Prediction of Channel State for Cognitive Radio Using Higher-Order Hidden Markov Model

Zhe Chen, Robert C. Qiu
Department of Electrical and Computer Engineering
Center for Manufacturing Research
Tennessee Technological University
Cookville, TN 38505, USA
Email: {zchen42, rqiu}@tntech.edu

Abstract—Spectrum sensing detects the availability of the radio frequency spectrum, which is essential and vital to cognitive radio. Traditional techniques for spectrum sensing fail to take the latency between spectrum sensing and data transmission into consideration. However, such latency does exist in hardware implementation. Prediction can be utilized to diminish the negative effect of such latency. In this paper, this latency is illustrated, and an approach for prediction of channel state using higher-order hidden Markov model (HMM) is proposed. The predicted channel states are output together with corresponding likelihood probabilities that are helpful to subsequent decision making. Wi-Fi signals have been recorded using a latest advanced ultra-performance digital phosphor oscilloscope (DPO), which are employed to evaluate the performance of the proposed approach. Experimental results show that the proposed approach for prediction of channel state is effective. The proposed approach for prediction of channel state can be used together with traditional spectrum sensing techniques for spectrum sensing with the latency taken into consideration. And it can also be utilized to provide predictive information to upper-level modules of cognitive radio.

I. INTRODUCTION

Cognitive radio (CR) has been put forward to make more efficient use of the radio frequency spectrum. It introduces “intelligence” beyond software defined radio (SDR). Spectrum sensing (SS) is the cornerstone of CR, which detects the availability of the radio frequency spectrum for secondary user (SU). The effectiveness of SS decides the efficiency of spectrum utilization of CR.

A. Spectrum Sensing, Prediction and Hidden Markov Model

There are some traditional techniques for spectrum sensing, such as energy detection, matched filter detection, cyclo-stationary feature detection, covariance-based detection, and wavelet-based detection [1], [2], [3], [4], [5]. Moreover, a multiband joint detection scheme is proposed based on energy detection in [6] for wideband spectrum sensing.

There are some related studies on prediction and hidden Markov model (HMM). The idea of predictive dynamic spectrum access has been introduced in [7], which aims at the distribution of period length of a channel being idle. Binary time series for spectrum occupancy characterization and prediction has been proposed in [8]. While multi-step-ahead prediction to control the interference time ratio has been utilized in [9]. The existence of Markov chain for sub-band utilization by primary users (PU) has been validated in [10]. HMM has been used to predict the usage behavior of a frequency band based on channel usage patterns in [11] for making the decision of moving to another frequency band or not. And a channel status predictor using HMM based pattern recognition has been proposed in [12]. However, none of the previous works takes the latency between spectrum sensing and data transmission (SSDT latency) into consideration for prediction. The SSDT latency does exist in terms of hardware implementation. Furthermore, the idea of using HMM for prediction in previous works is based on pattern recognition or classification.

B. Contribution of This Paper

Spectrum sensing slots (SS slots) and the SSDT latency are illustrated in this paper. Prediction is proposed to compensate the SSDT latency for spectrum sensing. An approach for prediction of channel state based on SS slots using higher-order HMM is introduced. The performance of the proposed approach is evaluated by the real Wi-Fi data captured by a latest advanced ultra-performance digital phosphor oscilloscope (DPO).

The whole radio frequency spectrum can be divided into multiple frequency segments in logic. And a frequency segment can be viewed to be usually consisted of multiple frequency points. The samples acquired at multiple frequency points within a frequency segment of interest are jointly considered as input to the proposed approach. By this way, prediction of channel state for multiple frequency segments can be implemented by using multiple instances of the proposed approach concurrently. Traditional training algorithms for generating parameters of higher-order HMM are not required in the proposed approach. Instead, parameters of higher-order HMM are calculated using a statistical method. The states of higher-order HMM in the proposed approach are associated with physical meanings, such as “busy”, “idle”, and “fuzzy”. The outputs of the proposed approach are predicted channel states for SS slots, as well as likelihood probabilities for each possible state. The proposed approach for prediction of channel state can work together with traditional spectrum sensing techniques to diminish the negative effect introduced by the SSDT latency. Moreover, the predictive information output by the proposed approach will be helpful for upper-level modules of CR, such as spectrum management and traffic.
control.

