Spectrum Occupancy State Predictor Based on Recurrent Neural Network

Rong Fan¹, Hesong Guo¹, Lujie Di², and Xing Ling²

¹Aviation Engineering Institute, Civil Aviation Flight University of China, Guanghan, Sichuan Province, 618307, China.
²School of Information and Communication Engineering University of Electronic Science and Technology of China, Chengdu, Province, China.

Corresponding author’s e-mail: fanrong@cafuc.edu.cn

Abstract. To cope with the scarcity of spectrum resources and to improve the efficiency of spectrum utilization, spectrum occupancy prediction (or named channel state prediction) technology has been addressed into cognitive radio (CR) in recent years. In the paper, we first address the issue how to model the primary user behaviour in CR. And based on the presented behaviour model, a recurrent neural network (RNN) is chosen to design a spectrum occupancy state predictor. By exploiting the learning capacity of recurrent neural network, it can predict the spectrum occupancy state of primary user behaviour. Meanwhile, in order to reveal the proposed predictor’s advantages, the spectrum occupancy state predictor based on RNN and another existed two spectrum occupancy state predictors are also adapted in the simulation section of the paper. Numerical simulation results illustrate the advantages of the proposed method.

1. Introduction

Cognitive radio (CR) is a content-aware smart radio that based on the software radio platform, which can realize self-reconfiguration through learning the varieties of the wireless communication environment. However, as one of the significant technology in CR, spectrum occupancy state prediction has been addressed for decades. And it has been used to alleviate the lack of spectrum resource, dynamically manage the spectrum and improve the spectrum utilization.

Up to now, many spectrum prediction technologies based on machine learning, especially those based on neural network, have becoming research hotspots [1-2]. In CR, a variety of neural network (NN) is used in spectrum sensing [3], dynamic channel selection [4], channel sensing and learning etc [5-7]. In [8], the NN based on forward propagation is used as a channel state predictor. However, it cannot be used the scenario that multiple primary users interact information with each other. Besides, the channel state prediction based on the hidden Markov model (HMM) used in [9] assumed that primary user behaviour is subject to Poisson distribution. Another spectrum prediction algorithm is also proposed in [10] and interested readers can refer to it. Different form the previous literatures, the learning ability of NN is focused in the paper. We first model the primary user behaviour which is used to as approximate real scenario as possible. And then multi-channel state predictor based on recurrent neural network (RNN) is devised. As RNN can share the model parameters at each time slot, it enables the prediction model can remember the historical information. And thus we can use the...
channel state historical information of each channel to train the NN parameters. And finally, the spectrum evolution is performed.

| AB   | State meaning                                      |
|------|---------------------------------------------------|
| 00   | Both are idle                                     |
| 01   | User A idle and user B busy (B sends information to A) |
| 10   | User A busy and user B idle (A sends information to B) |
| 11   | Duplex status (user A and user B send information to each other at the same time) |

The rest of this paper is organized as follows. The behaviour model of primary users is discussed in section 2. The basic RNN model is described in section 3 and the spectrum predictor for multi-channels is presented in section 4. Simulation results and conclusions are presented in section 5 and section 6, respectively.

2. Primary User Behavior Model

Considering a primary user system with $M$ channels, where $m \in \{1, 2, \ldots, M\}$ denotes the $m$-th channel, the band width is $B$ Hz for each channel. It is assumed that the primary user channels are logically divided into mutually independent. The time slots that are not occupied by primary users are called spectrum holes or idle spectrum [1]. If the channel is sensed idle for a certain period of time, the secondary users can access the channel. The channel state in one time slot can be expressed as idle or busy according to whether primary users active or not. Meanwhile, we assume that the channel used by each primary user is fixed and the behaviour of the primary user on a channel is considered to obey the Hidden Markov Model (HMM). Therefore, when the primary user starts to occupy the channel, the time duration obeys the exponential distribution. To make the simulation close to the real environment, the primary users can interact information with each other. For example, the primary user A and primary user B are assumed as a pair of interaction objects. As shown in Table 1, there are four interaction states. $10 \rightarrow 01$ indicates after the user A transmit the information to the user B, and user B responds to user A; $10 \rightarrow 00$ indicates that user A has completed the information transmission to user B. Obviously, Markov state transition probability matrix can be used to describe this kind of state transition. In this way, different state transition probability matrices represent the intra-group user interactions of different behaviours. As a matter of fact, the primary user behaviour model generated by the Markov state transition probability matrix is reasonable, which complies with the common law of primary user channel usage behaviour. And thus a binary primary sequence can be expressed as

$$c_{-\text{sta}_m} = \{c_{-\text{sta}_m}(1), \ldots, c_{-\text{sta}_m}(t), \ldots, c_{-\text{sta}_m}(T)\} \quad (1)$$

Where $c_{-\text{sta}_m}(t)$ denotes the primary user occupancy state at the $m$-th channel in the $t$-th time slot. Expressions $c_{-\text{sta}_m}(t) = 0$ (or $c_{-\text{sta}_m}(t) = 1$) represent the $m$-th channel is idle (or busy) in the $t$-th time slot.

