Predictive Behavior Classification For Cognitive Radio

Introduction and Preliminary Results

Daniel DePoy and Tamal Bose
Virginia Polytechnic Institute and State University
Wireless@VT
Blacksburg, USA
drdepoy@vt.edu, bose@vt.edu

Abstract— **Cognitive Radio systems rely heavily on artificial intelligence capabilities to perform a variety of tasks. Sharing spectrum resources more efficiently, self organization, and interference mitigation are just a few examples. For many CR applications, a primary goal is to decentralize and distribute network functions among participant nodes. As a consequence, any given node in a CR network may be required to coordinate with not only its peers, but also with a number of unknown transmitters. Thus, it is desirable that individual nodes be capable of predicting future states of non-peer transmitters in order to better optimize their own operation. In this paper we introduce methods for identifying cognitive behavior in an unknown transmitter and predicting likely future states based on physical spectrum observations. We discuss the problem in the context of our Universal DSA Network Simulation (UDNS) and present two behavior classification algorithms used to this end.**

(Coginitive Radio, Behavior Classification, Naive Bayes, AODE, Dynamic Spectrum Access)

I. Introduction

Cognitive Radio (CR) is a term used to describe a broad range of applications whereby a radio transceiver may dynamically alter certain parameters or behaviors in a manner which allows it to optimize some aspect(s) of its operation. Generally, this process involves some combination of spectrum sensing, decision making and adaptive behavior which is governed by an artificial intelligence or machine learning mechanism, called a Cognitive Engine (CE). Through the use of CEs and specially adapted networking protocols, the responsibility of coordinating transmission between and within discrete wireless networks can be increasingly shifted to network nodes themselves – making the need for centralized infrastructure and spectrum licensing progressively more obsolete. However, while the CR concept shows promise towards providing novel solutions to numerous wireless networking problems, the reduced reliance on central control networks and block licensing will present entirely new challenges for designing multiple access and network coordination protocols.

Currently, much of the elementary research in the CR field revolves around the broad concept of Dynamic Spectrum Access (DSA) – i.e., the ability to find and make use of currently unoccupied spectrum. In turn, DSA applications are often discussed within the context of licensed, or Primary Users (PUs) which have priority on a given channel, and unlicensed, or Secondary Users (SUs) which may use spectrum opportunistically – only in the absence of primary users. Assuming a CR user can detect the transmission of a static, non-cognitive PU, avoiding interference is a relatively simple proposition. However, this scenario is clearly oversimplified. In most non-trivial applications, we must also consider the presence of non-peer SUs which may also be attempting to opportunistically access spectrum resources.

Conventional wisdom suggests that we could simply consider any unknown transmission to originate from a PU, and avoid it as such. This tactic may be suitable in the presence of few SUs, but will scale poorly in the presence of many SUs since non-peer transmitters have no way of directly coordinating their DSA behavior. In this situation, non-peer SUs may respond to the introduction of a wideband PU or jamming signal by continuously hopping to the same vacant channel. This represents a likely scenario if it is assumed that each SU network is sensing the same spectrum, in a similar manner as other nearby SU networks. Upon detecting unknown transmissions on a newly established channel, the network nodes will assume a PU is present and vacate the channel once again. This process may potentially occur repeatedly if non-peer SU networks continue to select the same next frequency over multiple channel selection events. Such behavior is clearly inefficient from a media access standpoint, as channel evacuation behavior introduces significant network overhead, which reduces throughput. The channel evacuation loop scenario is especially inefficient if we consider the presence of vacant spectrum which was not selected by a SU network for one reason or another. To remedy this problem, we propose the integration of predictive machine learning algorithms into the CE framework which will enable a CR to observe an unknown transmitter, determine if it possess adaptive capabilities, and infer information about these capabilities towards the goal of predicting likely future states (e.g. - next channel selection). Based on these predictions, a CR network can tailor its own operation in a manner which will reduce the probability of interference with non-peer SUs, while maximizing the time between channel evacuation/selection events in a congested network. This paper presents preliminary work on the CR behavior classification problem by considering the simple case of classifying channel selection behavior heuristically. Using simulation, we present a basic classification framework built around two common predictive algorithms, and qualify the utility of our framework for predicting future CR behaviors.

