APPLICATION OF COMPRESSION SENSING AND BELIEF PROPAGATION FOR CHANNEL OCCUPANCY DETECTION IN COGNITIVE RADIO NETWORKS

by

Sadiq Jafar Sadiq

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

Copyright © 2011 by Sadiq Jafar Sadiq
Abstract

Application of Compressive Sensing and Belief Propagation for Channel Occupancy Detection in Cognitive Radio Networks

Sadiq Jafar Sadiq
Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2011

Wide-band spectrum sensing is an approach for finding spectrum holes within a wide-band signal with less complexity/delay than the conventional approaches. In this thesis, we propose four different algorithms for detecting the holes in a wide-band radio spectrum and finding the sparsity level of compressive signals. The first algorithm estimates the spectrum in an efficient manner and uses this estimation to find the holes. This approach adds a new dimension to the scenario through ignoring specific portions of the time samples. The second algorithm detectes the spectrum holes by reconstructing channel energies instead of reconstructing the spectrum itself. In this method, the signal is fed into a number of filters, less than the number of available channels. The energies of the filter outputs are then used as the compressed measurement to reconstruct the signal energy in each channel. The third algorithm employes two information theoretic algorithms (MDL and PDL) to find the sparsity level of a compressive signal and the last algorithm employs belief propagation for detecting the sparsity level. The performance of these algorithms is investigated through simulations.
Dedication

To my beloved parents and my sweet wife.
Acknowledgements

I would like to express my sincere thanks to my supervisor, Professor Shahrokh Valae, whose guidance and support made this work possible. His knowledge and encouragement has invaluable help in producing this work. I would also like to thank Veria Havary-Nassab, who I have been working with and who has supported me in this project.
## Contents

1 Introduction

1.1 Motivation .................................................. 1
1.2 Advantages to Applying Compressive Sensing in Spectrum Sensing ........ 4
1.3 Scope and Objectives ........................................ 5
1.4 Contributions ................................................ 7

2 Background

2.1 Compressive Sensing Theory ................................ 9
2.2 Spectrum Sensing in Cognitive Radio Networks .................. 11
  2.2.1 Techniques of Spectrum Sensing .......................... 12
  2.2.2 Cooperation in Spectrum Sensing ......................... 15
2.3 Wide-Band Spectrum Sensing in Cognitive Radio Networks ............. 16
  2.3.1 Wide-Band Spectrum Sensing Techniques .................. 16
  2.3.2 Compressive Sensing For Wide-Band Cognitive Radios ........ 17
2.4 Signal Enumeration Via Information Theoretic Criteria .................. 19
  2.4.1 Related Work ............................................. 21
2.5 Belief Propagation ............................................. 22
2.6 Compressive Sensing Via Belief Propagation ........................ 24
  2.6.1 The two-state mixture Gaussian model ..................... 25
  2.6.2 Spectrum Sensing Via Belief Propagation .................. 26
Chapter Summary .......................................................... 26

3 Compressive Detection For Wide-Band Spectrum Sensing 28

3.1 Problem Statement .................................................. 29

3.2 Compressive Wide-Band Spectrum Sensing With Reduced Delay and Complexity ........................................... 29

3.2.1 Compressive Wide-Band Spectrum Sensing With Smaller Time Aperture ......................................................... 30

3.2.2 Cognitive Radio with Bayesian Energy Detector ............. 31

3.2.3 Compressive Spectrum Sensing With Smaller Sampling Rate .... 32

3.3 Compressive Detection For Wide-Band Spectrum Sensing ......... 33

3.3.1 System Model .................................................... 33

3.3.2 Compressive Detection ........................................... 35

3.3.3 Cooperative Spectrum Sensing in Ad-hoc Networks .......... 37

3.3.4 Advantages over Current Algorithms ............................ 39

3.4 Chapter Summary .................................................... 40

4 Sparsity Level Detection 42

4.1 Sparsity Level Detection For Wide-Band Spectrum Sensing Via Information Theoretic Algorithms .......................... 43

4.1.1 System Model .................................................... 43

4.1.2 Estimating The Number of Occupied Channels ................ 47

4.2 Sparsity Level Detection Via Belief Propagation .................. 50

4.2.1 System Model .................................................... 51

4.2.2 Sparsity Level Detection Algorithm ............................... 52

4.2.3 Advantages Over Current Algorithms ............................ 55

4.2.4 Channel Occupancy Detection in CR Networks ................ 58

4.3 Chapter Summary .................................................... 59
5 Simulation Results

5.1 Compressive Wide-Band Spectrum Sensing with Reduced Delay and Complexity 60
5.2 Compressive Detection For Wide-Band Spectrum Sensing 65
5.3 Sparsity Level Detection Via Information Theoretic Algorithms 69
5.4 Sparsity Level Detection Via Belief Propagation 72
5.5 Chapter Summary 77

6 Conclusions and Future Work 80

6.1 Future Work 81

Bibliography 82
List of Tables

5.1 Number of detected channels by MDL and PDL for 100 independent runs 68
List of Figures

1.1 Radio Spectrum Utilization Results Over 6 Locations [1] .......................... 2

2.1 Sampling with traditional method and compressive sensing ....................... 10
2.2 The factor graph for the global function factorization in (2.15) .................... 23
2.3 The two-state mixture Gaussian model for \( x \) ........................................ 25

3.1 The hidden terminal problem in ad-hoc networks ..................................... 36

4.1 The block diagram of the system ................................................................. 44
4.2 The factor graph for the proposed algorithm .............................................. 53
4.3 The two-state mixture Gaussian model for cognitive radio ........................... 58

5.1 Amplitude of the spectrum of the noisy signal before compression ............. 61
5.2 Reconstructed spectrum with \( N/R = 50\%, S/R = 50\% \) ............................. 62
5.3 Reconstructed spectrum with \( N/R = 50\%, S/R = 30\% \) ............................. 63
5.4 Reconstructed spectrum with \( N/R = 50\% \) and a sampling rate of 2/3 of the Nyquist rate ................................................................. 64
5.5 Probability of error for different values of \( N \) and \( S \) ................................. 65
5.6 PDF of energy in an occupied and an unoccupied channel for different number of filters \( N \) ................................................................. 66
5.7 Probability of detection error versus SNR for different number of filters .......................... 67
5.8 The eigenvalues of the sample covariance matrix averaged over 100 independent runs for \( \text{SNR} = 0 \text{ dB} \) ................................................................. 69
5.9 The probability of detection for MDL and PDL methods over 100 independent runs .................................................. 70
5.10 The detected number of occupied channels as a function of the observation window. The number of signals changes from 4 to 5 at \( q = 40 \) ........................................... 71
5.11 The distribution of model variable \( m \) over 30 iterations. \( M = 20, K = 4, N = 20 \) and SNR = 0 dB .................................................. 73
5.12 The probability of detection for different number of measurements and iterations. \( M = 20, K = 4 \) and SNR = 0 dB. ........................................... 74
5.13 The probability of detection for different SNR for different number of iterations, observations and information theoretic methods. \( N = 20 \). ........................................... 75
5.14 The threshold SNR over different number of collected observations for BP-algorithm with 20 iterations, MDL and PDL. ........................................... 77
5.15 The average running times for different number of iterations for BP-algorithm and PDL. \( N = M/2, \) SNR = 0 dB. ........................................... 78
5.16 The number of occupied channels have been changed from 4 to 5 in the 26th iteration. The actual and the detected models are shown. \( M = 20, N = 20 \) and SNR = 0 dB. ........................................... 79
Chapter 1

Introduction

1.1 Motivation

Spectrum division among users in current wireless communication systems is decided by regulatory and licensing bodies like the Federal Communication Commission (FCC) in the United States. The most commonly spectrum management approach used is based on a static spectrum allocation model known as Command and Control. In this model, the radio spectrum is divided into fixed and non-overlapping blocks, separated by guard bands, and these blocks are assigned to different services and wireless technologies. After that, these blocks are licensed for exclusive use to carriers, radio and TV broadcasters, mobile operators, etc. In the license, the amount of radiated power and out-of-band emission which is interference to neighboring frequencies within each band are rigidly defined. Although this approach provides the users with protection against interference, it also provides a limited support for coexistence capabilities.

One study done by FCC Spectrum Policy Task Force showed a significant variations in the spectrum utilization in different times and locations, ranging from 15 – 85% in the bands below 3GHz, and greatly lower utilization at frequencies above 3GHz [2].

In [1], another study shows spectrum utilization in the frequency bands between
Chapter 1. Introduction

30MHz and 3GHz averaged over six different locations, as illustrated in Fig. 1.1. While some frequency bands are overcrowded, other bands are rarely used. Hence, the low utilization of the licensed spectrum is significantly due to inefficient static frequency allocations, not because of the shortage of radio spectrum.

The rapid growth in the number of wireless applications and services makes efficient spectrum utilization a necessity. Cognitive radio (CR) promises to increase the utilization of frequency bands that are under-utilized by providing opportunistic spectrum access. In a CR system, we have 2 types of users: primary users which are the licensed ones and secondary users which have no license but can access the spectrum opportunistically.
whenever there is a space. These empty spaces are known as spectrum holes. Thus, a spectrum hole is a band of frequencies assigned to a primary user, but, at a particular time and a specific location, the band is not being utilized by that user. Spectrum utilization can be improved significantly by making it possible for a secondary user to access a spectrum hole unoccupied by the primary user at the right location and time.

CR provides a number of benefits that would result in increased access to spectrum which means higher utilization of spectrum. Moreover, it allows new and improved communication services to be available. Some of the capabilities that CR can provide are the following

1. Frequency Agility: the ability of a radio to change its operating frequency to provide optimal operation.

2. Dynamic Frequency Selection: the ability to sense signals from other neighboring transmitters to choose the optimal operating environment.

3. Adaptive Modulation: the ability to modify transmission characteristics and waveforms to get the opportunity to use the spectrum.

4. Transmit Power Control: to allow transmission at high power limits when necessary, but constrain the transmission power to a lower level to have a higher level of sharing of the spectrum when higher power is not required.

5. Location Awareness: the ability for a device to determine its location as well as the location of other transmitters, and determine whether it is allowed to transmit at all, then to select the operating parameters such as the power and frequency allowed at its location.

Spectrum sensing plays an essential role in cognitive radio since secondary users need to detect primary signals in order to make decisions about the occupancy of the spectrum bands. The conventional approach to search for available bandwidth in a wide spectrum
band is to scan the band channel-by-channel to find unoccupied channels. To do this, we need to provide an RF front-end with a bank of tunable and narrow bandpass filters. Each filter will then measure the energy or examine the features of the filtered signal to determine if a particular band is occupied or not. Such an approach is however highly complicated and needs numerous RF components. Furthermore, such method introduces large latency to the spectrum sensing process.

1.2 Advantages to Applying Compressive Sensing in Spectrum Sensing

Compressive sensing is a method to recover sparse signals from far fewer measurements than needed by the traditional sampling. The under-utilization in most of the assigned spectrum bands results in signals that are sparse in frequency domain. Such sparsity has motivated the use of compressive sensing in reconstructing the frequency representation of the signal with far-less time samples than that the Nyquist theorem imposes. Using wide-band spectrum sensing techniques, CR nodes can scan the whole spectrum at once and avoid the delay and complexity of channel-by-channel scanning. As an example, consider a spectrum of 1GHz with channels of 5MHz each. Scanning channel by channel requires \(\frac{1GHz}{5MHz} = 200\) scans. However, with wide-band spectrum sensing, assuming a 100MHz wide-band filter, we require \(\frac{1GHz}{100MHz} = 10\) scans only. Furthermore, assuming a low communication load in the network, compressive sensing can be exploited to take advantage of the sparsity in the spectrum occupancy pattern.

Compressive sensing reduces the sampling by using a smaller number of linear combinations or measurements. Compressive sensing gets its importance since in some environments, the channel occupancy changes rapidly. Hence, fast methods for spectrum sensing are required.

An important issue in compressive sensing is to find the optimum number of mea-
surements needed for successful reconstruction of the signal. This requires the detection of the sparsity level (i.e. the number of non-zero elements) of the sparse signal. Traditionally, when the range of the spectrum to be observed and the number of primary users active in this range are known, statistics and measurements can determine an upper bound on the level of sparsity. On the other hand, if such a priori knowledge is not accessible, signal enumeration techniques can be used to determine the level of sparsity of the problem.

1.3 Scope and Objectives

The work presented in this thesis has two main goals. The first one is to find efficient methods for detecting the holes in a wide-band radio spectrum by taking advantage of the compressive sensing theory. The second goal is to find the sparsity level of compressive signals which optimizes the performance of the compressive sensing algorithm.

To achieve the first goal, we propose two different algorithms. The first algorithm estimates the spectrum in an efficient manner and uses this estimation to find the holes. On the other hand, the second algorithm detects the spectrum holes by reconstructing channel energies instead of reconstructing the spectrum itself.

For the second goal, we also propose two different algorithms. The first algorithm employs two information theoretic algorithms, namely minimum description length (MDL) and predictive description length (PDL) while belief propagation is adopted in the second algorithm.

In Chapter 2, the relevant background material is presented. The basics of compressive sensing theory are discussed. Next, a review of spectrum sensing in cognitive radio networks is presented. Traditional techniques as well as techniques based on Wide-band sensing and compressive sensing are studied. A review of two common information theoretic techniques that were proposed in the literature for signal enumeration is pro-
vided. The belief propagation algorithm in addition to some of the methods that use this algorithm to solve the compressive sensing problem are also reviewed.

In Chapter 3, a new technique based on compressive sensing for estimating the radio spectrum is proposed that results in decreasing the time delay or the sampling rate. This approach adds a new dimension to the scenario through ignoring specific portions of the time samples taking an advantage of the compressive sensing theory. In addition, an energy detector is introduced that uses the estimated spectrum to calculate the energy of each channel and decides on the occupancy of the channel using a threshold.

Next, a novel method of compressive detection for wide-band spectrum sensing is proposed. As a cognitive radio is just interested in detecting the channel occupancy, an approach to detect the unoccupied channels without estimating the spectrum is presented. In this method, the signal is fed into a number of filters, much less than the number of channels within the wide-band spectrum. The energies of the filter outputs are used as the compressed measurement to reconstruct the signal energy in each channel. The energy vector is then compared with a threshold vector to detect the spectrum holes. Moreover, the application of cooperative spectrum sensing in ad-hod networks is considered that is used to solve the hidden-terminal problem. The advantages of the compressive detection algorithm over current algorithms in the literature are also discussed.

In Chapter 4, two techniques for sparsity level detection of compressive signals based on information theoretic algorithms and the belief propagation algorithm are proposed. First, the compressive detection algorithm proposed in Chapter 3 is completed. In compressive detection, the number of filters needed to successfully detect the unoccupied channels requires a-priori knowledge about the number of occupied ones. It is important to use the least number of filters as this lowers the delay and the complexity of the system. In this chapter, information theoretic algorithms are suggested for estimating the number of occupied channels. This estimate is then used to determine the required
number of filters.