3. Recurrent Neural Network Model

RNN is different with the normal neural network which shares the model parameters at each time step. The mechanism of the RNN enables the model can adapt samples of different length and generation. RNN share the statistical strength of different length or different time step in the time dimension and the training data that RNN needs is less than the normal neural network model. Figure 1 shows the calculation graph of classic RNN. The update formula for the forward propagation of the RNN is

$$h_t = f^{(h)}(b + Wh_{t-1} + Ux_t)$$

$$\hat{y}_t = f^{(y)}(c + Vh_t) \quad (2)$$
Where $f^{(h)}$ and $f^{(y)}$ denotes the action function of hidden layer and output layer, respectively. The biases $b$ and $c$, weight matrix $U$, $V$ and $W$ denote the connection from input to hidden, hidden to output and hidden to hidden, respectively. The total loss between $y$ and the input sequence $x$ is the sum of the losses for all time steps. And RNN reaches the best parameters by minimize the loss. A big mathematical challenge is learning the long-term dependence of RNN, like the problems of gradient disappearance and gradient explosion. In real applications, gated RNN is a good method to solve these problems. It includes long short term memory (LSTM) and gated recurrent unit (GRU) [11]. Figure 2 represents the structure of GRU.

![Figure 1. The calculation graph of classic RNN.](image)

![Figure 2. The structure of GRU.](image)

4. Spectrum Occupancy State Predictor

To predict the future channel state, the historical channel states is used. As the spectrum states are not independent and have a certain time correlation and the interaction of each primary user is similar with the time correlation which is consistent with the features of the RNN. We use the binary sequence generated by primary user behavior model to represent the historical information of the primary user state in the channel and use this as input. The final output of the recurrent network is the channel state at the next time slot. Because all primary users have interaction with each other, we combine the channel occupancy state information of multiple primary users to make a unified prediction. Figure 3 shows the structure of the RNN proposed in this paper.

Multiple channel occupancy state information at the same time slot is feed into the network as a time step, historical occupancy state information for previous $L$ time slots is used to predict the channel state at time slot $L+1$, and then use the predict channel state at $L+1$ time slot and the previous $L$-1 historical information predict the channel state at time $L+2$, and so on, until the required length of time slot is predicted. The output layer of the spectrum occupancy state predictor uses the sigmoid activation function

$$out(x) = \frac{1}{1 + e^{-x}}$$

(3)

![Figure 3. Multi-primary user channel state structure.](image)

Table 2 gives the parameters of the RNN used in the paper. Since the output layer uses sigmoid function as activation function, the output of the neural network is a value range from 0 to 1. So we set a decision threshold $l$. If the output is greater than the threshold, we believe that the predicted time slot is occupied by primary user. Otherwise, the predicted time slot is idle. For the secondary user, the
spectrum occupancy state predictor is used to obtain the status of the next time slot of the channel that the primary user possible to use, and a channel whose idle duration is consistent with the time required by service of the secondary user is selected to transmit data.

Table 2. Parameters of spectrum occupancy state predictor.

| Hidden layer | Neural unit | Learning rate $\eta$ |
|--------------|-------------|---------------------|
| 1            | 40          | 0.001               |

In order to verify the performance of the spectrum occupancy state predictor, we introduce detection probability and false alarm probability. (a) Detection probability $P_d$: predictor predicts the time slot is busy and the actual state is busy. (b) False alarm probability $P_f(Idle)$: predictor predicts the time slot is busy but the actual state is idle. (c) Missed detection probability $P_f(Busy)$: predictor predicts the time slot is idle but the actual state is busy. The missed detection probability can be represented as

$$P_f(Busy) = 1 - P_d$$

(4)

it is noted that $P_f(Idle)$ is an important measurement standard of spectrum occupancy state predictor. If $P_f(Idle)$ increased, secondary users have less opportunity to use idle channels. On the other hand, high $P_d$ can minimize secondary users’ interference with primary users. When $P_f(Idle)$ is low and $P_d$ is high, the channel state predictor performance will be nice. The ideal situation is $P_f(Idle) = 0$ and $P_d = 1$.

Table 3. Performance comparison.

| Method     | Accuracy | $P_f(Busy)$ | $P_f(Idle)$ |
|------------|----------|-------------|-------------|
| Proposed method | 85.25% | 13.57% | 16.20% |
| KRLS$^{[12]}$ | 81.5% | 16.5% | 21% |
| HMM$^{[13]}$ | 82.56% | 16.3% | 18.75% |

5. Simulation and Results

The simulated primary user behavior data is generated with the model described in Section 2. We divide the data into two categories: (a) Training data, used to train the model, in order to determine weights and bias of the RNN; (b) Test data, used to test the performance of predictor. We suppose that there are four primary users active. And thus there are 16 interaction modes in simulation.

