TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In order to meet exponentially growing consumer demand for wireless data, radio spectrum regulators considering opportunistic spectrum access policies. Historic spectrum regulation focussed on exclusive frequency assignments (licensing), with spatial and frequency separation to mitigate interference between users. However, this approach leads to considerable underutilisation in both space and time. Thus, faced with a need to provide 1000 times the bandwidth in 10 years, regulators are considering agile access technologies on a licence exempt basis.

Before opportunistic access is a reality, speedy, robust, and accurate estimation of frequency spectra must be made. This is a challenging statistical and engineering problem, limited by characteristics of wireless channels such as multipath fading, and shadowing. The proposed bands have a large bandwidth, containing sub-channels which are not contiguous but statistically correlated, and radio-wave fading environment which can mask high-powered transmissions. Traditional methods are un-viable in these conditions; either requiring expensive hardware to meet the data rate required to perform the sampling, or a large number of RF components to turn a single wideband channel into many narrowband ones.

This project addresses the issue of estimating available frequencies for opportunistic transmission, from a set of underdetermined measurements. The compete set of measurements may be available to a particular sensor. Or, the measurements may be distributed over a network of sensors, improving estimation accuracy, and

The growing number of wireless devices is placing increasing demand on radio spectrum. Consumers are demanding faster speeds and better quality connections in more places. However, there is a limited amount of frequencies to transmit on. Consequently, demand for frequencies which provide sufficient bandwidth, good range and in-building penetration is high.

Not all frequencies are used at all times and in all places. Judicious spectrum management could alleviate the issue of too few frequencies being available to devices by developing approaches to interleaving opportunistic transmissions within established bands.

There are benefits to spectrum sharing beyond simply satisfying consumer demand. Innovative wireless applications such as wireless rural broadband, remote monitoring, and machine-to-machine applications will be made viable without the need to purchase exclusive access to a specific frequency.

The historic allocation process has placed unused spectrum between adjacent radio channels, in an attempt to avoid interference between consecutive users. However, technical changes in transmission standards are making some bands available as modern coding and modulation techniques are more spectrally efficient. In particular, in the UK, the switchover from analogue to digital terrestrial broadcast has freed many previously used bands.

Recent regulatory focus has been on frequencies in the TV broadcast bands 470-790MHz. In the UK, TV channels are broadcast using up to six multiplexes, each requiring an 8MHz channel. A total of 32 of these channels are allocated for TV broadcast, whilst only 6 of these channels are required to receive the 6 multiplexes at any given location. This is because TV broadcast is high powered and needs spatial separation between coverage areas to avoid interference. As such, the majority of TV frequencies reserved for TV broadcast are unused in any given place. These white spaces are spectrum which has been left over to prevent interference between primary users (such as TV broadcast). These frequencies can be used on an opportunistic basis by relatively low-powered devices. Whilst current focus is on the TVWS bands, the work presented in this thesis can be used in other, as yet unrealised frequencies.

Currently access to spectrum is managed in two ways: licensed and license exempt access. Licensing authorises a particular user/users to exclusive use of a specific frequency band. License exempt access allows any user to access a band, provided they meet technical requirements intended to limit interference in on other users. This is a particularly pressing issue for the co-existence of licensed users and licence exempt users in the same band.

Devices seeking to access white spaces need a robust mechanism for learning which frequencies can be used at a particular time and location. The approach currently being taken by OFCOM and the FCC is to maintain a set of databases, which map the location of white spaces based on knowledge of existing spectrum users. An alternative approach is for devices to monitor the use of spectrum individually.

One approach to white-space access is to maintain a database of currently available frequencies. The database would contain up to date information about incumbents, including television transmitters and wireless microphones. It would also need to maintain information about currently operative secondary users. Devices would register with the database, based on their geo-location and the service would determine availability for the device, either based on in situ measurements, or from a propagation model.

Location based measurements are a costly approach, as they need to be redone every time

primary user transmission characteristics change, and they cannot be done economically in difficult to reach places. Propagation models can achieve good accuracy, but at the cost of being complex and computationally intensive. Simplistic models do not agree with ground truth measurements.

Devices must also supply geographic information to a database service. How this is done has yet to be codified, but current methodologies all have drawbacks. GPS positioning works well outdoors, but is inaccurate indoors. Cellular location can yield errors of up to a mile. Using the base-station location as a proxy for the device location results in an unacceptably high loss of whitespaces, as decisions would have to be needlessly conservative.

An alternative to maintaining a database of available frequencies, is for secondary users to independently sense spectrum. This method is subject to a number of technical limitations. The proposed bands have an ultra-wide bandwidth. Traditional sensing approaches to this have been either to divide the band into a number of contiguous narrow bands, or to simply use a high rate processing device to capture the necessary samples. Both methods are prohibitively expensive, and require dedicated and energy hungry hardware. Further, traditional statistical spectral detection and estimation techniques make assumptions about the primary users signal. Such assumptions include cyclo-stationarity, the presence of specific waveform patters, the use of known pilot sequences, and sufficiently high powered transmissions that energy detection is viable. In practice, some (or all) of these assumptions are violated; either because assumed patterns are not present in the transmission, or because the radio environment has a deep fade between the primary and secondary user masking high powered transmissions.

The biggest issues for single node sensing are multipath fading, shadowing, and PU receiver uncertainty. Cooperation between a network of nodes can improve sensing performance through spatial diversity. In cooperative sensing information from geographically diverse nodes is aggregated in the decision making process. Combining observations can overcome the deficiencies of observations unique to individual cognitive radios. Spatial diversity makes it unlikely that all radios will experience the same fading and receiver uncertainty. Approaches to cooperative sensing based on Nyquist sensing typically involve energy detection at each node, along with a gain combining procedure (which is either performed at a fusion node, or via a decentralised algorithm). Cooperative sensing can use such simple methods at each node as receiver sensitivity can be mitigated by using multiple statistically independent observations. This reduces the complexity and cost of the individual cognitive radios.

Cooperative sensing has drawbacks however. Spatial diversity does not necessarily mean statistically independent observations. A subset of nodes all blocked by the same object will make measurements which are correlated and corrupted by the same shadowing for example. Cooperative sensing also involves an overhead with the being part of a network. This is any extra time, energy and processing when compared to sensing individually.

An alternative sensing strategy is compressive sampling (CS). This is a new paradigm in signal processing which has emerged over the last decade, and has had significant success in imaging

problems. In particular compressive sampling strategies have reduced the time needed for an MRI scan from 2 minutes to 30 seconds. Patients previously had to hold their breath for the entire duration. The central innovation in compressive sensing is that randomness is an effective strategy for sensing sparse signals. In practice much of the TVWS spectrum is unoccupied, and CS can leverage this sparsity to reduced the sampling rate of the CRS.

Group Testing is a model sparse inference problem, first introduced by Dorfmann in the context of testing for rare disease. Given a population of items, some small fraction of which has an interesting property (labelled defective), Group Testing proposes algorithms to find those items efficiently. What allows these items to be discovered, in fewer tests than the total number of items, is testing pools of items. That is, items aren't tested individually, but several together. In testing for rare diseases the blood samples are mixed and the mixture is tested. This allows fast elimination of non-defective items. Popular pooling designs are randomised - items are included in a test with independent probabilities. We consider probabilistic algorithms with non uniform probabilities of each item's defectivity. These non-uniform priors could come from a TVWS database of possibly occupied spectrum, or from summarising risk and family history in disease testing.

In this thesis, we argue that in order to perform spectrum sensing n the TVWS band, at reasonable sampling rates we must exploit the new sensing paridigm of compressive sensing. We show how we can quantify the reduction in absolute number of samples needed to accuractely reconstruct a signal using this methodology. We diwcuss how the sample rate at a single node may still be too high even with the reduced rate, and we provide two approaches to reducing it even further. We also discuss how with use the TVWS band may no longer be sparse, in the frequency basis. To recify this, we propose a distributed model of the spectral gradient, which allows us to accurately reconstruct the frequency bamd. We do this via an improved version of distributed ADMM. To reduce the smapling rate at a single device, we propose to aviod reconstruction altogether. We show it is possible to determine whether a frequency range is occupied with as little as 5

The structure of this thesis is as follows. Chapter 2 covers the relevant material on Nyquist sensing theory and Nyquist approaches to the spectrum sensing problem. Nyqusit sensing is introuduced, and how to reconstruct the signal from its samples is covered. Then, what the spectrum sensing problem is is defined, along with narrowband approaches such as energy detection, feature detection, and matched filtering - in order of increasing complexity. Distributed and cooperative approaches to the problem are then dicussed, in particular the AND and OR rules of equal gain combining. We discuss that distrbuted approaches have better statistical performance due to spatial diversity, but that each sensor still needs to use the same number of samples as the single sensor problem. It is argued that the data rates demanded by TVWS sensing makes this Nyquist approach unfeasible, and that some mechanism of reducing the absolute number samples is required for spectrum sensing in the TVWS band to be a reality.

Chapter 3 covers the theory of compressive sensing and some previous applications to the Wide-band Spectrum sensing. In particular, the conditions under which a sparse high-dimensional vector

can be recovered from undersampled representations are introduced - the Restriced Isometry Property (RIP) and stable embeddings along with which sensing operators satisfy these conditions. Then three families of reconstruction algorithm are discussed: greedy, convex and Bayesian. The relative merits and demerits of these alogorithms are compared in terms of speed, accuracy, complexity, and undersampling performance are discussed, concluding that more accurate algorithms which can reconstruct from fewer samples are slower and more complex. Two compressive approches to Wideband Spectrum sensing are then discussed in depth: the Random Demodulator and the Modulated Wideband Coverter, before moving on to discuss distributed compressive sensing.

Chapter 4 covers the theory of ADMM. It starts with the a potted history of the algorithm - from solving the heat equation to optimisation in Hilbert spaces - and some applictaions, including imaging and predictive control problems. Then the theory of the Proximity operator is covered, and its connection to ADMM, along with examples pertinent to high-dimensional statistical signal processing. The introduction of the Prox operator is motivated by how it allows simple calculation of the sub-problems in ADMM. Thus, difficult non-smooth optimisation can be quickly performed in simpler stages.

