Adaptive Spectrum Access and RF Classification for Collaborative Intelligent Radio Networks

ABSTRACT

This paper details solutions for problems proposed by the DARPA Spectrum Collaboration Challenge (SC2). The first problem entails the application of machine learning classification algorithms to the domain of radio signals. It requires designing a classifier that can detect the presence of differently modulated signals viz. analog FM, QPSK, GMSK in a noisy 3 MHz segment of radio spectrum. The channel is broken up into 100 KHz channels. Some of the channels may not necessarily have any signal at all. The output of the classifier should be one out of the 4 possible classes viz. empty, FM, QPSK, GMSK. The classifier is judged to be successful if its score (calculated based on a formula) is better than 68%. To solve this problem, the signal was channelized via a polyphase filter bank. The channels were then clustered based on signal characteristics and the strongest channel within a cluster was fed into a random forest classifier. This approach was successfully able to exceed the metrics provided by DARPA.

The second problem is building Collaborative Intelligent Radio Networks (CIRNs) that can coexist with other networks in the spectrum by learning and predicting other networks' spectrum access behavior. In this work, an agent is built that contends with the DARPA agent in selecting a particular frequency range. The designed agent is rewarded if it predicts the DARPA agent's state correctly, avoids collision with the DARPA agent, and penalized severely on colliding with the DARPA agent. To solve this problem, an exploratory stage learned the transition table governing the DARPA automata via a Monte-Carlo Markov Chain method. The transition table and observed reward states were then used to directly predict the value function for all possible actions. This approach exceeded the required metrics provided by DARPA.

INTRODUCTION

Since the invention of wireless communication, the airwaves have continued to fill with more and more data passed around in increasingly intermingled and complex communications networks. In order for communications networks to avoid interfering with one another, they must be either be geospatially separated from one another such that they do not leak significant power into neighboring networks or be modulated at a frequency which does not interfere with nearby ongoing communications. The current solution for dividing the airwaves is to designate blocks of frequency ranges where a block operator can operate autonomously. However, not all parts of the spectrum are occupied all the time, and such a commercialized allocation of the wireless spectrum is inefficient. Large segments of a spectrum owned by one particular network remain unoccupied while some other network might be running out of its limited spectrum resources. In other words, the spectrum is usually only partially utilized, and the inefficiency arises because the needs of those permitted to operate within a band can fluctuate.

With the explosion of wireless communications and need for more tightly packed airwaves, DARPA is seeking to utilize advances in machine learning and AI to tackle the problem of spectral resource scarcity [1]. By building Collaborative Intelligent Radio Networks (CIRNs), DARPA is hoping to build radios capable of intelligently communicating data from node to node without interfering with the current existing communications networks and other CIRNs. The Spectrum Collaboration Challenge 2 will pit several teams against one another in order to promote the creation of this next-generation radio network. As part of the challenge, all teams wishing to join must complete competency hurdles to show their capabilities in the vital competencies of the challenge.

Out of the three problems presented in [1], this work addresses the last two (hurdles 5.2 and 5.3) as they involve application of Machine Learning. Problem 5.2 assesses basic radio signal detection. A 3 MHz noisy channel is divided into 30 bins of 100 kHz each, and a classifier has to identify which bins are occupied by signals and which ones are just noise. The Probability of Detection P_D is the number of occupied bins that the classifier reports as occupied, divided by the number of occupied bins. The Probability of False Alarm P_{FA} is the number of unoccupied bins that the classifier reported as occupied, divided by the number of unoccupied bins. The signals that may be present are: analog FM, QPSK, and GMSK. The SNR seen by any individual signal type will be greater than or equal to 15 dB, and multiple or no instances of a particular signal type may be present. The classifier also needs to return the modulation scheme of the signal for each bin that it believes is occupied; P_T is the fraction of type reports that are correct. Such an evaluation ensures that false alarms are penalized twice – they increase P_{FA} and decrease P_T – because false classifications harm downstream learning algorithms more than the missed defects. The success condition for this problem is $S \geq 0.68$ where $S = P_D * (1 - P_{FA}) * P_T$.

