2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems

Wireless Spectrum Occupancy Prediction Based on Partial Periodic Pattern Mining

Pei Huang, Chin-Jung Liu, Li Xiao, Jin Chen
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48823, USA
{huangpe3, liuchinj, lxiao, jinchen}@msu.edu

***Abstract*—Cognitive radio appears as a promising technology to allocate wireless spectrum between licensed and unlicensed users in an efficient way. The availability of spectrum holes vastly affects the throughput and delay of unlicensed users. Predictive methods for inferring the availability of spectrum holes can help to improve spectrum extraction rate and reduce collision rate. In this paper, a spectrum occupancy prediction model based on Partial Periodic Pattern Mining (PPPM) is introduced. The mining aims to identify frequent spectrum occupancy patterns that are hidden in the spectrum usage of a channel. The mined frequent patterns are then used to predict future channel states (i.e., busy or idle). Based on the prediction, unlicensed users will be able to make use of spectrum holes efficiently without introducing significant interference to licensed users. PPPM outperforms traditional Frequent Pattern Mining (FPM) by considering real patterns that do not repeat perfectly due to noise, sensing errors, and irregular behaviors. Using real life network activities we show a significant reduction on miss rate in channel state prediction. With the proposed prediction mechanism, the performance of Dynamic Spectrum Access (DSA) is substantially improved.**

***Index Terms*—Cognitive Radio; Dynamic Spectrum Access (DSA); Occupancy Prediction; Partial Periodic Pattern Mining;**

I. Introduction

The proliferation of wireless services has resulted in a dense allocation of frequency bands. Traditional static frequency allocation does not account for variations of usage in both spatial and temporal domains. Recent measurement studies reveal that a large part of licensed wireless spectrum bands are underutilized [1], whereas many services such as WLAN, Bluetooth, ZigBee, are confined to crowded unlicensed bands. To exploit the temporally unused spectrum, the concept of Cognitive Radio (CR) is proposed. CR is a radio that is aware of and can learn from its surrounding environment and adjust its operating parameters accordingly [2]. In a CR network (CRN), unlicensed users known as secondary users (SUs) sense the licensed channels and use the available slots (spectrum holes) that are unused by primary users (PUs). SUs must immediately vacate the channels upon PUs' return [3].

Dynamic behaviors of PUs require accurate sensing and fast switch strategies to convince service providers to allow SUs to operate within their licensed bands. Recently due to the difficulty of accurate sensing for low power signals, FCC has decided to adopt the geo-location and database access method, eliminating the spectrum sensing requirement for unlicensed TV band devices [4]. Although this approach provides accurate information of spectrum availability, all SUs operating in the same area have identical information about spectrum availability and thus spectrum sharing among SUs is still a tough problem. Due to power asymmetry or different physical layer standards, transmissions of some SUs are prone to be corrupted by the other SUs. Therefore, some SUs actually act as virtual PUs towards the other SUs. Spectrum occupancy prediction helps a CR device extract spectrum holes while avoiding time slots of high collision probability. With the controllable collision rate achieved by prediction-assisted DSA, more service providers will be willing to open their spectrum for additional return on investment.

Wireless communications are designed to follow certain protocols. Patterns are expected to be observed in spectrum usage. Recently, a Frequent Pattern Mining (FPM) method has been presented in [5] to show encouraging prediction performance by using the discovered frequent patterns. In this paper, a Partial Periodic Pattern Mining (PPPM) based spectrum occupancy prediction (i.e., channel state prediction) method is introduced. Partial periodic patterns, which are different from full periodic patterns, identify regularity of behaviors at some but not all points of time [6]. For example, a pattern disclosing that the channel utilization of a channel is high during certain hours every day but not regular in other time is a partial periodic pattern. A spectrum occupancy pattern is affected by various factors such as noise, sensing errors, and irregularity of users' behaviors. Hence, the periodicity might partially be observable. PPPM algorithm is tailored for identification of realistic patterns that are irregular in nature. Using PPPM, more occupancy patterns are identified, leading to extra channel state prediction rules that reduce the miss rate in channel state prediction.

Besides accurate prediction, timely estimation of spectrum availability is also critical. Hence, we speed up the mining process in two phases. First, for each candidate frequent pattern, we reduce the number of subsequences that need to be examined in a time series by introducing an index list structure. Using the index list structure, counting the occurrences of a pattern does not require us to scan the entire database and thus improves mining expedition. Second, we reduce the number of candidate frequent patterns by identifying patterns that cannot yield longer frequent patterns so as to stop mining on them early. Moreover, by checking whether a pattern will be absorbed by another pattern, we avoid redundant mining

1526-7539/12 $26.00 © 2012 IEEE
DOI 10.1109/MASCOTS.2012.16

on two branches that yield identical frequent patterns. In summary, the contributions of this work are as follows.

- An efficient Partial Periodic Pattern Mining (PPPM) algorithm is proposed to mine rules for predicting the channel state in the next time slot.
- An index list structure along with an Apriori-like property and a backward-extension rule are introduced to speed up the mining process.

We compared the performance of our PPPM with the FPM algorithm using real life network activities and data collected in the Personal Communication Service (PCS) bands. The results show that the proposed PPPM-based prediction significantly reduces the miss rate over FPM-based prediction. This proves that spectrum usage exhibits partial periodicity. In addition, we particularly observed that distinguishing low utilization periods from high utilization periods and mine rules in corresponding utilization periods will substantially improve the prediction performance.

In order to show the advantages of integrating prediction in DSA, we compared our prediction-assisted DSA with a statistical knowledge-based DSA and demonstrated that the prediction of channel state significantly improves the **spectrum extraction rate**, which is defined as the ratio between the number of idle slots that are actually obtained by SUs and the total number of available idle slots left by PUs.

