Sensing-Transmission Tradeoff for Multimedia Transmission in Cognitive Radio Networks

Xiang Sheng and Shaowei Wang
School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
Email: 151180109@smail.nju.edu.cn, wangsw@nju.edu.cn

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322250

***Abstract*—Efficient probing spectrum holes is one of the most challenging tasks for the secondary user (SU) in a cognitive radio (CR) network. In this paper, we introduce a novel spectrum sensing framework where the duration for sensing at each time slot is variable. Sensing more channels increases the probability of finding a spectral hole, however, it would spend more time for sensing inevitably, which reduces the time for data transmission at a given time slot. Considering the sensing-transmission tradeoff, the optimization goal of spectrum sensing strategy is set to maximize the expected achievable throughput of the SU, which is formulated as a partially observable Markov decision process (POMDP). Finding an optimal solution to this optimization problem is computationally expensive due to its large state space, as well as large action space. We develop a novel spectrum sensing strategy based on deep reinforcement learning, which converges fast and can deal with complex scenario. Numerical results show that our proposed strategy can improve system throughput significantly.**

***Index Terms*—Cognitive radio, deep reinforcement learning, sensing-transmission tradeoff.**

I. Introduction

Cognitive radio (CR), which can significantly improve spectrum utilization, has been widely studied in the past two decades [1]–[3]. The quality of service of multimedia transmissions in CR networks can be maximized by increasing the system throughput. In CR networks, the secondary users (SUs) are expected to exploit the licensed spectrum opportunistically without interfering with the licensed primary users (PUs). During its operation, the SU is required to execute spectrum sensing to detect active PUs. However, spectrum sensing is generally time-consuming due to hardware limitations. Considering a time-slotted CR network, it is difficult for the SU to sense all licensed spectrum at a given time slot. With a bigger fraction for sensing, more spectrum can be sensed, which increases the probability of finding a spectrum hole. However, less time remains for data transmission, which limits the achievable throughput of this time slot. As a result, the sensing-transmission tradeoff problem, which refers to how to divide a time slot into two parts for sensing and transmission, is important for the SU [4], [5]. In practice, the optimal sensing duration depends on the channel state transition characteristics that are unknown to the SU, which is a challenge for the design of efficient spectrum sensing strategies.

This work was partially supported by the National Natural Science Foundation of China (61671233,61931023,U1936202)

Spectrum sensing problem has been widely studied in the literature. The conventional model-based methods are devised from pure mathematical PU traffic models with artificial assumptions [6], [7]. In contrast, the novel learning-based strategies are directly built on the data generated by the CR network, which make them adaptable to more complex real-world models. In [8], an online learning method is employed to address the spectrum sensing problem in the scenario that the licensed spectrum spans multiple service providers. In [9], reinforcement learning is used to develop an Aloha-like spectrum sensing scheme for multi-user CR network, and its performance is evaluated in small state space case. However, for large state space and partial observability, reinforcement learning is inefficient. With the demand for a more powerful learning-based spectrum sensing algorithm, the use of deep reinforcement learning has been investigated in recent years. In [10], the authors consider the spectrum sensing problem in a CR network with multiple correlated channels, and formulate it as a partially observable Markov decision process (POMDP). They implemented a deep Q-network and showed that it can achieve near-optimal performance in complex scenarios. In [11], a deep actor-critic reinforcement learning based framework was proposed for the multi-user spectrum sensing problem. The proposed framework was simulated in channel correlated scenario which demonstrated its ability of handling large action spaces.