Next, a novel technique based on belief propagation is presented to find the sparsity level of a sparse signal. The problem is formulated as a multiple hypothesis test, where several hypotheses are examined, each corresponding to a particular sparsity level, and the hypothesis that has the highest probability is selected. The proposed algorithm can detect the sparsity level with a smaller portion of the delay of conventional approaches and with a slower growing complexity. This proposed algorithm is application-independent which means it can be applied to many different applications. Moreover, the advantages of this algorithm over current algorithms are discussed. For the scope of this thesis, the application of spectrum sensing in cognitive radio networks is considered.

In Chapter 5, the simulation results are presented. the performance of the 4 algorithms proposed in Chapters 3 and 4 is evaluated through simulations. Different scenarios are considered and the ability of the algorithms presented in Chapter 3 to detect the spectrum holes is studied. Sparsity level detection performance is evaluated using the algorithms proposed in Chapter 4. The ability of the algorithms to detect the changes in the environment is also discussed.

Finally, Chapter 6 presents the concluding remarks and some topics of future work.

1.4 Contributions

The contributions of this work are summarized as follows, including the chapters presenting them, and publications referring to them:

1. Compressive wide-band spectrum sensing with reduced delay and complexity: In [3], a compressive sensing approach is used to reconstruct the spectrum of the wide-band signal using time samples. In this thesis, a new approach to the compressive wide-band spectrum sensing process is introduced that uses a smaller time aperture and/or less sampling rate for estimating the spectrum. The proposed method can
achieve better performance than the method proposed in [3] in terms of delay and complexity. (Chapter 3 - Section 3.2)

2. Compressive detection for wide-band spectrum sensing: As a cognitive radio is just interested in detecting the channel occupancy, a novel wide-band spectrum sensing scheme that uses compressive sensing for detecting the spectrum holes is proposed. In this proposed algorithm, the frequency domain representation of the signal is not reconstructed. Instead, just the vector of channel energies is obtained by solving an optimization problem which reduces the complexity of the problem. Moreover, the compressive detection method enhances the detection performance of the receiver by suppressing the noise energy in the unoccupied bands. (Chapter 3 - Section 3.3 and [4])

3. Sparsity level detection for wide-band spectrum sensing via information theoretic algorithms: A feedback loop that uses the filter output energies to estimate the number of occupied channels is proposed. Two information theoretic algorithms are used for estimating the number of occupied channels. This estimate is then used to determine the required number of filters to be adopted in the compressive detection algorithm. Optimizing the filtering process enhances the performance of the compressive detector by providing smaller delay and less complexity. (Chapter 4 - Section 4.1)

4. Sparsity level detection via belief propagation: A novel technique based on belief propagation is proposed to find the sparsity level of a sparse signal. The proposed algorithm can detect the sparsity level with a smaller portion of the delay of conventional approaches and with a slower growing complexity. This algorithm can be applied in different applications. In this thesis, it has been applied to find the number of occupied channels in CR networks. (Chapter 4 - Section 4.2)
Chapter 2

Background

In this chapter, we discuss some of the concepts that are to be used for the compressive detection and the sparsity level detection algorithms. First, we discuss the basics of compressive sensing theory which is to be used in Chapter 3 to derive the compressive detection algorithm. Next, we review some of the spectrum sensing techniques in CR networks that are proposed in the literature. We put some emphasis in the methods that apply compressive sensing for the wide-band spectrum sensing process.

In Section 2.4, we review two common information theoretic techniques that were proposed in the literature for signal enumeration. We also review the belief propagation algorithm in Section 2.5. Moreover, in Section 2.6, we study some of the methods that use belief propagation to solve the compressive sensing problem. The information theoretic techniques and the belief propagation algorithm discussed in this chapter are to be used in Chapter 4 where the sparsity level of a sparse signal is to be detected.

2.1 Compressive Sensing Theory

Compressive sensing is a method to recover signals from far fewer measurements than needed by the traditional sampling. Assume that an $M \times 1$ vector $x$ is to be measured. Also suppose that there is a basis $\Psi$ in which $x$ is sparse. Mathematically, $x$ can be
written as

$$x = \Psi s$$

(2.1)

where the $M \times 1$ vector $s$ is the representation of $x$ in the basis $\Psi$ and has just $K \ll M$ non-zero elements.

The compressive sensing theory states that $x$ can be accurately recovered from $N \ll M$ measurements of the signal [5,6]. Assume that we use a set of $N$ linear combinations of the signal as the measurement vector $y$

$$y = \Phi x.$$

(2.2)

where $\Phi$ is the $N \times M$ sensing matrix. Then by properly choosing $N$ and $\Phi$, and based on the sparsity of the representation of $x$ in the $\Psi$ basis, $x$ can be recovered from $y$.

In the conventional Nyquist sampling, $\Phi$ is the $M \times M$ identity matrix. In compressive sensing, $\Phi$ is generally a $N \times M$ matrix where $N \ll M$ as shown in Fig. 2.1.

The value of $N$ depends on $M$, $K$ and the coherence (correlation) between the sensing matrix $\Phi$ and the basis matrix $\Psi$. The coherence between $\Phi$ and $\Psi$ should be small and is given by

$$\mu(\Phi, \Psi) = \sqrt{M} \cdot \max_{1 \leq i, j \leq M} | \langle \Phi_i, \Psi_j \rangle |$$

(2.3)

In other words, the coherence measures the largest correlation between the elements of the sensing and basis matrices. Compressive sensing requires both the sparsity and
incoherent sampling, so that the signal can be recovered exactly. In compressive sensing, if the number of measurements, $N$, satisfies the following inequality

$$N \geq cK \log(M/K)$$

the reconstruction will be successful. Here, $c$ is a constant and $K$ is the sparsity level. As the basis matrix is determined by the nature of the problem, choosing a sensing matrix having a low coherence with $\Psi$ will lead to a smaller $N$. This suggests choosing $\Phi$ to be a totally random matrix.

If the above conditions apply, then the sparse vector $s$ can be recovered from the measurement vector $y$ through an $\ell_1$ norm minimization

$$\hat{s} = \arg\min_s \|s\|_1$$

Subject to $y = \Phi \Psi s$

(2.5)

where $\| \cdot \|_1$ denotes the $\ell_1$ norm of a vector.

### 2.2 Spectrum Sensing in Cognitive Radio Networks

Spectrum sensing is the task of obtaining awareness about the spectrum usage and existence of primary users in a geographical area. Given that there is no cooperation between primary users and secondary ones, spectrum availability for secondary users is determined using spectrum sensing. Thus, the secondary user monitors the spectrum and if it finds a hole then it transmits. In fact, the first fundamental cognitive task of a CR is to use spectrum sensing for determining spectral availability. The spectrum bands can be classified into three types: [7]

1. Black spaces: which are occupied by high-power local interferers.

2. Grey spaces: which are partially occupied by low-power interferers.
3. White spaces: which are free of RF interferers except for ambient noise, which is made up of natural and artificial forms of noise.

White spaces and grey spaces (with less probability) are candidates for use by secondary users.

2.2.1 Techniques of Spectrum Sensing

For the practical implementation of CR systems, it is important to have a reliable strategy for the spectrum holes detection. This requires that the physical layer of CR networks should exploit all available degrees of freedom (time, frequency and space) in order to find candidate holes. The spectrum sensing functionality in CR systems can be divided into two subtasks:

1. Occupancy sensing: to detect the spectrum occupancy in the local area and identify the unused channels.

2. Identity sensing: to distinguish between the licensed usage of spectrum and the opportunistic usage of spectrum by other CR users. Such distinction is important especially with dense CR users. Since the licensed usage of spectrum is well protected, the public white spaces are likely to be shared by multiple CR users. Identity sensing is unique for CR systems and is possibly more challenging [8]. For the scope of this thesis, occupancy sensing is to be considered.

Physical layer spectrum sensing techniques can be classified into two categories: energy-based detection and feature-based detection (matched filtering and cyclostationary).

**Energy-based detection**

Also known as radiometry, it is the most common way of spectrum sensing because of its low computational and implementation complexities. It can be performed in both
the time and frequency domains. In the time domain, a band-pass filter is normally applied to the target signal in a particular frequency region, and the energy of the signal samples is then measured [9]. In the frequency domain, the simplest energy detector is based on the fast Fourier transform (FFT) of the time domain signal [8]. The principle of energy detector is to evaluate the power spectral density (PSD) of received signals in the local area and then set thresholds, based on which white and gray spaces are defined. Some advantages of energy-based detection are that the energy detector requires no prior knowledge of other systems which are operating in band. As a result, the processing requirement is generally lower. In addition, less time is required to achieve reliable sensing. A major drawback is that it is prone to false detections and usually works poorly when the target signal is low. Another drawback is that it cannot distinguish between the spectrum usage of the primary user and that of the secondary users [8].

Feature-Based Detection

1. Matched-Filter: The feature-based sensing usually takes into account the correlation with known signal patterns. The feature-based detection assumes that a CR has the physical-layer prior knowledge of other in-band radio services. The information is then used for the reliable detection. The optimal way of the feature-based detection is matched filtering, since it maximizes the received SNR. For the matched-filtering processing, a CR should have perfect knowledge of the primary users signaling features such as the bandwidth, center frequency, modulation type and order, pulse shaping and frame format. Since CR needs receivers for all signal types, the implementation complexity of the sensing unit is impractically large [8]. Also, the time latency introduced by the matched filtering is relatively large since additional channel training and tracking are needed.

2. Cyclostationary Detection: A suboptimal method called cyclostationary feature detection is also used [8]. Modulated signals have their built-in periodicity. This
periodicity is typically introduced intentionally in the signal format so a receiver can exploit it for the parameter estimation such as carrier phase, pulse timing, or direction of arrival. This then can be used for the detection of a random signal with a particular modulation type in a background of noise and other modulated signals. A common analysis of stationary random signals is based on the autocorrelation function and PSD. On the other hand, cyclostationary signals exhibit correlation between widely separated spectral components due to spectral redundancy caused by periodicity. The distinctive feature of spectral redundancy makes signal selectivity possible. The cyclostationary feature detection has advantages due to its ability to differentiate modulated signals, interference, and noise in low SNR ratios. It can reduce the processing requirements while still maintaining a decent detection error probability.

Several metrics can be used to evaluate the performance of spectrum sensing algorithms. Some of these metrics are bandwidth, resolution, and real-time capability. Bandwidth refers to the spectrum range that is covered by the sensing CR. Resolution is the smallest spectrum step based on which the whole bandwidth range is quantized. Real-time capability is the time that it takes for a CR to reliably sense the environment and then make adaptive decisions. Due to the time variant characteristics of wireless channels, in general, the sensing latency should not exceed the coherence time of the channel. It is clear that there is a tradeoff between the three metrics. The spectrum sensing algorithm design is mainly to find an optimized tradeoff.

Analyzing the above we find that both energy-based detection and feature-based detection (matched filtering and cyclostationary) have advantages and drawbacks. While the energy-based detection is more general, the feature-based detection outperforms it in sensing reliability and sensing convergence time. The feature-based sensing is only possible when the primary user signal contains known signal patterns. Energy-based sensing and feature-based sensing exhibit similar performance at high SNR. The energy-
based sensing, however, works poorly at low SNRs, while the feature-based sensing can achieve good performance even at low SNRs. In [8], authors proposed a method that combines the two methods. Thus, CR spectrum sensing is divided into two stages: coarse sensing and fine sensing. Coarse sensing refers to the energy-based detection technique where it quickly scans the wide-band spectrum and identifies some potential spectrum holes that can be used. These potential spectrum holes are further processed in the fine sensing stage by using a feature-based detector.

2.2.2 Cooperation in Spectrum Sensing

Performing reliable spectrum sensing is a challenging task for a CR. In a wireless channel, signal fading can cause the received signal strength to be significantly lower than what is predicted by path loss models. There are two types of fading: slow fading and fast fading. Slow fading is frequency independent, and it does not cause significant fluctuations in signal strength over small changes in receiver location, whereas fast fading is frequency dependent and can vary significantly with small changes in location. Cooperation is proposed as a solution to problems that arise in spectrum sensing due to these problems [10]. So, the sensing information is shared among local cognitive users so that the spectrum sensing is performed collectively rather than individually. Cooperative sensing relies on the variability of signal strengths at various locations.

Cooperation among CRs can be centralized or distributed [11].

1. Centralized Sensing: In centralized sensing, a central unit collects sensing information from cognitive devices, identifies the available spectrum, and broadcasts this information to other CRs or directly controls the CR traffic.

2. Distributed Sensing: In distributed sensing, cognitive nodes share information among each other but they make their own decisions as to which part of the spectrum they can use. Distributed sensing is more advantageous than centralized
sensing in the sense that there is no need for a backbone infrastructure and it has less cost.

2.3 Wide-Band Spectrum Sensing in Cognitive Radio Networks

When a wide-band spectrum is assigned to a number of primary users, secondary users can search for unoccupied channels (spectrum holes) within the wide-band spectrum and communicate in that band. The traditional way for detecting holes in a wide-band spectrum is channel-by-channel scanning. In order to implement this, an RF front-end with a bank of tunable and narrow bandpass filters is needed. The occupancy of each channel can be determined by measuring the energy of the signal at the output of each filter. In this section, we review some alternative methods that have been proposed in the literature to facilitate the wide-band sensing process.

2.3.1 Wide-Band Spectrum Sensing Techniques

In [12], a wide-band spectrum sensing approach based on maximizing the aggregate opportunistic throughput has been proposed. The maximization is subject to restrictions on the interference introduced to the primary user. In this approach, the energy at the output of each narrow-band channel is compared with a threshold to decide about the presence of primary signal. The thresholds are then obtained as a solution of an optimization problem that maximizes the aggregate rate of the CR over multiple bands.

Also, a cooperative approach has been studied based on the spatially distributed radios. In this approach, CRs share the statistics of their channel occupancy in order to get a more reliable estimate and compensate the effect of deep fade in one or a subset of the radios.

In [13], maximum likelihood (ML) estimation of the signal and noise power has been
used to detect the primary signals. An iterative asymptotic ML estimate has been proposed that can be simplified to obtain an efficient least squares estimator. The performance of this approach has been studied through simulations for different number of channels and different SNRs.

### 2.3.2 Compressive Sensing For Wide-Band Cognitive Radios

Different methods have been proposed in the literature for wide-band spectrum sensing that applies compressive sensing. In [3], a compressive sensing approach is used to reconstruct the spectrum of a wide-band signal using time samples. Upon receiving the time domain signal $x(t)$, the signal is sampled in time with the Nyquist rate. Let $T$ be the time interval that the sensing system needs to analyze and decide about the occupied channels and assume that $T_0$ is the sampling period for $x(t)$. Sampling $x(t)$ results in the $R \times 1$ time sequence vector $x_t$, where $R = T/T_0$ is the number of samples within the available time interval $[0, T]$. Considering that the frequency representation of the signal $x_t$ is sparse (i.e assuming that majority of the channels are unoccupied), the frequency representation of the signal $x_t$ is the sequence $x_f$ which is obtained using the Discrete Fourier Transform (DFT) as follows

$$x_f = F_R x_t$$

(2.6)

where $F\{\cdot\}$ represents the DFT of a sequence.

