexity of learning a C-clique with our algorithm. In conventional methods, when learning the edges in a C-clique, we need to determine the existence of a possible edge one by one. For example, to check the edge between $f_{1,t}$ and $f_{1,t+1}$, the corresponding conditional mutual information is given by (2) associated with a realization of $M = 3$ as shown in Fig. 2:

$$I(f_{1,t};f_{1,t+1}|f_{2,t},f_{3,t}) = \sum_{f_{1,t}} \sum_{f_{2,t}} \sum_{f_{3,t}} P(f_{1,t},f_{2,t},f_{3,t},f_{1,t+1}) \log \frac{P(f_{2,t},f_{3,t})P(f_{1,t},f_{2,t},f_{3,t},f_{1,t+1})}{P(f_{1,t},f_{2,t},f_{3,t})P(f_{1,t+1},f_{2,t},f_{3,t})},$$

where each probability is a empirical probability. The computational load of calculating $I(f_{1,t};f_{1,t+1}|f_{2,t},f_{3,t})$ is mainly determined by using the related observations to calculate both the empirical probability $P(f_{1,t},f_{2,t},f_{3,t},f_{1,t+1})$ and the structure of $I(f_{1,t};f_{1,t+1}|f_{2,t},f_{3,t})$ function itself. Apparently, the related computation requires all the realizations of $\{f_{1,t},f_{2,t},f_{3,t},f_{1,t+1}\}$, e.g. $\{f_{1,t} = 0, f_{2,t} = 0, f_{3,t} = 0, f_{1,t+1} = 0\}, \{f_{1,t} = 0, f_{2,t} = 0, f_{3,t} = 1, f_{1,t+1} = 1\}, \ldots$. As a result, the computational complexity of measuring an edge depends on computing $2^{3(T+1)}$ times of $P(f_{1,t},f_{2,t},f_{3,t},f_{1,t+1})\log \frac{P(f_{2,t},f_{3,t})P(f_{1,t},f_{2,t},f_{3,t},f_{1,t+1})}{P(f_{1,t},f_{2,t},f_{3,t})P(f_{1,t+1},f_{2,t},f_{3,t})}$.

In general, since there are $M^2$ edges in a C-clique, the computational complexity of learning a C-clique is determined by running $M^22^{3(T+1)}$ times of $P(f_{1,t},f_{2,t},f_{3,t},f_{1,t+1})\log \frac{P(f_{2,t},f_{3,t})P(f_{1,t},f_{2,t},f_{3,t},f_{1,t+1})}{P(f_{1,t},f_{2,t},f_{3,t})P(f_{1,t+1},f_{2,t},f_{3,t})}$.

Such outcome implies that the computational complexity of learning a C-clique could be reduced by improving the. On the other hand, each term in (2) is actually based on the conditional probability table considering that the C-clique is completely connected. Note, when each term in (2) is estimated by the empirical probabilities, the computational complexity of (2) approximates that of computing the conditional probability table. Evidently, if we could find an approach to learn the complete conditional probability table and then quantify each edge by close-form expressions, the issue of varying variables will be solved. Based on above two ideas, we next propose an efficient algorithm based on analyzing the completely connected structure and exploring the fact of binary observations.

B. Efficient Learning Algorithm for Conditional Probability Table

As explained before, the efficiency of the mutual information based CBN learning is determined by two factors: the measurement method of dependence and the adaptation to the number of variables. Among the two factors, the latter one plays a leading role. In this subsection, we show our effort to handle the second factor by utilizing characteristic 2 of our CBN.

1) Completely Connected Structure and Its Benefit:

In a completely connected BN, we need to compute $P(f_{i,t}|pa(f_{i,t}))$, $\forall i \in M$ and $\forall t \in T$, to obtain the conditional probability table\footnote{For the clarity of expression, the empirical probability and the true probability are expressed by the same notation system.}. Intuitively, the completely connected BN is of high computational complexity because of the large number of edges. However, the fact is just the opposite. Since the structure of a completely connected BN is perfectly symmetry, the algorithm of computing the conditional probability table could be designed efficiently, which will be discussed next.

For a variable $f_{i,t}$ with $\forall i \in M$ and $\forall t \in T$, consider its related observations contained in a column vector $\alpha_{i,t}$. For brevity, we take $M = 2$ along with binary observations to explain our idea, and then extend the related results to a general case with $M \geq 2$. From the perspective of the frequentist probability, the empirical conditional probability table of $P(f_{i,t}|pa(f_{i,t}))$ in a C-clique is given by

$$P(f_{i,t} = 1|f_{1,t},f_{2,t} = 1, f_{2,t-1} = 1) = \frac{\alpha_{1,t-1}^T \circ \alpha_{2,t-1}^T \circ \alpha_{1,t}}{\alpha_{1,t-1}^T \circ \alpha_{2,t-1}^T},$$

\footnote{When implementing this calculation in code, it needs $2^{3(T+1)}$-for loops.} where

\footnote{In $P(f_{i,t}|pa(f_{i,t}))$, $pa(f_{i,t})$ means $f_{i,t}$’s parents which are the nodes connecting to $f_{i,t}$ directly.}
where $\odot$ is the Hadamard product,

$$P(f_{1,t} = 1|f_{1,t-1} = 0, f_{2,t-1} = 0) = \frac{\hat{o}_{1,t-1}^T \odot \hat{o}_{2,t-1}^T}{\hat{o}_{1,t-1}^T \odot \hat{o}_{2,t-1}} o_{1,t},$$  \hspace{1cm} (4)

where $\hat{o}_{1,t-1}^T = 1 - o_{1,t-1}^T$,

$$P(f_{2,t} = 1|f_{1,t-1} = 1, f_{2,t-1} = 0) = \frac{\hat{o}_{1,t-1}^T \odot \hat{o}_{2,t-1}^T}{\hat{o}_{1,t-1}^T \odot \hat{o}_{2,t-1}} o_{2,t},$$  \hspace{1cm} (5)

and

$$P(f_{2,t} = 1|f_{1,t-1} = 1, f_{2,t-1} = 0) = \frac{\hat{o}_{1,t-1}^T \odot \hat{o}_{2,t-1}^T}{\hat{o}_{1,t-1}^T \odot \hat{o}_{2,t-1}} o_{2,t}.\hspace{1cm} (6)$$

It is easy to see that the empirically conditional probability could be calculated as multiplying the observation $o_{m,t}$ by a regular arithmetic operator denoted by $F$. We add a subscript $c$ to $F$ to express the condition, e.g., $F_{00}$ with $c = 00$ stands for the arithmetic operator under $f_{1,t-1} = 0$ and $f_{2,t-1} = 0$, such that we have $F_{00} = \hat{o}_{1,t-1}^T \odot \hat{o}_{2,t-1}$.

