Methods

Signal Classification

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 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 (PNOISE) = ARGMIN(P);

FOR EACH BIN :

SNR[BIN] = 10 \* LOG10 ( ( PNOISE – P[BIN] ) / PNOISE );

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.

ALGORITHM 1.2 SIGNAL CLASSIFICATION

FOR EACH CLUSTER :

TEST\_BIN = ARGMAX(SNR(CLUSTER->BIN))

BIN = RANDOM\_FORREST.CLASSIFY(BIN);

FOR EACH BIN OUTSIDE A 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 x and y. 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 of performance 1,000 realizations were tested and PD / PFA was computed.

Adaptive Spectrum Access

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, a 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 exploratory stage. The equation below then could be directly computed :

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

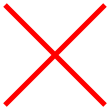
Signal Classification

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.

****

The confustion matrix shows the tendency for the GMSK and QPSK signals to be mixed relative to all other classes.

AI Agent

The DARPA metric was passed using the methods outlined above.