The sampled vector is compressed into a vector of lower dimension using a random sensing matrix as follows

$$y = \Phi x_t$$

(2.7)

where $\Phi$ is the $N \times R$ random sensing matrix, where $N < R$. Since the signal $x_f$ is sparse, using the compressive sensing theory, $x_f$ can be recovered using $N$ measurements.
by

\[
\hat{x}_f = \arg\min_{x_f} \|x_f\|_1 \\
\text{Subject to } y = \Phi F_R^{-1} x_f
\]  

(2.8)

Using the reconstructed spectrum, authors proposed a wavelet transform approach to determine the channel edges. The performance of the edge detector algorithm, in terms of the root mean square (MSE) error in edge detection, has been evaluated through simulations for different compression ratios and different SNRs.

In [3], an analog-to-digital converter has been used to transform the analog received signal into a digital signal by sampling at the Nyquist rate. On the other hand, in [14], the received analog signal is sampled at the information rate of the signal using an analog-to-information-converter (AIC). Here, the compressive sensing is embedded in the AIC. The same \(\ell_1\) norm minimization method is used to estimate the original spectrum. Similar to [3], authors of [14] implemented a wavelet edge detector to detect the channel borders in the estimated spectrum and the detection’s performance has been evaluated in terms of the MSE through simulations. It has been shown that the MSE of [3] is smaller than that of [14] for all compression rates, but their detection performances are comparable.

Authors of [15] proposed a compressive wide-band spectrum sensing scheme where a fusion center collects the autocorrelations of the signal s of each node. A compressive sensing recovery algorithm is then used to reconstruct an estimate of the signal spectrum. This estimate is then used to make decision about the occupancy of the channels. It has been shown that such a distributed scheme outperforms the scenario where single CR decides about the spectrum holes. As one can see in [3, 14, 15], the signal needs to be sampled at the Nyquist rate and then compressed.

In [16], authors proposed a mixed-signal parallel segmented compressive sensing architecture to perform wide-band spectrum sensing. The signal is segmented and compressive sensing is applied to each segment independently, then all samples are processed together
for the reconstruction of the signal.

### 2.4 Signal Enumeration Via Information Theoretic Criteria

Various techniques for signal enumeration have been proposed in the literature. The most common techniques apply information theoretic approaches. Minimum description length (MDL) [17] and predictive description length (PDL) [18] are two common information theoretic techniques that were proposed for signal enumeration. These methods use eigenvalue decomposition and partition the observation data into two orthogonal components in signal and noise subspaces.

In many applications, the observation vector $y$ can be represented as the output of a multiple-input/multiple-output channel with an additive white Gaussian noise as follows

$$y = Hx + n$$

(2.9)

where $y$ is the $N \times 1$ observation vector, $H$ is the $N \times M$ channel matrix and $x$ is the $M \times 1$ vector of signals. Here, $M$ is the dimension of the input signal and $N$ is the dimension of the channel output.

MDL and PDL are used to calculate the codelength that represents the data. This codelength is a cost function which when minimized, gives the shortest codelength required. Here, a set of $m$ models are considered where each model corresponds to a particular number of signals. The cost function is calculated for all models $m$. $m$ is selected from a set $\Gamma = \{0, 1, \ldots, N-1\}$. Hence, the problem can be viewed as a multiple hypothesis test, where the cost for all models is calculated and the best model is then selected.

For MDL, Coding of data is performed in two steps. First, data is encoded using a uniquely decodable prefix code. Then, the parameter vector is encoded and added as a
preamble to the codeword of data. Hence, the MDL cost consists of two parts corresponding to the log-likelihood function of the observation vector and an over-parameterization term.

The MDL cost for a model of order $m$ over a window of size $Q$ is given by [19]

$$\text{MDL}_m(Q) = - \log f(Y^Q | \hat{\psi}_m^Q) + \frac{\nu_m}{2} \log Q \quad (2.10)$$

where $f(X|\psi)$ is the conditional probability density function, $Y^Q$ is the matrix of observations up to time $Q$, $\hat{\psi}_m^Q$ is the ML estimate of the parameter vector $\psi^m$ based on the $Q$ observations and $\nu_m$ is the number of free elements.

In (2.10), the first term is the log-likelihood function of the observation vectors and the second term is the over-parameterization term. The over-parameterization term in MDL represents the number of digits required to encode the parameter vector to an optimal precision.

The MDL cost is calculated for all models and the smallest is selected as follows

$$\hat{m} = \arg \min_m \text{MDL}_m(Q). \quad (2.11)$$

The PDL cost for a model of order $m$ at time instant $q$ is given by [20]

$$\text{PDL}_m(Q) = - \sum_{q=1}^{Q} \log f(y_q | \hat{\psi}_m^{q-1}) \quad (2.12)$$

where $\hat{\psi}_m^{q-1}$ is the ML estimate of $\psi^m$ using the observations up to time $(q-1)$. The best model is selected as follows

$$\hat{m} = \arg \min_m \text{PDL}_m(Q). \quad (2.13)$$

MDL can only be applied to a batch of data and it is not suitable for online detection. On the other hand, PDL uses the ML estimate of the covariance matrix of the observation data at each time instant to calculate the cost function. Hence, PDL can be used online and can be applied to non-stationary and time-varying systems.
2.4.1 Related Work

Recently, much attention has been given to information theoretic criteria [21–24]. Authors of [25] proposed a modification to the MDL method so that it could be applied to non-Gaussian signals. This modification improves the performance of the method and also allows detecting a number of signals larger than the number of sensors. However, this comes with the price of higher computation complexity.

In [26], authors proposed an enumerator that exploits eigenvectors instead of sample eigenvalues. It has been shown that the proposed method outperforms MDL and PDL at low SNR regimes when the number of collected snapshots is small but has comparable performance with MDL and PDL for large number of snapshots. Authors of [27] proposed a minimum mean-square error (MMSE)-based MDL criterion for enumerating the sources.

A reduced-rank MDL method has been proposed in [28] in which the observation data is partitioned into the signal and noise subspace components using a recursive procedure. This approach avoids the estimation of the covariance matrix of the observations and its eigenvalue decomposition. Although the proposed method is more computationally efficient than the traditional MDL methods, it has poorer performance than PDL for low SNR.

The exponentially embedded families (EEF) criterion for model order selection has been proposed in [29] and applied to source enumeration for array signal processing in [30]. This method allows the user to embed two or more pdfs into a more general family of pdfs. It has been proven that the EEF criterion performs well for closely spaced sources, at low SNR and small number of snapshots.
2.5 Belief Propagation

Belief propagation (BP) algorithm (also known as the sum-product algorithm) provides a means to calculate the marginal distributions for random variables through passing messages in factor graphs [31–33]. Suppose that a global function can be factorized into the product of several simpler local functions. In other words, the global function \( F(x_1, x_2, \ldots, x_n) \) can be written as

\[
F(x_1, x_2, \ldots, x_n) = \prod_i f_i(X_i)
\]  

(2.14)

where \( f_i(X_i) \) is a local function with \( X_i \subset \{x_1, x_2, \ldots, x_n\} \) as its arguments. Factor graphs are graphical representations used to model the factorization given in (2.14). Factor graphs consist of nodes and edges. Nodes in a factor graph can be listed in two categories. Namely, variable nodes representing independent variables, and factor nodes representing local functions. In a factor graph, edges connect a factor node \( f \) to a variable node \( x \) if and only if \( f \) is a function of \( x \). For example, the factor graph in Fig. 2.2 represents the factorization of the global function \( F(x_1, x_2, x_3, x_4) \) given by

\[
F(x_1, x_2, x_3, x_4) = f_1(x_1, x_3)f_2(x_2, x_4)f_3(x_3)f_4(x_3, x_4)
\]  

(2.15)

where the variable nodes are represented by circles and the factor nodes are represented by squares.

BP algorithm is based on exchanging messages between variable and factor nodes in a factor graph. At every iteration, each function node receives a-priori information from the variable nodes connected to it through edges. It then calculates the a-posteriori probabilities for the connected variable nodes and passes the information back to the variable nodes. Upon receiving the updated information, the variable nodes calculate their new a-priori information. Hence, two types of messages can be found in the BP algorithm. The first type is the message passing on the edge from the variable node \( x \) to
the factor node $f$ which is denoted by $\mu_{x \rightarrow f}(x)$. The second type is the message passing on the edge from the factor node $f$ to the variable node $x$ and is denoted by $\mu_{f \rightarrow x}(x)$. These messages are exchanged according to the following update rules [31].

The message from variable node $x$ to function node $f$ is updated as follows

$$
\mu_{x \rightarrow f}(x) = \prod_{u \in n(x) \setminus \{f\}} \mu_{u \rightarrow x}(x) \tag{2.16}
$$

where $n(x)$ is the set of all factor nodes connected to $x$. Equation (2.16) implies that the message sent from variable node $x$ to function node $f$ is the product of all messages received by variable node $x$ on all other edges.

The message from function node $f$ to variable node $x$ is updated as follows

$$
\mu_{f \rightarrow x}(x) = \sum_{\sim \{x\}} \left( f(X) \prod_{w \in n(f) \setminus \{x\}} \mu_{w \rightarrow f}(w) \right) \tag{2.17}
$$

where $X = n(f)$ is the set of all variable nodes connected to $f$ and $\sim \{x\}$ is the set of all variable nodes connected to $f$ excluding $x$. In other words, the updated message sent from function node $f$ to variable node $x$ is the local function $f$ multiplied by all messages received by function node $f$ on all other edges. The result is then marginalized to be a function of $x$ only.

When the BP algorithm converges, the marginal distribution $f(x)$ for variable node
$x$ can be computed by taking the product of all messages received by $x$, that is

$$f(x) = \prod_{u \in n(x)} \mu_{u \rightarrow x}(x) \quad (2.18)$$

When the BP algorithm is implemented on a factor graph which is a tree, the algorithm is guaranteed to converge, and the marginal functions calculated in (2.18) are exact after a number of iterations equal to the depth of the tree. However, in the case that the factor graph contains cycles (a cycle is a loop that starts and ends at the same node), the BP algorithm does not have a natural termination and it needs to be stopped after a sufficient number of iterations where improvements in the messages are minor. Although convergence is not guaranteed in the case of factor graphs containing cycles, numerical results have shown that the algorithm achieves near-optimal results [34, 35].

### 2.6 Compressive Sensing Via Belief Propagation

The BP algorithm has been used for compressive sensing reconstruction. Authors of [36] proposed an algorithm for compressive sensing via belief propagation. They consider a sparse encoder matrix and a BP decoder to accelerate compressive sensing encoding and decoding under the Bayesian framework. The method depends on encoding the signal using sparse $\{0, 1, -1\}$ low density parity check (LDPC)-like matrices. They compute the measurements $y = \Phi x$ using a sparse compressive sensing matrix $\Phi$ with entries restricted to $\{0, 1, -1\}$. Hence the measurements are just sums and differences of small subsets of the coefficients of $x$. The sparse measurement matrix $\Phi$ can be represented as a sparse bipartite graph.

The decoding approach is based on message passing over graphs to solve a Bayesian inference problem. Using the two-state mixture Gaussian distribution as a prior model for the signal components, an estimate of $x$ can be computed that fits the measurements and best matches the prior. However, authors of [36] assume prior knowledge of the sparsity level of the signal which in general is not known in advance.
Chapter 2. Background

2.6.1 The two-state mixture Gaussian model

The two-state mixture Gaussian model has been well studied in the literature [39, 40]. This mixture distribution is used to model the prior for the elements of a sparse signal.

Let $\mathbf{x}$ be a sparse signal of dimension $M \times 1$. Since $\mathbf{x}$ is sparse, it consists of a small number of large elements and a large number of small elements. Therefore, each element of the signal can be associated with a state variable that can be either high or low, corresponding to an element of large or small magnitude, respectively. Let the variables $l_i \in \{0, 1\}$ denote the state of each element taking the value 0 when the element has a small magnitude, and the value 1 when the element has a large magnitude. The random vector $\ell = [l_1, l_2, \ldots, l_M]$ represents the state variables of the $M$ elements. Then, the probability function of each element $p(x_i)$, is associated with the state variable $l_i$. For $l_i = 1$, a high variance zero mean Gaussian distribution is chosen, and for $l_i = 0$, a low variance zero mean Gaussian distribution is chosen.
variance zero mean Gaussian distribution is used. Hence, the conditional distributions of $x_i$, $p(x_i|l_i)$ are as follows

$$p(x_i|l_i = 1) \sim \mathcal{N}(0, \sigma_1^2)$$
$$p(x_i|l_i = 0) \sim \mathcal{N}(0, \sigma_0^2)$$ (2.19)

where $\sigma_1^2 > \sigma_0^2$. The two-state mixture Gaussian model is illustrated in Fig. 2.3.

### 2.6.2 Spectrum Sensing Via Belief Propagation

The BP algorithm has also been used for spectrum sensing in CR networks. In [41], reconstructing the spectrum occupancies for CR networks has been formulated as a matrix completion problem, and BP algorithm has been applied to solve it. A distributed algorithm based on BP to detect channel occupancy in CR networks has been proposed in [42]. The proposed method is based on the representation of the network as a factor graph where the exchanged messages are the actual packets sent by network nodes to neighbors.

### 2.7 Chapter Summary

In this chapter, we discussed some of the concepts that are to be used to derive the compressive detection and the sparsity level detection algorithms. We reviewed the basics of compressive sensing theory. Next, we reviewed some of the spectrum sensing techniques in CR networks that are proposed in the literature. In addition, two common information theoretic techniques that were proposed in the literature for signal enumeration were discussed. We also studied the belief propagation algorithm and discussed some of the methods that use belief propagation to solve the compressive sensing problem.

In Chapter 3, we propose a method to enhance the performance of the technique suggested in [3] that was discussed in Section 2.3.2. Also, we apply the compressive
sensing theory to derive the compressive detection algorithm for wide-band spectrum sensing. In Chapter 4, the information theoretic techniques and the belief propagation algorithm discussed in this chapter are used to detect the sparsity level of a sparse signal.
Chapter 3

Compressive Detection For Wide-Band Spectrum Sensing

In this chapter we propose two techniques for wide-band spectrum sensing in CR networks. First, we propose a new technique based on compressive sensing for estimating the spectrum that results in decreasing the time delay or the sampling rate. The proposed technique can achieve even better performance than the method proposed in [3]. This approach adds a new dimension to the scenario suggested in [3] through ignoring specific portions of the time samples using the freedom of choosing the random sensing matrix in compressive sensing.