The aforementioned methods focus on fixed time slot structure, i.e., the SU senses one channel at a given time slot. In practice, the performance upper bound of these methods is limited by the fixed time slot structure, e.g., it is better to sense two channels instead of one at a given time slot if channels are highly likely to be busy [12]. Moreover, the reward of the SU in these methods is either a success or a failure due to the fixed time slot structure, resulting in the sparsity of the reward. Note that spectrum sensing with adaptive time slot structure can efficiently handle the above-mentioned problems, thereby highly improving system performance. In [13], the authors look into the problem of deriving the optimal sensing duration in CR network where the idle times of the PUs obey hyperexponentially distribution with known parameters. In [14], the authors focus on how to improve system throughput and energy efficiency by skipping channel sensing for a time slot or more, which triggers a tradeoff between the interference caused to the PUs and the benefit obtained. However, the time

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

slot structure in [13], [14] is model-based adaptive, which means that the sensing duration is adjusted based on the solution of predefined optimization problem. In practice, the SU only has a partial view of the whole CR network, which makes the computation of optimal solutions intractable in general. These facts motivate us to set up a reasonable adaptive time slot structure and develop efficient learning-based method to get a better tradeoff between sensing and transmission.

In this paper, we investigate spectrum sensing problem with adaptive time slot structure for system throughput maximization in CR networks. We show that the sensing-transmission tradeoff problem can be formulated as a POMDP. To tackle the partial observability of states, large state space and large action space, a novel spectrum sensing algorithm based on deep reinforcement learning is developed, which has a great potential of handling channel correlated scenario. Moreover, the reward in our setting is more detailed and consists of several discrete values, which mitigates the impact of the reward sparsity on convergence speed. We first show that our proposed method can efficiently exploit both the temporal correlation and the channel correlation of wireless channels. Numerical results indicate that the system throughput can be increased by 9%-24% using our proposed algorithm, compared to the conventional learning-based spectrum sensing schemes.

The remainder of our paper is organized as follows. We give our system model and formulate the optimization objective as a POMDP in section II. In section III, we give a brief introduction to deep reinforcement learning and propose the solution for the sensing-transmission tradeoff problem. Section IV provides simulation results of both joint Markovian model and real data trace. Finally, we present conclusions in section V.

II. System model

In this paper, we consider a CR network with single SU and a set $\mathcal{K} = \{1, 2, ..., K\}$ of licensed channels. The CR network operates in a time-slotted fashion, where the behaviors of the PUs stay the same at each time slot. That is, the idle/busy state of licensed channels can only vary at the beginning of each time slot and remains the same in the rest of this time slot. The state of licensed channels at time slot t is denoted by a length-K vector $s_t = [s_{t,1}, \cdots, s_{t,K}]$, where $s_{t,i} = 1$ if i-th channel is idle and $s_{t,i} = 0$ if i-th channel is busy. Note that the state of each channel is not independent of each other as discussed in [10]. Due to the temporal correlation and the channel correlation of wireless channels, we model the channel state switching patterns as a 2^K-state Markov chain, whose transition matrix is denoted by $\mathcal{P}$.

The single time slot of the SU, T_f, consist of sensing duration and transmission duration, as shown in Fig 1. In the sensing duration, the SU determines how many licensed channels to sense. At time slot t, k_t channels are sensed by the SU. We denote a length-K vector $a_t = [a_{t,1}, \cdots, a_{t,K}]$ as the sensing indictor at time slot t, where $a_{t,i} = 1$ indicates that the SU senses i-th channel, $a_{t,i} = 0$ indicates that the SU does not sense i-th channel and $k_t = \sum_{i=1}^{K} a_{t,i}$. Each sensing

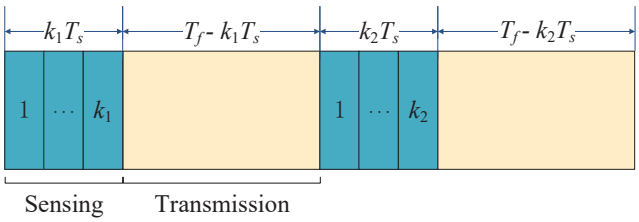

Fig. 1. Time slot structure of the SU in CR network.