Chapter 5 is new material and covers the application of and optimisation on graphs.

Chapter 6 introduces a model of

Chapter 7 covers compressive inference

Chapter 8 covers the application of our methods to real world data sets.

Finally, Chapter 9 covers group testing and our approach to .

Despite the ubiquity, capacity and apparent efficacy, modern communication systems are wasteful, inefficient and in need of reform. Most of the bits of data collected by our sensing systems are unessential, and only serve to necessitate data compression wasting computation time before transmission. For example, people regularly use a camera with a resolution of several megapixels only to upload a file of a few kilobytes to Facebook. Devices are unable to make dynamic decisions about how to transmit this data, leading to both spectral exhaustion on some frequencies whilst much of the available radio spectrum lies fallow.

This project addresses these issues, by reviewing a novel acquisition and decompression framework for data: a way in which we need only sense the most informative bits of data. This framework is then applied to the problem of sensing spectral opportunities dynamically, to make better use of available spectrum.

The key uniting both these applications is that data and spectra are *sparse*: that is they have a representations which are 'smaller' than their respective dimension. For example, images and audio can be compressed into file formats much smaller than when initially recorded (compare the relative sizes of bitmap and JPEG images).

The sole focus of this research is to use the sparsity of the spectrum to uncover transmission opportunities, allowing better use of spectrum more generally.

We are motivated by the need to send more data over wireless networks, whilst at the same time

having a constrained frequency set over which to transmit this information. This issue could be alleviated by users dynamically allocating spectrum on a per-transmission basis: without the ability to gain knowledge of spectral conditions this can never become a reality however.

The requirement for increasing bandwidth isn't just a pressing issue for today: in the next decade it is forecast that network operators will need to provide for three-orders of magnitude (1000 times) more capacity. Demand is continually outstripping supply - motivated by the ubiquity of smartphones, and the consumers appetites for media.

At the same time as this demand for ever more data, there is an increasing scarcity of radio spectrum over which to transmit. New frequencies are rarely cleared for commercial purposes, and when they are they go for high prices. A decade ago the UK auction for 3G spectrum licenses raised an estimated 22.4 billion pounds for the UK treasury, indicating the seriousness of the market players requirements for new spectrum. The recent 4G spectrum auction raised 2.3 billion pounds with initial networks being rolled out by the end of 2013.

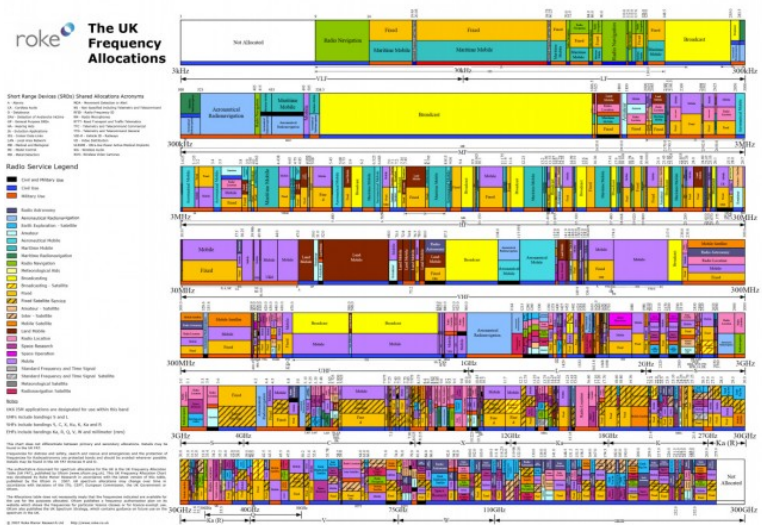

Figure 1.1: A digram of current Spectral allocation [**?**]

However, a closer inspection of the frequency allocation suggests this scarcity is artificial, it's more a product of regulatory oversight over time. As the constraints on spectrum requirement became more complex, so did the solutions to that problem - at the cost of leaving much of the spectrum idle for most of the time.

For example: much of the spectrum is allocated to TV broadcast, radio broadcast and mobile. However, if we look closer, the allocations aren't even for specific companies - they're simply categories. Within these, OFCOM may have many licensees within each category.

Also interesting to note is how much frequency the Government allocates to itself (the red bar underneath the blocks indicates Government use). Compare this to the actual utilisation of spectrum: much of it is not used at all. Figure 1.2 shows a snapshot of frequency utilisation in three diverse

locations in the UK over te radio specturm, note that many frequencies are not utilised (coloured blue) whilst others have significant activity (coloured yellow). Note that the plot for Southwark (central London) is barely different from Braddock - a rural area.

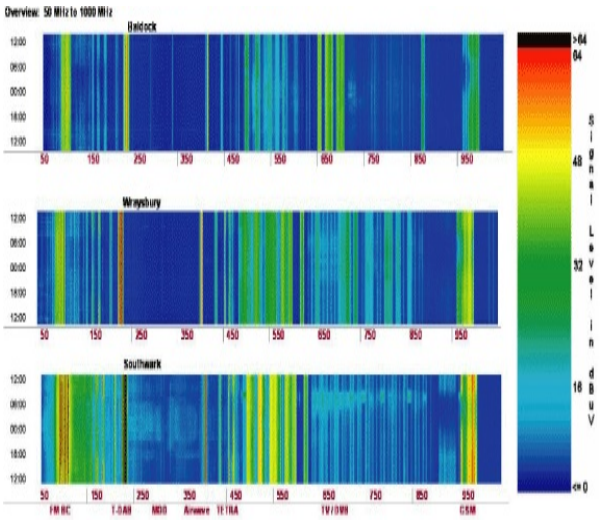

Figure 1.2: A snapshot of frequency utilisation in various areas: many frequencies are not used at all, whilst there is significant activity on others [17]

How do we then go about solving this issue - how can we obtain the most significant bits of information from our sensing mechanism, whilst obviating the need to compress the data once we are done? How do we dynamically assign spectrum? The work of Candes, Tao [22] and Donoho [38], has shown that instead of measuring the information we require directly (and then compressing it), we can measure 'holographic' and non-linear random projections between our measurement space and the space where our data is sparse. This requires only the knowledge that the signal is compressible via some transform - both the acquisition protocol and the reconstruction algorithm are agnostic to the type of signal. What is surprising is that the sampling kernels are fixed independently of the signal, are non-adaptive and these projections are sufficient to reconstruct the signal - as if we had an Oracle to tell us where the non-zero components of our signal are.

This work has had a large impact in medical imaging since it's inception: for example, it's now possible to take an image of a patient's heart within a single breath, as well as dynamic imaging of the heart ([39] figures 7 and 9).

Modern digital signal processing techniques (such as modulation techniques) are far more spectrally efficient than their historic analogue counterparts, which has in part contributed to the spectrum crisis. All this is changing though: from the beginning of 2013 all TV in the UK will transmitted digitally. Historically, television in the UK was broadcast using analogue signals requiring 32 multiplexes. Digital TV requires 6 multiplexes, on the other hand.

This freeing up of TV frequencies represents an opportunity: these frequencies have good propagation characteristics (they suffer less with free space path loss relative to higher frequencies), whilst

sill providing good bandwidth for data transmission. These TV frequencies are being opened up to civilian and commercial users: spectral holes will be able to be exploited opportunistically by devices, so long as they don't interfere with the reception of TV. Historically, this is the single largest gift of new spectrum, and because there is no requirement for licensing this spectrum is free.

As with all technological innovations, this will not only improve existing infrastructure but also new classes of devices to transmit, for instance; applications such as passive sensor networks) which only need spectrum intermittently to transmit monitoring results), inter-vehicle communication for real time traffic monitoring and wireless internet at broadband data rates have all been proposed.

Despite all of this hype, dynamic spectrum access won't become a reality unless spectral holes can be robustly detected. The requirement that secondary users exploit the new spectrum politely, without interference to primary user makes spectrum sensing essential to TV white-space (TVWS) technologies. The realisation of any Cognitive Radio standard (such as IEEE 802.22), requires the co-existence of primary (TV users) and secondary (everybody else who wants to use TVWS spectrum) users of the frequency spectrum to ensure proper interference mitigation and appropriate network behaviour.

Users of TVWS (Cognitive Radios) must sense whether spectrum is available, and must be able to detect very weak primary user signals. Furthermore they must sense over a wide bandwidth (due to the amount of TVWS spectrum proposed), which challenges traditional Nyquist sampling techniques, because the sampling rates required are not technically feasible with current RF or Analogue-to-Digital conversion technology.

Sensing should enable devices to detect the presence of TV signals in a band and provide smart and adaptive (and possibly distributed) solutions to band identification.

Spectrum sensing should involve:

1. Sensing to detect white spaces.
2. Co-existence with similar devices.
3. Frequency monitoring of other devices.
4. Interference management.
5. Spectrum mobility and transmission power control when needed.

As described earlier, the available spectrum is highly underutilised, and can be thought of as a collection of narrowband transmissions over a wideband channel. As such, the spectrum we're sensing is sparse. This makes it an ideal candidate for sparse recovery techniques such as Compressive Sensing.

The report is divided into three chapters: the remainder of this chapter describes methods for sensing narrowband signals (i.e. channels where the frequency response is approximately flat, and

where the bandwidth is smaller than the coherence bandwidth of the channel), and the limitations of these are highlighted for the problem of sensing spectrum for Cognitive Radios.

Classical and Compressive Sensing are then contrasted, including the main ideas such as incoherence and the Restricted Isometry Property, and illustrating the number of samples required for full reconstruction. Somme approaches to solving the optimisation problems posed by the new framework are also discussed.

Chapter 2 introduces Group Testing, and covers new work which has been accepted for publication at Allerton 2014. After some preliminary remarks the Capacity of a Group Testing problem is defined. Then, previous work on variations of an algorithm by Hwang are discussed, including upper and lower bounds on Capacity. Finally, a new algorithm is presented along with an analysis of the average number of tests the algorithm will execute.