Next, we propose a novel wide-band spectrum sensing scheme using compressive sensing for detecting the holes in the spectrum and we call it Compressive Detection. In this approach, the wide-band signal is fed into a number of wide-band filters and the outputs of the filters are used to reconstruct the vector of channel energies through the $\ell_1$ norm minimization. An energy detection is then performed by comparing the obtained vector to a vector of energy thresholds to decide about the occupancy of each channel.


3.1 Problem Statement

Suppose that a total spectrum of $W$ Hz is considered to be shared among different nodes in a CR network. Assume that each node needs a bandwidth of $B$ Hz for the communication with other nodes. Define $M = \frac{W}{B}$ to be the number of available channels where the $i$th channel is the frequency band $[f_i - B/2, f_i + B/2]$, $i = 1, 2, \ldots, M$. Assume that a wide-band antenna is listening to this spectrum receiving the time domain signal $x(t)$. Let $T$ be the time interval that the sensing system needs to analyze and decide about the occupied channels and assume that $T_0$ is the sampling period for $x(t)$. Sampling $x(t)$ at the input of the CR node, we obtain the time sequence vector $x_t$ as follows

$$x_t = \{x(t)\}_{t=nT_0}, \ n = 1, 2, \ldots R. \tag{3.1}$$

Here, $R = T/T_0$ denotes the number of samples within the available time interval $[0, T]$. The frequency representation of the signal $x_t$ is the sequence $x_f = \mathcal{F}\{x_t\}$ which is obtained as follows

$$x_f = F_R x_t \tag{3.2}$$

$$\{F_R\}_{i,k} = \frac{1}{\sqrt{R}} e^{j2\pi(i-1)(k-1)/(R-1)} \ i, k = 1, 2, \ldots R \tag{3.3}$$

The problem is then to efficiently find the holes in the spectrum of the received signal.

3.2 Compressive Wide-Band Spectrum Sensing With Reduced Delay and Complexity

In this section, we start by proposing the algorithm that uses less time samples to estimate the radio spectrum. Next, an energy detector is introduced that uses the estimated spectrum to calculate the energy of each channel and decides on the occupancy of the channel using a threshold. Finally, the proposed algorithm is modified to lower the sampling rate.
3.2.1 Compressive Wide-Band Spectrum Sensing With Smaller Time Aperture

As stated in Section 2.3.2, the time domain signal $x(t)$ is sampled with the Nyquist rate and then compressed with a $N \times R$ sensing matrix $\Phi$ to form the measurement vector $y$ as follows

$$y = \Phi x_t. \quad (3.4)$$

Then, $x_f$ can be recovered using the $\ell_1$-norm minimization as

$$\hat{x}_f = \text{arg min}_{x_f} \|x_f\|_1$$

Subject to $y = \Phi F^{-1} x_f \quad (3.5)$

Compressive sensing reduces the complexity by using smaller number of linear combinations or measurements. While the analysis in [3] uses a fully random matrix $\Phi$, the freedom in selecting this matrix can be creatively used to decrease the processing delay. In [3], all $R = T/T_0$ samples of the signal are collected and then compressed. The alternative way is using only a fraction of the total time interval $T$ to collect the samples. Suppose that we use the first $T_s$ seconds of the total time $T$. In other words, we decrease the time aperture and just use $S = T_s/T_0$ first samples of $x_t$ to generate $y$ where $S < R$. This can be done by selecting a matrix $\Phi$ with the first $S$ columns randomly chosen from a Gaussian distribution and the last $R - S$ columns set to be all zero, that is

$$y = \Phi x_t = [S | 0_{N \times (R-S)}] x_t = Sx_t^S \quad (3.6)$$

where $S$ is a random $N \times S$ matrix and $x_t^S$ is a vector consisting of the first $S$ elements of $x_t$. The last equality, clearly explains how this method will lead to a faster decision. Instead of waiting for $T = RT_0$ seconds, after $T_s = ST_0$ seconds (or $S$ samples) the cognitive system can start to recover $x_f$ and hence find the spectrum holes.
3.2.2 Cognitive Radio with Bayesian Energy Detector

With the introduction of the $S$ parameter, there are two different dimensions that can affect spectrum estimation. One is reducing the number of linear combinations $N$ (lower complexity), and the other is the number of time samples $S$ (less delay). Using less time samples apparently degrades the spectrum estimation performance. However, as a cognitive radio is interested in detection of energy in each channel, as far as an adequate detection performance is achievable, the accurate spectrum estimation is tolerable.

As shown in Section 2.1, if the number of measurements, $N$, satisfies the following inequality

$$N \geq cK \log \left( \frac{M}{K} \right)$$

the reconstruction will be successful. Here, $K$ denotes the sparsity level.

Therefore, based on (3.7), if we use a smaller number of time samples, spectrum reconstruction can be successful with a $N$ even lower than the threshold needed for $S = R$. However, we can not go too far in decreasing $S$, since with very low $S$, the input does not bear enough information about the wide-band signal. As a result, the signal can be reconstructed with much less complexity (small values of $N$) and less delay (small values of $S$).

Another issue in the above problem is finding the sparsity level. In other words, the number of channels that are occupied on the average determines the level of sparsity and hence the minimum of sampling size. Traditionally, when the range of the spectrum to be observed and the primary users active in this range are known, statistics and measurements can determine an upper bound on the level of sparsity and hence, a safe number of samples $N$ can be adopted. On the other hand, if such a priori knowledge is not accessible, signal enumeration techniques can be used to determine the level of sparsity of the problem. Finding this sparsity level $K$ will be discussed in Chapter 4.

In order to numerically evaluate the detection performance of a CR adopting specific
values for parameters $S$ and $N$, we introduce an energy detector that uses the estimated spectrum to calculate the energy of each channel and decides on the occupancy of the channel using a threshold. Such a detector merely adds up (integrates) the components of the reconstructed signal in each frequency band to get an estimate of the energy being received in that band:

$$\hat{E}_i = \sum_{j=f_{i-1}}^{f_i} |x_{f,j}|^2, \quad i = 1, 2, \ldots M$$  \hspace{1cm} (3.8)

where $x_{f,j}$ denotes the $j$th entry of vector $x_f$ and $|\cdot|$ denotes the absolute value of a complex number. Using (3.8) the CRs can use the vector $\hat{E} = [\hat{E}_1 \ldots \hat{E}_M]$ to decide about the channel occupancy.

One approach that the CR can adopt is to use a Bayesian detector with a priori knowledge of probabilities of channel occupancy. This knowledge can be statistically obtained and is closely related to the sparsity level of the signal. Such a detector sets a threshold on the reconstructed energy vector to minimize the Bayesian probability of error

$$P_e = P(\mathcal{H}_0)P_{Fa} + P(\mathcal{H}_1)(1 - P_D)$$  \hspace{1cm} (3.9)

where $\mathcal{H}_0$ and $\mathcal{H}_1$ denote the hypothesis of the channel being unoccupied and occupied respectively and $P_{Fa}$ and $P_D$ are the probabilities of false alarm and detection. The probability of error metric along with complexity and delay constraints associated with each application can be used to determine an optimal $(S, N)$ pair to be adopted by the CR. In Section (5.1), we have investigated the performance of the Bayesian detector for a specific wide-band signal and different values of $S$ and $N$.

### 3.2.3 Compressive Spectrum Sensing With Smaller Sampling Rate

A similar approach can be used to lower the time sampling rate. Consider a case where we choose every one out of three columns of $\Phi$ to be a zero column. In the delay decrease
paradigm, where we pad the last columns with zeros, this means that $S/R = 2/3$ and hence the delay is decreased by one third. But with distributing the zero columns in the $\Phi$ matrix, we see that only two out of three consequent samples of $x_t$ are used. In other words, we have decreased the sampling rate to $2/3$ of the Nyquist rate. This is specially of interest in wide-band sensing where the Nyquist rate can be very high and hard to achieve. Simulations show that, as far as we are interested in the detection of channel occupancy, this scenario can be adopted with minor performance loss.

3.3 Compressive Detection For Wide-Band Spectrum Sensing

In this section, we introduce the novel method of wide-band compressive channel occupancy detection. First, the channel occupancy estimator is suggested. Second, the compressive detection algorithm is introduced. In this algorithm, CR nodes estimate the signal energy of all channels compressively and decide on the occupancy of the channels. The application of the proposed method in ad-hoc networks and the advantages of the approach over current wide-band spectrum sensing algorithms are also discussed.

3.3.1 System Model

As shown in Section 3.1, the received time signal $x(t)$ is sampled to obtain the time sequence vector $x_t$. The frequency representation of the signal $x_t$ is the sequence $x_f$ given in (3.2).

Define $r = \frac{R}{M}$ as the sampling resolution of the channels which is the number of frequency samples per channel. $r$ depends on the resolution of the spectrum reconstruction. Consider $\Omega_i$ as the set of all $r$ frequencies that are sampled in the $i$th channel. Let us denote the portion of the received signal’s energy in the $i$th channel by $E_i$. Mathemati-
cally,

\[ E_i = \sum_{j \in \Omega_i} |x_{f,j}|^2, \quad i = 1, 2, \ldots M. \] (3.10)

Then, we can define \( \mathbf{e} = [E_1, E_2, \ldots E_M]^T \) as the vector of energies of the received signal in different channels. Since most of the spectrum bands are not occupied, the vector of channel energies \( \mathbf{e} \) is sparse.

Here, each node generates \( N \) wide-band filters \( \{H_n(f)\}_{n=1}^N \) such that

\[ H_n(f_i) = [\Phi]_{n,i}, \quad n = 1, 2, \ldots N, \quad i = 1, 2, \ldots M \] (3.11)

where \( H_n(f) \) represents the transfer function of the \( n \)th filter. \( \Phi \) is a random \( N \times M \) matrix which is provided to each node. Here, \( N < M \) depends on the sparsity of the channel occupancy. The node then feeds the wide-band signal into the \( N \) filters and the output at the \( n \)th filter is defined as

\[ \mathbf{z}_n = \text{Conv}(\mathbf{x}_t, \mathbf{h}_n) \] (3.12)

where \( \text{Conv}(\cdot, \cdot) \) denotes the convolution operation and \( \mathbf{h}_n \) is the impulse response sequence of the \( n \)th filter. The energy of the output signal of each filter is then measured to get the \( N \times 1 \) energy vector \( \mathbf{y} \)

\[ y_n = \mathbf{z}_n^H \mathbf{z}_n, \quad n = 1, 2, \ldots N \] (3.13)

\[ \mathbf{y} = [y_1, y_2, \ldots y_N]^T \] (3.14)

where \( (\cdot)^T \) and \( (\cdot)^H \) represent transpose and conjugate transpose of a matrix, respectively. Assume that the frequency response of each filter is approximately constant throughout each channel and equals \( H_n(f_i) = [\Phi]_{n,i} \) for the \( n \)th filter and the \( i \)th channel. Hence the energy at the output of the \( n \)th filter can be represented as

\[ y_n = \sum_{i=1}^M |H_n(f_i)|^2 E_i, \quad n = 1, 2, \ldots N. \] (3.15)
Define $\tilde{\Phi}$ as a matrix whose elements are square absolute values of the elements of the random matrix $\Phi$

$$
\tilde{\Phi} = \begin{bmatrix}
|H_1(f_1)|^2 & |H_1(f_2)|^2 & \ldots & |H_1(f_M)|^2 \\
|H_2(f_1)|^2 & |H_2(f_2)|^2 & \ldots & |H_2(f_M)|^2 \\
\vdots & \vdots & \ddots & \vdots \\
|H_N(f_1)|^2 & |H_N(f_2)|^2 & \ldots & |H_N(f_M)|^2
\end{bmatrix}
$$

(3.16)

Now, (3.15) can be written as

$$
y = \tilde{\Phi}e. 
$$

(3.17)

The goal is to estimate the length $M$ vector $e$ using the length $N$ measurements vector $y$.

### 3.3.2 Compressive Detection

It is now straightforward to establish the correspondence between our filter-based node design and the compressive sensing theory. We assume that at each node and at each instance of time, only a small portion of the channels are occupied. This is equivalent to assuming that the energy vector $e$ is sparse. Therefore, by properly choosing the number of the filters, $N$, based on the compressive sensing theory, the channel energy vector $e$ can be recovered from the measurement vector $y$ as

$$
\hat{e} = \arg \min_{e} \|e\|_1
$$

Subject to $y = \tilde{\Phi}e$

Each node reconstructs the energy vector $e$ from the vector of measurements $y$. Next, a threshold is adopted and the values of $e$ are compared with the threshold to decide on the occupancy of the channels.
In CR networks, the threshold adopted in each channel depends on the maximum level of interference allowed by the primary user. Assume that the distance from the primary transmitter to the primary receiver is denoted by $C$. If the guaranteed signal-to-interference ratio (SIR) for the primary communication is $\gamma$, the interference range of the primary receiver, $D$, can be determined by

$$\frac{P_pL(C)}{P_sL(D) + P_b} = \gamma$$

(3.18)

where $P_p$ and $P_s$ are the primary transmitter and the CR’s transmit powers, $P_b$ is the power of background interference at the primary receiver and $L(d)$ is the total path loss at distance $d$ [43]. Consequently, the CR node should be able to sense any signal coming from a distance of maximum $C + D$ or equivalently any signal with power equal to or greater than $P_{min} = P_pL(D + C)$. So in each channel, if $P_{min} > BN_o$, where $N_o$ is the noise spectral density, the threshold should be set above the noise level and below $P_{min}$. Otherwise the CR node is not in the interference range of the primary receiver and can always transmit in the underlying channel. The parameters $\gamma$, $C$ and $P_b$ should be provided by the regulator or the corresponding primary system [43].
3.3.3 Cooperative Spectrum Sensing in Ad-hoc Networks

The proposed compressive detection method could be adopted by ad-hoc networks for efficient spectrum utilization. Assume a number of nodes communicating within an ad-hoc network and spectrum of $W$ Hz is assigned to the whole network. This spectrum is divided into $M$ service channels and a low bandwidth control channel. The control channel is used to convey control commands such as connection initialization commands and channel occupancy estimation. One-to-one communication is assumed so whenever one node in the network has information to share with another, an empty channel has to be selected and used for the transmission.

A major challenge in such scenario is the hidden terminal problem. This is illustrated in Fig. 3.1. Assume that $N_1$ wants to transmit the data to $N_2$. If $N_1$ senses the channel, it may choose the same channel in which $N_5$ is transmitting. As $N_1$ is not in the range of $N_5$, it will not detect energy in $N_5$'s channel. On the other hand, $N_2$ is in the transmission range of $N_5$. So if $N_1$ uses this channel, its signal will collide with $N_5$'s at the receiver ($N_2$).

