

# Wideband Spectrum Sensing and Signal Classification for Autonomous Self-Learning Cognitive Radios

by

**Mario Bkassiny**

B.E., Electrical Engineering, Lebanese American University, 2008

M.S., Computer Engineering, Lebanese American University, 2009

DISSERTATION

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

Doctor of Philosophy  
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

May 28th, 2013

SUBMITTED BY: Mario Bkassiny

SUPERVISOR: Dr. Sudharman K. Jayaweera  
Department of Electrical and Computer Engineering

COMMITTEE MEMBERS: Dr. Christos G. Christodoulou  
Department of Electrical and Computer Engineering

Dr. Chaouki T. Abdallah  
Department of Electrical and Computer Engineering

Dr. Guoyi Zhang  
Department of Mathematics and Statistics

©2013, Mario Bkassiny

# Dedication

*To my parents, Tannous and Josephine, for their support and encouragement.*

*“Give me six hours to chop down a tree and I will spend the first four sharpening  
the axe”*

*– Abraham Lincoln*

# Acknowledgments

First, I would like to thank my advisor, Professor Sudharman K. Jayaweera, for his support and guidance over the past years at UNM. His valuable advice and assistance were essential in achieving this work and in helping me to develop my research career. Working with Professor Jayaweera has been a great learning experience for me. I am very grateful for his efforts and encouragement during this work period. I would like also to thank my committee members, Professor Chaouki T. Abdallah, Professor Christos G. Christodoulou and Professor Guoyi Zhang, for their help and support. I am very thankful for them serving as my committee members.

I would like to thank my lab colleagues for making the work environment at UNM a more friendly place. I am so grateful for their collaboration and the time we spent together throughout our journey at UNM. I want also to thank all my friends, especially my Lebanese friends who have been a great support for me during my stay in Albuquerque. Their presence has made my life in Albuquerque a more enjoyable experience.

Last, but not least, I would like to thank my extended family for their support and encouragement. Special thanks to my parents, Tannous and Josephine, for their patience, their sacrifice and their continual support in all my pursuits. Thanks to my siblings, Maya and Gaby, for being so encouraging and supportive during all this period of time.

# Wideband Spectrum Sensing and Signal Classification for Autonomous Self-Learning Cognitive Radios

by

**Mario Bkassiny**

B.E., Electrical Engineering, Lebanese American University, 2008

M.S., Computer Engineering, Lebanese American University, 2009

Ph.D., Electrical Engineering, University of New Mexico, 2013

## Abstract

In this dissertation, we develop a novel cognitive radio (CR) architecture, referred to as the *Radiobot* [1], whose goals go beyond dynamic spectrum access (DSA) to achieve the main features of cognition, notably, self-learning and self-reconfiguration. The proposed CR architecture is based on a sequence of signal processing and machine learning techniques that enable the Radiobot to sense a wide frequency band and act autonomously by learning from past experience. To achieve its goals, the proposed CR is equipped with the following functionalities: 1) Wideband spectrum sensing, 2) non-parametric signal classification, 3) unsupervised learning and reasoning and 4) decentralized decision-making.

To this end, we implement a blind spectrum sensing method based on joint energy/cyclostationary detection. Optimal wideband energy detector is designed based on the Neyman-Pearson (NP) criterion which maximizes the detection probability of primary signals, subject to a certain false alarm rate. Cyclostationary detection is

proposed as a means of extracting the underlying cyclic properties of the detected signals in order to identify the types of signals in each frequency band. Once the signal features are extracted, a Bayesian non-parametric classifier based on the Dirichlet process is applied to determine the different types of wireless systems in the surrounding radio frequency (RF) environment. In this dissertation, we extend the Dirichlet process mixture model (DPMM)-based classifier to allow for a mixture of Gaussian and non-Gaussian vector observation models, compared to existing DPMM's with scalar Gaussian observation models. We also develop a sequential DPMM classifier that can be implemented at a low processing cost, making it suitable for real-time operation. Upon identifying the RF activities in the surrounding environment, the Radiobot uses machine learning techniques for decision-making. Thus, we propose a reinforcement learning (RL) algorithm that enables the Radiobot to learn by interacting with its environment. The learning process is formulated in a decentralized partially observable Markov decision process (DEC-POMDP) framework and is shown to lead to a near-optimal policy with little knowledge about the environment. As a result, using its sensing and learning capabilities, the Radiobot can switch among multiple modes of operation to adapt to a dynamic RF environment.

# Publications

In the following, we present a list of journal and conference publications that have resulted from the work in this dissertation.

## Journal Publications

1. M. Bkassiny and S. K. Jayaweera, “Robust Wideband Spectrum Sensing for Cognitive Radios”, *IEEE Transactions on Wireless Communications*, 2013, [In preparation].
2. M. Bkassiny and S. K. Jayaweera, “Sequential Bayesian Non-parametric Signal Classification Algorithms for Wideband Cognitive Radios”, *IEEE Transactions on Wireless Communications*, May 2013, [In review].
3. M. Bkassiny, S. K. Jayaweera and Y. Li, “Multidimensional Dirichlet Process-based Non-Parametric Signal Classification for Autonomous Self-Learning Cognitive Radios”, *IEEE Transactions on Wireless Communications*, May 2012, [Second round review].
4. M. Bkassiny, Y. Li and S. K. Jayaweera, “A survey on machine-learning techniques in cognitive radios”, *IEEE Communications Surveys and Tutorials*, 2012, DOI: 10.1109/SURV.2012.100412.00017.

## Publications

5. M. Bkassiny, Y. Li, G. El-Howayek, S. K. Jayaweera and C. G. Christodoulou, “Recent Patents on Spectrum Sensing Methods and RF Architectures for Cognitive Radios”, *Journal of Recent Patents on Computer Science, Special Issue on Recent Advances in Cognitive Radio Communications, Bentham Science Publishers Ltd.*, vol. 5, no. 2, pp. 83-92, Aug. 2012.
6. M. Bkassiny, S. K. Jayaweera, Y. Li, and K. A. Avery, “Wideband Spectrum Sensing and Non-Parametric Signal Classification for Autonomous Self-Learning Cognitive Radios”, *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2596-2605, July 2012.
7. Y. Li, S. K. Jayaweera, M. Bkassiny, and K. A. Avery, “Optimal myopic sensing and dynamic spectrum access in cognitive radio networks with low-complexity implementations”, *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2412-2423, July 2012.
8. S. K. Jayaweera, M. Bkassiny, and K. A. Avery, “Asymmetric cooperative communications based spectrum leasing via auctions in cognitive radio networks”, *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2716-2724, Aug. 2011.
9. Y. Tawk, M. Bkassiny, G. El-Howayek, S. K. Jayaweera, and C. G. Christodoulou, “Reconfigurable front-end antennas for cognitive radio applications”, *IEE Proceedings IET Microwaves, Antennas and Propagation*, Nov. 2010.

## Conference Publications

1. M. Bkassiny, S. K. Jayaweera, Y. Li and K. A. Avery, “Blind cyclostationary feature detection based spectrum sensing for autonomous self-learning cogni-

## Publications

- tive radios”, *IEEE International Conference on Communications (ICC '12)*, Ottawa, Canada, June 2012.
2. S. K. Jayaweera, Y. Li, M. Bkassiny, C. G. Christodoulou and K. A. Avery, “Radiobots: Architecture, Algorithms, and Realtime Reconfigurable Antenna Designs for Autonomous, Self-learning Future Cognitive Radios”, *IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS '11)*, Chiangmai, Thailand, Dec. 2011.
  3. M. Bkassiny, S. K. Jayaweera, Y. Li, and K. A. Avery “Optimal and Low-complexity Algorithms for Dynamic Spectrum Access in Centralized Cognitive Radio Networks with Fading Channels,” *IEEE Vehicular Technology Conference (VTC-spring 2011)*, Budapest, Hungary, May 2011.
  4. Y. Li, S. K. Jayaweera, M. Bkassiny and K. A. Avery “Optimal Myopic Sensing and Dynamic Spectrum Access with Low-Complexity Implementations,” *IEEE Vehicular Technology Conference (VTC-spring 2011)*, Budapest, Hungary, May 2011.
  5. M. Bkassiny, S. K. Jayaweera, and K. A. Avery “Distributed Reinforcement Learning Based MAC Protocols for Autonomous Cognitive Secondary Users,” *Wireless and Optical Communications Conference (WOCC 2011)*, Newark, NJ, Apr. 2011.
  6. S. K. Jayaweera and M. Bkassiny, “Learning to thrive in a leasing market: an auctioning framework for distributed dynamic spectrum leasing (D-DSL),” *IEEE Wireless Communications and Networking Conference (WCNC 2011)*, Cancun, Mexico, Mar. 2011.
  7. M. Bkassiny and S. K. Jayaweera, “Optimal channel and power allocation for secondary users in cooperative cognitive radio networks,” *2nd Interna-*

*Publications*

*tional Conference on Mobile Lightweight Wireless Systems (MOBILIGHT),*  
Barcelona, Spain, May 2010. [Invited Paper]

# Contents

<b>Publications</b>	<b>ix</b>
<b>List of Figures</b>	<b>xviii</b>
<b>Glossary</b>	<b>xxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Wideband Spectrum Sensing . . . . .	4
1.2 Signal Classification . . . . .	8
1.3 Learning and Reasoning . . . . .	11
1.4 Dissertation Contributions . . . . .	13
1.5 Structure of the Dissertation . . . . .	16
1.6 Notation . . . . .	17
<b>2 Wideband Spectrum Sensing</b>	<b>18</b>
2.1 Introduction . . . . .	18

## Contents

2.2	System Model . . . . .	20
2.2.1	Observed Signals Model . . . . .	23
2.2.2	Detection of RF Activities . . . . .	25
2.2.3	Spectral Correlation Function of Multiple Superposed Digital Signals . . . . .	28
2.2.4	Feature Extraction: Baud Rate and Coding Properties . . . . .	29
2.3	Impact of Channel Fading on the Cyclostationary Features . . . . .	30
2.4	Impact of the Doppler Shift on the Detected Carrier Frequencies . . . . .	34
2.5	Self-Reconfiguration of the Spectrum Sensing Module . . . . .	35
2.6	Simulation Results . . . . .	37
2.7	Conclusion . . . . .	41
<b>3</b>	<b>Machine Learning in CR's</b>	<b>42</b>
3.1	Introduction . . . . .	42
3.2	Need of learning in CR's . . . . .	47
3.2.1	Definition of the Learning Problem . . . . .	47
3.2.2	Unique Characteristics of CR Learning Problems . . . . .	48
3.2.3	Types of Learning Paradigms: Supervised versus Unsupervised Learning . . . . .	51
3.2.4	Learning Problems in CR . . . . .	52
3.3	Decision-making in CR's . . . . .	55

*Contents*

3.3.1	Centralized Policy-making under Markov States: Reinforcement Learning . . . . .	55
3.3.2	Centralized Policy-making with Non-Markovian States: Gradient-policy Search . . . . .	61
3.3.3	Decentralized Policy-making: Game Theory . . . . .	65
3.3.4	Decision Rules under Uncertainty: Threshold-learning . . . . .	70
3.4	Feature Classification in CR's . . . . .	71
3.4.1	Non-parametric Unsupervised Classification: The DPMM . . . . .	71
3.4.2	Supervised Classification Methods in CR's . . . . .	77
3.5	Centralized and Decentralized Learning in CR . . . . .	85
3.6	Conclusion . . . . .	88
<b>4</b>	<b>Bayesian Non-Parametric Classification using the Dirichlet Process</b>	<b>90</b>
4.1	Introduction . . . . .	90
4.2	Data Clustering based on the DPMM and the Gibbs Sampling . . . . .	93
4.2.1	DPMM-based Clustering with a Gaussian Observation Model . . . . .	95
4.2.2	DPMM-based Clustering with a Mixture of Gaussian and non-Gaussian Priors for $\theta_i$ . . . . .	97
4.2.3	Prior and Posterior Distributions for $\alpha_0$ . . . . .	102
4.3	Bayesian Prediction (Density Estimation) of the Observation Variables	103
4.4	Convergence of the DPMM-based Classification Algorithm . . . . .	106

## Contents

4.5	Mean-Squared Error Analysis of the Estimated Cluster Means . . . . .	107
4.6	A Low-complexity Biased Parameter Selection Policy for Gibbs Sampling	109
4.7	A Sequential Gibbs Sampler for DPMM Classifiers . . . . .	111
4.8	Simulation Example: Signal Classification in the ISM Band . . . . .	112
4.8.1	Simulation Example: Performance of the Simplified and Sequential DPMM Classifiers . . . . .	118
4.9	Conclusion . . . . .	126
<b>5</b>	<b>Distributed Reinforcement Learning for CRN's</b>	<b>127</b>
5.1	Introduction . . . . .	127
5.2	System Model . . . . .	130
5.3	Channel Access Mechanism . . . . .	132
5.4	Sensing Policies of Distributed Secondary Users . . . . .	134
5.4.1	The Reward and Value Functions . . . . .	136
5.4.2	Reinforcement Learning for DEC-POMDP . . . . .	136
5.5	Simulation Results . . . . .	137
5.6	Conclusion . . . . .	141
<b>6</b>	<b>Summary of the Dissertation and Research Directions</b>	<b>144</b>
6.1	Summary of the Dissertation . . . . .	144
6.2	Future Research Directions . . . . .	146

*Contents*

<b>A Derivation of the ROC for Carrier Frequency Detection</b>	<b>149</b>
<b>References</b>	<b>152</b>

# List of Figures

1.1	A block diagram of the proposed CR model. . . . .	4
2.1	The actions of the cognitive engine: Sensing and PHY/MAC recon- figurations. . . . .	21
2.2	The wide spectrum of interest is divided into $N$ disjoint wide sub- bands for the purpose of sequential processing. . . . .	22
2.3	The cyclostationarity based RF signal detection with a scanning su- perheterodyne receiver. . . . .	24
2.4	Carrier frequencies are estimated as the midpoints of the intersections between the PSD curve and the threshold line. . . . .	27
2.5	Comparison between the ROC's of the sliding-window and conven- tional energy detections. The sliding-window length is $L = 11$ . . . .	38
2.6	Probability of identification of feature points with a sampling fre- quency $f_s = 200MHz$ and sliding-window length $L = 59$ . The de- tected signal is a 4-QAM with symbol rate of 5 Mbauds and down- converted to a carrier frequency of $20MHz$ . The performance is compared under both non-fading and Rayleigh fading channels. . . .	39

*List of Figures*

2.7	Learning curves of the cyclic sub-profile threshold $\zeta$ and the false alarm probability $P_f$ with a learning rate $\psi = 0.2$ and a desired false alarm probability $\phi$ (with $L = 11$ ). . . . .	40
2.8	Detection of multiple users in 2 separate sub-bands. . . . .	40
3.1	An intelligent design can transform the acquired information into knowledge by learning. . . . .	48
3.2	The cognition cycle of an autonomous CR (referred to as the Radiobot) [1]. Decisions that drive Actions are made based on the Observations and Learnt knowledge. The impact of actions on the system performance and environment leads to new Learning. The Radiobot's new Observations are guided by this Learnt Knowledge of the effects of past Actions. . . . .	49
3.3	Supervised and unsupervised learning approaches for CR's. . . . .	49
3.4	Typical problems in CR and their corresponding learning algorithms. . . . .	52
3.5	The RL cycle: At the beginning of each learning cycle, the agent receives a full or partial observation of the current state, as well as the accrued reward. By using the state observation and the reward value, the agent updates its policy (e.g. updating the Q-values) during the learning stage. Finally, during the decision stage, the agent selects a certain action according to the updated policy. . . . .	58
3.6	One realization of the Dirichlet process. . . . .	73
3.7	A diagram showing the basic idea of SVM: optimal separation hyperplane (solid red line) and two margin hyperplanes (dashed lines) in a binary classification example; Support vectors are bolded. . . . .	83

*List of Figures*

3.8 A comparison among the learning algorithms that are presented in this survey. . . . . 86

4.1 Signal Classification of 2 WiFi and a Bluetooth signal. The feature point is denoted by  $(f_c, \alpha, B)$ , where  $f_c$  is the carrier frequency,  $\alpha$  is the cyclic frequency component corresponding to the symbol rate and  $B$  is the estimated bandwidth. Energy detection is applied for  $30\mu s$  at an SNR of 5 dB with Rayleigh fading (fast fading). The probability of correct classification is 100% after 20000 Gibbs sampling iterations. 114

4.2 K-means classification with  $K = 4$  gives a classification accuracy of 100% with arbitrary initialization of centroid locations. . . . . 115

4.3 K-means classification with  $K = 4$  gives a classification accuracy of 79.41% with a different initialization of centroid locations. . . . . 115

4.4 X-means classification with estimated  $X = 12$  gives a classification accuracy of 55.88%. . . . . 116

4.5 DPMM-based classification with estimated  $K = 4$  gives a classification accuracy of 100% . . . . . 116

4.6 Signal Classification of 2 WiFi and a Bluetooth signal. The feature point is denoted by  $(f_c, B)$ , where  $f_c$  is the carrier frequency and  $B$  is the estimated bandwidth of the signal. Energy detection is applied for  $30\mu s$  at an SNR of 5 dB with Rayleigh fading (fast fading). The probability of correct classification is 100% after 5000 Gibbs sampling iterations. . . . . 118

*List of Figures*

4.7 DPMM classification of four different QAM signals at a received  $SNR = 10dB$  using Algorithm 4. Feature vectors assigned to the same cluster are represented with identical markers. Classification accuracy is equal to 100%. . . . . 119

4.8 Convergence of the DPMM classifier using random uniform, Round-Robin and random biased parameter selection policies for Gibbs sampling. . . . . 120

4.9 Sequential DPMM classification of different QAM signals at a received  $SNR = 10dB$  using Algorithm 5. Classification accuracy is equal to 100%. . . . . 121

4.10 X-means classification of different QAM signals at a received  $SNR = 10dB$ . Classification accuracy is equal to 71%. . . . . 122

4.11 Misclassification rates of the DPMM algorithm for different SNR levels. Misclassification rates of both training and sequential stages are obtained over 25 sensing intervals for each stage (denoted as "DPMM Training" and "DPMM Sequential", respectively). The overall misclassification rate of the DPMM, including both training and sequential stages, is denoted as "Overall DPMM". Four different QAM signals are being transmitted simultaneously and are sampled at a rate  $f_s = 200$  MHz for a duration  $T = 50\mu s$  in each sensing interval. . . . 124

*List of Figures*

4.12	Misclassification rates of the DPMM and X-means algorithms for different SNR levels. Overall DPMM misclassification rates are obtained over 50 sensing intervals, including both training and sequential stages. Misclassification rates of the X-means are also obtained over the whole 50 sensing intervals. Combined feature extraction errors and signal classification errors are computed using DPMM and X-means classifiers, and are denoted as "Overall DPMM + Feature Extraction" and "X-means + Feature Extraction", respectively. Four different QAM signals are being transmitted simultaneously and are sampled at a rate $f_s = 200$ MHz for a duration $T = 50\mu s$ in each sensing interval. . . . .	125
5.1	Cognitive radio network with distributed secondary nodes . . . . .	131
5.2	Channel Access Policies . . . . .	132
5.3	Sensing and Updating the Beliefs . . . . .	135
5.4	Average Utilization of Primary channels for $\alpha_0 = 0.1$ . . . . .	139
5.5	Average Utilization of Primary channels for $K_s = 3$ . . . . .	139
5.6	Collision rates with Primary channels for $K_s = 3$ . . . . .	140

# Glossary

ACS	Almost-cyclostationary
ADC	Analog-to-digital converter
AIC	Akaike information criterion
AM	Amplitude modulation
AMC	Automatic modulation classification
ANN	Artificial neural network
BER	Bit-error rate
BIC	Bayesian information criterion
BPSK	Binary phase shift keying
BSC	Binary symmetric channel
CR	Cognitive radio
CRN	Cognitive radio network
CRP	Chinese restaurant process
DEC-POMDP	Decentralized partially observable Markov decision process

## Glossary

DPMM	Dirichlet process mixture model
DFT	Discrete Fourier transform
DSA	Dynamic spectrum access
DSL	Dynamic spectrum leasing
DSS	Dynamic spectrum sharing
FD	Finite difference
FFT	Fast Fourier transform
FM	Frequency modulation
GMM	Gaussian mixture model
GMSK	Gaussian minimum shift keying
HMM	Hidden Markov model
IF	Intermediate frequency
i.i.d.	independent identically distributed
LTV	Linear time-variant
MAC	Medium access control
MAP	Maximum <i>a posteriori</i> probability
MCMC	Markov chain Monte Carlo
MDP	Markov decision process
MFNN	Multilayered feed-forward neural network

## *Glossary*

MLP	Multilayer perceptron
MSE	Mean-squared error
NG	Natural policy gradient
NP	Neyman-Pearson
NRM	Network reconfiguration manager
ODAL	Observe-Decide-Act-Learn
OSA	Opportunistic spectrum access
pdf	Probability density function
pmf	Probability mass function
POMDP	Partially observable Markov decision process
PSD	Power spectral density
QAM	Quadrature amplitude modulation
QoS	Quality of Service
RF	Radio frequency
RL	Reinforcement learning
ROC	Receiver operating characteristic
RRM	Radio resource management
SCF	Spectral correlation function
SDR	Software-defined radio

## *Glossary*

SNR	Signal-to-noise ratio
SSDC	Secondary system decision center
SVM	Support vector machine
TD	Temporal difference
TDMA	Time division multiple access
TRM	Terminal reconfiguration manager
UWB	Ultra-wideband
VPG	Vanilla policy gradient
WRAN	Wireless regional area network
WSS	Wide-sense stationary
WSSUS	Wide-sense stationary uncorrelated scattering

# Chapter 1

## Introduction

Since its inception by Mitola in 1999, the term cognitive radio (CR) has been used to refer to intelligent radio devices that are capable of learning and adapting to their environment [2, 3]. This concept is considered as an evolution of software-defined radios (SDR's) in which most of the signal processing tasks are being handled by general-purpose processors, instead of specific-purpose hardware. However, the transition from SDR's to CR's can be achieved by introducing cognition to the radio devices themselves, making them aware of their radio frequency (RF) environment [2].

In [4], Haykin envisioned CR's to be *brain-empowered* wireless devices that are specifically aimed at improving the utilization of the electromagnetic spectrum. According to Haykin, a CR is assumed to use the methodology of *understanding-by-building* and is aimed at achieving two primary objectives, which are permanent reliable communications and efficient utilization of the spectrum resources [4]. With this interpretation of CR's, a new era of CR's began, focusing on dynamic spectrum access (DSA) techniques to improve the spectrum utilization [4–8]. This led to research on various aspects of communications and signal processing required for DSA

## Chapter 1. Introduction

networks [4, 9–23]. These included underlay, overlay and interweave paradigms for spectrum co-existence by secondary CR's in licensed spectrum bands [7, 24]. In spectrum underlay, a CR is allowed to communicate over a wide frequency band below a certain power level such that it does not cause harmful interference to existing primary users. This paradigm can be implemented using power control techniques which ensure that the interference caused by secondary users to the primary receivers is below a certain interference cap [25, 26]. In spectrum overlay, however, CR's are assumed to know the primary message and they can use this knowledge in order to reduce the interference at the primary and secondary receivers using sophisticated implementation techniques. For example, spectrum overlay can be implemented in an asymmetric cooperative architecture in which the secondary transmitter spends a portion of its power to transmit its own signal, while the other portion is dedicated to relay the primary signal to its destination [7, 24, 27]. On the other hand, a CR operating in spectrum interweave performs spectrum sensing to determine where and when it may transmit. By locating the spectrum holes<sup>1</sup>, secondary users can access idle primary channels in the absence of primary users. To be specific, the interweave mode permits secondary users to efficiently utilize the unused spectrum holes, while avoiding, or limiting, collisions with primary transmissions. This technique was envisioned by the DARPA Next Generation (XP) program and it was denoted as the opportunistic spectrum access (OSA) [24].

To perform its cognitive tasks, a CR should be aware of its RF environment. It should sense its surrounding environment and identify all types of RF activities. Thus, spectrum sensing was identified as a major ingredient in CR's [28, 29]. Many sensing techniques have been proposed over the last decade based on matched filter, energy detection, cyclostationary detection, wavelet detection and covariance detection [18, 29–36]. In addition, cooperative spectrum sensing was proposed as a

---

<sup>1</sup>A spectrum hole is a licensed spectrum band that is owned by a primary system and is not utilized at a specific time and in a particular region.

## *Chapter 1. Introduction*

means of improving the sensing accuracy by addressing the hidden terminal problems inherent in wireless networks in [20, 21, 30, 32, 37–39].

In this dissertation, however, we focus on blind autonomous wideband sensing techniques since we are interested in detecting signals with unknown characteristics. This type of spectrum sensing applications is suitable for CR's that are operating over wide frequency bands in unknown RF environments in which multiple signals can be transmitted simultaneously at multiple unknown center frequencies [40, 41]. Thus, spectrum sensing is identified as the first stage in the signal processing chain and can be followed by signal classification and decision-making methods. In this work, we implement a signal processing and decision-making framework for CR's, as illustrated in Fig. 1.1. This architecture depicts the main functions of the proposed CR system and shows the structural dependence among these functions. Clearly, the first step of the signal processing chain starts with spectrum sensing which detects the active signals within a wide frequency band of interest. Spectrum sensing, itself, encompasses multiple signal processing capabilities, including energy detection, cyclostationary detection and matched-filter detection. Obviously, these detection methods have different processing costs and can achieve different performance levels. However, by using an SDR platform, all of these detection methods can be implemented on a single CR platform, thus enabling a wide range of signal detection capabilities. After detecting the active signals and extracting their corresponding features, signal classification is performed to identify the number and types of wireless systems in the surrounding environment. In the absence of any prior knowledge about the number of wireless systems, we refer to unsupervised non-parametric approaches to perform signal classification [42]. Our proposed non-parametric approach is implemented based on the Dirichlet process mixture model (DPMM) framework which is able to infer certain hidden characteristics about the surrounding environment [42, 43]. Both spectrum sensing and signal classification outcomes are used to construct an RF mapping of the on-going RF activities. This RF mapping characterizes the state of

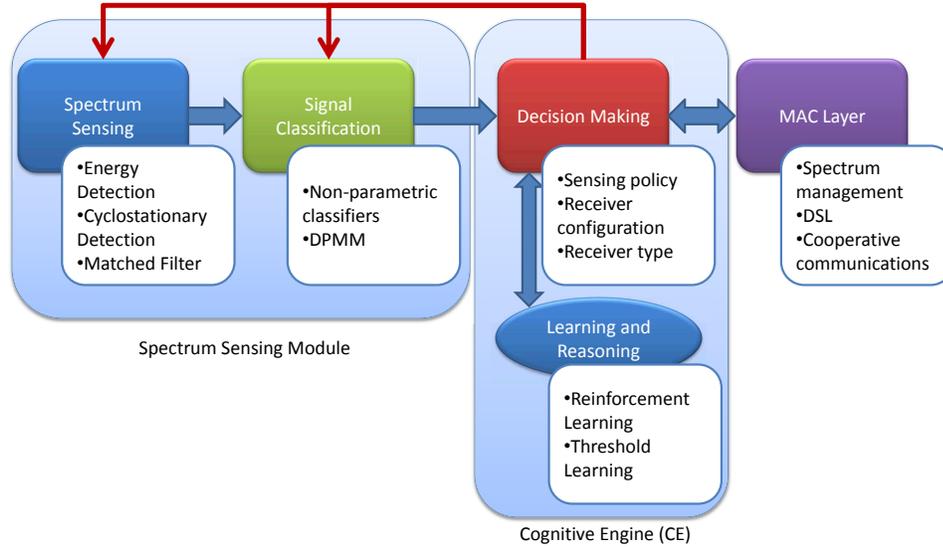


Figure 1.1: A block diagram of the proposed CR model.

the environment and assists the CR in decision-making. Decision-making is located at the heart of the cognitive engine which provides the required artificial intelligence tools for proper cognitive operation [44–49]. The decision-making policies are then used to control the different modules of the CR architecture.

In the following, we introduce each of the above mentioned CR functions and outline the main aspects of this novel CR architecture.

## 1.1 Wideband Spectrum Sensing

Wideband spectrum sensing has been addressed in recent CR applications such as [40, 41, 50–59]. Such wideband capability enables a CR to operate over a wide frequency band, thus improving the efficiency of DSA in exploiting a wide range of frequency bands. Furthermore, a CR equipped with wideband sensing and com-

## *Chapter 1. Introduction*

munication abilities can achieve multi-operability over widely spaced frequencies, as proposed in [1, 50, 60]. However, wideband operations present a real challenge for CR implementations. These challenges are manifested in the RF front-end design as well as in the wideband signal processing implementation [24, 40, 41, 50, 60–62]. In order to understand the nature of these limitations, we should first note that sensing a wide frequency band requires a high sampling rate since wideband RF signals are to be sampled at least at double their bandwidths in order to avoid aliasing while satisfying the Nyquist rate requirement. The high sampling rate requirements of wideband signals incur high power consumption and high quantization error at the analog-to-digital converter (ADC) [63, 64]. In addition, certain wideband signals may require sampling rates that may not be achievable by current state-of-the-art ADC's [63, 64].

In order to address the signal processing challenges of wideband spectrum sensing, compressive sensing (or compressed sampling) has been proposed as a means of sampling a wide frequency band at sub-Nyquist rates [51, 58, 65]. This approach is found to be successful in reducing the sampling rate requirements of ADC's. However, compressive sensing assumes sparse signals in the frequency domain, which may not be a valid assumption, in general [65]. Hence, a more general approach is required to address this issue in a more realistic and practical way.

On the other hand, certain wideband limitations are due to the RF front-end design itself, prohibiting wideband operation. More precisely, most of the RF components are designed to operate on certain nominal frequencies. These RF components may experience performance degradation when the operating frequency changes drastically over a wide frequency range. This is a major issue that should be accounted for when designing RF front-ends for wideband operation. Hence, reconfigurable RF components and reconfigurable antennas are proposed as a solution to allow radio devices to operate over a wide frequency range [24, 61, 62]. With these reconfig-

## *Chapter 1. Introduction*

urable abilities, CR's can switch among widely separated frequency bands, without compromising their performance.

Our approach to wideband spectrum sensing considers both signal processing and RF front-end characteristics of CR's. Hence, our proposed wideband spectrum sensing algorithms are designed in light of existing wideband RF front-end architectures, thus leading to a realistic signal processing framework for wideband operation [40,50,61,62,66]. Our proposed method is based on segmenting a wide frequency band of interest into smaller sub-bands that could be sensed, sequentially, at a lower rate, compared to the high sampling rate that might be required for the original unsegmented wide frequency band [40]. Thus, we bring the sampling rate requirements within a feasible range that can be achieved by existing ADC's. On the RF front-end side, we assume a reconfigurable antenna that can switch among different frequency bands to cover the desired frequency sub-bands, as in [61,62,66]. By matching the desired frequency sub-bands with the reconfigurable antenna frequency range, we can achieve proper sensing and sampling of the RF signals in each of the sub-bands of interest. Hence, having a reconfigurable hardware at this stage is essential to ensure proper operation of the RF front-end over a wide frequency range. However, we should note that the only limitation of this model is that different sub-bands can not be sensed simultaneously, which requires special sub-band selection policies. This problem, however, is out of the scope of this dissertation and is addressed in [67].

When the CR operates in a particular sub-band, it receives the corresponding RF signal, down-converts it into either baseband or intermediate frequency (IF) and then samples it using the ADC [40]. At this stage, digital signal processing methods will be applied to identify both number and types of the existing signals in the corresponding sub-band. Note that, the sub-bands are still considered as wide frequency bands and they may include multiple signals at different center frequencies within the whole sensed sub-band. Hence, most of the existing signal detection methods,

## *Chapter 1. Introduction*

such as energy detection and cyclostationary detection, which assume single narrow-band signals, can not be applied at this stage [22, 68, 69]. Thus, in this dissertation, we propose a joint energy/cyclostationary detection algorithm that is able to detect the active signals within each wide sub-band and to identify their characteristics using cyclostationary detection [40, 41, 70]. The energy detection will be implemented using a smoothed periodogram method and is able to detect the center frequencies of the active signals, subject to a certain desired false alarm probability [41]. Our proposed energy detector will be designed based on the Neyman-Pearson (NP) criterion to maximize the signal detection probability, while satisfying a certain false alarm probability constraint. Once the center frequencies are determined, cyclostationary detection will be applied to identify certain characteristics about each detected signal [40, 70–73]. Note that, several cyclostationary detection algorithms have been proposed for detecting only a single signal within the detected RF waveform [22, 68, 69]. In this dissertation, however, we propose a cyclostationary detection algorithm that is able to detect simultaneously multiple signals within a wide frequency band, while identifying the cyclic features of each signal. This leads to an efficient implementation of cyclostationary detection for wideband spectrum sensing. Note that, a cyclostationary detector can identify underlying periodicities in both analog and digital signals [72, 73]. Such periodicities are due to carrier frequencies, symbol rates, coding rates, etc. Furthermore, one of the main advantages of the cyclostationary detector is its immunity to stationary noise processes, which makes it robust in the low signal-to-noise ratio (SNR) regime [70–73].

In addition to its wideband characteristics, the proposed energy/cyclostationary detector is considered as a blind detector since it does not require any prior knowledge about the RF environment. This is a fundamental characteristic that makes our proposed detector suitable for autonomous CR applications. In particular, the proposed energy detector is considered as a blind detector since it does not require any prior knowledge about the candidate center frequencies nor the signals' band-

widths. By using the NP-based smoothed periodogram, the energy detector is able to detect the active channels within a wide frequency band. On the other hand, the cyclostationary detector is considered a blind detector since it does not require any prior knowledge about candidate cyclic frequencies, as opposed to existing cyclostationary detectors such as [22,34,68,69]. Furthermore, in the presence of simultaneous RF transmissions, our proposed cyclostationary detection algorithm is able to separate and identify the corresponding cyclic frequencies of each signal, which makes it superior to existing detectors. Hence, the proposed spectrum sensing algorithm can be used to extract special features from the detected signals, which can be used to construct an RF mapping of the signal activity in the wide frequency band of interest, as we shall discuss next.

## **1.2 Signal Classification**

After detecting the active signals and identifying their features and characteristics, a CR may construct an RF mapping of the on-going RF activity in order to learn certain characteristics about its RF domain. The RF mapping is assumed to accumulate the acquired knowledge about the environment over time, which can help to assist a CR in its future decision-making. Such RF mapping can be constructed based on feature vectors that are extracted from the sensed signals. By using appropriate probabilistic methods, the extracted feature vectors can be used to infer certain properties about the environment. In particular, if the feature vectors are assumed to be drawn from a mixture model, then classification methods can be used to identify the feature vectors or RF signals that belong to a certain wireless system. As a result of a properly designed classification process, the CR can identify whether a certain detected signal belongs to a known system or to an alien interferer or jammer [74–76]. Such knowledge can help a CR to decide whether to access or avoid a

certain frequency band or channel.

Several feature classification methods have been proposed in the literature. For example, [68] proposed a feature classification algorithm based on neural networks [77] and support vector machines (SVM's) [78], but these methods required training data to initialize the classifiers' parameters. On the other hand, parametric classifiers were proposed based on the Gaussian mixture model (GMM) or K-means algorithm, which do not require training data. However, these techniques assume a fixed number of clusters, which may not be known in general. As an alternative, the authors in [79] proposed to use the X-means algorithm [80] for unsupervised signal classification when the number of clusters is unknown. This approach is based on the K-means algorithms but approximates the number of clusters  $X$  by maximizing either the Bayesian information criterion (BIC) or the Akaike information criterion (AIC) [80]. However, similarly to the K-means algorithm, the X-means algorithm assumes spherical Gaussian data, which does not offer enough flexibility when dealing with observations having an arbitrary noise distribution [80]. Moreover, the K-means algorithm can only converge to a local minimum of the distortion measure and its performance heavily depends on the choice of initial center points [80].

To resolve these drawbacks, we resort to non-parametric classification approaches, in particular, the DPMM which assumes no prior knowledge about the number of clusters [43]. Note that, the DPMM-based classifier is considered to be a Bayesian non-parametric method in the sense of allowing the structure of the model (i.e. number of clusters) to grow with the complexity of the data [43, 81–84]. However, the individual observations of the DPMM can still be drawn from parametric distributions. The DPMM-based classifier can infer the number of clusters (or mixture components) from the data itself, making it a suitable candidate for unsupervised and autonomous classifiers in CR applications. This approach has been previously applied for galaxy clustering [85], speaker diarization [86], speaker adaptation [87],

## Chapter 1. Introduction

image segmentation [88] and compressive sensing [89]. However, in most of these applications, it is assumed that the observation or feature vectors are drawn from a GMM, the condition that may not be justified in wideband spectrum sensing since the feature vectors are extracted from highly dynamic wireless environments that are subject to fading and interference effects. Thus, we propose a novel DPMM-based classifier which assumes that the feature vectors are drawn from a mixture of several probability distributions, including both Gaussian and non-Gaussian families. Based on this framework, the DPMM classifier not only selects the optimal cluster for each detected feature, but it also selects the best probability distribution that matches the observation model. Hence, the proposed classification algorithm can find the best observation model that fits the observed data, which may improve the accuracy of the DPMM classifier [42].

Note that, all of the above mentioned DPMM classifiers are implemented using Markov chain Monte Carlo (MCMC) models and require an extensive number of Gibbs sampling iterations to converge to the stationary distribution of the corresponding Markov chain. This makes them computationally prohibitive for real-time CR operation. Hence, in this dissertation, we propose a novel Gibbs sampling algorithm, referred to as the *simplified Gibbs sampler*, which improves the convergence rate of the DPMM classifier, especially for large scale problems [90]. The proposed algorithm is based on a biased *parameter selection policy* that carefully selects specific parameters to be updated at each Gibbs sampling iteration, instead of sequentially or randomly selecting these parameters. Hence, the proposed algorithm is shown to improve the efficiency of the Gibbs sampling-based DPMM classifier and makes a suitable candidate for large-scale classification problems. Furthermore, we propose a *sequential Gibbs sampler* that is more suitable for real-time operation, compared to the *simplified Gibbs sampler* [90]. The proposed sequential Gibbs sampler selects a current observation feature and classifies it into a certain cluster. However, in order to achieve good performance results, the sequential Gibbs sampler requires a training

period, which can be based on the above simplified Gibbs sampler. As a result, the obtained sequential Gibbs sampler ensures real-time classification, which makes it an alternative solution to simple parametric classifiers, yet without requiring additional information about the observation model.

### 1.3 Learning and Reasoning

In addition to being aware of its environment, a CR should be equipped with the abilities of learning and reasoning [1–3, 60]. These capabilities can be achieved through a cognitive engine which forms the core of a CR [44–49]. A cognitive engine uses machine learning tools to coordinate the actions of the CR. However, only in recent years there is a growing interest in applying machine learning techniques to CR's [91–94].

According to [4], a CR is defined as *an intelligent wireless communication system that is aware of its environment and uses the methodology of understanding-by-building to learn from the environment and adapt to statistical variations in the input stimuli*. Based on this definition, it becomes clear that any CR should be equipped with a learning ability allowing it to adapt to its surrounding environment. In this dissertation, we focus on unsupervised learning methods that enable a CR to act autonomously without supervision [95]. By using unsupervised learning techniques, a learning agent can explore the environment and update its policies based on its observations and rewards. In particular, the reinforcement learning (RL) is a type of unsupervised learning techniques that has been used for controlling robots and is recently presented as a promising solution for CR applications [27, 95–99]. Reinforcement learning algorithms allow an agent to learn by *trial-and error* and consist of a combined exploitation/exploration strategy [95]. The exploration strategy helps to avoid locally optimal policies whereas exploitation strategies try to maximize the

## Chapter 1. Introduction

expected return. Thus, the learning agent needs to balance between its exploration and exploitation strategies to behave optimally in its environment.

In our work, we analyze the ability of CR's to learn and adapt in a decentralized multi-agent environment and propose an unsupervised learning algorithm to achieve this goal [96]. This is a challenging problem in machine learning literature since it requires *action coordination* among multiple agents (i.e. multiple CR's), yet without having any information exchange among them [100, 101]. In this setup, the reward of each agent is a function of the joint action of all the agents. However, since each agent only has control over its own actions, it needs to estimate the actions of other agents in order to select the proper action maximizing its reward function. By following an unsupervised learning approach, we show that decentralized CR's are able to reach near-optimal performance, without incurring any control overhead among agents [96].

To be concrete, we consider a decentralized cognitive radio network (CRN) in which several CR's try to access a set of primary channels without colliding with either primary users or other CR's. In the absence of any control channel within the secondary network, each CR needs to acquire a decentralized spectrum sensing policy in order to maximize a certain reward function while satisfying certain quality of service (QoS) requirements. The network is modeled as a decentralized partially observable Markov decision process (DEC-POMDP) in which the primary channels' occupancy is denoted as the system state [96, 102–104]. These states are assumed to evolve according to a discrete-time Markov chain. The actions of CR's define the channels to be sensed at each time period in order to maximize the average utilization of primary channels [96].

The optimal solution of the partially observable Markov decision process (POMDP) can be obtained by using dynamic programming approaches [104]. However, this is a computationally prohibitive approach due to the continuity of the environ-

ment states which are defined as a continuous belief vector [104]. The solution becomes even more complicated with the DEC-POMDP in which each agent does not have complete observation of the other agents, which adds more uncertainty to the decision-making problem [105]. In order to address this problem, we resort machine learning techniques, in particular, the Q-learning algorithm which can lead to satisfactory solutions in DEC-POMDP frameworks with little knowledge about the system environment [96, 99, 106]. In particular, we will show that, under certain conditions, the Q-learning algorithm, which is a type of unsupervised RL algorithms, can lead to a near-optimal solution for the channel access problem in decentralized CRN's [96, 107]. More importantly, this solution is obtained without any supervision by external agents nor communications among CR's, which makes it suitable for autonomous CR operation.

## 1.4 Dissertation Contributions

The main contributions of this dissertation can be summarized as follows:

- We propose a novel CR architecture that is aimed at wideband operation in alien RF environments. The proposed model is able to sense a wide frequency band of interest and detect the ongoing RF activities, without any prior knowledge about the active signals. This is achieved using a joint energy/cyclostationary detection method to identify the center frequencies and cyclic frequencies of the detected signals. In contrast with similar cyclostationary detection method, our approach assumes no prior knowledge about the candidate cyclic frequencies and it is able to detect simultaneously multiple signals within a wide frequency band.
- We design a wideband energy detector for CR's based on the NP criterion to

## Chapter 1. Introduction

maximize the detection probability of active signals, subject to a certain false alarm requirement. The energy detector is implemented using a sliding-window technique, which is shown to increase the detection probability for a given false alarm rate. Hence, the sliding-window energy detector can overcome the poor detection performance of energy detectors in CR applications.

- We derive an analytical expression for the decision threshold of the sliding-window energy detector in *frequency-domain*. This threshold is applied to a smoothed periodogram, which allows for signal detection in a wide frequency band. This approach is different from existing energy detectors in two ways: First, the frequency-domain decision threshold is derived based on the NP criterion, which maximizes the detection probability of RF signals subject to a certain false alarm rate. In contrast, most of the wideband energy detectors use arbitrary thresholds in frequency-domain, which does not guarantee any desired performance level. Second, optimal thresholds for energy detection are commonly obtained analytically in time-domain using a time-series representation of the signal. Although this method can guarantee a certain optimality criterion, it can not be used in wideband applications in which the signals activity is not homogeneous over the wide frequency band of interest. Hence, by properly designing our energy detector in frequency-domain, we optimize the signal detection performance for wideband applications.
- We design a cyclostationary detection algorithm that is able to identify, simultaneously, the cyclic frequencies of multiple RF signals in a wide frequency band. This is achieved by first establishing the superposition property of cyclostationary processes and then defining a cyclic sub-profile for each detected signal. Thus, each cyclic sub-profile represents the cyclic properties (or RF signature) of a single RF signal, which allows to separate the cyclic frequency components of multiple superposed signals in a wide frequency band.