From (3)-(6) and (7)-(49), we see that the computation of the conditional probability table is transformed to obtaining the arithmetic operator $F_c$ with every realization of $c$, where the structure of $F_c$ is very regular, which benefits from characteristic 2 of our CBN. Based on the regularity of $F_c$, the edges no longer need to be learned one by one, which is explained as follows.

2) Binary Observation based Conditional Probability Table (BbCPT) Learning Algorithm: In CR, the on/off behavior of a primary base station is expressed by a binary value, which leads to our learning algorithm exploring this fact.

Given binary observations, the number of realizations of $c$ is $2^M$, based on which we define $x \in \{0, 1, \ldots, 2^M - 1\}$. Accordingly, for a given $x$, we could use (11) defined below to generate a corresponding vector $c_x$ leading to $c = c_x[1] \times c_x[2] \times \cdots \times c_x[M]$, with

$$c_x[i] = \left\lfloor \frac{x}{2^{i-1}} \right\rfloor \pmod{2}$$ \hspace{1cm} (11)

where $\pmod{}$ denotes the modulo operation and $i \in \{1, 2, \ldots, M\}$.

For example, when $M = 2$ and $x = 2$, $c_2 = [1 \ 0]$, which means $c = 10$. Further, we could generate a matrix $C$ where each row contains a realization of $c$ given $x$:

$$C[x, M - i + 1] = \left\lfloor \frac{x}{2^{i-1}} \right\rfloor \pmod{2}, \quad i \in M.\hspace{1cm} (12)$$

On the other hand, when the observation is binary and we take $M = 2$ as in the previous subsection, the numerators of $F_{11}, F_{10}, F_{01},$ and $F_{00}$ could be reformulated as

$$o_{1,t-1}^T \odot o_{2,t-1}^T = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$c_0[01, o_{1,t-1}, o_{2,t-1}] = \begin{bmatrix} 1 - c_1 \left( 1 - \left[ o_{1,t-1}^T, o_{2,t-1}^T \right] \right) \\ c_1 \left( 1 - \left[ o_{1,t-1}^T, o_{2,t-1}^T \right] \right) \end{bmatrix}$$

$$c_0[10, o_{1,t-1}, o_{2,t-1}] = \begin{bmatrix} 1 - c_2 \left( 1 - \left[ o_{1,t-1}^T, o_{2,t-1}^T \right] \right) \\ c_2 \left( 1 - \left[ o_{1,t-1}^T, o_{2,t-1}^T \right] \right) \end{bmatrix}$$

respectively. Let nom($x$) and dnom($x$) respectively denote the numerator and denominator of $x$, and $F$ be a matrix containing all $F_c$‘s. Then, we arrive at a simple form for each realization of $F_c$ as below.

$$\text{nom}(F) = \text{nom}(F_{00}) ; \text{nom}(F_{01}) ; \text{nom}(F_{10}) ; \text{nom}(F_{11})$$

$$= \frac{CO_{t-1} + (1 - C) (1 - O_{t-1})}{2}$$ \hspace{1cm} (17)

where

$$O_{t-1} = [o_{1,t-1}, o_{2,t-1}]^T,$$ \hspace{1cm} (18)
with \( C \) obtained from (12).

On the other hand, since \( \text{nom}(\mathcal{F}_e) \) is a matrix, we have \( \text{dnom}(\mathcal{F}_e) = \text{nom}(\mathcal{F}_e)1_N \), where \( 1_N \) is a column vector of \( N \) ones. Thus, the denominator of \( \mathcal{F} \) is given by

\[
\text{dnom}(\mathcal{F}) = \text{nom}(\mathcal{F})1_N.
\] (20)

Extending (17) to an arbitrary value of \( M \) yields a general arithmetic operator \( \mathcal{F} \),

\[
\mathcal{F} = \text{nom}(\mathcal{F})/(\text{nom}(\mathcal{F})1_N \otimes 1_N^T),
\] (21)

where \( \otimes \) denotes the Kronecker product, and

\[
\text{nom}(\mathcal{F}) = \left[ CO_{t-1} + (1 - C)(1 - O_{t-1}) \right]/M,
\] (22)

\[
O_{t-1} = [o_{1,t-1}, o_{2,t-1}, \ldots, o_{M,t-1}]^T.
\] (23)

Hence, the empirical conditional probability table could be calculated as

\[
B_{t-1} = [B_{1,t-1}, \ldots, B_{M,t-1}] = \mathcal{F}O_{t-1}^T,
\] (24)

where \( B_{i,t-1} \) is a column vector with size \( 2^M \) containing every probability that \( f_{i,t} = 1 \) holds conditioned on \( c \).

Consequently, the BbCPT learning algorithm is given by (12) and (21)-(24). When compared with conventional methods, the proposed BbCPT algorithm effectively reduces the number of for-loops to just two for-loops (with matrix computation), and the empirical conditional probability table could be obtained by a sequence of closed-form function evaluations efficiently.

At the beginning of this subsection, we stated that the total computational overhead is affected by two factors, with the latter one already discussed in this subsection. Next, we study the first one to show how to efficiently measure the dependence between any pair of two variables in a C-clique.

C. Efficient CBN Structure Learning Algorithm

For the overall efficient CBN structure learning algorithm, we will also derive our metrics based on the conditional mutual information as discussed before, which further integrates the previous results on C-clique and conditional probability table learning.

Given a discrete random variable \( X \) with support \( \mathcal{X} \) and \( Y \) with support \( \mathcal{Y} \), the conditional entropy between \( X \) and \( Y \) is given by

\[
H(Y|X) = \sum_{x \in \mathcal{X}} P(x)H(Y|X = x) = \sum_{x \in \mathcal{X}} P(x)\sum_{y \in \mathcal{Y}} P(y|x)\log P(y|x) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(x,y)\log \frac{P(x)}{P(x,y)}
\] (25)

If \( Y \) is completely determined by \( X \), we have \( H(Y|X) = 0 \); if \( Y \) is independent of \( X \), we have \( H(Y|X) = H(Y) \). Hence, \( H(Y|X) \in [0, H(Y)] \) reflects the quantification of dependence of \( Y \) on \( X \). According to information theory, \( H(Y|X) + H(X|Y) \) is equivalent to \( I(X;Y) \) when measuring the dependence between \( X \) and \( Y \).