The rest of the paper is organized as follows. In Section II, we discuss related work in spectrum prediction and partial periodic pattern mining. We formalize our mining problem in Section III. In Section IV, we study partial periodic pattern mining and rule extraction. Performance studies are reported in Section V, followed by concluding remarks in Section VI.

II. Related Work

Various models have been developed to provide prediction for signal power, duty cycle, and channel state. We give a brief review over them in this section and include some related work on partial periodic pattern mining.

Several papers [7] [8] [9] model the variation of signal power on a channel. In [7], a second-order autoregressive (AR-2) process is used to model the channel variation. Once the parameters of the AR-2 model are computed, the signal power can be predicted and the channel state can be estimated via a Kalman filter. In [8], a moving average model (MA) is introduced and it is integrated with the AR model to yield an autoregressive moving average model (ARMA) for estimating the signal power in a channel. The results show that the time series of all TV channels fall into the moving average model. An ARMA model requires that the time series is stationary, which means the statistical properties of a time series is similar to those of time shifted series [10]. An autoregressive integrated moving average (ARIMA) model is introduced in [10] to handle non-stationary time series. It models the change of the band occupancy, which is defined as the ratio between the number of occupied channels and the total number of channels in a given band. An inconvenience is that a time series must be converted to a stationary and periodic time series in order to analyze it.

Instead of predicting signal power, many papers consider that the channel is either detected as occupied or unoccupied and the channel states constitutes a binary time series. Most studies assume that a Markov chain exists in such a binary time series and thus hidden Markov models (HMMs) are the most commonly used models for channel state prediction. Assuming that the primary user's traffic follows a Poisson process, a HMM-based dynamic spectrum access scheme is introduced in [11]. In [12], several deterministic traffic patterns are studied for the HMM-based channel state predictor. A more detailed analysis of the HMM-based predictor design is presented in [13], where a multilayer perceptron (MLP) based channel state predictor is also analyzed. The existence of Markov chain in spectrum utilization by PUs has been validated in [14] by using data collected in the paging bands. Introducing one more state named "fuzzy" (unknown availability), a higher-order HMM channel state predictor is introduced in [15]. The Markov chain has also been used to model the change of the duty cycle [16]. Recently, a FPM-based method [5] demonstrates that it provides better channel state prediction performance over the first-order HMM-based predictor. In this paper, we show that PPPM can provide more prediction rules than FPM, resulting in fewer unpredictable slots.

Mining patterns with gap constraints has been studied for sequence databases [17] [18] [19] [20]. Their methods, however, are not suitable for our problem. The number of entries in a sequence database is fixed. On the contrary, the number of sequences in a time series is changing along with the length of a sequence. Partial periodic pattern mining in time series databases has been studied in [6] [21], but the max-subpattern hit algorithm aims at enumerating all partial periodic patterns of a single period. It has to construct a max-subpattern tree for each single period. In our problem, the periods of patterns are unknown. Therefore, for a set of periods, the max-subpattern hit algorithm incurs high overhead for looping over single period mining. In spectrum occupancy prediction, we do not need a complex structure for enumerating all partial periodic patterns of a single period. Many of them are inappropriate for use because they include too many uncertain symbols. Therefore, we need a gap constraint (introduced in Section III) to filter out useless patterns. We have the same general concept of partial periodic pattern, but our definition of partial periodic pattern and the corresponding mining method are unique to the spectrum occupancy pattern mining. We also step further to extract prediction rules from mined patterns and utilize them to improve wireless communication performance.

III. Problem Formulation

In this section, we give a formal definition of partial periodic pattern mining in a spectrum usage database.

In CRNs, the SUs are responsible for sensing the channel before transmission so as to guarantee that the collision probability perceived by the PUs is under certain limit. Multiple PUs can be regarded as a virtual PU. If the spectrum usage of the virtual PU exhibits some patterns, SUs can learn from the sensing results and utilize these frequent patterns to reduce collision rate. In performance study we show that there indeed

exist patterns due to social behavior and common habits. We assume that SUs can recognize their own transmissions via signal features or they set silence periods to get training set.

Let '1' denote busy and '0' represent idle. The channel states in a period can be represented by a binary time series. Let I be the alphabet of all possible values that occur in the time series, we have $I = \{0, 1\}$. A **wild-card** (denoted by a single $*$) is a special character that matches any value in I. A **gap** is a sequence of wild-cards, and the *size* of a gap is the number of wild-cards in it. We use g^m to represent a gap whose size is within the range $[0, m]$, which means the wild-card $*$ can be repeated at most m times in a gap. Here, m is the **gap constraint**. If a pattern has too many wild cards, any sequence of channel state observations can match the pattern. The prediction power of the pattern is weak. Therefore, we stop developing a pattern if it violates the gap constraint. Same as other PPPM algorithms [17] [18] [19] [20], it is a user-defined parameter that needs to be input to the mining algorithm.

A **symbol** in a pattern is a **value** in I or a wild-card $*$. Given a pattern P, we use $P[i]$ to represent the ith symbol of P and p_i to represent the ith value of P. A pattern $P = p_1 g^m p_2 g^m ... p_{q-1} g^m p_q$ is a set of values and gaps. We define the *length* of a pattern P (denoted by $l = |P|$) as the number of symbols in P, which means the wild-cards are also counted towards the pattern's length. If $l \geq 2$, the substring that contains the first $l-1$ symbols is called the *prefix* of P (denoted by *prefix(P)*), and the substring that contains the last $l-1$ symbols is the *suffix* of P (denoted by *suffix(P)*).