The focus of Chapter 3 is again compressive sensing, this time in a distributed setting - given a connected network of nodes, how can we organise sensing to reconstruct a wideband signal? The chapter begins by discussing various signal models in the literature, and justifies the use of a single model. Then the sensing model is presented as a multinode extension of the Modulated Wideband Converter ([75]). There follows an extended discussion of constrained convex optimisation, and an introduction to the Alternating Direction Method of Multipliers. Finally, how to sense the requisite signals and use a distributed setup to solve the system is explained. Some tentative results of simulations are also presented.

CHAPTER 2

CLASSICAL SENSING

2.1 Introduction

This chapter deals with classical, or Nyquist sensing and some approaches to spectrum sensing based on this sensing paradigm. This is the sensing paradigm based upon the maximum frequency in a signal.

Section **??** outlines some of the mathematical rerequisites underlying modern signal procesing: representations in time and frequency of analog signals and the Nyquist theorem for sampling bandlimited signals. Conditions for perfect recovery of such a signal from a finite set of samples are also discussed.

Section 2.3 discusses techniques of spectrum based on this theory.

A necessary condition for perfect signal recovery, is that the signal be sampled at (or above) a rate equivalent twice the highest frequency contained in the signal. For example, an FM radio at 445 MHz, would need to be sampled at 890MHz, or above. This theorem was established by Nyquist and Shannon in the 20s and has formed the basis for all radio technology since.

As it relates to TV white spaces, which have a large bandwidth, this means that classical sensing schemes will require a large (and expensive) sampling rate. This chapter discusses some spectrum sensing schemes based upon Nyquist sensing. This is mainly for comparison with Compressive Techniques, described in Chapter 3.

The simplest approach to spectrum sensing, is simply to spilt the Wideband channel into a series of contiguous narrowband channels and measure the energy in each channel independently. Should the energy observed exceed a threshold (depending upon the noise variance), the channel is declared occupied; energy below the threshold is declared unoccupied. However simple, this approach has several drawbacks - most notably in the low SNR regime, it is impossible to distinguish between occupied and unoccupied states, no matter how long the channel is observed. Also, for a contiguous

set of narrowband channels, a hypothesis test must be done in each to decide whether the channel is occupied, or not. The number of tests grows exponentially with the number of channels, thus rendering obsolete the relative simplicity of single-band energy detection.

There are more sophisticated methods of spectrum sensing, most notably cyclostationary feature detection and Matched Filtering. Cyclostationary feature detection aims to detect periodic signals, in periodless white noise. It is superior to energy detection, in that it can distinguish between primary and secondary user transmissions, but it requires prior knowledge of the periods of the signals it will detect.

Communication theory tells us that a Matched Filter is the optimum detector for a signal is noise. This comes at a considerable implementational complexity. Matched filtering is subject to similar prior knowledge issues as other feature detection schemes: if there is a mismatch between the expected features and the signal, the scheme performs poorly. Often this difficulty is overcome by having a bank of different filters, and passing the signal through each of them, at the cost of more complexity.

Finally we discuss distributed approaches to wideband spectrum sensing. In particular equal gain combining, where a network of sensors each performing energy detection combines their measurements to gain statistical strength. We discuss several combination rules (in particular the AND and OR rules), as well as centralised and decentrailised approaches to decision making. We show that the probability of false detection goes down exponentially with the number of sensors, but that the SNR-wall phenomenon and exponential number of tests still persists.

2.2 Classical Sensing

Classically, for perfect signal reconstruction, we must sample a signal such that the sampling rate must be at least twice the maximum frequency in the bandlimited signal. The continuous time signal can then be recovered using an appropriate reconstruction filter (e.g. a sinc filter). For example, we can represent a sampled continuous signal as a multiplication of the signal with a train of Dirac delta functions at multiples of the sampling period T.

Informally, for a bandlimited signal (of bandwidth W) with compact frequency support on $(-B, B)$, a set of ordinates with spacing $\frac{1}{2B}$ seconds are enough to completely specify the function.

In [103], Shannon formalised this intuition, howver this idea had been implied in the work of Nyquist in 1928 [82].

Definition 2.2.1 (Fourier Transform)**.** *We write the Fourier transform:*

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-i\omega t}\, dt \tag{2.1}$$

where $\omega = 2\pi f$ is the angular frequency.

with inverse:

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{i\omega t}\, dt \tag{2.2}$$

Definition 2.2.2 (Bandlimited Signal)**.** *We say a signal is bandlimited if there exists a finite B such that:*

$$\int_{-\infty}^{\infty} x(t)\, e^{-i\omega t}\, dt = 0 \; |f| > B \tag{2.3}$$

This can be expressed in the frequency domain as:

$$X(\omega) = X(\omega)\, \Pi\left(\frac{f}{2B}\right) \tag{2.4}$$

Where

$$\Pi(x) = 1 \; if |x| \leq 1 \tag{2.5}$$

so a bandlimited function may be represented as:

$$x(t) = \int_{-B}^{B} X(f)\, e^{i\omega t}\, dt \tag{2.6}$$

Definition 2.2.3 (Periodic function)**.** *We say a function is periodic if there exists a T such that:*

$$x(t) = x(t - T) \tag{2.7}$$

We can represent the samples of the signal via the formula:

$$x(nT) = \text{Ш}(x)\, x(t) \tag{2.8}$$

where

$$\text{Ш}(x) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{2.9}$$

It should be noted that the Fourier Transform of (2.2) (the Shah, or comb function) is another Shah function:

$$\text{Ш}(\omega) = \frac{1}{T} \sum_{s=-\infty}^{\infty} \delta\left(s - \frac{1}{T}\right) \tag{2.10}$$

Theorem 2.2.4 (Nyquist [82])**.** *A signal $x(t)$ with bandwidth B may be sampled at a rate greater than $\frac{1}{2B}$, and can be recovered from an infinite set of samples with*

$$x(t) = \text{Ш}(x)\, x(t) \star \hat{\Pi}(B) \tag{2.11}$$

$$x(t) = \sum_{n=-B}^{B} x(nT)\, sinc\left(\frac{t_n T}{T}\right) \tag{2.12}$$

Definition 2.2.5 (Wide-sense Stationary Signal)**.** *Given a (bandlimited) signal $x(t)$, we say it is wide sense stationary, if the following conditions hold:*

- *For all t:*

$$\mathbb{E}x(t) = \mu \; \forall t \tag{2.13}$$

- *For all t_1, t_2 there exists a function $C_x(\circ)$ such that:*

$$\mathbb{E}\big(x(t_1) - \mu\big)\big(x(t_2) - \mu\big) = C_x(t_1 - t_2) \tag{2.14}$$

i.e. the mean of the signal must be constant, and the autocorrelation must depend only on the lag.

From this we can define the power spectral density of the signal $x(t)$:

Definition 2.2.6 (Power Spectral Density)**.**

$$S_{xx}(\omega) = \mid X(\omega) \mid^2 \tag{2.15}$$

where $X(\omega)$ is the Fourier transform of $x(t)$.

Definition 2.2.7 (Autocorrelation)**.** *The autocorrelation of a signal $R_{xx}(t_1, t_2)$ is defined as:*

$$R_{xx}(t_1, t_2) = \mathbb{E}x\big(t_1 - \mu\big)\, x\big(t_2 - \mu\big) \tag{2.16}$$

For wide sense stationary signals, $R_{xx}(t_1, t_2) = R_{xx}(t_1 - t_2, 0) = R_{xx}(\tau)$

The problem of spectrum sensing [128] is to decide whether a particular band is available, or not. That is, we wish to discriminate between the following two hypotheses:

$$H_0 : y[n] = w[n], \; \forall n = 1, \ldots, N \tag{2.17}$$

$$H_1 : y[n] = x[n] + w[n], \; \forall n = 1, \ldots, N \tag{2.18}$$

where x is the (deterministic) primary user's signal, having a specific structure which stems from modern coding and modulation techniques, w is additive white Gaussian noise and y is the received signal.

Any detection strategy is a function, $f : \mathbb{R}^n \rightarrow \{0, 1\}$, mapping the output of sensing to $\{0, 1\}$. 0 means the received signal is noise, whilst 1 means that a Primary User signal is present.

To decide whether the observations $\mathbf{y}$ were generated under H_0 or H_1 is accomplished by forming a test statistic $\Gamma\big(y\big)$ and then comparing this statistic with a predefined threshold λ. Both classical methods, where the hypotheses are assumed to be deterministically true and the goal is to minimise

the false detection probability, and Bayesian methods, where it is assumed that the source selects the true hypothesis at random according to some prior probabilities, agree that the test statistic should be likelihood ratio:

$$\Gamma\left(\mathbf{y}\right) = \frac{p\left(\mathbf{y} \mid H_1\right)}{p\left(\mathbf{y} \mid H_0\right)} \tag{2.19}$$

where a large value of Γ implies that we accept H_0.

The performance of a detector is quantified in terms of the probability of detection

$$P_D = Pr\left(\Gamma\left(\mathbf{y}\right) > \lambda \mid H_1\right) \tag{2.20}$$

and the probability of false alarm

$$P_{FA} = Pr\left(\Gamma\left(\mathbf{y}\right) > \lambda \mid H_0\right) \tag{2.21}$$

By varying λ the operating point of a detector can be chosen anywhere along its receiver operating characteristics curve.

2.3 Classical Sensing Techniques

There are several proposed spectrum sensing methods that enable cognitive radios identify bands and perform dynamic frequency selection. Some of the common (narrowband) spectrum sensing techniques are described below.

2.3.1 Energy Detection

Energy detection is the simplest form of spectrum sensing: this method simply compares the signal energy in a frequency band to a pre-defined threshold. If the threshold is exceeded the band is declared occupied. This method is quite generic as receivers need no knowledge of the primary users signal, and in theory, this form of detection works irrespective of the type of Primary User signalling used. Energy detect ions is a common method for the detection of unknown signals in noise, due to low computational and implementation complexity.