For the CR case where \( \mathcal{X} = \emptyset \) and \( \mathcal{Y} = \emptyset \), the conditional entropy is given by

\[
H(Y|X) = -\sum_{x \in \mathcal{X}} P(x)P(y=1|x)\log P(y=1|x) - \sum_{x \in \mathcal{X}} P(x)P(y=0|x)\log P(y=0|x)
\] (26)

which is a unimodal and symmetric function with respect to \( P(y=1|x = 1) \) and \( P(y=1|x = 0) \). No matter what the value of \( P(x) \) is, the maximum value of (26) is achieved when the following equations hold:

\[
\log P(y=0|x = 0) - \log P(y=0|x = 1) = 0,
\] (27)

\[
\log P(y=1|x = 0) - \log P(y=1|x = 1) = 0.
\] (28)

Evidently, given \( P(x) \), larger values of

\[
\log P(y=0|x = 0) - \log P(y=0|x = 1)\] and
\[
\log P(y=1|x = 0) - \log P(y=1|x = 1)
\] lead to smaller values of \( H(Y|X) \), which implies higher dependence between \( y \) and \( x \). Similarly, we could conclude the same observations about \( H(X|Y) \). Therefore, we arrive at the following expression, where \( D_p(y;x) \) is termed as the conditional probability based distance (Cpbd),

\[
D_p(y;x) = \sum_{y \in \mathcal{Y}} |\log P(y|x = 0) - \log P(y|x = 1)|
\] (29)

which is also symmetrical with respect to \( P(y=1|x = 1) \) and \( P(y=1|x = 0) \). The proposed Cpbd \( D_p(y;x) \) is not affected by \( P(y) \) and \( P(x) \), such that the value range of \( D_p(y;x) \) is \([0, +\infty)\). For multiple-variable cases, the Cpbd between \( x \) and \( y \) conditioned on \( z \) is given by

\[
D_p(y;z|x) = \sum_{y \in \mathcal{Y}} |\log P(y|z, x = 0) - \log P(y|z, x = 1)|
\] (30)

Taking the example of \( M = 2 \) in CBN, the Cpbd between \( f_{1,t-1} \) and \( f_{1,t} \) under \( f_{2,t-1} \) is given by

\[
D_p(f_{1,t}, f_{1,t-1}, f_{2,t-1}) = |\log P(f_{1,t} = 1|f_{1,t-1} = 1, f_{2,t-1} = 1)|
- |\log P(f_{1,t} = 1|f_{1,t-1} = 0, f_{2,t-1} = 1)|
+ |\log P(f_{1,t} = 1|f_{1,t-1} = 1, f_{2,t-1} = 0)|
- |\log P(f_{1,t} = 1|f_{1,t-1} = 0, f_{2,t-1} = 0)|
\] (31)

In probability theory and information theory, \( H(Y|X) + H(X|Y) \) is a distance measure between variables for evaluating the variation of information or shared information distance.
where each term inside the log operation could be directly obtained from the conditional probability table derived in Section IV-B. Moreover, each term has a regular relative index corresponding to the conditional probability table. As shown in the previous subsection, the conditional probability table with respect to \( f_{1,t-1} \) is given as 
\[
B_{1,t-1} = \begin{bmatrix} B_{00}, B_{01}, B_{10}, B_{11} \end{bmatrix}^T.
\]
Let \( B_{1,t-1} = 1 - B_{1,t-1} \); we have \( \log B_{1,t-1} = [\log B_{00}, \log B_{01}, \log B_{10}, \log B_{11}]^T \) and \( \log B_{1,t-1} = \log(1 - B_{1,t-1}) \). Then \( D_p(f_{1,t}; f_{1,t-1}|f_{2,t-1}) \) and \( D_p(f_{1,t}; f_{2,t-1}|f_{1,t-1}) \) can be calculated as 
\[
D_p(f_{1,t}; f_{1,t-1}|f_{2,t-1}) = (\log B^T_{t,t-1}L_1) \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix},
\]
(32)
where
\[
L_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix},
\]
(34)
(33)

For every possible edge pointing to \( f_{1,t} \), the corresponding CPbD is given by 
\[
D_p(f_{1,t}; f_{1,t-1}|f_{2,t-1}) = (\log B^T_{t,t-1}L_1) \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix},
\]
(36)
where
\[
L = \begin{bmatrix} L_1, L_2 \end{bmatrix},
\]
(37)
\[
S = I \otimes 1_2,
\]
(38)

Similarly as the above, it is easy to check that \( L \) and \( S \) obtained for \( f_{1,t} \) is still suitable for \( f_{2,t} \). Thus, when \( M = 2 \), the CPbD of the C-clique at \( t - 1 \) is given by 
\[
D_{t-1} = (\log B^T_{t-1}L) \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix},
\]
(39)
where 
\[
D_{t-1} = \begin{bmatrix} D_p(f_{1,t}; f_{1,t-1}|f_{2,t-1}) & D_p(f_{1,t}; f_{2,t-1}|f_{1,t-1}) \\ D_p(f_{2,t}; f_{1,t-1}|f_{2,t-1}) & D_p(f_{2,t}; f_{2,t-1}|f_{1,t-1}) \end{bmatrix},
\]
(40)
\[
B_{t-1} = [B_{1,t-1}, B_{2,t-1}],
\]
(41)
\[
L = \begin{bmatrix} L_1, L_2 \end{bmatrix},
\]
(42)
\[
S = I \otimes 1_2.
\]
(43)

Dp \((f_{1,t}; f_{1,t-1}|f_{M,t-1}) \geq Dp \((f_{1,t}; f_{i,t-1}|f_{M,m,t-1}) \), \( \forall i, m \in \mathbb{M} \) and \( m \neq i \). Hence, the entries in \( D_{t-1} \) could be normalized as 
\[
D_{t-1} = \Lambda(D_{t-1})^{-1}D_{t-1},
\]
(44)
where
\[
D_{t-1} = (\log B^T_{t-1}L) \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix},
\]
(46)
\[
B_{t-1} = [B_{1,t-1}, B_{2,t-1}, \cdots, B_{M,t-1}],
\]
(47)
\[
S = I \otimes 1_M.
\]
(48)

It is worth emphasizing that matrix \( L \) is not dependent on the number of C-cliques, but only on the number of variables in a C-clique. Hence, matrix \( L \) only needs to compute once for a given \( M \). It could be computed offline before the online CBN learning. This paper does not study the optimal approach for obtaining it. Alternatively, we provide a feasible method to arrive at \( L \), given in Table I.