A pattern $P = P[1]P[2]...P[l_1]$ is a *subpattern* of $C = C[1]C[2]...C[l_2]$ if $l_2 > l_1$ and there exists an integer $1 \leq i \leq l_2 - l_1 + 1$ such that $P[1] = C[i], P[2] = C[i+1], ..., P[l_1] = C[i+l_1-1]$, and C is a *superpattern* of P, which is denoted by $P \sqsubseteq C$. For example, $P = \langle 11 * 00 \rangle$ is a pattern of length 5, *prefix*$(P) = \langle 11 * 0 \rangle$ and *suffix*$(P) = \langle 1 * 00 \rangle$, $\langle 1 * 0 \rangle$ is a subpattern of P but $\langle 100 \rangle$ is not. We do not regard $\langle 100 \rangle$ as a subpattern of $\langle 1 * 0 \rangle$ because that violates the Apriori property in data mining (i.e., the support of a pattern cannot exceed the support of any of its subpatterns [22]).

Our goal is to identify frequent patterns in a time series $S = s_1 s_2 ... s_n$ where s_i is the ith value of S. We need to define the term *frequency* and how often a pattern should occur before we can consider it *frequent* in S. We define the *frequency* of a pattern P by the probability of observing P if we randomly pick a subsequence of length l in S. The time series S can be divided into N_l subsequences of equal length l that start at different starting points: $S_i = s_i s_{i+1} ... s_{i+l-1}$ where $1 \leq i \leq n-l+1$ and $N_l = n-l+1$. A subsequence S_i *matches* a pattern P of length l if for each position $1 \leq j \leq l$, $s_{i+j-1} \in P[j]$. The **support** of P in S (denoted by $sup(P)$) is the number of subsequences in S that match the pattern. The **confidence** of P (denoted by $conf(P)$) is defined as the division of its support by the total number of subsequences of length l in S, which well reflects the probability of observing P in S.

We say that a pattern is a *frequent partial periodic pattern* in a time series if its confidence exceeds a user-specified threshold ρ_c, which is defined as the **confidence constraint**.

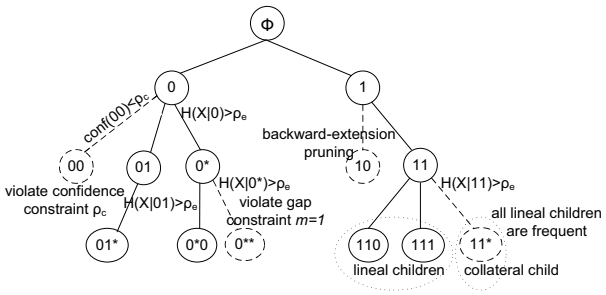

Fig. 1. An illustration of the frequent partial periodic pattern tree.

For example, if a pattern of length 3 $P = \langle 0 * 1 \rangle$ appears 5 times in the time series $S = \langle \mathbf{00}11\mathbf{00}110\mathbf{0}01 \rangle$, the confidence of P is $conf(P) = sup(P)/N_l = 5/10 = 1/2$. If $\rho_c = 0.5$, P can be considered as a frequent pattern. We use confidence instead of support to measure the frequency because the supports of long patterns are usually lower than that of short patterns. This is because the total number of sequences of length l (i.e., N_l) is monotonically decreasing along with the increasing length l. The support is thus normalized by N_l to correctly reflect the frequency.

IV. Partial Periodic Pattern Mining

In this section, we describe how to generate a candidate frequent pattern and how to count its support as well as how to prune candidate frequent patterns that need to be checked.

A. Frequent Pattern Enumeration

The complete search space of frequent partial periodic patterns forms a tree as shown in Fig. 1. Starting at a root node that is labeled with Φ, we can recursively extend a node of level L on the tree by concatenating a value in I or a wild-card that satisfies the gap constraint to get a child node on the next level $L+1$. Whether a parent node Q will develop a child node by concatenating a wild-card largely depends on the conditional entropy $H(X|Q)$ defined in Definition 2. Entropy is a measure of the uncertainty inherent in the distribution of a random variable. When the entropy of a random variable is large, it implies that the uncertainty as to the value of that random variable is large. We hence may consider it as a noise, but the noise should not affect our discovery of long patterns. We give the definitions of entropy and conditional entropy as follows.

Definition 1 *The entropy of a discrete random variable X is defined as $H(X) = -\sum_{x \in I} p(x) log_2 p(x)$ where $p(x)$ is the occurrence probability of x and $I = \{0, 1\}$.*

Definition 2 *For a given node Q, the conditional entropy of a discrete random variable X after node Q is defined as $H(X|Q) = -\sum_{x \in I} p(x|Q) log_2 p(x|Q)$ where $p(x|Q)$ denotes the fraction of subsequences with prefix Q in S that match $P = Q \oplus x$ and $\oplus$ denotes concatenation.*

If the conditional entropy of a node $H(X|Q)$ is higher than a user-specified threshold ρ_e, the node will develop a

child node by extending itself with a wild-card if both the gap constraint and the confidence constraint are not violated and at least one child with a value extension cannot exceed the confidence threshold. The ρ_e specifies how the user defines the uncertainty. In our case, if no value (i.e., either 0 or 1) appears after a pattern with a probability higher than 75%, we consider that the next slot following the pattern is uncertain. Consequently, a child with the extension of $*$ may be generated and it is the *collateral child* of the pattern. The other two children with a value extension are the *lineal children* of the pattern. The children of a node are generated and arranged according to a lexicographical order. If a node is not frequent, we stop developing its subtree. To save memory, we grow a pattern in a depth-first search way so that we can remove a branch when it is done.

The introduction of the collateral child has two advantages. First, channel sensing is not perfect. Misdetections will reduce the observed occurrences of a pattern. The wild-cards can save some frequent patterns that are near the decision boundary. For example, two patterns $\langle 00011 \rangle$ and $\langle 00010 \rangle$ may occur 987 and 324 times respectively. However, the confidence constraint requires that a pattern of length 5 must occur at least 1000 times to be considered frequent. If most occurrences of the two patterns are followed by a '0', we miss an important frequent pattern $\langle 0001 * 0 \rangle$ and any frequent pattern that has the prefix of $\langle 0001 * 0 \rangle$. Noise and sensing errors can make the occurrences of a pattern fall below the support threshold. Therefore, when the lineal children of a node cannot pass the confidence constraint while its collateral child can, we go on developing the subtree of this node so as to avoid missing any possible frequent pattern. Mining on this subtree will terminate soon either due to violation of gap constraint (we do not allow too many consecutive *) or due to violation of confidence constraint (the support of the pattern is already very low).