## Chapter 1. Introduction

- In this dissertation, we focus on the importance of machine learning in CR design and present a wide spectrum of machine learning techniques that can be applied in this context. We classify these learning algorithms according to a hierarchical representation showing the conditions under which each approach (or algorithm) can be applied. This hierarchical representation provides the guidelines for selecting the appropriate learning algorithm for a particular situation. We also present a brief and concise description of each machine learning approach, while comparing the advantages and disadvantages of each technique.
- We propose a Bayesian non-parametric signal classifier for CR's based on the DPMM framework. In contrast with similar DPMM-based classifiers, our proposed model is generalized to the multi-dimensional case and is extended to both Gaussian and non-Gaussian observation models. This generalization is important for wideband spectrum sensing and signal classification applications in which the simple Gaussian assumption may not be valid. Furthermore, using an unsupervised non-parametric classifier enables the CR to infer the number of systems (clusters) from the observed data itself. This improves the autonomous abilities of our proposed CR architecture.
- By investigating the underlying properties of the DPMM classifier, we propose a simplified Gibbs sampler to improve the convergence rate of the DPMM classification algorithm. This algorithm is implemented by introducing a parameter selection policy, enabling the Gibbs sampler to select the DPMM parameters more efficiently. On the other hand, the existing Gibbs sampling-based DPMM classifiers suffer from an excessive number of Gibbs sampling iterations and are limited to small-scale applications. By using our proposed simplified algorithm, however, we extend the applications of DPMM classifiers to large-scale problems, while reducing the computational burden of such classifiers in CR's.

- We implement a sequential DPMM classifier that can be used in real-time CR applications. In contrast with the existing DPMM classifiers that assume a fixed number of feature vectors, the sequential DPMM classification algorithm is implemented in a recursive process, allowing for real-time classification of new upcoming signal features. By combining advantages of both simplified and sequential DPMM classifiers, we present a novel non-parametric signal classification framework that allows the CR to autonomously classify the detected RF signals in real-time. Note that, the proposed simplified and sequential DPMM classifiers can be beneficial, not only to signal classification problems, but to general unsupervised feature classification problems, as well.
- We propose a reinforcement learning algorithm to derive a spectrum sensing policy for CR's in a decentralized CRN. The proposed algorithm is implemented in a DEC-POMDP framework, which is known to have a very challenging and untractable solution, in general. By using the Q-learning algorithm, however, we obtain a near-optimal sensing policy with little knowledge about the RF environment. Furthermore, the resulting policy is shown to achieve action coordination among CR users, while limiting the collision probability with primary licensed channels. To the best of our knowledge, this is the first time that the Q-learning algorithm is used in such OSA contexts.

## 1.5 Structure of the Dissertation

The remainder of this dissertation is organized as follows: Chapter 2 proposes a wideband spectrum sensing framework for CR's using a joint energy/cyclostationary detection method. In Chapter 3, we present the state-of-the art machine learning techniques that can be applied to CR's. Next, in Chapter 4, we propose a machine learning technique to perform signal classification based on a non-parametric

## *Chapter 1. Introduction*

Bayesian DPMM formulation. In Chapter 5, another machine learning technique is proposed to obtain a spectrum sensing policy for CR's in a decentralized CRN. The sensing policy is obtained using the Q-learning algorithm by assuming a DEC-POMDP framework. In each of these chapters, we validate our models and algorithms using both analytical and simulated results. Finally, we conclude the dissertation in Chapter 6.

## **1.6 Notation**

Throughout this dissertation, we use bold characters to refer to vector and matrix quantities.

# Chapter 2

## Wideband Spectrum Sensing

### 2.1 Introduction

The Radiobot proposed in [1] is considered as a *rational* radio agent that can interact with its RF environment to achieve several functions such as inter-operability in heterogeneous RF network environments, multi-mode operability and spectrum coexistence with other primary users [1]. A cognitive engine constitutes the *brain* of the Radiobot and coordinates its decision-making actions [1, 40, 44–49, 60]. For example, the cognitive engine may determine the sensing policy, the sensing antenna configurations, spectrum sensing algorithm, etc., for spectrum awareness. A high-level system architecture of a Radiobot is shown in Fig. 2.1 which highlights two major functions of the cognitive engine: 1) Controlling the sensing module and 2) controlling the PHY/MAC communication modules. In order to realize a complete Radiobot system, both autonomous sensing and PHY/MAC decision-making need to be developed. In this chapter, however, we restrict our attention to the spectrum sensing module and develop blind autonomous sensing algorithms that can be adapted, through cognitive learning, to unknown RF environments [40, 41].

Decision-making through machine learning techniques will be addressed in the following chapters.

According to [1], one of the most important abilities of a Radiobot is to be aware of the RF environment in order to self-characterize the best possible communication mode. Spectrum sensing is considered as an essential step towards this end and has been identified as a fundamental requirement for any CR [28, 29]. However, it is not sufficient for a Radiobot to just detect the existence of RF activities in its environment, but also it has to be able to identify the *types* of active signals. For example, if the Radiobot were able to identify a certain signal as a jammer, it might need to avoid it so that it preserves the security and reliability of its communication [74–76]. Hence, detecting the type of signal activity is essential in this context, which requires special signal processing algorithms to identify the type of each wireless signal based on its underlying physical properties.

In order to detect and identify RF activities, we develop a growingly sophisticated signal processing sequence based on blind joint energy/cyclostationary detection [40]. In the first step, energy detection is applied to detect the active carrier frequencies in the frequency range of interest. Next, a cyclostationarity-based feature extraction algorithm is used to detect the cyclic frequency components at each detected carrier frequency. In contrast with similar two-stage spectrum sensing architectures that assume prior knowledge of the primary channels [108, 109], our proposed spectrum sensing does not require any *a priori* knowledge of the existing channels, which makes it a suitable platform for autonomous Radiobots that operate in unknown RF environments. The performance of the carrier frequency detector is evaluated through its receiver operating characteristic (ROC) and the cyclostationary detection is evaluated for a wide range of SNR and for different sensing times.

After each action and/or observation, the Radiobot applies a learning algorithm to improve its future sensing and communications techniques based on its past ex-

perience, as encapsulated in the Observe-Decide-Act-Learn (ODAL) cognition cycle of [1]. Several learning algorithms have been previously applied to CR's for PHY/MAC decision-making. In particular, the RL has been applied for power control [99] and for distributed Medium Access Control (MAC) in CRN's [96,97]. In our case, however, the Radiobot employs a learning algorithm similar to [110], allowing online self-reconfiguration of the spectrum sensing module. The learning algorithm controls the threshold value of the cyclostationary detector to achieve a certain false alarm probability. In [110], the algorithm estimates the false alarm probability during a training period in which the signals are drawn from a null-hypothesis (denoting no signals). In our case, however, by using the energy detection results, the false alarm probability of the cyclostationary detector can be updated during the normal operation of the Radiobot when no signals are detected.

The remainder of this chapter is organized as follows: In Section 2.2, we introduce both the wideband sensing model and the feature extraction method, and analyze the impact of superposed multiple RF signals on the feature extraction operation. In Sections 2.3 and 2.4, we analyze the impact of wireless channel fading on both the cyclostationary and carrier frequency detectors, respectively. In Section 2.5, we present the self-reconfiguration of the sensing module. We show the simulation results in Section 2.6 and conclude this chapter in Section 2.7.

## **2.2 System Model**

The ability to sense the surrounding RF spectrum is crucial to everything a Radiobot can perform and achieve, due to the fact that spectrum sensing measurements are to be used in (a) detecting, identifying and classifying the signals present in the Radiobots RF environment, and (b) making decisions on its operating mode and subsequent sensing. In practice, a critical limitation of spectrum sensing systems

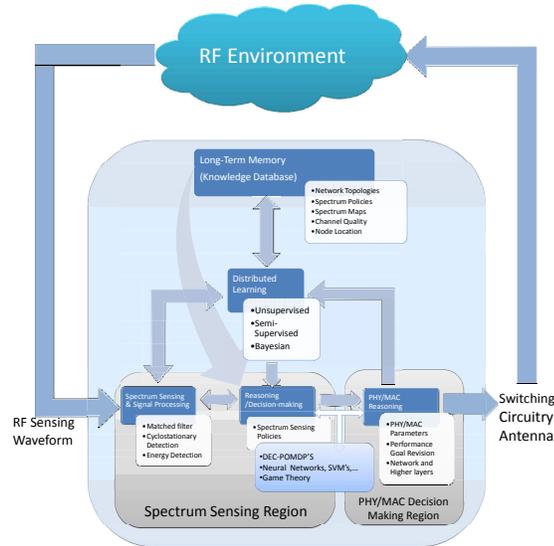


Figure 2.1: The actions of the cognitive engine: Sensing and PHY/MAC reconfigurations.

stems from the sampling hardware and the ADC's [111]. The tradeoff is between the sampling rate and the resolution. For example, recent research has led to an ADC that can sample at a rate of  $16GS/s$  but only with a 6-bit resolution [64]. Better resolutions can only be obtained at the expense of lower sampling rates, as in the case of the  $1GS/s$  ADC *ADS5400* [112] which allows 12-bit resolution. In order to avoid aliasing, the sampling rate is required to be at least as large as the Nyquist frequency. In our case, since the total bandwidth of the spectrum of interest is generally in the scale of several Giga Hertz, it may not be realistic at the current state of the art to expect an ADC to sample, for example, the whole ultra-wideband (UWB) spectrum at a sufficiently high sampling rate with sufficient resolution. A solution is to segment the spectrum of interest into several sub-bands and down-convert each sub-band to baseband or IF for sampling. Another solution for wideband spectrum sensing based on sub-Nyquist sampling was proposed in [113]. However, this technique can only be applied when the signals are sparse, the condition which can hardly be satisfied,

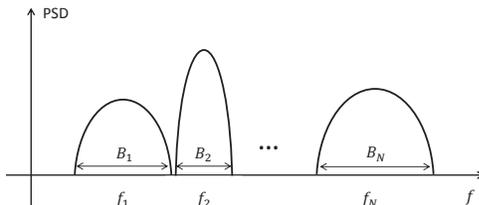


Figure 2.2: The wide spectrum of interest is divided into  $N$  disjoint wide sub-bands for the purpose of sequential processing.

in general.

In addition, as in many wireless mobile communication systems, hardware compactness is a major concern. Hence, it is desirable to reduce the number of RF components and to avoid parallel RF hardware redundancies in any such system. For instance, a communication system may have to be restricted to a limited number of RF mixers used for IF conversions. To address such hardware limitations, we propose a Round-Robin style joint energy/cyclostationary spectrum sensing strategy, which can achieve multi-band operation using a single reconfigurable RF chain. For this, we assume that the RF environment of interest is firstly segmented into a number of  $N$  disjoint, still wide, sub-bands, as shown in Fig. 2.2. By using the Round-Robin strategy, the Radiobot can sequentially switch among these sub-bands to detect the on-going RF activities.

We assume that these sub-bands are arbitrarily centered at frequencies  $f_1, \dots, f_N$ , with bandwidths of  $B_1, \dots, B_N$ . It is expected that this segmentation of the spectrum of interest into sub-bands will essentially be determined by the sensing antenna system in use. For example, the reconfigurable sensing antenna system that was developed in [61], is capable of scanning the UWB spectrum by segmenting it into  $N = 5$  sub-bands. In particular, this wideband sensing antenna was shown to be able to scan the spectrum from  $2GHz$  to  $10GHz$  in  $N = 5$  bands, with  $f_1 = 2.55GHz$ ,  $f_2 = 3.2GHz$ ,  $f_3 = 4.48GHz$ ,  $f_4 = 5.8GHz$ , and  $f_5 = 8.15GHz$  [61].

We assume that the Radiobot system sequentially picks one of the  $N$  sub-bands to sample at each time instant. In order to reduce the requirements on the sampling rate, as shown in Fig. 2.3, a local variable oscillator with frequency  $f_{I_n}$  and a corresponding digital bandpass filter is used to convert the received signal into an IF signal, where we denote by  $f_{I_n}$  the local oscillator frequency tuned for the  $n$ -th sub-band. By sequentially sensing the  $N$  sub-bands, the Radiobot can scan the whole spectrum without requiring parallel hardware nor unrealistic ADC's. Note that, sequential spectrum sensing may lead to certain limitations. For example, if the sensing and processing durations are too long, the Radiobot may miss certain changes in RF conditions in the currently non-sensed sub-bands. On the other hand, short sensing durations may lead to inaccurate sensing results. To address such problems, sub-band selection policies may be designed to determine the optimal selection of frequency sub-bands at each time instant. However, the problem of sub-band selection policies is out of the scope of this dissertation and is being addressed in [67].

Since our proposed detection procedure applies to each of the sub-bands in the same way, in the following, we present the model formulation for a particular sub-band  $n \in \{1, \dots, N\}$ . Hence, for brevity of notation, we drop the frequency sub-band index  $n$  in the following sections.

### 2.2.1 Observed Signals Model

We denote by  $N_s$  the total number of signals at time  $t$  in the sub-band of interest. The corresponding IF signal  $y(t)$  in Fig. 2.3 can be expressed as [114]:

$$y(t) = \Re \left\{ \sum_{l=1}^{N_s} \left( \int_0^\infty x_l(t - \tau) h_l(\tau, t) d\tau \right) e^{j2\pi(f_{c_l} - f_I)t} \right\} + w(t), \quad (2.1)$$

where  $x_l(t)$  denotes the  $l$ -th baseband signal that is to be modulated at a carrier frequency  $f_{c_l}$ . The  $l$ -th baseband equivalent linear time-variant (LTV) impulse re-

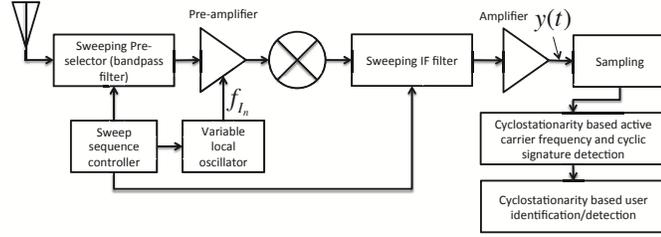


Figure 2.3: The cyclostationarity based RF signal detection with a scanning superheterodyne receiver.

sponse  $h_l(\tau, t)$  denotes the response of the channel at the time  $t$  to an impulse that stimulated the channel between the  $l$ -th signal source and the Radiobot at time  $t - \tau$  [114, 115]. The receiver noise, denoted by  $w(t)$ , is assumed to be a white noise process with double-sided power spectral density (PSD) of  $\frac{N_0}{2}$ . The average noise power at the output of the sweeping IF filter will be  $P_n = N_0 B$ , where  $B$  is the IF filter bandwidth. The resulting SNR at the output of the IF filter will thus be  $SNR = \frac{P_s}{P_n}$  where  $P_s$  is the received signal power.

Note that, for single-path (flat-fading) time invariant channel models, the channel impulse response is equal to  $h_l(\tau, t) \triangleq h_l \delta(\tau - \tau_l)$ , where  $\delta$  denotes the Dirac delta function,  $\tau_l$  is the propagation delay of the single channel's path and  $h_l$  is the complex channel gain. In this case, the received signal can be expressed as:

$$\begin{aligned}
 y(t) &= \Re \left\{ \sum_{l=1}^{N_s} \left( \int_0^{\infty} x_l(t - \tau) h_l \delta(\tau - \tau_l) d\tau \right) e^{j2\pi(f_{c_l} - f_I)t} \right\} + w(t), \\
 &= \Re \left\{ \sum_{l=1}^{N_s} h_l x_l(t - \tau_l) e^{j2\pi(f_{c_l} - f_I)t} \right\} + w(t).
 \end{aligned}$$

## 2.2.2 Detection of RF Activities

In order to detect active RF signals, we propose to identify their carrier frequencies and the associated cyclic frequencies that are induced by the underlying periodicities of those signals. Note that, it is well-known that almost all man-made signals exhibit such underlying periodicities due to, for example, their symbol rates, coding schemes, packet/frame header structures and training symbol sequences, etc. [71–73]. In the following discussion, however, we will explicitly focus on the cyclic properties induced by the symbol and coding rates<sup>1</sup>. Using the discrete-frequency smoothing method [71] described below, we compute an estimate  $\tilde{S}_x^\alpha(t, f)$  of the spectral correlation function (SCF)  $S_x^\alpha(f)$  using a discrete signal  $\{x(t - kT_s)\}_{k=0}^{M-1}$ , for each sub-band, where  $T_s$  is the sampling period, and  $M$  is the number of samples. Note that, this implies that the total time duration over which the particular frequency sub-band was scanned is  $T = (M - 1)T_s$ .

The discrete Fourier transform (DFT)  $\tilde{X}(t, f)$  of the sequence  $\{x(t - kT_s)\}_{k=0}^{M-1}$  is defined in (2.2) over the set of frequencies  $\{-\frac{f_s}{2}, -\frac{f_s}{2} + F_s, \dots, \frac{f_s}{2} - F_s\}$ , where  $f_s = \frac{1}{T_s}$  is the sampling rate and  $F_s = \frac{f_s}{M}$  is the frequency increment and  $a(t)$  is a triangular data tapering window [71].

$$\tilde{X}(t, f) = \sum_{k=0}^{M-1} a(t - kT_s)x(t - kT_s)e^{-j2\pi f(t - kT_s)}. \quad (2.2)$$

An estimate of the SCF can then be obtained as [71] based on the discrete-frequency smoothing method:

$$\tilde{S}_x^\alpha(t, f) = \frac{1}{LT} \sum_{\nu=-(L-1)/2}^{(L-1)/2} \tilde{X}(t, f + \frac{\alpha}{2} + \nu F_s)\tilde{X}^*(t, f - \frac{\alpha}{2} + \nu F_s), \quad (2.3)$$

where  $\alpha$  is the cyclic frequency and  $L$  (an odd number) is the spectral smoothing window length. Note that, (2.3) can be evaluated for discrete values of  $\alpha \in$

---

<sup>1</sup>It is fairly straightforward to generalize the method to include other periodicities that might be present in any given signal.

## Chapter 2. Wideband Spectrum Sensing

$\{0, \pm 2F_s, \pm 4F_s, \dots\}$  since the DFT  $\tilde{X}(t, f)$  is computed at discrete spectral frequencies  $\{-\frac{f_s}{2}, -\frac{f_s}{2} + F_s, \dots, \frac{f_s}{2} - F_s\}$ . Using this method, the SCF is evaluated at spectral frequencies corresponding to the spectral frequency resolution  $F_s = \frac{f_s}{M}$ , while the SCF is computed at cyclic frequencies  $\{0, \pm 2F_s, \pm 4F_s, \dots\}$ , which does not match the cyclic resolution  $F_s = \frac{f_s}{M}$ . However, in order to compute the SCF at cyclic frequencies corresponding to the cyclic frequency resolution, i.e. at  $\alpha \in \{0, \pm F_s, \pm 2F_s, \pm 3F_s, \dots\}$ , we may have to apply zero-padding to the sampled signal sequence  $\{x(t - kT_s)\}_{k=0}^{M-1}$ . However, this modification is not necessary, in general, if the frequency increment  $F_s$  is small enough to resolve the desired cyclic frequencies.

By setting  $\alpha = 0$ , we first obtain an estimation of the PSD of the discrete signal  $\{x(t - kT_s)\}_{k=0}^{M-1}$ :

$$\tilde{S}_x^0(t, f) = \frac{1}{LT} \sum_{\nu=-(L-1)/2}^{(L-1)/2} \left| \tilde{X}(t, f + \nu F_s) \right|^2. \quad (2.4)$$

The active carrier frequencies in the spectrum sub-band of interest is determined by setting a threshold on the above PSD. As shown in Appendix A, the threshold  $\eta_{PSD}$  shown below can be derived based on the NP test [41]:

$$\eta_{PSD} = \frac{\gamma^{-1}(L; (1 - \alpha_F) \Gamma(L)) P_n}{T_s L}, \quad (2.5)$$

where  $\alpha_F$  is the false alarm probability,  $\gamma^{-1}$  is the inverse lower incomplete gamma function (where  $\gamma(k; x) = \int_0^x t^{k-1} e^{-t} dt$  and the inverse is with respect to the second argument),  $\Gamma(k) = \int_0^\infty t^{k-1} e^{-t} dt$  is the gamma function and  $P_n$  is the noise power. A rough estimate of the noise power  $P_n$  can be first obtained from all the frequency components as  $\hat{P}_n = T_s \sum_{f=-f_s/2}^{f_s/2} \tilde{S}_x^0(t, f)$ . Then, a more accurate estimate of  $P_n$  is obtained from the periodogram where no signal has been detected, similar to

---

<sup>2</sup>Note that, the SCF can have the same resolution in both spectral and frequency domain when using the frequency-smoothing approach [71].

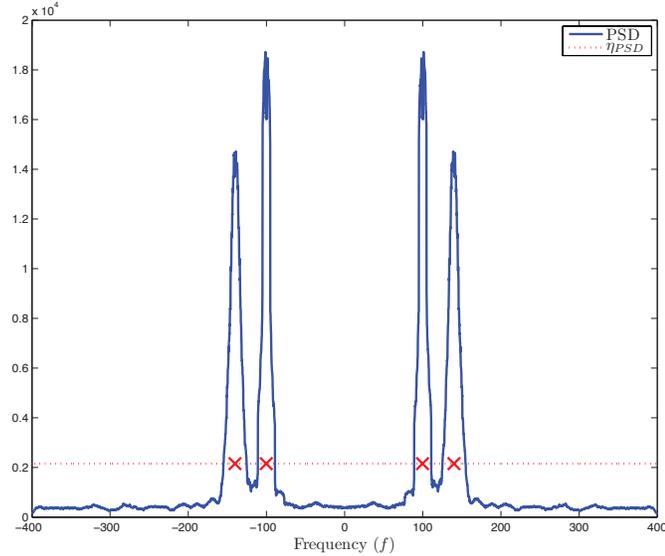


Figure 2.4: Carrier frequencies are estimated as the midpoints of the intersections between the PSD curve and the threshold line.

[116]. Other methods can also be used to estimate  $P_n$ , for example, based on the spectral kurtosis of the smallest values of the periodogram, as proposed in [117]. The impact of noise power uncertainty was discussed and analyzed in [118, 119] where the deterioration of the detector performance was upper-bounded by an expression involving the peak-to-peak range of noise uncertainty [118].

Using the periodogram, the carrier frequencies are estimated as the midpoints of the segments formed by the intersection between the PSD curve and the threshold line  $\eta_{PSD}$ , as shown in Fig. 2.4. We denote by  $\mathcal{A}$  the set of all detected carrier frequencies in the sub-band of interest.

Next, an estimate of the spectral autocohereance function magnitude [71] is computed as:

$$|\tilde{C}_x^\alpha(t, f)| = \frac{|\tilde{S}_x^\alpha(t, f)|}{\sqrt{\tilde{S}_x^0(t, f + \alpha/2)\tilde{S}_x^0(t, f - \alpha/2)}}. \quad (2.6)$$

Note that  $|\tilde{C}_x^\alpha(t, f)|$  is normalized to be between 0 and 1. Due to the fact that for

each carrier, the associated cyclic components show up peaks in a close range of the carrier, we define the *cyclic sub-domain* profile of carrier  $f_c \in \mathcal{A}$  as:

$$\tilde{I}_x(t, \alpha, f_c) = \max_{f \in [f_c - \Delta f_L(f_c), f_c + \Delta f_U(f_c)]} |\tilde{C}_x^\alpha(t, f)|, \quad (2.7)$$

where the lines  $f = f_c - \Delta f_L(f_c)$  and  $f = f_c + \Delta f_U(f_c)$  ( $\forall f_c \in \mathcal{A}$ ) partition the  $(f, \alpha)$ -plane into Voronoi cells whose point sites [120] are located at the detected carrier frequency points  $\{(f_c, 0) : f_c \in \mathcal{A}\}$ .

In [73], it is shown that digital signals exhibit cyclostationarity at multiples of their baud rates. Moreover, the digital signals may exhibit other periodicities as well, for example, due to coding. We denote the RF signature of the signal centered at  $f_c$  as  $\text{RF}(f_c) = \{\alpha \neq 0 : \mathcal{I}_E \tilde{I}_x(t, \alpha, f_c) \geq \zeta\}$ , where  $\mathcal{I}_E$  denotes the indicator function of event  $E = \{\tilde{I}_x(t, \alpha, f_c) \text{ is a local maximum}\}$ , and  $\zeta \in (0, 1)$  is a threshold for the peak detection in the cyclic sub-domain profile.

### 2.2.3 Spectral Correlation Function of Multiple Superposed Digital Signals

In practice, the Radiobot is more likely to deal with multiple RF activities in each spectrum sub-band of interest. Thus, it needs to know the corresponding SCF properties of superposed digital signals, in order to identify the number and types of the detected signals accurately.

In order to analyze the impact of the superposition of multiple signals on the SCF of a signal  $y(t)$ , let us assume that  $y(t) = w(t) + \sum_{m=1}^{N_s} x_m(t)$ , where  $\{x_m(t)\}_{m=1}^{N_s}$  are *independent zero-mean* random processes (denoting  $N_s$  superposed signals) and  $w(t)$  is an independent white noise process with a double-sided PSD of  $\frac{N_0}{2}$ . The autocorrelation function of  $y(t)$  is  $R_{yy}(t, \tau) = \frac{N_0}{2} \delta(\tau) + \sum_{m=1}^{N_s} R_{x_m x_m}(t, \tau)$ , where

## Chapter 2. Wideband Spectrum Sensing

$R_{x_m x_m}(t, \tau)$  is the autocorrelation functions of  $x_m(t)$ , for  $m = 1, \dots, N_s$ . First, we define a Fourier transform for the cyclic autocorrelation function as [121]:

$$\begin{aligned}
 R_{yy}^\alpha(\tau) &\triangleq \lim_{T \rightarrow \infty} \frac{1}{T} \int_{T/2}^{T/2} R_{yy}(t, \tau) e^{-j2\pi\alpha t} dt \\
 &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{T/2}^{T/2} \left[ \frac{N_0}{2} \delta(\tau) + \sum_{m=1}^{N_s} R_{x_m x_m}(t, \tau) \right] e^{-j2\pi\alpha t} dt \\
 &= \frac{N_0}{2} \delta(\tau) \delta(\alpha) + \sum_{m=1}^{N_s} R_{x_m x_m}^\alpha(\tau).
 \end{aligned} \tag{2.8}$$

The SCF of  $y(t)$  can then be expressed as:

$$S_y^\alpha(f) = \int_{\mathbb{R}} R_{yy}^\alpha(\tau) e^{-j2\pi f \tau} d\tau = \frac{N_0}{2} \delta(\alpha) + \sum_{m=1}^{N_s} S_{x_m}^\alpha(f). \tag{2.9}$$

This result shows that the superposition of multiple independent signals results in a superposition of spectral peaks in the  $(f, \alpha)$  domain. In other words, the SCF of the superposition of multiple signals has peaks at cyclic frequencies corresponding to integer multiples of, for example, the data rates of each signal.

### 2.2.4 Feature Extraction: Baud Rate and Coding Properties

The RF signature  $\text{RF}(f_c)$  vector itself can be used as a feature for classifying detected signals. For compactness, it is more convenient, however, to represent this vector by fewer elements. To achieve this, we define two feature elements  $\alpha_1$  and  $\alpha_2$  that are extracted from the RF signature, with  $\alpha_1$  representing the baud rate induced cyclic frequency and  $\alpha_2$  representing the coding induced cyclic frequency. Based on the cyclostationarity properties, the cyclic profile exhibits high peaks at the induced cyclic frequencies  $\alpha_1$  and  $\alpha_2$ . Moreover, since the code length is usually a multiple of the symbol duration, the coding induced cyclic frequency  $\alpha_2$  is smaller than the data rate induced cyclic frequency  $\alpha_1$ . By using this information, in Algorithm 1,

a feature extraction procedure for determining  $\alpha_1$  and  $\alpha_2$  is proposed. Note that, in this algorithm,  $\rho \in (0, 1)$  with  $\rho \gg 0$ . These feature elements can be used to construct certain feature vectors for each detected signal in order to be classified using proper procedures.

---

**Algorithm 1** Feature Extraction Procedure

---

```

for each  $f_c \in \mathcal{A}$  do
     $F = [f_c - \Delta f_L, f_c + \Delta f_R]$ 
     $V_1 = \text{RF}(f_c)$ ,  $M_1 = \arg \max_{\alpha \in V_1} \tilde{I}(t, \alpha, f_c)$ 
     $V_2 = \text{RF}(f_c) \setminus M_1$ ,  $M_2 = \arg \max_{\alpha \in V_2} \tilde{I}(t, \alpha, f_c)$ 
    if  $M_1 < M_2$  then
        if  $\rho \tilde{I}(t, M_1, f_c) > \tilde{I}(t, M_2, f_c)$  then
             $(\alpha_1, \alpha_2) = (M_1, 0)$ 
        else
             $(\alpha_1, \alpha_2) = (M_2, M_1)$ 
        end if
    else
         $(\alpha_1, \alpha_2) = (M_1, M_2)$ 
    end if
end for

```

---

## 2.3 Impact of Channel Fading on the Cyclostationary Features

In this section, we show that the cyclostationary features of signals can essentially be preserved even in the presence of channel fading. In other words, we show that the proposed cyclostationarity based detection method is robust against channel fading effects.

Chapter 2. Wideband Spectrum Sensing

A continuous-time real-valued stochastic process  $x(t)$  is said to be second-order *cyclostationary in the wide sense* if its mean  $\mathbb{E}\{x(t)\}$  and autocorrelation function  $R_{xx}(t, \tau) \triangleq \mathbb{E}\{x(t + \tau)x(t)\}$  are periodic with some period, say  $T_0$ :

$$\mathbb{E}\{x(t + T_0)\} = \mathbb{E}\{x(t)\}, R_{xx}(t + T_0, \tau) = R_{xx}(t, \tau), \quad (2.10)$$

for all  $t$  and  $\tau$  [70]. We consider a cyclostationary digital signal  $x(t)$  and an LTV fading channel (i.e. due to the Doppler effect), having an impulse response of  $h(\tau', t)$ . According to the definition of cyclostationarity, we know that the autocorrelation function of  $x(t)$  is a periodic function of  $t$ , such that  $R_{xx}(t + T_0, \tau) = R_{xx}(t, \tau)$ , for some period  $T_0$ . The received signal  $y(t)$  through the LTV fading channel can be expressed as:

$$y(t) = \int_0^\infty x(t - \tau')h(\tau', t)d\tau' + w(t), \quad (2.11)$$

where  $w(t)$  is an additive wide-sense stationary (WSS) noise process. The autocorrelation function of the received signal  $y(t)$  can then be expressed as:

$$\begin{aligned} R_{yy}(t, \tau) &= \mathbb{E}\{y(t + \tau)y(t)\} \\ &= \mathbb{E}\left\{\left[\int_0^\infty x(t + \tau - \tau'_1)h(\tau'_1, t + \tau)d\tau'_1 + w(t + \tau)\right] \times \right. \\ &\quad \left. \times \left[\int_0^\infty x(t + \tau - \tau'_2)h(\tau'_2, t)d\tau'_2 + w(t)\right]\right\} \\ &= \mathbb{E}\left\{\int_0^\infty \int_0^\infty x(t + \tau - \tau'_1)x(t + \tau - \tau'_2) \times \right. \\ &\quad \left. \times h(\tau'_1, t + \tau)h(\tau'_2, t)d\tau'_1d\tau'_2\right\} + \mathbb{E}\{w(t + \tau)w(t)\} \\ &= \int_0^\infty \int_0^\infty \mathbb{E}\{x(t + \tau - \tau'_1)x(t + \tau - \tau'_2)\} \mathbb{E}\{h(\tau'_1, t + \tau)h(\tau'_2, t)\} d\tau'_1d\tau'_2 + R_{ww}(t, \tau) \\ &= \int_0^\infty \int_0^\infty R_{xx}(t, \tau - \tau'_1 + \tau'_2)R_{hh}(\tau'_1, \tau'_2; t + \tau, t)d\tau'_1d\tau'_2 + R_{ww}(\tau), \end{aligned}$$

where  $R_{hh}(\tau'_1, \tau'_2; t_1, t_2) \triangleq \mathbb{E}\{h(\tau'_1, t_1)h(\tau'_2, t_2)\}$  is the autocorrelation of the channel impulse response  $h(\tau', t)$ , and  $R_{ww}(t, \tau) = R_{ww}(\tau)$  is the autocorrelation function of the WSS noise.

Chapter 2. Wideband Spectrum Sensing

According to empirical studies, the channel can be considered as WSS as long as the mobile unit covers a distance in the dimension of a few tens of the wavelength of the carrier signal in an observation period [122]. We also assume that scattering components with different propagation delays are statistically uncorrelated. These channel models are called US (uncorrelated scattering) channel models or US models [114]. The most important class of stochastic LTV channel models is represented by models belonging both to the class of WSS and to the class of US. These channel models are called WSS uncorrelated scattering (WSSUS) models which are almost exclusively employed in current literature for modeling frequency selective mobile radio channels [114, 122–125].

Under this common assumption of WSSUS, the autocorrelation function of the impulse response of the LTV fading channel can be expressed as [114]:

$$R_{hh}(\tau'_1, \tau'_2; t + \tau, t) = \delta(\tau'_2 - \tau'_1) S_{hh}(\tau'_1, \tau), \quad (2.12)$$

where  $S_{hh}(\tau'_1, \tau)$  is called the *delay cross-power spectral density* [114]. We substitute (2.12) back into (2.12) to obtain:

$$\begin{aligned} R_{yy}(t, \tau) &= \int_0^\infty \int_0^\infty R_{xx}(t, \tau - \tau'_1 + \tau'_2) \times \delta(\tau'_2 - \tau'_1) S_{hh}(\tau'_1, \tau) d\tau'_1 d\tau'_2 + R_{ww}(\tau) \\ &= \int_0^\infty R_{xx}(t, \tau) S_{hh}(\tau'_1, \tau) d\tau'_1 + R_{ww}(\tau) \\ &= R_{xx}(t, \tau) \int_0^\infty S_{hh}(\tau'_1, \tau) d\tau'_1 + R_{ww}(\tau) \end{aligned} \quad (2.13)$$

so that

$$\begin{aligned} R_{yy}(t + T_0, \tau) &= R_{xx}(t + T_0, \tau) \int_0^\infty S_{hh}(\tau'_1, \tau) d\tau'_1 + R_{ww}(\tau) \\ &= R_{xx}(t, \tau) \int_0^\infty S_{hh}(\tau'_1, \tau) d\tau'_1 + R_{ww}(\tau) \\ &= R_{yy}(t, \tau). \end{aligned} \quad (2.14)$$

This shows that the autocorrelation function of the received signal  $y(t)$  is also periodic with the same period  $T_0$  as the transmitted signal  $x(t)$ . As a result, the received signal  $y(t)$  is also cyclostationary with the same cyclic components as  $x(t)$ .

A more general class of stochastic processes is obtained if the autocorrelation function  $R_{xx}(t, \tau)$  is almost periodic in  $t$  for each  $\tau$  [121]: A continuous-time real-valued stochastic process  $x(t)$  is said to be *almost-cyclostationary (ACS) in the wide sense* if its autocorrelation function  $R_{xx}(t, \tau)$  is an almost periodic function of  $t$  (with frequencies not depending on  $\tau$ ) [70]. When the input signal  $x(t)$  is considered as ACS, the output signal  $y(t)$  through the LTV fading channel is also ACS with the same cyclic components as  $x(t)$ , since we can see from (2.13) and (2.14) the autocorrelation function  $R_{yy}(t, \tau)$  is also almost periodic with the same period as  $R_{xx}(t, \tau)$ .

As a result, we see that when fading channels are considered as general LTV systems, the cyclostationary properties of the transmitted signals are not altered at the output of the channel, or the received signal at the Radiobot. This justifies the robustness of the proposed cyclostationarity based detection method in the presence of channel fading. Note that, the proposed cyclostationarity based detection method introduced in Section 2.2.2 also applies to the ACS assumption, since the SCF is also defined under the assumption of ACS and it has been shown that an ACS signal exhibits cyclostationarity at cycle frequency  $\alpha$  if  $R_{xx}^\alpha(\tau) \neq 0$ , similarly to the cyclostationary stochastic processes [70, 121].

## 2.4 Impact of the Doppler Shift on the Detected Carrier Frequencies

The cyclic autocorrelation function  $R_{yy}^\alpha(\tau)$  of the received signal  $y(t)$  is defined as  $R_{yy}^\alpha(\tau) \triangleq \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} R_{yy}(t, \tau) e^{-j2\pi\alpha t} dt$  [121]. Replacing  $R_{yy}(t, \tau)$  by its value in (2.13), we obtain:

$$R_{yy}^\alpha(\tau) = H(\tau)R_{xx}^\alpha(\tau) + R_{ww}(\tau)\delta^K(\alpha), \quad (2.15)$$

where  $H(\tau) = \int_{-\infty}^{\infty} S_{hh}(\tau'_1, \tau) d\tau'_1$  and  $\delta^K$  denotes the Kronecker delta function. We may compute the PSD  $S_y^0(f)$  of the received signal  $y(t)$  as the Fourier transform (denoted by the operator  $\mathcal{F}$ ) of  $R_y^\alpha(\tau)$  at  $\alpha = 0$ , such that:

$$\begin{aligned} S_y^0(f) &= \mathcal{F} \left\{ \int_{-\infty}^{\infty} S_{hh}(\tau'_1, \tau) d\tau'_1 \right\} * S_x^0(f) + S_w(f) \\ &= \int_{-\infty}^{\infty} \mathcal{F} \{ S_{hh}(\tau'_1, \tau) \} d\tau'_1 * S_x^0(f) + S_w(f) \\ &= \int_{-\infty}^{\infty} S(\tau'_1, f) d\tau'_1 * S_x^0(f) + S_w(f) \end{aligned} \quad (2.16)$$

$$= S_{\mu\mu}(f) * S_x^0(f) + S_w(f), \quad (2.17)$$

where  $S(\tau'_1, f)$  and  $S_{\mu\mu}(f)$  are, respectively, the *scattering function* and the *Doppler power spectral density*, and  $S_x^0(f)$  is the PSD of the transmitted signal. Note that (2.16) and (2.17) are obtained using (7.37) and (7.42) in [114], respectively.

The Doppler PSD is usually defined over a range  $[-f_{max}, f_{max}]$ , where  $f_{max}$  is the maximum Doppler frequency shift [114]. Thus, the received PSD can be expressed as:

$$S_y^0(f) = \int_{-f_{max}}^{f_{max}} S_{\mu\mu}(\nu) S_x^0(f - \nu) d\nu + S_{ww}(f). \quad (2.18)$$

Based on (2.18), the convolution of  $S_x^0(f)$  with a window of length  $2f_{max}$  causes the PSD to spread at most by  $\pm f_{max}$  at each point. If the Doppler PSD  $S_{\mu\mu}(f)$

is symmetric (such as Jakes' type [114]), the carrier frequency components of the detected feature points do not shift since the main lobes of the PSD are spread evenly in both left and right directions. However, if  $S_{\mu\mu}(f)$  is not symmetric (such as Rice's, Gauss I or Gauss II types [114]), the detected carrier frequencies will shift by an amount smaller than  $f_{max}$ . Therefore, due to the Doppler shift, it may not be possible to detect and distinguish signals that are separated by less than  $f_{max}$  in the spectrum. However, based on the users activity and by using appropriate learning algorithms, the Radiobot might be able to detect each of the signals when they are the only transmitted signals. Then using this knowledge, it may be able to distinguish them when both signals are transmitted simultaneously. This again emphasizes the importance of true learning from past experience during the signal detection and classification steps.

## 2.5 Self-Reconfiguration of the Spectrum Sensing Module

The performance of the Radiobot is related to the quality and accuracy of the sensing observations. It is required to optimize the sensing module so that it best estimates the RF activity in the surrounding environment. Several parameters may need to be optimized during the sensing process, such as the sensing duration, detector thresholds, spectrum sensing policies, etc. based on the particular RF environment it encounters at a given time. It is the task of the learning and reasoning abilities of the Radiobot to make the cognitive engine dynamically adapt these parameters based on its past experience. To be specific, assume that the Radiobot needs to optimize its cyclic sub-profile threshold  $\zeta$  such that it achieves a certain false alarm probability. Of course, it is almost impossible to obtain analytical solutions to this problem due to the complexity of the cyclic profile equation and to the uncertainty

in the surrounding environment. A possible solution is to learn the optimal threshold value iteratively based on the sensing observations, as in [110].

An online learning algorithm was proposed in [110] to adapt the threshold value of NP test when the probability distribution of the detected signals is unknown. The threshold is thus dynamically updated to achieve a desired false alarm probability. The learning process is conducted during a training period in which the observed data are drawn from a null hypothesis. In our case, however, we do not assume a training period and we propose a learning algorithm that updates the cyclic sub-profile threshold  $\zeta$  during the normal operation time itself to achieve a desired false alarm probability  $\phi$ . By the help of the energy detection, the learning algorithm identifies the absence of transmitted signals to perform the learning process. The objective of the learning algorithm is to minimize the Kullback-Leibler distance  $K(P||Q)$  between two probability distributions  $P$  and  $Q$ , similar to [110], where:

$$K(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \quad (2.19)$$

We denote by  $P$  and  $Q$  the desired and actual probability distributions of the cyclostationary detector output, conditioned on the absence of transmitted signals. These probability distributions correspond to Bernoulli random variables, representing whether a signal is (1) or is not (0) detected. By defining  $\phi$  and  $P_f(\zeta)$  as the desired and actual false alarm probabilities (for a given threshold  $\zeta$ ), respectively, the Kullback-Leibler distance can then be expressed as:

$$K(P||Q) = K(\phi, P_f(\zeta)) = \phi \log \frac{\phi}{P_f(\zeta)} + (1 - \phi) \log \frac{1 - \phi}{1 - P_f(\zeta)}. \quad (2.20)$$

Note that  $K(\phi, P_f(\zeta)) = 0$  iff  $\phi = P_f(\zeta)$ . Due to its convexity in  $P_f(\zeta)$ , the Kullback-Leibler distance guarantees a global minimum. Moreover, it was shown in [110] that  $K(\phi, P_f(\zeta))$  is convex in  $\zeta$  iff  $P_f(\zeta)$  is monotonous, which is satisfied in our case. However, since the analytical expression of  $P_f(\zeta)$  is unknown, it can be estimated as the ratio of sample points that exceed the threshold  $\zeta$  in the cyclic profile  $I(\alpha)$ , when

there is no transmitted signals. As noted in [110], to achieve accurate estimate for  $P_f(\zeta)$ , the recursive adaptation in  $\zeta$  should not be too frequent. This is taken into account in the proposed learning algorithm (Algorithm 2), in which the threshold  $\zeta$  is updated after each  $N_c > 1$  updates of the false alarm probability  $P_f(\zeta)$ .

---

**Algorithm 2** Learning algorithm to control the cyclic sub-profile threshold  $\zeta$

---

Initialize:  $counter = 1$ .

**while** No signal is detected by the energy detector **do**

    Update the false alarm probability  $P_f(\zeta)$  and  $counter = counter + 1$ .

**if**  $counter = N_c$  **then**

        Update  $\zeta$  such that:  $\zeta \leftarrow \zeta + \psi (P_f(\zeta) - \phi)$ .

        Reset  $counter = 1$ .

**end if**

**end while**

---

The update rule in Algorithm 2 minimizes the Kullback-Leibler function since it follows a gradient descent direction that reduces the difference  $|P_f(\zeta) - \phi|$  at a learning rate of  $\psi > 0$ . Moreover, due to the convexity of the Kullback-Leibler function, this algorithm is guaranteed to converge to a unique optimal threshold value.