C. Organization of This Paper

The rest of this paper is organized as follows. Section II formulates the problem. Section III introduces the proposed approach for channel prediction using higher-order HMM. Section IV reports the performance evaluation of the proposed approach. And Section V concludes this paper.

II. PROBLEM FORMULATION

In this section, a scenario of spectrum sensing is described. And SS slot and the SSDT latency are introduced.

A. Spectrum Sensing Slots

As shown in Fig. 1, an SU communicates with another SU or a secondary base station (SBS) through full-duplex wireless channels. Both uplink channel and downlink channel are comprised of a series of SS slots. Assume the time length of the SS slots are constant. Each SS slot has two phases, i.e., sensing phase (the first phase) and transmission phase (the second phase). During an SS slot, an SU or SBS senses the availability of the channel in sensing phase, then it may start data transmission in transmission phase if the channel is available.

In [13], acknowledgment (ACK) and negative acknowledgment (NAK) messages are employed to indicate whether the transmission is successful or not. The idea is borrowed here. In this paper, it is proposed that ACK or NAK messages are sent from receiver to sender together with other kinds of data in the transmission phase of SS slots, as shown in Fig. 1. Since it takes time for the receiver to process received data and send ACK or NAK messages to the sender, there is a latency for the sender between sending data to the receiver and receiving the corresponding ACK or NAK message from the receiver.

B. Latency between Spectrum Sensing and Data Transmission

A simplified data path of SU is shown in Fig. 2. Received radio frequency signals are amplified and down-converted before they are converted into digital signals by analog-to-digital converter (A/D). Then the digital signals are fed to data processing module where digital signal processing is performed, through a digital data interface. The down-conversion module is optional. In [14], down-conversion is employed. While in [15], wideband spectrum is measured directly without down-conversion.

From hardware implementation point of view, latencies exist between A and B, and C and D, which are shown in Fig. 2. Take the small form factor (SFF) software defined radio development platform of Texas Instruments (TI) for example. There is an interface called Video Processing Front End (VPFE) in the data interface module between A and B in Fig. 2. VPFE introduces a latency of 61-79 µs. And there is an interface called Video Processing Front End (VPBE) in the data interface module between C and D. VPBE introduces a latency of 70-186 µs [16]. Since data processing takes time, there is also a latency between B and C. Fig. 3 illustrates the latencies. The latencies between A and B, B and C, and C and D are denoted as $t_{rl}$, $t_{pl}$, and $t_{tl}$, respectively. Then the total latency $t_l$ is

$$t_l = t_{rl} + t_{pl} + t_{tl} \quad (1)$$

The total latency is denoted as SSDT latency in this paper. If the SSDT latency is comparable with or greater than the time length of an SS slot, it can not be omitted. Fig. 3 shows the case that the SSDT latency equals to the time length of one SS slot. In this case, using spectrum sensing techniques without consideration of the SSDT latency, the result of spectrum sensing in the sensing phase of SS slot 0 will be applied to the transmission phase of SS slot 1 implicitly, which is
unexpected. If the real channel state changed between SS slot 0 and SS slot 1, a detection error of spectrum sensing would occur. In order to fill in the gap caused by the SSDT latency and reduce detection error rates of spectrum sensing, it is proposed in this paper to use the data obtained from the sensing phase of SS slot 0 and those from prior SS slots to predict the availability of channel in SS slot 1.

III. HIGHER-ORDER HIDDEN MARKOV MODEL

In [17], first-order HMM has been employed to predict channel state for spectrum sensing. A major limitation of first-order HMM is that a state only depends on one immediate previous state. Thus, information of historical states is not thoroughly exploited for prediction of future states. In this paper, prediction using higher-order HMM is proposed to make a better use of the information of historical states that contains abundant hints for prediction of future states.