In order to prevent the hidden terminal problem, the nodes communicating in an ad-hoc network should cooperate in finding the empty channels. Whenever data is available at one node intended to be sent to another, the destination is also notified through the control channel and then both nodes sense the spectrum and exchange their estimates of the available channels through the control channel. The estimates made at the two nodes might be different since each may receive signals from close by nodes that are communicating through one of the $M$ channels that are not detected by the peer node. Next, based on the two estimates of the occupancy pattern, they agree on one or a number of channels that are empty.

The exchange of spectrum estimates can be performed by sending $M$ bits over the control channel in which 1's show the locations of occupied channels. Upon receiving this bit stream, each node can use bitwise OR operation to obtain available channels at
To find the thresholds that nodes should adopt in the channel occupancy detection, assume that a minimum frequency reuse distance $D$ is determined for the network. In other words, the same channel can be re-used to connect two other nodes if both are in a distance of at least $D$ from the nodes initially using the channel. Using the same path loss function $L(\cdot)$, the threshold at each channel can be set to $\gamma = PL(D)$ where $P$ denotes the maximum transmit power of the nodes. As both nodes detect the channel occupancy using the proposed compressive detection approach, it is guaranteed that no interference or hidden terminal problem occurs.

The cooperation between the network nodes in the ad-hoc network occurs through a control channel. The information to be shared among the nodes could be either in the form of hard or soft decision. While in the case of hard decision only the channels’ occupancy information is shared among nodes, more accurate decision could be taken by using soft decision where some reliability information are sent as well. In each CR, after compressively sensing the signal and obtaining the vector $y$, the energy vector $e$ is reconstructed. Using this vector, CR has to detect the presence of a primary user in each frequency band. To do this, a threshold should be chosen and the components of $e$ should be compared with the threshold. Choosing this threshold is based on the probability of false alarm or detection requirements for the underlying application. Specially in cognitive systems, probability of detection needs to be set high as failing to detect the presence of the primary user will consequent in introducing interference to the owner of the frequency band. This makes it more reasonable to chose a threshold that corresponds to a point in the ROC that has a high probability of detection.

For the case of hard decision, each CR takes a definite decision on each channel but provides no reliability information about this decision. The values 1 and 0 are used here to denote a decision that a certain channel is occupied/unoccupied and are not related to the energies of the channels. Thus, a vector of size $M$ bits containing binary values
is shared with other nodes. When a CR that is trying to occupy a channel receives this vector, it compares the occupancy information from that CR with its information. This way, a channel will be chosen if and only if all the nodes decide on it to be empty. On the other hand, for the case of soft decision, additional information are shared to increase the reliability of the decision.

3.3.4 Advantages over Current Algorithms

The proposed algorithm has several advantages over the algorithms already suggested in the literature. First, unlike methods suggested in [3, 14], in our compressive detection, the frequency domain representation of the signal is not reconstructed. Instead, just the vector of channel energies is obtained by solving the $\ell_1$ norm minimization. This benefits the complexity of the problem in two aspects.

First, the energy vector in our algorithm has exactly $M$ elements and hence the optimization problem has dimension $M$, while in spectrum reconstruction, the dimension is $R = rM$ where $r$ is the number of samples per channel.

Second, the optimization variable in spectrum reconstruction is a complex vector. In [44], it is shown that the $\ell_1$ norm minimization in this case can be solved using second order cone programming. On the other hand, in the proposed energy detection, the optimization variable is a vector of real, nonnegative numbers. Authors of [45] show that in this case, the optimization problem can be solved with linear programming.

To compare the complexities of the two methods, as an example, suppose that a spectrum consisting of 20 channels is being sensed. If 10 samples per channel are used in the spectrum reconstruction scheme, the complexity factor of the proposed method is $400000 \left(\frac{(rM)^3}{M}\right)$ times smaller than that of the spectrum estimator.

It is also worth mentioning that comparing with non wide-band spectrum sensing methods, i.e. channel-by-channel scanning, the proposed method outperforms in complexity in spite of the added compressive sensing algorithm. First, in channel-by-channel
scanning, a bank of $M$ narrow-band filters are needed to scan each channel while the proposed method exploits just $N \ll M$ filters. Second, the filters used in the proposed method are wide-band filters having a much shorter impulse response and hence lower filtering complexity. Considering that our $N$ filters have bandwidth $M$ times larger than a narrow band single channel filter, and the fact that we have only $N$ filters, leads to the conclusion that the filtering complexity of the proposed method over the conventional channel-by-channel scanning is $\frac{1}{M} \times \frac{N}{M} = \frac{N}{M^2}$.

Another advantage of the proposed method, is the effect of the compressive sensing algorithm on the detection performance in the presence of noise. If there is no noise, the energy vector $e$ has lots of zeros and a few nonzero elements representing the occupied channels. This sparse vector hence can be reconstructed based on the compressive sensing theory. In real situations, the energy of the unoccupied channels is $B N_o$ (the noise energy) and therefore $e$ has no zeros. In this case, the reconstructed signal is not an exact copy of the energy vector as the sparsity has changed. Nevertheless, numerical results suggest that as the compressive sensing algorithm searches for a vector with least number of nonzero elements, in the reconstructed energy vector, the noise effect has been suppressed compared to the original energy vector. In other words, as far as a reasonable signal power is present in the receiver, the output of the compressive sensing algorithm has a higher SNR.

### 3.4 Chapter Summary

In this chapter we proposed two techniques for wide-band spectrum sensing in CR networks. The first method is used to estimate the spectrum of the received signal with decreasing the time delay or the sampling rate compared to the method suggested in [3].

The compressive detection technique for detecting the holes in the spectrum was also proposed. The signal was fed into a number of wide-band filters and the outputs of
the filters were used to reconstruct the vector of channel energies through the $\ell_1$ norm minimization. An energy detection was then performed to detect the spectrum holes. The compressive detection method offers the receiver with lower complexity by reducing the number of filters and higher detection performance by suppressing the noise energy in the unoccupied bands.
Chapter 4

Sparsity Level Detection

An important issue in compressive sensing is to find the optimum number of measurements needed for successful reconstruction of a sparse signal. As discussed in Section 2.1, in order to get a successful reconstruction of the signal, the number of measurements, \( N \), should satisfy the following inequality

\[
N \geq cK \log \left( \frac{M}{K} \right) \quad (4.1)
\]

Therefore, sparsity level \( K \) is required to find the sufficient number of measurements \( N \) needed for successful reconstruction of the signal.

In this chapter, we propose two techniques for sparsity level detection of compressive signals based on information theoretic algorithms (MDL and PDL) and the belief propagation algorithm.

First, we complete the compressive detection algorithm proposed in Section 3.3. In compressive detection, the number of filters needed to successfully detect the unoccupied channels, depends on the total number of channels as well as the number of occupied ones. The lower the number of occupied channels, the smaller the number of filters required. It is important to use the least number of filters as this lowers the delay and the complexity of the system. In Section 3.3, we assumed the maximum number of channels that are occupied simultaneously is known and hence, the minimum number of required filters was
determined through simulations. However, in practice the number of occupied channels is changing and needs to be updated on a regular basis. In this chapter, information theoretic algorithms are used for estimating the number of occupied channels. This estimate is then used to determine the required number of filters which is then looped back to the filter set block. Optimizing the filtering process enhances the performance of the compressive detector by providing smaller delay and less complexity.

Next, we propose a novel technique based on belief propagation to find the sparsity level of a sparse signal. The problem is formulated as a multiple hypothesis test, where several hypotheses are examined, each corresponding to a particular sparsity level, and the hypothesis that has the highest probability is selected. The proposed algorithm can detect the sparsity level with a smaller portion of the delay of conventional approaches and with a slower growing complexity.

### 4.1 Sparsity Level Detection For Wide-Band Spectrum Sensing Via Information Theoretic Algorithms

In this section, we use the two information theoretic methods (MDL and PDL) that were discussed in Section 2.4 to detect the number of occupied channels in CR networks. First, we discuss the system model. Next, the concept of decomposing the observation vector to signal and noise subspaces is studied. Then, the enumeration methods of PDL and MDL are proposed.

#### 4.1.1 System Model

From the discussion in Section 3.3, the system model is given by the following

\[ y = \hat{\Phi} e. \]  

(4.2)
Figure 4.1: The block diagram of the system

The block diagram of the system is shown in Fig. 4.1. As seen in this diagram, the system consists of a feed-forward path and a feedback loop. The feed-forward path was introduced in Section 3.3. In the feed-forward path, the received wide-band signal is fed into the $N$ wide-band filters defined in (3.11). The energies of the filter outputs are then used to reconstruct the channel occupancy pattern in the whole spectrum using the $\ell_1$-norm minimization as follows

$$\hat{e} = \arg \min_{e} \|e\|_1$$

subject to $y = \tilde{\Phi}e$. \hspace{1cm} (4.3)

Energy detection is then performed by comparing the reconstructed vector of energies $\hat{e}$ to a vector of energy thresholds to find the unoccupied channels. Assuming no knowledge about the sparsity of the channel occupancy, a full set of filters $M$ must be adopted to capture all information about the signal. This large number of filters $M$ adds to the complexity of the system. In this section, we introduce the feedback loop in Fig. 4.1 which plays an essential role in finding the required number of filters $P < M$.

The feedback loop consists of a noise whitening filter and the enumerator. The whitened energies of the filter outputs are fed into PDL/MDL enumerator which estimates the number of occupied channels. The required number of filters to be adopted, $P$ follows from the compressive sensing theory, where $P$ should satisfy the inequality
given in (4.1). Using this inequality, the required number of filters, \( P < M \) is determined and this number is then fed into the set of filters to manage the number of filters to be used.

The number of filters in the feed-forward path is crucial since the outputs of the filters are used to solve the \( \ell_1 \)-norm minimization problem in (4.3). By activating the feedback loop periodically, for example every minute, the enumerator can estimate the sparsity of the channel occupancy. Then, using (4.1), the number of filters to be used by the feed-forward path is calculated and only \( P \) filters are used during that minute.

Next, in the system model given in (4.2), the received signal has an additive noise. Hence, channel energies are noisy and vector \( e \) can be written as

\[
e = \bar{e} + n
\]  

(4.4)

where \( \bar{e} \) is the \( M \times 1 \) vector of channel energies and \( n \) is the \( M \times 1 \) vector of noise energies. Let the mean and the covariance matrix of vector \( n \) be denoted as \( \mu_n \) and \( C_n \), respectively. Since all channels have the same bandwidth \( B \), the noise energy for all channels is equal and given by \( \sigma_n^2 \). Also, using the assumption that centralized noise energy samples are independent for different channels, the covariance matrix of vector \( n \) is given by \( C_n = E[(n - \mu_n)(n - \mu_n)^T] = \sigma_n^2 I \). Hence, the model in (4.2) can be rewritten as

\[
y = \tilde{\Phi} \bar{e} + \tilde{\Phi} n.
\]  

(4.5)

Clearly the noise component at (4.5) is colored and needs to be whitened. Define \( \tilde{n} \triangleq \tilde{\Phi} n \) to be the noise component in \( y \) in (4.5). The covariance matrix for \( \tilde{n} \) is \( C_{\tilde{n}} = E[(\tilde{n} - \tilde{\Phi} \mu_n)(\tilde{n} - \tilde{\Phi} \mu_n)^T] = \sigma_n^2 \tilde{\Phi} \tilde{\Phi}^T \). Since \( C_{\tilde{n}} \) is positive definite, it has a unique Cholesky decomposition, i.e., there exists a unique lower-triangular, non-singular matrix
\( \mathbf{L} \) such that \( \mathbf{L}\mathbf{L}^T = \tilde{\Phi}\tilde{\Phi}^T \). Multiplying both sides of (4.5) by \( \mathbf{L}^{-1} \) we get

\[
\tilde{\mathbf{y}} = \mathbf{L}^{-1}\mathbf{y} = \mathbf{L}^{-1}(\tilde{\Phi}\tilde{\mathbf{e}} + \tilde{\Phi}\tilde{\mathbf{n}})
\]

\[
= \underbrace{\mathbf{L}^{-1}\tilde{\Phi}\tilde{\mathbf{e}}} + \underbrace{\mathbf{L}^{-1}\tilde{\Phi}\tilde{\mathbf{n}}}. \tag{4.6}
\]

We can easily show that the noise component \( \tilde{\mathbf{n}} = \mathbf{L}^{-1}\tilde{\Phi}\tilde{\mathbf{n}} \) has the covariance matrix

\[
\mathbf{C}_{\tilde{\mathbf{n}}} = E[(\tilde{\mathbf{n}} - \mu_{\tilde{\mathbf{n}}})(\tilde{\mathbf{n}} - \mu_{\tilde{\mathbf{n}}})^T] = \sigma_n^2\mathbf{I}.
\]

Hence, the system model in (4.5) can now be written as

\[
\tilde{\mathbf{y}} = \tilde{\Phi}\tilde{\mathbf{e}} + \tilde{\mathbf{n}} \tag{4.7}
\]

where \( \tilde{\mathbf{y}} \) is the \( M \times 1 \) vector of observations, \( \tilde{\Phi} \) is an \( M \times M \) random matrix, \( \tilde{\mathbf{e}} \) is the \( M \times 1 \) vector of channel energies and \( \tilde{\mathbf{n}} \) is the \( M \times 1 \) vector of whitened noise energies. We assume that \( \tilde{\Phi} \) is a full-rank matrix which means that the columns of \( \tilde{\Phi} \) are linearly independent.

We assume that the received signal snapshots form an independent identically distributed (i.i.d.) sequence of Gaussian random vectors, \( \mathbf{x}_t \sim \mathcal{N}(0,\sigma^2) \). The received signal is sampled using the sampling resolution \( r \) which represents the number of frequency samples per channel. As the DFT matrix \( \mathbf{F}_M \) in (3.3) is unitary, the frequency domain sequence \( \mathbf{x}_f \) has the same distribution as \( \mathbf{x}_t \), i.e. \( \mathbf{x}_f \sim \mathcal{N}(0,\sigma^2) \). Since \( \mathbf{x}_f \) follows a Gaussian distribution, \( \left( \frac{\mathbf{x}_f}{\sigma} \right)^2 \sim \chi^2(1) \) where \( \chi^2(1) \) denotes chi-squared distribution with one degree of freedom. The signal energy in the \( i \)th channel, \( E_i \) is given in (3.10). The channel energy is the summation of the energies of all \( r \) samples in that channel. Hence, channel energies follow a chi-squared distribution with \( r \) degrees of freedom. Mathematically,

\[
\frac{E_i}{\sigma^2} = \sum_{j \in \Omega_i} \left( \frac{\mathbf{x}_{f,j}}{\sigma} \right)^2 \sim \chi^2(r), \quad i = 1, 2, \ldots M. \tag{4.8}
\]

Using a high resolution for sampling means that the degree of freedom is large, and the chi-squared distribution approaches the Gaussian distribution. Hence, we can assume
that channel energy snapshots form an (i.i.d.) sequence of Gaussian random vectors with unknown mean and unknown covariance matrix $C_e$. The noise snapshots are assumed to form an (i.i.d.) sequence of Gaussian random vectors with unknown mean and covariance matrix $\sigma^2I$. Hence, the observation vector is a Gaussian process with the covariance matrix

$$C = E[(\tilde{y} - \mu)(\tilde{y} - \mu)^T|\tilde{\Phi}, \sigma^2, C_e]$$

$$= \tilde{\Phi}C_e\tilde{\Phi}^T + \sigma^2I. \quad (4.9)$$

The probability density function of the observation vector is given by

$$f(y|C) = \frac{\exp\left(-\frac{1}{2}(\tilde{y} - \mu)^TC^{-1}(\tilde{y} - \mu)\right)}{(\sqrt{2\pi})^M\sqrt{\det(C)}} \quad (4.10)$$

Our goal is to find the sparsity level of $\tilde{e}$, which is the same as detecting the number of occupied channels in the CR network.