## 2.6 Simulation Results

In order to demonstrate the performance of our proposed cyclostationarity-based autonomous signal detection procedure, we simulate several signals in the  $2.4GHz$  ISM band. These signals are assumed to have carriers at  $2.412GHz$ ,  $2.437GHz$  and  $2.462GHz$  and symbol rates of 10, 12 and 14 Mbauds, respectively. The signals are allowed to use different quadrature amplitude modulation (QAM) schemes and are equally likely to be in ON or OFF states during each sensing period. Wireless

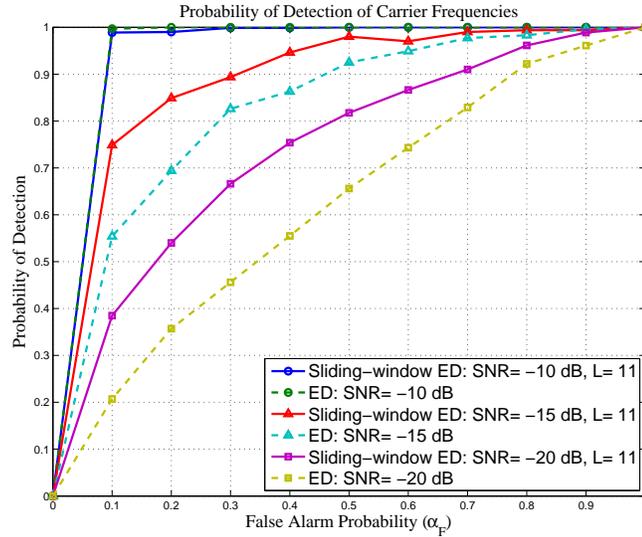


Figure 2.5: Comparison between the ROC's of the sliding-window and conventional energy detections. The sliding-window length is  $L = 11$ .

channel is assumed to be Rayleigh fading. The fading channels coefficients  $h$  are normalized, such that  $\mathbb{E}\{h^2\} = 1$ . Also, the Radiobot's receiver is subjected to white Gaussian noise.

We assume that the sensed signal is downconverted to IF band with an IF oscillator with frequency of  $f_I = 2.35GHz$ . After IF conversion, the three signals are supposed to be centered at 62, 87 and 112 MHz. Each sensing observation takes  $12\mu s$  with a receiver SNR of  $20dB$ .

In Fig. 2.5, we show the ROC curves of the adopted sliding-window energy detection scheme [33, 41]. This detector is compared to the conventional energy detection and it shows superior detection performance. Next, we show in Fig. 2.6 the detection performance of the cyclostationary detection for different values of SNR's and for different sensing times. The results show that 95% of detection probability can be achieved at an SNR of  $-6dB$  and with a sensing time of  $T = 30\mu s$ . Afterwards,

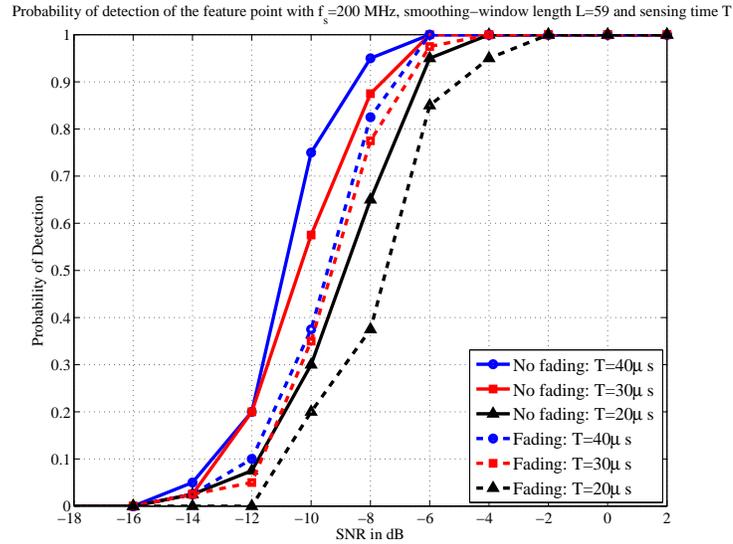


Figure 2.6: Probability of identification of feature points with a sampling frequency  $f_s = 200\text{MHz}$  and sliding-window length  $L = 59$ . The detected signal is a 4-QAM with symbol rate of 5 Mbauds and down-converted to a carrier frequency of  $20\text{MHz}$ . The performance is compared under both non-fading and Rayleigh fading channels.

we verify, in Fig. 2.7, the convergence of the learning algorithm proposed in Section 2.5. We let  $\phi$  to be the desired false alarm probability of the cyclostationary detection and let  $\zeta$  be the control threshold. Starting from  $\zeta = 0$ , Algorithm 2 converges to constant threshold at which the actual false alarm probability  $P_f(\zeta)$  converges to  $\phi$ . The learning rate is set to  $\psi = 0.2$  and the threshold  $\zeta$  is updated after each  $N_c = 20$  updates of the false alarm probability  $P_f(\zeta)$ . Note that a similar learning procedure could be applied to adapt the energy detector threshold  $\eta_{PSD}$ . However, this step is not required in our case since we have an analytical expression for  $\eta_{PSD}$  in (A.6).

Finally, in order to verify the multi-band operability of the Radiobot, we simulate, in Fig. 2.8, the sequential sensing in two different sub-bands. Each sub-band has 2 different systems and we assume that these users can be either ON (1) or OFF (0) at each time instant, as shown in the user activity curves of Fig. 2.8. The Radiobot

Chapter 2. Wideband Spectrum Sensing

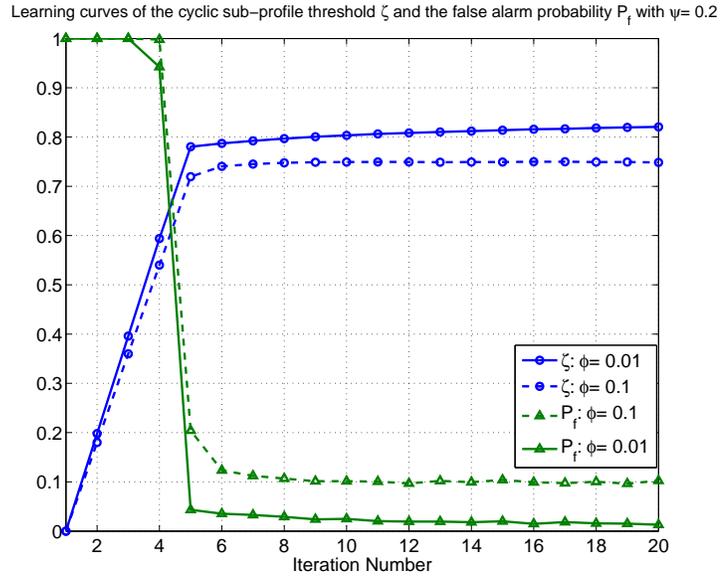


Figure 2.7: Learning curves of the cyclic sub-profile threshold  $\zeta$  and the false alarm probability  $P_f$  with a learning rate  $\psi = 0.2$  and a desired false alarm probability  $\phi$  (with  $L = 11$ ).

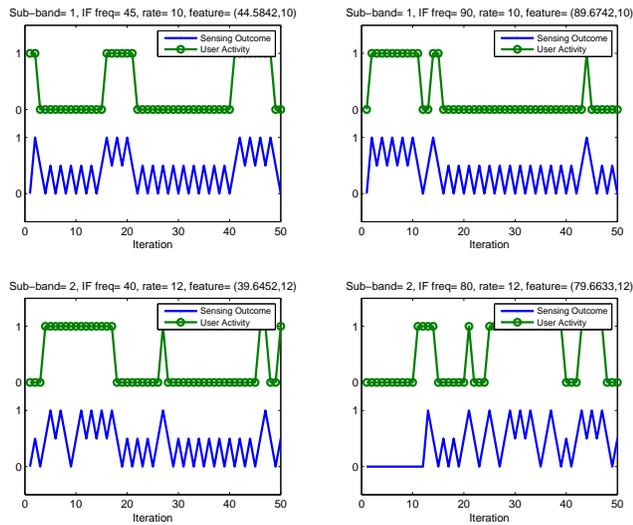


Figure 2.8: Detection of multiple users in 2 separate sub-bands.

senses sequentially these sub-bands. We plot the sensing outcomes and represent by 1 (resp. 0) whether the corresponding system is detected (resp. not detected). An outcome of 0.5 implies that the corresponding sub-band is not sensed at a certain time. The results in Fig. 2.8 show that the Radiobot can accurately detect the different systems and allocate them to appropriate clusters, while switching between different sub-bands.

## **2.7 Conclusion**

In this chapter, we have presented an autonomous CR architecture, referred to as the Radiobot [1]. This model is aimed at emphasizing the cognitive aspects of CR's by requiring that the Radiobot is able to achieve self-learning and self-reconfigurability. The proposed Radiobot architecture employs a joint energy/cyclostationary detection to extract different features from the sensed signals. A learning algorithm is proposed to allow self-reconfigurability of the Radiobot sensing module to match its RF environment. We analyzed the performance of the energy detection through the ROC and showed the robustness of the cyclostationary detection to fading and to WSS noise. We verified, through simulations, the expected convergence of the proposed learning algorithm and the multi-band operability of the Radiobot architecture with the proposed wideband spectrum sensing approach.

# Chapter 3

## Machine Learning in CR's

### 3.1 Introduction

The term *cognitive radio* has been used to refer to radio devices that are capable of learning and adapting to their environment [2, 3]. Cognition, from the Latin word *cognoscere* (to know), is defined as a process involved in gaining knowledge and comprehension, including thinking, knowing, remembering, judging and problem solving [126]. A key aspect of any CR is the ability for self-programming or autonomous learning [1, 127]. In [4], Haykin envisioned CR's to be *brain-empowered* wireless devices that are specifically aimed at improving the utilization of the electromagnetic spectrum. According to Haykin, a CR is assumed to use the methodology of *understanding-by-building* and is aimed to achieve two primary objectives: Permanent reliable communications and efficient utilization of the spectrum resources [4].

In order to be really *cognitive*, a CR should be equipped with the abilities of learning and reasoning [1–3, 60, 93, 128]. These capabilities are to be embedded in a cognitive engine which has been identified as the core of a CR [44–49], following the pioneering vision of [3]. The cognitive engine is to coordinate the actions of the

### Chapter 3. Machine Learning in CR's

CR by making use of machine learning algorithms. However, only in recent years there has been a growing interest in applying machine learning algorithms to CR's, as shown in [40, 41, 91, 92, 96].

In general, learning is necessary if *the precise effects of the inputs on the outputs of a given system are not known* [91]. In other words, if the input-output function of a given system is unknown, learning techniques are required to estimate that function in order to design proper inputs. For example, in wireless communications, the wireless channels are non-ideal and may cause uncertainty. If it is desired to reduce the probability of error over a wireless link by reducing the coding rate, learning techniques can be applied to estimate the wireless channel characteristics and to determine the specific coding rate that is required to achieve a certain probability of error [91]. In this case, the probability of error, as a function of a specific coding rate, is considered to be unknown and to be estimated using learning tools. The problem of channel estimation is relatively simple and can be solved via estimation algorithms [129]. However, in the case of CR's and cognitive CRN's, problems become more complicated with the increase in the degrees of freedom of wireless systems especially with the introduction of highly-reconfigurable SDR's. In this case, several parameters and policies need to be adjusted simultaneously (e.g. transmit power, coding scheme, modulation scheme, sensing algorithm, communication protocol, sensing policy, etc.) and no simple formula may be able to determine these setup parameters simultaneously. This is due to the complex interactions among these factors and their impact on the RF environment. Thus, learning methods can be applied to allow efficient adaption of the CR's to their environment, yet without the complete knowledge of the dependence among these parameters [95].

The problem becomes even more complicated with heterogeneous CRN's. In this case, a CR not only has to adapt to the RF environment, but also it has to coordinate its actions with respect to the other radios in the network. With only a limited

### *Chapter 3. Machine Learning in CR's*

amount of information exchange among nodes, a CR needs to estimate the behavior of other nodes in order to select its proper actions. For example, in the context of DSA, CR's try to access idle primary channels while limiting collisions with both licensed and other secondary cognitive users [96]. In addition, if the CR's are operating in unknown RF environments [1], conventional solutions to the decision process (e.g. Dynamic programming in the case of Markov decision processes (MDP's) [102]) may not be feasible since they require complete knowledge of the system. On the other hand, by applying special learning algorithms such as the RL [95,96,99], it is possible to arrive at the optimal solution to the MDP, without knowing the transition probabilities of the Markov model. Therefore, given the reconfigurability requirements and the need for autonomous operation in unknown and heterogeneous RF environment, CR's may use learning algorithms as a tool for adaptation to the environment and to coordinate with peer radio devices. Moreover, incorporation of low-complexity learning algorithms can lead to reduced system complexities in CR's [90].

A look at the recent literature on CR's reveals that both supervised and unsupervised learning techniques have been proposed for various learning tasks. The authors in [46,78,130] have considered supervised learning based on neural networks and SVM's for CR applications. On the other hand, unsupervised learning, such as RL, has been considered in [131,132] for dynamic spectrum sharing (DSS) applications. The distributed Q-learning algorithm has been shown to be effective in a particular CR application in [99]. For example, in [133], CR's used the Q-learning to improve detection and classification performance of primary signals. Other applications of RL to CR's can be found, for example, in [97,98,134,135]. Recent work in [136] introduces novel approaches to improve the efficiency of RL by adopting a weight-driven exploration. Unsupervised Bayesian non-parametric learning based on the Dirichlet process was proposed in [137] and was used for signal classification in [41,42]. A robust signal classification algorithm was also proposed in [79], based on unsupervised learning.

### Chapter 3. Machine Learning in CR's

Although the RL algorithms (such as Q-learning) may provide a suitable framework for autonomous unsupervised learning, their performance in partially observable, non-Markovian and multi-agent systems can be unsatisfactory [100, 138–140]. Other types of learning mechanisms such as evolutionary learning [138, 141], learning by imitation, learning by instruction [142] and policy-gradient methods [139, 140] have been shown to outperform RL on certain problems under such conditions. For example, the policy-gradient approach has been shown to be more efficient in partially observable environments since it searches directly for optimal policies in the policy space [139, 140].

Similarly, learning in multi-agent environments has been considered in recent years, especially when designing learning policies for CRN's. For example, [143] compared a cognitive network to a human society that exhibits both individual and group behaviors, and a strategic learning framework for cognitive networks was proposed in [144]. An evolutionary game framework was proposed in [145] to achieve adaptive learning in cognitive users during their strategic interactions. By taking into consideration the distributed nature of CRN's and the interactions among the CR's, optimal learning methods can be obtained based on cooperative schemes, which helps to avoid the selfish behaviors of individual nodes in a CRN.

One of the main challenges of learning in distributed CRN's is the problem of action coordination [100]. To ensure optimal behavior, centralized policies may be applied to generate optimal joint actions for the whole network. However, centralized schemes are not always feasible in distributed networks. Hence, the aim of cognitive nodes in distributed networks is to apply decentralized policies that ensure near-optimal behavior while reducing the communication overhead among nodes. For example, a decentralized technique that was proposed in [126, 146] was based on the concept of *docitive networks*, from the Latin word *docere* (to teach), which establishes knowledge transfer (i.e. teaching) over the wireless medium [126]. The objective

### Chapter 3. Machine Learning in CR's

of docitive networks is to reduce the cognitive complexity, speed up the learning rate and generate better and more reliable decisions [126]. In a docitive network, radios teach each others by interchanging knowledge such that each node attempts to learn from a *more intelligent* node. The radios are not only supposed to teach end-results, but rather elements of the methods of getting there [126]. For example, in a docitive network, newly joint radios can acquire certain policies from existing radios in the network. Of course, there will be communication overhead during the knowledge transfer process. However, as it is demonstrated in [126, 146], this overhead is compensated by the policy improvement achieved due to cooperative docitive behavior.

In this chapter, we discuss the role of learning in CR's and emphasize how crucial the autonomous learning ability in realizing a real CR device. We present a survey of the state-of-the-art achievements in applying machine learning techniques to CR's.

In this chapter, specifically, we focus on the challenges that are encountered in applying machine learning techniques to CR's, given the importance of learning in CR applications. In particular, we provide in-depth discussions on the different types of learning paradigms in the two main categories: supervised learning and unsupervised learning. The machine learning techniques discussed in this chapter include those that have been already proposed in the literature as well as those that might be reasonably applied to CR's in future. The advantages and limitations of these techniques are discussed to identify perhaps the most suitable learning methods in a particular context or in learning a particular task or an attribute. Moreover, we provide discussions on the centralized and decentralized learning techniques as well as the challenging machine learning problems in the non-Markovian environments.

The remainder of this chapter is organized as follows: Section 3.2 defines the learning problem in CR's and presents the different learning paradigms. Sections 3.3 and 3.4 present the decision-making and feature classification problems, respectively.

In Section 3.5, we describe the learning problem in centralized and decentralized CRN's and we conclude this chapter in Section 3.6.

## 3.2 Need of learning in CR's

### 3.2.1 Definition of the Learning Problem

A CR is defined to be “an intelligent wireless communication system that is aware of its environment and uses the methodology of understanding-by-building to learn from the environment and adapt to statistical variations in the input stimuli” [4]. As a result, a CR is expected to be intelligent by nature. It is capable of learning from its experience by interacting with its RF environment [1]. According to [147], learning should be an indispensable component of any intelligent system.

As identified in [147], there are three main conditions for intelligence: 1) Perception, 2) learning and 3) reasoning, as illustrated in Fig. 3.1. Perception is the ability of sensing the surrounding environment and the internal states to acquire information. Learning is the ability of transforming the acquired information into knowledge by using methodologies of classification and generalization of hypotheses. Finally, knowledge is used to achieve certain goals through reasoning. As a result, learning is at the core of any intelligent device including, in particular, CR's. It is the fundamental tool that allows a CR to acquire knowledge from its observed data.

In the followings, we discuss how the above three constituents of intelligence are built into CR's. First, *perception* can be achieved through the sensing measurements of the spectrum. This allows the CR to identify on-going RF activities in its surrounding environment, as presented in Chapter 2. After acquiring the sensing observations, the CR tries to *learn* from them in order to classify and organize the observations into suitable categories (knowledge). Finally, the *reasoning* ability

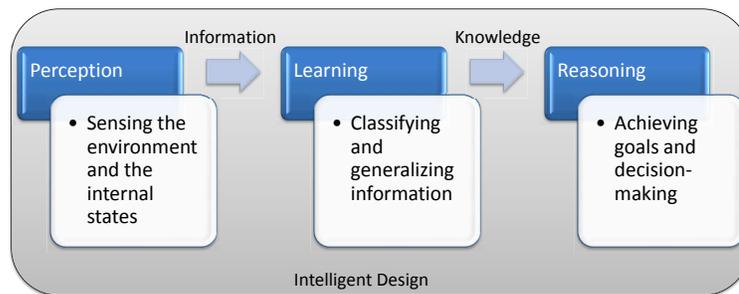


Figure 3.1: An intelligent design can transform the acquired information into knowledge by learning.

allows the CR to use the knowledge acquired through learning to achieve its objectives. These characteristics were initially specified by Mitola in defining the so-called *cognition cycle* [2]. We illustrate in Fig. 3.2 an example of a simplified cognition cycle that was proposed in [1] for autonomous CR's, referred to as *Radiobots* [60]. Figure 3.2 shows that Radiobots can learn from their previous actions by observing their impact on the outcomes. The learning outcomes are then used to update, for example, the sensing (i.e. observation) and channel access (i.e. decision) policies in DSA applications [4, 9, 22, 96].

### 3.2.2 Unique Characteristics of CR Learning Problems

Although the term *cognitive radio* has been interpreted differently in various research communities [1], perhaps the most widely accepted definition is as a radio that can sense and adapt to its environment [1, 3, 4, 91]. The term *cognitive* implies *awareness, perception, reasoning* and *judgement*. As we already pointed out earlier, in order for a CR to derive reasoning and judgement from perception, it must possess the ability for learning [147]. Learning implies that the current actions should be based on *past and current observations* of the environment [148]. Thus, history plays a major role in the learning process of CR's.

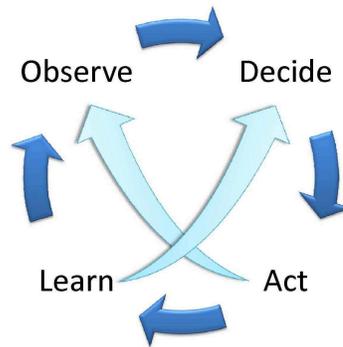


Figure 3.2: The cognition cycle of an autonomous CR (referred to as the Radiobot) [1]. Decisions that drive Actions are made based on the Observations and Learnt knowledge. The impact of actions on the system performance and environment leads to new Learning. The Radiobot's new Observations are guided by this Learnt Knowledge of the effects of past Actions.

Several learning problems are specific to CR applications due to the nature of the CR's and their operating RF environments. First, due to noisy observations and sensing errors, CR's can only obtain partial observations of their state variables. The learning problem is thus equivalent to a learning process in a partially observable environment and must be addressed accordingly.

Second, CR's in CRN's try to learn and optimize their behaviors simultaneously.

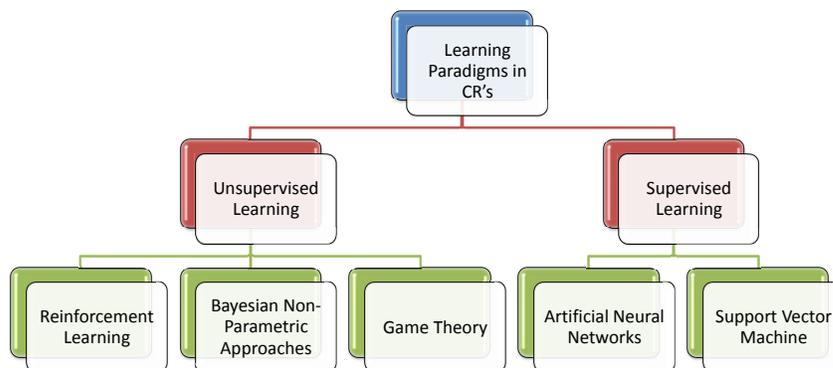


Figure 3.3: Supervised and unsupervised learning approaches for CR's.

### *Chapter 3. Machine Learning in CR's*

Hence, the problem is naturally a multi-agent learning process. Furthermore, the desired learning policy may be based on either cooperative or non-cooperative schemes and each CR might have either full or partial knowledge of the actions of the other cognitive users in the network. In the case of partial observability, a CR might apply special learning algorithms to estimate the actions of the other nodes in the network before selecting its appropriate actions, as in, for example, [100].

Finally, autonomous learning methods are desired in order to enable CR's to learn on its own in an unknown RF environment. In contrast to licensed wireless users, a truly CR may be expected to operate in any available spectrum band, at any time and in any location [1]. Thus, a CR may not have any prior knowledge of the operating RF environment such as the noise or interference levels, noise distribution or user traffics. Instead, it should possess autonomous learning algorithms that may reveal the underlying nature of the environment and its components. This makes the unsupervised learning a perfect candidate for such learning problems in CR applications.

To sum up, the three main characteristics that need to be considered when designing efficient learning algorithms for CR's are:

1. Learning in partially observable environments.
2. Multi-agent learning in distributed CRN's.
3. Autonomous learning in unknown RF environments.

A CR design that embeds the above capabilities will be able to operate efficiently and optimally in any RF environment.

### 3.2.3 Types of Learning Paradigms: Supervised versus Un-supervised Learning

Learning can be either supervised or unsupervised, as depicted in Fig. 3.3. Un-supervised learning may particularly be suitable for CR's operating in alien RF environments [1]. In this case, autonomous unsupervised learning algorithms permit exploring the environment characteristics and self-adapting actions accordingly without having any prior knowledge [1,40]. However, if the CR has prior information about the environment, it might exploit this knowledge by using supervised learning techniques. For example, if certain signal waveform characteristics are known to the CR prior to its operation, training algorithms may help CR's to better detect signals with those characteristics.

In [142], the two categories of supervised and unsupervised learning are identified as learning by *instruction* and learning by *reinforcement*, respectively. A third learning regime is defined as the learning by *imitation* in which an agent learns by observing the actions of similar agents [142]. In [142], it was shown that the performance of a learning agent (learner) is influenced by its learning regime and its operating environment. Thus, to learn efficiently, a CR must adopt the best learning regime for a given learning problem, whether it is learning by *imitation*, by *reinforcement* or by *instruction* [142]. Of course, some learning regimes may not be applicable under certain circumstances. For example, in the absence of an instructor, the CR may not be able to learn by instruction and may have to resort to learning by reinforcement or imitation. An effective CR architecture is the one that can switch among different learning regimes depending on its requirements, the available information and the environment characteristics.

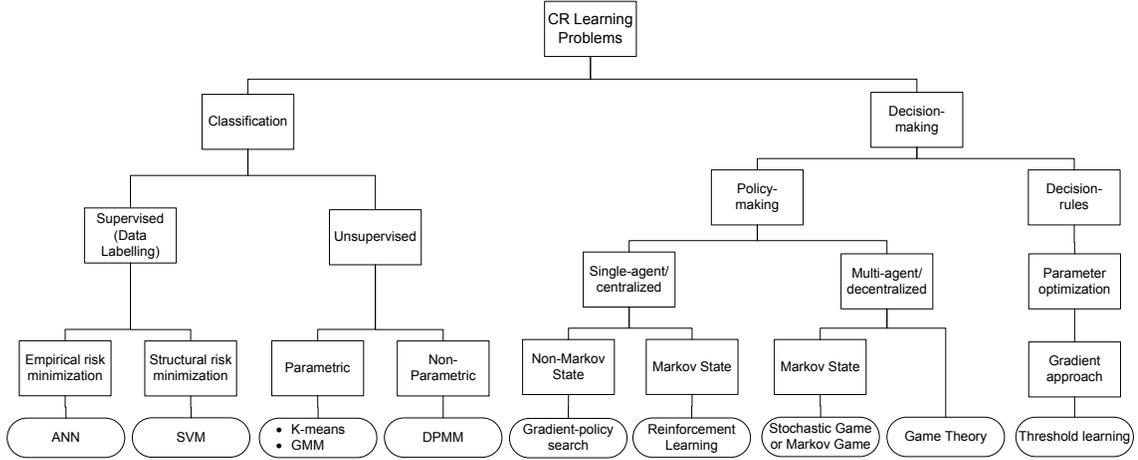


Figure 3.4: Typical problems in CR and their corresponding learning algorithms.

### 3.2.4 Learning Problems in CR

Several learning algorithms can be used by CR's to achieve different goals. In order to obtain a better insight on the functions and similarities among the presented algorithms, we identify two main problem categories and show the learning algorithms under each category. The hierarchical organization of the learning algorithms and their dependence is illustrated in Fig. 3.4.

Referring to Fig. 3.4, we identify two main CR problems (or tasks) as:

1. Decision-making.
2. Feature classification.

These problems are general in a sense that they cover a wide range of CR tasks. For example, classification problems arise in spectrum sensing while decision-making problems arise in determining the spectrum sensing policy, power control or adaptive modulation. Learning algorithms can be classified under the above two tasks, and can be applied under specific conditions, as illustrated in Fig. 3.4. For example,

the classification algorithms can be split into two different categories: Supervised and unsupervised. Supervised algorithms require training with labeled data and include, among others, the artificial neural network (ANN) and SVM algorithms. The ANN algorithm is based on empirical risk minimization and does not require prior knowledge of the observed process distribution, as opposed to structural models [149, 150]. However, SVM algorithms, which are based on structural risk minimization, have shown superior performance, in particular for small training examples, since they avoid the problem of *overfitting* [149, 151].

For instance, consider a set of training data denoted as  $\{(x_1, y_1), \dots, (x_N, y_N)\}$  such that  $x_i \in X, y_i \in Y, \forall i \in \{1, \dots, N\}$ . The objective of a supervised learning algorithm is to find a function  $g : X \rightarrow Y$  that maximizes a certain score function [149]. In ANN,  $g$  is defined as the function that minimizes the empirical risk:

$$R(g) = R_{emp}(g) = \frac{1}{N} \sum_{i=1}^N L(y_i, g(x_i)) , \quad (3.1)$$

where  $L : Y \times Y \rightarrow \mathbb{R}^+$  is a loss function. Hence, ANN algorithms find the function  $g$  that best fits the data. However, if the function space  $G$  includes too many candidates or the training set is not sufficiently large (i.e. small  $N$ ), empirical risk minimization may lead to high variance and poor generalization, which is known as *overfitting*. In order to prevent overfitting, structural risk minimization can be used, which incorporates a regularization penalty to the optimization process [149]. This can be done by minimizing the following risk function:

$$R(g) = R_{emp}(g) + \lambda C(g) , \quad (3.2)$$

where  $\lambda$  controls the bias/variance tradeoff and  $C$  is a penalty function [149].

In contrast with the supervised approaches, unsupervised classification algorithms do not require labeled training data and can be classified as being either parametric or non-parametric. Unsupervised parametric classifiers include the K-means and GMM

### *Chapter 3. Machine Learning in CR's*

algorithms and require prior knowledge of the number of classes (or clusters). On the other hand, non-parametric unsupervised classifiers do not require prior knowledge of the number of clusters and can estimate this quantity from the observed data itself, for example using methods based on the DPMM [41–43].

Decision-making is another major task that has been widely investigated in CR applications [22, 25, 96, 99, 152–159]. Decision-making problems can in turn be split to policy-making and decision rules. Policy-making problems can be classified as either centralized or decentralized. In a policy-making problem, an agent determines its optimal set of actions over a certain time duration, thus defining an optimal policy (or an optimal strategy in game theory terminology). In a centralized scenario with a Markov state, RL algorithms can be used to obtain optimal solution to the corresponding MDP, without prior knowledge of the transition probabilities [95, 102]. In non-Markov environments, optimal policies can be obtained based on gradient policy search algorithms which search directly for solutions in the policy space. On the other hand, for multi-agent scenarios, game theory is proposed as a solution that can capture the distributed nature of the environment and the interactions among users. With a Markov state assumption, the system can be modeled as a Markov game (or a stochastic game), while conventional game models can be used, otherwise. Note that learning algorithms can be applied to the game-theoretic models (such as the no-regret learning [160–162]) to arrive at equilibrium under uncertainty conditions.

Finally, decision rules form another class of decision-making problems which can be formulated as hypothesis testing problems for certain observation models. In the presence of uncertainty about the observation model, learning tools can be applied to implement a certain decision rule. For example, the threshold-learning algorithm proposed in [41, 110] was used to optimize the threshold of the NP test under uncertainty about the noise distribution.

## Chapter 3. Machine Learning in CR's

In brief, we have identified two main classes of problems and have determined the conditions under which certain algorithms can be applied for these problems. For example, the DPMM algorithm can be applied for classification problems if the number of clusters is unknown, whereas the SVM may be better suited if labeled data is available for training.

The learning algorithms that are presented in this chapter help to optimize the behavior of the learning agent (in particular the CR) under uncertainty conditions. For example, the RL leads to the optimal policy for MDP's [95] while game theory leads to Nash equilibrium, whenever it exists, of certain types of games [163]. The SVM algorithm optimizes the structural risk by finding a global minimum, whereas the ANN only leads to local minimum of the empirical risk [150,151]. The DPMM is useful for non-parametric classification and converges to the stationary probability distribution of the Markov chain in the MCMC Gibbs sampling procedure [43,164]. As a result, the proposed learning algorithms achieve certain optimality criterion within their application contexts.

### 3.3 Decision-making in CR's

#### 3.3.1 Centralized Policy-making under Markov States: Reinforcement Learning

Reinforcement learning is a technique that permits an agent to modify its behavior by interacting with its environment [95]. This type of learning can be used by agents to learn autonomously without supervision. In this case, the only source of knowledge is the feedback an agent receives from its environment after executing an action. Two main features characterize the RL: *trial-and-error* and *delayed reward*. By *trial-and-error* it is assumed that an agent does not have any prior knowledge about the

### Chapter 3. Machine Learning in CR's

environment, and executes actions blindly in order to *explore* the environment. The *delayed reward* is the feedback signal that an agent receives from the environment after executing each action. These rewards can be positive or negative quantities, telling *how good or bad* an action is. The agent's objective is to maximize these rewards by *exploiting* the system.

Reinforcement learning is distinguished from *supervised learning* by not having a supervisor to tell whether an action is correct or wrong. Therefore, the learning agent only relies on its interactions with the environment and tries to learn on its own. This makes the RL a basic algorithm for autonomous learning.

A key concept in RL is that the agent should observe the reward for each action in each situation. By repetition, the agent attempts to learn to favor the actions that lead to positive rewards, and avoids the actions that lead to negative rewards. Moreover, a learning agent can use the RL to choose the actions that permit avoiding certain bad situations. After several repetitions, the agent acquires an optimal policy and adapts its actions and behavior to the environment.

The theory of RL has evolved along three main threads [95]. The first thread is the learning by *trial-and-error* which has its roots in the psychology of animals. This approach goes back to 1898 and has led to the revival of the RL in the early 1980's [165]. For example, in his analysis of animal behavior, Thorndike observed that animals tend to reselect actions that are followed by good outcomes, and they try to avoid the actions that lead to bad outcomes [166].

The second thread originates from the problem of optimal control and its dynamic programming-based solution. One approach to this problem was developed in the mid 1950's by Bellman and others by extending the theory of Hamilton and Jacobi. The dynamic programming is found to be the most efficient solution to the optimal control problem. However it suffers from what Bellman called "the curse of dimensionality"

because the complexity of dynamic programming increases exponentially with the number of state variables [95]. Also, it requires complete knowledge of the system.

The third thread that led to the RL is the *temporal difference* concept which was first applied to learning problems by Samuel [167]. This idea consists of updating an evaluation function about the environment in order to improve the total reward. The three threads that constitute the RL were joined together in 1989 by Watkins when he developed the Q-learning algorithm [107, 168].

An RL-based cognition cycle for CR's was defined in [132], as illustrated in Fig. 3.5. It shows the interactions between the CR and its RF environment. The learning agent receives an observation  $o_t$  of the state  $s_t$  at time instant  $t$ . The observation is accompanied by a delayed reward  $r_t(s_{t-1}, a_{t-1})$  representing the reward received at time  $t$  resulting from taking action  $a_{t-1}$  in state  $s_{t-1}$  at time  $t-1$ . The learning agent uses the observation  $o_t$  and the delayed reward  $r_t(s_{t-1}, a_{t-1})$  to compute the action  $a_t$  that should be taken at time  $t$ . The action  $a_t$  results in a state transition from  $s_t$  to  $s_{t+1}$  and a delayed reward  $r_{t+1}(s_t, a_t)$ . It should be noted that here the learning agent is not passive and does not only observe the outcomes from the environment, but also affects the state of the system via its actions such that it might be able to drive the environment to a desired state that brings the highest reward to the agent.

### An MDP Framework for RL

Reinforcement learning algorithms are applied under the assumption that the agent-environment interaction forms an MDP. An MDP is characterized by the following elements [102]:

- A set of *decision epochs*  $T$  including the point of times at which decisions are made. The time interval between decision epoch  $t \in T$  and decision epoch  $t + 1 \in T$  is denoted as *period*  $t$ .

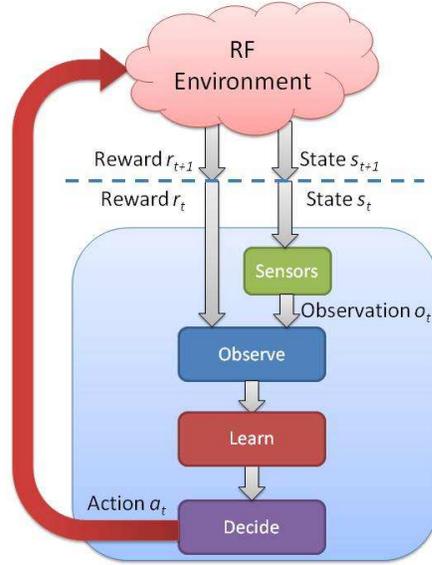


Figure 3.5: The RL cycle: At the beginning of each learning cycle, the agent receives a full or partial observation of the current state, as well as the accrued reward. By using the state observation and the reward value, the agent updates its policy (e.g. updating the Q-values) during the learning stage. Finally, during the decision stage, the agent selects a certain action according to the updated policy.

- A finite set  $\mathcal{S}$  of states for the agent (i.e. secondary user).
- A finite set  $\mathcal{A}$  of actions that are available to the agent. In particular, in each state  $s \in \mathcal{S}$ , a subset  $\mathcal{A}_s \subseteq \mathcal{A}$  might be available.
- A non-negative function  $p_t(s'|s, a)$  denoting the probability that the system is in state  $s'$  at time epoch  $t + 1$ , when the decision-maker chooses action  $a \in \mathcal{A}$  in state  $s \in \mathcal{S}$  at time  $t$ . Note that, the subscript  $t$  might be dropped from  $p_t(s'|s, a)$  if the system is stationary.
- A real-valued function  $r_t^{MDP}(s, a)$  defined for state  $s \in \mathcal{S}$  and action  $a \in \mathcal{A}$  to denote the value at time  $t$  of the reward received in period  $t$  [102]. Note that, in RL literature, the reward function is usually defined as the delayed reward  $r_{t+1}(s, a)$  that is obtained at time epoch  $t + 1$  after taking action  $a$  in state  $s$

at time  $t$  [95].

At each time epoch  $t$ , the agent observes the current state  $s$  and chooses an action  $a$ . An optimum policy maximizes the total expected rewards, which is usually discounted by a discount factor  $\gamma \in [0, 1)$  in case of an infinite time horizon. Thus, the objective is to find the optimal policy  $\pi$  that maximizes the expected *discounted return* [95]:

$$R(t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}(s_{t+k}, a_{t+k}) , \quad (3.3)$$

where  $s_t$  and  $a_t$  are, respectively, the state and action at time  $t \in \mathbb{Z}$ .

The optimal solution of an MDP can be obtained by using several methods such as the *value iteration* algorithm based on dynamic programming [102]<sup>1</sup>. Given a certain policy  $\pi$ , the value of state  $s \in \mathcal{S}$  is defined as the expected discounted return if the system starts in state  $s$  and follows policy  $\pi$  thereafter [95, 102]. This value function can be expressed as [95]:

$$V^\pi(s) = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}(s_{t+k}, a_{t+k}) \mid s_t = s \right\} , \quad (3.4)$$

where  $\mathbb{E}_\pi\{.\}$  denotes the expected value given that the agent follows policy  $\pi$ . Similarly, the value of taking action  $a$  in state  $s$  under a policy  $\pi$  is defined as the *action-value function* [95]:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}(s_{t+k}, a_{t+k}) \mid s_t = s, a_t = a \right\} . \quad (3.5)$$

The value iteration algorithm finds an  $\varepsilon$ -optimal policy assuming stationary rewards and transition probabilities (i.e.  $r_t(s, a) = r(s, a)$  and  $p_t(s'|s, a) = p(s'|s, a)$ ). The algorithm initializes a  $v^0(s)$  for each  $s \in \mathcal{S}$  arbitrarily and iteratively updates

---

<sup>1</sup>There are other algorithms that can be applied to find the optimal policy of an MDP such as *policy iteration* and *linear programming* methods. Interested readers are referred to [102] for additional information regarding these methods.

Chapter 3. Machine Learning in CR's

$v^n(s)$  (where  $v^n(s)$  is the estimated value of state  $s$  after the  $n$ -th iteration) for each  $s \in \mathcal{S}$  as follows [102]:

$$v^{n+1}(s) = \max_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \sum_{j \in \mathcal{S}} p(j|s, a) v^n(j) \right\} . \quad (3.6)$$

The algorithm stops when  $\|v^{n+1} - v^n\| < \varepsilon \frac{1-\gamma}{2\gamma}$  and the  $\varepsilon$ -optimal decision  $d_\varepsilon(s)$  of each state  $s \in \mathcal{S}$  is defined as:

$$d_\varepsilon(s) = \arg \max_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \sum_{j \in \mathcal{S}} p(j|s, a) v^{n+1}(j) \right\} . \quad (3.7)$$

Obviously, the *value iteration* algorithm requires explicit knowledge of the transition probability  $p(s'|s, a)$ . On the other hand, an RL algorithm, referred to as the Q-learning, was proposed by Watkins in 1989 [168] to solve the MDP problem without knowledge of the transition probabilities and has been recently applied to CR's [96, 99, 133, 169]. The Q-learning algorithm is one of the important *temporal difference* (TD) methods [95, 168]. It has been shown to converge to the optimal policy when applied to single agent MDP models (i.e. centralized control) in [168] and [95]. However, it can also generate satisfactory near-optimal solutions even for DEC-POMDP's, as shown in [99]. The *one-step* Q-learning is defined as follows:

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \left[ r_{t+1}(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a) \right] . \quad (3.8)$$

The learned action-value function,  $Q$  in (3.8), directly approximates the optimal action-value function  $Q^*$  [95]. However, it is required that all state-action pairs need to be continuously updated in order to guarantee *correct convergence* to  $Q^*$ . This can be achieved by applying an  $\varepsilon$ -greedy policy that ensures that all state-action pairs are updated with a non-zero probability, thus leading to an optimal policy [95]. If the system is in state  $s \in \mathcal{S}$ , the  $\varepsilon$ -greedy policy selects action  $a^*(s)$  such that:

$$a^*(s) \left\{ \begin{array}{ll} = \arg \max_{a \in \mathcal{A}} Q(s, a) & , \text{ with Pr} = 1 - \varepsilon \\ \sim U(\mathcal{A}) & , \text{ with Pr} = \varepsilon \end{array} \right. , \quad (3.9)$$

where  $U(\mathcal{A})$  is the discrete uniform probability distribution over the set of actions  $\mathcal{A}$ .

### 3.3.2 Centralized Policy-making with Non-Markovian States: Gradient-policy Search

While RL and *value-iteration* methods [95, 102] can lead to optimal policies for the MDP problem, their performance in non-Markovian environments remains questionable [139, 140]. Hence, the authors in [138–140] proposed the *policy-search* approach as an alternative solution method for non-Markovian learning tasks. Policy-search algorithms directly look for optimal policies in the policy space itself, without having to estimate the actual states of the systems [139, 140]. In particular, by adopting policy gradient algorithms, the policy vector can be updated to reach an optimal solution (or a local optimum) in non-Markovian environments.

The value-iteration approach has several other limitations as well: First, it is restricted to deterministic policies. Second, any small changes in the estimated value of an action can cause that action to be, or not to be selected [139]. This would affect the optimality of the resulting policy since optimal actions might be eliminated due to an underestimation of their value functions.

On the other hand, the gradient-policy approach has shown promising results, for example, in robotics applications [170, 171]. Compared to value-iteration methods, the gradient-policy approach requires fewer parameters in the learning process and can be applied in model-free setups not requiring perfect knowledge of the controlled system.

The policy-search approach can be illustrated by the following overview of policy-gradient algorithms from [140]. We consider a class of stochastic policies that are

Chapter 3. Machine Learning in CR's

parameterized by  $\theta \in \mathbb{R}^K$ . By computing the gradient with respect to  $\theta$  of the average reward, the policy could be improved by adjusting the parameters in the gradient direction. To be concrete, assume  $r(X)$  to be a reward function that depends on a random variable  $X$ . Let  $q(\theta, x)$  be the probability of the event  $\{X = x\}$ . The gradient with respect to  $\theta$  of the expected performance  $\eta(\theta) = \mathbb{E}\{r(X)\}$  can be expressed as:

$$\nabla\eta(\theta) = \mathbb{E} \left\{ r(X) \frac{\nabla q(\theta, x)}{q(\theta, x)} \right\} . \quad (3.10)$$

An unbiased estimate of the gradient can be obtained via simulation by generating  $N$  independent identically distributed (i.i.d.) random variables  $X_1, \dots, X_N$  that are distributed according to  $q(\theta, x)$ . The unbiased estimate of  $\nabla\eta(\theta)$  is thus expressed as:

$$\hat{\nabla}\eta(\theta) = \frac{1}{N} \sum_{i=1}^N r(X_i) \frac{\nabla q(\theta, X_i)}{q(\theta, X_i)} . \quad (3.11)$$

By the law of large numbers,  $\hat{\nabla}\eta(\theta) \rightarrow \nabla\eta(\theta)$  with probability one. Note that the quantity  $\frac{\nabla q(\theta, X_i)}{q(\theta, X_i)}$  is referred to as the *likelihood ratio* or the *score function* [140]. By having an estimate of the reward gradient, the policy parameter  $\theta \in \mathbb{R}^K$  can be updated by following the gradient direction, such that:

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k \nabla\eta(\theta) , \quad (3.12)$$

for some step size  $\alpha_k > 0$ .