CROWNCOM 2012, June 18-20, Stockholm, Sweden

DOI 10.4108/icst.crowncom.2012.248518

II. Previous Work

A. Universal DSA Network Simulation (UDNS)

For this work, we employ a MATLAB simulation framework called the Universal DSA Network Simulation (UDNS) as a testing platform. The UDNS was developed by Wireless@VT as a tool to analyze the performance of different SU DSA channel selection strategies in the presence of several PUs [2], and was modified to support our predictive classification research [6]. Prior to running a simulation, the user must define the channel, PU and SU parameters for each scenario to be simulated. PUs are assigned fixed channels, SUs are assigned a channel selection method, and both are assigned an activity lambda which defines how often a transmitter will access the current channel. In addition, each PU and SU is assigned an SNR value which defines how visible the transmissions are to the SU energy detectors.

When a scenario is executed, the PUs begin to simulate transmission according to their configured parameters for several iterations before the SUs come online. This delay provides the SUs with an opportunity to observe the PU transmissions, and aggregate channel observations before making a selection. Once online, if an SU detects the presence of a PU on its current channel, it will evacuate the channel and make a new selection according to its selection algorithm. Each time a SU selects a new channel, the SU ID number is recorded, along with the current state of each channel over the corresponding iteration. The resulting channel state matrix entries correspond to the channel energy, time since the last hop, and the access/occupancy rates of the SU on the given channel. Following the simulation, this information is passed to the preprocessor.

B. Channel Selection Methods

Five channel selection algorithms are used during the simulation, based on our previous work with the UDNS [2]. The selection algorithms available in the simulation are as follows:

- *Random selection*: SU selects a channel at random regardless of occupancy rate or channel energy.
- *Least energy:* SU selects the channel with the least energy at the time of selection.
- *Least recently occupied:* SU selects the channel it has occupied least recently.
- *Least frequently accessed:* SU selects the channel with the lowest observed activity.
- *Least frequently occupied:* SU selects the channel it has occupied least often.

These algorithms represent a survey of some basic channel selection strategies which may be encountered in a typical CR deployment scenario, though they do not represent a complete list by any means. In a real world deployment, channel selection behavior will be only one observable manifestation of data link and CE behavior, which may be classified in a similar manner.

III. Predictive Behavior Classification

Predictive behavior classification is a well studied topic with applications in a wide variety of fields – from financial analysis to traffic modeling [1]. Despite this, behavior classification in the context of CR networks has received little literature attention thus far. As previously stated, the goal of our research is to apply these well understood machine learning and data mining principles in order to classify unknown transmitters, and identify their adaptive or cognitive capabilities. In this preliminary work, we approach the topic by defining two problems.

1) *Given sufficient observation (training) time, can we predict the channel selection algorithm in use by several transmitters through observation of their hopping patterns?*

2) *Given sufficient observation (training) time, can we predict the current channel selection algorithm in use by a single SU for which the channel selection algorithm in use is changed over time?*

We present multiple simulations based on these scenarios which will demonstrate the application of our classification framework for CR behavior prediction, as well as provide a basis for future research. To perform the actual classification, we use two classifiers, as well as a preprocessor – the details of which are described in the following sections.

A. Naive Bayes Classifier

One of the simplest, yet most widely implemented probabilistic classification algorithms is the Naïve Bayes (NB) classifier [7]. These classifiers directly apply Bayes' Theorem over a set of empirical input attributes by making a "naïve" assumption of independence between each observed attribute. By making this assumption, the conditional likelihood function can be tractably estimated according to (1).

$$p(A_1, A_2, \ldots, A_n | C) \propto \prod_{i=1}^{n} p(A_i | C) \tag{1}$$

Where C is the class being tested and A_n are the attribute observations corresponding to the event being classified. This expression (1) estimates the likelihood of a class based on the product of the conditional probabilities between a class and each observed attribute. Combining (1) with the known priors, $p(C)$, for our class and attribute probabilities (determined from the training set), the NB probability model can be expressed as (2).

$$p(C | A_1, A_2, \ldots, A_n) = \frac{p(C) \prod_{i=1}^{n} p(A_i | C)}{p(A_1, A_2, \ldots, A_n)} \tag{2}$$

In addition to the attribute independence assumption, the NB classifier requires inferring or generating a probability distribution for each attribute model. For the NB case, we use a Gaussian distribution to approximate the conditional likelihood parameters, $p(A_i|C)$. The NB classifier is analyzed using both the raw observations, as well as with the preprocessed data. Since the classifier is common and simple to implement, it provides a useful baseline level of effectiveness for our behavior classification concept as a whole.