Second, even when one lineal child of a node is frequent, we may want to develop an additional pattern that accounts for the irregularity of behaviors. Suppose one of the two patterns described above passes the confidence constraint with some irregular slots (e.g., $\langle 00010 \rangle$ occurs 1328 times and $\langle 00011 \rangle$ occurs 964 times). The irregularity will be detected by the entropy of their parent (i.e., $H(X \mid 0001)$). An additional pattern $\langle 0001* \rangle$ is developed along with $\langle 00010 \rangle$. The reason is that the regularity of behaviors during a certain period of time may not be identified. For example, most people make phone calls in a pure random manner. It may be unable to identify regular channel activities during certain periods of time but it is desirable to discover frequent patterns like $\langle 0001* * * * *0001 \rangle$ so that we can predict the next channel state no matter busy states $\langle 00011111 \rangle$ or idle states $\langle 00010000 \rangle$ are observed. These partial periodic patterns specify regularity of behaviors at some but not all points of time. Considering that partial periodicity exists ubiquitously in real life, more patterns can be discovered and more useful prediction rules can be extracted. If all of the lineal children are frequent, we consider them as two different frequent patterns and do not generate the collateral child.

B. Support Count

For each candidate pattern, we have to check whether it is frequent by counting its support. This requires us to scan the time series S once. Some algorithms traverse the entire time series as many times as needed. Since we perform depth-first search, we can use an index list structure to reduce the number of subsequences that need to be checked.

Given a time series S and a pattern P of length l, the *head index list* of P (denoted by $HIL(P)$) is a list of indices where the P appears in S. For example, if $S = \langle \mathbf{00}11010\mathbf{0}010 \rangle$ and $P = \langle 0 * 1 \rangle$, then $HIL(P) = 1, 2, 8$. Given $HIL(P)$, one can easily get the $sup(P)$ by returning the length of the HIL.

A pattern Q of length $l-1$ has three children: $P_1 = Q \oplus 0$, $P_2 = Q \oplus 1$, and $P_3 = Q \oplus *$. Their head index lists can be computed by examining subsequences that start at positions specified by $HIL(Q)$ using the following procedure.

```
for x ∈ HIL(Q) do
    if (x + l − 1) ≤ n then
        if S[x + l − 1] == '0' then
            Insert x in HIL(P1);
            sup(P1)++;
        else
            Insert x in HIL(P2);
            sup(P2)++;
        end
    end
end
```

For each head index x in $HIL(Q)$, if there exists a subsequence of length l that starts at x in S, we check the last value of the subsequence. If the value is equal to '0', the head index x is inserted to $HIL(P_1)$, otherwise the head index x is inserted to $HIL(P_2)$. The HIL of the collateral child P_3 is given by

$$HIL(P_3) = HIL(Q) \tag{1}$$

Because we do depth-first search, the length of the HIL is monotonically decreasing until we reach an invalid candidate pattern and stop developing the subtree. This speeds up the mining procedure quite a lot in practice because we skip a large number of subsequences and there is no time-consuming string comparison.

C. Candidate Pattern Pruning

A common problem in data mining is that the number of candidate patterns is huge. There exist about 3^l candidate patterns for a certain length l in our problem. It is infeasible to enumerate all candidate frequent patterns and count their supports. In this section, we introduce two constraints for efficient candidate pattern pruning.

1) Confidence constraint: For efficient pruning, most mining algorithms adopt the *Apriori* property, which states that any subpattern of a frequent pattern must also have the minimum support [22]. We can extend the *Apriori* property to the concept of confidence if the total number of sequences is fixed.

However, the number of unique sequences of length l in a time series is different for different values of l. For example, consider a pattern $P = \langle 001 \rangle$ and its subpattern $Q = \langle 00 \rangle$ in a time series $S = \langle 001001 \rangle$, we see that $sup(P) = sup(Q) = 2$, but $N_2 = 5$ and $N_3 = 4$. As a result, $conf(P) = \frac{2}{4}$ and $conf(Q) = \frac{2}{5}$, which shows that the confidence of a pattern may exceed the confidence of its subpattern. Therefore, we cannot prune a candidate pattern even though its confidence is lower than the user-specified threshold ρ_c; otherwise we are unable to discover long patterns.

To achieve efficient pruning, we derive an *Apriori-like* property in Theorem 1.

Theorem 1 *Given a length l pattern P and a length $(l-d)$ subpattern $Q = P[i]P[i+1]...P[i+l-d-1]$ of P, where $1 \leq i \leq d+1$, we have $conf(Q) \geq \frac{N_l}{N_{l-d}}\rho_c$.*

Proof: Because Q is a subpattern of P, we have $sup(Q) \geq sup(P)$. If a length l pattern P is frequent, then by definition, we have $conf(P) = sup(P)/N_l \geq \rho_c$. Now, consider a length $(l-d)$ subpattern Q of P, we have

$$conf(Q) = \frac{sup(Q)}{N_{l-d}} \geq \frac{sup(P)}{N_{l-d}} \geq \frac{N_l}{N_{l-d}}\rho_c = \lambda_{l,l-d} \cdot \rho_c \quad (2)$$

■

Theorem 1 implies that any length $(l-d)$ subpattern Q of a frequent pattern P must retain a confidence that is not less than $\lambda_{l,l-d} \cdot \rho_c$. This Apriori-like property allows us to prune a large number of candidate patterns from consideration. Suppose the length of the longest frequent pattern in the time series S is l_m. If the confidence of a length i pattern is less than $\lambda_{l_m,i} \cdot \rho_c$, we can stop growing the pattern. This requires that the user has a rough idea about the value of the l_m, and we can guarantee that all frequent patterns of length less than or equal to l_m will be discovered. There exist some periodicity detection methods [23] [24] that can be used to determine the length of the longest pattern. Users can also determine the pattern length that is meaningful (e.g., a length of 500 ms). Since a CR device needs to estimate the channel state instantly, the rule set cannot be too large. A max length of tens of bits is usually sufficient.