operation spends time T_s and so the sensing duration of time slot t is k_tT_s. In the transmission duration, if one of the sensed channels is idle or $a_t^{\mathrm{T}} s_t \geq 1$, the SU transmits data over the idle channel for $T_f - k_tT_s$. The throughput of time slot t is calculated as in [4], [14] using the formula given below,

$$D_t = \frac{T_f - k_tT_s}{T_f} C_t, \tag{1}$$

where

$$C_t = \begin{cases} \log_2(1 + \mathrm{SNR}_{\mathrm{sec}}) & ; \ a_t^{\mathrm{T}} s_t \geq 1 \\ 0 & ; \ a_t^{\mathrm{T}} s_t = 0 \end{cases} \tag{2}$$

and $\mathrm{SNR}_{\mathrm{sec}}$ is the signal to noise ratio of the SU at the SU receiver. If the SU selects too many channels to sense, the probability of $a_t^{\mathrm{T}} s_t \geq 1$ increases but less time remains for transmission. If the SU senses not enough channels, no idle channel is found or $a_t^{\mathrm{T}} s_t = 0$, which makes the remaining transmission duration wasted.

An SU should find an efficient tradeoff between sensing and transmission, so as to maximize the achievable throughput in total time slots T. Mathematically, the throughput maximization problem can be described as follows,

$$\begin{aligned} &\max_{a_1, a_2, \ldots, a_T} \quad \sum_{t=1}^{T} D_t, \\ &s.t. \quad C1 : a_{t,i} \in \{0, 1\}, \forall t, \forall i \\ &\qquad C2 : 1 \leq \sum_{i=1}^{K} a_{t,i} \leq \lfloor \frac{T_f}{T_s} \rfloor, t = 1, \ldots, T \end{aligned} \tag{3}$$

where $C1$ declares that each channel can only be sensed by the SU at most once at a given time slot, $C2$ is the time slot duration constraint.

Note that the expectation of D_t given the state transition matrix $\mathcal{P}$ can be denoted by:

$$E[D_t] = \frac{T_f - k_tT_s}{T_f} \log_2(1 + \mathrm{SNR}_{\mathrm{sec}}) P(a_t^{\mathrm{T}} s_t \geq 1 | s_{t-1}). \tag{4}$$

After substituting (4) into (3), the above optimization problem defined by (3) can be split into T sub-problems which are described mathematically as follows,

$$\max_{a_t} \quad \frac{T_f - k_tT_s}{T_f} P(a_t^{\mathrm{T}} s_t \geq 1 | s_{t-1}), \quad \forall t. \tag{5}$$

Each sub-problem illustrated in (5) is an integer programming task that is generally NP-hard. Even worse, the state transition

matrix $\mathcal{P}$ is unknown for the SU, and the SU only has a partial view of the whole CR network which makes s_{t-1} partially unknown at time slot t. All these factors make us to formulate the sensing-transmission tradeoff problem as a POMDP.

In POMDP, an autonomous agent without any prior knowledge of the environment model aims to find an optimal policy, so as to maximize its long-term reward. The agent only has a partial view on the system environment, and makes decisions based on its observation. The reward of the agent only depends on its action and the environment state. The POMDP is denoted as the tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{O}, F, U, R\}$, where $\mathcal{S}$ is the set of possible environment states, $\mathcal{A}$ is the set of available actions for the agent, $\mathcal{O}$ is the set of possible observations for the agent. The latent state at time t is denoted by $s_t \in \mathcal{S}$. When an action $a_t \in \mathcal{A}$ is carried out, a reward $r_t \in \mathbb{R}$ is obtained by the agent based on the distribution $r_t \sim R(r_t|s_t, a_t)$, and the environment state changes based on the transition distribution $s_{t+1} \sim F(s_{t+1}|s_t, a_t)$. Afterwards, a partially occluded observation $o_{t+1} \in \mathcal{O}$ is received by the agent, based on the distribution $o_{t+1} \sim U(o_{t+1}|s_{t+1}, a_t)$.