B. *Averaged One-Dependence Estimation Classifier*

The primary weakness of the NB classifier is the attribute independence assumption required to make estimation of the conditional likelihood function tractable. Simply put, any conditional dependence among discrete attributes will result in a corresponding increase in classification bias, and a decrease in the overall accuracy of the resulting classifier. One method for reducing the bias of a Bayesian classifier is to use One Dependence Estimation (ODE) to approximate the conditional likelihood function [8]. ODE supports a weaker independence assumption by estimating dependence of pair-wise attribute combinations during training, and constructing the probability model according to these pairs, rather than individual attribute observations alone. This can be thought of as a special case of the NB classifier, where the priors and likelihood function are conditioned by a single parent attribute, A_p, before posterior estimates are computed.

$$p(C|A_1,\ldots,A_n)=\frac{p(C,A_p)\prod p(A_p|C,A_i)}{p(A_1,A_2,\ldots,A_n)} \quad (3)$$

An Averaged One-Dependence Estimator (AODE) averages ODE posterior estimates for all possible parent attributes and class labels, and returns a posterior estimate for each class – In contrast to the NB classifier, which returns only the most likely class label.

$$p(C|A_1,\ldots,A_n)=\frac{\sum_{j=1}^{n} p(C,A_j)\prod_{i=1}^{n} p(A_j|C,A_i)}{p(A_1,A_2,\ldots,A_n)} \quad (4)$$

As a consequence, not only does AODE reduce estimation bias over conventional NB, it also provides a useful means of determining the confidence level of the returned maximum *a-posteriori* class label. For our implementation, the joint and conditional probability terms, $p(A_j|C,A_i)$ and $p(C,A_j)$ required to compute (4), are approximated using a joint frequency table and m-estimation, as described in [3]. Our AODE implementation does not estimate the likelihood function according to a continuous probability distribution model like NB, and can therefore only handle empirically discrete or discretized continuous inputs. In order to handle the requirement for discrete data, and reduce the number of attribute dimensions, a preprocessor is used to condition the attribute observations prior to training and prediction. It should be noted that this is a limitation of our AODE implementation only, and does not reflect a universal limitation of the classifier. Though it is most commonly implemented for discrete cases, as in [3], several algorithms for a continuous or hybrid AODE classifiers are presented in the literature [5], and are a topic of interest for our continuing research.

C. *Preprocessing*

When the simulation is run, each time a SU is forced off a channel by a PU, the SU ID, as well as information about each channel is recorded in two arrays. The first array records information about the time and frequency with which the SU has accessed the current channel, and the second array records the energy level, total access rate and total occupancy rate for each channel at the current time step. These arrays may be used to directly train the NB classifier, however we can improve the classification accuracy significantly by first passing the simulation outputs through a preprocessor. The preprocessor allows us to exploit heuristics within the data set in order to reduce the dimensionality of the probability model, which in turn will generally reduce the bias of the classifier [7].

Our approach is similar to the clustering, or latent class approaches described in [9] and [4]. For each recorded SU event, the channel parameters are interpreted to fit within one or more latent class labels based on the channel selection strategies we wish to classify. For example, if an SU selects a new channel that is both the one with the least energy and the one it has least recently occupied, the preprocessor output vector for this event will contain non-zero attribute entries corresponding to these latent classes. By performing this attribute hard-coding, we generate a superposition mapping raw attributes to latent class attributes which reflect all the relevant information from the simulation output, without completely discarding potentially useful attributes. This produces a 10 fold reduction in dimensionality for our simulation, which will significantly improve the classification accuracy of the NB classifier, and allows us to implement a computationally efficient AODE classifier which does not require the estimation of multivariate probability distributions as described in [5].

IV. Experimental Setup

Using the UDNS, we construct two scenarios for testing our classification strategies, as described in section 3. Ten arbitrary channels are configured, having additive noise properties between -3 and 0 dB, and one PU is configured for each channel. The PU configuration properties include transmit and idle rates, apparent SNR (as seen by the SUs) as well as the symbol rate, modulation and RRC alpha value. These values are used to model the basic behavior of a static PU transmitter. The SUs are configured with similar properties, but additionally require an entry for which channel selection strategy to simulate, as well as a list of channels to monitor. Assigning symbol rate, modulation and SNR properties to the SUs and PUs allows for a more realistic simulation in which each PU transmission, as seen by the SUs, is not simply a binary on-off state, but is rather deterministically sensed by SU energy detectors. Depending on the channel and PU configuration for a given time step, this accounts for the possibility of false alarm and missed detection events that will occur in a real CR network. Though not specifically addressed in this paper, detection and identification of false alarm events within a CR network is another research topic that we aim to address using similar classification strategies in subsequent work.