2) Backward-extension constraint: To further reduce the number of candidate frequent patterns that need to be checked, we identify patterns that can be absorbed by other patterns. Suppose $P = P[1]P[2]...P[l]$ is a pattern of length l in time series S. Given another pattern $C = p_0 P$ where $p_0 \in I$, if $sup(P) = sup(C)$, we say p_0 is a *backward-extension item* of P. Obviously, if P has a backward-extension item p_0, it can be absorbed by C. For example, if there is a '0' before any occurrence of '110', the pattern $\langle 110 \rangle$ can be absorbed by $\langle 0110 \rangle$. Any pattern developed under the tree of '110' has the '0110' as the prefix. It is redundant to develop both subtrees.

To check the backward-extension item of a pattern P, we extract the head indices from $HIL(P)$ and examine whether the values located one symbol before all occurrences of P are the same value in I. Since $I = \{0, 1\}$, we use summation to check the condition as follows.

```
Cnt ← 0;
boundary ← False;
for x ∈ HIL(P) do
    if x ≥ 2 then
        Cnt ← sum (Cnt, S[x − 1])
    else                    /* reach boundary */
        boundary ← True
    end
end
if Cnt == 0 or Cnt == len (HIL(P)) or (Cnt
== len (HIL(P))-1 and boundary) then
    P can be safely pruned;
end
```

When we count the support for a pattern P, we also add up the value that is one symbol ahead of each matched subsequence. If the sum is 0, it implies that there is a '0' before any occurrence of P. If the sum is equal to $len(HIL(P))$, it implies that there is a '1' before any occurrence of P. If there exists a $x = 1$ in $HIL(P)$, we reach the boundary and we can regard that there is a '1' before any occurrence of P if the sum is equal to $len(HIL(P)) - 1$. In any of the three cases, the pattern P can be absorbed by another pattern and we do not develop the subtree of P. The pruning method is very efficient because if we use a simple candidate-maintenance-and-test method, in a case where $\langle 10 \rangle$ can be absorbed by $\langle 110 \rangle$, $\langle 110 \rangle$ must be mined before $\langle 10 \rangle$ so that $\langle 10 \rangle$ can be compared with $\langle 110 \rangle$. However, in our method, we can discover that $\langle 10 \rangle$ will be absorbed by $\langle 110 \rangle$ even if we do depth-first search for $\langle 10 \rangle$ before we do depth-first search for $\langle 11 \rangle$ as shown in Fig. 1. The redundant development of $\langle 10 \rangle$ is efficiently avoided.

D. Channel State Prediction

The mining algorithm outputs all frequent patterns and extracts prediction rules at the same time. A prediction rule is defined as $P \Rightarrow C$, where C is a superpattern of P. The confidence of a rule is defined as $conf(P \Rightarrow C) = sup(C)/sup(P)$. For example, if $\langle 0001 \rangle$ appears 1000 times and $\langle 00010 \rangle$ appears 926 times, the confidence of the prediction rule $\langle 0001 \rangle \Rightarrow \langle 00010 \rangle$ is 92.6%, which declares that the channel state is predicted to be idle in the next slot with a confidence of 92.6% when past channel states $\langle 0001 \rangle$ are observed. In the following discussions, we present a rule $P \Rightarrow C$ as $P \Rightarrow c$, where $C = P \oplus c$.

We discover both partial periodic patterns and full periodic patterns. To improve prediction efficiency by reducing the number of rules, we examine whether a rule can be absorbed by another rule. Suppose two rules $P_1 \Rightarrow c$ and $P_2 \Rightarrow c$ yield the same prediction of c. If they have the same length, and $P_1[i] \in P_2[i]$ for each position i, and $conf(P_1 \Rightarrow c) \leq conf(P_2 \Rightarrow c)$, the first rule can be absorbed by the latter one. For example, R_1 can be absorbed by R_2 if $conf(R_1 : \langle 01010 \rangle \Rightarrow 1) \leq conf(R_2 : \langle 0 * 010 \rangle \Rightarrow 1)$, but the fusion cannot be performed if $conf(R_1) > conf(R_2)$.

Finally, in the prediction stage we start with patterns of the longest length. If past observations of channel state match the

TABLE I
THE CHANNEL STATE PREDICTION RESULTS UNDER DIFFERENT RULE CONFIDENCE CONSTRAINTS.

rule confidence constraint R_c		0.9	0.8	0.7
FPM	accuracy	90.24%	85.53%	79.53%
	miss rate	53.27%	33.79%	9.53%
PPPM	accuracy	89.74%	84.10%	79.05%
	miss rate	47.86%	24.69%	4.73%

pattern P in a rule $P \Rightarrow c$, we predict the channel state in the next slot as c. Because a wild-card $*$ matches any value, it is possible that patterns from multiple rules are matched. In such a case, we adopt the rule with the largest number of matched values. If several patterns have the same number of matched values, we adopt the rule with the highest confidence. We stop searching for shorter length rules once prediction can be made. The reason is that a longer match usually leads to a more accurate prediction. A shorter pattern may provide a prediction of higher confidence, but it includes contributions from multiple long patterns. For example, $\langle 010 \rangle$ appears whenever $\langle 1010 \rangle$ or $\langle 0010 \rangle$ appears. When channel states '1010' are observed, it is better to use the rule $\langle 1010 \rangle$ instead of $\langle 010 \rangle$. If no pattern can be matched, we do not predict and it is regarded as a miss. We define **prediction accuracy** as the ratio between the number of correctly predicted slots and the total number of predicted slots. The ratio between the number of unpredictable slots and the total number of slots is defined as the **miss rate**.