Authors in [170, 171] identify two major steps when performing policy gradient methods:

1. A policy evaluation step in which an estimate of the gradient  $\nabla\eta(\theta)$  of the expected return  $\eta(\theta)$  is obtained, given a certain policy  $\pi_\theta$ .

### Chapter 3. Machine Learning in CR's

2. A policy improvement step which updates the policy parameter  $\theta$  through steepest gradient ascent  $\theta_{k+1} = \theta_k + \alpha_k \nabla \eta(\theta)$ .

Note that, estimating the gradient  $\nabla \eta(\theta)$  is not straight-forward, especially in the absence of simulators that generate the  $X_i$ 's. To resolve this problem, special algorithms can be designed to obtain reasonable approximations of the gradient. Indeed, several approaches have been proposed to estimate the gradient policy vector, mainly in robotics applications [170, 171]. Three different approaches have been considered in [171] for policy gradient estimation:

1. Finite difference (FD) methods.
2. Vanilla policy gradient (VPG) methods.
3. Natural policy gradient (NG) methods.

Finite difference methods, originally used in stochastic simulations literature, are among the oldest policy gradient approaches. The idea is based on changing the current policy parameter  $\theta_k$  by small perturbations  $\delta\theta_i$  and computing  $\delta\eta_i = \eta(\theta_k + \delta\theta_i) - \eta(\theta_k)$ . The policy gradient  $\nabla \eta(\theta)$  can be thus estimated as:

$$\mathbf{g}_{FD} = (\mathbf{\Delta}\mathbf{\Theta}^T \mathbf{\Delta}\mathbf{\Theta})^{-1} \mathbf{\Delta}\mathbf{\Theta} \mathbf{\Delta}\mathbf{\eta}, \quad (3.13)$$

where  $\mathbf{\Delta}\mathbf{\Theta} = [\delta\theta_1, \dots, \delta\theta_I]^T$ ,  $\mathbf{\Delta}\mathbf{\eta} = [\delta\eta_1, \dots, \delta\eta_I]^T$  and  $I$  is the number of samples [170, 171]. Advantages of this approach is that it is straightforward to implement and does not introduce significant noise to the system during exploration. However, the gradient estimate can be very sensitive to perturbations (i.e.  $\delta\theta_i$ ) which may lead to bad results [171].

Instead of perturbing the parameter  $\theta_k$  of a deterministic policy  $u = \pi(x)$  (with  $u$  being the action and  $x$  being the state), the VPG approach assumes a stochastic policy  $u \sim \pi(u|x)$  and obtains an unbiased gradient estimate [171]. However,

Chapter 3. Machine Learning in CR's

in using the VPG method, the variance of the gradient estimate depends on the squared average magnitude of the reward, which can be very large. In addition, the convergence of the VPG to the optimal solution can be very slow, even with an optimal baseline [171]. The NG approach which leads to fast policy gradient algorithms can alleviate this problem. Natural gradient approaches use the Fisher information  $F(\theta)$  to characterize the information about the policy parameters  $\theta$  that is contained in the observed path  $\tau$  [171]. A path (or a trajectory)  $\tau = [x_{0:H}, u_{0:H}]$  is defined as the sequence of states and actions, where  $H$  denotes the horizon which can be infinite [170]. Thus, the Fisher information  $F(\theta)$  can be expressed as:

$$F(\theta) = \mathbb{E} \left\{ \nabla_{\theta} \log p(\tau|\theta) \nabla_{\theta} \log p(\tau|\theta)^T \right\} , \quad (3.14)$$

where  $p(\tau|\theta)$  is the probability of trajectory  $\tau$ , given certain policy parameter  $\theta$ . For a given policy change  $\delta\theta$ , there is an information loss of  $l_{\theta}(\delta\theta) \approx \delta\theta^T F(\theta) \delta\theta$ , which can also be seen as the change in path distribution  $p(\tau|\theta)$ . By searching for the policy change  $\delta\theta$  that maximizes the expected return  $\eta(\theta + \delta\theta)$  for a constant information loss  $l_{\theta}(\delta\theta) \approx \varepsilon$ , the algorithms searches for the highest return value on an ellipse around the current parameter  $\theta$  and then goes in the direction of the highest values. More formally, the direction of the steepest ascent on the ellipse around  $\theta$  can be expressed as [171]:

$$\delta\theta = \arg \max_{\delta\theta \text{ s.t. } l_{\theta}(\delta\theta)=\varepsilon} \delta\theta^T \nabla_{\theta} \eta(\theta) = F^{-1}(\theta) \nabla_{\theta} \eta(\theta) . \quad (3.15)$$

This algorithm is further explained in [171] and can be easily implemented based on the Natural Actor-Critic algorithms [171].

By comparing the above three approaches, the authors in [171] showed that NG and VPG methods are considerably faster and result in better performance, compared to FD. However, FD has the advantage of being simpler and applicable in more general situations.

### 3.3.3 Decentralized Policy-making: Game Theory

Game theory [172] presents a suitable platform for modeling rational behavior among CR's in CRN's. There is a rich literature on game theoretic techniques in CR, as can be found in [173–183]. Game theory [172] is a mathematical tool that attempts to model the behavior of rational entities in an environment of conflict. This branch of mathematics has primarily been popular in economics, and has later found its way into biology, political science, engineering and philosophy [163]. In wireless communications, game theory has been applied to data communication networking, in particular, to model and analyze routing and resource allocation in competitive environments [184].

A game model consists of several rational entities that are denoted as the players. Assuming a game model  $\mathbf{G} = (\mathcal{N}, (\mathcal{A}_i)_{i \in \mathcal{N}}, (U_i)_{i \in \mathcal{N}})$ , where  $\mathcal{N} = \{1, \dots, N\}$  denotes the set of  $N$  players and each player  $i \in \mathcal{N}$  has a set  $\mathcal{A}_i$  of available actions and a utility function  $U_i$ . Let  $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$  be the set of strategy profiles of all players. In general, the utility function of an individual player  $i \in \mathcal{N}$  depends on the actions taken by all the players involved in the game and is denoted as  $U_i(a_i, a_{-i})$ , where  $a_i \in \mathcal{A}_i$  is an action (or strategy) of player  $i$  and  $a_{-i} \in \mathcal{A}_{-i}$  is a strategy profile of all players except player  $i$ . Each player selects its strategy in order to maximize its utility function. A Nash equilibrium of a game is defined as a point at which the utility function of each player does not increase if the player deviates from that point, given that all the other players' actions are fixed. Formally, a strategy profile  $(a_1^*, \dots, a_N^*) \in \mathcal{A}$  is a Nash equilibrium if [161]:

$$U_i(a_i^*, a_{-i}) \geq U_i(a'_i, a_{-i}), \forall i \in \mathcal{N}, \forall a'_i \in \mathcal{A}_i. \quad (3.16)$$

A key advantage of applying game theoretic solutions to CR protocols is in reducing the complexity of adaptation algorithms in large cognitive networks. While optimal centralized control can be computationally prohibitive in most CRN's, due

to communication overhead and algorithm complexity, game theory presents a distributed platform to handle such situations [94]. Another justification for applying game theoretic approaches to CR's is the assumed cognition in the CR behavior, which induces *rationality* among CR's, similar to the players in a game.

### Game Theoretic Approaches in Wireless Communications

There are two major game theoretic approaches that can be used to model the behavior of nodes in a wireless medium: Cooperative and non-cooperative games. In a non-cooperative game, the players make rational decisions considering only their individual payoff. In a cooperative game, however, players are grouped together and establish an enforceable agreement in their group [163].

A non-cooperative game can be classified as either a complete or an incomplete information game. In a complete information game, each player can observe the information of other players such as their payoffs and their strategies. On the other hand, in an incomplete information game, this information is not available to other players. A game with incomplete information can be modeled as a Bayesian game in which the game outcomes can be estimated based on Bayesian analysis. A Bayesian Nash equilibrium is defined for the Bayesian game, similar to the Nash equilibrium in the complete information game [163].

In addition, a game can also be classified as either static or dynamic. In a static game, each player takes its actions without knowledge of the strategies taken by the other players. This is denoted as a one-shot game which ends when actions of all players are taken and payoffs are received. In a dynamic game, however, a player selects an action in the current stage based on the knowledge of the actions taken by the other players in the current or previous stages. A dynamic game is also called a sequential game since it consists of a sequence of repeated static games. The common equilibrium solution in dynamic games is the subgame perfect Nash

equilibrium which represents a Nash equilibrium of every subgame in the original game [163].

### **Applications of Game Theory to CR's**

Several types of games have been adapted to model different situations in CRN's [94]. For example, supermodular games (the games having the following important and useful property: there exists at least one pure strategy Nash equilibrium) have been used for distributed power control in [185, 186] and for rate adaptation in [187]. Repeated games were applied for DSA by multiple secondary users that share the same spectrum hole in [188]. In this context, repeated games are useful in building reputations and applying punishments in order to reinforce a certain desired outcome. The Stackelberg game model can be used as a model for implementing CR behavior in cooperative spectrum leasing where the primary users act as the game-leaders and secondary cognitive users as the followers [189].

Auctions are one of the most popular methods used for selling a variety of items, ranging from antiques to wireless spectrum. In auction games the players are the buyers who must select the appropriate bidding strategy in order to maximize their perceived utility (i.e., the value of the acquired items minus the payment to the seller). The concept of auction games has successfully been applied to cooperative dynamic spectrum leasing (DSL) in [27, 190], as well as to spectrum allocation problems in [191]. The basics of the auction games and the open challenges of applying auction games to the field of spectrum management are discussed in [192].

Stochastic games (or Markov games) can be used to model the greedy selfish behavior of CR's in a CRN, where CR's try to learn their best response and improve their strategies over time [193]. In the context of CR's, stochastic games are dynamic, competitive games with probabilistic actions played by secondary spectrum users.

The game is played in a sequence of stages. At the beginning of each stage, the game is in a certain state. The secondary users choose their actions, and each secondary user receives a reward that depends on both its current state and its selected actions. The game then moves to the next stage having a new state with a certain probability, which depends on the previous state as well as the actions selected by the secondary users. The process continues for a finite or infinite number of stages. The stochastic games are generalizations of repeated games that have only a single state.

### **Learning in Game Theoretic Models**

There are several learning algorithms that have been proposed to estimate unknown parameters in a game model (e.g. other players' strategies, environment states, etc.). In particular, no-regret learning allows initially uninformed players to acquire knowledge about their environment state in a repeated game [160]. This algorithm does not require prior knowledge of the number of players nor the strategies of other players. Instead, each player will learn a better strategy based on the rewards obtained from playing each of its strategies [160].

The concept of regret is related to the benefit a player feels after taking a particular action, compared to other possible actions. This can be computed as the average reward the player gets from a particular action, averaged over all other possible actions that could be taken instead of that particular action. Actions resulting in lower regret are updated with higher weights and are thus selected more frequently [160]. In general, no-regret learning algorithms help players to choose their policies when they do not know the other players' actions. Furthermore, no-regret learning can adapt to a dynamic environment with little system overhead [160].

No-regret learning was applied in [160] to allow a CR to update both its transmission power and frequencies simultaneously. In [162], it was used to detect ma-

### Chapter 3. Machine Learning in CR's

licious nodes in spectrum sensing whereas in [161] no-regret learning was used to achieve a correlated equilibrium in OSA for CR's. Assuming the game model  $\mathbb{G} = (\mathcal{N}, (\mathcal{A}_i)_{i \in \mathcal{N}}, (U_i)_{i \in \mathcal{N}})$  defined above, in a correlated equilibrium, a strategy profile  $(a_1, \dots, a_N) \in \mathcal{A}$  is chosen randomly according to a certain probability distribution  $p$  [161]. A probability distribution  $p$  is a correlated strategy, if and only if, for all  $i \in \mathcal{N}$ ,  $a_i \in \mathcal{A}_i$ ,  $a_{-i} \in \mathcal{A}_{-i}$  [161]:

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} p(a_i, a_{-i}) [U_i(a'_i, a_{-i}) - U_i(a_i, a_{-i})] \leq 0, \forall a'_i \in \mathcal{A}_i. \quad (3.17)$$

Note that, every Nash equilibrium is a correlated equilibrium and Nash equilibria correspond to the special case where  $p(a_i, a_{-i})$  is a product of each individual player's probability for different actions, i.e. the play of the different players is independent [161]. Compared to the non-cooperative Nash equilibrium, the correlated equilibrium in [161] was shown to achieve better performance and fairness.

Recently, [194] proposed a game-theoretic stochastic learning solution for OSA when the channel availability statistics and the number of secondary users are unknown a priori. This model attempts to resolve non-feasible OSA solution which requires prior knowledge of the environment and the actions taken by the other users. By applying the stochastic learning solution in [194], the communication overhead among the CR users is reduced. Furthermore, the model in [194] provides an alternative solution to OSA schemes proposed in [152, 154] that do not consider the interactions among multiple secondary users in a POMDP framework [194].

Thus, learning in a game theoretic framework can help CR's to adapt to environment variations given a certain uncertainty about the other users' strategies. Therefore, it provides a potential solution for multi-agent learning problems under partial observability assumptions.

### 3.3.4 Decision Rules under Uncertainty: Threshold-learning

A CR may be implemented on a mobile device that changes location over time and switches transmissions among several channels. This mobility and multi-band/multi-channels operability may pose a major challenge for CR's in adapting to their RF environments. A CR may encounter different noise or interference levels when switching among different bands or when moving from one place to another. Hence, the operating parameters (e.g. test thresholds and sampling rate) of CR's need to be adapted with respect to each particular situation. Moreover, CR's may be operating in unknown RF environments and may not have perfect knowledge of the characteristics of the other existing primary or secondary signals, requiring special learning algorithms to allow the CR to explore and adapt to its surrounding environment. In this context, special types of learning can be applied to directly learn the optimal values of certain design and operation parameters.

*Threshold-learning* presents a technique that permits such dynamic adaptation of operating parameters to satisfy the performance requirements, while continuously learning from the past experience. By assessing the effect of previous parameter values on the system performance, the learning algorithm optimizes the parameters values to ensure a desired performance. For example, in considering energy detection, after measuring the energy levels at each frequency, a CR decides on the occupancy of a certain frequency band by comparing the measured energy levels to a certain threshold. The threshold levels are usually designed based on NP tests in order to maximize the detection probability of primary signals, while satisfying a constraint on the false alarm. However, in such tests, the optimal threshold depends on the noise level. An erroneous estimation of the noise level might cause sub-optimal behavior and violation of the operation constraints (for example, exceeding a tolerable collision probability with primary users). In this case, and in the absence of perfect knowledge about the noise levels, threshold-learning algorithms can be devised to learn the

optimal threshold values. Given each choice of a threshold, the resulting false alarm rate determines how the test threshold should be regulated to achieve a desired false alarm probability. An example of application of threshold learning can be found in [195] where a threshold learning algorithm was derived for optimizing spectrum sensing in CR's. The resulting algorithm was shown to converge to the optimal threshold that satisfies a given false alarm probability.

## **3.4 Feature Classification in CR's**

### **3.4.1 Non-parametric Unsupervised Classification:**

#### **The DPMM**

A major challenge an autonomous CR can face is the lack of knowledge about the surrounding RF environment, in particular, when operating in the presence of unknown primary signals [1]. Even in such situations, a CR is expected to be able to adapt to its environment while satisfying certain requirements. For example, in DSA, a CR must not exceed a certain collision probability with primary users. For this reason, a CR should be equipped with the ability to autonomously explore its surrounding environment and to make decisions about the primary activity based on the observed data. In particular, a CR must be able to extract knowledge concerning the statistics of the primary signals based on measurements [1, 41]. This makes unsupervised learning an appealing approach for CR's in this context. In the following, we may explore a Dirichlet process prior based [196, 197] technique as a framework for such non-parametric learning and point out its potentials and limitations. The Dirichlet process prior based techniques are considered as unsupervised learning methods since they make few assumptions about the distribution from which the data is drawn [43], as can be seen in the following discussion.

### Chapter 3. Machine Learning in CR's

A Dirichlet process  $DP(\alpha_0, G_0)$  is defined to be the distribution of a random probability measure  $G$  over a measurable space  $(\Theta, \mathcal{B})$ , such that, for any finite measurable partition  $(A_1, \dots, A_r)$  of  $\Theta$ , the random vector  $(G(A_1), \dots, G(A_r))$  is distributed as a finite dimensional Dirichlet distribution with parameters  $(\alpha_0 G_0(A_1), \dots, \alpha_0 G_0(A_r))$  such that:

$$(G(A_1), \dots, G(A_r)) \sim Dir(\alpha_0 G_0(A_1), \dots, \alpha_0 G_0(A_r)) , \quad (3.18)$$

where  $\alpha_0 > 0$ . A vector  $(X_1, \dots, X_n) \sim Dir(a_1, \dots, a_n)$  is said to be distributed according to a Dirichlet distribution with parameters  $(a_1, \dots, a_n)$  if:

$$f(x_1, \dots, x_n | a_1, \dots, a_n) = \frac{\Gamma(\sum_{i=1}^n a_i)}{\prod_{i=1}^n \Gamma(a_i)} \prod_{i=1}^n x_i^{a_i-1} , \quad (3.19)$$

subject to  $\sum_{i=1}^n x_i = 1$ , with  $x_i > 0$ ,  $a_i > 0$ , for all  $i = 1, \dots, n$ .

We denote  $G \sim DP(\alpha_0, G_0)$  to represent the probability measure  $G$  that is drawn from the Dirichlet process  $DP(\alpha_0, G_0)$ . In other words,  $G$  is a *random probability measure* whose distribution is given by the Dirichlet process  $DP(\alpha_0, G_0)$  [43]. That is, the realizations  $G$  of a Dirichlet process are *random probability distributions*, in contrast with *random variables* or *random processes* that are usually assumed in probabilistic models.

### Construction of the Dirichlet Process

Teh [43] describes several ways of constructing the Dirichlet process. A first method is a direct approach that constructs the random probability distribution  $G$  based on the *stick-breaking* method. The *stick-breaking* construction of  $G$  can be summarized as follows [43]:

1. Generate independent i.i.d. sequences  $\{\pi'_k\}_{k=1}^\infty$  and  $\{\phi_k\}_{k=1}^\infty$  such that

$$\begin{cases} \pi'_k | \alpha_0, G_0 \sim Beta(1, \alpha_0) \\ \phi_k | \alpha_0, G_0 \sim G_0 \end{cases} , \quad (3.20)$$

Chapter 3. Machine Learning in CR's

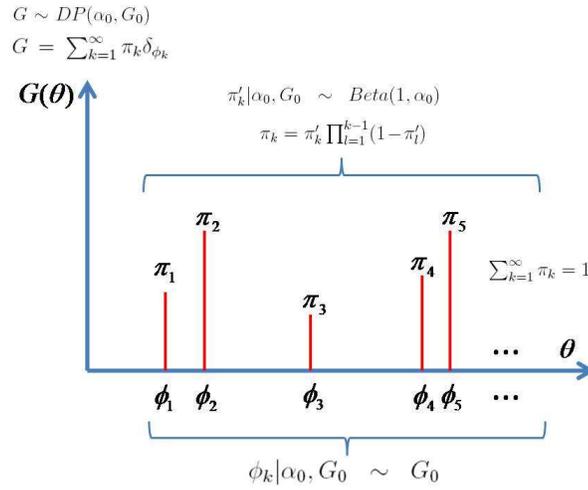


Figure 3.6: One realization of the Dirichlet process.

where  $\text{Beta}(a, b)$  is the beta distribution whose probability density function (pdf) is given by  $f(x, a, b) = \frac{x^{a-1}(1-x)^{b-1}}{\int_0^1 u^{a-1}(1-u)^{b-1} du}$ .

2. Define  $\pi_k = \pi'_k \prod_{l=1}^{k-1} (1 - \pi'_l)$ . We can write  $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots) \sim GEM(\alpha_0)$ , where  $GEM$  stands for Griffiths, Engen and McCloskey [43]. The  $GEM(\alpha)$  process generates the vector  $\boldsymbol{\pi}$  as described above, given a parameter  $\alpha_0$  in (3.20).
3. Define  $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$ , where  $\delta_{\phi}$  is a probability measure concentrated at  $\phi$  (and  $\sum_{k=1}^{\infty} \pi_k = 1$ ).

In the above construction  $G$  is a random probability measure distributed according to  $DP(\alpha_0, G_0)$ . The randomness in  $G$  stems from the random nature of both the weights  $\pi_k$  and the weights positions  $\phi_k$ . A sample distribution  $G$  of a Dirichlet process is illustrated in Fig. 3.6, using the steps described above in the *stick-breaking* method. Since  $G$  has an infinite discrete support (i.e.  $\{\phi_k\}_{k=1}^{\infty}$ ), this makes it a suitable candidate for non-parametric Bayesian classification problems in which the number of clusters is unknown *a priori* (i.e. allowing for infinite number of clusters),

with the infinite discrete support (i.e.  $\{\phi_k\}_{k=1}^{\infty}$  being the set of clusters. However, due to the infinite sum in  $G$ , it may not be practical to construct  $G$  directly by using this approach in many applications. An alternative approach to construct  $G$  is by using either the Polya urn model [197] or the Chinese restaurant process (CRP) [198]. The CRP is a discrete-time stochastic process. A typical example of this process can be described by a Chinese restaurant with infinitely many tables and each table (cluster) having infinite capacity. Each customer (feature point) that arrives at the restaurant (RF spectrum) will choose a table with a probability proportional to the number of customers on that table. It may also choose a new table with a certain fixed probability.

A second approach to constructing a Dirichlet process does not define  $G$  explicitly. Instead, it characterizes the distribution of the drawings  $\theta$  of  $G$ . Note that  $G$  is discrete with probability 1. For example, the Polya urn model [197] does not construct  $G$  directly, but it characterizes the draws from  $G$ . Let  $\theta_1, \theta_2, \dots$  be i.i.d. random variables distributed according to  $G$ . These random variables are independent, given  $G$ . However, if  $G$  is integrated out,  $\theta_1, \theta_2, \dots$  are no more conditionally independent and they can be characterized as:

$$\theta_i | \{\theta_j\}_{j=1}^{i-1}, \alpha_0, G_0 \sim \sum_{k=1}^K \frac{m_k}{i-1+\alpha_0} \delta_{\phi_k} + \frac{\alpha_0}{i-1+\alpha_0} G_0, \quad (3.21)$$

where  $\{\phi_k\}_{k=1}^K$  are the  $K$  distinct values of  $\theta_i$ 's and  $m_k$  is the number of values of  $\theta_i$  that are equal to  $\phi_k$ . Note that, this conditional distribution is not necessarily discrete since  $G_0$  might be a continuous distribution (in contrast with  $G$  which is discrete with probability 1). The  $\theta_i$ 's that are drawn from  $G$  exhibit a clustering behavior since a certain value of  $\theta_i$  is most likely to reoccur with a nonnegative probability (due to the point mass functions in the conditional distribution). Moreover, the number of distinct  $\theta_i$  values is infinite, in general, since there is a nonnegative probability that the new  $\theta_i$  value is distinct from the previous  $\theta_1, \dots, \theta_{i-1}$ . This conforms with the definition of  $G$  as a probability mass function (pmf) over an infinite

discrete set. Since  $\theta_i$ 's are distributed according to  $G$ , given  $G$ , we denote:

$$\theta_i|G \sim G . \tag{3.22}$$

### Dirichlet Process Mixture Model

The Dirichlet process makes a perfect candidate for non-parametric classification problems through the DPMM. The DPMM imposes a non-parametric prior on the parameters of the mixture model [43]. The DPMM can be defined as follows:

$$\left\{ \begin{array}{l} G \sim DP(\alpha_0, G_0) \\ \theta_i|G \sim G \\ \mathbf{y}_i|\theta_i \sim f_{\theta_i}(\mathbf{y}_i) \end{array} \right. , \tag{3.23}$$

where the likelihood function  $f_{\theta}(\mathbf{y}_i) \triangleq f(\mathbf{y}_i|\theta_i = \theta)$ , with  $\theta_i$  being the parameter of  $\mathbf{y}_i$ . In (3.23),  $G$  is drawn from a non-parametric set of distributions and is discrete with probability 1 [43]. Given a certain realization  $G$ , parameters  $\theta_i$ 's can be drawn from  $G$ , forming a set of mixture components for the DPMM. Feature vectors  $\mathbf{y}_i$  can thus be drawn from the distribution  $f_{\theta_i}(\mathbf{y}_i)$ .

### DPMM-based Classification using Gibbs Sampling

The problem of DPMM-based classification is to estimate the mixture component  $\theta_i$  for each feature vector  $\mathbf{y}_i$ , for all  $i \in \{1, \dots, N\}$ . In particular, we are interested in finding the maximum *a posteriori* probability (MAP) estimates of  $\theta_i$  ( $i = 1, \dots, N$ ), given the feature vectors  $\mathbf{y}_{1:N} \triangleq \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ . This can be obtained using MCMC methods, in particular, the Gibbs sampling to draw samples from the joint posterior distribution of  $(\theta_1, \dots, \theta_N)$  [42, 85]. The Gibbs sampling method samples each parameter  $\theta_i$ , given the other parameters  $\{\theta_j\}_{j \neq i}$ . Hence, it can be efficiently implemented with the DPMM framework in which a closed-form expression of the conditional distribution of  $\theta_i|\{\theta_j\}_{j \neq i}$  can be obtained.

$$\theta_i | \{\theta_j\}_{j \neq i}, \mathbf{y}_{1:N} \begin{cases} = \theta_j & \text{with prob. } q_j = \frac{f_{\theta_j}(\mathbf{y}_i)}{\alpha_0 f(\mathbf{y}_i) + \sum_{j=1, j \neq i}^N f_{\theta_j}(\mathbf{y}_i)} \\ \sim f(\theta_i | \mathbf{y}_i) & \text{with prob. } q_0 = \frac{\alpha_0 f(\mathbf{y}_i)}{\alpha_0 f(\mathbf{y}_i) + \sum_{j=1, j \neq i}^N f_{\theta_j}(\mathbf{y}_i)} \end{cases} . \quad (3.24)$$

However, it is hard to find analytical MAP estimates of  $\theta_i$ 's since the joint distribution of  $(\theta_1, \dots, \theta_N)$ , given  $\mathbf{y}_{1:N}$ , is unknown. As an alternative, we may use Monte Carlo methods to compute the MAP estimates by sampling from the posterior distribution of  $\theta_i$ 's, given  $\mathbf{y}_{1:N}$  [199, 200]. In particular, in situations that we have the conditional distribution of each  $\theta_i$ , given the other parameters  $\{\theta_j\}_{j \neq i}$ , we can construct an MCMC algorithm based on Gibbs sampling to draw samples from the joint posterior distribution of  $(\theta_1, \dots, \theta_N)$  [201].

The Gibbs sampling algorithm starts with arbitrary estimates of  $\theta_i$ 's and draws samples from the conditional distribution of each parameter  $\theta_i$ , given the other parameters  $\{\theta_j\}_{j \neq i}$ , where  $\{\theta_j\}_{j \neq i}$  take the values of their most recent estimates [201]. It can be shown that these samples converge in probability to the actual posterior distribution of  $(\theta_1, \dots, \theta_N)$ , giving an effective way to estimate  $\theta_i$ 's [85]. However, the Gibbs sampler usually samples the parameters  $\theta_i$ 's sequentially, which makes the process computationally prohibitive, especially for large  $N$ . As an alternative, we propose in Chapter 4 simplified and sequential Gibbs sampling algorithms to improve the convergence rate of the Gibbs sampling process, taking into consideration the clustering behavior of the DPMM classifier.

By assuming a DPMM framework, the posterior distribution of  $\theta_i | \{\theta_j\}_{j \neq i}, \mathbf{y}_{1:N}$  can be computed as in (3.24), where  $f(\mathbf{y}_i) = \int_{\theta} f_{\theta}(\mathbf{y}_i) G_0(\theta) d\theta$  is the marginal distribution of  $\mathbf{y}_i$ , assuming a prior  $G_0(\theta)$ , and  $f_{\theta}(\mathbf{y}_i) \triangleq f(\mathbf{y}_i | \theta_i = \theta)$ , for all  $\theta$ 's, where  $\theta_i$  stands for the parameter of the feature vector  $\mathbf{y}_i$  [164]. Note that, the required posterior distribution  $f(\theta_i | \mathbf{y}_i)$  can easily be obtained if  $\theta_i$  has a conjugate prior for the likelihood  $f_{\theta_i}(\mathbf{y}_i)$ . In this case,  $G_0(\theta_i)$  and  $f(\theta_i | \mathbf{y}_i)$  will belong to the same fam-

ily of distributions. A complete framework for DPMM-based signal classification algorithms in CR's is discussed thoroughly in Chapter 4.

In practice, the DPMM-based Gibbs sampling process can be described as in Algorithm 3. In this algorithm, the parameters  $\theta_i$ 's are selected sequentially in a Round-Robin scheme. This scheme is computationally inefficient since it keeps revisiting all the parameters uniformly, even after certain parameters may have converged. In order to improve the convergence rate of this process, we will define a parameter selection policy that selects specific parameters to be sampled at each iteration. This policy is described in details in Chapter 4.

---

**Algorithm 3** Gibbs sampling for DPMM classification.

---

Initialize  $\theta_i = \mathbf{y}_i, \forall i \in \{1, \dots, N\}$ .

**while** Convergence condition not satisfied **do**

**for**  $i = 1, \dots, N$  **do**

        Use Gibbs sampling to obtain  $\theta_i$  from the posterior distribution in (3.24).

**end for**

**end while**

---

### 3.4.2 Supervised Classification Methods in CR's

Unlike the unsupervised learning techniques discussed in the previous section that may be used in alien environments without having any prior knowledge, supervised learning techniques can generally be used in familiar/known environments with prior knowledge about the characteristics of the environment. In the following, we introduce some of the major supervised learning techniques that have been applied to classification tasks in CR's.

## **Artificial Neural Network**

The ANN has been motivated by the recognition that human brain computes in an entirely different way compared to the conventional digital computers [202]. A neural network is defined to be “a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use” [202]. An ANN resembles the brain in two respects [202]: 1) Knowledge is acquired by the network from its environment through a learning process and 2) interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

Some of the most beneficial properties and capabilities of ANN's include: 1) Nonlinear fitness to underlying physical mechanisms, 2) adaptation ability to minor changes in surrounding environment and 3) providing information about the confidence in the decision made. However, the disadvantages of ANN's are that they require training under many different environment conditions and their training outcomes may depend crucially on the choice of initial parameters.

Various applications of ANN's to CR's can be found in recent literature [150, 203–207]. The authors in [203], for example, proposed the use of multilayered feed-forward neural networks (MFNN's) as a technique to synthesize performance evaluation functions in CR's. The benefit of using MFNN's is that they provide a general-purpose black-box modeling of the performance as a function of the measurements collected by the CR; furthermore, this characterization can be obtained and updated by a CR at run-time, thus effectively achieving a certain level of learning capability. The authors in [203] also demonstrated in several IEEE 802.11 based environments how these modeling capabilities can be used for optimizing the configuration of a CR.

In [204], the authors proposed an ANN-based cognitive engine that learns how environmental measurements and the status of the network affect its performance

on different channels. In particular, an implementation of the proposed “cognitive controller” for dynamic channel selection in IEEE 802.11 wireless networks was presented. Performance evaluation carried out on an IEEE 802.11 wireless network deployment demonstrated that the cognitive controller is able to effectively learn how the network performance is affected by changes in the environment, and to perform dynamic channel selection thereby providing significant throughput enhancements.

In [205], an application of a feedbackward ANN in conjunction with the cyclostationary spectrum sensing was presented to perform spectrum sensing. The results showed that the proposed approach is able to detect the signals at considerably low SNR values. In [150], the authors designed a channel status predictor using a MFNN model. The authors argued that their proposed MFNN-based prediction is superior to the hidden Markov model (HMM)-based approaches, by pointing out that the HMM-based approaches require a huge memory space to store a large number of past observations with high computational complexity.

In [206], the authors proposed a methodology for spectrum prediction by modeling licensed user features as a multivariate chaotic time series, which is then input to an ANN that predicts the evolution of RF time series to decide if the unlicensed user can exploit the spectrum band. Experimental results showed a similar trend between predicted and observed values. This proposed spectrum evolution prediction method was done by exploiting the cyclostationary signal features to construct an RF multivariate time series that contain more information than the univariate time series, in contrast to most of the previously suggested modeling methodologies which focused on univariate time series prediction [208].

To illustrate the operation of ANN's in CR contexts, we present the model proposed in [130] and describe the main steps in the implementation of ANN's. In particular, [130] considers a multilayer perceptron (MLP) neural network which maps sets of input data onto a set of appropriate outputs. An MLP consists of multi-

### Chapter 3. Machine Learning in CR's

ple layers of nodes in a directed graph, which is fully connected from one layer to the next [130]. Except the input nodes, each node in the MLP is a neuron with a nonlinear activation function that computes a weighted sum of the up-layer output (denoted as the activation). An example of one of the most popular activation functions that is used in ANN's is the sigmoid function:

$$f(a) = \frac{1}{1 + e^{-a}} . \quad (3.25)$$

The ANN proposed in [130] has an input layer, output layer and multiple hidden layers. Note that, having additional hidden layers improves the nonlinear performance of the ANN in terms of classifying linearly non-separable data. However, adding more hidden layers makes the network more complicated and may require longer training time.

In the following, we consider an MLP network and let  $y_j^l$  to be the output of the  $j$ -th neuron in the  $l$ -th layer. Denote also by  $w_{ji}^l$  the weight between the  $j$ -th neuron in the  $l$ -th layer and the  $i$ -th neuron in the  $l - 1$ -th layer. The output  $y_j^l$  is given by:

$$y_j^l = \frac{1}{1 + e^{-\sum_i w_{ji}^l y_i^{l-1}}} . \quad (3.26)$$

During the training, the network tries to match the target value  $t_k$  to the output  $o_k$  of the  $k$ -th output neuron<sup>2</sup>. The error between the target and actual outputs is evaluated, for example, according to the mean-squared error (MSE):

$$MSE = \frac{1}{K} \sum_{k=1}^K (t_k - o_k)^2 , \quad (3.27)$$

where  $K$  is the number of output nodes. The update process will repeat until the MSE is smaller than a certain threshold.

The update rule can be performed according to a delta rule which adjusts the

---

<sup>2</sup>Since a certain target value (i.e. a label) is required during the training process, neural networks are considered as supervised learning algorithms.

weights  $w_{ji}^l$  by an amount [130]:

$$\Delta w_{ji}^l = \eta \delta_j^l y_i^{l-1} , \quad (3.28)$$

where  $\eta$  is a learning rate and  $\delta_j^l$  is defined as:

$$\delta_j^l = \begin{cases} o_j(t_j - o_j)(1 - o_j) & \text{if } l \text{ is the output layer} \\ y_j^l(1 - y_j^l) \sum_k \delta_k^{l+1} w_{kj}^{l+1} & \text{if } l \text{ is the hidden layer} \end{cases} .$$

The authors in [130] used the above described MLP neural network to implement a learner in a cognitive engine. By assuming a WiMax configurable radio technology, the learner is able to choose a certain modulation mode according to the SNR, such that a certain bit-error rate (BER) will be achieved. Thus, the inputs of the neural network consists of the code rate and SNR values and the output is the resulting SNR. By supplying training data to the neural network, the cognitive engine is trained to identify the BER that results from a certain choice of modulation, given a certain SNR level. By comparing the performance of different scales of neural networks, the simulation results in [130] showed that increasing the number of hidden layers reduces the speed of convergence but leads to a smaller MSE. However, more training data are required for larger number of hidden layers. Thus, given a certain set of training data, a trade-off must be made between the speed of convergence and the convergence accuracy of the neural network.

### Support Vector Machine

The SVM, developed by Vapnik and others [209], has been used for many machine learning tasks such as pattern recognition and object classifications. The SVM is characterized by the absence of local minima, the sparseness of the solution and the capacity control obtained by acting on the margin, or on other dimension independent quantities such as the number of support vectors [209]. SVM-based techniques have

achieved superior performances in a wide variety of real world problems due to their generalization ability and robustness against noise and outliers [111].

The basic idea of SVM's is to map the input vectors into a high-dimensional feature space in which they become linearly separable. This mapping from the input vector space to the feature space is a non-linear mapping which is achieved by using kernel functions. Depending on the application different types of kernel functions can be used. A common choice for classification problems is the Gaussian kernel which is a polynomial kernel of infinite degree. In performing classification, a hyperplane which allows for the largest generalization in this high-dimensional space is found. This is so-called a maximal margin classifier [210]. Note that, the margin is defined as the distance from a separating hyperplane to the closest data points. As shown in Fig. 3.7, there could be many possible separating hyperplanes between the two classes of data, but only one of them allows for the maximum margin. The corresponding closest data points are named support vectors and the hyperplane allowing for the maximum margin is called an optimal separating hyperplane. The interested reader is referred to [78, 151, 211] for insightful discussion on SVM's.

An SVM-based classifier was described in [151] for signal classification in CR's. The classifier in [151] assumed a training set  $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$  with  $x \in \mathbb{R}^N$  and  $y \in \{-1, 1\}$ . The objective is to find a hyperplane:

$$\mathbf{w}^T \varphi(\mathbf{x}) + b = 0 , \tag{3.29}$$

where  $\varphi$  can be a non-linear function that maps  $\mathbf{x}$  into a higher dimensional Hilbert space [211],  $\mathbf{w}$  is a weight vector and  $b$  is a scalar parameter. In general, it is not possible to obtain an expression for the mapping function  $\varphi$ . However, this function can be characterized by a Kernel function  $K(\mathbf{x}_i, \mathbf{x}_j)$  and, as it turns out fortunately, the Kernel function is sufficient to optimize the parameters  $\mathbf{w}$  and  $b$  in (3.29) [211].

The hyperplane in (3.29) is assumed to separate the data into two classes such that the distance between the closest points of each class to the hyperplane is maximized.

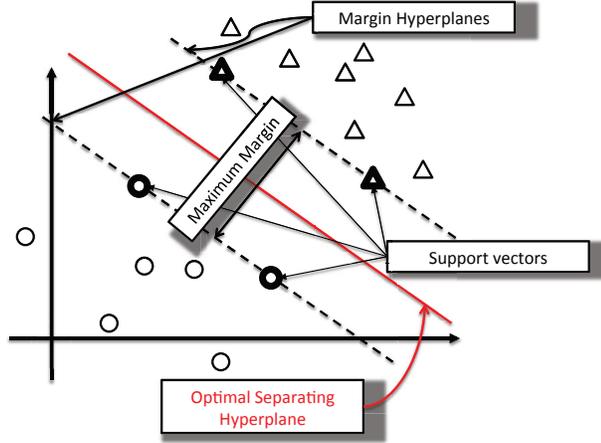


Figure 3.7: A diagram showing the basic idea of SVM: optimal separation hyperplane (solid red line) and two margin hyperplanes (dashed lines) in a binary classification example; Support vectors are bolded.

This can be achieved by minimizing the norm  $\|\mathbf{w}\|^2$  [211].

In order to solve the optimization problem, the slack variables  $\{\xi_i, i = 1, \dots, l\}$  are introduced and the optimization problem can be formulated as [151]:

$$\min_{\mathbf{w}, b, \xi_i} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i \quad (3.30)$$

$$\text{s.t. } y_i (\mathbf{w}^T \varphi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \forall i = 1, \dots, l \quad (3.31)$$

$$\xi_i \geq 0, \forall i = 1, \dots, l \quad (3.32)$$

where  $C$  is the penalty parameter that controls the training error.

The Lagrangian of the above optimization problem can be written as:

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \beta_i \xi_i - \sum_{i=1}^l \alpha_i [\mathbf{w}^T \varphi(\mathbf{x}_i) + b - 1 + \xi_i] , \quad (3.33)$$

where  $\alpha_i, \beta_i \geq 0$  are the Lagrange multipliers. By computing the derivatives with respect to  $\mathbf{w}$ ,  $b$  and  $\xi_i$ , the dual representation of the optimization problem can be

Chapter 3. Machine Learning in CR's

expressed as [151]:

$$\begin{aligned} \max_{(\alpha_1, \dots, \alpha_l)} \quad & \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \forall i = 1, \dots, l \\ & \sum_{i=1}^l y_i \alpha_i = 0 \end{aligned}$$

where  $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$  is the Kernel function.

In this case, the decision function (i.e. the learning machine [211]) is computed as:

$$f(x) = \text{sgn} \left\{ \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right\}. \quad (3.34)$$

Other applications of SVM's to CR can be found in current literature, including [46, 78, 111, 151, 212–217]. Most of these applications of the SVM in CR context, however, has been for performing signal classification.

In [214], for example, an MAC protocol classification scheme was proposed to classify contention-based and control-based MAC protocols in an unknown primary network based on SVM's. To perform the classification in an unknown primary network, the mean and variance of the received power are chosen as two features for the SVM. The SVM is embedded in a CR terminal of the secondary network. A time division multiple access (TDMA) and a slotted Aloha network were setup as the primary networks. Simulation results showed that TDMA and slotted Aloha MAC protocol could be effectively classified by the CR terminal and the correct classification rate was proportional to the transmission rate of the primary networks, where the transmission rate for the primary networks is defined as the new packet generating/arriving probability in each time slot. The reason for the increase in the correct classification rate when the transmission rate increases is the following: for slotted Aloha network, the higher transmission rate brings the higher collision

probability, and thus the higher instantaneous received power captured by a CR terminal; for TDMA network, however, there is no relation between transmission rate and instantaneous captured received power. Therefore, when the transmission rates of both primary networks increase, it makes a CR terminal easier to differentiate TDMA and slotted Aloha.

Support vector machine classifiers can not only be a binary classifier as shown in the previous example, but also it can be easily used as a multi-class classifiers by treating a  $K$ -class classification problem as  $K$  two-class problems. For example, in [215] the authors presented a study of multi-class signal classification based on automatic modulation classification (AMC) through SVM's. A simulated model of an SVM signal classifier was implemented and trained to recognize seven distinct modulation schemes; five digital (BPSK, QPSK, GMSK, 16-QAM and 64-QAM) and two analog (FM and AM). The signals were generated using realistic carrier frequency, sampling frequency and symbol rate values, and realistic raised-cosine and Gaussian pulse shaping filters. The results showed that the implemented classifier can correctly classify signals with high probabilities.

### 3.5 Centralized and Decentralized Learning in CR

Since noise uncertainties, shadowing, and multi-path fading effects limit the performance of spectrum sensing, when the received primary SNR is too low, there exists an SNR wall, below which reliable spectrum detection is impossible in some cases [218,219]. If secondary users cannot detect the primary transmitter, while the primary receiver is within the secondary users transmission range, a hidden terminal problem occurs [220,221], and the primary user's transmission will be interfered with. By taking advantage of diversity offered by multiple independent fading channels (multiuser diversity), cooperative spectrum sensing improves the reliability of

Chapter 3. Machine Learning in CR's

		Spectrum Sensing and MAC Protocols	Signal Classification and Feature Detection	Power Allocation and Rate adaptation	System Parameters Reconfiguration	Pros	Cons
Unsupervised learning techniques	Reinforcement learning (RL)	x				Optimal solution for MDP's	In general, suboptimal for POMDP's, DEC-MDP's and DEC-POMDP's
	Non-parametric Learning: DPMM		x			Does not require prior knowledge about the number of mixture components	Requires large number of iterations, compared to parametric methods
	Game theory-based Learning	x		x		Suitable for multi-player decision problems	Requires knowledge of different parameters (e.g. SINR, power, price from base stations, etc.) which is impractical in many situations
	Threshold Learning				x	Suitable for controlling specific parameters under uncertainty conditions	Requires training data
Supervised learning techniques	Artificial Neural Network (ANN)		x			Does not require prior knowledge of the distribution of the observed process	<ul style="list-style-type: none"> <li>▪ Suffers from <i>overfitting</i></li> <li>▪ Requires data labeling</li> </ul>
	Support Vector Machine (SVM)		x			Has better performance for small training examples, compared to ANN	<ul style="list-style-type: none"> <li>▪ Requires prior knowledge of the distribution of the observed process</li> <li>▪ Requires data labeling</li> </ul>

Figure 3.8: A comparison among the learning algorithms that are presented in this survey.

spectrum sensing and the utilization of idle spectrum [158, 159], as opposed to non-cooperative spectrum sensing.