The agent, the state space $\mathcal{S}$, the action space $\mathcal{A}$, the observation space $\mathcal{O}$ and the reward function R for the formulated POMDP is defined as follows:

- Agent: In this CR network, the agent represents the SU who is executing the sensing and transmission process.
- State: $s_t \in \mathcal{S}$ is the environment state at time slot t, which consists of the occupation feature of the channels. The occupation feature of i-th channel represents whether i-th channel is occupied by the PU.
- Action: $a_t \in \mathcal{A}$ is the action taken by the agent at time slot t, which consists of the sensing indictor of the channels. The sensing indictor of i-th channels denotes whether the agent senses i-th channel.
- Observation: At time slot t, the agent receives the observation $o_t \in \mathcal{O}$, which consists of the sensing result of the channels at time slot $t-1$. Mathematically, $o_t = [o_{t,1}, \cdots, o_{t,K}]$, where $o_{t,i}$ is 1 if $a_{t-1,i}$ is 1 and $s_{t-1,i}$ is 1, $o_{t,i}$ is -1 if $a_{t-1,i}$ is 1 and $s_{t-1,i}$ is 0, and $o_{t,i}$ is 0 if $a_{t-1,i}$ is 0.
- Reward: In our problem, the SU may transmit data or not at each time slot. We define the reward at time slot t as $R(s_t, a_t) = \frac{1}{\log_2(1+\mathrm{SNR}_{\mathrm{sec}})} D_t$, which is the normalized throughput of the SU.

In the formulated POMDP, the objective of the agent is to obtain an optimal policy Π^* that maximizes the expected cumulative discounted reward over a finite period T:

$$\max_{\Pi} \quad E[\sum_{t=1}^{T} \gamma^{t-1} R(s_t, a_t)]. \tag{6}$$

where $\gamma \in [0, 1]$ is the discount rate, determining the effect of the future reward. In general, finding the exact solution of a POMDP is of exponential computation complexity [15]. Even worse, both the state space and the action space of the formulated POMDP have 2^K elements. All these factors make it intractable to find the optimal solution to our formulated POMDP as the system size increases.

III. Proposed solution

A. Proximal Policy Optimization Algorithm

The proximal policy optimization (PPO) is a powerful deep reinforcement learning algorithm combining the advantages of the neural network function approximators to the policy gradient structure [16]. It has been widely applied in various sequential decision problems, primarily due to its ability of effectively handling high-dimensional action spaces and implementing stochastic policies. Specifically, the PPO creates two neural network function approximators for the policy function and the critic function. The policy network is used to select actions in the POMDP and denoted by $\pi_\theta : \mathcal{O} \to P(\mathcal{A})$, where $P(\mathcal{A})$ is the set of probability measures on $\mathcal{A}$, θ is the parameters of the policy network, and $\pi_\theta(a_t|o_t)$ is the conditional probability of a_t under o_t. The critic network is used to estimate the value function $V^\pi(o)$ and denoted by $V_\phi : \mathcal{O} \to \mathbb{R}$, where $V^\pi(o)$ is the expected long term reward of being in observation o and then following policy π, and ϕ is the parameters of the critic network.

The basic idea behind the policy gradient structure is to apply a gradient ascent algorithm on the policy parameters θ to the gradient estimated. The gradient of θ is estimated in Monte Carlo fashion by running its policy π_θ in the POMDP environment for N timesteps to get a trajectory τ and then obtaining a sample of the performance objective $J(\theta)$:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[\sum_t R\left(s_t, a_t\right) \right] = \mathbb{E}_{\tau \sim \pi_\theta(\tau)}[R(\tau)] \tag{7}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[\left(\sum_{t=1}^{N} \nabla_\theta \log \pi_\theta\left(a_t|o_t\right) \right) R(\tau) \right] \tag{8}$$

where τ represents trajectories of the form $\{(o_1, a_1, r_1), ..., (o_N, a_N, r_N)\}$, N is the total timesteps of a sampling epoch. In practice, these gradients are calculated by applying automatic differentiation software [17] on a surrogate objective, and are then backpropagated through the policy network to update θ.