V. PERFORMANCE STUDY

In this section, we compare the prediction performance of our partial periodic pattern mining (PPPM) with the frequent pattern mining (FPM) [5] in which the patterns must be full periodic. We show that more prediction rules are extracted from patterns mined by PPPM. Comparing with the optimal statistical knowledge-based dynamic spectrum access strategy [25], we show that the spectrum extraction rate is improved with the prediction.

A. Prediction performance

To get real-life network activities, we use the network traffic trace collected in the SIGCOMM'08 [26] as a reference. The wireless network activities were captured by eight 802.11a monitors with 2 monitors on each channel for four days. We divide the time into slots of 20 ms and convert the trace into a binary time series of channel states: if channel activities are detected in a slot, we output a '1'; otherwise we output a '0'. There is a beacon every 100 ms and thus 5 bits cover a beacon interval. Depending on the application, the slot size can vary. Short idle periods are hard to exploit for use in practice and thus an entire slot can be considered as busy from a practical point of view if channel activities are detected.

To investigate the prediction accuracy and the miss rate, we use one day's data for training and one day's data for test. Table I summarizes the prediction results under different rule confidence constraints (R_c). As introduced in Section IV-D, the confidence of a rule $P \Rightarrow c$ is defined as the fraction of pattern P's occurrences that are followed by c ($conf(P \Rightarrow c) = sup(P \oplus c)/sup(P)$). In order to bound the prediction accuracy, only rules of confidence that is higher than R_c are used. Generally, PPPM reduces the miss rate by around $5\% \sim 9\%$ with a sacrifice of about 1% on prediction accuracy. The lower miss rate implies that more useful rules are extracted from patterns mined by PPPM, which validates that some patterns only exhibit partial periodicity and they cannot be discovered by FPM. It is observable that the miss rate is significantly reduced when rules of lower confidences are used. It implies that most rules only have confidences between 0.7 and 0.9. The prediction accuracy thus will be pulled down to around 80% if most slots are predicted.

The major reason for prediction errors and high miss rate is the changing utilization. A channel may be busy in some time periods and is underutilized in the other time periods. Fig. 2 shows the change of channel utilization over one days' measurement. We define the channel utilization as the ratio between the number of slots that are utilized by PUs and the total number of slots. Using all data together makes the prediction power of patterns weak. The same pattern P has a higher probability to be followed by a '1' during high utilization periods but has a lower probability to be followed by a '1' in low utilization periods. If we mix data of high and low utilization periods, pattern P is unable to yield any prediction. This causes the high miss rate. Another possibility is that in a global view we may get a rule $P \Rightarrow 1$. In low utilization periods, however, P is more frequently followed by a '0' rather than a '1'. The rule $P \Rightarrow 1$ thus results in many prediction errors during low utilization periods. Therefore, rules mined during high utilization periods may not be applicable in low utilization periods and vice versa. Mixing them can cause errors in their counterparts. It is desirable to distinguish low channel utilization periods from high channel utilization periods and mine rules in corresponding utilization periods.

We divide data into two sets based on the channel utilization. If the channel utilization is above a certain threshold δ, we accumulate data in the high utilization set. If the channel utilization drops below δ, we accumulate data in the low utilization set. Rules mined on the two sets are used to predict channel states in corresponding utilization periods. With $\delta = 0.4$ and $R_c = 0.8$, the performance of prediction is summarized in Table II. In low channel utilization periods, the regularity of channel usage is easy to identify (there is a beacon every 100 ms). The prediction performance is much better than that in high utilization periods.

In high utilization periods, patterns are still expected to be observable because wireless communications are designed to follow certain protocols. The behaviors of a single device are slightly easy to predict. When the aggregate behaviors of multiple devices are considered, the regularity becomes obscure. The prediction performance in high utilization periods is thus not as good as that in low utilization periods. Here PPPM outperforms FPM by identifying partial periodicity of patterns. The miss rate is reduced by $10\% \sim 15\%$ in high utilization periods. A transmission can happen at any time.

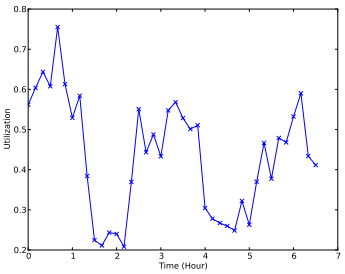

Fig. 2. Channel utilization of Wi-Fi in a resolution of ten minutes.

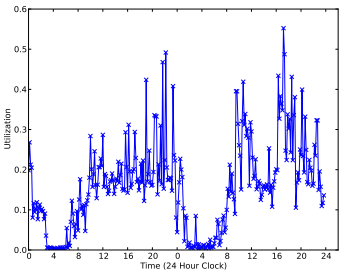

Fig. 3. Channel utilization on a channel in the PCS bands in a time resolution of ten minutes.

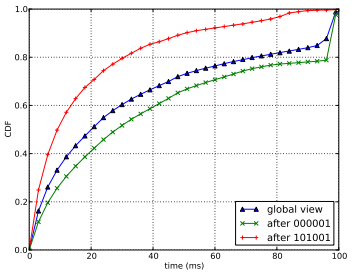

Fig. 4. CDF of the idle durations in Wi-Fi networks.

TABLE II
THE CHANNEL STATE PREDICTION RESULTS IN DIFFERENT CHANNEL UTILIZATION PERIODS. WE USE EITHER PRIOR DAY'S DATA OR THE SAME DAY'S DATA FOR TRAINING.