Higher-order HMM generalizes first-order HMM by extending the dependency from one immediate previous state to $R$ states, which is defined by a tuple $\lambda = \{A, B\}$ [18], [19], [20]. $A$ is the state transition probability matrix with $R + 1$ dimensions,

$$(A)_{i_1,i_2,...,i_{R+1}} = a_{i_1i_2...i_{R+1}} = \Pr(q_{t+1} = \theta_{i_{R+1}}|q_t = \theta_{i_1},...,q_{t-R+2} = \theta_{i_2},q_{t-R+1} = \theta_{i_1},i_1,i_2,...,i_{R+1} = 1,2,...,N)$$

(2)

where $(\bullet)_{i_1,i_2,...,i_{R+1}}$ denotes the entry of the matrix with indexes of $i_1,i_2,...,i_{R+1}$, and $\Pr(\bullet)$ means probability. $R$ is the order of higher-order HMM, $N$ is the number of states of Markov chain, $\{\theta_1,...,\theta_N\}$ are the $N$ states, $q_t$ represents the state at time $t$, $q_t \in \{\theta_1,...,\theta_N\}$. And $B$ is the emission probability matrix,

$$(B)_{j,k} = b_{jk} = \Pr(o_t = v_k|q_t = \theta_j)$$

(3)

where $M$ is the number of possible observation values in the observation space $\{v_1,...,v_M\}$, $o_t$ represents the observed value at time $t$, $o_t \in \{v_1,...,v_M\}$.

Given a parameter tuple $\lambda$ and a sequence of observed values $o = \{o_1,...,o_T\}$, the state sequence that is most likely to have generated the input sequence $o$ and the likelihood can be calculated using the following algorithm [18], [19], [20], which is denoted as HO-HMM algorithm in this paper.

1) Initialization. For each $t = 2,3,...,R$ and $i_1,i_2,...,i_R = 1,2,...,N$, calculate

$$\delta_1(i_1) = a_{0i_1}b_{i_1}(o_1)$$

(4)

$$\delta_t(i_1,i_2,...,i_t) = \delta_{t-1}(i_1,i_2,...,i_{t-1})a_{i_ti_{t+1}}b_{i_{t+1}}(o_t)$$

(5)

$$\varphi_1(i_1,i_2,...,i_R) = 0$$

(6)

2) Iteration. For each $t = R + 1, R + 2,...,T - 1, T$ and $i_2,i_3,...,i_{R+1} = 1,2,...,N$, calculate

$$\delta_t(i_2,i_3,...,i_{R+1}) = \max_{i_1}\{\delta_{t-1}(i_1,i_2,...,i_{R+1})a_{i_1i_2...i_{R+1}}b_{i_{R+1}}(o_t)\}$$

(7)

$$\varphi_t(i_2,i_3,...,i_{R+1}) = \arg\max_{i_1}\{\delta_{t-1}(i_1,i_2,...,i_{R+1})a_{i_1i_2...i_{R+1}}\}$$

(8)

3) Termination.

$$P^* = \max_{i_1,i_2,...,i_R}\{\delta_T(i_1,i_2,...,i_R)\}$$

(9)

$$q_t^* = \varphi_t(i_1,i_2,...,i_R)$$

(10)

4) Backtraсing. For $t = T - R, T - R - 1,...,2,1$, look up

$q_t^* = \varphi_{t+1}(q_{t+1}^*,q_{t+2}^*,...,q_T^*)$.

where $P^*$ is the calculated likelihood probability and $q_T^*$ is the estimated state at time $T$.

As the order $R$ goes higher, the complexity of the HO-HMM algorithm increases sharply. In order to utilize the information from more prior states and to reduce computational complexity, an algorithm named AA-HMM is proposed in this paper as follows. It derives from the Viterbi algorithm for first-order HMM [20].