A typical implementation for energy detection would be to centre a bandpass filter on the band of interest, followed by a squaring device to measure the received energy and an integrator to determine the observation interval. Finally the output of the integrator is compared with a threshold to determine the presence of a signal. This threshold is determined based upon the noise variance of the channel. I.e. we have a decision metric of the following form:

$$M = \sum_{n=0}^{N} |y\left[n\right]|^2 \tag{2.22}$$

Theorem 2.3.1. *Modelling the signal and noise as zero-mean Gaussian random variables with variances σ_s, and σ_n respectively, we can derive expressions for the metric, the detection probability and the false alarm probability under the rule* (2.2) *above [128]:*

$$M = \begin{cases} \frac{\sigma_w^2}{2}\chi_{2N}^2 & \text{under } H_0 \\ \frac{\sigma_w^2+\sigma_s^2}{2}\chi_{2N}^2 & \text{under } H_1 \end{cases} \tag{2.23}$$

$$P_D = 1 - \Gamma\left(1, \frac{\lambda}{1+\frac{\sigma_s^2}{\sigma_w^2}}\right) \tag{2.24}$$

$$P_{FA} = 1 - \Gamma\left(1, \frac{\lambda}{\sigma_w^2}\right) \tag{2.25}$$

Where $\Gamma(1,x)$ is the incomplete gamma function.

From these equations it's clear to see that the performance of energy detection based sensing faces challenges at low SNR values. See [128] figure 3 for curves quantifying the performance.

Also energy detectors perform poorly under extreme fading conditions as they are unable to distinguish primary users and noise. Further this type of detector is not efficient at detecting spread spectrum signals.

For energy detection we wish to maximise P_D subject to a constraint on P_{FA}. This is done via a threshold λ, which trades off these two probabilities.

Choosing λ requires knowledge of the Primary User transmissions, as well as estimates of the noise power. Given these, calculating the optimal λ is straightforward [124]. Estimating the PU power is dependent on the radio environment between the PU transmitter and the CR. Noise power estimation isn't flawless and a small noise power estimation error can cause significant performance loss [52], [95].

Noise power uncertainty can be mitigated by using an adaptive algorithm [130], or an a variant of the MUSIC algorithm which separates signal and noise subspaces [85]. These significantly increase the complexity of the energy detector, making it less attractive relative to other methods.

A more serious concern for energy detection is the SNR wall [112]: an SNR below which an energy detector will fail to detect the presence of a PU signal no matter how long the detector observes the channel. This is because at low SNRs the PU signal is no longer well separated from the noise. There has been some work in overcoming this wall using cross correlation between multiple antennas [86].

2.3.2 Cyclostationary Feature Detection

Because the signals used in practical communication systems contain distinctive features that can be exploited for detection, it is possible to achieve a detection performance which substantially surpasses the energy detector [125], [60]. This is in contrast to the predictions of information theory where maximum entropy signals will be statistically white and Gaussian (if this were the case, then

we could do no better than the energy detector). More importantly, known signal features can be exploited to estimate unknown parameters such as noise power.

Examples of well known patterns include pilot signals and spreading sequences. Other examples include preambles and midambles: known sequences transmitted before and in the middle of each slot, respectively. Others include redundancy added by coding, modulation and burst formatting used by the transmitter.

This method of cyclostationary feature detection exploits cyclostationary features of received signals: man made periodicity in the signal (for example symbol rate, chip rate, cyclic prefix etc) or its statistics - mean, autocorrelation. A cyclic correlation function is used instead of PSD (or autocorrelation sequence) for detecting signals present in a given spectrum. This is able to differentiate noise from primary users signals since noise is wide-sense stationary with no correlation but modulated signals are cyclostationary due to the redundancy of signal correlations.

For clarity, the random processes encountered by a cognitive radio will have a period in both expectation and autocorrelation:

$$\mathbb{E}(t) = \mathbb{E}(t + mT) = \mathbb{E}[x(t)] \tag{2.26}$$

$$\mathbb{R}(\tau) = \mathbb{E}\left[x(t)\, x\left(\bar{t} + \tau\right)\right] \tag{2.27}$$

where t is time, τ is the autocorrelation lag, $x(t)$ is the random process we are considering and m is an integer.

Due to the periodicity of the autocorrelation, it can be expressed as a Fourier series over integer multiples of the fundamental frequency in the signal as well as integer multiples of sums and differences of this frequency:

$$\mathbb{R}(\tau) = \sum_{\alpha} r(\alpha, \tau)\, e^{2\pi j \alpha t} \tag{2.28}$$

with Fourier coefficients:

$$r(\alpha, \tau) = \frac{1}{T} \int_0^T x\left(t + \frac{\tau}{2}\right) x\left(\bar{t} + \frac{\tau}{2}\right) e^{-2\pi j \alpha t}\, dt \tag{2.29}$$

where α is the cyclic frequency

From this we can define the Cyclic Power Spectrum of the signal:

$$S(f) = \int_{-\infty}^{\infty} r(\alpha, \tau)\, e^{-2\pi j f \tau}\, d\tau \tag{2.30}$$

For a fixed lag τ, (2.3.2) can be re-written as:

$$R_{xx}(t, \tau) = R_{xx}(\tau) + \sum_{\alpha} r(\alpha, \tau)\, e^{2\pi j \alpha t} \tag{2.31}$$

i.e. a part dependent on the lag only (the cyclic frequency is zero), and a part which is a periodic function of time.

Under both hypotheses, (2.2), (2.2), the continuous portion of the signal exists, but the cyclostationary portion only exists under (2.2) when $\alpha \neq 0$. Thus we only need to test for the presence of a cyclostationary component.

To this end re-write the hypotheses as:

$$H_0 : y[n] = S_w^\alpha[n], \text{n} = 1 \dots N \tag{2.32}$$

$$H_1 : y[n] = S_x^\alpha[n] + S_w^\alpha[n], \text{n} = 1 \dots N \tag{2.33}$$

where S_x^α is the CPS of white noise which is zero for $\alpha \neq 0$. Using the test statistic:

$$\chi = \sum_{\alpha \neq 0} \sum_n S_x^\alpha \overline{S_x^\alpha} \tag{2.34}$$

we can formulate the cyclostationary detector as:

$$d = \begin{cases} 0 & \chi < \lambda \\ 1 & \chi \geq \lambda \end{cases} \tag{2.35}$$

where λ is some pre-determined threshold [48].

The advantages of this type of sensing over energy detection are that its possible to distinguish primary user transmissions (as well as distinguish between different PU signals) [66]. It is also possible to distinguish noise from PU signals as the noise spectrum has no cyclic correlation [19], [18]. However, cyclic frequencies have to be assumed to be known [48].

2.3.3 Matched Filtering

If all the probability distributions and parameters - noise variance, signal variance, channel coefficients etc - are known under both hypotheses, and the signal to be detected is perfectly known then the optimal test statistic is a matched filter [19], [128].

A matched filter is the convolution of a test signal with a template signal (or window) and detects the presence of the template in the unknown signal (as the convolution measures the overlap of two signals).

For example: for a given TV signal, $r(t)$ defined over $0 \leq t \leq T$ the corresponding matched filter is $h(t) = r(T - t)$.

A test statistic can be formed by sampling the output of the filter every nT seconds and choosing 2.2 if the statistic is below some threshold and 2.2 otherwise.

When compared to other methods, matched filtering takes a shorter time to achieve a threshold probability of false alarm. However, matched filtering requires that radios demodulate received

signals, and so requires perfect knowledge of primary users signalling features. Matched filtering also requires a prohibitively large power consumption, as various algorithms need to be executed for detection.

The paper [9], compares the performance of Energy Detection, Matched Filtering and Cyclostationary detection. It concludes that cyclostationarity based detection has the best performance (based on a lower P_{FA} for a given P_D), as this for of detection is naturally insensitive to noise uncertainty as the test statistic for cyclic detection doesn't require knowledge of the noise variance.

2.3.4 Distributed Approaches to Spectrum Sensing

Spectrum sensing can also be achieved with multiple nodes, co-operating on the sensing task to efficiently decide which part of the spectrum to use.

There are several advantages to considering distributed approaches to spectrum sensing. As the received spectrum is highly variable geographically, a device caught in a deep fade may inadvertently decide that a frequency which is occupied by a Primary User (PU) is unoccupied. By performing sensing with a geographically distributed network of nodes, the estimate can be improved via spatial diversity. Observing the signal through multiple independent channels mitigates the effect of multipath fading and shadowing for any single node. This improved statistical accuracy is paid for by the increased system complexity and communication overhead between nodes and/or a fusion centre.

By utilising multiple sensors, the problem of multiple hypothesis testing (described in section **??**) may be alleviated. It may be more practical to divide the band into multiple sub-bands which are sensed individually, this allows for simpler and lower powered sensor architectures. In [84] Oksanen et al propose a measure of spatial diversity as the maximum slope of the probability of missed diction curve on the logarithmic scale. This definition is insensitive to the number of samples used, and is low for correlated channels. This definition allows the system designer to avoid a policy of all sensors sensing all sub-bands, whilst also avoiding the problem of having only a single sensor on each sub-band.

In the previous section (**??**), we described several techniques with which a single sensor could decide whether a PU was present, or not. Any of these techniques can be used by the sensors in a network, and either the results are combined via a data fusion technique or local decisions are made based upon communication between neighbouring nodes. For example in [29], Cho and Narieda describe a weighted liner combing method cooperative cyclostationary feature detection. The technique involves individual nodes estimating the cyclic autocorrelation function of the spectrum, and these estimated correlation functions are then combined - as opposed to the estimates of the signal being combined.

However, far more common is simple energy detection at each node. followed by a voting rule. This system is described by Ma et al in [**?**]. In this scenario each sensor takes energy measurements as described in section (2.3.1) and either are combined at a Fusion Centre (FC), or shared locally

between neighbouring nodes. A simple, and low communication complexity data fusion method is Hard Combining. Here, each node simply transmits a binary decision for each sub-channel it senses, and a voting rule is applied to decide on sub-band occupancy.