In typical policy gradient structure, the surrogate objective of the policy network requires a trajectory sampled from the policy being optimized, which is no longer applicable for the improved policy network after a single gradient ascent. To increase the efficiency of the trajectory, the PPO applies importance sampling to get the expectation of the trajectory sampled from an old policy $\pi_{\theta_{old}}$ under the new policy π_θ. Here, θ_{old} is the parameters of the policy network before the gradient ascent, and θ is the parameters of the policy network we want to improve. In this way, each trajectory can be exploited for several gradient ascent steps. The surrogate objective that the PPO maximizes is defined as follows:

$$L^{CLIP}(\theta) = \sum_{t=1}^{N} [\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)], \tag{9}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{old}}(a_t|o_t)}$ is the probability ratio, the clip operator moves the variable in the first argument to the range supplied by the two following arguments, and $\hat{A}_t$ measures how good a_t is compared to the other actions in the observation o_t and is estimated by: $\hat{A}_t = r_t + \gamma V_\phi(o_{t+1}) - V_\phi(o_t)$. As the new policy is improved, the divergence between the two policies increases, which reduces the rationality of importance sampling. For the surrogate objective to be valid, the divergence between the two policies is limited by clipping the probability ratio $r_t(\theta)$ to the region $[1-\epsilon, 1+\epsilon]$. Note that the critic network can be updated by minimizing its mean square error and then the surrogate objective of the critic network is:

$$L^{VF}(\phi) = -\sum_{t=1}^{N}[r_t + \gamma V_\phi(o_{t+1}) - V_\phi(o_t)]^2. \quad (10)$$

B. Learning-Based Spectrum Sensing

The architecture of the PPO algorithm proposed to solve the sensing-transmission tradeoff problem is shown in Fig 2.

1) LSTM Layer: Since the environment state is partially observable for the agent and the channel state transition characteristic may be non-Markovian, classic neural network does not work well in this setting. Thus, we introduce a Long Short Term Memory (LSTM) layer to the PPO, which can aggregate observations over time and then give the agent the ability of estimating the true state [18]. The input observation $o \in \mathcal{O}$ of the LSTM layer is a vector of size K, and the output feature is also a vector of size K.

2) Policy and Critic Layers: Another improvement we adopted is parameter sharing between policy network and critic network, as suggested in [19], which can speed up the convergence of the PPO algorithm. The intuition behind the architecture lies in saving the trouble of training two LSTM feature extractors separately. The policy layer is a fully connected layer, which takes the output of LSTM layer as its input and then outputs the action distribution $\pi_\theta(a|o)$. In practical, the action distribution is a general distribution with unknown parameters, and so the policy layer just needs to output its parameters. Note that $\mathcal{A}$ has 2^K elements and the computational complexity of the policy layer is proportional to the size of its output, which means that designing an appropriate action distribution is important. By exploiting the correlation of 2^K actions, we find that a joint distribution of K Bernoulli distributions is enough, where k-th Bernoulli distribution decides the probability of sensing k-th channel. Thus, the output of the policy layer is a vector of size K. The critic layer is a fully connected layer, which takes the output of LSTM layer as its input and then outputs the value $V_\phi(o)$.

In our experiments, the PPO algorithm updates its parameters in an online manner. The agent splits the total time T into fixed-length trajectory segments. Each iteration, the agent collects N timesteps of data. Then we construct the surrogate objectives $L^{CLIP}(\theta)$ and $L^{VF}(\phi)$ based on those N timesteps of data, and optimize them with a gradient method. The full framework is given in Algorithm 1.

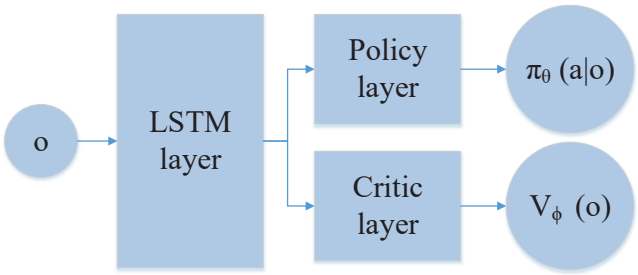

Fig. 2. An illustration of the architecture of the proposed PPO algorithm for spectrum sensing.