In centralized cooperative spectrum sensing [158, 159], a central controller collects local observations from multiple secondary users, decides the spectrum occupancy by using decision fusion rules, and informs the secondary users which channels to access. In distributed cooperative spectrum sensing [222, 223], on the other hand, secondary users within a CRN exchange their local sensing results among themselves without requiring a backbone or centralized infrastructure. On the other hand, in the non-cooperative decentralized sensing framework, no communications are assumed among the secondary users [224].

### *Chapter 3. Machine Learning in CR's*

In [144], the authors showed how various centralized and decentralized spectrum access markets (where CR's can compete over time for dynamically available transmission opportunities) can be designed based on a stochastic game (discussed above in Section 3.3.3) framework and solved using a learning algorithm. Their proposed learning algorithm was to learn the following information in the stochastic game: state transition model, state and the policy of other secondary users and the network resource state. The proposed learning algorithm was similar to Q-learning. However, the main difference compared to Q-learning was that it explicitly considered the impact of other secondary user actions through the state classifications and transition probability approximation. The computational complexity and performance were also discussed in [144].

In [27], the authors proposed and analyzed both a centralized and a decentralized decision-making architecture with RL for the secondary CRN. In this work, a new way to encourage primary users to lease their spectrum was proposed: the secondary users place bids indicating how much power they are willing to spend for relaying the primary signals to their destinations. In this formulation, the primary users achieve power savings due to asymmetric cooperation. In the centralized architecture, a secondary system decision center (SSDC) selects a bid for each primary channel based on optimal channel assignment for secondary users. In a decentralized CRN architecture, an auction game-based protocol was proposed in which each secondary user independently places bids for each primary channel and receivers of each primary link pick the bid that will lead to the most power savings. A simple and robust distributed RL mechanism was developed to allow the users to revise their bids and to increase their subsequent rewards. The performance results given in [27] showed the significant impact of RL in both improving spectrum utilization and meeting individual secondary user performance requirements.

In general, there is always a trade-off between the centralized and decentralized

control in radio networks. This is also true for CRN's. While the centralized schemes ensure efficient management of the spectrum resources, they often suffer from signaling and processing overhead. On the other hand, a decentralized scheme can reduce the complexity of the decision-making in cognitive networks. However, radios that act according to a decentralized scheme may adopt a selfish behavior and try to maximize their own utilities, at the expense of the sum-utility of the network (social welfare), leading to overall network inefficiency. This problem can become particularly severe when considering heterogeneous networks in which different nodes belong to different types of systems and have different objectives (usually conflicting objectives). To resolve this problem, [225] proposes a hybrid approach for heterogeneous CRN's where the wireless users are assisted in their decisions by the network which broadcasts aggregated information to the users [225]. At some states of the system, the network manager imposes its decisions on users in the network. In other states, the mobile nodes may take autonomous actions in response to the information sent by the network center. As a result, the model in [225] avoids having a completely decentralized network, due to possible inefficiency of such non-cooperative networks. Nevertheless, a large part of the decision-making is still delegated to the mobile nodes to reduce the processing overhead at the central node.

## **3.6 Conclusion**

In this chapter, we have characterized the learning problems in CR's and stated the importance of machine learning in developing real CR's. We have presented the state-of-the-art learning methods that have been applied to CR's classifying them under supervised and unsupervised learning. A discussion of some of the most important, and commonly used, learning algorithms was provided along with their advantages and disadvantages. We also showed some of the challenging learning

### *Chapter 3. Machine Learning in CR's*

problems encountered in CR's and presented possible solution methods to address them. In the following chapters, we present two machine learning frameworks for CR's addressing both signal classification and decision-making methods, respectively.

# Chapter 4

## Bayesian Non-Parametric Classification using the Dirichlet Process

### 4.1 Introduction

Signal classification has been identified as an important task for CR's [42, 79, 93]. Several feature detection and signal classification methods have been proposed in the literature. For example, [68] proposed a cyclostationarity-based feature detection and an HMM-based signal classification for CR's. However, this technique requires prior training with ideal feature vectors for each signal type, which may not be possible if the CR is operating in an unknown environment without any prior knowledge of the existing signal types. Other classification methods have also been proposed based on neural networks [77] and SVM's [78], but they also required training data to initialize the classifiers' parameters. On the other hand, feature classification can be performed based on parametric classification approaches such as the GMM or K-

means algorithm that do not require training data. However, these techniques assume a fixed number of classes, which may not be known in an alien RF environment in which the number of active wireless systems is unknown *a priori*. As an alternative, the authors in [79] proposed to use the X-means algorithm [80] for unsupervised signal classification when the number of clusters is unknown. This approach is based on the K-means algorithms but approximates the number of clusters  $X$  by maximizing either the BIC or the AIC [80]. However, similarly to the K-means algorithm, the X-means algorithm assumes spherical Gaussian data, which does not offer enough flexibility when dealing with observations having an arbitrary noise distribution [80]. Moreover, the K-means algorithm can only converge to a local minimum of the distortion measure and its performance heavily depends on the choice of initial center points [80].

To resolve these drawbacks, we resort to non-parametric classification approaches. In particular, the DPMM that assumes no prior knowledge of the number of clusters [43]. Note that, the DPMM-based classifier is considered to be a Bayesian non-parametric method in the sense of allowing the structure of the model (i.e. number of clusters) to grow with the complexity of the data [43, 81–84]. However, the individual observations of the DPMM can still be drawn from parametric distributions. The DPMM-based classifier can infer the number of clusters (or mixture components) from the data itself, making it a suitable candidate for unsupervised and autonomous classifiers. This approach has been previously applied for galaxy clustering [85], speaker diarization [86], speaker adaptation [87], image segmentation [88] and compressive sensing [89]. In this chapter, we propose the DPMM classification approach to infer the number and types of wireless systems that are sensed by a CR in an unknown environment. The non-parametric nature of the DPMM allows for an arbitrary number of clusters and helps the CR to learn and act autonomously in any RF environment.

Note that, most of the existing DPMM classifiers assume Gaussian observation models, which may not accurately represent complex observations encountered in wireless systems [40,41,43,87–89,164,226]. In this work, hence, we extend the DPMM framework to both Gaussian and non-Gaussian observation models by allowing the cluster parameters to be drawn from a mixture model where each mixture component is used to parameterize a particular observation model, including both Gaussian and non-Gaussian distributions. By applying the Gibbs sampling, we determine the observation model that best fits each cluster, while estimating the corresponding parameters. To the best of our knowledge, this is the first DPMM that assumes such a framework, thus offering flexibility in handling arbitrary observation models, as opposed to both K-means and X-means algorithms which assume spherical Gaussian observations [80].

Most of DPMM formulations, however, require an extensive number of Gibbs sampling iterations making them computationally prohibitive in real-time operation. Hence, we propose a novel Gibbs sampling algorithm, referred to as the *simplified Gibbs sampler*, which improves the convergence rate of the DPMM classifier. The proposed algorithm is based on a *parameter selection policy* that carefully selects specific parameters to be updated at each Gibbs sampling iteration, instead of sequentially or randomly selecting all parameters. Hence, the proposed algorithm is shown to improve the efficiency of the Gibbs sampling-based DPMM classifier and makes a suitable candidate for large-scale classification problems.

Furthermore, we propose a *sequential Gibbs sampler* that is suitable for real-time operation. Given a new feature vector input, the proposed sequential Gibbs sampler first classifies it into a certain cluster. The DPMM approach allows this to be either one of the already existing clusters or a new cluster. In order to achieve good performance, the sequential Gibbs sampler requires a training period, which can be implemented using the above simplified Gibbs sampler. As a result, the

obtained sequential Gibbs sampler ensures real-time classification, which makes it an alternative solution to simple parametric classifiers, yet without requiring additional information about the observation model.

The remainder of this chapter is organized as follows: In Section 4.2, we describe the Bayesian DPMM classification method and in Section 4.3 we derive the predictive distribution of the observed feature points. The convergence of the algorithm is discussed in Section 4.4 and we derive the MSE of the cluster means in Section 4.5. Simplified and sequential DPMM-based Gibbs samplers are proposed in Sections 4.6 and 4.7, respectively. Simulation results are presented in Section 4.8 and we conclude the paper in Section 4.9.

## 4.2 Data Clustering based on the DPMM and the Gibbs Sampling

Consider a sequence of observations  $\mathbf{y}_{1:N} \triangleq \{\mathbf{y}_i\}_{i=1}^N$ , where  $\mathbf{y}_i \triangleq [y_{i,1}, \dots, y_{i,d}]^T \in \mathbb{R}^d$ , and assume that these observations are drawn from a mixture model. If we do not know the number of mixture components, it is reasonable to assume a non-parametric model, such as the DPMM which allows the number of mixture components to increase with the complexity of the data. Thus, let us assume that the mixture components  $\theta_i$  are drawn from a  $G \sim DP(\alpha_0, G_0)$ , for  $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$ , where  $\phi_k$  are the unique values of  $\theta_i$  and  $\pi_k$  their corresponding probabilities.

The problem is to estimate the mixture component  $\hat{\theta}_i$  for each observation  $\mathbf{y}_i$ , for all  $i \in \{1, \dots, N\}$ . In particular, we are interested in finding MAP estimates of  $\theta_i$  ( $i = 1, \dots, N$ ), given the observations  $\mathbf{y}_{1:N}$ . However, it is hard to find analytical MAP estimates of  $\theta_i$ 's since the joint distribution of  $(\theta_1, \dots, \theta_N)$ , given  $\mathbf{y}_{1:N}$ , is unknown. As an alternative, we may use Monte Carlo methods to compute the MAP

estimates by sampling from the posterior distribution of  $\theta_i$ 's, given  $\mathbf{y}_{1:N}$  [199, 200]. In particular, in situations that we have the conditional distribution of each  $\theta_i$ , given the other parameters  $\{\theta_j\}_{j \neq i}$ , as in the DPMM formulation, we can construct an MCMC algorithm based on Gibbs sampling to draw samples from the joint posterior distribution of  $(\theta_1, \dots, \theta_N)$  [201]. The Gibbs sampling algorithm starts with arbitrary estimates of  $\theta_i$ 's and draws samples from the conditional distribution of each parameter  $\theta_i$ , given the other parameters  $\{\theta_j\}_{j \neq i}$ , where  $\{\theta_j\}_{j \neq i}$  take the values of their most recent estimates [201]. It can be shown that these samples converge in probability to the actual posterior distribution of  $(\theta_1, \dots, \theta_N)$ , thus leading to an efficient method for estimating  $\theta_i$ 's [85].

By assuming a DPMM framework, the posterior distribution of  $\theta_i | \{\theta_j\}_{j \neq i}, \mathbf{y}_{1:N}$  can be computed as [164]:

$$\theta_i | \{\theta_j\}_{j \neq i}, \mathbf{y}_{1:N} \begin{cases} = \theta_j & \text{with prob. } q_j = \frac{f_{\theta_j}(\mathbf{y}_i)}{\alpha_0 f(\mathbf{y}_i) + \sum_{j=1, j \neq i}^N f_{\theta_j}(\mathbf{y}_i)} \\ \sim f(\theta_i | \mathbf{y}_i) & \text{with prob. } q_0 = \frac{\alpha_0 f(\mathbf{y}_i)}{\alpha_0 f(\mathbf{y}_i) + \sum_{j=1, j \neq i}^N f_{\theta_j}(\mathbf{y}_i)} \end{cases}, \quad (4.1)$$

where  $f(\mathbf{y}_i) = \int_{\theta} f_{\theta}(\mathbf{y}_i) G_0(\theta) d\theta$  is the marginal distribution of  $\mathbf{y}_i$ , assuming a prior  $G_0(\theta)$ , and  $f_{\theta}(\mathbf{y}_i) \triangleq f(\mathbf{y}_i | \theta_i = \theta)$ , for all  $\theta$ 's, where  $\theta_i$  stands for the parameter of observation  $\mathbf{y}_i$ . In other words, the assumption of an underlying DPMM for the cluster parameters  $\theta_i$ 's implies that  $\theta_i$  is equal to  $\theta_j$  with probability  $q_j$ , or it is a new value drawn according to the conditional distributions  $f(\theta_i | \mathbf{y}_i)$  with probability  $q_0$ . Note that, the required posterior distribution  $f(\theta_i | \mathbf{y}_i)$  can easily be obtained if  $\theta_i$  has a conjugate prior for the likelihood  $f_{\theta_i}(\mathbf{y}_i)$ <sup>1</sup>. In this case,  $G_0(\theta_i)$  and  $f(\theta_i | \mathbf{y}_i)$  will belong to the same family of distributions. In particular, if both the prior distribution  $G_0(\theta_i)$  and the likelihood function  $f_{\theta_i}(\mathbf{y}_i)$  are Gaussian, then the posterior

---

<sup>1</sup>If the posterior distribution  $p(\theta|x)$  is in the same family as the prior probability distribution  $p(\theta)$ , the prior and posterior are then called conjugate distributions, and the prior is called a *conjugate prior for the likelihood*. All the members of the exponential family have conjugate priors. In particular, the normal, gamma, exponential, Wishart and inverse-Wishart distributions have conjugate priors [227].

distribution  $f(\theta_i|\mathbf{y}_i)$  will also be Gaussian. Thus, most of the literature on DPMM problems assumes conjugate priors [43, 88, 164]. In the following, we first present the Gibbs sampling algorithm for the multivariate Gaussian case and then generalize the model to a mixture of Gaussian and non-Gaussian observations.

### 4.2.1 DPMM-based Clustering with a Gaussian Observation Model

A Gibbs sampling algorithm for estimating the parameters  $\theta_i$  of a DPMM was proposed in [164], which showed that the outcomes of the developed algorithm converge, in probability, to those of the posterior distribution of  $(\theta_1, \dots, \theta_N)$ , given  $\mathbf{y}_{1:N}$ . However, [164] assumed that the prior distribution  $G_0(\theta_i)$  can be chosen as a uniform distribution, presuming prior knowledge of the range of the observations, which, in general, may not be available. In addition, it also assumed that the observations  $\mathbf{y}_{1:N}$  are distributed according to a standard Gaussian distribution, given the parameters  $\theta_i$ 's. This assumption was relaxed in [85] in which a Bayesian method was proposed to estimate both mean and variance of the Gaussian observation model from the observations  $\mathbf{y}_{1:N}$ .

In this section, we follow an approach similar to [85] in developing a multi-dimensional Bayesian non-parametric estimator for DPMM's. In the next section, we generalize this method to non-Gaussian observation models.

Let us assume a sequence of observations  $\mathbf{y}_{1:N}$  from a DPMM that are normally distributed given the mixture component parameters  $\theta_{1:N} \triangleq \{\theta_i\}_{i=1}^N$ . We may thus denote  $\mathbf{y}_i|\theta_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{V}_i)$ , where  $\theta_i = (\boldsymbol{\mu}_i, \mathbf{V}_i)$  for  $i \in \{1, \dots, N\}$ . The prior distribution  $G_0(\theta_i)$  can be modeled as the normal/inverse-Wishart conjugate prior such that  $\mathbf{V}_i^{-1} \sim W(\mathbf{S}/2, s/2)$ , where  $W(\mathbf{S}/2, s/2)$  is the Wishart distribution with a positive definite scale matrix  $\mathbf{S}/2$  and  $s/2$  degrees of freedom, and  $\boldsymbol{\mu}_i|\mathbf{V}_i \sim \mathcal{N}(\mathbf{m}, \tau\mathbf{V}_i)$ ,

for some mean  $\mathbf{m}$  and scale factor  $\tau > 0$ . Note that, this is the most commonly used conjugate prior distribution for the mean and the covariance matrix of a multivariate Gaussian observation model<sup>2</sup>. Furthermore, a large value of  $\tau$  implies a large dispersion among the cluster means, whereas parameter  $\mathbf{m}$  is a prior estimate of these means [85].

On the other hand, the parameter  $s$  reflects the confidence in the value of the covariance matrix  $\mathbf{V}_i$ . That is, a large value of  $s$  corresponds to the case where  $\mathbf{V}_i$  is believed to be approximately equal to its prior estimate  $\mathbf{S}$ . However, a small value of  $s$  corresponds to the case where little knowledge is available about  $\mathbf{V}_i$  [85].

The posterior distribution  $f(\theta_i|\mathbf{y}_i)$  is a bivariate normal/inverse-Wishart distribution whose components are [85]:

$$\begin{aligned}\mathbf{V}_i^{-1} &\sim W\left(\frac{\mathbf{S}_i}{2}, \frac{1+s}{2}\right), \\ \boldsymbol{\mu}_i|\mathbf{V}_i &\sim \mathcal{N}(\mathbf{x}_i, X\mathbf{V}_i),\end{aligned}$$

where  $\mathbf{S}_i = \mathbf{S} + \frac{(\mathbf{y}_i - \mathbf{m})(\mathbf{y}_i - \mathbf{m})^T}{1+\tau}$ ,  $X = \frac{\tau}{1+\tau}$  and  $\mathbf{x}_i = \frac{\mathbf{m} + \tau\mathbf{y}_i}{1+\tau}$ . The corresponding weights  $q_0$  and  $q_j$  in (4.1) can shown to be [85]:

$$q_0 \propto \frac{\alpha_0 c(s)}{|\mathbf{M}|^{1/2}} \left(1 + \frac{(\mathbf{y}_i - \mathbf{m})^T \mathbf{M}^{-1} (\mathbf{y}_i - \mathbf{m})}{s}\right)^{-(1+s)/2}$$

and

$$q_j \propto \frac{1}{\sqrt{2|\mathbf{V}_j|}} e^{-\frac{(\mathbf{y}_j - \boldsymbol{\mu}_j)^T \mathbf{V}_j^{-1} (\mathbf{y}_j - \boldsymbol{\mu}_j)}{2}},$$

for  $j \in \{1, \dots, N\}$ ,  $j \neq i$  and subject to  $\sum_{j=1, j \neq i}^N q_j = 1$ , with  $\mathbf{M} = \frac{1+\tau}{s}\mathbf{S}$  and  $c(s) = \Gamma(\frac{1+s}{2})\Gamma(\frac{s}{2})s^{-1/2}$ .

We may use the above posterior marginal distribution to perform Gibbs sampling. The resulting number of distinct values of  $\theta_{1:N}$  (denoted by  $\{\phi_k\}_{k=1}^K$ ) is then an

---

<sup>2</sup>Note that, families of conjugate priors are not unique. In particular, the set of all probability distributions is always a conjugate prior.

estimate of the number of components (or clusters) in the mixture model. Algorithm 3 summarizes this DPMM classification procedure based on the Gibbs sampling. Upon convergence, the observations  $\mathbf{y}_i$ 's that share identical values of  $\theta_i$ 's are assumed to belong to the same cluster.

### 4.2.2 DPMM-based Clustering with a Mixture of Gaussian and non-Gaussian Priors for $\theta_i$

Most of the existing DPMM-based classification problems assume that the observations  $\mathbf{y}_{1:N}$  are normally distributed, given the cluster parameters  $\theta_i$ 's [43, 88, 164]. In this work, however, we relax this condition to allow  $\mathbf{y}_i|\theta_i$  to be non-Gaussian distributed. In modifying the likelihood  $f_{\theta_i}(\mathbf{y}_i)$ , however, we also need to adapt the prior distribution of  $\theta_i$  accordingly so that it is a conjugate prior for the assumed likelihood. This is necessary since if we were to lose the conjugate property of the prior, a closed-form expression for the posterior distribution of  $\theta_i$ , as in (4.1), may not be possible. For example, the Gaussian prior is conjugate for the Gaussian likelihood. However, if we were to use a different likelihood function, such as the log-normal distribution, the Gaussian prior is no more conjugate for this particular likelihood. In this case, a possible conjugate prior would be the Gamma distribution [228]. Thus, modifying the likelihood  $f_{\theta_i}(\mathbf{y}_i)$  should be done in conjunction with adapting the prior distribution of  $\theta_i$ , accordingly.

Hence, we allow the likelihood function  $f_{\theta_i}(\mathbf{y}_i)$  to belong to one of the  $L$  different distributions (e.g. Gaussian, or Gamma or log-normal, etc.). The parameter  $\theta_i$  denotes the distribution parameter and we let  $Z_i \in \{1, \dots, L\}$  to denote the distribution index, which specifies the type of the distribution  $f_{\theta_i}(\mathbf{y}_i)$ . Clearly,  $\theta_i$  can be modeled as a *mixture model* of  $L$  components where each component is a random parameter drawn from a certain set  $\mathcal{S}_l$ , for  $l = 1, \dots, L$ . The set  $\mathcal{S}_l$  contains all pos-

sible parameters of the  $l$ -th distribution model. By following a Bayesian approach, we can estimate the parameters  $\theta_i$ 's, given the observations  $\mathbf{y}_i$ 's, by using (4.1).

We denote a discrete prior distribution for  $Z_i$  such that  $P\{Z_i = l\} \triangleq \kappa_l$ , for  $l = 1, \dots, L$ . Given a certain observation model  $Z_i = l$  for the observation  $\mathbf{y}_i$ , we denote the conditional prior distribution of  $\theta_i$  as  $\theta_i|\{Z_i = l\} \sim G_0^{(l)}(\theta_i)$ , where  $\theta_i \in \mathcal{S}_l$ .

We define  $f_{\theta}^{(l)}(\mathbf{y}_i) \triangleq f(\mathbf{y}_i|\theta_i = \theta, Z_i = l)$ , for all  $\theta \in \mathcal{S}_l$ , to be the likelihood function of the observation  $\mathbf{y}_i$ , given that  $Z_i = l$ . Thus, we can write  $\mathbf{y}_i|\{\theta_i, Z_i\} \sim f_{\theta_i}^{(1)}(\mathbf{y}_i)\mathcal{I}_{\{Z_i=1\}} + \dots + f_{\theta_i}^{(L)}(\mathbf{y}_i)\mathcal{I}_{\{Z_i=L\}}$ , where the indicator function  $\mathcal{I}_A$  is defined as  $\mathcal{I}_A = 1$  if the event  $A$  is true, and 0 otherwise. Note that, the distribution of  $\mathbf{y}_i|\{\theta_i, Z_i\}$  is defined for  $\theta_i \in \mathcal{S}_{Z_i}$  such that  $\theta_i$  is a valid parameter for the  $Z_i$ -th distribution model.

Under the above formulation, the posterior distribution of the parameter  $\theta_i$ , given the observation  $\mathbf{y}_i$ , is defined over the set  $\mathcal{S} \triangleq \bigcup_{l=1}^L \mathcal{S}_l$  such that:

$$\begin{aligned}
 f(\theta_i|\mathbf{y}_i) &= \sum_{l=1}^L f(\theta_i, Z_i = l|\mathbf{y}_i) \\
 &= \sum_{l=1}^L f(\theta_i|\mathbf{y}_i, Z_i = l)P\{Z_i = l|\mathbf{y}_i\} \\
 &= \sum_{l=1}^L \hat{\kappa}_{l,i}f(\theta_i|\mathbf{y}_i, Z_i = l) ,
 \end{aligned} \tag{4.2}$$

where

$$\begin{aligned}
 \hat{\kappa}_{l,i} &\triangleq P\{Z_i = l | \mathbf{y}_i\} \\
 &= \frac{P\{Z_i = l\}f(\mathbf{y}_i | Z_i = l)}{\sum_{l'=1}^L P\{Z_i = l'\}f(\mathbf{y}_i | Z_i = l')} \\
 &= \frac{\kappa_l f(\mathbf{y}_i | Z_i = l)}{\sum_{l'=1}^L \kappa_{l'} f(\mathbf{y}_i | Z_i = l')} \\
 &= \frac{\kappa_l \int_{\theta \in \mathcal{S}_l} f_{\theta}^{(l)}(\mathbf{y}_i) G_0^{(l)}(\theta) d\theta}{\sum_{l'=1}^L \kappa_{l'} \int_{\theta \in \mathcal{S}_{l'}} f_{\theta}^{(l')}(\mathbf{y}_i) G_0^{(l')}(\theta) d\theta} , \tag{4.3}
 \end{aligned}$$

and  $f(\theta_i | \mathbf{y}_i, Z_i = l) = 0$  if  $\theta_i \notin \mathcal{S}_l$ . In general, if a closed-form expression can not be obtained for (4.3),  $\hat{\kappa}_{l,i}$  can be evaluated numerically.

The expression in (4.2) implies that  $\theta_i$  can be sampled from the posterior distribution  $f(\theta_i | \mathbf{y}_i, Z_i = l)$  with a probability  $\hat{\kappa}_{l,i}$ , for  $l = 1, \dots, L$ . In other words, given an observation  $\mathbf{y}_i$ , the distribution index  $Z_i$  is first sampled from the discrete set  $\{1, \dots, L\}$ , with corresponding probabilities  $\{\hat{\kappa}_{l,i}\}_{l=1}^L$ . Given the sampled value of  $Z_i$ ,  $\theta_i$  can be sampled from  $\mathcal{S}_{Z_i}$  using the posterior distribution  $f(\theta_i | \mathbf{y}_i, Z_i)$ . Furthermore, if  $f(\theta_i | \mathbf{y}_i, Z_i = l)$  and  $G_0^{(l)}(\theta_i)$  are conjugate for the likelihood  $f_{\theta_i}^{(l)}(\mathbf{y}_i)$ ,  $\forall l \in \{1, \dots, L\}$ , then the posterior in (4.2) can be expressed in closed-form. If not, the posterior may not be derived in closed-form. However, the approach can still be used with numerical methods.

The marginal distribution of the observation  $\mathbf{y}_i$  can be computed as:

$$f(\mathbf{y}_i) = \sum_{l=1}^L \kappa_l \int_{\theta \in \mathcal{S}_l} f_{\theta}^{(l)}(\mathbf{y}_i) G_0^{(l)}(\theta) d\theta . \tag{4.4}$$

By substituting (4.2) and (4.4) in (4.1), we obtain the posterior distribution of  $\theta_i | \{\theta_j\}_{j \neq i}, \mathbf{y}_{1:N}$ .

An Example (Clustering with a mixture of Gamma, log-normal and Gaussian observation models):

For example, let us assume that  $\mathbf{y}_i = [y_{i,1}, \dots, y_{i,d}]^T \in \mathbb{R}^d$  and  $L = 3$ , so that each  $\mathbf{y}_i|\theta_i$  is a mixture of Gaussian, Gamma and log-normal distributions. For analytical tractability, the likelihood functions of the observations  $\mathbf{y}_i$ 's are selected so that the prior and posterior distributions of  $\theta_i$  are conjugate. We also assume that the elements of  $\mathbf{y}_i$ 's are independent in the case of non-Gaussian observation models.

First, as in Section 4.2.1, we may define  $\mathcal{S}_1 \triangleq \mathbb{R}^d \times \mathbb{R}^{d \times d}$  to be the set of possible parameters of the Gaussian likelihood function corresponding to  $\theta_i|\{Z_i = 1\} \triangleq (\boldsymbol{\mu}_i, \mathbf{V}_i)$ . In this case, the likelihood  $f_{\theta_i}^{(1)}(\mathbf{y}_i)$ , the posterior  $f(\theta_i|\mathbf{y}_i, Z_i = 1)$ , the marginal  $\int_{\theta \in \mathcal{S}_1} f_{\theta}^{(1)}(\mathbf{y}_i)G_0^{(1)}(\theta)d\theta$  and the prior  $G_0^{(1)}(\theta_i)$  can be computed as described in Section 4.2.1.

Next, we define  $\mathcal{S}_2 \triangleq \mathbb{R}^d$ , such that  $\theta_i|\{Z_i = 2\} \triangleq \mathbf{a}$ , where  $\mathbf{a} = [a_1, \dots, a_d]^T$  are the shape parameters of a Gamma distributed likelihood function (assuming fixed rate parameters  $\{b_k\}_{k=1}^d$ ) such that:

$$f_{\theta}^{(2)}(\mathbf{y}_i) = \prod_{k=1}^d \frac{b_k^{a_k}}{\Gamma(a_k)} y_{i,k}^{a_k-1} e^{-b_k y_{i,k}}, \quad (4.5)$$

where we have let  $\theta = \mathbf{a}$ , i.e.  $y_{i,k}|\{\theta_i = \theta, Z_i = 2\} \sim Ga(a_k, b_k)$  and are independent. Note that, (4.5) denotes the likelihood of observation  $\mathbf{y}_i$  joining a cluster with parameter  $\theta$ . In this case, to preserve the conjugate property, the prior distribution of  $\mathbf{a}$  is assumed to be equal to:

$$G_0^{(2)}(\theta_i) = G_0^{(2)}(\mathbf{a}) = \prod_{k=1}^d \frac{1}{J(a_0, b_0, b_k, c_0)} \cdot \frac{a_0^{a_k-1} b_k^{c_0 a_k}}{\Gamma(a_k)^{b_0}}, \quad (4.6)$$

where  $a_0, b_0$  and  $c_0$  are the corresponding hyper-parameters and  $J(a_0, b_0, b_k, c_0) \triangleq \int_0^{\infty} \frac{a_0^{x-1} b_k^{c_0 x}}{\Gamma(x)^{b_0}} dx$  is the normalization term. The posterior distribution of  $\theta_i$  can be

obtained as in [228] and can be shown to be equal to:

$$f(\theta_i | \mathbf{y}_i, Z_i = 2) = f(\mathbf{a} | \mathbf{y}_i) = \prod_{k=1}^d \frac{1}{J(a_0 y_{i,k}, b_0 + 1, b_k, c_0 + 1)} \cdot \frac{(a_0 y_{i,k})^{a_k - 1} b_k^{(c_0 + 1) a_k}}{\Gamma(a_k)^{b_0 + 1}}. \quad (4.7)$$

The marginal distribution of  $\mathbf{y}$  can thus be computed as:

$$f(\mathbf{y}_i | Z_i = 2) = \prod_{k=1}^d \int_0^\infty \frac{b_k^z}{\Gamma(z)} y_{i,k}^{z-1} e^{-b_k y_{i,k}} \frac{a_0^{z-1} b_k^{c_0 z}}{\Gamma(z)^{b_0}} \left[ \int_0^\infty \frac{a_0^{t-1} b_k^{c_0 t}}{\Gamma(t)^{b_0}} dt \right]^{-1} dz. \quad (4.8)$$

Note that, in practice, the above marginal distribution of  $\mathbf{y}$  can be estimated using numerical methods since it has to be only evaluated for a particular value of  $\mathbf{y}_i$ .

Finally, we define  $\mathcal{S}_3 \triangleq \mathbb{R}^d$  such that  $\theta_i | \{Z_i = 3\} \triangleq \boldsymbol{\rho}$ , where  $\boldsymbol{\rho} = [\rho_1, \dots, \rho_d]^T$  are the log-scale parameters of a log-normal likelihood function (assuming fixed shape parameters  $\{\xi_k\}_{k=1}^d$ ) such that:

$$f_\theta^{(3)}(\mathbf{y}_i) = \prod_{k=1}^d \frac{1}{y_{i,k} \sqrt{2\pi \xi_k^2}} e^{-\frac{(\ln y_{i,k} - \rho_k)^2}{2\xi_k^2}}, \quad (4.9)$$

where we let  $\theta = \boldsymbol{\rho}$ , i.e.  $y_{i,k} | \{\theta_i = \theta, Z_i = 3\} \sim \ln \mathcal{N}(\rho_k, \xi_k^2)$  and are independent.

The prior distribution of  $\boldsymbol{\rho}$  is assumed to be equal to:

$$G_0^{(3)}(\theta_i) = G_0^{(3)}(\boldsymbol{\rho}) = \prod_{k=1}^d \frac{1}{\sqrt{2\pi \xi_{0,k}^2}} e^{-\frac{(\rho_k - \rho_{0,k})^2}{2\xi_{0,k}^2}}, \quad (4.10)$$

i.e.  $\rho_k \sim \mathcal{N}(\rho_{0,k}, \xi_{0,k}^2)$ , where  $\rho_{0,k}$  and  $\xi_{0,k}$  ( $k = 1, \dots, d$ ) are the corresponding hyper-parameters. The posterior distribution of  $\theta_i$  is equal to [228]:

$$f(\theta_i | \mathbf{y}_i, Z_i = 3) = f(\boldsymbol{\rho} | \mathbf{y}_i) = \prod_{k=1}^d \frac{1}{\sqrt{2\pi \psi_k}} e^{-\frac{(\rho_k - \nu_k)^2}{2\psi_k}}, \quad (4.11)$$

i.e.  $\rho_k | y_{i,k} \sim \mathcal{N}(\nu_k, \psi_k)$ , where  $\nu_k = \frac{\xi_{0,k}^2 \rho_{0,k} + \xi_k^2 y_{i,k}}{\xi_{0,k}^2 + \xi_k^2}$  and  $\psi_k = \xi_{0,k}^2 + \xi_k^2$ . The marginal distribution of  $\mathbf{y}$  can thus be computed as:

$$f(\mathbf{y}_i | Z_i = 3) = \prod_{k=1}^d \frac{1}{2\pi y_{i,k} \sqrt{\xi_k^2 \xi_{0,k}^2}} \int_{-\infty}^\infty e^{-\frac{(\ln y_{i,k} - \rho)^2}{2\xi_k^2}} e^{-\frac{(\rho - \rho_{0,k})^2}{2\xi_{0,k}^2}} d\rho. \quad (4.12)$$

which can again be estimated numerically.

Once we have the marginal posterior distributions characterized as above, we can apply the Gibbs sampling as in Algorithm 3 to find the best observation model that fits each cluster.

### 4.2.3 Prior and Posterior Distributions for $\alpha_0$

In [229], it was shown that the posterior distribution for  $\alpha_0$  can be represented in a simple conditional form, given a certain class of prior distributions for  $\alpha_0$  [84]. In particular, if the prior distribution of  $\alpha_0$  follows the Gamma distribution, such that  $\alpha_0 \sim Ga(a, b)$  with shape  $a > 0$  and scale  $b > 0^3$ , then the conditional posterior distribution of  $\alpha_0$  may be expressed as a mixture of two Gamma distributions, where the mixing parameter follows a Beta distribution, such that:

$$\alpha_0|x, K \sim \pi_x Ga(a + K, b - \log(x)) + (1 - \pi_x) Ga(a + K - 1, b - \log(x)) \quad , \quad (4.13)$$

where  $K > 1$  is the number of clusters and  $x|\alpha_0, K \sim Beta(\alpha_0 + 1, N)$  with  $Beta$  denoting the Beta distribution [84, 229]. The mixing parameter  $\pi_x$  is defined such that:

$$\frac{\pi_x}{1 - \pi_x} = \frac{a + K - 1}{N(b - \log(x))} \quad , \quad (4.14)$$

It should be noted that  $\alpha_0$  and  $K$  should be sampled at each iteration of the Gibbs sampling and that the prior distribution of  $K$  is given by [229]:

$$P(K|\alpha_0, N) = c_N(K)N!\alpha_0^K \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + N)} \quad , \quad (4.15)$$

where  $c_N(K) = P(K|\alpha_0 = 1, N)$  can be computed using recurrence formulae for Stirling numbers [229]. Note that this prior distribution depends only on the number of data points  $N$  and on the concentration parameter  $\alpha_0$ .

---

<sup>3</sup>It is very hard to estimate  $a$  and  $b$  from real-world data. However, it is noticed in [84] that small values of  $a$  and  $b$  lead to nearly similar values of the  $\alpha$  probability density, thus resulting in a lack of variability in the distribution of  $\theta_i$ .

Moreover, for large  $N$ , the number of clusters generated by this model can be approximated as  $K = X + 1$ , where  $X$  is a Poisson random variable with mean  $\alpha_0 (\gamma + \log(N))$  and  $\gamma \approx 0.5772156649$  being the Euler constant [229]. This approximation is useful if the number of clusters  $K$  is much smaller than the number of data points  $N$ , when  $N$  is large [229]. In wireless applications, we may assume that different wireless systems form different clusters. The data points within each cluster may represent the signals corresponding to that system (cluster). If the signals are detected frequently with respect to the operation time of a certain system, a large number of feature points will be observed in a single cluster, which makes the number of feature points  $N$  to grow at a much faster rate compared to  $K$ , thus justifying the use of above approximation.

On the other hand, in order to compute the posterior distribution of  $K$ , given the observed data points, the authors in [85,229] proposed a Monte Carlo approach. This method was based on counting the number of distinct mixture components at each Gibbs iteration and updating the posterior of  $K$  accordingly. Hence, the empirical posterior probability of  $K$  can be approximated by the histogram of the number of mixture components that are encountered throughout the Gibbs sampling iterations.

### 4.3 Bayesian Prediction (Density Estimation) of the Observation Variables

Upon observing and classifying  $N$  feature points, a CR may need to predict the occurrence of a particular observation  $\mathbf{y}_{N+1}$  in the next time step. The predictive probability distribution of the random observation  $\mathbf{Y}_{N+1}$  can help to achieve this goal by using the previously observed features. Such predictive distribution can be useful in decision-making applications, allowing CR's to coordinate their actions with

other wireless users by predicting their behavior.

The posterior distribution of  $\mathbf{Y}_{N+1}$ , given the observations  $\mathbf{y}_{1:N}$  and the cluster parameters  $\theta_{1:N}$ , is denoted by  $P(\mathbf{Y}_{N+1}|\theta_{1:N}, \mathbf{y}_{1:N})$ . Since  $\{\mathbf{Y}_i\}_{i=1}^N$  are independent, given  $\theta_{1:N}$ , we have  $P(\mathbf{Y}_{N+1}|\theta_{1:N}, \mathbf{y}_{1:N}) = P(\mathbf{Y}_{N+1}|\theta_{1:N})$  which may be evaluated as  $\int P(\mathbf{Y}_{N+1}|\theta_{N+1}) dP(\theta_{N+1}|\theta_{1:N})$  [85]. According to [85], the probability distribution of  $\mathbf{Y}_{N+1}$ , given the components  $\theta_{1:N}$ , can be computed as:

$$(\mathbf{Y}_{N+1}|\theta_{1:N}) \sim \frac{\alpha_0}{\alpha_0 + N} f(\mathbf{y}_{N+1}) + \frac{1}{\alpha_0 + N} \sum_{i=1}^N f_{\theta_i}(\mathbf{y}_{N+1}) , \quad (4.16)$$

where  $f(\mathbf{y}_{N+1})$  is the marginal distribution of  $Y_{N+1}$  which was defined in (4.4). If the observation model follows the Gaussian distribution, then the marginal distribution  $f(\mathbf{y}_{N+1})$  follows the Student-t distribution such that:

$$(\mathbf{Y}_{N+1}|\theta_{1:N}) \sim \frac{\alpha_0}{\alpha_0 + N} T_s(\mathbf{m}, \mathbf{M}) + \frac{1}{\alpha_0 + N} \sum_{i=1}^N f_{\theta_i}(\mathbf{y}_{N+1}) , \quad (4.17)$$

where  $T_s(\mathbf{m}, \mathbf{M})$  is the Student- $t$  distribution whose pdf is given by:

$$f(\mathbf{x}) = |s\mathbf{M}|^{1/2} \frac{\Gamma[(1+s)/2]}{\Gamma(s/2)\Gamma(1/2)} \left[ 1 + \frac{(\mathbf{x} - \mathbf{m})^T \mathbf{M}^{-1} (\mathbf{x} - \mathbf{m})}{s} \right]^{-(1+s)/2} , \quad (4.18)$$

with  $s$  degrees of freedom, mode  $\mathbf{m}$  and scale factor  $\mathbf{M} = \frac{(1+\tau)\mathbf{S}}{s}$ .

In general, we may re-write (4.16) as:

$$(\mathbf{Y}_{N+1}|\theta_{1:N}) \sim \frac{\alpha_0}{\alpha_0 + N} f(\mathbf{y}_{N+1}) + \frac{N}{\alpha_0 + N} \sum_{k=1}^K \frac{n_k}{N} f_{\theta_k}(\mathbf{y}_{N+1}) , \quad (4.19)$$

where  $n_k$  is the number of data points in cluster  $k \in \{1, \dots, K\}$ . Note that (4.19) implies that the observation  $\mathbf{Y}_{N+1}$  is drawn from a mixture of a Student t-distribution and an observation mixture model with mixing parameters  $\frac{\alpha_0}{\alpha_0 + N}$  and  $\frac{n}{\alpha_0 + N}$ , respectively. In wireless applications, it is reasonable to assume that a detected signal may belong to a previously detected system (cluster) with a probability proportional to the number of signals observed from that system. However, since we assume that

the number of systems (clusters) is unknown, *a priori*, a signal belonging to a new system may arise with a probability proportional to  $\alpha_0$ . Thus, the probability distribution in (4.16) may be used to predict the occurrence of a certain signal, given past information.

Since past information may consist of only noisy observations  $\mathbf{y}_{1:N}$ , in the following, we show the predictive distribution of  $\mathbf{Y}_{n+1}$ , given the past observations  $\mathbf{y}_{1:N}$ . Thus, we integrate out the cluster parameters  $\theta_{1:N}$  from the posterior distribution of  $\mathbf{Y}_{N+1}$  since these parameters are not fully observable by the classifier. Hence, the Bayesian prediction, or density estimation, problem can be solved by evaluating the unconditional predictive distribution:

$$P(\mathbf{Y}_{N+1}|\mathbf{y}_{1:N}) = \int P(\mathbf{Y}_{N+1}|\theta_{1:N}) dP(\theta_{1:N}|\mathbf{y}_{1:N}) . \quad (4.20)$$

The complexity of the above expression stems from the inherent complexity of the posterior  $P(\theta_{1:N}|\mathbf{y}_{1:N})$ . However, by using the Monte Carlo approach of [85,164], it is possible to obtain an approximation for this density function, iteratively. For a given  $\mathbf{m}$  and  $\tau$  parameters, the estimated density function is given by [85]:

$$P(\mathbf{Y}_{N+1}|\mathbf{y}_{1:N}) \approx \frac{1}{N_r} \sum_{r=1}^{N_r} P(\mathbf{Y}_{N+1}|\theta_{1:N}(r)) \quad (4.21)$$

$$= \frac{1}{N_r} \sum_{r=1}^{N_r} \left[ \frac{\alpha_0(r)}{\alpha_0(r)+N} f(\mathbf{y}_{N+1}) + \frac{1}{\alpha_0(r)+N} \sum_{i=1}^N f_{\theta_i(r)}(\mathbf{y}_{N+1}) \right] \quad (4.22)$$

where  $N_r$  is the number of Gibbs sampling iterations,  $\theta_i(r)$  and  $\alpha_0(r)$  are the sampled parameters at the  $r$ -th iteration. In particular, if the observation model is assumed to have a Gaussian distribution, then (4.22) can be expressed as:

$$P(\mathbf{Y}_{N+1}|\mathbf{y}_{1:N}) \approx \frac{1}{N_r} \sum_{r=1}^{N_r} \left[ \frac{\alpha_0(r)}{\alpha_0(r)+N} T_s(\mathbf{m}, \mathbf{M}) + \frac{1}{\alpha_0(r)+N} \sum_{i=1}^N f_{\theta_i(r)}(\mathbf{y}_{N+1}) \right] \quad (4.23)$$

where  $T_s(\mathbf{m}, \mathbf{M})$  is the Student-t distribution that is defined in (4.18).

The authors in [85] have shown the convergence of the above estimate to the actual predictive distribution  $P(\mathbf{Y}_{N+1}|\mathbf{y}_{1:N})$  for almost all starting values. That is:

$$\lim_{N_r \rightarrow \infty} \frac{1}{N_r} \sum_{r=1}^{N_r} P(\mathbf{Y}_{N+1}|\theta_{1:N}(r)) = P(\mathbf{Y}_{N+1}|\mathbf{y}_{1:N}) . \quad (4.24)$$

The above identity shows that the predictive distribution of  $Y_{N+1}$  is equivalent to the average likelihood function of  $Y_{N+1}$ , averaged over the Gibbs sampling iterations.

## 4.4 Convergence of the DPMM-based Classification Algorithm

The convergence of Algorithm 3 has been proven in [85, 164] based on the MCMC approach. The convergence result can be stated as follows.