Two separate experiments are presented. For the first experiment, five SUs are configured, each using a separate channel selection strategy as defined in section 2.b. The simulation is run until 10,000 channel selection events are recorded, at which point the simulation output is run through the preprocessor and training routine. The goal for the first experiment is to determine the accuracy with which the classifiers can correctly predict the channel selection strategy for a given SU channel selection event, in the presence of several users. We qualify the classifier as a function of the training set length (*n*), in terms of the probability of correctly classifying the final 1000 recorded SU events. The first *n* simulated events and their *a-priori* class labels are used for training, and the events to be classified are drawn from the end of the list in order to ensure there is no overlap between the two data sets.

For the second experiment a single SU is configured to randomly choose a new channel selection strategy every 500 events. Ten PUs are additionally configured in the same manner as the first scenario. The output from the first simulation, where each radio was assigned a separate, static selection algorithm is used to train the classifier for the second experiment. The goal of this scenario is to determine how the classifier responds to a variable selection strategy in a single radio, as well as qualifying which errors the classifier makes most frequently (e.g. - which channel selection strategies are most difficult to differentiate). In the interest of space, only the AODE classifier is presented for this scenario, and will be used to predict the channel selection method for each decision event, independent of each previous event. The accuracy of classification will be based on how many predictions the classifier correctly makes over 5000 decisions, after 5000 training events are recorded.

A. Results

The results from the individual experiments are presented in the figures below. Figure 1 shows the classification accuracy for the AODE classifier, as well as for the NB classifier with and without preprocessing, as described in section 3.c. The minimum training length for the NB classifier is around 200 events in order to ensure that the in-class variance estimate for each attribute is non-zero. The AODE implementation has no such requirements, but in the interest of visual consistency, only a subset of the results are displayed in the plot. For the first experimental scenario, figure 1 confirms the utility of both the NB and AODE classifiers in predicting the correct channel selection strategy with reasonable accuracy. Comparing the NB output for the non-preprocessed case to the preprocessed case additionally confirms that our latent class clustering algorithm functions as expected, and produces classification results with higher accuracy and lower variance due to the reduction in attribute dimensions. Compared to the NB classifier, the AODE case shows a moderate, but consistent improvement in classification accuracy, in addition to a slight reduction in variance between training sets.

Figure 2 illustrates the ability of the AODE classifier to track changes in channel selection behavior for a single SU in the presence of several PUs. The Y-axis represents one of the four channel selection algorithms used in this simulation (random selection excluded), while the X-axis represents simulation time. The thick dashed line shows the actual progression between individual selection algorithms, while the lighter solid line represents the predicted class for each SU decision event. Channel selection index 1 corresponds to the least energy algorithm, and index 2, 3 and 4 correspond to least frequently accessed, least recently accessed and least occupied rate algorithms respectively. A divergence of the solid line from the dashed line indicates a classification error and shows which class was incorrectly selected.