1) Initialization.

$$\hat{a}_{i_Ri_{R+1}} = a_{0i_Ri_{R+1}}$$

$$\hat{a}_{i_{R+1}} = 1,2,...,N$$

(12)

$$\delta_1(i_{R+1}) = \hat{a}_{i_0i_{R+1}}b_{i_{R+1}}(o_1)$$

(13)

2) Iteration. For $i_{R+1} = 1,2,...,N$ and $t = 2,3,...,T$, calculate

$$\delta_t(i_{R+1}) = \max_{1 \leq i_R \leq N}\{\delta_{t-1}(i_R)a_{i_Ri_{R+1}}b_{i_{R+1}}(o_t)\}$$

(14)

$$\varphi_t(i_{R+1}) = \arg\max_{1 \leq i_{R+1} \leq N}\{\delta_{t-1}(i_R)a_{i_Ri_{R+1}}\}$$

(15)

3) Termination.

$$P^* = \max_{1 \leq i_{R+1} \leq N}\{\delta_T(i_{R+1})\}$$

(16)

$$q_t^* = \varphi_{t+1}(q_{t+1}^*)$$

(17)
IV. PREDICTION OF CHANNEL STATE USING HIGHER-ORDER HIDDEN MARKOV MODEL

In spectrum sensing, what SUs concern is the availability of channel (or channel state) within some frequency segments of interest. An architecture for prediction of channel state using higher-order HMM is proposed, which is shown in Fig. 4. In the sensing phase of every SS slot, received digital signals are transformed into frequency domain using fast Fourier transform (FFT). Then multiple frequency points within a frequency segment of interest are then quantified and fed to higher-order HMM as an observed value. Multiple frequency segments of interest can be processed in parallel using multiple higher-order HMMs.

Fig. 5 illustrates observed values (denoted as $o_1, o_{i-1}, o_{i-2}, \ldots$) and channel states (denoted as $q_{t+2}, \ldots, q_{i-2}, \ldots$). At the end of sensing phase of every SS slot, an observed value is obtained and input to higher-order HMM. During each SS slot, channel is in a certain state. This state can be denoted as “busy”, “idle”, or “fuzzy”. How to obtain actual channel states for each SS slot is an open problem for discussion. One approach is to utilize the ACK/NAK message sent by other SU or SBS. For example, if an ACK message for a certain SS slot is received, then this SS slot can be marked as “idle”, since other SU or SBS has successfully received the data transmitted during this SS slot. Note that there is also a latency between the data transmission in that SS slot and the reception of the ACK/NAK message for that SS slot. In this paper, the value of this latency is denoted as $Y$, in the unit of SS slots.

The task of prediction of channel state is using observed values and prior channel states to predict future channel states. Higher-order HMM is employed for prediction. Channel state is directly mapped to the state of higher-order HMM. And observed values are used as inputs to higher-order HMM. The outputs of higher-order HMM are predicted channel state as well as its corresponding likelihood probability. Here, $A$ and $B$ of higher-order HMM are defined as below.

\[
(A)_{i_1,i_2,\ldots,i_R+1} = a_{i_1,i_2,\ldots,i_R+1} = \Pr(q_{t+1}=(X+Y) = \theta_{i_{R+1}}|q_{t} = \theta_{i_{R}}, \ldots, q_{t-(R-1)}=(X+Y) = \theta_{i_1}) \tag{18}
\]

\[
(B)_{j,k} = b_{j,k} = \Pr(o_t = v_k|q_{t+X} = \theta_j) \tag{19}
\]

where $X$ is the prediction span, in the unit of SS slots.