**TABLE I. ALGORITHM OF OBTAINING L.**

| 1) | \( \text{for } m = 1 : 1 : M \) |
| 2) | \( \epsilon_1 = 0_2 \times \epsilon_1[2^m, 1] = -1, \epsilon_1[2^m/2, 1] = 1 \) |
| 3) | generate a circulant matrix \( e \) based on \( \epsilon_1 \), each row is a backward shifting of \( \epsilon_1 \), |
| 4) | \( \epsilon_1 = e[1: m] \) |
| 5) | for \( n = 2 : 1 : m \) |
| 6) | \( \epsilon_l = I_{2 \times 2} \otimes L_{n-1}, L_{n+1} = \epsilon_{l-1} \) |
| 7) | end |
| 8) | end |
| 9) | \( L = [L_1, L_2, \cdots, L_M] \) |

**D. Efficient C-clique Learning Procedure**

Based on the previously discussed BbCPT and CPbD-based structure learning, the procedure of learning a C-clique is summarized as:

1) Initialization of \( C, L, \) and \( S \):
2) Obtain the conditional probability table by (21)-(24);
3) Obtain the CPbD of all edges by (45)-(48).

As we have emphasized before, \( C \) and \( L \) are only required to calculate once according to the number of variables in a C-clique, which means that they could be calculated and stored priorly. When regarding \( C \) and \( L \) as prior knowledge, the above procedures show that our learning algorithm is just a sequence of closed-form function evaluations and has a simpler dependence check function. Consequently, the proposed learning method has lower computational complexity and is capable of adapting to different numbers of variables.
V. LEARNING CBN WITH UNKNOWN T

The previous section introduces the CBN learning algorithm with a given T. Sometimes, the value of T is not known a priori. For example, we do not know the exact value of T in our CR system model. The uncertainty of T implies that the number of variables in the temporal domain is unknown in the CBN model. Consequently, after a number of observations being collected, we do not exactly know which temporal variables that generated the collected observations. In the structure learning literature, observations are usually assumed already associated with different variables correctly. As such, the uncertainty of T is a challenging issue, less studied in the BN community. For this problem, we propose a heuristic but efficient solution as follows.

To estimate T, we first convert this uncertainty issue into a mathematical problem. Let \( D_p(x) = \sum_{i,k} D_p(f_{i,t+x}; f_{k,t}; f_{1:M,t}(f_{k,t})) \), which is the sum of the dependence measures over every edge between the variables at times t and t + x. In addition, let \( T_p, T_s, \) and \( T^* \) denote the period of CPbD, the interval between two sequential observations of the same variable, and the minimal value of period T, respectively. Note that all the periods in set \( \{T | T = kT^*, k \in \mathbb{Z} \} \) are feasible for the BN structure (here \( \mathbb{Z} \) denotes the positive integer set). But a longer period implies higher computational complexity. Hence, we pick the minimal possible value \( T^* \) as the optimal choice. For \( T^* \), it not only needs to ensure that the observations are statistically periodic along temporal variables but also has to guarantee that the observations of a variable are statistically independent. Accordingly, when \( N \to \infty \), the optimal period \( T^* \) could be obtained by solving

\[
T^* = \arg \min_{t} \frac{t}{kT_p}, \quad k \in \mathbb{Z}, \quad t \geq T^* - 1,
\]

where \( T_s = \min\{T_s | D_p(T_s) = 0\} \).

A. Learning \( T_s \)

Specifically, the period T should ensure that the observations of any variable \( f_{i,t} \) are statistically independent over \( T_s \). Regarding our system, the temporal correlation of two observations decreases as the their interval increases. Obviously, it is better to set an enough long period \( T_s \) to ensure the statistically independent condition. However, the number of available observations \( N \) is limited in practice, so that we can only obtain the empirical \( T_{p} \) and \( T_{s} \), and it is impossible to find a feasible value of \( T_s \) to ensure the perfect condition \( D_p(T_s) = 0 \). It is worth noting that: i) it is impossible to deduce the true probability or distribution of a variable from limited observations, and ii) the empirical statistical information is widely used in real system analysis. From the perspective of both statistics and engineering, the empirical statistical information is really valuable and useful as long as it is close to the true one. Hence, given the finite observations, we propose to obtain the empirical \( T_s \) by finding the first valley value of the CPbD between \( f_{1:M,t} \) and \( f_{1:M,t+T_s} \), denoted by \( D_{t,t+T_s} \). Here \( f_{1:M,t} \) is short for \( f_{1,t}, \ldots, f_{M,t} \). The pseudo-code of the proposed algorithm denoted by \( T_s \) is given in Table II, where \( \frac{|D_p|}{s} \) is the averaged \( D_{t,t+T_s} \) over \( T_s \) and edges.

### Table II. Pseudo-code of \( T_s \)

| Step | Description |
|------|-------------|
| 1)   | set \( l = 2 \) |
|      | for \( T_s = 2 : 1 : 2^l + 1 \) |
|      | if \( t = 1 : 1 : T_{p} \) |
|      | \( D_p[t] = D_{t,t+T_{s}} \); |
|      | end |
|      | \( D[T_s - 1] = \frac{|D_p|}{s} \); |
|      | for \( t = 2 : 1 : T_s - 1 \) |
|      | if \( D_p[t] \leq D_p[t-1] \) and \( D_p[t] \leq D_p[t+1] \) |
|      | then break and output \( T^* = t \); |
|      | end |
|      | if \( T_s = 2^l \) |
|      | then set \( l = l + 1 \) and goto 2). |

Next, we give our algorithm which obtains \( T_p \) by using the regularity of CPbD.

B. Learning \( T_p \)

As proven above, the CPbD has capability of reflecting the dependence between variables. Hence, when finding the correct value of \( T_p \), the CPbD should demonstrate a certain regularity which could help to learn \( T_p \). This subsection first shows the existence of CPbD’s regularity mathematically. We need to emphasize that the observations are assumed to have certain correlation in user and temporal domain, though we do not know the statistical characteristics of the correlation, since if all observations are independent there will be no edges in the BN structure. To ensure the unknown \( T_p \) could be obtained by using the observations, we have the following theoretical studies.