Algorithm 1 Proximal Policy Optimization for Spectrum Sensing

Initialize the policy network $\pi_\theta(a|o)$ with weights θ
Initialize the critic network $V_\phi(o)$ with weights ϕ
for iteration $i = 1, 2, ..., \frac{T}{N}$ **do**
  $\tau = \emptyset$
  for $t = 1, 2, ...N$ **do**
    With the observation o_t, the SU takes action a_t according to the stochastic policy $\pi_\theta(a_t|o_t)$ and obtains a reward r_t
    $\tau = \tau \cup (o_t, a_t, r_t)$
  end for
  $\theta_{old} \leftarrow \theta$
  for $j \in \{1, 2, ..., M\}$ **do**
    Use τ to compute probability ratios $r_t(\theta)$
    Use τ to compute advantage estimates $\hat{A}_t$
    $L_t^{CLIP}(\theta) = \sum_{t=1}^{N}[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$
    $L_t^{VF}(\phi) = -\sum_{t=1}^{N}(r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t))^2$
    Update θ by a gradient method w.r.t. $L_t^{CLIP}(\theta)$
    Update ϕ by a gradient method w.r.t. $L_t^{VF}(\phi)$
  end for
end for

IV. Simulation Results

We validate the proposed PPO algorithm in both joint Markovian model and real data trace, and compare it with three conventional spectrum sensing algorithms, i.e., the deep Q-learning based (DQN) [10], the Thompson sampling based (TS) [8] and the random channel sense (RA) algorithms. In the DQN algorithm, the deep Q-network takes the state-action pair as its input and then outputs the corresponding Q-value. The channel that maximizes the Q-value is sensed by the SU at each time slot. In the TS algorithm, the SU chooses the channel that maximizes the sampling results of the defined posterior channel idle probability. In the RA algorithm, the probability of each channel being sensed is equal at each time slot. The simulation parameters of the proposed PPO algorithm are summarized in Table I.

A. joint Markovian model

The joint Markovian model is a 2^K-state Markov chain, which is difficult to implement completely. Thus, we adopt the

TABLE I
SIMULATION PARAMETERS

Parameter	Value
T_f	1 ms
T_s	0.1 ms
γ	0.99
ϵ	0.3
N	20
M	32
learning rate	0.004
optimizer	Adam
activation function	ReLU/Softmax

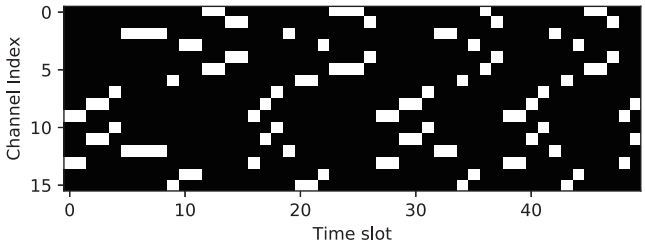

Fig. 3. A capture of the fixed-pattern channel switching model.

fixed-pattern channel switching model, as discussed in [10], [11], which is a special case of the joint Markovian model and can perfectly realize the temporal correlation and the channel correlation of wireless channels. In the simulation, K channels are evenly divided into several subsets that become available in turn with the fixed switching probability P_s. In Fig. 3, we give a pixel illustration to show how the idle/busy states of channels change in the fixed-pattern channel switching model over 50 time slots, where only two of the $K = 16$ channels are in idle condition at a given time slot and the switching probability is $P_s = 0.7$.

In Fig. 4, we show the reward as a function of the iterations process of deep reinforcement learning, in which $K = 16$ channels are evenly divided into 8 subsets and the switching probability is $P_s = 0.9$. Note that in the case $P_s = 0.9$, the optimal sensing duration is to sense one channel at a time slot, and so the fixed time slot structure of the DQN algorithm just right meets the system needs. Even so, the curve convergence speed of the proposed PPO algorithm is not inferior to that of the DQN algorithm. The PPO algorithm requires only less than 200 iterations (4 s since each iteration corresponds to 20 time slots) to approach within 90% of the optimal throughput, which implies that the convergence difficulty brought by the big action spaces in our problem is solved by the flexible reward design and the efficient algorithm design.