Let  $Q_I(\theta_{1:N}(0), A)$  be the probability that, with an initial value  $\theta_{1:N}(0)$  and after one iteration, Algorithm 3 produces a sample value that is contained in the measurable set  $A$ , i.e.  $Q_I(\theta_{1:N}(0), A) = P\{\theta_{1:N}(1) \in A|\theta_{1:N}(0)\}$ .  $Q_I(.,.)$  is called the transition kernel of the Markov chain.

Similarly, let  $Q_I^s(\theta_{1:N}(0), A) = P\{\theta_{1:N}(s) \in A|\theta_{1:N}(0), s\}$  and let us denote by  $P(\theta_{1:N}|\mathbf{y}_{1:N})$  the posterior distribution of  $\theta_{1:N}$ .

Theorem 1 of [85] states that, for almost all starting values of  $\theta_{1:N}(0)$ , the probability measure  $Q_I^s$  (defined over the measurable space  $\Omega \supset A$ ) converges in total variation norm to the posterior distribution as  $s$  goes to infinity. That is, for almost all  $\theta_{1:N}(0)$ ,  $\lim_{s \rightarrow \infty} \|Q_I^s(\theta_{1:N}(0), \cdot) - P(\theta_{1:N}|\mathbf{y}_{1:N})\| = 0$ . Of course, this convergence *in probability* is a weaker type of convergence, compared to the *almost sure* convergence for which  $P\{\lim_{r \rightarrow \infty} \|\theta_{1:N}(r) - \theta_{1:N}\| > \delta\} = 0$ , for some  $\delta > 0$ . In other words, Theorem 1 of [85] does not state that  $\theta_i(r) \rightarrow \theta_i$  for all  $i \in \{1, \dots, N\}$ . However, it ensures that the Gibbs sampling outcomes  $\theta_{1:N}(r)$  will be distributed according to

the actual posterior distribution of  $\theta_{1:N}|\mathbf{y}_{1:N}$ , for large  $r$ . This result is particularly important to justify the use of the Gibbs sampling outcomes in constructing the posterior distribution of  $\theta_{1:N}|\mathbf{y}_{1:N}$  and finding an estimation of  $\theta_{1:N}$ .

## 4.5 Mean-Squared Error Analysis of the Estimated Cluster Means

In this section, we derive the MSE of the estimated cluster means and, under certain regularity conditions, we establish an asymptotic upper bound on the MSE. Denote by  $\hat{\boldsymbol{\mu}}_k$  and  $\boldsymbol{\mu}_k$  to be, respectively, the estimated and actual mean vectors of cluster  $k \in \{1, \dots, K\}$ .

By assuming that the DPMM-based classifier results in correct clustering of the observation points (after sufficiently many Gibbs sampling iterations), the MSE of the estimated cluster means  $\boldsymbol{\mu}_k$  can be expressed as:

$$MSE_k = tr \left( \frac{1}{n_k} \mathbf{V}_k \right) = \frac{1}{n_k} tr (\mathbf{V}_k) , \quad (4.25)$$

where  $\mathbf{V}_k$  is the covariance matrix of the observations in cluster  $k$ , and  $n_k$  is the number of data points belonging to cluster  $k$ .

In a DPMM with  $N$  data points and with  $K$  clusters, the average MSE becomes:

$$MSE = \mathbb{E} \left\{ \frac{1}{K} \sum_{k=1}^K MSE_k | N \right\} , \quad (4.26)$$

where the prior distribution of  $K$  is as given in (4.15). For large  $N$ ,  $K$  can be approximated with a Poisson random variable such that [229]:

$$P \{K = k | \alpha_0, N\} = \frac{e^{-\alpha_0(\gamma + \log N)} [\alpha_0 (\gamma + \log N)]^k}{k!} , \text{ for } k = 0, 1, \dots \quad (4.27)$$

Thus, we have:

$$MSE = \mathbb{E} \left\{ \frac{1}{K} \sum_{k=1}^K MSE_k | N \right\} = \mathbb{E} \left\{ \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} tr(\mathbf{V}_k) | N, n_k \neq 0 \right\}. \quad (4.28)$$

Due to the complexity of the distribution of  $\frac{1}{n_k}$ , it is hard to obtain a closed form for the above MSE expression. However, if the observations are equally partitioned among the clusters (i.e.  $n_k = \frac{N}{K}$ ), we have:

$$\begin{aligned} MSE &= \mathbb{E} \left\{ \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} tr(\mathbf{V}_k) | N, n_k \neq 0 \right\} \\ &= \mathbb{E} \left\{ \frac{1}{K} \sum_{k=1}^K \frac{K}{N} tr(\mathbf{V}_k) | N \right\} \\ &\leq \frac{1}{N} \mathbb{E} \left\{ \sum_{k=1}^K V_{max} | N \right\} \\ &= \frac{1}{N} V_{max} \mathbb{E} \{ K | N \} \\ &= \frac{1}{N} V_{max} (\gamma + \log N) \mathbb{E} \{ \alpha_0 \} \\ &= \frac{ab}{N} V_{max} (\gamma + \log N) \\ &= \overline{MSE}, \end{aligned} \quad (4.29)$$

where  $V_{max} = \max_{k=1, \dots, K} tr(\mathbf{V}_k)$  and  $\alpha_0 \sim Ga(a, b)$ . Thus, under the above assumed conditions and for large  $N$ , an upper bound for MSE of the cluster mean estimates can be taken to be proportional to:

$$\overline{MSE} \propto \frac{\log N}{N}.$$

This result shows that the MSE of the cluster mean estimates decreases with  $N$ . However, the convergence of the Gibbs sampling algorithm becomes slower as  $N$  increases. Thus, a tradeoff should be made between the estimation accuracy and the convergence speed when selecting a particular data set of size  $N$  for clustering.

The above asymptotic bound is valid for large values of  $N$ , which can be justified in spectrum sensing applications when the sensing periods are very short, as in [40]. In this case, we consider a time window that includes a large number of sensing intervals as the processing period. Feature points are extracted after each sensing interval, thus leading to a large number of feature points  $N$  during this time window. These  $N$  feature points are then used in DPMM classification, justifying the use of large  $N$  in the above result. In addition, if the RF activities remain constant during the time window, feature points will be observed from the same clusters over successive sensing intervals. Then, we may assume that the total number of feature points will be equally partitioned among all the clusters.

## 4.6 A Low-complexity Biased Parameter Selection Policy for Gibbs Sampling

After running the Gibbs sampler in Algorithm 3 for a certain number of iterations, certain parameters are not likely to change if they were assigned to large clusters. In other words, if feature vector  $\mathbf{y}_i$  is assigned to a certain cluster with large number of elements, it is most likely to be re-assigned to the *same cluster* if its corresponding parameter  $\theta_i$  is sampled again using (4.1). This is a direct consequence of the underlying CRP property of the DPMM in which the probability of assigning a feature vector to a certain cluster is proportional to the number of elements in that cluster [42, 43]. Hence, frequently sampling the parameters  $\theta_i$ 's that correspond to large clusters may be unnecessary.

On the other hand, given the CRP property, a feature vector  $\mathbf{y}_i$  that is assigned to a small cluster is more likely to join a different cluster if its parameter  $\theta_i$  is re-sampled using (4.1). Thus, in order to improve the convergence rate of the Gibbs sampler,

the parameters  $\theta_i$ 's should be selected for sampling in a non-uniform way such that parameters belonging to small clusters are sampled more frequently, compared to parameters belonging to large clusters.

For this, we define a *parameter selection policy* that selects specific parameters  $\theta_i$ 's to be sampled at a particular iteration. The selection process should favor elements belonging to small clusters. Hence, the parameter selection policy is defined using  $\{w_i\}_{i=1}^N$  such that  $w_i$  is the probability of selecting parameter  $\theta_i$  to be sampled according to (4.1). We let  $w_i$  to be inversely proportional to the number of elements of the cluster containing  $\mathbf{y}_i$  such that  $\sum_{i=1}^N w_i = 1$ . In other words, if  $\theta_i = \phi_k$ , then  $w_i \propto \frac{1}{m_k}$ , where  $m_k$  is the number of elements of the cluster with parameter  $\phi_k$ <sup>4</sup>. Since, initially, each point is assigned to a different cluster, we have  $m_k = 1$  for all  $k = 1, \dots, K$ , implying that parameters  $\theta_i$ 's are selected uniformly in that case. The proposed biased selection policy is described in Algorithm 4.

After several Gibbs sampling iterations, Algorithm 4 becomes biased to selecting parameters  $\theta_i$ 's belonging to smaller clusters. Note that, other parameter selection methods can be proposed based, for example, on the Boltzman distribution which offers certain flexibility in modifying the selection policy over time. Other parameter selection policies can be proposed to achieve specific purpose algorithms, such as the sequential classification algorithm that is proposed in Section 4.7.

---

**Algorithm 4** A biased selection strategy for DPMM-based Gibbs sampling.

---

Initialize  $\theta_i = \mathbf{y}_i, \forall i \in \{1, \dots, N\}$ .

**while** Convergence condition not satisfied **do**

Select  $\theta_i$  from  $\{\theta_i\}_{i=1}^N$  with probability  $w_i$ .

Update  $\theta_i$  by sampling from the posterior distribution in (4.1).

Update the parameter selection policy  $\{w_i\}_{i=1}^N$ .

**end while**

---



---

<sup>4</sup>Recall that  $\phi_k$ 's are the unique elements of  $\{\theta_i\}_{i=1}^N$

## 4.7 A Sequential Gibbs Sampler for DPMM Classifiers

The DPMM-based Gibbs sampler that is presented in Algorithm 3 can be considered as an offline classifier since it assumes a fixed number  $N$  of feature vectors and requires large number of iterations, which makes it unfeasible for real-time operation. The biased parameter selection-based Gibbs sampler that was proposed in Section 4.6 may reduce the computational complexity of Gibbs sampling, yet it cannot be considered as a sequential algorithm since it also assumes that the number of feature vectors is fixed. In this section, however, we propose a sequential formulation for DPMM classification by assuming that the number of feature vectors increases over time.

The proposed sequential DPMM classification algorithm is defined as a recursive process that is performed in two consecutive stages:

1. The Gibbs sampler selects a newly detected feature vector  $\mathbf{y}_{N+1}$  to be classified according to the DPMM approach. The parameter  $\theta_{N+1}$  of  $\mathbf{y}_{N+1}$  is thus sampled according to (4.1). Hence, the new feature vector  $\mathbf{y}_{N+1}$  will be assigned to either an existing or a new cluster, and the parameter selection policy  $\{w_i\}_{i=1}^{N+1}$  is updated accordingly.
2. A parameter  $\theta_i$  is selected from  $\{\theta_i\}_{i=1}^{N+1}$  with a probability  $w_i$  and is sampled according to (4.1). This ensures that all parameters  $\theta_i$ 's are being continuously sampled as time progresses and new feature vectors are detected. Note that, if required, the operation in this second stage may be repeated consecutively  $J$  times in order to improve the classification results.

The sequential DPMM-based classification algorithm can be characterized as in Algorithm 5. Note that, for efficient implementation of this algorithm, an offline

training step is introduced before running the sequential classification method in order to initialize the clusters and improve the real-time performance.

---

**Algorithm 5** A sequential DPMM-based Gibbs sampling.

---

**Training:** Apply offline DPMM classification to  $\mathbf{y}_{1:N}$ .

**Sequential classification:**

**for**  $k = N + 1, \dots$  **do**

    Sample  $\theta_k$  using the posterior distribution in (4.1).

    Update the parameter selection policy  $\{w_i\}_{i=1}^k$ .

**for**  $j = 1, \dots, J$  **do**

        Select  $\theta_i$  from  $\{\theta_i\}_{i=1}^k$  with probability  $w_i$ .

        Update  $\theta_i$  by sampling from the posterior distribution in (4.1).

        Update the parameter selection policy  $\{w_i\}_{i=1}^k$ .

**end for**

**end for**

---

## 4.8 Simulation Example: Signal Classification in the ISM Band

In this section, we apply above developed non-parametric signal classification algorithm based on DPMM to the problem of RF mapping. In particular, to start with, we consider 2 IEEE 802.11.b WiFi signals (channels 2 and 13) transmitting at 2.417 and 2.472GHz, respectively. We also consider a Bluetooth signal transmitting at 2.45 GHz during the sensing process. The SNR at the receiver is 5 dB and each sensing window is  $30\mu s$ . We assume a fast-fading Rayleigh channel with normalized fading coefficients  $h$  such that  $\mathbb{E}\{h^2\} = 1$ .

After each  $30\mu s$  sensing time, feature points  $(f_c, \alpha, B)$  are extracted from the

sensed signal, where  $f_c$  denotes the carrier frequency (down-converted to zero-IF),  $\alpha$  is the cyclic frequency component corresponding to the symbol rate and  $B$  is the estimated signal bandwidth. The carrier frequencies  $f_c$  and cyclic frequencies  $\alpha$  are obtained by applying the energy and cyclostationary detection algorithms in [41] and the signal bandwidth is estimated from the smoothed PSD of the received signal. In this setup, each WiFi signal has a bandwidth of 22 MHz and the Bluetooth signal has a bandwidth of 1 MHz. Furthermore, the Bluetooth signal has a symbol rate of 1 Mbaud and the WiFi has a chiprate of 11 Mchips/s that is manifested in the  $\alpha$  component of the feature points.

We perform 50 repetitions of the sensing process (over a total sensing time of  $50 \times 30 \mu s$ ) and obtain the feature points. We then apply our proposed DPMM-based feature classification algorithm to classify the observed feature points. The feature points that are marked with the same marker shape in Fig. 4.1 are assigned to the same cluster. We show in Fig. 4.1 the results of the DPMM classification in a 3D feature space where the two WiFi signals are estimated to have Gaussian observation models while the Bluetooth signal is assigned a log-normal model. The classification accuracy, denoting the percentage of feature points classified into correct clusters, is estimated as 100% in this setup.

In the next set of simulations, we compare performance of the proposed DPMM-based classification algorithm to that of the approach proposed in [79] based on the K-means and X-means algorithms [80]. In the simulation setup, we consider an additional 4-QAM digital signal transmitting at 2440 MHz. For simplicity, we limit the feature vectors to be 2-D data  $(f_c, B)$ . Figures 4.2 and 4.3 show the performance of the K-means algorithm for  $K = 4$  with different initial cluster centroids. Figure 4.2 shows that the data are clustered perfectly for  $K = 4$ . However, in some cases, the QAM and Bluetooth signals are merged into a single cluster, whereas each of the WiFi clusters is split into multiple clusters, as shown in Fig. 4.3. This is because

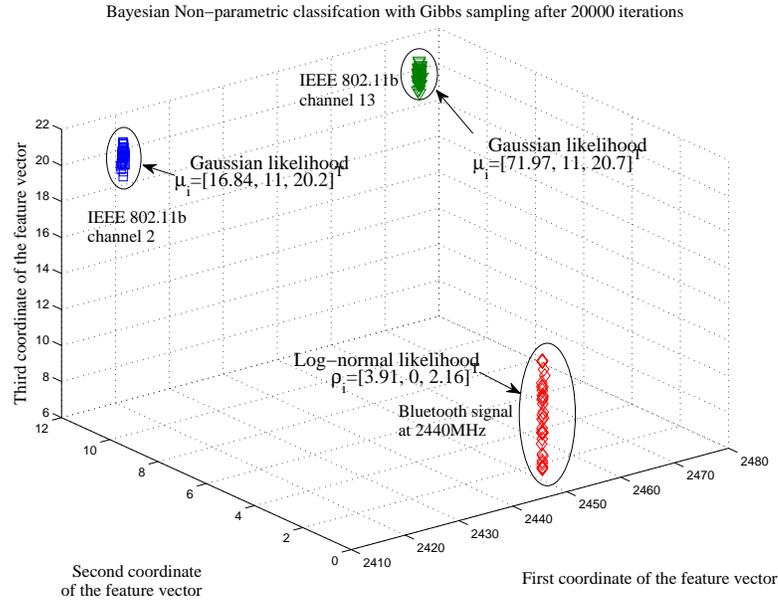


Figure 4.1: Signal Classification of 2 WiFi and a Bluetooth signal. The feature point is denoted by  $(f_c, \alpha, B)$ , where  $f_c$  is the carrier frequency,  $\alpha$  is the cyclic frequency component corresponding to the symbol rate and  $B$  is the estimated bandwidth. Energy detection is applied for  $30\mu s$  at an SNR of 5 dB with Rayleigh fading (fast fading). The probability of correct classification is 100% after 20000 Gibbs sampling iterations.

the K-means algorithm depends on the choice of initial centroid locations and it converges to a local minimum of the distortion measure [79, 80]. Hence, the initial centroid in Fig. 4.2 has lead to a perfect classification result. However, for a different initial centroid, as in Fig. 4.3, the K-means lead to a poor classification performance. Furthermore, the K-means algorithm assumes that the observation noise is circular Gaussian with a covariance matrix that is a scaled version of the identity matrix. Thus, the performance of K-means-based clustering deteriorates in the presence of observation models with arbitrary noise characteristics.

Next, we apply the X-means and the proposed DPMM-based algorithms to the same data, and we show the results in Figs. 4.4 and 4.5, respectively. The X-means

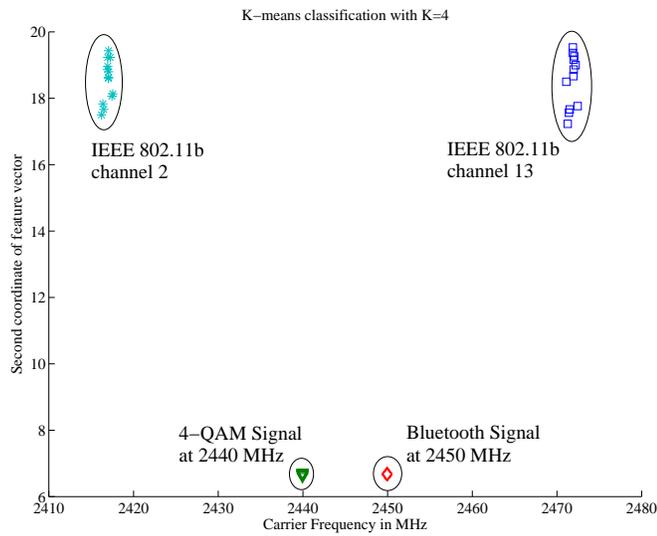


Figure 4.2: K-means classification with  $K = 4$  gives a classification accuracy of 100% with arbitrary initialization of centroid locations.

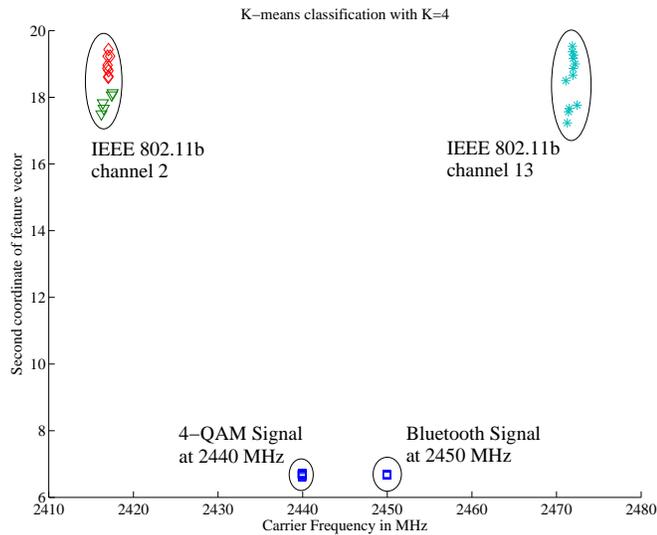


Figure 4.3: K-means classification with  $K = 4$  gives a classification accuracy of 79.41% with a different initialization of centroid locations.

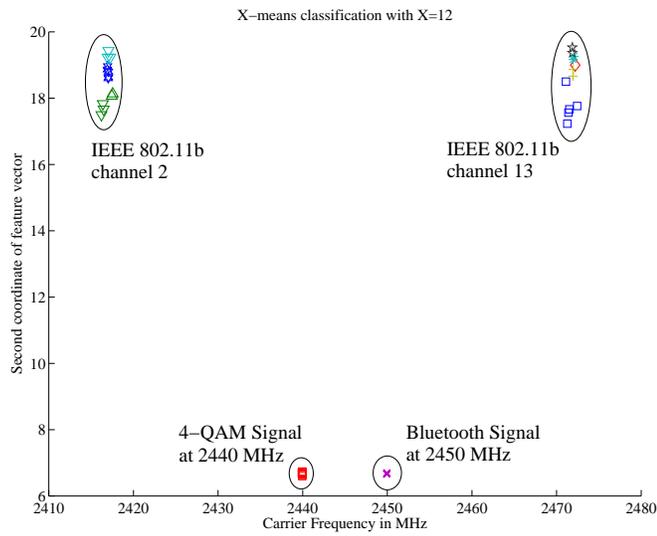


Figure 4.4: X-means classification with estimated  $X = 12$  gives a classification accuracy of 55.88%.

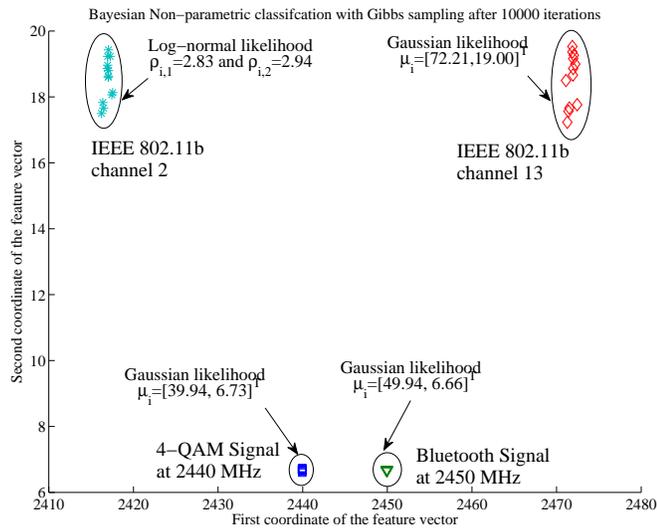


Figure 4.5: DPMM-based classification with estimated  $K = 4$  gives a classification accuracy of 100%

algorithm suffers from the same drawbacks of the K-means by not being able to match the distributions of the observed data. In this simulation, the feature points of WiFi channels were split into multiple clusters. On the other hand, the results of the proposed DPMM-based classification in Fig. 4.5 show perfect clustering due to the ability of the DPMM to estimate the noise characteristics and to infer the number of clusters from the observed data. In this case, the classification accuracy was estimated to be 100% with the DPMM classifier, whereas X-means achieved a classification accuracy of only 55.88%. On the other hand, the K-means algorithm achieved 100% accuracy if  $K = 4$  was given as prior information, as shown in Fig. 4.2, but the performance still dropped to 79.41% depending on the initial choice of centroid locations, as in Fig. 4.3. Clearly, the advantage of the proposed DPMM-based method is that it does not require *neither the number of clusters  $K = 4$  nor their centroids as prior information*.

In Fig. 4.6, we plot the predictive probability distribution of future feature points. For simplicity of representation, we again consider a 2D feature space with feature points  $(f_c, B)$  and represent the pdf of the predictive distribution in contour lines. The result shows four main clusters corresponding to the WiFi, Bluetooth and QAM signals where the feature points corresponding to channel 2 of the WiFi system is estimated to have a log-normal distribution while the other feature points are estimated to have Gaussian distributions. The obtained distribution forms an RF mapping of the RF environment and can help CR's to adapt their actions by using this information (beyond the scope of this work).

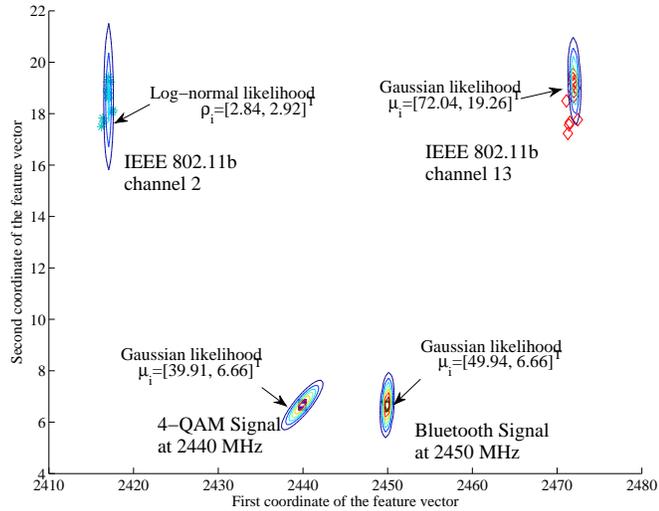


Figure 4.6: Signal Classification of 2 WiFi and a Bluetooth signal. The feature point is denoted by  $(f_c, B)$ , where  $f_c$  is the carrier frequency and  $B$  is the estimated bandwidth of the signal. Energy detection is applied for  $30\mu s$  at an SNR of 5 dB with Rayleigh fading (fast fading). The probability of correct classification is 100% after 5000 Gibbs sampling iterations.

### 4.8.1 Simulation Example: Performance of the Simplified and Sequential DPMM Classifiers

In this section, we demonstrate the efficiency of the proposed simplified Gibbs sampler (Algorithm 4) in improving the convergence rate of the DPMM classifier. Thus, we consider four different QAM signals (BPSK, 4-QAM, 16-QAM and 64-QAM) that are transmitted simultaneously at baseband-equivalent center frequencies  $-80, -40, 0$  and  $50$  MHz and with respective symbol rates 2, 2, 5 and 5 MBauds. We perform spectrum sensing for  $T = 50\mu s$  in each sensing interval at a sampling rate  $f_s = 200$  MHz and with smoothing window length  $L = 151$ . We extract  $N = 100$  feature vectors (whose components are the center frequencies, cyclostationary feature and kurtosis feature) after 25 sensing intervals. We assume a multi-path Rayleigh fading

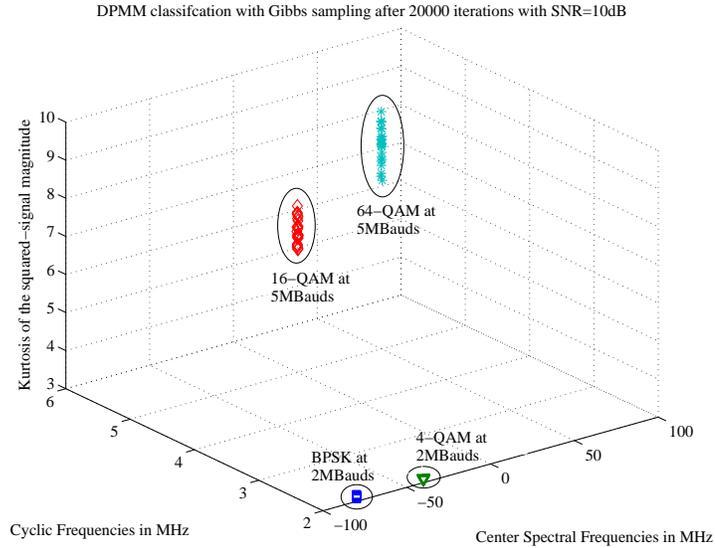


Figure 4.7: DPMM classification of four different QAM signals at a received  $SNR = 10dB$  using Algorithm 4. Feature vectors assigned to the same cluster are represented with identical markers. Classification accuracy is equal to 100%.

channel with unity paths gains and we assume a received SNR of 10 dB.

We apply DPMM-based classification using three different parameter selection policies for Gibbs sampling: Random parameter selection, Round-Robin parameter selection and biased parameter selection (i.e. simplified Gibbs sampler), respectively. The hyper-parameters of the DPMM are selected such that:  $\kappa_1 = \kappa_2 = 0.5$ ,  $\mathbf{m} = [0, 0, 0]^T$ ,  $\tau = 1$ ,  $s = 5$ ,  $\mathbf{S} = \text{diag}(0.1, 0.1, 1)$ ,  $\boldsymbol{\rho}_0 = [\log(50), \log(2), \log(5)]^T$ ,  $\boldsymbol{\xi}_0 = [1, 1, 1]^T$  and  $\alpha_0 = 1$ . After 20000 Gibbs sampling iterations, all of the above parameter selection policies lead to perfect classification results, as shown in Fig. 4.7 where we represent features assigned to the same cluster using identical markers. By computing the number of clusters  $K$  at each iteration step, we observe that each of the parameter selection policies converges to the optimal number of clusters (i.e.  $K = 4$ ) after different numbers of iterations, as shown in Fig. 4.8. The simplified Gibbs sampler with random biased parameter selection policy converges after

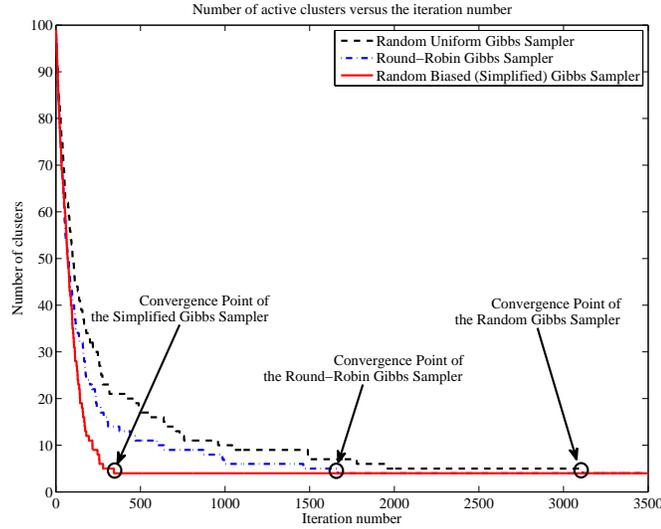


Figure 4.8: Convergence of the DPMM classifier using random uniform, Round-Robin and random biased parameter selection policies for Gibbs sampling.

343 iterations, compared to 1665 and 3113 iterations for the Round-Robin and random uniform selection policies, respectively. This shows that the proposed simplified Gibbs sampler can improve the convergence rate of the traditional Gibbs sampler by 5 times, in this case. This result can be interpreted by observing that the number of clusters stabilizes for longer durations in both random and Round-Robin selection policies before updating the proper feature vector parameters  $\theta_i$ 's to lead to convergence. On the other hand, by biasing the selection of parameters  $\theta_i$ 's, the proposed simplified Gibbs sampler selects more efficiently the parameters  $\theta_i$ 's to be updated according to (4.1), which leads to faster convergence.

The above simplified Gibbs sampler-based DPMM classifier is also applied during the training stage of the sequential DPMM classifier (Algorithm 5) to initialize the clusters' parameters. In the training stage, we consider the same observation model as above. However, during real-time operation, we assume that the signal transmitting at  $-80\text{MHz}$  changes its symbol rate to 5MBauds and its modulation scheme to 16-

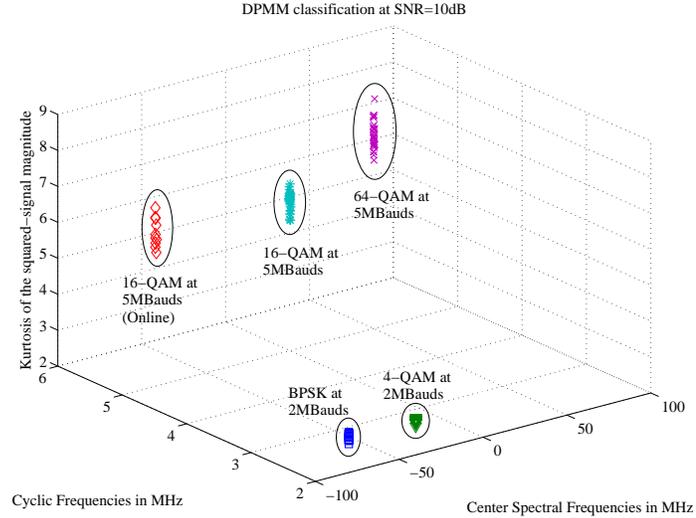


Figure 4.9: Sequential DPMM classification of different QAM signals at a received  $SNR = 10dB$  using Algorithm 5. Classification accuracy is equal to 100%.

QAM, while the other signals' parameters remain unchanged. We run the spectrum sensing and feature extraction algorithm for 10 sensing intervals while extracting the signals features at each sensing interval and classifying the extracted features using the sequential DPMM algorithm. Note that, in contrast with the simplified algorithm, the sequential algorithm classifies the detected features at the end of each sensing interval. We show, in Fig. 4.9, the resulting feature space at the end of the 10 sensing intervals, including feature vectors that are observed during both training and real-time stages. These results show that a new cluster has been created to include feature points corresponding to the 16-QAM signal that is transmitting at  $-80MHz$ . However, the other signals still correctly join the old clusters, resulting in perfect overall classification performance.

It is also worth comparing the proposed sequential DPMM classifier to a non-parametric unsupervised classifier referred to as the X-means algorithm [79] to identifying under what conditions and in what scenarios the proposed algorithm can be

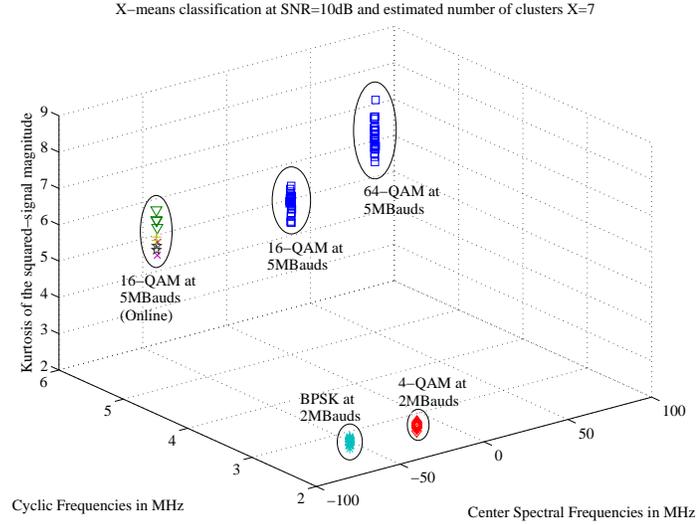


Figure 4.10: X-means classification of different QAM signals at a received  $SNR = 10dB$ . Classification accuracy is equal to 71%.

most advantageous. By applying the X-means algorithm to the same set of data assumed in Fig. 4.9, we observe, in Fig. 4.10, that many feature vectors have not been classified correctly, which resulted in a classification accuracy of only 71%. The primary reason of the poor performance is that the X-means algorithm was not able to estimate the number of clusters in this case. Indeed, it was estimated to be 11, instead of 5. This is due to the underlying spherical Gaussian observation model assumed by the X-means algorithm, which is not a good representation of our feature vectors distribution [42].

Note that, in detecting and classifying signals in the presence of noise, there can be two types of errors: 1) Feature extraction errors and 2) signal classification errors. In our problem, the feature extraction errors are due to the generation of feature vectors not corresponding to existing signals, which is equivalent to false alarm errors in signal detection literature [230]. The feature extraction errors depend mainly on the design and characteristics of the feature extraction algorithm. On the other hand,

the classification error is defined as the misclassification rate of the detected feature vectors, which mainly depends on the classifier design. Hence, we may define the following types of errors to be considered for further performance evaluation:

1. Feature extraction error: Defined as the proportion of feature vectors that do not correspond to existing signals.
2. Misclassification rate: Defined as the proportion of feature vectors that are not classified into correct clusters. In this case, correct classification implies that feature vectors corresponding to different signals are classified into different clusters and erroneous features (due to false alarm) are classified into separate clusters.
3. Overall feature extraction and signal classification error: Defined as the proportion of feature vectors that do not correspond to existing signals and/or are misclassified. This is the complementary rate of the overall classification accuracy.

We first compute the misclassification rates of the DPMM during both training and sequential classification stages (denoted as "DPMM Training" and "DPMM Sequential" in Fig. 4.11). We also compute the overall misclassification rate of DPMM over the whole training and sequential stages (denoted as "Overall DPMM" in Fig. 4.11). The training stage is performed over 25 sensing intervals, followed by 25 sensing intervals for sequential classification. During the whole time horizon, we consider a similar setup of QAM signals as described above. The misclassification rates of both training and sequential stages are shown in Fig. 4.11. At low SNR values ( $-10\text{dB}$  and  $-5\text{dB}$ ), the sequential misclassification rate is higher than the training misclassification rate. However, the misclassification rate drops to 0% for large SNR values, resulting in perfect overall DPMM classification.

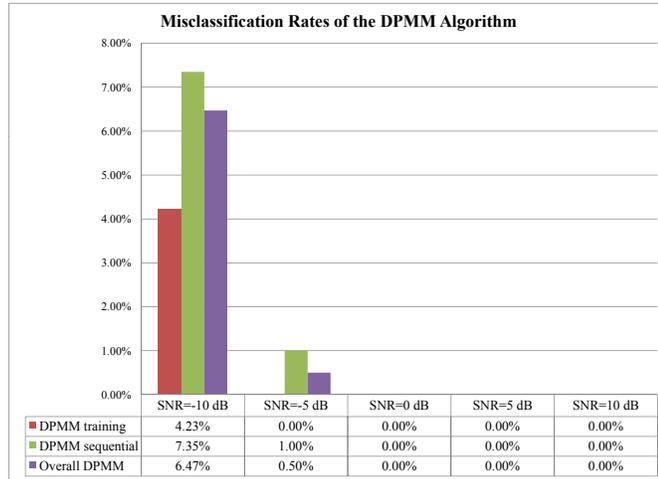


Figure 4.11: Misclassification rates of the DPMM algorithm for different SNR levels. Misclassification rates of both training and sequential stages are obtained over 25 sensing intervals for each stage (denoted as "DPMM Training" and "DPMM Sequential", respectively). The overall misclassification rate of the DPMM, including both training and sequential stages, is denoted as "Overall DPMM". Four different QAM signals are being transmitted simultaneously and are sampled at a rate  $f_s = 200$  MHz for a duration  $T = 50\mu s$  in each sensing interval.

Next, we compute the feature extraction error (denoted as "Feature Extraction" in Fig. 4.12) as well as both DPMM and X-means misclassification rates (denoted as "Overall DPMM" and "X-means", respectively). Combined feature extraction errors and signal classification errors are also computed for each of the DPMM and X-means classifiers, and are denoted as "Overall DPMM + Feature Extraction" and "X-means + Feature Extraction" in Fig. 4.12, respectively. The results show that the feature extraction error drops as the SNR increases. In the DPMM classification case, Fig. 4.12 shows that the overall classification accuracy is degraded mostly due to the high level of feature extraction errors at low SNR. However, perfect feature extraction and DPMM classification is obtained at high SNR. In other words, as long as the feature extraction errors are small, we may expect the performance of

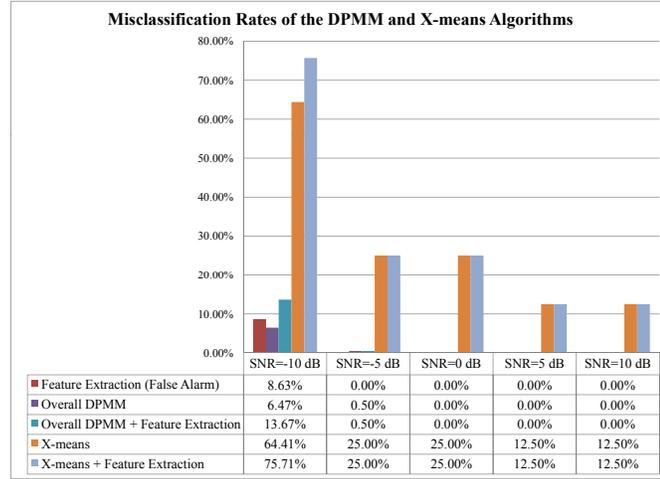


Figure 4.12: Misclassification rates of the DPMM and X-means algorithms for different SNR levels. Overall DPMM misclassification rates are obtained over 50 sensing intervals, including both training and sequential stages. Misclassification rates of the X-means are also obtained over the whole 50 sensing intervals. Combined feature extraction errors and signal classification errors are computed using DPMM and X-means classifiers, and are denoted as "Overall DPMM + Feature Extraction" and "X-means + Feature Extraction", respectively. Four different QAM signals are being transmitted simultaneously and are sampled at a rate  $f_s = 200$  MHz for a duration  $T = 50\mu s$  in each sensing interval.

the DPMM classification to be very good. On the other hand, the X-means shows poor classification performance at low SNR. However, in contrast to the DPMM, the X-means does not lead to perfect classification even when the SNR is high. This is because the X-means classification algorithm leads to a significant amount of its own errors when the features are not spherical Gaussian distributed as presumed.

## 4.9 Conclusion

In this chapter, we proposed a Bayesian non-parametric signal classification method to identify/classify active wireless systems in an unknown RF environment. This proposed technique is suitable for autonomous CR's, such as Radiobots of [1] and [60], in performing spectrum sensing and signal classification in alien RF bands. Since our non-parametric technique does not require any prior knowledge of the existing signals in the sensed spectrum, it can ensure autonomous operation of CR's. The proposed DPMM framework extends to both Gaussian and non-Gaussian observation models and it uses the Gibbs sampling to estimate the appropriate distribution for each cluster. We derived an upper bound for the MSE of the estimate of the cluster means as a function of the number of feature points  $N$ . A Bayesian predictive distribution was also derived to construct an RF mapping for the on-going RF activity. A sequential Gibbs sampler was also proposed to improve the computational efficiency of the DPMM-based classification algorithm. Simulation results were presented to compare the performance of the proposed DPMM-based algorithm to those of existing classifiers such as K-means and X-means. These example results show that the new DPMM-based non-parametric classification algorithm, and its sequential version, can be highly effective in unknown RF environments compared to comparable algorithms such as X-means.

# Chapter 5

## Distributed Reinforcement Learning for CRN's

### 5.1 Introduction

Opportunistic spectrum access [231] has been envisioned as a promising technique to exploit the spectrum vacancies, which permits unlicensed secondary users to access the primary channels opportunistically when the primary users who own the spectrum rights are not transmitting. Cognitive radio devices provide a platform to realize such OSA techniques. In general, CR's are assumed to be able to sense and adapt to their RF environment.

In this chapter, we consider a decentralized CRN in which each secondary user tries to obtain, independently, the best estimate of the status of the primary channels based on its own local information. In particular, when the primary channel states follow a Markovian evolution, a cognitive user can utilize its history of observations and actions in order to derive a better sensing/accessing policy. This problem can then be formulated as a DEC-POMDP and has been discussed in several recent stud-

ies. For example, in [152], the authors suggested a MAC protocol for decentralized ad-hoc CRN's by modeling the system as a POMDP that is equivalent to a MDP with an infinite number of states. The corresponding optimal sensing policy that maximizes the total discounted return was shown to be *computationally prohibitive*. Thus, an optimal myopic policy was derived such that it maximizes the instantaneous rewards. The myopic policy that was formulated in [152] is optimal for a single-user setup, and is suboptimal when applied to a multiuser setting because it would lead to collisions between secondary users when more than one user try to access the same channel. On the other hand, in [232] the authors proposed three different sensing policies for multiuser OSA: The first policy is based on a cooperative protocol in which secondary users exchange their beliefs about the channel states at each time slot. The second policy applies learning techniques to obtain an estimate of the other users' beliefs, and the third policy is based on a single-user approach in which the cognitive users act non-cooperatively. We note that [232] assumes perfect sensing of the primary channels, which we do not assume throughout this work.

In [38], a suboptimal sensing/access policy was derived for *cooperative* cognitive networks since it is not easy to solve the Bellman equation that corresponds to the formulated POMDP model. However, the assumed model did not ensure full utilization of spectrum resources because only one primary channel was accessed at each time instant collectively by all secondary users. This leads to low network throughput since all the secondary users are assumed to sense the same primary channel at a time. The main advantage of this model, however, was that it achieves better sensing performance. The trade-off between the sensing accuracy and the secondary throughput has been discussed recently in [233].