Let $u_i \in \{0,1\}$ be the decision made by Cognitie Radio (CR), and $u \in \{0,1\}$ be the global decision. A 0 indicates PU absence. Two common voting rules are the AND rule, where a sub-band is declared occupied if $u_i = 1$ for all i, and the OR rule, where a band is declared occupied if $u_i = 1$ for any i.

2.3.5 Limitations

The methods described above, are appropriate for sensing whether a single channel is available for transmission, based upon the result of measurements of that channel. However, Cognitive Radios aim to exploit spectral holes in a wide band spectrum (i.e. a channel whose frequency response is not flat over the bandwidth) and will usually have to make a decision regarding transmission from measurements from this type of channel.

There are two proposed approaches to this: Multiband sensing and Compressive Sensing. Multiband sensing splits the wideband spectrum into a number of independent (not necessarily contiguous) sub-channels (whose frequency response is flat), and performs the hypothesis test for each sub-channel. However, in practice, there are correlations across sub-channels that this method fails to address. For example, digital TV signals are transmitted as spread spectrum signals so that primary user occupancy is correlated across channels. A related issue is that noise variance could be unknown but correlated across bands. Binary hypothesis testing then fails in this case, needing to be replaced by composite hypothesis tests which grow exponentially with the number of sub-channels. Such problems are typically non-convex and require prohibitively complex detectors. We will describe Compressive Sensing in Chapter 3

CHAPTER 3

COMPRESSIVE SENSING

3.1 Introduction

This chapter discusses Compressive Sensing Compressive Sensing (CS): an alternative signal acquisition method to Nyquist sampling, which is capable of accurate sensing at rates well below those predicted by the Nyquist theorem. This strategy hinges on using the structure of a signal, and the fact that many signals we are interested in can be compressed successfully. Thus, Compressive Sensing acquires the most informative parts of a signal directly.

The first section surveys the mathematical foundation of CS. It covers the Restricted Isometry Property and Stable Embeddings - necessary and sufficient conditions on which a signal can be successfully acquired. Informally, these conditions suggest that the sensing operator preserves pairwise distances between points when projected to a (relatively) low dimensional space from a high dimensional space. We then discuss which operators satisfy this condition, and why. In particular, random ensembles such as the Bernoulli/Gaussian ensembles satisfy this property - we discuss how this can be applied to the problem of wideband spectrum sensing. We also survey a small amount of the theory of Wishart matrices.

The section concludes with an overview of reconstruction algorithms for CS - methods for unpacking the original signal from its compressed representation. We give insight into the minimum number of samples for reconstruction. We survey convex, greedy,and Bayesian approaches; as well as algorithms which are blends of all three. The choice of algorithm is affected by the amount of undersampling required for system performance, the complexity of the algorithm itself, and the desired reconstruction accuracy. In general: the more complex the algorithm, the better the reconstruction accuracy. The allowable undersampling depends upon the signal itself, and the prior knowledge available to the algorithm. Greedy algorithms are the simplest class, making locally

optimal updates within a predefined dictionary of signal atoms. Greedy algorithms have relatively poor performance, yet are the fastest algorithms. Convex algorithms are the next most complex, based upon minimising a global functional of the signal. This class is based upon generalised gradient descent, and has no fixed number of steps. Finally Bayesian algorithms are the slowest and most complex, but offer the best performance - both in terms of undersampling (as these algorithms incorporate prior knowledge in an elegant way), and in terms of reconstruction accuracy.

We survey some distributed approaches to compressive sensing, in particular some models of joint sparsity, and joint sparsity with innovations.

Finally, we survey some of the approaches to wideband spectrum sensing based upon compressive sensing. In particular we survey the Random Demodulator and the Modulated Wideband converter. Both of these systems make use of low frequency chipping sequences (also used in spread spectrum communication systems). These low frequency sequences provide the basis for CS - several different sequences each convoluted with the signal are sufficient to accurately sense the signal.

3.2 Preliminaries

Compressive sensing is a modern signal acquisition technique in which randomness is used as an effective sampling strategy. This is contrast to traditional, or Nyquist, sampling, which requires that a signal is sampled at regular intervals. The motivation for this new method comes from two disparate sources: data compression, and sampling hardware design.

The work of Shannon, Nyquist and Whittaker [67, 119] has been an extraordinary success - digital signal processing enables the creation of sensing systems which are cheaper, more flexible, and which offer superior performance to their analogue counterparts. For example, radio dongles, such as those which support the RTLSDR standard, which can process millions of samples per second can now be bought for as little as £12. However, the sampling rates underpinning these advances have a doubling time of roughly 6 years - this is due to physical limitations in Analogue to Digital conversion hardware. Specifically, these devices will always be limited in bandwidth and dynamic range (number of bits), whilst applications are creating a deluge of data to be processed downstream.

Data compression means that in practice many signals encountered 'in the wild' can be fully specified by much fewer bits than required by the Nyquist sampling theorem. This is either a natural property of the signals, for example images have large areas of similar pixels, or as a conscious design choice, as with training sequences in communication transmissions. These signals are not statistically white, and so these signals may be compressed (to save on storage). For example, lossy image compression algorithms can reduce the size of a stored image to about 1% of the size required by Nyquist sampling. In fact the JPEG standard uses Wavelets to exploit the inter-pixel redundancy of images.

Whilst this vein of research has been extraordinarily successful, it poses the question: if the reconstruction algorithm is able to reconstruct the signal from this compressed representation, why

collect all the data in the first place, when most of the information can be thrown away?

Compressed Sensing answers these questions, by way of providing an alternative signal acquisition method to the Nyquist theorem. Specifically, situations are considered where fewer samples are collected than traditional sensing schemes. That is, in contrast to Nyquist sampling, Compressive Sensing is a method of measuring the informative parts of a signal directly without acquiring unessential information at the same time.

These ideas have not come out of the ether recently however. Prony, in 1795, [37], proposed a method for estimating the parameters of a number of exponentials corrupted by noise. This work was extended by Caratheodory in 1907 [24], who proposed in the 1900s a method for estimating a linear combination of k sinusoids for the state at time 0 and any other $2k$ points. In the 1970's Geophysicists proposed minimising the ℓ_1 norm to reconstruct the structure of the earth from seismic measurements. Clarebuot and Muir proposed in 1973, [30], using the ℓ_1 norm as an alternative to Least squares. Whilst Taylor, Banks and McCoy show in [113] how to use the ell_1 norm to deconvolve spike trains (used for reconstructing layers in the earth). Santosa and Symes in [96] introduced the constrained ℓ_1 program to perform the inversion of band-limited reflection seismograms. The innovation of CS is to tells us under which circumstances these problems are tractable.

The key insight in CS, is that for signals which are sparse or compressible - signals which are non-zero at only fraction of the indices over which they are supported, or signals which can be described by relatively fewer bits than the representation they are traditionally captured in - may be measured in a non-adaptive way through a measurement system which is orthogonal to the signal's domain.

Examples of sparse signals are:

1. A sine wave at frequency ω is defined as a single spike in the frequency domain yet has an infinite support in the time domain

2. An image will have values for every pixel, yet the wavelet decomposition of the image will typically only have a few non-zero coefficients

Informally, CS posits that for s-sparse signals $\alpha \in \mathbb{R}^n$ - signals with s non-zero amplitudes at unknown locations) - $\mathcal{O}(s \log n)$ measurements are sufficient to exactly reconstruct the signal.

In practice this can be far fewer samples than conventional sampling schemes. For example a megapixel image requires 1,000,000 Nyquist samples, but can be perfectly recovered from 96,000 compressive samples in the wavelet domain [23].

The measurements are acquired linearly, by forming inner products between the signal and some arbitrary sensing vector:

$$y_i = \langle \alpha, \psi_i \rangle \tag{3.1}$$

or

$$y = \Psi\alpha \tag{3.2}$$

where, y_i is the i^{th} measurement, $\alpha \in \mathbb{R}^n$ is the signal, and ψ_i is the i^{th} sensing vector. We pass to (3.2) from (3.2), by concatenating all the y_i into a single vector. Thus the matrix Ψ has the vectors ψ_i as columns.

Note that the measurements may be corrupeted by noise, in which case our model is:

$$y = Ax + n \tag{3.3}$$

where $n \in \mathbb{R}^m$, and each component is sampled from a $\mathcal{N}(0, 1/n)$ distribution.

We require that sensing vectors satisfy two technical conditions (described in detail below): an Isotropy property, which means that components of the sensing vectors have unit variance and are uncorrelated, and an Incoherence property, which means that sensing vectors are almost orthogonal. Once the set of measurements have been taken, the signal may be reconstructed from a simple linear program. We describe these conditions in detail in the next section.

3.2.1 RIP and Stable Embeddings

We begin with a formal definition of sparsity:

Definition 3.2.1 (Sparsity)**.** *A high-dimensional signal is said to be s-sparse, if at most s coefficients x_i in the linear expansion*

$$\alpha = \sum_{i=1}^{n} \phi_i x_i \tag{3.4}$$

are non-zero, where $x \in \mathbb{R}$, $\alpha \in \mathbb{R}$, and ϕ_i are a set of basis functions of $\mathbb{R}^n$.
We can write (3.2.1) *as:*

$$\alpha = \Phi x \tag{3.5}$$

We can make the notion of sparsity precise by defining Σ_s as the set of s-sparse signals in $\mathbb{R}^n$:

$$\Sigma_s = \{x \in \mathbb{R}^n : |\text{supp}(x)| \leq s\} \tag{3.6}$$

where supp(x) *is the set of indices on which x is non-zero.*

We may not be able to directly obtain these coefficients x, as we may not posses an appropriate measuring device or one may not exist, or there is considerable uncertainty about where the non-zero coefficients are.