Hurdle 5.3, which is the second problem that this paper addresses, deals with spectrum contention in wireless communication. For CIRNs to work in tandem with other networks, an agent has to be built that can learn and predict other networks' spectrum access behavior. The primary objective is to avoid collisions with other spectrum users. Our agent (Agent **A**) will vie for spectrum access against DARPA's agent (Agent **D**) a number of times. In each turn, two things happen simultaneously: i) Agent **D** makes its choice of spectrum resource by choosing an integer D_0, and ii) Agent **A** returns a pair of integers, its own choice of spectrum resource A_0, and its prediction P_{D0} of Agent **D**'s choice. Both D_0 and A_0 are integers between 1 and *M*. D_0 depends, in some undisclosed way, on A_0. Agent **D** is an automaton having *N* states, and there is a state transition matrix giving the probability of the next state as a function of A_0. Each state is associated with an integer between 1 and *M* which specifies Agent **D**'s output i.e. D_0. Multiple distinct states out of the *N* states may output the same D_0. Agent **A** is rewarded +1 if $A_0 \neq D_0$, +3 if $PD_0 = D_0$, and -12 if $A_0 = D_0$. The success condition is if the Agent **A**'s total payoff is greater than 3000, after the final 1000 turns played against Agent **D** that has 10 states ($N = 10$).

BACKGROUND

[2] studies the application of Convolutional Neural Networks (CNNs) to complex valued radio signals, and why it is a good and viable to implement blind temporal learning on large and densely encoded time series. [2] also address the problem of Modulation Recognition by treating it as n N-class decision problem. This is the latest work that's being done for such problems and shows promising results in achieving Modulation Recognition.

In order to effectively model MDP, one of the techniques often used are MCMC, Markov Chain methods, or transition table evaluation. Mean field theory MCMC utilizes several adjacent data points to come up with an aggregate field which draws the random walk of the MCMC process in a general direction as it explores the decision space. In effect, a field is created from the individual data points and their likelihoods. After each iteration probabilities are updated using Bayes rule and fitting a model indicative of the probabilities of the data.

METHODS

Signal classification for hurdle 5.2:
DARPA transmitted to teams a continuous sample stream of 3 MHz of bandwidth at base-band. Teams were asked to divide the spectrum into 30 100 KHz bins, and identify the modulation scheme within a bin. Rather than classify and then label bins, the approach was taken to split the spectrum first into 30 channels using a polyphase filter bank. For all challenges, DARPA heavily made use of the open-source library *gnuradio*, which contains an implementation of a polyphase filter bank based channelizer. Using the gnuradio implementation, the signal was separated into 30 bins. Not all bins contain a signal, some are just noise, which are excellent candidates for broadcasting in. To classify noise, a more practical approach was taken utilizing the specifications provided by DARPA. The algorithm for noise detection is described below.

ALGORITHM 1.0 – Noise Detection

```
for each bin :
     p[bin] = mean(abs(samples))
     noise_level = argmin(p);
for each bin :
```

$$\texttt{snr[bin]} = 10 * \log 10\left(\frac{\text{noise\_level} - \text{p[bin]}}{\text{noise\_level}}\right)$$

```
     noise[bin] = snr[bin] < (argmin(snr) + threshold)
```

Once the Noise was determined for each bin, a simple clustering algorithm was applied to group adjacent bins. The rationale for grouping adjacent bins came from DARPA's implementation of the spectrum, where signals always were surrounded by empty "guard" bins to reduce the probability of interference from other signals. The clustering algorithm is described below.

ALGORITHM 1.1 Signal Clustering

```
for each bin :
    if previous_bin equals edge or noise :
        if bin is not noise :
            create new signal cluster label
    if bin is signal :
        append to newest signal cluster label
```

Once the groups of signals were clustered by associated bins, the max SNR bin was fed into a random forest classifier, trained on data synthetically generated by the same code used by DARPA for the test vectors.

```
for each cluster :
    test_bin = argmax(snr(bin in cluster))
    bin      = random_forest.classify(bin);
for each bin not in cluster :
    bin      = "noise"
```

The Feature space used for classification was generated from the magnitude and phase coordinates generated from the RAW signal data. In other words, IQ space was used to plot the feature geometries in Cartesian coordinates. Each feature vector contained 256 values down-sampled from the original sample space.

10,000 spectrum realizations were generated, all with differing power levels and signal characteristics and fed into the random forest classifier. For analysis of performance 1,000 realizations were tested and P_D / P_{FA} was computed.