First, we derive Proposition 1 and Proposition 2 to show that the calculation of the empirical \( P(f_{i,t+1} = 1 | f_{i,t} = 1) \) has regular outcomes when considering \( T \) is variable. Let \( T_{f} \) and \( T_{r} \) denote the sets containing the false and true periods respectively. Obviously, there is \( T := T_{f} \cup T_{r} \). And let \( Z_{c} := S(k : k \) is positive integer) and \( Z_{0} := \{2k + 1 : k \) is positive integer).\n
Proposition 1: When \( T_{f}, T_{f}, 1 \in T_{f}, T_{r} \in T_{r}, \) and \( T_{r} \in Z_{0}, \) there exist:

P1.1) \( P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{r}} \neq P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \)

P1.2) \( P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \neq P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \)

Proposition 2: When \( T_{f}, T_{f}, 1 \in T_{f}, T_{r} \in T_{r}, \) and \( T_{r} \in Z_{c}, \) there exist:

P2.1) \( P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{r}} \neq P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \)

P2.2) \( P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \neq P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \) under the condition that \( T_{f}, T_{f}, 1 \in Z_{c} \).

P2.3) \( P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \neq P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \) under the condition that \( T_{f}, T_{f}, 1 \in Z_{c} \).

P2.4) \( P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \neq P(f_{i,t+1} = 1 | f_{i,t} = 1) | T_{T-T_{f}} \) under the condition that \( T_{f} \in Z_{0} \) and \( T_{f}, 1 \in Z_{c} \).
Proof: We prove Proposition 1 at first. Let \( \{o_{1,1}, o_{1,2}, \ldots, o_{1,N}\} \) denote the N observations of user-1. Then, given the period \( T \), the observations are sorted by as follows.

- When \( T = 2 \), we have \( o_{1,1}[T-2] = [o_{1,1,1}, o_{1,3,1}, \ldots]^T \) and \( o_{1,2}[T-2] = [o_{1,1,2}, o_{1,4,1}, o_{1,6,1}, \ldots]^T \).
- When \( T = 3 \), we have \( o_{1,1}[T-3] = [o_{1,1,1}, o_{1,4,1}, o_{1,7,1}, \ldots]^T \), \( o_{1,2}[T-3] = [o_{1,2,1}, o_{1,5,1}, o_{1,8,1}, \ldots]^T \), and \( o_{1,3}[T-3] = [o_{1,3,1}, o_{1,6,1}, o_{1,9,1}, \ldots]^T \).
- When \( T = 4 \), we have \( o_{1,1}[T-4] = [o_{1,1,1}, o_{1,5,1}, o_{1,9,1}, \ldots]^T \), \( o_{1,2}[T-4] = [o_{1,2,1}, o_{1,6,1}, o_{1,10,1}, \ldots]^T \), \( o_{1,3}[T-4] = [o_{1,3,1}, o_{1,7,1}, o_{1,11,1}, \ldots]^T \), and \( o_{1,4}[T-4] = [o_{1,4,1}, o_{1,8,1}, o_{1,12,1}, \ldots]^T \).
- When \( T = 5 \), there exist \( o_{1,1}[T-5] = [o_{1,1,1}, o_{1,6,1}, o_{1,11,1}, \ldots]^T \), \( o_{1,2}[T-5] = [o_{1,2,1}, o_{1,7,1}, o_{1,12,1}, \ldots]^T \), \( o_{1,3}[T-5] = [o_{1,3,1}, o_{1,8,1}, o_{1,13,1}, \ldots]^T \), \( o_{1,4}[T-5] = [o_{1,4,1}, o_{1,9,1}, o_{1,14,1}, \ldots]^T \), and \( o_{1,5}[T-5] = [o_{1,5,1}, o_{1,10,1}, o_{1,15,1}, \ldots]^T \).
- When \( T = 6 \), there exist \( o_{1,1}[T-6] = [o_{1,1,1}, o_{1,7,1}, o_{1,13,1}, \ldots]^T \), \( o_{1,2}[T-6] = [o_{1,2,1}, o_{1,8,1}, o_{1,14,1}, \ldots]^T \), \( o_{1,3}[T-6] = [o_{1,3,1}, o_{1,9,1}, o_{1,15,1}, \ldots]^T \), \( o_{1,4}[T-6] = [o_{1,4,1}, o_{1,10,1}, o_{1,16,1}, \ldots]^T \), \( o_{1,5}[T-6] = [o_{1,5,1}, o_{1,11,1}, o_{1,17,1}, \ldots]^T \), and \( o_{1,6}[T-6] = [o_{1,6,1}, o_{1,12,1}, o_{1,18,1}, \ldots]^T \).

Accordingly, when \( T_r = 5 \), we have

\[
P(f_{1,2} = 1 | f_{1,1} = 1, T_T = 2, T_r = 5) = \lim_\text{N} \sum_{t=1}^{T_f} o_{1,1} T_{1,1} + o_{1,5} T_{1,5} S(o_{1,1}) \mid_{T = T_T = 5}
\]

where \( S(.) \) denotes an operation shifting the entries of a vector with a single step circularly.

\[
P(f_{1,2} = 1 | f_{1,1} = 1, T_T = 3, T_r = 5) = \lim_\text{N} \sum_{t=1}^{T_f} o_{1,2} T_{1,2} \mid_{T = T_T = 3}
\]

To see that \( P(f_{1,2} = 1 | f_{1,1} = 2, T_T = 1, T_r = 5) = P(f_{1,2} = 1 | f_{1,1} = 1, T_T = 3, T_r = 5) = P(f_{1,2} = 1 | f_{1,1} = 1, T_T = 4, T_r = 5) = P(f_{1,2} = 1 | f_{1,1} = 1, T_T = 5) \) by a similar approach, we show that such a relationship holds for general \( T_f \) and \( T_r \) (where \( T_r \in \mathbb{Z}_0 \)) as follows.

- When \( T = T_f \), there exist \( o_{1,1}[T-T_f] = [o_{1,1,1}, o_{1,3,1}, o_{1,5,1}, \ldots]^T \), \( o_{1,2}[T-T_f] = [o_{1,2,1}, o_{1,4,1}, o_{1,6,1}, \ldots]^T \), \( o_{1,3}[T-T_f] = [o_{1,3,1}, o_{1,5,1}, o_{1,7,1}, \ldots]^T \), \( o_{1,4}[T-T_f] = [o_{1,4,1}, o_{1,6,1}, o_{1,8,1}, \ldots]^T \), \( o_{1,5}[T-T_f] = [o_{1,5,1}, o_{1,7,1}, o_{1,9,1}, \ldots]^T \), and \( o_{1,6}[T-T_f] = [o_{1,6,1}, o_{1,8,1}, o_{1,10,1}, \ldots]^T \).
- When \( T = T_r \), there exist \( o_{1,1}[T-T_r] = [o_{1,1,1}, o_{1,3,1}, o_{1,5,1}, \ldots]^T \), \( o_{1,2}[T-T_r] = [o_{1,2,1}, o_{1,4,1}, o_{1,6,1}, \ldots]^T \), \( o_{1,3}[T-T_r] = [o_{1,3,1}, o_{1,5,1}, o_{1,7,1}, \ldots]^T \), \( o_{1,4}[T-T_r] = [o_{1,4,1}, o_{1,6,1}, o_{1,8,1}, \ldots]^T \), \( o_{1,5}[T-T_r] = [o_{1,5,1}, o_{1,7,1}, o_{1,9,1}, \ldots]^T \), and \( o_{1,6}[T-T_r] = [o_{1,6,1}, o_{1,8,1}, o_{1,10,1}, \ldots]^T \).