$\delta = 0.4$ $R_c = 0.8$		prior day		same day	
		FPM	PPPM	FPM	PPPM
Low	accuracy	90.78%	90.5%	90.7%	90.67%
	miss rate	7.57%	5.63%	7.2%	5.53%
High	accuracy	80.22%	79.14%	83.22%	81.48%
	miss rate	56.25%	46.12%	57.82%	42.83%
$\delta = 0.4$ $R_c = 0.7$		prior day		same day	
		FPM	PPPM	FPM	PPPM
Low	accuracy	89.14%	89.15%	89.15%	89.29%
	miss rate	1.89%	0.71%	2.06%	0.82%
High	accuracy	71.87%	71.94%	76.51%	74.87%
	miss rate	22.37%	13.37%	21.74%	9.42%

As a result, in high utilization periods, the confidence that a specific value will follow a pattern is low. We have to loose the rule confidence constraint. As shown in Table II, the miss rate can be reduced by more than 30% in high utilization periods if rules of confidence between 0.7 and 0.8 are allowed to use. This tells us that there are patterns in randomness. They are just obscure and hard to identify.

When we use one day's data to predict channel states in the next day, the spectrum usage behaviors may have changed. Therefore, we change to do 10-fold cross validation on one day's data. There is a slight improvement as demonstrated in Table II. The results show that the spectrum usage patterns are similar for the same wireless service. We do not need to perform pattern mining frequently.

In low utilization periods, the improvement of PPPM over FPM is not significant. The reason is that the beacon broadcast is rather regular in Wi-Fi networks. In other wireless services, the regularity may not be easily identified. We use USRP1 [27] with WBX to monitor channel activities in the Personal Communication Service (PCS) bands. Fig. 3 shows the channel utilization on a channel in the PCS bands for two days. It exhibits some patterns in accordance with common habits. The channel utilization is low between 01:00 and 08:00. After 10:00 the channel becomes busy. We focus on the analysis of daytime activities. Table III summarizes the prediction results. Different from Wi-Fi channels, there is no explicit periodicity of channel activities in PCS bands. Therefore, few rules have confidence of prediction higher than 0.9 and we have to loose

TABLE III
THE CHANNEL STATE PREDICTION RESULTS IN THE PCS BANDS.

rule confidence constraint R_c		0.8	0.7
FPM	accuracy	81.39%	73.35%
	miss rate	67.55%	25.26%
PPPM	accuracy	79.94%	72.31%
	miss rate	49.17%	9.99%

the rule confidence constraint. PPPM shows its advantage in discovering patterns in true traffic. The miss rate of PPPM is $16\% \sim 18\%$ lower than that of FPM.

B. DSA with prediction

We study how dynamic spectrum access can benefit from the prediction. When licensed bands are open for SUs' use, the PUs often impose a collision probability constraint η. In our PPPM-assisted DSA, we use the prediction rules mined by PPPM to estimate the collision probability. When a sequence of channel state observations match a frequent pattern, the decision of channel access is made based on the prediction. If the prediction is busy, we do not access the channel in the next slot; otherwise, we utilize the next slot if the collision probability is below the collision constraint η. We compare the performance of our PPPM-assisted DSA with an optimal DSA introduced in [25]. The optimal DSA is a statistical knowledge based access (SKA) strategy that estimates the risk of accessing the channel based on the probability density function (PDF) of idle durations $f(\cdot)$, the cumulative distribution function (CDF) of idle durations $F(\cdot)$, and the PU collision probability constraint η. Study in [28] again demonstrates that the SKA strategy improves spectrum extraction rate by 2-3 times over no knowledge based access with data measured in various bands.

In previous work, the PDF and the CDF are calculated in a global view of the entire training set. Even if we divide data into two sets of high and low utilization, the subtle variations of PDF and CDF in each data set are still concealed in the global view. In light traffic periods, the probability of having a long idle duration is larger than that in periods of slightly heavier traffic loads. For example, in Wi-Fi networks, Fig. 4 shows that 20% of the idle durations observed after pattern $\langle 000001 \rangle$ are 99 ms while most of the idle durations observed after pattern $\langle 101001 \rangle$ are much shorter. The CDF obtained in a global view only reveals the synthesized effect. In PPPM,

TABLE IV
THE SPECTRUM EXTRACTION RATE OF DIFFERENT DYNAMIC SPECTRUM ACCESS STRATEGIES UNDER DIFFERENT COLLISION CONSTRAINTS.

collision constraint η			0.1	0.2	0.3
Wi-Fi	SKA	extraction rate	20.2%	42.76%	68.45%
		collision rate	6.04%	14.06%	24.65%
	PPPM	extraction rate	70.47%	87.7%	93.19%
		collision rate	9.09%	16.34%	25.77%
PCS	SKA	extraction rate	20.22%	28.62%	39.68%
		collision rate	7.58%	13.85%	20.81%
	PPPM	extraction rate	33.8%	44.44%	55.88%
		collision rate	9.16%	15.65%	23.19%

frequent patterns are identified. The confidence of predicting '0' and '1' is no longer given based on a global view of the idle duration distribution. Instead, it is given based on observed patterns, distinguishing prediction in low risk periods from high risk periods. PPPM-assisted DSA can utilize more idle slots without increasing the collision rate as shown in Table IV. PPPM improves the spectrum extraction rate by 50% over SKA when the constraint of collision probability is tight (i.e., 10%) in Wi-Fi. The reason is that SKA is too conservative for a tight collision constraint. PPPM, however, opportunistically utilizes some slots that are dangerous in a global view but safe for use in idle periods. The significant improvement validates the importance of mining patterns if there exists regularity in spectrum usage; otherwise, even though there is sufficient underutilized spectrum, the actual number of spectrum holes that SUs can utilize are few.