Adaptive Spectrum Access for hurdle 5.3:
DARPA created an automaton which utilizes a fixed size higher dimensional Markov Chain to make moves based on observations of its opponent. In order to play the DARPA agent, an AI was developed to estimate the Automaton's Markov Chain by playing 29000 exploratory rounds again the DARPA agent, then directly compute the highest reward move it should make for the last 1000 rounds. For predicting the DARPA agents next move, an AI was rewarded 3 points. For avoiding a collision with the DARPA agent, the AI was awarded 1 point. If the DARPA agent and AI made the same move, the AI was penalized 12 points.

Three different methods were used to compute the transition tables governing the DARPA agent. First a brute force method of simply trying every move possible to the agent over and over was implemented. Second, an epsilon greedy algorithm was implemented to attempt to find the high probability transitions that dominate the Markov Chain. Finally, a Monte-Carlo Markov Chain method was implemented to explore the transition table. The MCMC method had the closest Kullback-Leibler divergence from the true distribution, and therefore was chosen as the most appropriate method for the exploration stage.

Following the exploration stage, a greedy approach was implemented, where the most likely event was always chosen. A direct solution for the linear programming approach was also implemented. Where the reward was also kept track of, not just the observed transitions in the

$$\pi(s) := \arg\max_a \left\{ \sum_{s'} P_a(s,s') \left(R_a(s,s') + \gamma V(s') \right) \right\}$$

exploratory stage. The equations below then could be directly computed:

$$V_{i+1}(s) := \max_a \left\{ \sum_{s'} P_a(s,s') \left(R_a(s,s') + \gamma V_i(s') \right) \right\},$$

The Probabilities were given from the exploration stage as well as the Mean rewards. V is computed recursively for 5 steps. The discount factor gamma was set to 0.4, as there is a strong preference for transitions which stay on a "path". Within the transition table DARPA added paths with 0.6 probability for the automaton to follow. By evaluating future moves and weighing them heavily, the AI was able to weight more heavily transitions that would remain on a path.

RESULTS

Signal classification for hurdle 5.2:

The DARPA metric was passed using the methods outlined above. Each individual step performed as described in this section.

Noise was detected in 1,000 realizations with 98.56 % accuracy. The issues normally arose in wide-band signals with low signal strength, where a bin was confused as a guard bin or had a strong enough dip to lower the average power below the noise threshold.

Peak signal power was detected with 93% probability of detection 1.5 % probability of false alarm, mostly due to small errors in noise could cause multiple clusters to form or clusters to be collapsed into a single cluster if noise went undetected. The signal was classified with 86.7% Accuracy.

Confusion matrix	GMSK	QPSK	FM
GMSK	322	51	2
QPSK	33	234	12
FM	22	13	311

The confusion matrix shows the tendency for the GMSK and QPSK signals to be mixed relative to all other classes. As per the scoring metric, S = 0.69907077.

Adaptive Spectrum Access for hurdle 5.3:
The DARPA metric was passed using the methods outlined above.

Kullback-Leiber Divergence 10,000 trials:

Epsilon Greedy	Brute Force	MCMC
0.8	0.6	0.11

DARPA Score MCMC last 1000 evaluation Steps, avg. 10000 trials:

Epsilon Greedy	Linear Programming	Markov Decision Process
2800 ±1133	3200 ±350	1244 ±2032

(For 5.3, show plots of rewards vs. iterations.)

CONCLUSION

We successfully showed two methods for completing the spectrum sharing and classification hurdles proposed by DARPA. In comparison with modern methods, the spectrum sharing agent performed favorably. However, the classification stage has much room for improvement.

For future research, the classification stage and adaptive spectrum fairing stage must be combined into a single process which takes into account the uncertainty in detection. Recent steps in reinforcement learning and convolution neural nets may hint that this joint process could be implemented as a q-learning deep net. Furthermore, the hardware must be brought in to give real world data rather than synthetic data. Though the signal data should be consistent with synthetic data, there are other network characteristics yet to be modeled.

BIBLIOGRAPHY

[1] https://spectrumcollaborationchallenge.com/wp-content/uploads/Entrance-Hurdles-Revision1.pdf

[2] Timothy J. O. Shea, Johnathan Corgan, T. Charles Clancy, “Convolutional Radio Modulation Recognition Networks”, arXiv preprint arXiv:1602.04105