Accordingly, we have

\[
o_{1,1}[T][o_{1,2} T_{1,1} + o_{1,5} T_{1,5} S(o_{1,1})]
\]

where \( n \) and \( l \) are index, \( t_r \) and \( t_f \) respectively denote \( t_{T-T_f} \) and \( t_{T-T_r} \), and

\[
t_r = \left[ t_f + (n-1) T_f \right] T_r + 1.
\]

Because of \( T_r \in \mathbb{Z}_0 \), the modulo operation in (54) leads to \( t_r \in \{0, 1, 2, \ldots, T_r - 1\} \) when \( n \) increasing. Hence, we could obtain

\[
P(f_{1,t+1} = 1 | f_{1,t} = 1, T_f, T_r) = \lim_\text{N} \sum_{t=1}^{T_f} o_{1,1} T_{1,1} \mid_{T = T_f}
\]

and then could extend it to the following result,

\[
P(f_{1,t+1} = 1 | f_{1,t} = 1, T_f, T_r) = \lim_\text{N} \sum_{t=1}^{T_f} o_{1,1} T_{1,1} \mid_{T = T_f}
\]

Hence, Proposition 1 is proved. Next, we prove Proposition 2.

When \( T_r = 6 \), we have

\[
P(f_{1,2} = 1 | f_{1,1} = 1, T_T = 2, T_r = 6)
\]

\[
= \lim_\text{N} \sum_{t=1}^{T_f} o_{1,1} T_{1,1} \mid_{T = T_T = 6}
\]
\[ P(f_{1,2} = 1 | f_{1,1} = 1, T_f = 3, T_r = 6) = \lim_{N \to \infty} \frac{\sum_{i=1}^{N} \sigma_i^T \sigma_i^{T+1} + \sigma_i^T \sigma_i^{T+1}}{(\sum_{i=1}^{N} \sigma_i^T)} 1^{T \to T_r = 6}, \]

\[ = \lim_{N \to \infty} \frac{\sigma_{i,t+1}^T \sigma_{i,t+1}^{T+1} + \sigma_{i,t}^T \sigma_{i,t}^{T+1}}{(\sum_{i=1}^{N} \sigma_i^T)} 1^{T \to T_r = 6}, \]

\[ P(f_{1,2} = 1 | f_{1,1} = 1, T_f = 4, T_r = 6) = \lim_{N \to \infty} \frac{\sigma_{i,t+1}^T \sigma_{i,t+1}^{T+1} + \sigma_{i,t}^T \sigma_{i,t}^{T+1}}{(\sum_{i=1}^{N} \sigma_i^T)} 1^{T \to T_r = 6}, \]

\[ = \lim_{N \to \infty} \frac{\sigma_{i,t+1}^T \sigma_{i,t+1}^{T+1} + \sigma_{i,t}^T \sigma_{i,t}^{T+1}}{(\sum_{i=1}^{N} \sigma_i^T)} 1^{T \to T_r = 6}, \]

\[ \text{From (58)-(61), it is obvious that } P(f_{1,2} = 1 | f_{1,1} = 1, T_f = 2, T_r = 6) = P(f_{1,2} = 1 | f_{1,1} = 1, T_f = 4, T_r = 6) = P(f_{1,2} = 1 | f_{1,1} = 1, T_f = 3, T_r = 6) = P(f_{1,2} = 1 | f_{1,1} = 1, T_f = 5, T_r = 6) = P(f_{1,2} = 1 | f_{1,1} = 1, T_f = 6). \]

By a similar approach, we show that such a relationship holds for general \(T_f\) and \(T_r\) (where \(T_r \in \mathbb{Z}_c\)) as follows.

When \(T_r \in \mathbb{Z}_c\), \(T_f \in \mathbb{Z}_d\) and \(T_f \in \mathbb{Z}_e\), the modulo operation in (54) leads to \(t_r \in \{1, 3, \ldots, T_r - 1\}\); when \(T_r \in \mathbb{Z}_c\), \(T_f \in \mathbb{Z}_d\) and \(T_f \in \mathbb{Z}_e\), the modulo operation in (54) leads to \(t_r \in \{2, 4, \ldots, T_r\}\). Hence, we have

\[ P(f_{1,t+1} = 1 | f_{1,t} = 1, T_f, T_r) = \lim_{N \to \infty} \frac{\sigma_{i,t+1}^T \sigma_{i,t+1}^{T+1}}{(\sum_{i=1}^{N} \sigma_i^T)} 1^{T \to T_r} \]

\[ = \begin{cases} \lim_{N \to \infty} \frac{\sum_{i=1}^{N} \sigma_i^T \sigma_i^{T+1}}{(\sum_{i=1}^{N} \sigma_i^T)} 1^{T \to T_r} & \text{when } \lfloor T \to T_r \in \mathbb{Z}_c, \\
\lim_{N \to \infty} \frac{\sum_{i=1}^{N} \sigma_i^T \sigma_i^{T+1}}{(\sum_{i=1}^{N} \sigma_i^T)} 1^{T \to T_r} & \text{when } \lfloor T \to T_r \in \mathbb{Z}_e. \end{cases} \]

On the other hand, when \(T_r \in \mathbb{Z}_c\) and \(T_f \in \mathbb{Z}_c\), the modulo operation in (54) leads to the same results as (57). Thus, Proposition 2 is proved.

Proposition 1 and Proposition 2 show that there are some regular patterns as changing the value of \(T\). In addition, such patterns are different when the true period is odd or even.

Based on the derived two propositions, we have the following two corollaries for the CPBD.