Parameters of higher-order HMM, i.e., $A$ and $B$, can be obtained by a statistics method. The proposed equations are listed as follows.

\[
\alpha_{i_1,i_2,\ldots,i_{R+1}}^{L-(X+Y)} = \sum_{t=1+(R-1)\times(X+Y)}^{L-(X+Y)} \text{Eval}(q_{t+(X+Y)} = \theta_{i_{R+1}}|q_{t} = \theta_{i_{R}}, \ldots, q_{t-(R-1)}=(X+Y) = \theta_{i_1}) \tag{20}
\]

\[
b_{j,k} = \frac{\sum_{t=X+1}^{L-X} \text{Eval}(o_t = v_k|q_{t+X} = \theta_j)}{\sum_{t=X+1}^{L-X} \text{Eval}(q_{t} = \theta_j)} \tag{21}
\]

where $L$ is the length of training sequence, and

\[
\text{Eval}(\bullet) = \begin{cases} 1 & (\bullet) \text{stands} \\ 0 & \text{otherwise} \end{cases} \tag{22}
\]

With known parameters of higher-order HMM and observed values, predicted channel state $q_T^*$ and corresponding likelihood probability $P^*$ can be calculated either by the HO-HMM algorithm or by the AA-HMM algorithm. $T$ is set to 1.

There are some options for using higher-order HMM, which should be considered in practical usage.

a) Offline Training or Online Adaptive Training: The process of obtaining the parameters of higher-order HMM is called training. Training can be done either offline or online. Offline training means that parameters are pre-calculated before they come into use and they usually do not vary after they are calculated. However, sometimes training sequences are hard to be obtained beforehand. And sometimes the environment changes a lot with time. Thus pre-calculated parameters may not fit anymore. Online adaptive training do not need training sequences in advance. It updates the parameters of higher-order HMM in real-time when higher-order HMM works. Although it takes some time to make the parameters stable, it can obtain the most suitable parameters adaptively.
b) 2 States or 3 States: Physical meanings are associated with the states of higher-order HMM in this paper. Since usually the availability of channel is represented as “busy” (unavailable for SU) and “idle” (available for SU), a 2-state higher-order HMM can be defined. Another state named “fuzzy” (unknown availability) is proposed as a medium state. With this additional state, a 3-state higher-order HMM can be defined. The “fuzzy” state is introduced to feed a dedicated state to higher-order HMM while actual channel availability is unknown. Moreover, using “fuzzy” state can help reduce the interference to PU, since in the unknown state SU is unlikely to access the channel. However, training of 3-state higher-order HMM needs more training data to figure out more parameters.

c) Scalar Quantization or Vector Quantization: Data from one or more adjacent frequency point are quantized into $M$ levels and then fed to higher-order HMM as observed values. The quantization before HMM in Fig. 4 can be either scalar quantization (SQ) or vector quantization (VQ). SQ quantifies the data from only the center frequency point of a frequency segment of interest into $M$ integers. While VQ quantifies the data from all the frequency points of a frequency segment of interest into $M$ integers. Linde-Buzo-Gray (LBG) algorithm can be used to design the codebook for VQ [20].

d) Last States Information: Last states refer to the past states just before current state. Using last states information (LSI) is an option for the HO-HMM algorithm. The algorithm described by (4) (5) (6) (7) (8) (9) (10) (11) is denoted as the HO-HMM algorithm without LSI, since in the initialization phase no information of last states is utilized. With (4) (5) (6) simplified by (23), LSI can be used for the HO-HMM algorithm.

\[
\delta_t(i_1, i_2, \ldots, i_R) = \begin{cases} 
1 & i_1 = q_1, i_2 = q_2, \ldots, i_R = q_R \\
0 & \text{Otherwise} 
\end{cases}
\]

(23)

where $q_1, q_2, \ldots, q_R$ denote the last $R$ states. The AA-HMM algorithm presented by (12) (13) (14) (15) (16) (17) takes advantage of LSI. Without LSI, the order of higher-order HMM will not affect the AA-HMM algorithm.

V. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed approach, Wi-Fi is employed as PU. In the experiment, the proposed approach is used to predict channel states for SU with the SSDT latency taken into consideration.