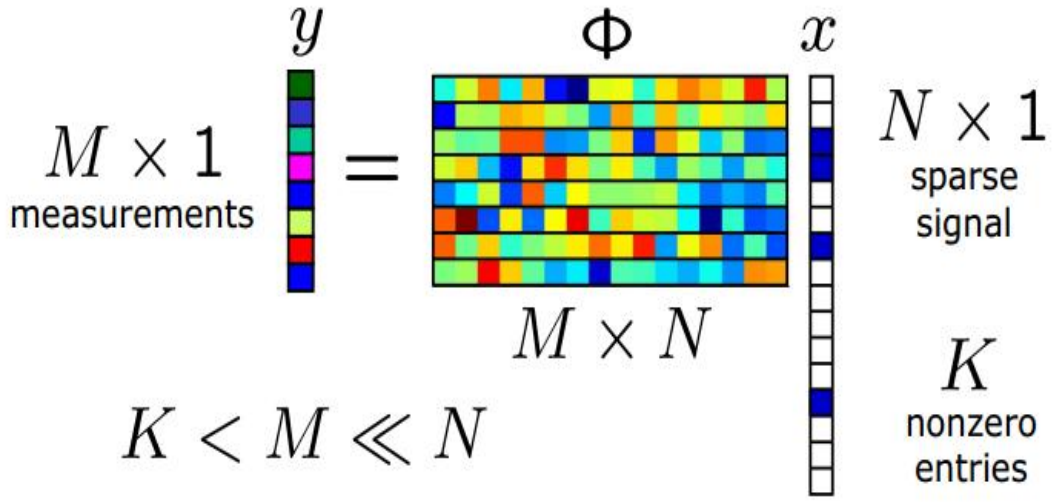

Figure 3.1: A visualisation of the Compressive Sensing problem as an under-determined system

Given a signal $\alpha \in \mathbb{R}^n$, a matrix $A \in \mathbb{R}^{m \times n}$, with $m \ll n$, we can acquire the signal via the set of linear measurements:

$$y = \Psi\alpha = \Psi\Phi x = Ax \tag{3.7}$$

where we have combined (3.2), and (3.2.1). In this case A represents the sampling system (i.e each column of A is the product of Φ, with the columns of Ψ). We can work with the abstract model (3.2.1), bearing in mind that x may be the coefficient seqeunce of the object in the proper basis.

In contrast to classical sensing, which requires that $m = n$ for there to be no loss of information, it is possible to reconstruct x from an under-determined set of measurements as long as x is sparse in some basis.

There are two conditions the matrix A needs to satisfy for recovery below Nyquist rates:

1. Restricted Isometry Property.

2. Incoherence between sensing and signal bases.

Definition 3.2.2 (RIP)**.** *We say that a matrix A satisfies the RIP of order s if there exists a $\delta \in (0,1)$ such that for all $x \in \Sigma_s$:*

$$(1-\delta)\,||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta)\,||x||_2^2 \tag{3.8}$$

i.e. A approximately preserves the lengths of all s-sparse vectors in $\mathbb{R}^n$.

Remark 3.2.3. *Although the matrix A is not square, the RIP (3.2.2) ensures that $A^T A$ is close to the identity, and so A behaves approximately as if it were orthogonal. This is formalised in the following lemma from [102]:*

Lemma 3.2.4. *Let A be a matrix which satisfies the RIP of order 2s with RIP constant δ. Then for two disjoint subsets $I, J \subset [n]$ each of size at most s, and for any vector $u \in \mathbb{R}^n$:*

$$\langle Au_I, Au_J \rangle \leq \delta \, ||u_I||_2 \left|\left|u_J\right|\right|_2 \tag{3.9}$$

where u_I is the vector with component u_i if $i \in I$ and zero elsewhere.

Remark 3.2.5 (Information Preservation)**.** *A necessary condition to recover all s-sparse vectors from the measurements Ax is that $Ax_1 \neq Ax_2$ for any pair $x_1 \neq x_2$, $x_1, x_2 \in \Sigma_s$, which is equivalent to $||A(x_1 - x_2)||_2^2 > 0$.*

This is guaranteed as long as A satisfies the RIP of order 2s with constant δ - as the vector $x_1 - x_2$ will have at most 2s non-zero entries, and so will be distinguishable after multiplication with A. To complete the argument take $x = x_1 - x_2$ in definition 3.2.2, guaranteeing $||A(x_1 - x_2)||_2^2 > 0$, and requiring the RIP order of A to be 2s.

Remark 3.2.6 (Stability)**.** *We also require that the dimensionality reduction of compressed sensing is the preservation of relative distances: that is if x_1 and x_2 are far apart in $\mathbb{R}^n$ then their projections Ax_1 and Ax_2 are far apart in $\mathbb{R}^m$. This will guarantee that the dimensionality reduction is robust to noise.*

A requirement on the matrix A that satisfies both of these conditions is the following:

Definition 3.2.7 (δ-stable embedding)**.** *We say that a mapping is a δ-stable embedding of $U, V \subset \mathbb{R}^n$ if*

$$(1-\delta)\,||u-v||_2^2 \leq ||Au - Av||_2^2 \leq (1+\delta)\,||u-v||_2^2 \tag{3.10}$$

for all $u \in U$ and $v \in V$.

Remark 3.2.8. *Note that a matrix A, satisfying the RIP of order 2s is a δ-stable embedding of Σ_s, Σ_s.*

Remark 3.2.9. *Definition 3.2.7 has a simple interpretation: the matrix A must approximately preserve Euclidean distances between all points in the signal model Σ_s.*

3.3 Incoherence

Given that we know a basis in which our signal is sparse, ϕ, how do we choose ψ, so that we can accomplish this sensing task? In classical sensing, we choose ψ_k to be the set of T_s-spaced delta

functions (or equivalently the set of $1/T_s$ spaced delta functions in the frequency domain). A simple set of ψ_k would be to choose a (random) subset of the delta functions above.

In general, we seek waveforms in which the signals' representation would be dense.

Definition 3.3.1 (Incoherence)**.** *A pair of bases is said to be incoherent if the largest projection of two elements between the sensing (ψ) and representation (ϕ) basis is in the set* $[1, \sqrt{n}]$*, where n is the dimension of the signal. The coherence of a set of bases is denoted by* μ*.*

Examples of pairs of incoherent bases are:

- Time and Fourier bases: Let $\Phi = \mathbf{I}_n$ be the canonical basis and $\Psi = \mathbf{F}$ with $\psi_i = n^{-\frac{1}{2}} e^{i\omega k}$ be the Fourier basis, then $\mu(\phi, \psi) = 1$. This corresponds to the classical sampling strategy in time or space.
- Consider the basis Φ to have only entries in a single row, then the coherence between Φ and any fixed basis Ψ will be $\sqrt{n}$.
- Random matrices are incoherent with any fixed basis Ψ. We can choose Φ by creating n orthonormal vectors from n vectors sampled independently and uniformly on the unit sphere. With high probability $\mu = \sqrt{n \log n}$. This extends to matrices whose rows are created by sampling independent Gaussian or Bernoulli random vectors.

This implies that sensing with incoherent systems is good (in the sine wave example above it would be better to sample randomly in the time domain as opposed to the frequency domain), and efficient mechanisms ought to acquire correlations with random waveforms (e.g. white noise).

Theorem 3.3.2 (Reconstruction from Compressive measurements [22])**.** *Fix a signal $f \in \mathbb{R}^n$ with a sparse coefficient basis, x_i in ϕ. Then a reconstruction from m random measurements in ψ is possible with probability $1 - \delta$ if:*

$$m \geq C\mu^2(\phi, \psi) S \log\left(\frac{n}{\delta}\right) \tag{3.11}$$

where $\mu(\phi, \psi)$ is the coherence of the two bases, and S is the number of non-zero entries on the support of the signal.

3.3.1 Random Matrix Constructions

To construct matrices satisfying definition 3.2.7, given m, n we generate A by A_{ij} being i.i.d random variables from distributions with the following conditions [34]

Condition 1 (Norm preservation)**.** $\mathbb{E}A_{ij}^2 = \frac{1}{m}$

Condition 2 (sub-Gaussian)**.** $\mathbb{E}\left(e^{A_{ij}t}\right) \leq e^{C^2 t^2/2}$

Random variables A_{ij} satisfying conditions (1) and (2) satisfy the following concentration inequality [6]:

Lemma 3.3.3 (sub-Gaussian)**.**

$$\mathbb{P}\left(\left|||Ax||_2^2 - ||x||_2^2\right| \geq \varepsilon ||x||_2^2\right) \leq 2e^{-cM\varepsilon^2} \tag{3.12}$$

Then in [6] the following theorem is proved:

Theorem 3.3.4. *Suppose that m, n and $0 < \delta < 1$ are given. If the probability distribution generating A satisfies condition* (**??**)*, then there exist constants c_1, c_2 depending only on δ such that the RIP* (3.2.2) *holds for A with the prescribed δ and any $s \leq \frac{c_1 n}{\log n/s}$ with probability $\geq 1 - 2e^{-c_2 n}$*

For example, if we take $A_{ij} \sim \mathcal{N}(0, 1/m)$, then the matrix A will satisfy the RIP, with probability [6]

$$\geq 1 - 2\left(\frac{12}{\delta}\right)^k e^{-c_0 \frac{\delta}{2} n} \tag{3.13}$$

where δ is the RIP constant, and c_0 is a constant/

3.3.2 Wishart Matrices

Let $\{X_i\}_{i=1}^r$ be a set of i.i.d $1 \times p$ random vectors drawn from the multivariate normal distribution with mean 0 and covariance matrix H.

$$X_i = \left(x_1^{(i)}, \ldots, x_p^{(i)}\right) \sim N(0, H) \tag{3.14}$$

We form the matrix X by concatenating the r random vectors into a $r \times p$ matrix.

Definition 3.3.5 (Wishart Matrix)**.** *Let*

$$W = \sum_{j=1}^{r} X_j X_j^T = XX^T \tag{3.15}$$

Then $W \in \mathbb{R}^{r \times r}$ has the Wishart distribution with parameters

$$W_r\left(H, p\right) \tag{3.16}$$

where p is the number of degrees of freedom.