**Corollary 1:** When \(T_f, T_f, T_r \in \mathbb{Z}_r\), and \(T_r \in \mathbb{Z}_c\), there exist:

\[ D(f_{1,t+1} = 1; f_{1,t} = 1) | T_f \neq 0 = D(f_{1,t+1} = 1; f_{1,t} = 1) | T_f = 0; \]

\[ D(f_{1,t+1} = 1; f_{1,t} = 1) | T_r \neq 0 = D(f_{1,t+1} = 1; f_{1,t} = 1) | T_r = 0. \]

**Corollary 2:** When \(T_f, T_f, T_r \in \mathbb{Z}_r\), and \(T_r \in \mathbb{Z}_c\), there exist:

\[ D(f_{1,t+1} = 1; f_{1,t} = 1) | T_f \neq 0 = D(f_{1,t+1} = 1; f_{1,t} = 1) | T_f = 0; \]

\[ D(f_{1,t+1} = 1; f_{1,t} = 1) | T_r \neq 0 = D(f_{1,t+1} = 1; f_{1,t} = 1) | T_r = 0. \]
conventional one [12]. Then we present the outcomes of BN structure learning, which are obtained by the proposed algorithms in Section IV and Section V.

A. Comparison of Computational Complexity

The computational complexity is evaluated by running the Matlab codes of the proposed algorithm and the conventional one in a same desktop. The corresponding running time is recorded to reflect the real computational cost. In this subsection, the learning complexity is evaluated when considering the CBN structure that only consists of one C-clique, since the learning time of the whole CBN structure is linear to that of a C-clique. Additionally, the impacts of observation numbers and variable numbers are considered for a comprehensive comparison. In our simulations, the observations are generated based on \( M \) binomial distributions \(^6\). The simulated results are presented in Fig. 3 and Fig. 4, where the computational cost is the total running time for obtaining the conditional probability table and the measurement of every edge. It is obvious that the computational cost of the proposed algorithm is much less than the conventional one, especially as the number of base stations and observations increasing. In particular, when \( M = 12 \) and \( N = 36000 \), the conventional algorithm demands for 502.2 seconds to learn. It implies that the conventional algorithm is not suitable for online learning even when the number of variables is not large. In Fig. 4, we draw the ratio between the “Conv” and “Prop” costs over \( M \), where it is obvious that the proposed scheme is more efficient. According to Fig. 4, the proposed algorithm only requires \( 1/38 \) of the time cost as the conventional one when \( M = 12 \) and \( N = 36000 \). In addition, note that the proposed algorithm could adapt to any number of variables, while the conventional one cannot.

B. Learning Outcomes

The simulation scenario is illustrated by Fig. 1, where mobile users move from the right to the left and access one of the three primary base stations according to their locations. The base stations are respectively labeled #1, #2, and #3 from right to left. The simulation configuration is as follows: the arrival of mobile users follows a Poisson distribution with mean \( \lambda_s = 1 \); the length of the road is 600 meters, covered by the three cells; the data traffic arrival rate at each user follows a Poisson distribution with mean \( \lambda_d = 0.002 \);\(^7\) the service time of data traffic follows an exponential distribution with a mean of 2 seconds; and the spectrum sensing is performed once per second.

In this paper, we are interested in the relationship among base station activities, which is introduced by the user movement. Such statistical correlation is useful for CR operation but has not been well studied in the past. To show the learning results, the data of observation is collected in three different cases corresponding to the uniformly distributed speed regions as [43.2, 72], [72, 129.6], and [100.8, 158.4] kilometers per hour, respectively. The learning results of \( T \) (empirical period of CBN structure) are presented in Table V, where the unit of \( T \) is second. It is obvious that the period \( T \) is similar under different numbers of observations, which shows that the proposed algorithm could work robustly. According to the learning results, it could be concluded that the statistical relationship of the observed three base stations are mainly affected by the user speed, and the period is inversely proportional to the user speed, which is consistent with our hypothesis.

---

\(^6\) Actually, the distributions of observations do not impact on the computational cost of learning at all.

\(^7\) The value of \( \lambda_d \) is set small since we have to ensure what the primary base stations can have idle status.
speed. Since the case of \( N = 3600 \times 10 \) has the most number of observations, it could show the most trustable results when compared with the other two. Interestingly, we observe that \( T \times v \approx \frac{DN}{2} \) from our simulation results, where \( v \) is the averaged speed over the speed region. Actually, such relationship is reasonable, since \( \frac{DN}{2} \) is the mean of the distances between the mobile users and the end of the simulated road on the left.

| Speed Region | \( N = 3600 \) | \( N = 3600 \times 5 \) | \( N = 3600 \times 10 \) |
|--------------|----------------|-----------------------|-------------------------|
| \([43.2 \ 72]\) | \( T = 16 \) | \( T = 16 \) | \( T = 16 \) |
| \([72 \ 129.6]\) | \( T = 10 \) | \( T = 12 \) | \( T = 12 \) |
| \([100.8 \ 158.4]\) | \( T = 8 \) | \( T = 10 \) | \( T = 8 \) |

Next, we show the learning results of CPbD. For clarity of expression, we only show the results in the case of \([100.8 \ 158.4]\) and \( N = 3600 \times 10 \). When \( T = 8 \), the learned statistical pattern \( D_{1:7} \) is given as follows,

\[
D_1 = \begin{bmatrix} 1.0000 & 0.3818 & 0.0917 \\ 0.0841 & 1.0000 & 0.3592 \end{bmatrix}, \quad (63)
\]

\[
D_2 = \begin{bmatrix} 1.0000 & 0.5724 & 0.2733 \\ 0.0000 & 1.0000 & 0.4926 \end{bmatrix}, \quad (64)
\]

\[
D_3 = \begin{bmatrix} 1.0000 & 0.5041 & 0.1137 \\ 0.3804 & 1.0000 & 0.5247 \end{bmatrix}, \quad (65)
\]

\[
D_4 = \begin{bmatrix} 1.0000 & 0.4337 & 0.0345 \\ 0.0527 & 1.0000 & 0.3740 \end{bmatrix}, \quad (66)
\]

\[
D_5 = \begin{bmatrix} 1.0000 & 0.4379 & 0.1431 \\ 0.0000 & 1.0000 & 0.6035 \end{bmatrix}, \quad (67)
\]

\[
D_6 = \begin{bmatrix} 1.0000 & 0.5247 & 0.3372 \\ 0.2607 & 1.0000 & 0.5282 \end{bmatrix}, \quad (68)
\]

\[
D_7 = \begin{bmatrix} 1.0000 & 0.5490 & 0.2898 \\ 0.2796 & 1.0000 & 0.5051 \end{bmatrix}, \quad (69)
\]

\[
D_8 = \begin{bmatrix} 1.0000 & 0.5120 & 1.0000 \end{bmatrix},
\]

Apparently, the entries in the upper triangular part of \( D \) is generally larger than the entries in the lower triangular part, which implies that the status of base stations #2 and #3 is heavily impacted by base station #1, but the reverse does not hold. In addition, we use the learned CPbD \( D_{1:7} \) to draw a CBN shown in Fig. 5, where we see that the dependence between base station #1 and base station #3 increases as \( t \) increases during a period. It is due to the fact that the users served by base station #1 at the beginning of a period will arrive in the range of base station #3 at the end of the period. Note that if we add binary thresholding to make decisions on the existence of edges, which is used for binary testing of each edge in [12], the learning outcomes will be similar to that in [12]. Corresponding to the CPbD, the conditional probability table \( B_{1:7} \) is given by (70)-(76). Based on the learned conditional probability table, we could generate the joint distribution of the on/off behavior of three primary base stations and predict the status of network for the future, to serve broader CR applications.