### 4.1.2 Estimating The Number of Occupied Channels

**Subspace Decomposition**

In [46], it has been shown that the space of the observation vector $\tilde{y}$ can be decomposed into two orthogonal subspaces, namely the signal subspace and the noise subspace. The signal subspace is the subspace that is spanned by the column vectors of $\tilde{\Phi}$ and it matches the span of the $m$ eigenvectors that correspond to the largest $m$ eigenvalues of the sample covariance matrix. The presence of noise causes $\tilde{y}$ to have components in the orthogonal complement subspace which is the noise subspace. The dimensions of the signal and noise subspaces for model $m$ are $m$ and $M - m$, respectively. The ML estimate of the signal subspace for model $m$ corresponds to the span of the eigenvectors that correspond to the largest $m$ eigenvalues of the ML estimate of the covariance matrix. It follows from decomposing the observation vector into the signal and noise subspaces that the signal and noise eigenvalues are distinguished. The largest $m$ eigenvalues of the sample
covariance matrix are shown to be much larger than the remaining $M - m$ eigenvalues that correspond to the noise.

**Predictive Description Length**

In PDL, the cost function is calculated for all models $m \in \Gamma = \{0, 1, \ldots, M - 1\}$ where each model corresponds to a particular number of occupied channels, and the model that has the least cost is selected. From (4.7), the system output for the model of order $m$ at time instant $q$ is given by

$$\bar{y}_q = \bar{\Phi}e_q + n_q.$$  \hfill (4.11)

It is worth mentioning that, since a full set of filters is used for the enumeration process, then we have $M = P$. Since the length of the observation vector $\bar{y}_q$ is $M$, we collect $M$ observations from which we form the sample covariance matrix $\bar{C}_M$ as follows

$$\bar{C}_M = \frac{1}{M} \sum_{q=1}^{M} (\bar{y}_q - \mu_q)(\bar{y}_q - \mu_q)^T$$  \hfill (4.12)

where $\mu_q$ is the sample mean vector and is given by

$$\mu_q = \frac{1}{q} \sum_{i=1}^{q} \bar{y}_q.$$  \hfill (4.13)

The sample covariance matrix at time instant $q$ for $q > M$ is as follows

$$\bar{C}_q = \frac{q-1}{q} \bar{C}_{q-1} + \frac{1}{q}(\bar{y}_q - \mu_q)(\bar{y}_q - \mu_q)^T.$$  \hfill (4.14)

The PDL criterion for calculating the cost function is given in (2.12). Since the first $M$ samples are used to form $\bar{C}_q$, the PDL for a model of order $m$ is given by

$$\text{PDL}_{m}(Q) = - \sum_{q=M+1}^{Q} \log f(y_q | \psi^m_{q-1})$$  \hfill (4.15)

where $Q$ is the window size which is the number of collected observations.

Define $\bar{\lambda}_{q,1}, \ldots, \bar{\lambda}_{q,M}$ as the eigenvalues of $\bar{C}_q$ where $\bar{\lambda}_{q,1} \geq \bar{\lambda}_{q,2} \geq \ldots \geq \bar{\lambda}_{q,M}$. Also, define $\bar{u}_{q,1}, \ldots, \bar{u}_{q,M}$ as the corresponding eigenvectors of $\bar{C}_q$. The parameter vector $\psi^m$
at time $q$ is defined as

$$
\psi^m_q = (\bar{\lambda}_{q,1}, \ldots, \bar{\lambda}_{q,M}, \bar{u}_{q,1}, \ldots, \bar{u}_{q,M}, \sigma^2_m) \tag{4.16}
$$

where $\sigma^2_m$ is the noise variance. Using (4.10) and (4.15), by ignoring the constant terms that are independent of the model, the PDL criterion is given by

$$
PDL_m(Q) = \sum_{q=M+1}^{Q} \left( \log |\hat{\mathbf{C}}^m_{q-1}| + (\bar{\mathbf{y}}_q - \bar{\mathbf{\mu}}_q)^T [\hat{\mathbf{C}}^m_{q-1}]^{-1}(\bar{\mathbf{y}}_q - \bar{\mathbf{\mu}}_q) \right). \tag{4.17}
$$

The sample covariance matrix $\hat{\mathbf{C}}_q$ can be used to calculate the ML estimate of the true correlation matrix $\hat{\mathbf{C}}_q$ as follows [47]

$$
\hat{\lambda}_i = \bar{\lambda}_i, \quad i = 1, 2, \ldots, m \tag{4.18}
$$

$$
\hat{\lambda}_i = \hat{\sigma}^2_m = \frac{1}{M-m} \sum_{i=m+1}^{M} \bar{\lambda}_i, \quad i = m + 1, \ldots, M \tag{4.19}
$$

$$
\hat{\mathbf{u}}_i = \bar{\mathbf{u}}_i, \quad i = 1, 2, \ldots, M \tag{4.20}
$$

where $\hat{\sigma}^2_m$ is the ML estimate of the noise variance. Equation (4.19) follows from the fact that the true covariance matrix $\mathbf{C}_q$ has all $M-m$ eigenvalues that correspond to the noise to be identical and equal to $\sigma^2_m$. Hence, the ML estimate of $\sigma^2_m$, namely $\hat{\sigma}^2_m$, is the arithmetic average of the $M-m$ smallest eigenvalues of $\hat{\mathbf{C}}^m_q$. Using (4.18), (4.19) and (4.20), we get the ML estimate of the covariance matrix $\hat{\mathbf{C}}^m_q$. The determinant of $|\hat{\mathbf{C}}^m_q|$ is given by

$$
|\hat{\mathbf{C}}^m_q| = \prod_{i=1}^{m} \hat{\lambda}_{q,i} \cdot (\sigma^2_m)^{M-m} \tag{4.21}
$$

Using (4.17) and (4.21), the PDL criterion is given by

$$
PDL_m(Q) = \sum_{q=M+1}^{Q} \left( \sum_{i=1}^{m} \log \bar{\lambda}_{q-1,i} + (M-m) \log \hat{\sigma}^2_{m,q-1} \right.

\left. + (\bar{\mathbf{y}}_q - \bar{\mathbf{\mu}})^T [\hat{\mathbf{C}}^m_{q-1}]^{-1}(\bar{\mathbf{y}}_q - \bar{\mathbf{\mu}}) \right). \tag{4.22}
$$

Subtracting the term $\sum_{q=M+1}^{Q} \sum_{i=1}^{M} \log \bar{\lambda}_{q-1,i}$ from all models as it is independent of the model selected, the PDL criterion is given by

$$
PDL_m(Q) = \sum_{q=M+1}^{Q} \left( - \log \left[ \prod_{i=m+1}^{M} \bar{\lambda}_{q-1,i} \right] \left( \bar{\lambda}_{q-1,i} \right)^{M-m} \right)

\left. + (\bar{\mathbf{y}}_q - \bar{\mathbf{\mu}})^T [\hat{\mathbf{C}}^m_{q-1}]^{-1}(\bar{\mathbf{y}}_q - \bar{\mathbf{\mu}}) \right). \tag{4.23}
$$
where the first term measures the multiplicity of the smallest eigenvalue by calculating the logarithm of the ratio of the geometric average to the arithmetic average of the \((M - m)\) smallest eigenvalues of the sample covariance matrix \(\bar{C}_q\).

**Minimum Description Length**

In MDL, a cost function is calculated for all models \(m \in \Gamma\), and the best model that represents the data with the least cost is selected. In PDL, we need to construct the sample covariance matrix \(\bar{C}_q\) for all \(M < q \leq Q\) in order to use them to update the cost function in (4.23). However, the cost function for MDL (2.10) shows that we are just interested in the sample covariance matrix at time \(Q\), \(\bar{C}_Q\). The sample covariance matrix \(\bar{C}_Q\) is given by the following

\[
\bar{C}_Q = \frac{1}{Q} \sum_{q=1}^{Q} (\bar{y}_q - \mu_q)(\bar{y}_q - \mu_q)^T
\]

where \(\mu_q\) is defined in (4.13). Using (2.10) and (4.21), the MDL cost function can be written as

\[
\text{MDL}_m(Q) = Q \log |\hat{C}_m^{\mu_m}| + \frac{\nu_m}{2} \log Q.
\]

The number of free elements is known to be \(\nu_m = m(2M - m + 1)\) [17]. Using an approach similar to PDL, the MDL criterion for model \(m\) is given by

\[
\text{MDL}_m(Q) = -Q \log \left( \frac{\Pi_{i=m+1}^{M} \hat{\lambda}_{Q,i}}{(\hat{\sigma}_{m,Q}^2)^{M-m}} \right) + \frac{m(2M - m + 1)}{2} \log Q
\]

where the first term measures the multiplicity of the smallest eigenvalue and the second term is the over-parameterization factor.

**4.2 Sparsity Level Detection Via Belief Propagation**

In this section, the System model is discussed. Then, the sparsity level detection algorithm is proposed. The advantages of the proposed algorithm over other algorithms are explained. Finally, the application of the algorithm in CR networks is discussed.
4.2.1 System Model

In this section, our goal is to find $K$, the number of nonzero elements of $x$ which is equivalent to detecting the sparsity level of a compressive signal. In compressive sensing, the measurement vector $y$ consists of a set of $N$ linear combinations of the signal $x$, i.e.

$$y = \Phi x$$

(4.27)

where $y$ is the $N \times 1$ vector of measurements, $\Phi$ is the $N \times M$ sensing matrix and $x = [x_1, x_2, \ldots, x_M]^T$ is the $M \times 1$ sparse signal. We assume that the elements of $x$ are independent and identically distributed (i.i.d.) Gaussian random variables. We further assume that $\Phi$ is a full column rank matrix which means that any combination of $N$ or smaller number of columns of $\Phi$ are linearly independent.

Before running the BP algorithm, the factor graph should be constructed by setting all variable and factor nodes and establishing all the edges connecting the nodes. In our proposed algorithm, each element of the signal $x$ is represented by a variable node. In order to find the sparsity level $K$ of the signal, a sequence of binary hypothesis tests is considered for all the $M$ elements of the signal with null and alternative hypotheses

$$H_0 : x_i = 0$$

$$H_1 : x_i \neq 0$$

(4.28)

In the initiation of a factor graph, variable nodes send messages that are their prior distributions to function nodes. In our proposed algorithm, since our problem can be viewed as a sequence of binary hypothesis tests given by (4.28), and since the elements of $x$ are (i.i.d.) Gaussian, we incorporate the two-state mixture Gaussian distribution as our prior distribution of the signal’s elements.

Let the variables $l_i \in \{0, 1\}$ denote the state of each element taking the value 0 when $H_0$ holds and the value 1 when $H_1$ occurs. The random vector $\ell = [l_1, l_2, \ldots, l_M]$ represents the state variables of the $M$ elements. Next, we associate the probability
function of each element, $p(x_i)$, with the state variable $l_i$. For $l_i = 1$, we choose a Gaussian distribution with mean $\mu_1$ and variance $\sigma_1^2$ and for $l_i = 0$, we choose a Gaussian distribution with mean $\mu_0$ and variance $\sigma_0^2$. Hence, the conditional distributions of $x_i$, $p(x_i|l_i)$ are as follows

$$
p_1(x_i) \triangleq p(x_i|l_i = 1) \sim \mathcal{N}(\mu_1, \sigma_1^2)$$
$$p_0(x_i) \triangleq p(x_i|l_i = 0) \sim \mathcal{N}(\mu_0, \sigma_0^2)$$ (4.29)

### 4.2.2 Sparsity Level Detection Algorithm

Here, we propose an algorithm to detect the sparsity level $K$ of a compressive signal using belief propagation. This algorithm finds the probability for all models $m$, where each model corresponds to a particular number of non-zero elements of $x$. $m$ is selected from a set $\Gamma = \{0, 1, \ldots, M\}$. Hence, our problem can be viewed as a multiple hypothesis test, in which several hypotheses are examined, each corresponding to a particular model $m$, and the hypothesis (model) which has the highest probability is selected.

Our algorithm employs BP by message passing over factor graphs as shown in Fig. 4.2. Since this factor graph has loops, we need to run the BP algorithm for several iterations before it converges. As shown in Fig. 4.2, the variable nodes are represented by circles and the function nodes are represented by squares. The factor graph consists of two main parts. The first part is responsible for finding the marginal distributions for each of the $M$ elements $\{x_i\}_{i=1}^M$. The second part performs a multiple hypothesis test and selects the best model $m$ which represents the sparsity level $K$.