We believe that the solution to these issues is to make the so-called CR's indeed cognitive, i.e. to achieve smart performance, the CR's should have the ability to learn from their observed environment and the past actions. Indeed, it can be argued that

learning from experience must be at the heart of any cognitive system. Recently, this view is gaining importance within the CR research community as is evident by the application of learning techniques to CR's, as we discussed in Chapter 3. For example, the multi-agent RL algorithm, known as Q-Learning, was applied in [99] to achieve interference control in decentralized wireless regional area networks (WRAN's). In [234], the authors developed a Q-learning algorithm for an auction-based DSA protocol, which is different from the DEC-POMDP structure of our proposed model. To the best of our knowledge, none of the CR studies that assume an underlying POMDP structure has used the Q-learning algorithm to solve the OSA problem [152, 232]. The literature on learning techniques to achieve CR goals is still at an infancy, although there is a rich literature on machine learning in computer science and classical statistical learning that provides a great starting point [95].

In this chapter, we formulate the channel sensing in decentralized cognitive networks as a DEC-POMDP problem. Unlike [152], our approach considers a multi-user setting and we propose a channel sensing policy that takes into account the collisions among secondary users. Our proposed *sensing* policy is based on the distributed RL. Note that, we use the RL to derive the sensing policy rather than to obtain interference control as in [99]. This algorithm achieves two main goals: Deriving a sensing policy based on the history of actions and observations, and minimizing the collisions between secondary users while competing for channel access opportunities. On the other hand, we propose a channel *access* mechanism that limits the collisions between primary and secondary users when secondary users have noisy observations about the primary channels. Our channel access scheme ensures high accuracy and robustness in controlling the collision probability with primary channels, thus guaranteeing the QoS requirements of primary users.

The remainder of this chapter is organized as follows: Section 5.2 defines the system model. In sections 5.3 and 5.4, we derive both the accessing and sensing policies for cognitive users. We show the simulation results in section 5.5. Section

5.6 concludes the paper.

## 5.2 System Model

We consider a wireless network having a set of primary channels  $\mathcal{C} = \{1, \dots, L\}$ . The channels' occupancy states are assumed to be independent and following a Markovian evolution. A set of distributed users form a secondary network that is assumed to rely on cognitive techniques to access these primary channels when they are idle. The set of secondary users in the system is denoted by  $\mathcal{K}_s = \{1, \dots, K_s\}$ . The secondary network forms a multiple access channel in which each secondary user independently searches for a spectrum opportunity in order to communicate with a secondary base station, as depicted in Fig. 5.1. Every secondary user  $j \in \mathcal{K}_s$  is assumed to be able to sense only one primary channel at a time, and we assume that secondary users do not cooperate. This is a reasonable assumption in decentralized networks in which there is no control channels for ensuring collaboration among secondary users.

We identify the overall system made of primary channels and the  $K_s$ -secondary users as a DEC-POMDP [103] by defining the state of the system as:

$$\mathbf{s}(k) = (s_1(k), \dots, s_L(k)) \in \mathcal{S}, \quad (5.1)$$

where  $s_i(k) \in \{0, 1\}$  represents the state of channel  $i \in \mathcal{C}$  as being idle (0) or busy (1) in time slot  $k$ , and  $\mathcal{S}$  is the set of all possible states  $\mathbf{s}(k)$ . We define  $\mathbf{a} \triangleq (a_1, \dots, a_{K_s})$  as the joint action of all secondary users (agents) and  $P(s, \mathbf{a}, s')$  to be the probability of transition from state  $s$  to  $s'$  when taking the joint action  $\mathbf{a}$ . The transitions of every channel's state are independent of the other states and these transitions are assumed to follow a Markovian evolution as mentioned above. The state transition matrix  $\mathbb{P}$  of the state vector  $\mathbf{s}(k)$  is therefore  $\mathbb{P} = \mathbb{P}_1 \otimes \dots \otimes \mathbb{P}_L$ , where  $\mathbb{P}_i$  is the state transition matrix of channel  $i$ , and  $\otimes$  denotes the Kronecker product.

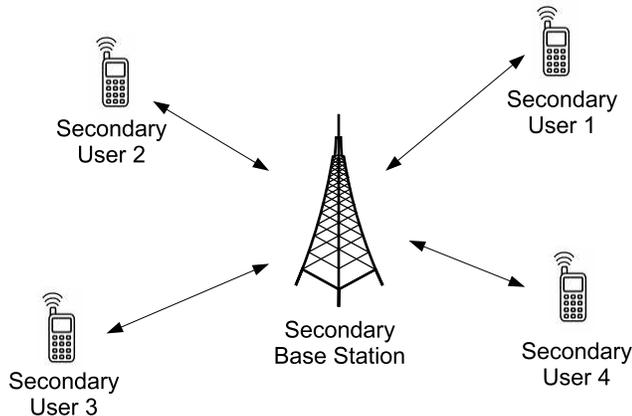


Figure 5.1: Cognitive radio network with distributed secondary nodes

Note that, the transition probabilities  $P(s, \mathbf{a}, s')$  (for  $(s, s') \in \mathcal{S}^2$ ) are independent of the secondary user actions since they are determined by the evolution of the primary channels states, i.e.  $P(s, \mathbf{a}, s') = P(s, s')$ , where  $P(s, s')$  is obtained from the state transition matrix  $\mathbb{P}$ . Similarly, for an individual channel  $i \in \mathcal{C}$ , the transition probabilities  $P_i(l, l')$  (for  $(l, l') \in \{0, 1\}^2$ ) are obtained from  $\mathbb{P}_i$ .

The action of secondary user  $j \in \mathcal{K}_s$  at time  $k$  is denoted by  $a_j(k) \in \mathcal{C}$  which represents the index of the primary channel that user  $j \in \mathcal{K}_s$  should sense during time slot  $k$ . We define  $Y_i(k, j)$  to be the observation of secondary  $j \in \mathcal{K}_s$  on channel  $i \in \mathcal{C}$  in time slot  $k$  which is assumed to be the output of a Binary Symmetric Channel (BSC) where  $\Pr\{Y_i(k, j) \neq s_i(k)\} = \nu_i$  is the crossover probability. As a result,  $Y_i(k, j)$  is a discrete random variable with distinct pmf  $f_0$  and  $f_1$  when  $s_i(k) = 0$  and  $s_i(k) = 1$ , respectively.

Let  $\mathbf{Y}_i^k(j)$  denotes the vector of observations up to time slot  $k$  obtained by secondary  $j \in \mathcal{K}_s$  on channel  $i \in \mathcal{C}$ . Let  $\mathbf{K}_i^k(j)$  denote the time slot indices up to slot  $k$  when channel  $i$  was sensed by secondary user  $j$ . Also, let  $\mathbf{Y}^k(j) = \{\mathbf{Y}_i^k(j) : i \in \mathcal{C}\}$

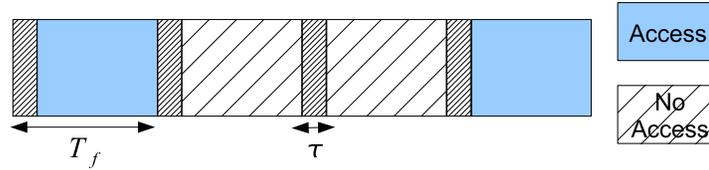


Figure 5.2: Channel Access Policies

be the collection of observations up to slot  $k$  on all primary channels obtained by the  $j$ -th secondary user.

### 5.3 Channel Access Mechanism

The sensing and access operations of the secondary users are scheduled as is shown in Fig. 5.2, where we consider that a secondary user senses a primary channel during the sensing period  $\tau$ . Primary users are assumed to always start their transmission at the beginning of a frame of duration  $T_f$  so that a primary channel will remain free during the secondary access duration if it was free during the corresponding sensing period.

A cognitive device that has sensed a channel can access that channel during the remaining frame duration of  $T_f - \tau$ . In order to avoid collisions among secondary users, we assume that each secondary user generates a random *backoff time* before transmitting [152]. If more than one secondary users decide to access the same channel, the channel access will be granted to the secondary user that has the smallest *backoff time*.

After sensing channel  $i = a_j(k)$ , secondary user  $j \in \mathcal{K}_s$  decides whether to access channel  $i$  based on its observation sequence  $\mathbf{y}_i^k(j) \triangleq \{y_i(k', j) : k' \in \mathbf{K}_i^k\}$  where  $y_i(k', j)$  is a realization of  $Y_i(k', j)$ . In order to achieve a probability of collision be-

low a certain bound, we may apply an NP type detector [230]. An optimal access decision for the  $j$ -th secondary user would choose one of the two possible hypothesis  $H_1 = \{s_i(k) = 0\}$  or  $H_0 = \{s_i(k) = 1\}$  in time slot  $k$  based on the whole observation sequence  $\mathbf{y}_i^k(j)$ . However, implementing such an optimal detector becomes too complicated due to the need for computing the distribution of the likelihood ratio of  $\mathbf{Y}_i^k(j)$  which is a random sequence whose length increases linearly with time. Hence, we simplify the detection rule by assuming that the decision to access a channel in time slot  $k$  is based only on the current observation.

Let  $\alpha$  be the false alarm probability such that  $\alpha \leq 0.5$ . The optimal NP detector then is as randomized access decision rule  $\tilde{\delta}_i(k, j)$  for secondary  $j$  to access channel  $i$  at time  $k$ . This access decision can be viewed as a Bernoulli random variable denoted by  $\delta_i(k, j)$  whose parameter  $\tilde{\delta}_i(k, j)$  is given by:

$$\tilde{\delta}_i(k, j) = \begin{cases} \frac{\alpha}{\nu_i} \mathcal{I}_{\{y_i(k,j)=0\}} I_{i,j}^{(k)} & \text{if } \alpha < \nu_i \\ \left( \mathcal{I}_{\{y_i(k,j)=0\}} + \frac{\alpha - \nu_i}{1 - \nu_i} \mathcal{I}_{\{y_i(k,j)=1\}} \right) I_{i,j}^{(k)} & \text{if } \alpha \geq \nu_i \end{cases}$$

where  $I_{i,j}^{(k)} = \mathcal{I}_{\{a_j(k)=i\}}$ , and  $\mathcal{I}_B = 1$  if condition  $B$  is satisfied, and 0 otherwise. Therefore, secondary user  $j$  decides to access a sensed channel  $i$  in time slot  $k$  only if  $\delta_i(k, j) = 1$ , which happens with probability  $\tilde{\delta}_i(k, j)$ .

It can be observed that the collision probability on a particular channel can go beyond the desired threshold because the accessing rule in a decentralized network follows an OR-rule. For that reason, we will design a channel access mechanism that guarantees a certain collision probability with the primary channels.

We define  $E_{j,i}(k)$  to be the event that secondary user  $j \in \mathcal{K}_s$  decides to access channel  $i \in \mathcal{C}$  at time  $k$ , given that secondary user  $j$  has sensed channel  $i$  at time  $k$ . Also, we let  $E_i(k)$  to be the event that channel  $i \in \mathcal{C}$  is busy at time  $k$ . When several secondary users sense and try to access the same primary channel  $i \in \mathcal{C}$ , we define the resulting collision probability as  $P_c(i) = \Pr \left\{ \bigcup_{j \in \mathcal{Z}_i(k)} E_{j,i}(k) | E_i(k) \right\}$ , where  $\mathcal{Z}_i(k)$

is the set of secondary users that sense channel  $i$  in time slot  $k$ .

Note that the events  $\{E_{j,i}(k)|E_i(k) : j \in \mathcal{K}_s\}$  are independent because each secondary user makes its access decision independently of the other users, after having sensed the channel  $i$ . As a result, the collision probability on channel  $i$  can be expressed as  $P_c(i) = 1 - (1 - \alpha)^{Z_i(k)}$ , where  $Z_i(k) = |\mathcal{Z}_i(k)|$  and  $\alpha = \Pr\{E_{j,i}(k)|E_i(k)\}$  is the false alarm probability of each secondary detector that results from claiming  $H_1 = \{s_i(k) = 0\}$  (or equivalently  $\{\delta_i(k, j) = 1\}$ ) when  $H_0 = \{s_i(k) = 1\}$  is true. Therefore, in order to ensure an overall collision probability  $P_c(i) = \alpha_0$  in channel  $i$ , each secondary user  $j \in \mathcal{Z}_i(k)$  should set its false alarm probability to  $\alpha = 1 - (1 - \alpha_0)^{1/Z_i(k)}$ .

Since each secondary user does not know the total number of users  $Z_i(k)$  that are sensing primary channel  $i \in \mathcal{C}$  at a particular time  $k$ , it uses the expected value of  $Z_i(k)$  to compute its false alarm probability such that  $\alpha = 1 - (1 - \alpha_0)^{1/\mathbb{E}\{Z_i(k)\}}$ . We will compute this expected value in the followings and show, through simulations, that the proposed access technique can guarantee an upper bound on the collision between primary and secondary users.

## 5.4 Sensing Policies of Distributed Secondary Users

We define the belief vector of channel  $i \in \mathcal{C}$  as  $\mathbf{p}(k, j, i) = [p_0(k, j, i), p_1(k, j, i)]$  where  $p_l(k, j, i) = \Pr\{s_i(k) = l | \mathbf{Y}_i^{k-1}(j)\}$  which represents the probability of  $s_i(k)$  being in state  $l \in \{0, 1\}$  in time slot  $k$ , given the past observations  $\mathbf{Y}_i^{k-1}(j)$ . Let  $\mathbf{b}_j(k) = [b_j(1, k), \dots, b_j(2^L, k)]$  be the belief vector of the primary system according to secondary user  $j$ , where

$$b_j(u(\mathbf{s}(k)), k) = \prod_{i=1}^L p_{s_i(k)}(k, j, i), \quad (5.2)$$

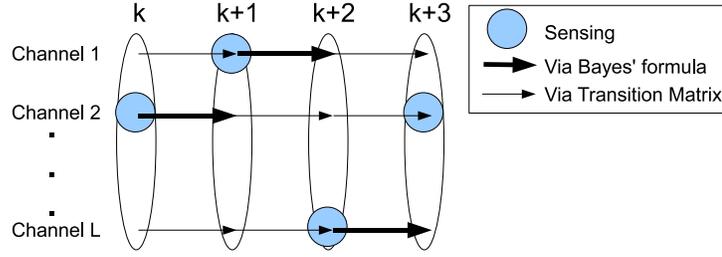


Figure 5.3: Sensing and Updating the Beliefs

given that  $u(\mathbf{s}) \in \mathcal{U} = \{1, \dots, 2^L\}$  is the index of state  $\mathbf{s}(k) = (s_1(k), \dots, s_L(k))$ . The belief vector  $\mathbf{b}_j(k)$  is a sufficient statistic for an optimal OSA protocol in a single-user setup [152]. However, in our case, we consider a distributed multi-user scenario and  $\mathbf{b}_j(k)$  is no longer a sufficient statistic for optimal decisions. But since we are interested in applying RL techniques to solve the DEC-POMDP problem, we may still use belief vector  $\mathbf{b}_j(k)$  to obtain a reasonably good suboptimal solution in a distributed multi-user setting, as shown in [99]. This would simplify the problem, yet leading to near-optimal solutions.

At each time slot, each secondary user updates its belief vector about the states of the channels in the next slot. Suppose secondary user  $j$  senses channel  $i = a_j(k)$  in time slot  $k$  and observes  $Y_i(k, j)$ . Then it updates its belief about the state of channel  $i$  in time  $k + 1$  using Bayes' formula as follows:

$$p_m(k + 1, j, i) = \frac{\sum_{l=0}^1 P_i(l, m) f_l(Y_i(k, j)) p_l(k, j, i)}{\sum_{l=0}^1 f_l(Y_i(k, j)) p_l(k, j, i)}, \quad (5.3)$$

where  $m \in \{0, 1\}$ . For the *unsensed* primary channels  $i' \neq a_j(k)$ , the  $j$ -th secondary user's belief vector is simply updated based on the assumed Markovian evolution:  $\mathbf{p}(k + 1, j, i') = \mathbf{p}(k, j, i') \mathbf{P}_{i'}, \forall i' \neq a_j(k)$ .

Figure 5.3 shows the update procedure in which thick arrows represent the updates using Bayes' formula, whereas thin arrows represent the updating of beliefs based only on the assumed Markovian nature of the channels.

### 5.4.1 The Reward and Value Functions

We define the total discounted return of user  $j \in \mathcal{K}_s$  in time slot  $k$  as  $R_j(k) = \sum_{n=0}^{\infty} \gamma^n r_j(k+n)$ , where  $r_j(k)$  is the reward of secondary user  $j$  in time slot  $k$  and  $\gamma \in (0, 1)$  is a discounting factor. In a fully observable MDP, an agent  $j \in \mathcal{K}_s$  may define the value of a state  $\mathbf{s}$  in slot  $k$  and under a policy  $\pi_j$  as [95]:

$$V_j^{\pi_j}(\mathbf{s}, k) = \mathbb{E} \{R_j(k) | \mathbf{s}(k) = \mathbf{s}\}. \quad (5.4)$$

Similarly, the function  $Q_j(s, a)$  is defined as the expected return starting from state  $s$ , taking the action  $a$ , and then following a policy  $\pi_j$  thereafter as:

$$Q_j^{\pi_j}(\mathbf{s}, a, k) = \mathbb{E} \{R_j(k) | \mathbf{s}(k) = \mathbf{s}, a_j(k) = a\}. \quad (5.5)$$

In the case of a POMDP, however, the actual state of the system is the belief vector  $\mathbf{b}_j(k)$ . Hence, the resulting process is an infinite state MDP which makes the solutions of (5.4) and (5.5) computationally expensive. In particular, our assumed model of a DEC-POMDP is a non-cooperative multi-agent system whose solution is shown to be NEXP-hard [103]. Hence, we will solve this problem by finding the Q values of the DEC-POMDP model by using the underlying MDP model [106], as explained in the next section.

### 5.4.2 Reinforcement Learning for DEC-POMDP

In the following, we extend the Q-learning algorithm that is defined for centralized fully observable environments in [95] by extending it to the partially observable channel sensing problem. This can be made by assigning a  $Q(s, a)$  table for each secondary user  $j$ , where  $s \in \mathcal{S}$  is the channels' states vector with  $u(s) \in \mathcal{U} = \{1, \dots, 2^L\}$  being the index of state  $s$  and  $a \in \mathcal{C}$  is the index of the sensed channel. However, we do not use the belief vector  $\mathbf{b}_j(k)$  as the actual state. Instead, we

solve for the values of  $Q(s, a)$  in the underlying MDP model by using  $\mathbf{b}_j(k)$  as a weighting vector, as described in [106]. Although this is not the optimal solution of the DEC-POMDP problem, [106] shows that this approach leads to a near-optimal solution with a very low computational complexity if the algorithm adopts an  $\varepsilon$ -greedy policy [95].

Since the secondary users cannot fully observe the state of the primary system in the POMDP environment, the sensing policy of each secondary user is based on the belief vector  $\mathbf{b}_j(k) = [b_j(1, k), \dots, b_j(2^L, k)]$ . We describe the Q-learning procedure for each user  $j \in \mathcal{K}_s$  in Algorithm 6. Given a belief vector  $\mathbf{b} = [b(1), \dots, b(2^L)]$ , we define the Q-value of the belief vector  $\mathbf{b}$  as:

$$Q_{\mathbf{b}}(a) = \sum_{s \in \mathcal{S}} b(u(s))Q(s, a), \quad (5.6)$$

and the update function as:

$$\Delta Q_{\mathbf{b}}(s, a) = \xi b(u(s)) \left[ r_j(k) + \gamma \max_{a' \in \mathcal{C}} Q_{\mathbf{b}'}(a') - Q(s, a) \right].$$

We define  $\xi$  to be the learning rate. The Q-value  $Q(s, a)$  is updated after taking every action using:

$$Q(s, a) \leftarrow Q(s, a) + \Delta Q_{\mathbf{b}}(s, a). \quad (5.7)$$

This update is done for every state  $s \in \mathcal{S}$ .

## 5.5 Simulation Results

We assume that all primary channels  $i \in \mathcal{C}$  have the same transition probabilities that are governed by the transition matrix:

$$\mathbb{P}_i = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}. \quad (5.8)$$

We define the average spectrum hole utilization as:

$$U = \frac{\sum_{j=1}^{K_s} \sum_{k=1}^{\infty} \mathcal{I}_{\{r_j(k)=1\}}}{\sum_{i=1}^L \sum_{k=1}^{\infty} \mathcal{I}_{\{s_i(k)=0\}}}. \quad (5.9)$$

The reinforcement values (rewards) are selected as follows:

1.  $r_j(k) = 1$  if secondary  $j$  successfully accesses channel  $a_j(k)$  at time  $k$ .
2.  $r_j(k) = -0.5$  if secondary  $j$  back-off due to collision with another secondary user, and conditioned on the channel being idle.
3.  $r_j(k) = 0$  if the sensed channel is busy.

In the random sensing scenario, the average number of secondary users that are sensing a given primary channel is  $\mathbb{E}\{Z_i(k)\} = \frac{K_s}{L(1-(1-1/L)^{K_s})}$ , where  $Z_i(k) \in \{1, \dots, K_s\}$  is a zero-truncated binomial random variable with parameters  $K_s$  and  $1/L$ . Thus, in the random sensing scenario, we set the false alarm probability of each secondary user to  $\alpha = 1 - (1 - \alpha_0)^{1/\mathbb{E}\{Z_i(k)\}}$ .

On the other hand, when applying the Q-learning algorithm, the secondary users will be evenly distributed over the channels. Therefore,  $\mathbb{E}\{Z_i(k)\} = \frac{K_s}{L}$  if  $K_s \geq L$ , and  $\mathbb{E}\{Z_i(k)\} = 1$  otherwise.

We note that  $\mathbb{E}\{Z_i(k)\}$  is conditioned on the channel  $i$  being sensed (i.e. conditioned on  $\{Z_i(k) \neq 0\}$ ).

In the following simulations, we model the sensing observations of channel  $i \in \mathcal{C}$  as the output of a BSC with cross-over probability  $\nu_i$ , and we let  $\nu = [\nu_1, \dots, \nu_L]$ . The use of a BSC permits to simplify the analysis, yet it is applicable to different channel environments since  $\nu_i$  can depend on the channel fading model, the detector type, the signal and noise power, and the prior distributions of the information message. Interested readers are referred to [158, 159, 235] for the computation of  $\nu_i$  under different channel conditions and with different detection methods.

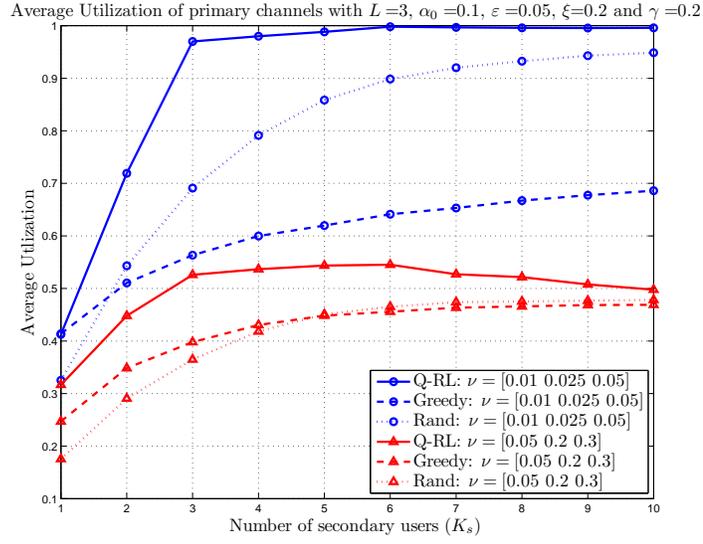


Figure 5.4: Average Utilization of Primary channels for  $\alpha_0 = 0.1$ .

We compare the performance of our proposed channel access/sensing mechanism

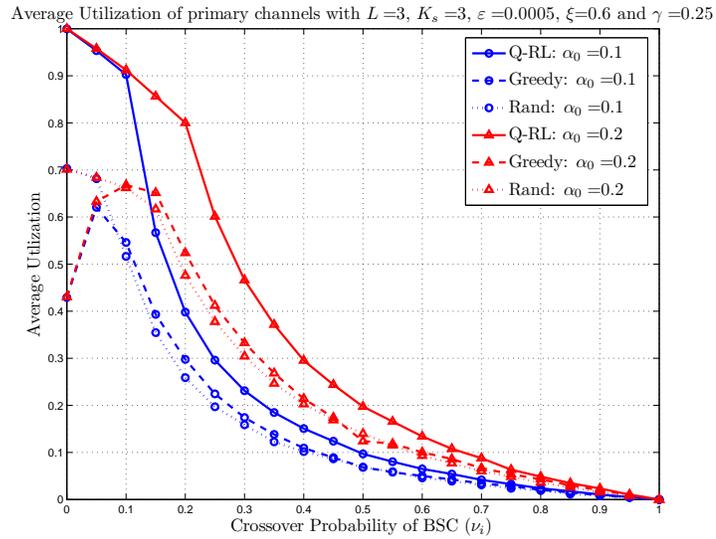


Figure 5.5: Average Utilization of Primary channels for  $K_s = 3$ .

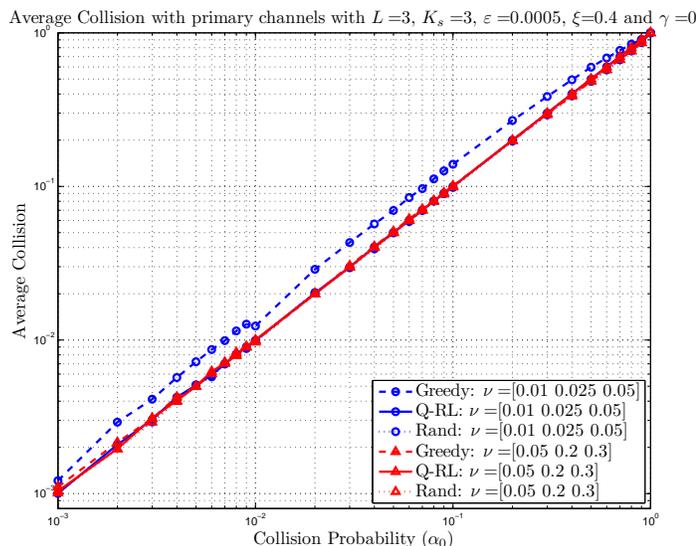


Figure 5.6: Collision rates with Primary channels for  $K_s = 3$ .

to the greedy approach that was proposed in [152]. This greedy approach is equivalent to the *single-user approach* that is defined in [232] and which is applied as a non-cooperative myopic policy in multiuser OSA. In Fig. 5.4, we observe that RL permits to achieve high utilization of the spectrum opportunities in the primary channels. In particular, in the low-noise regime, the spectrum utilization approaches 100%. Moreover, the RL algorithm has a significant advantage over the greedy algorithm of [152] because the greedy algorithm makes most of the secondary users to sense the channel that is most likely to be idle, thus ignoring other possible spectrum opportunities and causing collisions among secondary users, as stated in [232]. This is expected because the greedy algorithm is an optimal *myopic* strategy for a single-user case and can only be a suboptimal strategy in a multiuser context. On the other hand, a simple random sensing policy that selects randomly a channel at each time instant can outperform the greedy algorithm of [152] as the number of secondary users  $K_s$  increases. That is because a random policy reduces the collisions among the secondary users, compared to the greedy policy of [152].

Next, we assume all primary channels to have the same crossover probability  $\nu_i$  and we show in Fig. 5.5 the impact of the sensing noise on the performance of both the Q-learning and random sensing systems. We see that the performance drops at a higher rate when the crossover probability of the sensing BSC ( $\nu_i$ ) becomes greater than the false alarm probability  $\alpha$  of *each* secondary user.

In Fig. 5.6, we analyze the collision probability that results from our designed NP detectors. Here we are controlling the collision probability with the primary channels during the time slots in which a primary channel is being sensed. Figure 5.6 shows the accuracy of the proposed decentralized collision probability control in maintaining the collision rate equal to the prescribed threshold  $\alpha_0$ , by using either of the RL or the random sensing protocols that are proposed in this chapter. From Fig. 5.6 it can be seen that these algorithms are robust against channel impairments as captured by  $\nu_i$ . The efficiency of these algorithms is due to the fact that they estimate the number of secondary users that are sensing each channel, and based on this information, the channel access rule is updated so that the collision rate with primary users is maintained within the required bound. We observe also that the greedy policy violates the prescribed collision probability with primary users when the observation noise  $\nu_i$  is low. However, in this case, the excess in collision probability is not very large, compared with  $\alpha_0$ , because most of the users sense the most likely idle channel, whereas a small number of users would sense a busy channel according to the greedy approach.

## 5.6 Conclusion

In this chapter, we derived channel sensing and accessing protocols for secondary users in decentralized cognitive networks. The sensing policy is completely decentralized and is obtained by using RL. The proposed policy ensures efficient utilization

of the spectrum resources since it exploits the Markovian nature of the primary channel traffic and limits the collisions among competing secondary users. Also, we have designed a secondary detector that maximizes the detection probability of the idle channels while satisfying the collision probability constraint imposed by primary users. The designed policies are characterized by their robustness and accuracy, and help to enhance the cognitive capabilities of secondary users.

---

**Algorithm 6** Q-learning Algorithm for agent  $j \in \mathcal{K}_s$

---

**for** each  $s \in \mathcal{S}$ .  $a \in \mathcal{C}$  **do**

Initialize  $Q(s, a) = 0$ .

**end for**

Initialize the belief vector  $\mathbf{b}$  arbitrarily.

**for** each time slot  $k$  **do**

Generate a random number  $rnd$  between 0 and 1.

**if**  $rnd < \varepsilon$  **then**

Select action  $a^*$  randomly.

**else**

Select action  $a^* = \arg \max_a Q_{\mathbf{b}}(a)$ .

**end if**

Execute action  $a^*$  (i.e. sense channel  $a^*$ ).

Receive the immediate reward  $r_j(k)$ .

Update  $p_0(a^*, k, j)$  using the observation  $y(k)$ :

$$p_0(a^*, k, j) \leftarrow \frac{f_0(y(k))p_0(a^*, k, j)}{\sum_{l=0}^1 f_l(y(k))p_l(a^*, k, j)}$$

Update the current belief  $\mathbf{b}$  according to  $p_0(a^*, k, j)$ .

Evaluate the next belief vector  $\mathbf{b}'$  based on (5.3).

Update the table entries as follows:

$$Q(s, a^*) \leftarrow Q(s, a^*) + \Delta Q_{\mathbf{b}}(s, a^*), \forall s \in \mathcal{S}.$$

$\mathbf{b} \leftarrow \mathbf{b}'$ .

**end for**

---

# Chapter 6

## Summary of the Dissertation and Research Directions

In this dissertation, we have developed a CR architecture that is equipped with autonomous sensing and learning abilities to adapt to alien radio environments. The proposed architecture uses sophisticated spectrum sensing tools as well as machine learning techniques to explore and act autonomously in the surrounding RF environment.

In the followings, we summarize the main aspects and contributions of this dissertation, and propose possible research directions that can be addressed in future works.

### 6.1 Summary of the Dissertation

In Chapter 2, we proposed an autonomous CR architecture that is characterized by its wideband operation and self-learning ability. The proposed architecture uses so-

## *Chapter 6. Summary of the Dissertation and Research Directions*

phisticated signal processing and machine learning tools. Throughout our study, we focused mainly on developing wideband signal processing algorithms to detect and identify the active signals in a certain RF domain. The proposed sensing algorithm is based, in part, on a sliding-window energy detector that maximizes the detection probability of active signals, subject to a certain false alarm constraint. Cyclostationary detection was used at the second stage of spectrum sensing to detect the underlying cyclic properties of the detected signals. As a result, feature vectors can be extracted from the detected signals, characterizing the active signals in the surrounding environment. We analyzed the performance of the proposed sensing algorithm in both non-fading and Rayleigh fading environments. The results showed efficient signal detection, even at relatively low SNR's.

In Chapter 3, we presented a survey of machine learning techniques in CR's, focusing mainly on signal classification and decision-making methods. We identified the unique nature of learning in CR's and showed its importance in achieving a real CR system. We also presented several machine learning techniques that can be applied in CR applications, and listed them under different categories in a hierarchical order. As a result, we have identified the most appropriate machine learning algorithm that can be used in a particular situation. Furthermore, we provided a brief description of the presented machine learning tools, while showing the advantages and disadvantages of each technique.

In Chapter 4, we considered the problem of signal classification using a Bayesian non-parametric approach based on the DPMM framework. The DPMM-based classifier was shown to be a suitable candidate for our classification problem due to its non-parametric support probability distribution, which allows for an infinite number of mixture components, and therefore, an infinite number of clusters. The DPMM classifier was extended to both Gaussian and non-Gaussian observation models to improve the flexibility of this algorithm in matching the feature observation model.

In addition, we proposed a simplified and a sequential DPMM classifier that reduce the computational burden of the DPMM classifier by exploiting the underlying CRP structure of the Dirichlet process. This was achieved by defining a parameter selection policy for the Gibbs sampler, which was shown to improve the efficiency of the Gibbs sampling-based DPMM classifier. We have shown, through simulations, that our proposed DPMM-based classification algorithm can lead to perfect classification results in most of the scenarios.

Finally, in Chapter 5, we developed a spectrum sensing and channel access policy for CR's in distributed CRN's based on DSA. The proposed policy is aimed at multi-agent scenarios in which multiple CR's try to sense and access, independently, a set of primary channels. The problem was formulated in a DEC-POMDP framework which is well-known to be an open problem in decision-making literature. Hence, we proposed an efficient RL algorithm that can be applied in this setup in order to reach action coordination among CR's in a distributed CRN. The proposed algorithm was shown to achieve near-optimal policy without incurring any control overhead among cognitive users. The overall system can achieve spectrum awareness and self-learning abilities, thus laying down the fundamental structure of autonomous CR's.

## **6.2 Future Research Directions**

The work that is presented in this dissertation can be extended along several directions, focusing on either spectrum sensing or decision-making applications.

### **Robust Signal Detection**

In spectrum sensing applications, we may consider the problem of robust signal detection in the presence of outliers and contaminating non-Gaussian noise models [236].

This problem is particularly important for wideband spectrum sensing applications in which CR's can be subject to a wide range of RF activities, including potential interferers and jammers. The conventional energy detection method is vulnerable to such non-Gaussian noise scenarios, which may cause degradation in the detection performance. Hence, in order to overcome this problem, robust detection methods can be implemented by assuming a certain contaminating noise model or by using robust cost functions for power spectral estimation [230, 236]. The obtained robust signal detectors would be able to improve the reliability of spectrum sensing in highly dynamic RF environments.

### **Compressive Sensing**

Another aspect of spectrum sensing applications may consider compressive sensing methods to reduce the computational complexity of wideband CR applications [65]. Although this problem usually assumes sparse RF signals in the spectral frequency domain, it is worth to be addressed in cyclostationary detection applications. In particular, compressive sensing can be applied to cyclostationary feature detection due to the sparsity of the SCF in the 2-dimensional  $(f, \alpha)$ -plane [237]. Therefore, compressive sensing can be a perfect candidate for cyclostationary detection of wideband signals and can help to reduce both computational burden and hardware cost of such techniques.

### **Multi-agent RL in decentralized CRN's**

The problem of decentralized decision-making in CRN's can be investigated further in multi-agent scenarios. Although our proposed RL algorithm was shown to lead to a satisfactory solution in decentralized networks, it is initially aimed at fully observable single-agent decision-making scenarios. Hence, the RL algorithm should be analyzed

further in decentralized control problems. This is still an open problem and has been identified as the multi-agent RL problem which aims at achieving optimal action coordination among distributed learning agents [100].

In contrast with RL-based MDP solutions, the optimal policy of a multi-agent RL is not necessarily deterministic [238]. Therefore, any formulation of the multi-agent RL problem should be based on stochastic policies. Several attempts have been made to address this problem using a Markov game formulation, as in [238]. The proposed solution in [238] is obtained by using the *minimax* approach, but this solution does not scale very easily with the number of learning agents. On the other hand, [239] proposed a Q-learning algorithm for multi-agent RL with partially observable environments and showed the convergence of this algorithm. However, the performance of this algorithm was only analyzed numerically without any analytical interpretation.

# Appendix A

## Derivation of the ROC for Carrier Frequency Detection

Consider a sampled data sequence  $\{x(k)\}_{k=0}^{M-1}$ , with  $T_s$  as the sampling period. We denote by  $\{X(n)\}_{n=0}^{M-1}$  its DFT obtained by fast Fourier transform (FFT) algorithm:

$$X(n) = \sum_{k=0}^{M-1} x(k)e^{-j2\pi n \frac{k}{M}}, \text{ for } n = 0, \dots, M-1. \quad (\text{A.1})$$

The average power in a spectral window of odd length  $L$ , centered at  $n$ , can be approximated by  $T(n) = \sum_{l=-(L-1)/2}^{(L-1)/2} |X(n+l)|^2$ . In order to derive the ROC of the NP detector, we determine the distribution of  $T(n)$  under the two hypotheses:

$$\mathcal{H}_0 : x(k) = w(k), \quad (\text{A.2})$$

$$\mathcal{H}_1 : x(k) = s(k) + w(k), \quad (\text{A.3})$$

where  $\{w(k)\}_{k=0}^{M-1}$  are modeled as i.i.d. Gaussian random variables, s.t.  $w(k) \sim \mathcal{N}(0, P_n)$ . The signal  $\{s(k)\}_{k=0}^{M-1}$  in (A.3) can be modeled as i.i.d. Gaussian random variables, s.t.  $s(k) \sim \mathcal{N}(0, P_s)$ . This is a reasonable assumption for signals that are perturbed by propagation through turbulent media and multipath fading [240]. It is

Appendix A. Derivation of the ROC for Carrier Frequency Detection

well-known that the energy detector is the optimal detector for unknown (random) signals.

Let  $\mathbf{x} = [x(0), \dots, x(M-1)]^T$ ,  $\mathbf{X} = [X(0), \dots, X(M-1)]^T$ ,  $\mathbf{X}^R = [\Re\{X(0)\}, \dots, \Re\{X(M-1)\}]^T$  and  $\mathbf{X}^I = [\Im\{X(0)\}, \dots, \Im\{X(M-1)\}]^T$ , where  $\Re\{\}$  and  $\Im\{\}$  denote the real and imaginary parts, respectively. The DFT in (A.1) can be expressed as:

$$\mathbf{X}^C \triangleq \begin{bmatrix} \mathbf{X}^R \\ \mathbf{X}^I \end{bmatrix} = \mathbf{A}\mathbf{x}, \quad (\text{A.4})$$

where  $\mathbf{A}$  is a  $2M$ -by- $M$  matrix of DFT coefficients. Since  $\{x(k)\}_{k=0}^{M-1}$  are zero-mean i.i.d. Gaussian random variables, then  $\mathbf{X}^C$  is a jointly Gaussian random vector. It can be shown that, under  $\mathcal{H}_0$ ,  $\mathbb{E}\{\mathbf{X}^C (\mathbf{X}^C)^T\} = \frac{MP_n}{2}I_M$  (where  $I_M$  is an  $M$ -by- $M$  identity matrix) and under  $\mathcal{H}_1$   $\mathbb{E}\{\mathbf{X}^C (\mathbf{X}^C)^T\} = \frac{M(P_s+P_n)}{2}I_M$ . Therefore, elements of  $\mathbf{X}^C$  are uncorrelated. Since  $\mathbf{X}^C$  is jointly Gaussian with uncorrelated elements, the elements of  $\mathbf{X}^C$  are then independent. Also, since all the elements have the same variance under the same hypothesis, elements of  $\mathbf{X}^C$  are assumed to be i.i.d. zero-mean Gaussian random variables with variance  $\frac{MP_n}{2}$  under  $\mathcal{H}_0$ , and  $\frac{M(P_n+P_s)}{2}$  under  $\mathcal{H}_1$ .

Under the above assumptions,  $T'(n) = \frac{2}{MP_n}T(n)$  is a sufficient statistic for the hypothesis testing and follows a  $\chi_{2L}^2$  distribution. The threshold  $\eta$  for carrier frequency detection is defined s.t.  $\Pr\{T'(n) > \eta | \mathcal{H}_0\} \leq \alpha_F$ , where  $\alpha_F$  is the acceptable false alarm probability. Note that the noise power  $P_n$  can be estimated, for example, by using the method proposed in [117].

The NP decision rule  $\delta$  for carrier frequency detection is then defined as:

$$\delta(T'(n)) = \begin{cases} 0 & \text{if } T'(n) < \eta \\ 1 & \text{otherwise} \end{cases}, \quad (\text{A.5})$$

where  $\eta = 2\gamma^{-1}(L; (1 - \alpha_F)\Gamma(L))$ ,  $\gamma^{-1}$  is the inverse lower incomplete gamma function (where  $\gamma(k; x) = \int_0^x t^{k-1}e^{-t}dt$  and the inverse is w.r.t. the second argument)

*Appendix A. Derivation of the ROC for Carrier Frequency Detection*

and  $\Gamma(k) = \int_0^\infty t^{k-1} e^{-t} dt$  is the gamma function. By applying this to the PSD in (2.4), the threshold is given by:

$$\eta_{PSD} = \frac{\eta P_n}{2T_s L} = \frac{\gamma^{-1}(L; (1 - \alpha_F) \Gamma(L)) P_n}{T_s L} . \quad (\text{A.6})$$

The resulting detection probability of this detector can be expressed as:

$$P_D = \Pr\{T'(n) > \eta | \mathcal{H}_1\} = 1 - \frac{\gamma\left(L; \frac{\eta}{2(1+SNR)}\right)}{\Gamma(L)} , \quad (\text{A.7})$$

which represents the ROC of the carrier frequency detector.

# References

- [1] S. K. Jayaweera and C. G. Christodoulou, “Radiobots: Architecture, algorithms and realtime reconfigurable antenna designs for autonomous, self-learning future cognitive radios,” University of New Mexico, Technical Report EECE-TR-11-0001, Mar. 2011. [Online]. Available: <http://repository.unm.edu/handle/1928/12306>
- [2] J. Mitola, III and G. Maguire, Jr., “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
- [3] J. Mitola, “Cognitive radio: An integrated agent architecture for software defined radio,” Ph.D. dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, 2000.
- [4] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [5] FCC, “Report of the spectrum efficiency working group,” FCC spectrum policy task force, Tech. Rep., Nov. 2002.
- [6] ———, “ET docket no 03-322 notice of proposed rulemaking and order,” Tech. Rep., Dec. 2003.
- [7] A. Goldsmith, S. Jafar, I. Maric, and S. Srinivasa, “Breaking spectrum gridlock with cognitive radios: An information theoretic perspective,” *Proceedings of the IEEE*, vol. 97, no. 5, pp. 894–914, May 2009.
- [8] N. Devroye, M. Vu, and V. Tarokh, “Cognitive radio networks,” *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 12–23, Nov. 2008.

## References

- [9] Q. Zhao and B. M. Sadler, “A survey of dynamic spectrum access,” *IEEE Sig. Proc. Magazine*, vol. 24, no. 3, pp. 79–89, May 2007.
- [10] G. Zhao, J. Ma, Y. Li, T. Wu, Y. H. Kwon, A. Soong, and C. Yang, “Spatial spectrum holes for cognitive radio with directional transmission,” in *2008 IEEE Global Telecommunications Conference (GLOBECOM 2008)*, Nov. 2008, pp. 1–5.
- [11] A. Ghasemi and E. Sousa, “Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs,” *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 32–39, April 2008.
- [12] B. Farhang-Boroujeny, “Filter bank spectrum sensing for cognitive radios,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 5, pp. 1801–1811, May 2008.
- [13] B. Farhang-Boroujeny and R. Kempter, “Multicarrier communication techniques for spectrum sensing and communication in cognitive radios,” *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 80–85, April 2008.
- [14] C. R. C. da Silva, C. Brian, and K. Kyouwoong, “Distributed spectrum sensing for cognitive radio systems,” in *Information Theory and Applications Workshop, 2007*, 29 2007-Feb. 2 2007, pp. 120–123.
- [15] C. Cordeiro, M. Ghosh, D. Cavalcanti, and K. Challapali, “Spectrum sensing for dynamic spectrum access of tv bands,” in *Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on*, Aug. 2007, pp. 225–233.
- [16] H. Chen, W. Gao, and D. G. Daut, “Signature based spectrum sensing algorithms for ieee 802.22 wran,” in *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007, pp. 6487–6492.
- [17] Y. Zeng and Y. Liang, “Maximum-minimum eigenvalue detection for cognitive radio,” in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, Sept. 2007, pp. 1–5.
- [18] —, “Covariance based signal detections for cognitive radio,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, April 2007, pp. 202–207.
- [19] X. Zhou, Y. Li, Y. H. Kwon, and A. Soong, “Detection timing and channel selection for periodic spectrum sensing in cognitive radio,” in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, Nov. 2008, pp. 1–5.