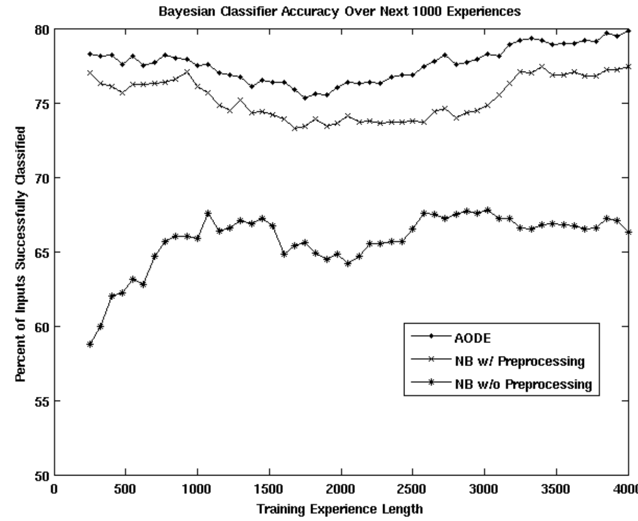

Figure 1. Classification Accuracy

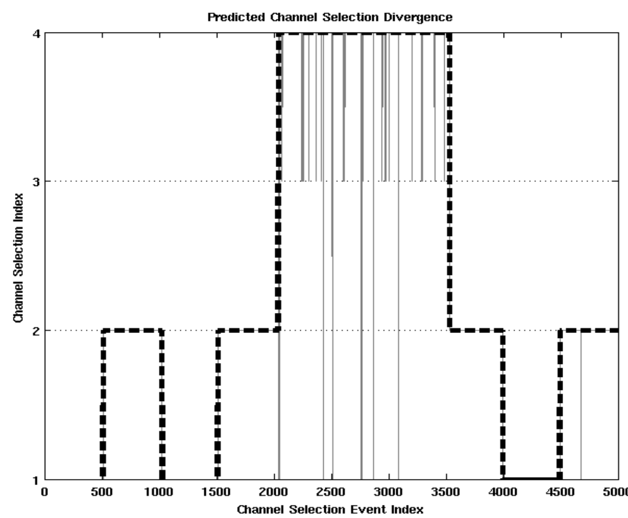

Figure 2. AODE Predicted Channel Selection Divergence

From figure 2, it is clear that the classifier performs better for some selection strategies than it does for others – specifically, the least occupied rate algorithm is occasionally wrongly predicted as the least recently occupied algorithm. This is not unexpected, as the two algorithms which are most commonly confused are the two most similar algorithms considered – and it is likely that the least recently occupied channel is also the channel that the SU has accessed the least for many events. For this simulation, Figure 2 shows a total of 51 errors over 5000 events classified, for an accuracy of greater than 98%. This represents a best case scenario where there is only a single SU to be classified. As illustrated in the first experiment, the classification accuracy is reduced when additional SUs must be classified at once.

B. Conclusions

This paper summarizes our preliminary proof of concept– work in analyzing the utility of behavior classification techniques for predicting the future states of a CR transmitter. Two predictive algorithms are analyzed for this purpose – Naïve Bayes and Averaged One-Dependence Estimation – and their relative performance compared. Using the AODE classifier and a basic clustering preprocessor, we were able to demonstrate up to 80% accuracy in classifying the channel selection strategy of several SUs (figure 1), and up to 98% accuracy in classifying a single SU which changes it's channel selection strategy over time (figure 2). We believe behavior classification represents a powerful tool with potential benefits for a number of CR areas, such as improving spectrum sharing efficiency in heterogeneous white space networks, as well as Electronic Countermeasures (ECM) applications.

C. Future Work

We hope to expand the scope of this work in several ways by expanding the UDNS and tuning the classification models to identify other cognitive capabilities, in addition to channel selection strategy. We plan to work the preliminary methods presented here into a larger classification framework, which will use tree-augmented machine learning to classify many other parameters of an unknown transmitter, including a more complete classification of the PHY and MAC layer technologies in use. Additionally, we plan to develop and test DSA protocols and CE algorithms which exploit these predictive capabilities for more efficient spectrum sharing, towards solving the repeated evacuation problem identified previously. We will also explore the application of our classification framework for the identification and classification of false alarm events in CR networks. Future work on the classification algorithms themselves will involve exploring the computational trade offs between different classifiers, such as those presented here. We will also explore additional preprocessing methods, such as tree-based augmentation, in addition to our current clustering approach. We believe there is significant potential in this relatively unstudied area, and hope to establish a basis for additional research within the CR community.

Acknowledgments

We would like to thank the faculty, staff and administrators from Wireless at Virginia Tech for their tireless dedication and hard work in support of this research. We would be lost without them.

References

[1] A. W. Moore and D. Zuev. "Internet traffic classification using bayesian analysis techniques." *SIGMETRICS Perform. Eval.* Rev. 33, 1 (June 2005), 50-60.

[2] B. Hilburn and T. Newman and T Bose and S. Kadambe. "A Survey of Basic Channel Selection Techniques for Cognitive Radios," *Technical Conference and Product Exposition, SDR '10*. 2010 pp.37-41.

[3] G. I. Webb and J. Boughton and Z. Wang. "Averaged One-Dependence Estimators: Preliminary Results." *University of Technology Sydney*. 2002, pg. 65-73.

[4] J. Abonyi and B. Feil . "Computational Intelligence in Data Mining." Informatica Vol. 29 Number 1: pg. 3-12. May 2005.

[5] M. J. Flores and J. A. Gámez and A. M. Martínez and J. M. Puerta. "GAODE and HAODE: two proposals based on AODE to deal with continuous variables." *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pg. 313-320.

[6] M. Rebholz and T. Bose and S. Kadambe, "Classification of DSA Channel Selection for Estimation of Secondary User Behavior," *Cybersecurity for Cognitive Radio, IEEE MWSCAS*. 2011, 7-11 Aug.

[7] N. Friedman and M. Goldszmidt and D. Geiger. "Bayesian Network Classifiers." *Machine Learning. Springer Netherlands*, Issue 2, Vol 29 pg. 131-163.

[8] N. Li and Y. Yu and Z. Zhou. "Semi-naive Exploitation of One-Dependence Estimators," *Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on* , vol., no., pp.278-287, 6-9 Dec. 2009

[9] T. Hofmann and J. Puzicha and M. Jordan. "Unsupervised Learning from dyadic data." *Advances in neural information processing systems* 1999: 466. 14 Jan 2012.