In the first part, the factor graph has three types of variable nodes: states $\{l_i\}_{i=1}^M$, signal’s elements $\{x_i\}_{i=1}^M$ and the observations $\{y_j\}_{j=1}^N$. Three types of function nodes can also be found in the factor graph. The first type is the prior function nodes which reflect our belief about the prior distributions of the signal’s elements and impose our prior knowledge on the state variables $l_i$. The prior distribution of the $i$th element $x_i$ is
Figure 4.2: The factor graph for the proposed algorithm

given by the probability mass function of the $i$th state variable $l_i$ as follows

$$f_{prior}(m, l_i) \triangleq p(l_i|m) = \begin{cases} \frac{m}{M} & l_i = 1 \\ 1 - \frac{m}{M} & l_i = 0 \end{cases} \quad (4.30)$$

Assuming that we have no prior knowledge about the sparsity level of the signal, we initialize the priors to be as follows

$$p(l_i) = \begin{cases} 0.5 & l_i = 1 \\ 0.5 & l_i = 0 \end{cases} \quad (4.31)$$

The second type of function nodes is used to connect the state variables and the signal’s elements variables and is given by

$$f_{mix}(x_i, l_i) \triangleq p(x_i|l_i). \quad (4.32)$$

Now, using (2.17) and (4.32), the prior distribution of $x_i$ is given by

$$p(x_i|m) = \frac{m}{M} \cdot p_1(x_i) + \left(1 - \frac{m}{M}\right) \cdot p_0(x_i). \quad (4.33)$$

The third type of function nodes is a delta function which connects each observation variable $y_j$ to a set of signal’s elements variables $X \subset \{x_1, x_2, \ldots, x_M\}$ that are involved in the calculation of that observation. That is
Chapter 4. Sparsity Level Detection

\[ f_{\text{obs}}(X, y_j) \triangleq \delta(y_j - \sum_j h_j x_j), \quad x_j \in X \] (4.34)

Implementing the factor graph in Fig. 4.2, and after running the algorithm, the marginal distributions \( p(x_i|y) \) are computed in each iteration using (2.18). Since the signal’s elements \( \{x_i\}_{i=1}^{M} \) are assumed to be independent, the joint conditional distribution of the signal \( x \) given the observation vector \( y \) is

\[ p(x|y) = \prod_{i=1}^{M} p(x_i|y). \] (4.35)

Next, we proceed to the second part of the factor graph which is designed to find the model \( m \). This part consists of a model variable node \( m \) and a model selector function node \( f_{\text{ms}} \). The function node \( f_{\text{ms}} \) is the conditional distribution of the model \( m \) given the distribution of the signal \( x \). Using Bayes rule we get

\[ f_{\text{ms}}(m, x) \triangleq p(m|x) = \frac{p(x|m)p(m)}{p(x)}. \] (4.36)

Given model \( m \) means that, out of the total \( M \) signal’s elements, \( m \) are non-zero and the remaining \( M - m \) elements are zeros. Clearly, there are \( \binom{M}{m} \) options for choosing the \( m \) elements that are non-zeros. Define \( S_k, \quad k = 1, 2, \ldots \binom{M}{m} \), to be all subsets of \( \Gamma \) whose cardinality is \( m \). Given model \( m \), the joint conditional distribution of the signal \( x \), for each subset \( S_k \), can be computed by assigning the marginal distribution \( p_1(x_i) \) for each of the \( m \) elements that are non-zero, and assigning the marginal distribution \( p_0(x_j) \) for each of the \( M - m \) elements that are zeros. Using the assumption that the signal \( x \) is (i.i.d.) Gaussian random vector, \( p(x|m) \) is the multiplication of the marginal distributions of all elements. The result is then summed over all the \( \binom{M}{m} \) choices. In other words,

\[ p(x|m) = \sum_{k=1}^{\binom{M}{m}} \left( \prod_{i \in S_k} p_1(x_i) \cdot \prod_{j \in S_k} p_0(x_j) \right) \] (4.37)
where $\bar{S}_k$ is the complement of the set $S_k$. Since we assumed that the sparsity level $K$ is unknown in advance, all models have the same probability to occur. Thus, the probability distribution of the model variable $m$ is assumed to be uniform and given by

$$p(m) = \frac{1}{M}. \quad (4.38)$$

Similarly, using (4.31) and (4.33), the initial a-priori joint distribution of $x$ is calculated as follows

$$p(x) = \prod_{i=1}^{M} [0.5p_1(x_i) + 0.5p_0(x_i)]. \quad (4.39)$$

Substituting (4.35) and (4.36) in (2.17), the probability distribution of the model variable $m$ is computed as follows

$$p(m|y) = \sum_{x_1} \sum_{x_2} \ldots \sum_{x_M} \left( p(m|x) \cdot \prod_{i=1}^{M} p(x_i|y) \right) \quad (4.40)$$

where $p(m|x)$ is the message sent from the function node $f_{ms}$ to the model variable node $m$. This distribution of $m$ is fed to the prior nodes to update our beliefs about the prior distribution of $x$. Using the distribution of $m$, we can extract the ML estimate $\hat{m}$ which is an approximation of $K$ using the following

$$\hat{m} = \arg \max_{m} p(m|y). \quad (4.41)$$

The steps of the proposed algorithm are summarized in Algorithm 1.

### 4.2.3 Advantages Over Current Algorithms

The algorithm proposed in this section has several advantages over the current algorithms in the literature. In MDL and PDL, a large number of observations, $Q$ must be collected in order to form the sample covariance matrix, which is used to find the ML estimate of the covariance matrix. Next, the multiplicity of the smallest eigenvalue is used to
Chapter 4. Sparsity Level Detection

Algorithm 1 Sparsity Level Detection Via Belief Propagation

for all nodes $x_i \in \mathbf{x}$: initialize signal’s elements priors as

$$p(x_i) = 0.5p_1(x_i) + 0.5p_0(x_i)$$

Initialize function $f_{ms}$ using (4.36)

for each iteration do

for all observations: compute messages $\mu_{f_{obs} \rightarrow x}(\mathbf{x})$ using (2.17)

for all sources: compute messages $\mu_{x_i \rightarrow f_{obs}}(x_i)$ using (2.16)

for all sources: compute the marginal distributions $p(x_i)$ using (2.18)

Compute the conditional distribution for variable $m$ using (4.40)

Update the signal’s elements priors

end for

Output the estimated sparsity level $\hat{m}$ using (4.41)

determine the sparsity level. However, in the BP algorithm proposed in this section, acquiring just one observation is enough to find the sparsity level. Using this observation and after running the algorithm for a number of iterations, the algorithm converges and (4.41) is used to estimate the sparsity level $K$.

In addition, the proposed algorithm can be used in real time and can be applied to non-stationary and time-varying systems. Hence, our proposed algorithm outperforms MDL which can only be applied to a batch of data and is not suitable for online detection. However, PDL is capable of detecting changes in the environment, thus, we need to compare the complexity of our algorithm to that of PDL.

To analyze the computational complexity of the proposed algorithm, assume that each message, passing in the factor graph, is represented by $s$ samples. For the first part of the factor graph which finds the marginal distribution for each of the $M$ elements $\{x_i\}_{i=1}^{M}$, the number of computations required at variable nodes to process the messages is $O(NMs)$ per iteration. This is because we have $M$ variable nodes, and for each node we need to multiply $N-1$ messages received at that node by the prior distribution.
For function nodes, the messages can be processed using convolution in the frequency domain, and since we have $N$ function nodes, the computational complexity is given by $O(NMs \log(s))$. Hence, the complexity required to find the marginal distributions is given by $O(NMs + NMs \log(s))$. This complexity can be reduced by using LDPC-like matrices for $\Phi$ [36].

Next, we consider the second part of the factor graph that finds the sparsity level. Finding the distribution of $m$ using (4.40) has an exponential complexity with the dimension of the signal $M$ ($M$ nested summations). In order to simplify the computation of the distribution of $m$, we notice that (4.36) can be factored as the multiplication of $M$ functions, where each is a function of one variable $\{x_i\}_{i=1}^M$ only. In other words, (4.36) can be factored as

$$f_{ms}(m, \mathbf{x}) = \prod_{i=1}^{M} f_i(x_i|m)$$

$$f_i(x_i|m) = \sum_{x_i} \left( \frac{\Phi_{p_{1}(x_i)}(\frac{M-m}{M}p_{0}(x_i))}{M \cdot (0.5p_{1}(x_i) + 0.5p_{0}(x_i))} \right).$$

Hence, (4.40) can be rewritten as

$$p(m|y) = \prod_{i=1}^{M} (f_i(x_i|m) \cdot p(x_i|y)).$$

Using this simplification, equation (4.43) is calculated for all models $m = [0, 1, \ldots, M]$, thus for each node variable $x_i$, we perform $O(sM)$ multiplications. Since we have $M$ node variables, the computation complexity for function (4.40) is $O(sM^2)$ per iteration.

Therefore, the overall complexity of the simplified algorithm is $O(sM^2 + NMs + NMs \log(s))$. Since the number of samples per message $s$ is constant, the complexity scales to $O(M^2 + cNM)$, where $c$ is a constant. Hence, our algorithm has lower computation complexity compared to PDL which involves eigenvalue decomposition and matrix inversion and has complexity given by $O(M^3)$. 
4.2.4 Channel Occupancy Detection in CR Networks

The technique proposed in this section is not limited to a specific scenario and can be applied to many different applications ranging from DS-CDMA signals [48] to array signal processing [49] and sinusoidal signal detection [50].

For CR networks, the proposed BP algorithm can be applied to detect the number of occupied channels in the network. To find the number of occupied channels, the sequence of binary hypothesis tests to be considered for all the $M$ channels available in the network has null and alternative hypotheses

$$\mathcal{H}_0 : \text{channel } i \text{ is unoccupied}$$

$$\mathcal{H}_1 : \text{channel } i \text{ is occupied} \quad (4.45)$$

CR energies can be modeled by the discussed two states, $l_i = 0$ for the unoccupied state and $l_i = 1$ for the occupied state. The two-state mixture Gaussian model is implemented that captures our prior beliefs about the occupancy of the channels. For the occupied state, a Gaussian distribution with a large mean $\mu_1$ is chosen, and for the unoccupied state, a Gaussian distribution with a small mean $\mu_0$ is used. Hence, $\mu_1 > \mu_0 > 0$ and $\sigma_1 \geq \sigma_0$. The two state mixture Gaussian model is illustrated in Fig. 4.3.
4.3 Chapter Summary

In this chapter, we proposed two techniques for sparsity level detection of compressive signals. We proposed information theoretic algorithms (MDL and PDL) for estimating the number of occupied channels in a wide-band shared spectrum. This estimate was then used to complete the compressive detection algorithm proposed in Chapter 3 to determine the required number of filters. Using the smallest possible number of filters reduces the delay and the complexity of the system.

Next, we proposed a novel technique based on belief propagation to find the sparsity level of sparse signals. The problem was formulated as a multiple hypothesis test, where several hypotheses were examined, each corresponding to a particular sparsity level, and the hypothesis that has the highest probability was selected. The proposed algorithm could detect the sparsity level with a smaller portion of the delay of conventional approaches and with a slower growing complexity.
Chapter 5

Simulation Results

In this chapter, the performance of the 4 algorithms proposed in Chapters 3 and 4 are evaluated through simulations. All the experiments in this chapter are conducted using MATLAB software.

This chapter first evaluates the method proposed in Section 3.2. We reconstruct the spectrum using different number of linear combinations and different number of time samples of the signal and we study the probability of error using the Bayesian detector described in Section 3.2.2. Next, the algorithm proposed in Section 3.3 is investigated for different number of filters. Sparsity level detection performance is evaluated using the algorithms proposed in Sections 4.1 and 4.2. The two proposed algorithms are compared in terms of performance and complexity. The ability of the algorithms to detect the changes in the environment is also discussed.

5.1 Compressive Wide-Band Spectrum Sensing with Reduced Delay and Complexity

In this section, we evaluate the performance of the method proposed in Section 3.2 that reduces the delay and the sampling rate for compressive wide-band spectrum sensing.
In Section 3.2, we suggested using $S < R$ samples for reconstructing the compressed signal. The introduction of the $S$ parameter resulted in two different dimensions that can affect spectrum estimation. Namely, reducing the number of linear combinations $N$ (lower complexity) and the number of time samples $S$ (less delay). In this section, we study the effect of using different values of $S$ and $N$ for a specific wide-band signal.

We consider a wide-band spectrum with total bandwidth of 100MHz. According to the Nyquist theorem, the sampling rate is $200 \times 10^6$ samples/second. The spectrum consists of 10 channels, and we assume that only 5 channels are active at a time.

Numerical solution of the $\ell_1$ optimization problem in (3.5) for different compression ratios shows that a compression ratio of at least $N/R = 45\%$ (for the case of 5 active channels) is required to get a reconstruction of the signal that resembles the original signal's spectrum.

Fig. 5.1 depicts the frequency representation of the noisy wide-band signal with SNR...
= 8 dB before compression. In Fig. 5.2, using compression ratio of \( N/R = 50\% \) which is just above the threshold, the signal is reconstructed using the first \( S/R = 50\% \) of the time samples. As seen in this figure, with only half of the samples, the reconstructed frequency response is still a good approximation of the wide-band signal. In Fig. 5.3, the compression ratio is kept at 50\% but just the first 30\% of the samples are used (70\% less delay). This figure shows that, while the spectrum reconstruction is not as accurate as before, the accumulation of energy in the channels is so that an energy detector can still detect the occupied channels.

Fig. 5.4 is the spectrum of the signal reconstructed with \( N/R = 50\% \) and a sampling rate 33\% lower than the Nyquist rate. In other words, we have decreased the sampling rate to 2/3 of the Nyquist rate. Again, although a fine spectrum reconstruction is not resulted, the energy detection is successful in retrieving the occupancy pattern.

Fig. 5.5 illustrates the probability of error of a Bayesian detector as described in
Section 3.2.2 versus SNR for different values of \( S \) and \( N \). The a priori probability of channel occupancy is 50% as, in the simulation, 5 out of 10 channels are set to be occupied. The best performance is that of \( N/R = S/R = 100\% \) which has no compression and uses all time samples. Next, using the full vector of time samples, the compression ratio is set to 50%. Fig. 5.5 shows that in this case SNR needs to be 3dB higher to keep the probability of error at \( P_e = 10^{-5} \). The same compression ratio has been used in the next experiment but only half of the time samples are used (50\% less delay). The price we pay here is around 3dB in SNR in \( P_e = 10^{-4} \).

Knowing that the threshold on \( N \) is 45\%, we have plotted the case of \( N/R = 33\% \) again with \( S/R = 50\% \). As seen in Fig. 5.5 and as expected, the degradation is significant. In the next experiment, we keep the compression ratio \( N/R = 33\% \) which is below threshold and this time decrease \( S \) to 33\%. As illustrated in Fig. 5.5 the performance is dramatically enhanced. This can be explained based on the discussion in Section-3.2.2.
Here, although the compression ratio is 33%, the threshold has decreased because the signal length is now just one third of the original signal. Hence, the used compression ratio falls above the threshold and the reconstruction algorithm performs much more accurate.

Fig. 5.5 confirms this discussion showing that with $N/R = S/R = 33\%$ we just need 1 dB more SNR to have the same performance as the $S/R = N/R = 100\%$ case at $P_e = 10^{-6}$. It is also observed that this case even outperforms the case of $N/R = 50\%$ and $S/R = 100\%$.

Being able to exploit just a portion of the time samples is specially important because for instance, with the configurations of the current simulations, just 30% of the nominal time can be dedicated for collecting samples and the decision can be made much faster.

We have also simulated a case where we have used a sampling rate equal to 2/3 of the
Nyquist sampling rate, keeping the compression ratio $N/R = 50\%$. Fig. 5.5 shows that the probability of error in this case is almost the same as that of the $N/R = 50\%$ and $S/R = 100\%$. Hence, as far as we are interested in the detection of channel occupancy, this scenario can be used with minor performance loss.