## References

- [20] G. Ganesan and Y. Li, “Cooperative spectrum sensing in cognitive radio, part i: Two user networks,” *Wireless Communications, IEEE Transactions on*, vol. 6, no. 6, pp. 2204–2213, June 2007.
- [21] —, “Cooperative spectrum sensing in cognitive radio, part ii: Multiuser networks,” *Wireless Communications, IEEE Transactions on*, vol. 6, no. 6, pp. 2214–2222, June 2007.
- [22] J. Chen, A. Gibson, and J. Zafar, “Cyclostationary spectrum detection in cognitive radios,” in *IET Seminar on Cognitive Radio and Software Defined Radios: Technologies and Techniques 2008*, London, UK, Sep. 2008, pp. 1–5.
- [23] S. Huang, X. Liu, and Z. Ding, “Opportunistic spectrum access in cognitive radio networks,” in *The 27th Conference on Computer Communications. IEEE INFOCOM '08*, Phoenix, AZ, Apr. 2008, pp. 1427–1435.
- [24] Y. Tawk, M. Bkassiny, G. El-Howayek, S. Jayaweera, K. Avery, and C. Christodoulou, “Reconfigurable front-end antennas for cognitive radio applications,” *IET Microwaves, Antennas Propagation*, vol. 5, no. 8, pp. 985–92, 2011.
- [25] S. K. Jayaweera and T. Li, “Dynamic spectrum leasing in cognitive radio networks via primary-secondary user power control games,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 3300–3310, July 2009.
- [26] S. K. Jayaweera, G. Vazquez-Vilar, and C. Mosquera, “Dynamic spectrum leasing: A new paradigm for spectrum sharing in cognitive radio networks,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2328–39, 2010.
- [27] S. K. Jayaweera, M. Bkassiny, and K. A. Avery, “Asymmetric cooperative communications based spectrum leasing via auctions in cognitive radio networks,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2716–2724, Aug. 2011.
- [28] S. Haykin, D. Thomson, and J. Reed, “Spectrum sensing for cognitive radio,” *Proceedings of the IEEE*, vol. 97, no. 5, pp. 849–877, May. 2009.
- [29] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 116–130, Quarter 2009.
- [30] K. Ben Letaief and W. Zhang, “Cooperative communications for cognitive radio networks,” *Proceedings of the IEEE*, vol. 97, no. 5, pp. 878–893, May 2009.

## References

- [31] J. Ma, G. Y. Li, and B. H. Juang, “Signal processing in cognitive radio,” *Proceedings of the IEEE*, vol. 97, no. 5, pp. 805–823, May 2009.
- [32] W. Zhang, R. Mallik, and K. Letaief, “Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5761–5766, Dec. 2009.
- [33] Y. M. Kim, G. Zheng, S. H. Sohn, and J. M. Kim, “An alternative energy detection using sliding window for cognitive radio system,” in *10th International Conference on Advanced Communication Technology (ICACT '08)*, vol. 1, Gangwon-Do, South Korea, Feb. 2008, pp. 481–485.
- [34] A. Dandawate and G. Giannakis, “Statistical tests for presence of cyclostationarity,” *IEEE Transactions on Signal Processing*, vol. 42, no. 9, pp. 2355–2369, Sep. 1994.
- [35] B. Deepa, A. Iyer, and C. Murthy, “Cyclostationary-based architectures for spectrum sensing in IEEE 802.22 WRAN,” in *IEEE Global Telecommunications Conference (GLOBECOM '10)*, Miami, FL, Dec. 2010, pp. 1–5.
- [36] J. Lunden, V. Koivunen, A. Huttunen, and H. Poor, “Collaborative cyclostationary spectrum sensing for cognitive radio systems,” *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4182–4195, 2009.
- [37] M. Gandetto and C. Regazzoni, “Spectrum sensing: A distributed approach for cognitive terminals,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 546–557, Apr. 2007.
- [38] J. Unnikrishnan and V. Veeravalli, “Cooperative sensing for primary detection in cognitive radio,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 18–27, Feb. 2008.
- [39] T. Cui, F. Gao, and A. Nallanathan, “Optimization of cooperative spectrum sensing in cognitive radio,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1578–1589, May 2011.
- [40] M. Bkassiny, S. K. Jayaweera, Y. Li, and K. A. Avery, “Wideband spectrum sensing and non-parametric signal classification for autonomous self-learning cognitive radios,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2596–2605, July 2012.
- [41] —, “Blind cyclostationary feature detection based spectrum sensing for autonomous self-learning cognitive radios,” in *IEEE International Conference on Communications (ICC '12)*, Ottawa, Canada, June 2012.

## References

- [42] M. Bkassiny, S. K. Jayaweera, and Y. Li, "Multidimensional Dirichlet process-based non-parametric signal classification for autonomous self-learning cognitive radios," *IEEE Transactions on Wireless Communications*, 2013, [In review].
- [43] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006. [Online]. Available: <http://www.jstor.org/stable/27639773>
- [44] A. El-Saleh, M. Ismail, M. Ali, and J. Ng, "Development of a cognitive radio decision engine using multi-objective hybrid genetic algorithm," in *IEEE 9th Malaysia International Conference on Communications (MICC 2009)*, Dec. 2009, pp. 343–347.
- [45] L. Morales-Tirado, J. Suris-Pietri, and J. Reed, "A hybrid cognitive engine for improving coverage in 3g wireless networks," in *IEEE International Conference on Communications Workshops (ICC Workshops 2009)*., June 2009, pp. 1–5.
- [46] Y. Huang, H. Jiang, H. Hu, and Y. Yao, "Design of learning engine based on support vector machine in cognitive radio," in *International Conference on Computational Intelligence and Software Engineering (CiSE '09)*, Wuhan, China, Dec. 2009, pp. 1–4.
- [47] Y. Huang, J. Wang, and H. Jiang, "Modeling of learning inference and decision-making engine in cognitive radio," in *Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, vol. 2, Apr. 2010, pp. 258–261.
- [48] Y. Yang, H. Jiang, and J. Ma, "Design of optimal engine for cognitive radio parameters based on the duga," in *3rd International Conference on Information Sciences and Interaction Sciences (ICIS 2010)*, June 2010, pp. 694–698.
- [49] H. Volos and R. Buehrer, "Cognitive engine design for link adaptation: An application to multi-antenna systems," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2902–2913, Sep. 2010.
- [50] Y. Li, S. K. Jayaweera, and C. G. Christodoulou, "Wideband PHY/MAC bandwidth aggregation optimization for cognitive radios," in *Third International Workshop on Cognitive Information Processing*, Parador de Baiona, Spain, May 2012.
- [51] H. Wu and S. Wang, "An efficient and robust approach for wideband compressive spectrum sensing," in *IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC '12)*, Hong Kong, China, Aug. 2012.

## References

- [52] Z. Tian and G. B. Giannakis, “A wavelet approach to wideband spectrum sensing for cognitive radios,” in *1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, Mykonos Island, Greece, June 2006, pp. 1–5.
- [53] G. Vazquez-Vilar, R. Lo andpez Valcarce, C. Mosquera, and N. Gonzalez Prelicic, “Wideband spectral estimation from compressed measurements exploiting spectral a priori information in cognitive radio systems,” in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP '10)*, Dallas, TX, Mar. 2010, pp. 2958–2961.
- [54] Z. Tian and G. Giannakis, “Compressed sensing for wideband cognitive radios,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 4, Honolulu, HI, Apr. 2007, pp. IV–1357–IV–1360.
- [55] A. Taherpour, S. Gazor, and M. Nasiri-Kenari, “Wideband spectrum sensing in unknown white gaussian noise,” *IET Communications*, vol. 2, no. 6, pp. 763–771, July 2008.
- [56] J. Kim and C. Christodoulou, “A simple reconfigurable microstrip antenna for wideband applications,” in *Antennas and Propagation Society International Symposium (APSURSI), 2010 IEEE*, 2010, pp. 1–4.
- [57] Z. Zhang, H. Li, D. Yang, and C. Pei, “Space-time bayesian compressed spectrum sensing for wideband cognitive radio networks,” in *IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN '10)*, Singapore, Singapore, Apr. 2010, pp. 1–11.
- [58] U. Nakarmi and N. Rahnavard, “Joint wideband spectrum sensing in frequency overlapping cognitive radio networks using distributed compressive sensing,” in *Military Communications Conference (MILCOM '11)*, Baltimore, MD, Nov. 2011, pp. 1035–1040.
- [59] M. Mishali and Y. Eldar, “Wideband spectrum sensing at sub-nyquist rates [applications corner],” *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 102–135, July 2011.
- [60] S. K. Jayaweera, Y. Li, M. Bkassiny, C. Christodoulou, and K. Avery, “Radiobots: The autonomous, self-learning future cognitive radios,” in *International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS '11)*, Chiangmai, Thailand, Dec. 2011, pp. 1–5.
- [61] Y. Tawk, J. Costantine, K. Avery, and C. Christodoulou, “Implementation of a cognitive radio front-end using rotatable controlled reconfigurable antennas,”

## References

- IEEE Transactions on Antennas and Propagation*, vol. 59, no. 5, pp. 1773–1778, May 2011.
- [62] Y. Tawk, J. Costantine, S. Hemmady, G. Balakrishnan, K. Avery, and C. Christodoulou, “Demonstration of a cognitive radio front end using an optically pumped reconfigurable antenna system (OPRAS),” *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 2, pp. 1075–1083, 2012.
- [63] B. Razavi and B. Sahoo, “A 12-bit 200-mhz cmos adc,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 9, pp. 2366–2380, Sep. 2009.
- [64] C.-C. Huang, C.-Y. Wang, and J.-T. Wu, “A cmos 6-bit 16-gs/s time-interleaved adc using digital background calibration techniques,” *IEEE Journal of Solid-State Circuits*, vol. 46, no. 4, pp. 848–858, Apr. 2011.
- [65] E. Candes and M. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [66] Y. Tawk, A. Albrecht, S. Hemmady, G. Balakrishnan, and C. Christodoulou, “Optically pumped reconfigurable antenna systems (OPRAS),” in *IEEE Antennas and Propagation Society International Symposium (APSURSI '10)*, Toronto, Canada, July 2010, pp. 1–4.
- [67] Y. Li, S. K. Jayaweera, M. Bkassiny, and C. Ghosh, “Learning-aided sub-band selection algorithms for spectrum sensing in wide-band cognitive radios,” 2013, [In preparation].
- [68] K. Kim, I. Akbar, K. Bae, J. sun Urn, C. Spooner, and J. Reed, “Cyclostationary approaches to signal detection and classification in cognitive radio,” in *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '07)*, Dublin, Ireland, Apr. 2007, pp. 212–215.
- [69] P. Sutton, K. Nolan, and L. Doyle, “Cyclostationary signatures in practical cognitive radio applications,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 13–24, Jan. 2008.
- [70] W. A. Gardner, *Statistical Spectral Analysis: A Nonprobabilistic Theory*. Englewood Cliffs, NJ: Prentice Hall, 1987.
- [71] W. Gardner, “Measurement of spectral correlation,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 5, pp. 1111–1123, Oct. 1986.

## References

- [72] ———, “Spectral correlation of modulated signals: Part i— analog modulation,” *IEEE Transactions on Communications*, vol. 35, no. 6, pp. 584 – 594, June 1987.
- [73] W. Gardner, W. Brown, and C.-K. Chen, “Spectral correlation of modulated signals: Part ii—digital modulation,” *IEEE Transactions on Communications*, vol. 35, no. 6, pp. 595 – 601, June 1987.
- [74] Y. Wu, B. Wang, and K. Liu, “Optimal Defense against Jamming Attacks in Cognitive Radio Networks Using the Markov Decision Process Approach,” in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Dec. 2010, pp. 1 –5.
- [75] A. Sampath, H. Dai, H. Zheng, and B. Zhao, “Multi-channel Jamming Attacks using Cognitive Radios,” in *Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN 2007)*, Aug. 2007, pp. 352 –357.
- [76] Q. Zhu, H. Li, Z. Han, and T. Bas andar, “A Stochastic Game Model for Jamming in Multi-Channel Cognitive Radio Systems,” in *2010 IEEE International Conference on Communications (ICC 2010)*, May 2010, pp. 1 –6.
- [77] J. Popoola and R. van Olst, “Application of neural network for sensing primary radio signals in a cognitive radio environment,” in *AFRICON 2011*, Livingstone, Zambia, Sep. 2011, pp. 1 –6.
- [78] M. Ramon, T. Atwood, S. Barbin, and C. Christodoulou, “Signal classification with an svm-fft approach for feature extraction in cognitive radio,” in *SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC '09)*, Belem, Brazil, Nov. 2009, pp. 286 –289.
- [79] T. Clancy, A. Khawar, and T. Newman, “Robust signal classification using unsupervised learning,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 4, pp. 1289 –1299, Apr. 2011.
- [80] D. Pelleg and A. Moore, “X-means: extending k-means with efficient estimation of the number of clusters,” in *Seventh International Conference on Machine Learning (ICML '00)*, Stanford, CA, June-July 2000.
- [81] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, “Bayesian nonparametric methods for learning markov switching processes,” *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 43 –54, Nov. 2010.

## References

- [82] ———, “Bayesian nonparametric inference of switching dynamic linear models,” *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1569–1585, Apr. 2011.
- [83] N. Bouguila and D. Ziou, “A dirichlet process mixture of dirichlet distributions for classification and prediction,” in *IEEE Workshop on Machine Learning for Signal Processing (MLSP '08)*, Oct. 2008, pp. 297–302.
- [84] A. Rabaoui, N. Viandier, J. Marais, and E. Duflos, “On selecting the hyperparameters of the dpm models for the density estimation of observation errors,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '11)*, Prague, Czech Republic, May 2011, pp. 4092–4095.
- [85] M. D. Escobar and M. West, “Bayesian density estimation and inference using mixtures,” *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 577–588, 1995. [Online]. Available: <http://www.jstor.org/stable/2291069>
- [86] K. Ishiguro, T. Yamada, S. Araki, T. Nakatani, and H. Sawada, “Probabilistic speaker diarization with bag-of-words representations of speaker angle information,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 447–460, Feb. 2012.
- [87] A. Harati Nejad Torbati, J. Picone, and M. Sobel, “Applications of Dirichlet process mixtures to speaker adaptation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '12)*, Kyoto, Japan, Mar. 2012, pp. 4321–4324.
- [88] S. Li, Z. Yanning, M. Miao, and T. Guangjian, “Sar image segmentation method using dp mixture models,” in *International Symposium on Computer Science and Computational Technology (ISCCT '08)*, vol. 2, Shanghai, China, Dec. 2008, pp. 598–601.
- [89] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin, “Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds,” *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6140–6155, Dec. 2010.
- [90] M. Bkassiny and S. K. Jayaweera, “Sequential bayesian non-parametric signal classification algorithms for wideband cognitive radios,” *IEEE Transactions on Wireless Communications*, 2013, [In review].
- [91] C. Clancy, J. Hecker, E. Stuntebeck, and T. O’Shea, “Applications of machine learning to cognitive radio networks,” *IEEE Wireless Communications*, vol. 14, no. 4, pp. 47–52, Aug. 2007.

## References

- [92] A. N. Mody, S. R. Blatt, N. B. Thammakhoue, T. P. McElwain, J. D. Niedzwiecki, D. G. Mills, M. J. Sherman, and C. S. Myers, “Machine learning based cognitive communications in white as well as the gray space,” in *IEEE Military Communications Conference. (MILCOM '07)*, Orlando, FL, Oct. 2007, pp. 1–7.
- [93] M. Bkassiny, Y. Li, and S. Jayaweera, “A survey on machine-learning techniques in cognitive radios,” *IEEE Communications Surveys and Tutorials*, 2012, DOI: 10.1109/SURV.2012.100412.00017.
- [94] A. He, K. K. Bae, T. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. Neel, Y. Zhao, J. Reed, and W. Tranter, “A survey of artificial intelligence for cognitive radios,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1578–1592, May 2010.
- [95] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- [96] M. Bkassiny, S. K. Jayaweera, and K. A. Avery, “Distributed reinforcement learning based mac protocols for autonomous cognitive secondary users,” in *20th Annual Wireless and Optical Communications Conference (WOCC '11)*, Newark, NJ, Apr. 2011, pp. 1–6.
- [97] J. Lunden, V. Koivunen, S. Kulkarni, and H. Poor, “Reinforcement learning based distributed multiagent sensing policy for cognitive radio networks,” in *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '11)*, Aachen, Germany, May 2011, pp. 642–646.
- [98] Y. Yao and Z. Feng, “Centralized channel and power allocation for cognitive radio networks: A q-learning solution,” in *Future Network and Mobile Summit*, Florence, Italy, June 2010, pp. 1–8.
- [99] A. Galindo-Serrano and L. Giupponi, “Distributed Q-learning for aggregated interference control in cognitive radio networks,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1823–1834, May 2010.
- [100] C. Claus and C. Boutilier, “The dynamics of reinforcement learning in cooperative multiagent systems,” in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, July 1998, pp. 746–752.
- [101] L. Busoniu, R. Babuska, and B. De Schutter, “Multi-agent reinforcement learning: A survey,” in *9th International Conference on Control, Automation, Robotics and Vision (ICARCV '06)*, Grand Hyatt, Singapore, Dec. 2006, pp. 1–6.

## References

- [102] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley and Sons, 1994.
- [103] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, “The complexity of decentralized control of Markov Decision Processes,” *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, Nov. 2002.
- [104] R. D. Smallwood and E. J. Sondik, “The Optimal Control of Partially Observable Markov Processes over a Finite Horizon,” *Operations Research*, vol. 21, no. 5, pp. 1071–1088, Sept.-Oct. 1973.
- [105] D. S. Bernstein, E. A. Hansen, and S. Zilberstein, “Bounded policy iteration for decentralized POMDPs,” in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, Edinburgh, United Kingdom, July 2005.
- [106] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, “Learning policies for partially observable environments: Scaling up,” *Readings in agents*, pp. 495–503, 1998.
- [107] C. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.
- [108] S. Maleki, A. Pandharipande, and G. Leus, “Two-stage spectrum sensing for cognitive radios,” in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP '10)*, Dallas, TX, Mar. 2010, pp. 2946–2949.
- [109] L. Luo, N. Neihart, S. Roy, and D. Allstot, “A two-stage sensing technique for dynamic spectrum access,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 3028–3037, June 2009.
- [110] D. Pados, P. Papantoni-Kazakos, D. Kazakos, and A. Koyiantis, “On-line threshold learning for Neyman-Pearson distributed detection,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 10, pp. 1519–1531, Oct. 1994.
- [111] T. Atwood, “RF channel characterization for cognitive radio using support vector machines,” Ph.D. dissertation, University of New Mexico, Nov. 2009.
- [112] 12-Bit, 1-GSPS Analog-to-Digital Converter. [Online]. Available: <http://focus.ti.com/lit/ds/symlink/ads5400.pdf>
- [113] D. Wieruch and V. Pohl, “A cognitive radio architecture based on sub-Nyquist sampling,” in *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '11)*, Aachen, Germany, May 2011, pp. 576–585.

## References

- [114] M. Patzold, *Mobile Fading Channels*, 1st ed. John Wiley & Sons, Ltd, 2002.
- [115] A. Goldsmith, *Wireless Communications*, 1st ed. New York, NY: Cambridge University Press, 2005.
- [116] C. Yeh and A. Robel, "Adaptive noise level estimation," in *International Conference on Digital Audio Effects (DAFx-06)*, Montreal, Quebec, Canada, Sep. 2006, pp. 145–148.
- [117] F. Millioz and N. Martin, "Estimation of a white gaussian noise in the short time fourier transform based on the spectral kurtosis of the minimal statistics: Application to underwater noise," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP '10)*, Dallas, TX, Mar. 2010, pp. 5638 –5641.
- [118] A. Sonnenschein and P. Fishman, "Radiometric detection of spread-spectrum signals in noise of uncertain power," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 654 –660, July 1992.
- [119] B. Shent, L. Huang, C. Zhao, Z. Zhou, and K. Kwak, "Energy detection based spectrum sensing for cognitive radios in noise of uncertain power," in *International Symposium on Communications and Information Technologies (ISCIT '08)*, Vientiane, Lao PDR, Oct. 2008, pp. 628 –633.
- [120] O. Setter, M. Sharir, and D. Halperin, "Constructing two-dimensional voronoi diagrams via divide-and-conquer of envelopes in space," in *Sixth International Symposium on Voronoi Diagrams (ISVD '09)*, Copenhagen, Denmark, June 2009, pp. 43 –52.
- [121] W. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a century of research," *Signal Processing*, vol. 86, no. 4, pp. 639–697, Apr. 2006. [Online]. Available: <http://dx.doi.org/10.1016/j.sigpro.2005.06.016>
- [122] J. Parsons and A. Bajwa, "Wideband characterisation of fading mobile radio channels," *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 129, no. 2, p. 95, april 1982.
- [123] D.-S. Yoo and W. Stark, "Characterization of wssus channels: normalized mean square covariance," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 4, pp. 1575 – 1584, july 2005.
- [124] P. Hoehner, "A statistical discrete-time model for the wssus multipath channel," *Vehicular Technology, IEEE Transactions on*, vol. 41, no. 4, pp. 461 –468, nov 1992.

## References

- [125] K.-W. Yip and T.-S. Ng, “Discrete-time model for digital communications over a frequency-selective rician fading wssus channel,” *Communications, IEE Proceedings-*, vol. 143, no. 1, pp. 37–42, feb 1996.
- [126] L. Giupponi, A. Galindo-Serrano, P. Blasco, and M. Dohler, “Docitive networks: an emerging paradigm for dynamic spectrum management [dynamic spectrum management],” *IEEE Wireless Communications*, vol. 17, no. 4, pp. 47–54, Aug. 2010.
- [127] T. Costlow, “Cognitive radios will adapt to users,” *IEEE Intelligent Systems*, vol. 18, no. 3, p. 7, May-June 2003.
- [128] J. Mitola, “Cognitive radio architecture evolution,” *Proceedings of the IEEE*, vol. 97, no. 4, pp. 626–641, Apr. 2009.
- [129] X. Gao, B. Jiang, X. You, Z. Pan, Y. Xue, and E. Schulz, “Efficient channel estimation for MIMO single-carrier block transmission with dual cyclic timeslot structure,” *IEEE Transactions on Communications*, vol. 55, no. 11, pp. 2210–2223, Nov. 2007.
- [130] X. Dong, Y. Li, C. Wu, and Y. Cai, “A learner based on neural network for cognitive radio,” in *12th IEEE International Conference on Communication Technology (ICCT '10)*, Nanjing, China, Nov. 2010, pp. 893–896.
- [131] B. Hamdaoui, P. Venkatraman, and M. Guizani, “Opportunistic exploitation of bandwidth resources through reinforcement learning,” in *IEEE Global Telecommunications Conference (GLOBECOM '09)*, Honolulu, HI, Dec. 2009, pp. 1–6.
- [132] K.-L. A. Yau, P. Komisarczuk, and P. D. Teal, “Applications of reinforcement learning to cognitive radio networks,” in *IEEE International Conference on Communications Workshops (ICC), 2010*, Cape Town, South Africa, May 2010, pp. 1–6.
- [133] Y. Reddy, “Detecting primary signals for efficient utilization of spectrum using q-learning,” in *Fifth International Conference on Information Technology: New Generations (ITNG '08)*, Las Vegas, NV, Apr. 2008, pp. 360–365.
- [134] M. Li, Y. Xu, and J. Hu, “A q-learning based sensing task selection scheme for cognitive radio networks,” in *International Conference on Wireless Communications Signal Processing (WCSP '09)*, Nanjing, China, Nov. 2009, pp. 1–5.
- [135] P. Venkatraman, B. Hamdaoui, and M. Guizani, “Opportunistic bandwidth sharing through reinforcement learning,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 3148–3153, July 2010.

## References

- [136] T. Jiang, D. Grace, and P. Mitchell, “Efficient exploration in reinforcement learning-based cognitive radio spectrum sharing,” *IET Communications*, vol. 5, no. 10, pp. 1309–1317, 1 2011.
- [137] Z. Han, R. Zheng, and H. Poor, “Repeated auctions with bayesian nonparametric learning for spectrum access in cognitive radio networks,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 890–900, Mar. 2011.
- [138] G. D. Croon, M. F. V. Dartel, and E. O. Postma, “Evolutionary learning outperforms reinforcement learning on non-markovian tasks,” in *Workshop on Memory and Learning Mechanisms in Autonomous Robots, 8th European Conference on Artificial Life*, Canterbury, Kent, UK, 2005.
- [139] R. Sutton, D. Mcallester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Proceedings of the 12th conference on Advances in Neural Information Processing Systems (NIPS ’99)*. Denver, CO: MIT Press, 2001, pp. 1057–1063.
- [140] J. Baxter and P. L. Bartlett, “Infinite-horizon policy-gradient estimation,” *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, 2001.
- [141] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, “Evolutionary algorithms for reinforcement learning,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 241–276, 1999.
- [142] F. Dandurand and T. Shultz, “Connectionist models of reinforcement, imitation, and instruction in learning to solve complex problems,” *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, pp. 110–121, Aug. 2009.
- [143] Y. Xing and R. Chandramouli, “Human behavior inspired cognitive radio network design,” *IEEE Communications Magazine*, vol. 46, no. 12, pp. 122–127, Dec. 2008.
- [144] M. van der Schaar and F. Fu, “Spectrum access games and strategic learning in cognitive radio networks for delay-critical applications,” *Proceedings of the IEEE*, vol. 97, no. 4, pp. 720–740, Apr. 2009.
- [145] B. Wang, K. Ray Liu, and T. Clancy, “Evolutionary cooperative spectrum sensing game: how to collaborate?” *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 890–900, Mar. 2010.
- [146] A. Galindo-Serrano, L. Giupponi, P. Blasco, and M. Dohler, “Learning from experts in cognitive radio networks: The docitive paradigm,” in *Proceedings*

## References

- of the Fifth International Conference on Cognitive Radio Oriented Wireless Networks Communications (CROWNCOM '10)*, Cannes, France, June 2010, pp. 1–6.
- [147] R. S. Michalski, “Learning and cognition,” in *World Conference on the Fundamentals of Artificial Intelligence (WOFAI '95)*, Paris, France, July 1995, pp. 507–510.
- [148] J. Burbank, A. Hammons, and S. Jones, “A common lexicon and design issues surrounding cognitive radio networks operating in the presence of jamming,” in *IEEE Military Communications Conference (MILCOM '08)*, San Diego, CA, Nov. 2008, pp. 1–7.
- [149] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [150] V. Tumuluru, P. Wang, and D. Niyato, “A neural network based spectrum prediction scheme for cognitive radio,” in *Communications (ICC), 2010 IEEE International Conference on*, may 2010, pp. 1–5.
- [151] H. Hu, J. Song, and Y. Wang, “Signal classification based on spectral correlation analysis and svm in cognitive radio,” in *22nd International Conference on Advanced Information Networking and Applications (AINA '08)*, Okinawa, Japan, Mar. 2008, pp. 883–887.
- [152] Q. Zhao, L. Tong, A. Swami, and Y. Chen, “Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 589–600, Apr. 2007.
- [153] J. Unnikrishnan and V. V. Veeravalli, “Algorithms for dynamic spectrum access with learning for cognitive radio,” *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 750–760, Feb. 2010.
- [154] Q. Zhao, L. Tong, and A. Swami, “Decentralized cognitive mac for dynamic spectrum access,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, Nov. 2005, pp. 224–232.
- [155] Y. Li, S. Jayaweera, M. Bkassiny, and K. Avery, “Optimal myopic sensing and dynamic spectrum access in cognitive radio networks with low-complexity implementations,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2412–2423, July 2012.

## References

- [156] S. K. Jayaweera and C. Mosquera, “A dynamic spectrum leasing (dsl) framework for spectrum sharing in cognitive radio networks,” in *43rd Annual Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2009.
- [157] K. Hakim, S. K. Jayaweera, G. El-Howayek, and C. Mosquera, “Efficient dynamic spectrum sharing in cognitive radio networks: Centralized dynamic spectrum leasing (C-DSL),” *IEEE Trans. Wireless Communications*, June 2010, accepted for publication.
- [158] Y. Li, S. K. Jayaweera, M. Bkassiny, and K. A. Avery, “Optimal myopic sensing and dynamic spectrum access with low-complexity implementations,” in *IEEE Vehicular Technology Conference (VTC-spring '11)*, Budapest, Hungary, May 2011.
- [159] M. Bkassiny, S. K. Jayaweera, Y. Li, and K. A. Avery, “Optimal and low-complexity algorithms for dynamic spectrum access in centralized cognitive radio networks with fading channels,” in *IEEE Vehicular Technology Conference (VTC-spring '11)*, Budapest, Hungary, May 2011.
- [160] B. Latifa, Z. Gao, and S. Liu, “No-regret learning for simultaneous power control and channel allocation in cognitive radio networks,” in *Computing, Communications and Applications Conference (ComComAp '12)*, Hong Kong, China, Jan. 2012, pp. 267–271.
- [161] Z. Han, C. Pandana, and K. Liu, “Distributive opportunistic spectrum access for cognitive radio using correlated equilibrium and no-regret learning,” in *IEEE Wireless Communications and Networking Conference (WCNC '07)*, Hong Kong, China, Mar. 2007, pp. 11–15.
- [162] Q. Zhu, Z. Han, and T. Basar, “No-regret learning in collaborative spectrum sensing with malicious nodes,” in *IEEE International Conference on Communications (ICC '10)*, Cape Town, South Africa, May 2010, pp. 1–6.
- [163] K. Akkarajitsakul, E. Hossain, D. Niyato, and D. I. Kim, “Game theoretic approaches for multiple access in wireless networks: A survey,” *IEEE Communications Surveys Tutorials*, vol. 13, no. 3, pp. 372–395, quarter 2011.
- [164] M. D. Escobar, “Estimating normal means with a Dirichlet process prior,” *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 268–277, Mar. 1994. [Online]. Available: <http://www.jstor.org/stable/2291223>
- [165] E. L. Thorndike, “Animal intelligence: An experimental study of the associative processes in animals,” Ph.D. dissertation, Columbia University, 1898.

## References

- [166] ———, *Animal Intelligence*. Hafner, Darien, CT, 1911.
- [167] A. L. Samuel, “Some studies in machine learning using the game checkers,” *IBM Journal on Research and Development*, vol. 3, pp. 211–229, 1959.
- [168] C. Watkins, “Learning from delayed rewards,” Ph.D. dissertation, University of Cambridge, United Kingdom, 1989.
- [169] H. Li, “Multi-agent q-learning of channel selection in multi-user cognitive radio systems: A two by two case,” in *IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, San Antonio, TX, Oct. 2009, pp. 1893–1898.
- [170] J. Peters and S. Schaal, “Policy gradient methods for robotics,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (2006)*, Beijing, China, Oct. 2006, pp. 2219–2225.
- [171] M. Riedmiller, J. Peters, and S. Schaal, “Evaluation of policy gradient methods and variants on the cart-pole benchmark,” in *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL '07)*, Honolulu, HI, Apr. 2007, pp. 254–261.
- [172] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.
- [173] P. Zhou, W. Yuan, W. Liu, and W. Cheng, “Joint power and rate control in cognitive radio networks: A game-theoretical approach,” in *Proc. IEEE International Conference on Communications (ICC'08)*, May 2008, pp. 3296–3301.
- [174] A. R. Fattahi, F. Fu, M. V. D. Schaar, and F. Paganini, “Mechanism-based resource allocation for multimedia transmission over spectrum agile wireless networks,” *IEEE Journ. Select Areas Commun.*, vol. 3, no. 25, pp. 601–612, Apr. 2007.
- [175] O. Ileri, D. Samardzija, and N. B. Mandayam, “Demand responsive pricing and competitive spectrum allocation via a spectrum server,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, Nov. 2005, pp. 194–202.
- [176] Y. Zhao, S. Mao, J. Neel, and J. Reed, “Performance evaluation of cognitive radios: Metrics, utility functions, and methodology,” *Proceedings of the IEEE*, vol. 97, no. 4, pp. 642–659, April 2009.
- [177] J. Neel, R. M. Buehrer, B. H. reed, and R. P. Gilles, “Game theoretic analysis of a network of cognitive radio,” in *45th Midwest Symp. on Circuits and Systems*, vol. 3, Aug. 2002, pp. III–409–III–412.

## References

- [178] M. R. Musku and P. Cotae, “Cognitive radio: Time domain spectrum allocation using game theory,” in *IEEE Int. Conf. on System and Systems Engineering (SoSE)*, April 2007, pp. 1–6.
- [179] W. Wang, Y. Cui, T. Peng, and W. Wang, “Noncooperative power control game with exponential pricing for cognitive radio network,” in *IEEE 65th Vehicular Technology Conf. (VTC)-Spring*, April 2007, pp. 3125–3129.
- [180] Z. Ji and K. J. R. Liu, “Cognitive radios for dynamic spectrum access- dynamic spectrum sharing: A game theoretical overview,” *IEEE Communications Magazine*, vol. 45, no. 5, pp. 88–94, May 2007.
- [181] N. Nie and C. Comaniciu, “Adaptive channel allocation spectrum etiquette for cognitive radio networks,” in *1st IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Nov. 2005, pp. 269–278.
- [182] R. G. Wendorf and H. Blum, “A channel-change game for multiple interfering cognitive wireless networks,” in *Military Communi. Conf. (MILCOM)*, Oct. 2006, pp. 1–7.
- [183] V. Krishnamurthy, “Decentralized spectrum access amongst cognitive agents - An interacting multivariate global games approach,” Nov. 2008, accepted.
- [184] Z. Qu, Z. Qin, J. Wang, L. Luo, and Z. Wei, “A cooperative game theory approach to resource allocation in cognitive radio networks,” in *The 2nd IEEE International Conference on Information Management and Engineering (ICIME '10)*, 2010, pp. 90–93.
- [185] J. Li, D. Chen, W. Li, and J. Ma, “Multiuser power and channel allocation algorithm in cognitive radio,” in *International Conference on Parallel Processing (ICPP '07)*, XiAn, China, Sep. 2007, p. 72.
- [186] X. Zhang and J. Zhao, “Power control based on the asynchronous distributed pricing algorithm in cognitive radios,” in *IEEE Youth Conference on Information Computing and Telecommunications (YC-ICT '10)*, Beijing, China, Nov. 2010, pp. 69–72.
- [187] L. Pillutla and V. Krishnamurthy, “Game theoretic rate adaptation for spectrum-overlay cognitive radio networks,” in *IEEE Global Telecommunications Conference (GLOBECOM '08)*, New Orleans, LA, Dec. 2008, pp. 1–5.
- [188] H. Li, Y. Liu, and D. Zhang, “Dynamic spectrum access for cognitive radio systems with repeated games,” in *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS '10)*, Beijing, China, June 2010, pp. 59–62.

## References

- [189] O. Simeone, I. Stanojev, S. Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz, "Spectrum leasing to cooperating secondary ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 203–213, Jan. 2008.
- [190] S. K. Jayaweera and M. Bkassiny, "Learning to thrive in a leasing market: an auctioning framework for distributed dynamic spectrum leasing (d-dsl)," in *IEEE Wireless Communications & Networking Conference (WCNC'2011)*, Cancun, Mexico, Mar. 2011, accepted.
- [191] L. Chen, S. Iellamo, M. Coupechoux, and P. Godlewski, "An auction framework for spectrum allocation with interference constraint in cognitive radio networks," in *IEEE INFOCOM '10*, San Diego, CA, Mar. 2010, pp. 1–9.
- [192] G. Iosifidis and I. Koutsopoulos, "Challenges in auction theory driven spectrum management," *Communications Magazine, IEEE*, vol. 49, no. 8, pp. 128–135, august 2011.
- [193] F. Fu and M. van der Schaar, "Stochastic game formulation for cognitive radio networks," in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '08)*, Chicago, IL, Oct. 2008, pp. 1–5.
- [194] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1380–1391, Apr. 2012.
- [195] S. Gong, W. Liu, W. Yuan, W. Cheng, and S. Wang, "Threshold-learning in local spectrum sensing of cognitive radio," in *IEEE 69th Vehicular Technology Conference (VTC Sp. '09)*, Barcelona, Spain, Apr. 2009, pp. 1–6.
- [196] T. Ferguson, "A bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.
- [197] D. Blackwell and J. B. MacQueen, "Ferguson distributions via polya urn schemes," *The Annals of Statistics*, vol. 1, no. 2, pp. 353–355, 1973. [Online]. Available: <http://www.jstor.org/stable/2958020>
- [198] M. Jordan. (2005) Dirichlet processes, Chinese restaurant processes and all that. [Online]. Available: <http://www.cs.berkeley.edu/~jordan/nips-tutorial05.ps>
- [199] M. DeGroot, *Optimal Statistical Decisions*. McGraw-Hill, 1970.
- [200] H. W. Sorenson, *Parameter Estimation: Principles and Problems*. Marcel Dekker, 1980.

## References

- [201] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721 –741, Nov. 1984.
- [202] S. S. Haykin, *Neural networks : A Comprehensive Foundation*, 2nd ed. Prentice Hall, Jul. 1999.
- [203] N. Baldo and M. Zorzi, “Learning and adaptation in cognitive radios using neural networks,” in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, jan. 2008, pp. 998 –1003.
- [204] N. Baldo, B. Tamma, B. Manojt, R. Rao, and M. Zorzi, “A neural network based cognitive controller for dynamic channel selection,” in *Communications, 2009. ICC '09. IEEE International Conference on*, june 2009, pp. 1 –5.
- [205] Y.-J. Tang, Q.-Y. Zhang, and W. Lin, “Artificial neural network based spectrum sensing method for cognitive radio,” in *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, sept. 2010, pp. 1 –4.
- [206] M. I. Taj and M. Akil, “Cognitive radio spectrum evolution prediction using artificial neural networks based multivariate time series modeling,” *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*, pp. 1 –6, april 2011.
- [207] J. Popoola and R. van Olst, “A novel modulation-sensing method,” *Vehicular Technology Magazine, IEEE*, vol. 6, no. 3, pp. 60 –69, sept. 2011.
- [208] M. Han, J. Xi, S. Xu, and F.-L. Yin, “Prediction of chaotic time series based on the recurrent predictor neural network,” *Signal Processing, IEEE Transactions on*, vol. 52, no. 12, pp. 3409 – 3416, dec. 2004.
- [209] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [210] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152. [Online]. Available: <http://doi.acm.org/10.1145/130385.130401>
- [211] M. Martinez-Ramon and C. G. Christodoulou, *Support Vector Machines for Antenna Array Processing and Electromagnetics*, 1st ed., C. A. Balanis, Ed. USA: Morgan and Claypool Publishers, 2006.

## References

- [212] G. Xu and Y. Lu, "Channel and modulation selection based on support vector machines for cognitive radio," in *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006. International Conference on*, sept. 2006, pp. 1–4.
- [213] L. Hai-Yuan and J.-C. Sun, "A modulation type recognition method using wavelet support vector machines," in *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, oct. 2009, pp. 1–4.
- [214] Z. Yang, Y.-D. Yao, S. Chen, H. He, and D. Zheng, "Mac protocol classification in a cognitive radio network," in *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*, may 2010, pp. 1–5.
- [215] M. Petrova, P. Ma andho andnen, and A. Osuna, "Multi-class classification of analog and digital signals in cognitive radios using support vector machines," in *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, sept. 2010, pp. 986–990.
- [216] D. Zhang and X. Zhai, "Svm-based spectrum sensing in cognitive radio," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, sept. 2011, pp. 1–4.
- [217] T. D. Atwood, M. Martnez-Ramon, and C. G. Christodoulou, "Robust support vector machine spectrum estimation in cognitive radio," in *Proceedings of the 2009 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting*, 2009.
- [218] Z. Sun, G. Bradford, and J. Laneman, "Sequence detection algorithms for phy-layer sensing in dynamic spectrum access networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 1, pp. 97–109, feb. 2011.
- [219] D. Cabric, "Addressing feasibility of cognitive radios," *Signal Processing Magazine, IEEE*, vol. 25, no. 6, pp. 85–93, november 2008.
- [220] Z. Han, R. Fan, and H. Jiang, "Replacement of spectrum sensing in cognitive radio," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 6, pp. 2819–2826, june 2009.
- [221] S. Jha, U. Phuyal, M. Rashid, and V. Bhargava, "Design of omc-mac: An opportunistic multi-channel mac with qos provisioning for distributed cognitive radio networks," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 10, pp. 3414–3425, october 2011.

## References

- [222] B. Wang and K. J. R. Liu, “Advances in cognitive radio networks: A survey,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 5–23, Feb. 2011.
- [223] B. Wang, K. Liu, and T. Clancy, “Evolutionary game framework for behavior dynamics in cooperative spectrum sensing,” in *IEEE Global Telecommunications Conference (IEEE GLOBECOM '08)*, Dec. 2008, pp. 1–5.
- [224] E. C. Y. Peh, Y.-C. Liang, Y. L. Guan, and Y. Zeng, “Power control in cognitive radios under cooperative and non-cooperative spectrum sensing,” *Wireless Communications, IEEE Transactions on*, vol. 10, no. 12, pp. 4238–4248, december 2011.
- [225] M. Haddad, S. Elayoubi, E. Altman, and Z. Altman, “A hybrid approach for radio resource management in heterogeneous cognitive networks,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 831–842, Apr. 2011.
- [226] N. Shetty, S. Pollin, and P. Pawelczak, “Identifying spectrum usage by unknown systems using experiments in machine learning,” in *IEEE Wireless Communications and Networking Conference (WCNC '09)*, Budapest, Hungary, Apr. 2009, pp. 1–6.
- [227] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. Chapman & Hall/CRC, 2003.
- [228] D. Fink, “A compendium of conjugate priors,” Tech. Rep., 1997.
- [229] M. West, “Hyperparameter estimation in dirichlet process mixture models,” Duke University, Tech. Rep., 1992.
- [230] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer, 1998.
- [231] Q. Xiao, Q. Gao, L. Xiao, S. Zhou, and J. Wang, “An optimal opportunistic spectrum access approach,” in *IEEE International Conference on Communications Workshops, 2009*, Dresden, Germany, 14-18 2009, pp. 1–5.
- [232] H. Liu, B. Krishnamachari, and Q. Zhao, “Cooperation and learning in multiuser opportunistic spectrum access,” in *IEEE International Conference on Communications Workshops, 2008*, Beijing, China, May. 2008, pp. 487–492.
- [233] Y. Zhang, L. Zhang, and C. Tang, “Joint detection of cyclostationary and energy in cognitive radio,” in *International Conference on Intelligent Systems*

## References

- and Knowledge Engineering (ISKE '10)*, Hangzhou, China, Nov. 2010, pp. 182–186.
- [234] Y. Teng, Y. Zhang, F. Niu, C. Dai, and M. Song, “Reinforcement learning based auction algorithm for dynamic spectrum access in cognitive radio networks,” in *IEEE 72nd Vehicular Technology Conference Fall (VTC '10-Fall)*, Ottawa, ON, Sep. 2010, pp. 1–5.
- [235] F. F. Digham, M.-S. Alouini, and M. Simon, “On the energy detection of unknown signals over fading channels,” in *IEEE International Conference on Communications (ICC '03)*, vol. 5, Anchorage, AK, May 2003, pp. 3575–3579 vol.5.
- [236] S. A. Kassam, *Signal Detection in Non-Gaussian Noise*. Springer-Verlag, 1988.
- [237] Z. Tian, Y. Tafesse, and B. Sadler, “Cyclic feature detection with sub-nyquist sampling for wideband spectrum sensing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 1, pp. 58–69, Feb. 2012.
- [238] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *In Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, NJ, Jul. 1994, pp. 157–163.
- [239] H. Li, “Multi-agent q-learning for competitive spectrum access in cognitive radio systems,” in *Fifth IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks*, Boston, MA, Jun. 2010, pp. 1–6.
- [240] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer, 1998, ch. 3, pp. 72–76.