Remark 3.3.6. *This distribution is a generalisation of the Chi-squared distribution: let $p = H = 1$.*

Theorem 3.3.7 (Expected Value)**.**

$$\mathbb{E}(W) = rH \tag{3.17}$$

Proof.

$$\begin{aligned}\mathbb{E}(W) &= \mathbb{E}\left(\sum_{j=1}^{r} X_j X_j^T\right) \\ &= \sum_{j=1}^{r} \mathbb{E}\left(X_j X_j^T\right) \\ &= \sum_{j=1}^{r} \left(\mathrm{Var}(X_j) + \mathbb{E}(X_j)\mathbb{E}(X_j^T)\right) \\ &= r H\end{aligned}$$

Where the last line follows as X_j is drawn from a distribution with zero mean. □

Remark 3.3.8. *The matrix $M = A^T A$, where A is constructed by the methods from section 3.3.1, will have a Wishart distribution. In particular, it will have $\mathbb{E}M = \frac{1}{m} I_n$*

The joint distribution of the eigenvalues is given by [62]:

$$p(\lambda_1, \ldots, \lambda_r) = c_r \prod_{i=1}^{r} e^{-\lambda_i} \prod_{i<j} (\lambda_i - \lambda_j)^2 \tag{3.18}$$

The eigenvectors are uniform on the unit sphere in $\mathbb{R}^r$.

3.4 Reconstruction Objectives

Compressive sensing places the computational load on reconstructing the coefficient sequnce x, from the set of compressive samples y. This is in contrast to Nyquist sampling, where the bottleneck is in obtaining the samples themselves - reconstructing the signal is a relatively simple task.

Many recovery algorithms have been proposed, and all are based upon minimising some functional of the data. This objective is based upon two terms: a data fidelity term, minimising the discrepancy between the reconstructed and true signal, and a regularisation term - biasing the reconstruction towards a class of solutions with desirable properties, for example sparsity. Typically the squared error $\frac{1}{2}\left|\left|y - Ax\right|\right|_2^2$ is chosen as the data fidelity term, whilst a number of regularisation terms have been introduced in the literature.

A particularly important functional is:

$$\arg\min_x ||x||_1 \text{ s.t } y = Ax \tag{3.19}$$

known as Basis Pursuit [27], with the following program known as the LASSO [117] as a noisy generalisation:

$$\arg\min_x \frac{1}{2}\left|\left|Ax - y\right|\right|_2^2 + \lambda ||x||_1 \tag{3.20}$$

The statistical properties of LASSO have been well studied. The program performs, both regularisation and variable selection: the parameter λ trades off data fidelity and sparsity with higher values of λ leading to sparser solutions.

The LASSO shares several features with Ridge regression [57]:

$$\arg\min_x \frac{1}{2} \left|\left| Ax - y \right|\right|_2^2 + \lambda \left|\left| x \right|\right|_2^2 \tag{3.21}$$

and the Non-negative garrote [15] used for best subset regression:

$$\arg\min_x \frac{1}{2} \left|\left| Ax - y \right|\right|_2^2 + \lambda \left|\left| x \right|\right|_0 \tag{3.22}$$

The solutions to these programs can all be related to each other - it can be shown [54], that the solution to (3.20) can be written as:

$$\hat{x} = S_\lambda \left(x^{OLS} \right) = x^{OLS} \text{sign} \left(x_i - \lambda \right) \tag{3.23}$$

where $x^{OLS} = (A^T A)^{-1} A^T y$ is the ordinary least squares solution, whereas the solution to Ridge regression can be written as:

$$\hat{x} = (1 + \lambda)^{-1} x^{OLS} \tag{3.24}$$

and the solution to the best subset regression (3.22) where $||x||_0 = \{|i| : x_i \neq 0\}$, can be written as:

$$\hat{x} = H_\lambda \left(x^{OLS} \right) = x^{OLS} \mathbb{I} \left(|x^{OLS}| > \lambda \right) \tag{3.25}$$

where $\mathbb{I}$ is the indicator function. From (3.25) and (3.24), we can see that the solution to (3.20), (3.23), translates coefficients towards zero by a constant factor, and set coefficients to zero if they are too small; thus the LASSO is able to perform both model selection (choosing relevant covariates) and regularisation (shrinking model coefficients).

Example 3.4.1 (Unitary A [45])**.**

Figure 3.2, provides a graphical demonstration of why the LASSO promotes sparse solutions. (3.20) can also be thought of as the best convex approximation of the ℓ_0 problem (3.22), as the ℓ_1-norm is the convex hull of the points defined by $||x||_p$ for $p < 1$ as $p \to 0$.

Other examples of regularisers are:

- Elastic Net: This estimator is a blend of both (3.20) and (3.21), found by minimising:

$$\arg\min_x \frac{1}{2} \left|\left| Ax - y \right|\right|_2^2 + \lambda \left|\left| x \right|\right|_2^2 + \mu \left|\left| x \right|\right|_1 \tag{3.26}$$

  The estimate has the benefits of both Ridge and Lasso regression: feature selection from the LASSO, and regularisation for numerical stability (useful in the under-determined case we

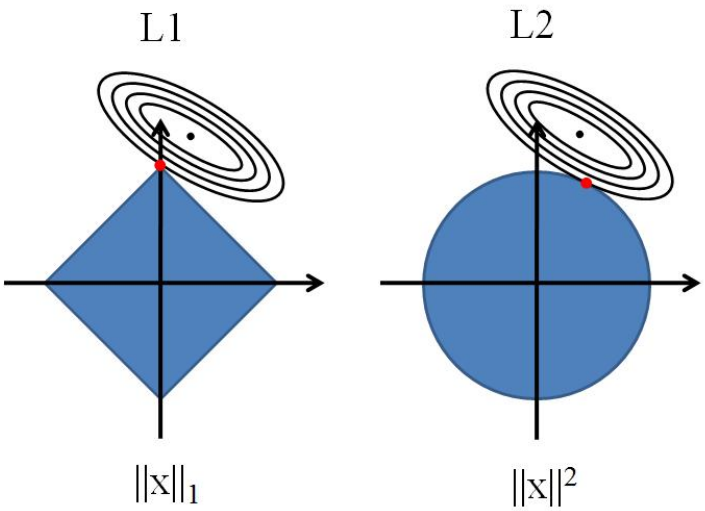

Figure 3.2: Solutions to the Compressive Sensing optimisation problem intersect the l_1 norm the points where all components (but one) of the vector are zero (i.e. it is sparsity promoting) [116]

consider here) from Ridge regression. The Elastic Net will outperform the LASSO ([132]) when there is a high degree of collfinearity between coefficients of the true solution.

- TV regularisation

$$\arg\min_x \frac{1}{2}\left|\left|Ax-y\right|\right|_2^2 + \lambda\,||\nabla x||_1 \tag{3.27}$$

This type of regularisation is used when preserving edges whilst simultaneously de-noising a signal is required. It is used extensively in image processing, where signals exhibit large flat patches alongside large discontinuities between groups of pixels.

- Candes and Tao in [21], propose an alternative functional:

$$\min_{x\in\mathbb{R}^n} ||x||_1 \text{ s.t. } \left|\left|A^T(Ax-y)\right|\right|_\infty \leq t\sigma \tag{3.28}$$

with $t = c\sqrt{2\log n}$. Similarly to the LASSO this functional selects sparse vectors consistent with the data, in the sense that the residual $r = y - Ax$ is smaller than the maximum amount of noise present. In [21] it was shown that the l_2 error of the solution is within a factor of $\log n$ of the ideal l_2 error. More recent work by Bikel, Ritov, and Tsybakov, [10], has shown that the LASSO enjoys similar properties.

3.5 Reconstruction Algorithms

Broadly reconstruction algorithms fall into three classes: convex-optimisation/linear programming, greedy algorithms, and Bayesian inference. Convex optimisation methods offer better performance, measured in terms of reconstruction accuracy, at the cost of greater computational complexity. Greedy methods are relatively simpler, but don't have the reconstruction guarantees of convex algorithms. Bayesian methods offer the best reconstruction guarantees, as well as uncertainty estimates about the quality of reconstruction, but come with considerable computational complexity.

Recovery Algorithm Summary			
Algorithm Type	Accuracy	Complexity	Speed
Greedy	Low	LowG	Fast
Convex	Medium	MediumA	Medium
Bayesian	High	HighB	Slow

3.5.1 Convex Algorithms

Convex methods cast the optimisation objective either as a linear program with linear constraints, or as a second order cone program with quadratic constraints. Both of these types of program can be solved with first order interior point methods. However, their practical application to compressive sensing problems is limited due to their polynomial dependence upon the signal dimension and the number of constraints.

Compressive Sensing poses a few difficulties for convex optimisation based methods. In particular, many of the unconstrained objectives are non-smooth: meaning methods based upon descent down a smooth gradient are inapplicable.

To overcome these difficulties, a series of algorithms originally proposed for wavelet-based image de-noising have been applied to CS, known as iterative shrinkage methods. These have the desirable property that they boil down to matrix-vector multiplications and component-wise thresholding.

Iterative shrinkage algorithms replace searching for a minimal facet of a complex polytope by a iteratively denoised gradient descent. The choice of the (component-wise) denoiser is dependent upon the regulaiser used in 3.20. These algorithms have an interpretation as Expectation-Maximisation [46] - where the E-step is performed as gradient descent, and the M-step is the application of the denoiser.