A. Wi-Fi Signal Measurement

Wi-Fi time-domain signals have been measured and recorded using a latest advanced ultra-performance DPO whose model is Tektronix DPO72004. It supports a maximum bandwidth of 20 GHz and a maximum sampling rate of 50 GS/s. In the measurement, a laptop accesses the Internet through a wireless Wi-Fi router. An antenna with a frequency range of 800MHz to 2500MHz is placed near the laptop and connected to DPO. The sampling rate of the DPO is set to 6.25 GS/s. Recorded Wi-Fi time-domain signals are shown in Fig. 6. The length of SS slot of SU is set to 2 $\mu$s and the length of sensing phase is set to 0.4 $\mu$s. The recorded time-domain signals within the sensing phase of every SS slot are transformed into frequency domain using FFT. For each SS slot, an FFT shot is obtained. Fig. 7 shows a part of the obtained FFT shots.

B. Performance Evaluation

Each FFT shot is associated with a time index. Data from one or multiple frequency points in an FFT shot are quantified into an observed value for higher-order HMM. In the experiment, the frequency segment of interest is centered at 2.422 GHz with a bandwidth of 5 MHz. Higher-order HMM is trained using online adaptive training. Let $M = 128$ and $Y = 1$. Rates of mis-detection and false alarm are employed as metrics for performance evaluation. Since both mis-detection rate and false alarm rate are expected to be minimal for spectrum sensing, average error rate (AER) is introduced.
Fig. 8. Error rates of prediction using adaptive training for higher-order HMM.

Fig. 9. Error rates of prediction using 2-state higher-order HMM.

Fig. 10. Error rates of prediction using 3-state higher-order HMM.

where $q_t$ denotes the predicted state at time $t$, $p_t$ denotes the actual state at time $t$, $I$ represents the “idle” state, $B$ represents the “busy” state, and $L'$ is the length of testing sequence. Thus the range of AER is $[0, 1]$.

Fig. 8 shows an example using adaptive training for higher-order HMM. The parameters for this example are set as follows: $X = 1$, $R = 1$, vector quantization and 3 states. It can be observed that the error rates converge and the AER gets close to convergence in few hundreds of shots. Fig. 9 and Fig. 10 show the AER of prediction with different combinations of options. Nearest neighbor prediction, which refers to the prediction that current channel state is predicted directly using the last available channel state, is used as a benchmarking approach. In these two figures, $R = 1$. It can be observed from the figures that the performance of the proposed approach is much better than that of nearest neighbor prediction, especially when prediction span increases, either in the case of 2 states or in the case of 3 states. In this case, among these combinations of options, the HO-HMM algorithm with vector quantization and without LSI gets the lowest AER.

Fig. 11 and Fig. 12 show the AER of prediction using the HO-HMM algorithm.

Given that $q_t$ denotes the predicted state at time $t$, $p_t$ denotes the actual state at time $t$, $I$ represents the “idle” state, $B$ represents the “busy” state, and $L'$ is the length of testing sequence, the Average Error Rate (AER) can be defined as:

$$AER = \frac{\sum_{t=1}^{L'} \text{Eval}(q_t = I | p_t = B) + \sum_{t=1}^{L'} \text{Eval}(q_t = B | p_t = I)}{\sum_{t=1}^{L'} \text{Eval}(p_t = B) + \sum_{t=1}^{L'} \text{Eval}(p_t = I)}$$

(24)
orders of higher-order HMM. The options for these two figures are set to vector quantization and 2 states with LSI. It can be observed that the AER decreases with the order of higher-order HMM increases and that the AER of the HO-HMM algorithm is slightly lower than that of the AA-HMM algorithm.

VI. Conclusion

The spectrum sensing slots and the latency between spectrum sensing and data transmission have been illustrated in this paper. In order to diminish the negative effect of this latency, an approach for prediction of channel state based on spectrum sensing slots using higher-order HMM has been proposed. Options for the proposed approach have been put forward as well. Performance evaluation using real Wi-Fi signals recorded by a latest advanced ultra-performance digital phosphor oscilloscope shows that the performance of the proposed approach is much better than that of nearest neighbor prediction, especially when the order of higher-order HMM increases.

Acknowledgment

This work is funded by National Science Foundation through grants (ECCS-0901420), (ECCS-0821658), and (ECCS-0622125), and Office of Naval Research through a contract (N00014-07-1-0529).

References