Figure 4 presents the transitions probability among modes. It is considered that the same interaction mode is more likely to continue in the next time slot, so the probability of the diagonal of the graph is larger than others. Figure 5 shows the occupancy of the channels used by the primary users, where white and black indicate that the time slot is idle and occupied by the primary user, respectively.

In the multi-channel state predictor based on RNN, the parameters of the neural network are shown in Table 1 and we chose GRU as the RNN cell. The weights and biases of the neural network are randomly initiated using the random function and the value range from -1 to 1.

Figure 6 shows the prediction accuracy for different time steps and different prediction step length. From the result, we can observe that with the increase of time step, better performance of the proposed multi-state predictor can be achieved. However, prediction performance does not continue to increase while the time step reaches a certain length and even decreases. This is probably because for certain length of time step, the historical information has strong correlation with the channel state at next
moment, and too long time step cannot provide useful information to predict or even become noise. So, along with the length of prediction increasing, the accuracy of prediction is decreased. That’s to say, the more steps length of prediction means more unpredictability.

Figure 4. Four-user state transition probability map.

Figure 5. Four-user state transition probability map.

Figure 6. Accuracy of different time step and prediction length.

6. Conclusion
In this paper, we first model a primary user behavior with HMM state transfer matrix. And then, a RNN is chosen to design a spectrum occupancy state predictor. By exploiting the learning capacity of recurrent neural network, it can predict the spectrum occupancy state of primary user behaviour. The
Simulation results show that our proposed spectrum predictor can predict the channel state for a period of time in the future. With this method, it has better performance than other existed algorithms and can provide more sufficient information to the secondary users to utilize spectrum holes.

Acknowledgment
The author would like to thank the anonymous reviews for their comments that helped to improve the quality of the paper. The work was supported by the projects on the Research Foundation of the Civil Aviation Flight University of China for Advanced Talents under Grant J2019-002.

References
[1] Das, D., Matolak, D.W., Das, S. (2018) Spectrum occupancy prediction based on functional link artificial neural network (FLANN) in ISM band. Neural Comput. Appl. 29: 1363-1376.
[2] Jian, Y., Zhao, S. (2015) Enhanced throughput of cognitive radio networks by imperfect spectrum prediction. IEEE Commun. Lett. 19: 1738-1741.
[3] Liu, C., Liu X. and Liang, Y. (2019) Deep CNN for Spectrum Sensing in Cognitive Radio. In: IEEE International Conference on Communications. Shanghai. pp. 1-6.
[4] Jia, L., Xu, Y., Zhang Y. and et al. (2017) A Distributed Anti-jamming Channel Selection Algorithm for Interference Mitigation-based Wireless Networks. In: IEEE 17th International Conference on Communication Technology. Chengdu. pp. 151-155.
[5] Baldo, N., Tamma, B.R., Manoj, B.S. and et al. (2009) A Neural Network Based Cognitive Controller for Dynamic Channel Selection. In: IEEE International Conference on Communications. Dresden. pp. 1-5.
[6] Tubachi, S., Venkatesan M., A. V. Kulkarni. (2017) Predictive Learning Model in Cognitive Radio Using Reinforcement Learning. In: IEEE International Conference on Power, Control, Signals and Instrumentation Engineering. Chennai. pp. 564-567.
[7] Morozs, N., Clarke T., Grace, D. (2016) Distributed heuristically accelerated q-learning for robust cognitive spectrum management in LTE cellular systems. (2016) IEEE Trans. Mobile Comput. 15: 817-825.
[8] Eltom, H., Kandeepan, S., Evans, R. J. and et al. (2018) Spectrum Occupancy Prediction Using a Hidden Markov Model. In: 9th International Conference on Signal Processing and Communication Systems. Cairns. pp.1-8.
[9] Tarek, D., Benslimane A., Darwish M. and et al. (2019) Cognitive Radio Networks Channel State Estimation Using Machine Learning Techniques. In: 15th International Wireless Communications & Mobile Computing Conference. Tangier. pp. 342-347.
[10] Goodfellow, I., Bengio, Y., Courville A. (2016) Deep learning. Vol. 1. Cambridge: MIT Press.
[11] Le, T., Kim J., Kim, H. (2016) Classification Performance Using Gated Recurrent Unit Recurrent Neural Network on Energy Disaggregation. In: International Conference on Machine Learning and Cybernetics. Jeju. pp. 105-110.
[12] Hainmueller, J., Hazlett, C. (2014) Kernel regularized least squares: reducing misspecification bias with a flexible and interpretable machine learning approach [J]. Polit. Anal. 22:143-168.
[13] Eltom, H., Kandeepan, S., Liang, Y.C. et al. (2018) Cooperative soft fusion for HMM-based spectrum occupancy prediction. IEEE Commun. Lett. 22: 2144-2147.