In brief, the simulated outcomes show that the proposed CBN structure learning algorithm not only can quantify the correlation but also correctly reflects the unidirectional statistical pattern caused by the mobile user one-way movement. Therefore, our learning scheme can efficiently learn the CBN structure to represent the true statistical behavior of the underlying network.

**VII. CONCLUSION**

In this paper, we proposed a learning scheme to obtain the statistical activity pattern of a primary network in both spatial and temporal domains simultaneously. The proposed scheme incurs evidently lower computational complexity when compared with the traditional ones. Additionally, it is capable of learning the statistical period of the CBN structure. By simulations, we show that the learning results also correctly reflect certain network behavior beyond spectrum usage, which could be useful for broader CR network control applications.
Fig. 5. Illustration of learned CBN. The width of edges reflects the dependence.

\[
    B_4 = \begin{pmatrix}
        0.7994 & 0.7994 & 0.9990 \\
        0.7772 & 0.7033 & 0.0380 \\
        0.8372 & 0.2438 & 0.7023 \\
        0.7815 & 0.0138 & 0.0215 \\
    \end{pmatrix}
\]

(73)

\[
    B_5 = \begin{pmatrix}
        0.1436 & 0.8921 & 0.7139 \\
        0.2030 & 0.7397 & 0.0161 \\
        0.0168 & 0.2140 & 0.7505 \\
        0.0136 & 0.0139 & 0.0204 \\
        0.9990 & 0.9990 & 0.9990 \\
        0.8254 & 0.7387 & 0.0227 \\
        0.6919 & 0.1290 & 0.8199 \\
        0.7568 & 0.0131 & 0.0083 \\
        0.1570 & 0.8431 & 0.6559 \\
        0.1948 & 0.7329 & 0.0241 \\
        0.0372 & 0.2062 & 0.7173 \\
        0.0113 & 0.0137 & 0.0196 \\
        0.9990 & 0.9990 & 0.5002 \\
        0.7495 & 0.7703 & 0.0218 \\
        0.7709 & 0.1436 & 0.7994 \\
        0.7599 & 0.0132 & 0.0156 \\
        0.2313 & 0.7687 & 0.6536 \\
        0.1843 & 0.7866 & 0.0214 \\
        0.0092 & 0.2055 & 0.7250 \\
        0.0119 & 0.0128 & 0.0219 \\
    \end{pmatrix}
\]

(74)

\[
    B_6 = \begin{pmatrix}
        0.7994 & 0.7994 & 0.9990 \\
        0.7772 & 0.7033 & 0.0380 \\
        0.8372 & 0.2438 & 0.7023 \\
        0.7815 & 0.0138 & 0.0215 \\
    \end{pmatrix}
\]

(75)

\[
    B_7 = \begin{pmatrix}
        0.7994 & 0.7994 & 0.9990 \\
        0.7772 & 0.7033 & 0.0380 \\
        0.8372 & 0.2438 & 0.7023 \\
        0.7815 & 0.0138 & 0.0215 \\
    \end{pmatrix}
\]

(76)

REFERENCES

[1] J. Mitola and J. Maguire, G.Q., “Cognitive radio: making software radios more personal,” Personal Communications, IEEE, vol. 6, no. 4, pp. 13–18, Aug 1999.

[2] S. Dudley, W. Headley, M. Lichtman, E. Imana, X. Ma, M. Abdelbar, A. Padaki, A. Ullah, M. Solal, T. Yang, and J. Reed, “Practical issues for spectrum management with cognitive radios,” Proceedings of the IEEE, vol. 102, no. 3, pp. 242–264, March 2014.

[3] J. Ma, G. Li, and B.-H. Juang, “Signal processing in cognitive radio,” Proceedings of the IEEE, vol. 97, no. 5, pp. 805–823, May 2009.

[4] W. Han, J. Li, Z. Li, J. Si, and Y. Zhang, “Spatial false alarm in cognitive radio network,” Signal Processing, IEEE Transactions on, vol. 61, no. 6, pp. 1375–1388, March 2013.

[5] Z. Quan, S. Cui, and A. Sayed, “Optimal linear cooperation for spectrum sensing in cognitive radio networks,” Selected Topics in Signal Processing, IEEE Journal of, vol. 2, no. 1, pp. 28–40, Feb 2008.

[6] K. Murphy, S. Mian et al., “Modelling gene expression data using dynamic bayesian networks,” Technical report, Computer Science Division, University of California, Berkeley, CA, Tech. Rep., 1999.

[7] R. Daly, Q. Shen, and S. Aitken, “Review: Learning bayesian networks: Approaches and issues,” Knowl. Eng. Rev., vol. 26, no. 2, pp. 99–157, May 2011.

[8] V. Mihajlovic and M. Petkovic, “Dynamic bayesian networks: A state of the art,” University of Twente, Centre for Telematics and Information Technology, Tech. Rep., 2001.

[9] K. P. Murphy, Machine learning: a probabilistic perspective. MIT press, 2012.

[10] P. Shaugnessy and G. Livingston, “Evaluating the causal explanatory value of bayesian network structure learning algorithms,” Research Paper, Department of Computer Science, University of Massachusetts Lowell., 2005.

[11] D. M. Chickering, D. Heckerman, and C. Meek, “Large-sample learning of bayesian networks is np-hard,” J. Mach. Learn. Res., vol. 5, pp. 1287–1330, Dec. 2004.

[12] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu, “Learning bayesian networks from data: An information-theory based approach,” Artif. Intell., vol. 137, no. 1-2, pp. 43–90, May 2002.

[13] D. P. Bertsekas, R. G. Gallager, and P. Humblet, Data networks, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ, 1987, vol. 2.

[14] W. Han, J. Li, Z. Li, J. Si, and Y. Zhang, “Efficient soft decision fusion rule in cooperative spectrum sensing,” Signal Processing, IEEE Transactions on, vol. 61, no. 8, pp. 1931–1943, April 2013.