In Fig. 5, we show the average reward of the learning process as a function of the switching probability P_s, where $K = 16$ channels are evenly divided into 8 subsets. It is obvious that our proposed PPO algorithm performs much better than others, especially when the switching probability is smaller. The reason for the improvement is that the adaptive time slot structure makes great use of the time correlation and channel correlation of wireless channels. As the decrease of

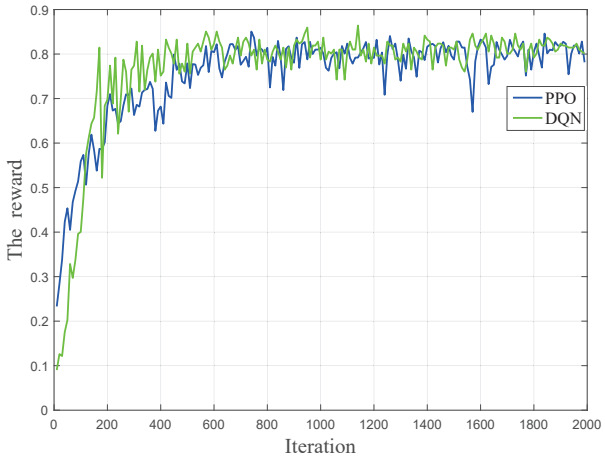

Fig. 4. The reward as a function of the iterations process of deep reinforcement learning under the switching probability $P_s = 0.9$.

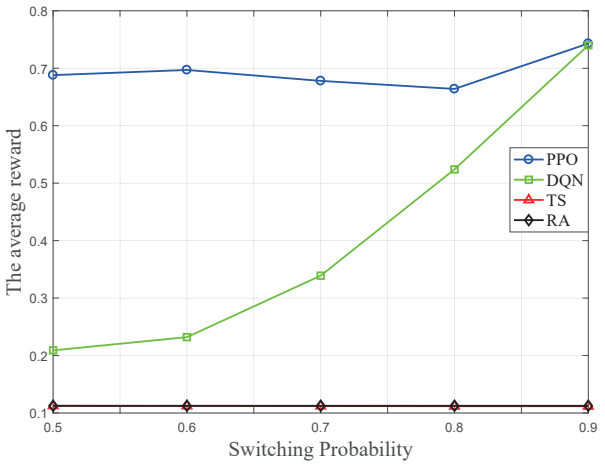

Fig. 5. The average reward of the learning process as a function of the switching probability P_s under 2000 iterations.

P_s, the uncertainty of the channel state increases, which makes the optimal sensing duration increase. In the TS algorithm, the correlation of channels is neglected, which makes it impossible to learn the channel switching pattern. In the DQN algorithm, the channel switching pattern is captured, but its fixed time slot structure makes its performance limited by the increased channel uncertainty. On the contrary, the PPO algorithm learns to optimize its time slot structure by exploiting the correlation of channels, thereby increasing the system throughput.

B. real data trace

The performance of the PPO algorithm is evaluated by real data trace obtained from our FSW43 spectrum analyzer. The measurement system operates on the frequency band 2635-2655 MHz, which has been allocated to China Telecom for TD-LTE services. We divide the range of 2635 MHz to 2655 MHz into 20 channels, each of which spans 1MHz. The

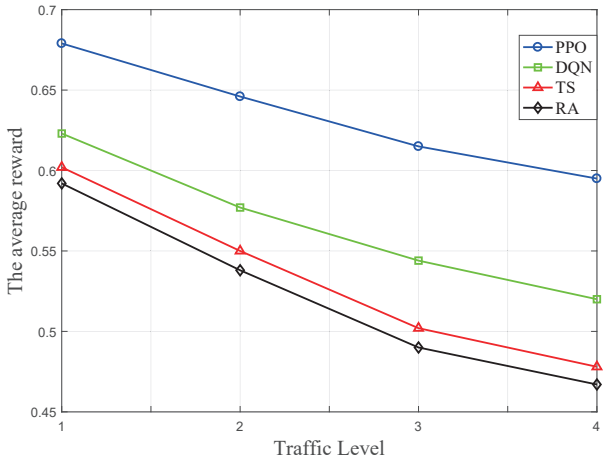

Fig. 6. The average reward as the traffic level under real data trace.

measurement data is collected indoor during 4 time periods, each time period lasting 5 minutes. We define the traffic level to measure how busy the channels are. The traffic level of each time period is related to its start time, i.e., 1 indicates $23:00$, 2 indicates $11:00$, 3 indicates $14:00$ and 4 indicates $20:00$.