VI. CONCLUSION

In this paper, a spectrum occupancy prediction model based on Partial Periodic Pattern Mining (PPPM) is introduced. The mining takes into account the irregularity of spectrum usage and thus is more suitable for true traffic. The proposed PPPM algorithm combines the gap-constrained pattern growth, the head index list structure, the Apriori-like property, and the backward-extension pruning to achieve fast and reliable partial periodic pattern mining. The partial periodicity of spectrum occupancy patterns and the performance of PPPM are validated with real life Wi-Fi network activities and data collected in the PCS bands. PPPM extracts more channel state prediction rules, leading to a significant reduction on miss rate in spectrum occupancy prediction compared with traditional FPM. We particularly observed that distinguishing low utilization periods from high utilization periods and mining rules in corresponding utilization periods will substantially improve the prediction performance. There is sufficient underutilized spectrum, but the actual number of spectrum holes that SUs can make use of are few due to PUs' tight collision constraints. We compared the PPPM-assisted DSA with a statistical knowledge based DSA to demonstrate that prediction of channel states significantly improves the spectrum extraction rate without introducing significant interference to PUs.

REFERENCES

[1] "General survey of radio frequency bands (30 MHz to 3 GHz): Vienna, Virginia, September 1-5, 2009."

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

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

[4] "FCC-10-174." [Online]. Available: http://www.fcc.gov/Daily_Releases/Daily_Business/2010/db0923/FCC-10-174A1.pdf

[5] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: a large-scale spectrum measurement study," in *Proc. ACM MobiCom*, Nov. 2009, pp. 13–24.

[6] J. Han, G. Dong, and Y. Yin, "Efficient mining of partial periodic patterns in time series database," in *Proc. of the 15th International Conference on Data Engineering (ICDE)*, 1999, pp. 106–115.

[7] Z. Wen, T. Luo, W. Xiang, S. Majhi, and Y. Ma, "Autoregressive spectrum hole prediction model for cognitive radio systems," in *Proc. IEEE ICC Workshops*, May 2008, pp. 154–157.

[8] J. Su and W. Wu, "Wireless spectrum prediction model based on time series analysis method," in *Proc. ACM CoRoNet*, Sep. 2009, pp. 61–66.

[9] C.-J. Yu, Y.-Y. He, and T.-F. Quan, "Frequency spectrum prediction method based on EMD and SVR," in *Proc. ISDA*, 2008, pp. 39–44.

[10] Z. Wang and S. Salous, "Spectrum occupancy statistics and time series models for cognitive radio," *J. of Signal Processing Systems*, Mar. 2009.

[11] I. A. Akbar and W. H. Tranter, "Dynamic spectrum allocation in cognitive radio using hidden Markov models: Poisson distributed case," in *Proc. IEEE SoutheastCon*, 2007, pp. 196–201.

[12] C.-H. Park, S.-W. Kim, S.-M. Lim, and M.-S. Song, "HMM based channel status predictor for cognitive radio," in *Proc. 2007 Asia-Pacific Microwave Conference (APMC)*, Dec. 2007, pp. 1–4.

[13] V. K. Tumuluru, P. Wang, and D. Niyato, "Channel status prediction for cognitive radio networks," *Wireless Commun. and Mobile Comput.*, Aug. 2010.

[14] C. Ghosh, C. Cordeiro, D. P. Agrawal, and M. B. Rao, "Markov chain existence and hidden Markov models in spectrum sensing," in *Proc. IEEE PerCom*, 2009, pp. 1–6.

[15] Z. Chen and R. C. Qiu, "Prediction of channel state for cognitive radio using higher-order hidden Markov model," in *Proc. IEEE SoutheastCon*, 2010, pp. 276–282.

[16] M. Lopez-Benitez and F. Casadevall, "Empirical time-dimension model of spectrum use based on a discrete-time markov chain with deterministic and stochastic duty cycle models," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 6, pp. 2519–2533, July 2011.

[17] C. Li and J. Wang, "Efficiently mining closed subsequences with gap constraints," in *Proc. of the SIAM International Conference on Data Mining (SDM)*, 2008, pp. 313–322.

[18] B. Ding, D. Lo, J. Han, and S.-C. Khoo, "Efficient mining of closed repetitive gapped subsequences from a sequence database," in *Proc. of the 25th International Conference on Data Engineering (ICDE)*, 2009.

[19] M. Zhang, B. Kao, D. W. Cheung, and K. Y. YIP, "Mining periodic patterns with gap requirement from sequences," *ACM Trans. Knowledge Discovery from Data*, vol. 1, no. 2, Aug. 2007.

[20] X. Zhu and X. Wu, "Mining complex patterns across sequences with gap requirments," in *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007, pp. 2934–2940.

[21] W. G. Aref, M. G. Elfeky, and A. K. Elmagarmid, "Incremental, online, and merge mining of partial periodic patterns in time-series databases," *IEEE Trans. Knowledge and Data Engineering*, vol. 16, no. 3, pp. 332–342, Mar 2004.

[22] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. VLDB*, 1994, pp. 487–499.

[23] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid, "Periodicity detection in time series databases," *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no. 7, Jul. 2005.

[24] M. Wellens, A. de Baynast, and P. Ma ho nen, "Exploiting historical spectrum occupancy information for adaptive spectrum sensing," in *Proc. IEEE WCNC*, 2008, pp. 717–722.

[25] S. Huang, X. Liu, and Z. Ding, "Optimal transmission strategies for dynamic spectrum access in cognitive radio networks," *IEEE Trans. Mobile Computing*, vol. 8, no. 12, pp. 1636–1648, Dec. 2009.

[26] "Anonymized packet traces and AP syslog." [Online]. Available: http://www.cs.umd.edu/projects/wifidelity/sigcomm08_traces/

[27] "Universal software radio peripheral," http://www.ettus.com/.

[28] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng, "On the feasibility of effective opportunistic spectrum access," in *Proc. IMC*, 2010.