Reinforcement-learning algorithm for cognitive users operating as independent agents in uncertain environments

Angeli K. V. Kordali, Panayotis G. Cottis

School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Athens, Greece

kordali@mail.ntua.gr, pcottis@central.ntua.gr

Primary User’s Traffic

Primary Users’ (PUs) traffic follows either deterministic patterns, as in TV transmission, or stochastic patterns, as in packet-switched or circuit-switched networks, where the packet arrival time follows the Poisson process [1].

Many approaches are based on traffic pattern learning to predict the future traffic in PUs channels such as in [2].

In [3], the authors classify the channels based on the history of collected data and apply constant monitoring whereas in [4] the channels are characterized by the probability of being idle based on statistics collected in a learning phase.

Both approaches address specific traffic patterns with static characteristics.

However, the traffic stochastic patterns cannot, in general, reflect the dynamic changes in the communication channels, especially when these channels are accessed by not registered users as the Secondary Users (SUs). The statistics of channel occupancy vary with time due to changes in traffic load. The SUs have to function in a completely unknown environment with no information about either the traffic pattern followed by the PUs or its specific characteristics (utilization level, frequency of state transitions, etc.)

Reinforcement Learning

The main principle of reinforcement learning (RL) is learning by selection and not by instruction [5].

The environment is represented by a discrete set of states S where decision makers/agents operate. In the general case, an agent receives an input from the environment, chooses an action a from a set of actions A and receives a reinforcement signal r for which it depends on the action taken and the current state s of the environment.

When it comes to determining the immediate reward, the actions of an agent/user have influence on the subsequent environment states and future rewards, the problem is modeled as a Markov decision process (MDP). Similarly to the formulation of the simple RL problem, an agent at state s, t ∈ S, can select an action a from a discrete set of actions A. This selection has two consequences: first, it offers a reward according to a reward function R: S x A → R; and, second, it leads to a new environment state s’, s’ ∈ S, following the state transition function T: S x A → B(S), where B(S) is a probability distribution over the set S.

Q-Learning Algorithm

It constitutes an online learning algorithm [6][7].

It is based on the following recursive equation:

\[ Q_{t+1}(s, a) = (1 - \alpha_t)Q_t(s, a) + \alpha_t[\gamma \max_{a_t} Q_t(s_{t+1}, a_t) + r_t] \]

where s and a denote the state and action taken at the following time instance (s_a, q_a) and r is the reward which is a discount factor to account for the contribution of future reinforcements (0 ≤ γ ≤ 1).

RL algorithms are characterized by two main components [8], the update rule which designates how an agent imports the accumulated experience into the update of the Q values of the actions and the learning policy which specifies the selection of the action at each time instance based on the Q-values.

In general, Q_t(s, a) = \sum_k \gamma^k \sum_j P_j(s_{t+k}, a_j) with probability 1 if the following conditions hold:

\[ \sum_k \gamma^k \sum_j P_j(s_{t+k}, a_j) = \infty \]

and \[ \sum_k \gamma^k \sum_j P_j(s_{t+k}, a_j) = \infty \]

where (q_a) is an indicator function taking value 1, if a = a and 0 otherwise.

System Description

M available channels occupied by M Primary Users.

S = \{(s_1, s_2, ..., s_M)\} is the set of the 2^n possible states of the M available channels.

A = \{a_1, a_2, ..., a_M\} is the set of the possible actions an SU can take. An action represents the channel chosen by an SU at a specific time instance, i.e., a_i denotes that channel c_i is chosen for transmission.

The SUs are equipped with only one transceiver; hence, parallel transmissions are not feasible and only one channel can be used at a time.

Time is divided into periods for sensing and transmission.

The Q-values, kept by the SUs, characterize solely their actions, i.e., their channel choices, and are independent of the current state, i.e., Q_t(s, a_j) = Q_t(s_j, a_j).

In case of a successful transmission, the received reward r_j \{s_j, a_j\} is the throughput related to the specific transmission as quantified by the number of successfully transmitted packets divided by the transmission duration.

The goal of the SU is to set the channels in a preference order based on the probability of being vacant and the estimated duration of the vacant period.

Proposed Algorithm

Step 1: Initialize Q-values \[ Q(s, a) = \begin{cases} 0 & \text{if } a \neq 0 \\ \infty & \text{if } a = 0 \end{cases} \]

Step 2: Evaluate \[ P_t(s, a) = \frac{\text{Temp}(s, a)}{\text{Temp}(s, a) + \text{Temp}(s, \text{null})} \text{ if } a \neq 0 \]

Step 3: Set sensing order \( E = \{e_1, e_2, ..., e_m\} \) based on the probability function P, j = 1

Step 4: Execute action a, which corresponds to order \( d_j \)

If channel \( c_j \) is vacant:

Transmit

Receive reward r_j \{s_j, a_j\}

else
goto Step 4 with \( j = j + 1 \)

Step 5: Update Q-values according to the update rule and go to Step 2.

Update Rules for learning procedure

Two update rules are considered for the learning procedure of Step 5:

- Learning with least squares parameter estimation (L-learning)
  In this successful transmission is completed, the Q-value of the probed channel is updated following the Q-learning model, i.e.,
  \[ Q_{t+1}(s, a_j) = (1 - \lambda_t)Q_t(s, a_j) + \lambda_t r_j(s, a_j) \]
  where r_j(s, a_j) is the reward and \( \lambda_t \) is the learning parameter quantifying the weight assigned to the latest information whereas \( 1 - \lambda_t \) is the weight assigned to the already accumulated experience. No future rewards are taken into account.

- Learning with discounted learning parameter (Time-Learning)
  It constitutes a modified RL update rule, where the SU counts the attempts made to access a primary channel \( c_j \), this count is denoted \( n_j \). The learning rate \( \lambda_t \) is not an a priori defined parameter. Instead, it is related to \( n_j \) via
  \[ \lambda_t = \frac{1}{n_j} \]
  thus
  \[ Q_{t+1}(s, a_j) = \left(1 - \frac{1}{n_j}\right)Q_t(s, a_j) + \frac{1}{n_j} r_j(s, a_j) \]
  And \( n_j \) is the number of times the SU has attempted to access channel \( c_j \).

Learning policy

In the proposed scheme the Boltzmann strategy is employed for the selection of a future action, i.e., which channel to access.

\[ P(A) = \frac{\exp(Q(A)/Temp)}{\sum_j \exp(Q(A)/Temp)} \]

Temp is the temperature parameter which is related to the variance of the Gumbel errors in a Logit discrete choice model. High Temp values favor exploration by reducing the importance of the variations of the Q values and low Temp values favor exploitation.

Simulations

- 10 available channels of different mean duration of vacant periods
- Mean duration of vacant periods (0.2-2.5s)
- Mean duration of vacant periods (0.2-1.75s)
- Metric Transmissions over switches

Conclusions

- Both L-learning and Time-Learning offer high exploitation of the available opportunities.
- The ratio of transmissions/hits is significantly higher than the case of no learning.
- The suggested algorithm works with any traffic pattern of Primary Users.
- Awareness is achieved based only on information collected by the SU.

Selected References


Acknowledgment

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.