In Fig. 6, we show the average reward as the traffic level under real data trace, in which the number of channels is $K = 20$. As we can see, the proposed PPO algorithm outperforms the DQN and the TS algorithms by $9\% - 15\%$ and $17\% - 24\%$, respectively. The improvement is higher when the traffic level is high. We show that existing learning-based algorithms perform unsatisfactorily in real data tests, since their system models make too many assumptions and their fixed time slot structure can not adapt to dynamic network conditions. The proposed PPO algorithm can learn the complex channel state transition characteristics and then adaptively adjust its time slot structure, thereby achieving significant improvements in SU throughput.

V. Conclusion

In this paper, we have considered the spectrum sensing problem in CR networks, where the adaptive time slot structure is introduced for system throughput maximization. We have formulated the sensing-transmission tradeoff problem as a partially observable Markov decision process and developed a novel spectrum sensing algorithm based on deep reinforcement learning. The proposed algorithm has a great potential of handling complex real-world models, and can obtain a good tradeoff between sensing and transmission. Simulation results of joint Markovian model have shown that the proposed algorithm can learn the correlation of wireless channels and then efficiently exploit the correlation through the adaptive time slot structure. In addition, for real date trace experiments, the throughput improvement is 9%-24%, compared with the conventional learning-based spectrum sensing schemes.

References

[1] Y. Zhang and S. Wang, "Resource allocation for cognitive radio-enabled femtocell networks with imperfect spectrum sensing and channel uncertainty," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7719–7728, 2016.

[2] C. Ran and S. Wang, "Resource allocation in heterogeneous cloud radio access networks: A workload balancing perspective," in *Proc. IEEE GLOBECOM'14*, 2014.

[3] S. Wang, Z.-H. Zhou, M. Ge, and C. Wang, "Resource allocation for heterogeneous cognitive radio networks with imperfect spectrum sensing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 464–475, 2013.

[4] Y.-C. Liang, Y. Zeng, E. C. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1326–1337, 2008.

[5] H. Jiang, T. Wang, and S. Wang, "Multi-agent reinforcement learning for dynamic spectrum access," in *Proc. IEEE ICC'19*, 2019.

[6] S. Sengottuvelan, J. Ansari, P. Maehoenen, T. Venkatesh, and M. Petrova, "Channel selection algorithm for cognitive radio networks with heavy-tailed idle times," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1258–1271, 2017.

[7] Y. Song, Y. Fang, and Y. Zhang, "Stochastic channel selection in cognitive radio networks," in *Proc. IEEE GLOBECOM'07*, 2007.

[8] M. Zhou, T. Wang, and S. Wang, "Spectrum sensing across multiple service providers: A discounted thompson sampling method," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2402–2406, 2019.

[9] H. Li, "Multiagent-learning for aloha-like spectrum access in cognitive radio systems," *EURASIP J. Wireless Commun. Netw*, vol. 2010, pp. 1–15, 2010.

[10] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, 2018.

[11] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1125–1139, 2019.

[12] J. Lai, E. Dutkiewicz, R. P. Liu, and R. Vesilo, "Opportunistic spectrum access with two channel sensing in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 1, pp. 126–138, 2015.

[13] S. Senthilmurugan and T. Venkatesh, "Optimal channel sensing strategy for cognitive radio networks with heavy-tailed idle times," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 1, pp. 26–36, 2017.

[14] V. Raj, I. Dias, T. Tholeti, and S. Kalyani, "Spectrum access in cognitive radio using a two-stage reinforcement learning approach," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 20–34, 2018.

[15] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Proc. NeurIPS'10*, 2010, pp. 2164–2172.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] P. Adam, G. Sam, C. Soumith, C. Gregory, Y. Edward, D. Zachary, L. Zeming, D. Alban, A. Luca, and L. Adam, "Automatic differentiation in pytorch," in *Proc. NeurIPS'17*, 2017.

[18] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *Proc. AAAI'15*, 2015.

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. ICML'16*, 2016, pp. 1928–1937.