### 5.2 Compressive Detection For Wide-Band Spectrum Sensing

In this section, we evaluate the performance of the compressive detection algorithm proposed in Section 3.3 through simulations. The input signal to each node, is the wide-band noisy signal which is fed into $N$ different filters and the energy of the output signals is then used in the $\ell_1$ norm optimization problem to obtain the energy vector $\mathbf{e}$ (3.18).

In the simulations, a spectrum bandwidth of 20 channels is considered and it is also
assumed that at each node and at each instance of time, not more than 6 channels are occupied. Measurements show that a minimum of 12 filters ($N = 12$) is needed for successful reconstruction of the energy vector. Additive white Gaussian noise is added to the received time signal.

Fig. 5.6 illustrates the probability density function (PDF) of the estimated channel energy $e$ for an occupied and an unoccupied channel with SNR of 5 dB. As seen in this figure, the PDF of the detected energies in an occupied channel are very close for $N = 15$ and $N = 12$. On the other hand, for $N = 10$ the mean of the detected energy degrades significantly. This shows the threshold effect of $N$ based on the compressive sensing theory. Interestingly, the mean of the unoccupied channel energy, which is the mean of the reconstructed noise energy, decreases with $N$ as far as $N$ remains above the threshold. In other words, the compressive sensing algorithm is suppressing the input
noise at the output while keeping the signal almost constant and hence increasing the SNR. This is of course true if \( N \) is above the threshold. As seen in Fig. 5.6, for \( N = 10 \) the noise has been actually amplified.

Fig. 5.7 depicts the probability of error in channel occupancy detection versus SNR for different number of filters \( N \). As seen in this figure, for the number of filters above the threshold \( N = 12 \), the performance of the detector is almost the same. Using \( N = 12 \) filters, we have a performance loss of 1 dB. The degradation is on the other hand apparent when less than 12 filters are used. This shows that finding the number of required filters is crucial. If the number of filters adopted falls below that number, the compressive detection algorithm fails in detecting the holes in the spectrum. Finding this required number of filters is discussed in the following sections.
Table 5.1: Number of detected channels by MDL and PDL for 100 independent runs

<table>
<thead>
<tr>
<th>Method</th>
<th>m</th>
<th>SNR (dB)</th>
<th>-10</th>
<th>-9</th>
<th>-8</th>
<th>-7</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDL</td>
<td>1</td>
<td>86</td>
<td>54</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12</td>
<td>31</td>
<td>23</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2</td>
<td>14</td>
<td>32</td>
<td>15</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>30</td>
<td>82</td>
<td>97</td>
<td>99</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>PDL</td>
<td>1</td>
<td>54</td>
<td>24</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>21</td>
<td>25</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>17</td>
<td>21</td>
<td>19</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>8</td>
<td>30</td>
<td>68</td>
<td>87</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
5.3 Sparsity Level Detection Via Information Theoretic Algorithms

In compressive detection, the number of filters needed to successfully detect the unoccupied channels, depends on the total number of channels as well as the number of occupied ones. As discussed in Section 4.1, MDL and PDL methods can be applied to detect the number of occupied channels. In this section, the performance of MDL and PDL methods is evaluated through simulations.

A spectrum bandwidth of $M = 20$ channels is considered and it is assumed that only 4 channels are being used by some CR nodes. The input signal is the wide-band signal which is fed into $N = 20$ different filters. The energy vector of filter outputs is then used in the enumerators to find the number of occupied channels. We use a resolution
Figure 5.9: The probability of detection for MDL and PDL methods over 100 independent runs

(r) of 50 samples per channel. A narrow guard-band is also inserted between adjacent channels to decrease the effect of interference. The energies of the samples are added for each channel and they form the channel energy vector $\tilde{e}$.

We perform MDL and PDL methods for 100 independent runs. The sample covariance matrix $\tilde{C}_q$ for PDL and MDL over a window size $Q = 60$ is calculated using (4.14) and (4.24), respectively. The eigenvalues of $\tilde{C}_q$ averaged over 100 runs for SNR = 0 dB are shown in Fig. 5.8. It follows from the subspaces decomposition that the eigenvalues can be decomposed into signal and noise eigenvalues. In Fig. 5.8, the distance between the 4th and the 5th eigenvalues is larger than the distance between the other subsequent eigenvalues. Hence, the first 4 eigenvalues represent the occupied channel energies while the remaining eigenvalues represent the noise.

To further investigate the performance of the two methods, we present the results of
Figure 5.10: The detected number of occupied channels as a function of the observation window. The number of signals changes from 4 to 5 at $q = 40$

performing 100 independent runs in Table 5.1. In this table, we compare the performance of MDL and PDL detectors. Each row in the table shows the number of times that the detector selected the corresponding model for the given SNR.

The probability of detection for this experiment is shown in Fig. 5.9. The probability of detection is defined as the ratio of the number of runs in which the true number of occupied channels has been detected to the total number of runs. From the figure, we can notice that SNR of -6 dB is required to achieve over 90% probability of detection and SNR of -4 dB is needed to achieve full performance. The performance of the two methods is comparable with some gain for PDL over MDL. Fig. 5.9 shows that the number of occupied channels can be determined with a high probability of detection. Hence, this number can be used in (4.1) to estimate the number of required filters that should be used to find the locations of the unoccupied channels.
The PDL method uses the ML estimate of the covariance matrix of the observation data at each time instant to calculate the cost function. Hence, PDL can be applied to non-stationary and time-varying systems. In the next experiment, we study the case where the number of occupied channels changes inside the window of observations. Suppose we have a window size of $Q = 60$ samples with SNR = 0 dB. We start the experiment with 4 occupied channels and change the number of occupied channels to 5 at $q = 40$ as shown in Fig. 5.10. The PDL cost is computed for all $1 \leq q \leq 60$. At each time instant, we find the number of occupied channels by locating the minimum PDL cost. In this experiment, since we have 20 channels, the PDL method requires minimum of 20 observations to operate. These observations are used to form the sample covariance matrix, which is used to find the ML estimate of the covariance matrix. As shown in Fig. 5.10, the number of detected occupied channels changes from 4 to 5 after the 40th time instant.

5.4 Sparsity Level Detection Via Belief Propagation

In Section 4.2, we developed an algorithm based on BP to detect the sparsity level of a compressive signal. For CR networks, the proposed algorithm can be applied to detect the number of occupied channels in the network. In this section, the performance of the proposed algorithm is evaluated through simulations.

Similar to the setup used in Section 5.3, a spectrum bandwidth of 20 channels is considered, $M = 20$, and it is assumed that only 4 channels are occupied at each time instance, $K = 4$. Upon receiving the input wide-band signal, it is fed into 20 different wide-band filters, $N = 20$, and the energy vector of filter outputs $y$ is then used as the observation vector.

We perform the proposed algorithm for 100 independent runs. Each message in the factor graph is represented by $s = 315 = 3^2 \times 5 \times 7$ samples which provides fast
Figure 5.11: The distribution of model variable $m$ over 30 iterations. $M = 20, K = 4, N = 20$ and SNR = 0 dB

computation of FFT. In each run, the probability distribution of the model variable $m$ is calculated using (4.40) with 30 iterations per run for SNR = 0 dB. The distribution of $m$ averaged over 100 runs is shown in Fig. 5.11. In this figure, one can see that model $m = 4$ has higher probability than other models. As a result, it follows from (4.41) that $K = \hat{m} = 4$ which reflects the actual number of occupied channels in the network.

To further evaluate the performance of the proposed method, we have performed different experiments. The probability of detection versus the number of measurements is shown in Fig. 5.12. The probability of detection over 100 independent runs has been plotted in Fig. 5.12 for SNR = 0 dB and different number of iterations. The number of measurements is the dimension of the observation vector $y$ and in this experiment also represents the number of wide-band filters used. From Fig. 5.12, we notice that running the algorithm for larger number of iterations improves the probability of detection. While
Figure 5.12: The probability of detection for different number of measurements and iterations. M=20, K=4 and SNR = 0 dB.

5 iterations are not enough to detect the actual number of occupied channels, we achieve a significant improvement by increasing the number of iterations to 7 or 10. Running the algorithm for 20 or 30 iterations does not introduce a significant change. Moreover, we note that as more filters are used, higher performance is achieved.

In Fig. 5.13, we illustrate the results of the simulation of the algorithm over 100 independent runs for \( N = 20 \) and different SNRs. The performance of the PDL method has also been plotted in the same figure. For PDL, we use the results from Section 5.3. As seen in Fig. 5.13, the performance of the proposed algorithm increases gradually with increasing SNR. Using just one observation, the probability of detection increases as we increase the number of iterations. In the figure, we have plotted the probability of detection for 10, 20 and 30 iterations. Using 30 iterations, the probability of detection exceeds 90% for \( \text{SNR} = -3 \text{ dB} \). Increasing the number of observations also enhances the
Figure 5.13: The probability of detection for different SNR for different number of iterations, observations and information theoretic methods. $N = 20$.

performance of the algorithm. With 10 observations and 20 iterations per observation, the probability of detection exceeds 90% with SNR that is 2 dB less than that required by using just one observation.

One advantage of the proposed algorithm is the smaller number of observations required to detect the sparsity level compared to other algorithms. This is illustrated in Fig. 5.14. In this figure, the SNR required to achieve 90% probability of detection is plotted over different number of collected observations ranging from 1 to 65 observations for MDL, PDL and the proposed algorithm with 20 iterations per observation. While the PDL method requires minimum of 20 observations to operate, the proposed BP algorithm achieves 90% performance with only one observation, in which it requires SNR of -3 dB. The SNR required drops to -5 dB with just 3 observations compared to 40 observations needed by the PDL method.
Next, we compare the running times of the proposed algorithm and PDL. We run the two methods with $N = M/2$ and $\text{SNR} = 0 \text{ dB}$ for different number of channels $M$ ranging from 5 to 55. The average running times over 100 independent runs for both algorithms, namely BP with different number of iterations and PDL are shown in Fig. 5.15. For PDL, the number of collected observations is 3 times the number of channels $M$. This choice of the number of observations insures achieving probability of detection similar to that shown in Fig. 5.13. However, for our proposed algorithm, acquiring just one observation is sufficient to produce the results shown in that figure. Fig. 5.15 shows that PDL performs faster than our proposed algorithm for small number of channels. However, the required time increases significantly by increasing the number of channels. From the previous experiment, although PDL performs better than BP for low SNR, the required time to find the number of occupied channels in PDL is much higher for large number of channels. These results agree with our complexity analysis discussed in Section 4.2.3. The numerical results suggest that our proposed method is more efficient in systems with larger number of channels than information theoretic methods.

The proposed algorithm can be used in real time and works well in non-stationary environments. In the last experiment, we study the case where the number of occupied channels changes between the collection of two observations. Using $M = 20$, $N = 20$ and $\text{SNR} = 0 \text{ dB}$, we collected one observation where 4 channels are occupied, $K = 4$. We ran the algorithm for 25 iterations. Next, another channel is occupied $K = 5$ and we collect another observation and ran the algorithm for 25 iterations. In Fig. 5.16, we have plotted the true and the detected number of occupied channels, $K$ and $\hat{m}$, respectively. The first iterations are not shown in the figure since the algorithm was not converged yet. From the figure, it took the method 4 iterations to detect the change in the environment.
Chapter 5. Simulation Results

Figure 5.14: The threshold SNR over different number of collected observations for BP-algorithm with 20 iterations, MDL and PDL

5.5 Chapter Summary

The performance of the 4 algorithms proposed in Chapters 3 and 4 was investigated through simulations.

For the compressive wide-band spectrum sensing algorithm with reduced delay, the tradeoff between complexity of the compressive sensing algorithm and the time delay was studied. It has been shown that by properly choosing these parameters, one achieve even better performance with smaller delay and less complexity.

For the compressive detection algorithm, numerical results suggested that the compressive sensing method offers the receiver with lower complexity by reducing the number of filters and higher detection performance by suppressing the noise energy in the unoccupied bands.
In the simulations for the sparsity level detection algorithm that uses MDL and PDL, numerical results showed that the number of occupied channels can be determined with high probability of detection.

Simulation results for the sparsity level detection algorithm that employs belief propagation showed that, the proposed algorithm could detect the sparsity level with a smaller portion of the delay of conventional approaches and with a slower growing complexity while maintaining the performance. It has also been shown that both PDL and the BP algorithm are able to detect the changes in the number of occupied channels in the surrounding.
Figure 5.16: The number of occupied channels have been changed from 4 to 5 in the 26th iteration. The actual and the detected models are shown. $M = 20$, $N = 20$ and SNR = 0 dB
Chapter 6

Conclusions and Future Work

With the rapid growth in wireless applications, spectrum resource becomes scarce. Although the current static spectrum management avoids interference effectively, this comes with the price of very low spectrum utilization. While some frequency bands are overcrowded, other bands are rarely used. CR promises to increase the utilization of frequency bands that are under-utilized by providing opportunistic spectrum access.

The under-utilization in most of assigned spectrum bands results in signals that are sparse in frequency domain. Such sparsity has motivated the use of compressive sensing in reconstructing the frequency representation of the signal with far-less time samples than Nyquist theorem imposes. Using wide-band spectrum sensing techniques, CR nodes can scan the whole spectrum at once and avoid the delay and complexity of channel-by-channel scanning.

The work presented in this thesis had two main objectives. The first one was finding efficient methods for detecting the holes in a wide-band radio spectrum and the second one is to find the sparsity level of compressive signals which optimizes the performance of the compressive sensing algorithm.

To achieve the first goal, we proposed two different algorithms. The first algorithm estimated the spectrum in an efficient manner and used the estimation to find the holes.
This approach added a new dimension to the scenario through ignoring specific portions of the time samples taking an advantage of the freedom in choosing the sensing matrix in compressive sensing. The second algorithm detected the spectrum holes by reconstructing channel energies instead of reconstructing the spectrum itself. In this method, the signal was fed into a number of filters, much less than the number of channels within the wide-band spectrum. The energies of the filter outputs were used as the compressed measurement to reconstruct the signal energy in each channel. The energy vector was then compared with a threshold vector to detect the spectrum holes.

For the second goal, we also proposed two different algorithms. The first algorithm employed two information theoretic algorithms (MDL and PDL) to find the sparsity level. In the second algorithm, belief propagation was adopted. This algorithm could detect the sparsity level with a smaller portion of the delay of conventional approaches and with a slower growing complexity.

The performance of the 4 algorithms proposed in this thesis was evaluated through simulations.

6.1 Future Work

In Chapter 3, the methods proposed for detecting the holes in a wide-band radio spectrum using compressive sensing or compressive detection depends on the numerical results which we got through simulations. Solid analysis is required to support the ideas presented in this chapter. Deriving an algorithm to find the number of samples needed to have certain performance with the least possible delay is in demand. Moreover, the sensing matrix considered in this work was random. However, it is important to investigate the restrictions that are to be applied to this sensing matrix. Finally, although the cognitive radio application was considered in this study, the work presented here could be applied to other applications as well.
Bibliography