There has been a line of work on algorithms directly minimising the ℓ_0 norm [122], [87], [76]. Even though this objective function is no longer convex, the proximity operator (see definition (4.3.1)) the ℓ_0-norm can be computed, and so this minimisation can be cast as shrunk gradient descent. Thus the direct inverse problem for CS can be solved exactly, and efficiently.

procedure IST($y, A, \mu, \tau, \varepsilon$)
  $x^0 = 0$
  while $\left|\left|x^t - x^{t-1}\right|\right|_2^2 \leq \varepsilon$ **do**
    $x^{t+1} \leftarrow S_{\mu\tau}\left(x^t + \tau A^T z^t\right)$
    $z^t \leftarrow y - Ax^t$
  end while
  return x^{t+1}
end procedure

Figure 3.3: The Iterative Soft Thresholding Algorithm

procedure OMP(y, A, K, ε)
  $x^0 = 0$, $r = y$, $\Omega = \varnothing$, $i = 0$
  while $\left|\left|x^t - x^{t-1}\right|\right|_2^2 \leq \varepsilon$ **do**
    $i \leftarrow i + 1$
    $b \leftarrow A^T r$
    $\Omega \leftarrow \Omega \cup \text{supp}\left(H_1\left(b\right)\right)$
    $x \restriction_\Omega \leftarrow A^T \restriction_\Omega x$
    $x \restriction_{\Omega^c} \leftarrow 0$
    $b \leftarrow y - A \restriction_\Omega A^T \restriction_\Omega x$
  end while
  return x
end procedure

Figure 3.4: The OMP recovery algorithm

3.5.2 Greedy Algorithms

Greedy methods are another family of solutions to 3.20. They offer reduced computational complexity with correspondingly worse reconstruction quality and poorer guarantees on sparsity and undersampling than convex algorithms. Examples of this type are Orthogonal Matching Pursuit (OMP)[118], Subspace Pursuit [32], and Compressive Sensing Orthogonal Matching Pursuit (CoSaMP).

Greedy pursuit algorithms abandon exhaustive searches of the solution space in favour of locally optimal single term updates. Representative of this type of algorithm is Orthogonal Matching Pursuit (OMP), which proceeds by approximating the solution by some active set of columns from the sensing matrix A. In detail the algorithm finding the index of the largest remaining element not in the support set via matched filtering and (hard) thresholding. Then a residual in the orthogonal complement of the space spanned by the current active set is computed. This guarantees a maximal reduction in l-2 error in each iteration.

These ideas are illustrated in in figure (3.5). Assuming the previous signal approximation is along the vertical axis, the residual from the current step will be in the subspace orthogonal to this direction (the pink plane) - labelled r. The elements of A which have not been selected do not have to lie in this subspace - they are illustrated as p_1 and p_2. OMP selects the largest element of A not currently

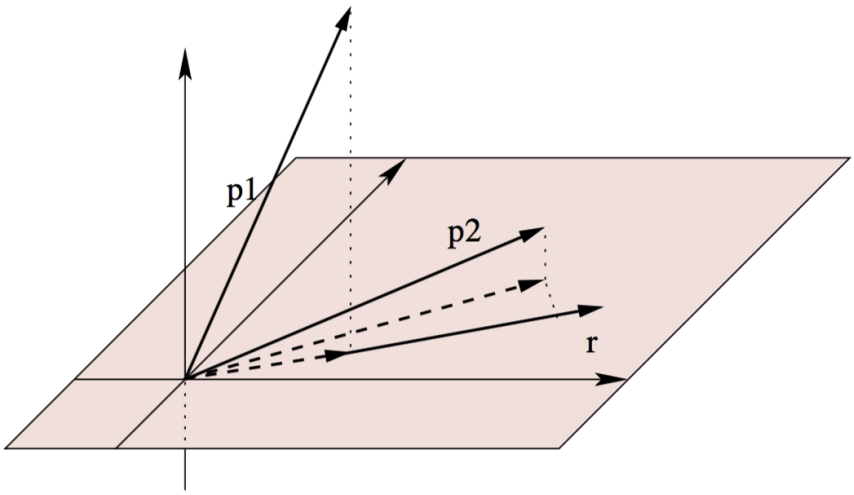

Figure 3.5: An illustration of the orthognalisation step of OMP. [11]

included, with the largest inner product with the current residual. In the figure this is p_2.

OMP has a long history in statistics and signal processing. It was first introduced in the statistics literature as stepwise least squares [50], and proposed by Pati et al in [89] as a modification of the Matching Pursuit algorithm of Mallet and Zhang [70] maintaining full backward orthogonality of the residual. The use of OMP for solving Compressive Sensing problems was studied by Tropp and Gilbert in [118].

Subspace pursuit and CoSaMP are matching pursuit algorithms specifically designed for compressive sensing problems. The main difference between these and OMP, is that they compute the T largest components in the matched filtering stage along with solving a least-squares problem to with the selected indices to identify the new support set.

Despite their computational simplicity, greedy algorithms have several drawbacks. Primarily they do not come with stable recovery guarantees, and they require a larger number of samples to recover the signal when compared to Bayesian and Convex recovery algorithms. Also, due to their greedy nature, these algorithms are not guaranteed to converge: in fact it was shown in [123] that there exist k-sparse vectors and sensing matrices A such that OMP fails to converge in k iterations.

3.5.3 Bayesian Algorithms

Bayesian methods reformulate the optimisation problem into an inference problem. These methods come with a unified theory, and standard methods to produce solutions. The theory is able to handle hyper-parameters in a elegant way, provides a flexible modelling framework, and is able to provide desirable statistical quantities such as the uncertainty inherent in the prediction.

Previous sections have discussed how the weights x may be found through optimisation methods such as basis pursuit or greedy algorithms. Here, an alternative Bayesian model is described.

procedure AMP(y, A, ε)
  $x^0 = 0$, $z^0 = A^T y$
  while $\left|\left|x^t - x^{t-1}\right|\right|_2^2 \leq \varepsilon$ **do**
    $x^{t+1} \leftarrow S_{\mu\tau}\left(x^t - \tau + A^T z^t\right)$
    $z^{t+1} \leftarrow y - Ax^t + \frac{||x||_0}{m} z^t$
  end while
  return x^{t+1}
end procedure

Figure 3.6: The AMP recovery algorithm

Equation (3.2) implies that we have a Gaussian likelihood model:

$$p\left(y \mid z,\sigma^2\right) = (2\pi\sigma^2)^{-K/2} \exp\left(-\frac{1}{2\sigma^2}\|y - Ax|_2^2\right) \tag{3.29}$$

The above has converted the CS problem of inverting sparse weight **w** into a linear regression problem with a constraint (prior) that **w** is sparse.

To seek the full posterior distribution over **w** and σ^2, we can chose a sparsity promoting prior. A popular sparseness prior is the Laplace density functions:

$$p\left(x \mid \lambda\right) = \left(\frac{\lambda}{2}\right)^N \exp -\lambda \sum_{i=1}^{N} |x_i| \tag{3.30}$$

Note that the solution the convex optimisation problem (3.20) corresponds to a maximum *a posteriori* estimate for x using this prior. I.e this prior is equivalent to using the l_1 norm as an optimisation function (see figure 3.7 [116]).

The full posterior distribution on x and σ^2 may be realised, by using a hierarchical prior instead. To do this, define a zero-mean Gaussian prior on each element of n:

$$p\left(n \mid a\right) = \prod_{i=1}^{N} \mathbb{N}\left(n_i \mid 0, \alpha_i^{-1}\right) \tag{3.31}$$

where α is the precision of the distribution. A gamma prior is then imposed on α:

$$p\left(\alpha \mid a, b\right) = \prod_{i=1}^{N} \Gamma\left(\alpha_i \mid a, b\right) \tag{3.32}$$

The overall prior is found by marginalising over the hyperparameters:

$$p\left(x \mid a, b\right) = \prod_{i=1}^{N} \int_0^\infty \mathbb{N}\left(w_i \mid 0, \alpha_i^{-1}\right) \Gamma\left(\alpha_i \mid a, b\right) \tag{3.33}$$

This integral can be done analytically and is a Student-t distribution. Choosing the parameters a, b appropriately we can make the Student-t distribution peak strongly around $x_i = 0$ i.e. sparsifying. This process can be repeated for the noise variance σ^2. The hierarchical model for this process

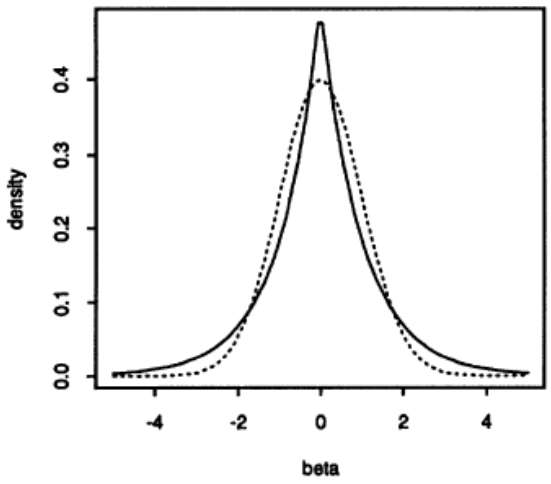

Figure 3.7: The Laplace (l_1-norm, bold line) and Normal (l_2-norm, dotted line) densities. Note that the Laplace density is sparsity promoting as it penalises solutions away from zero more than the Gaussian density. [116]

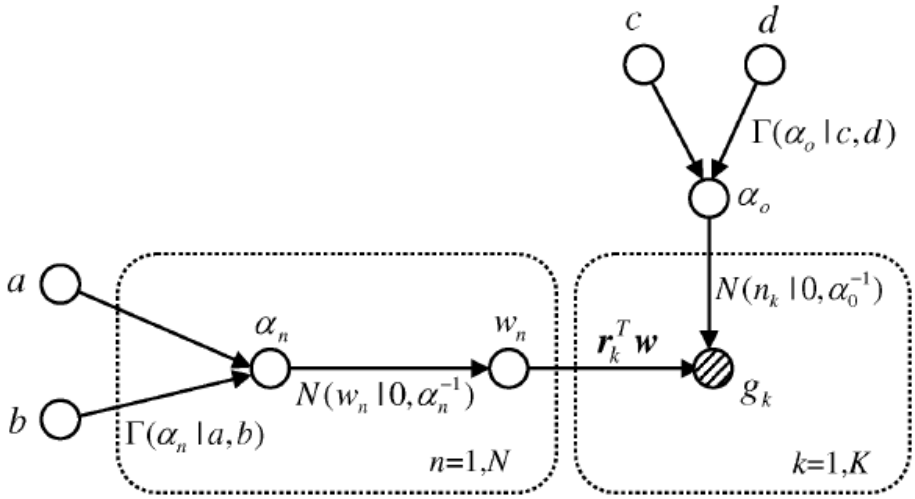

Figure 3.8: The hierarchical model for the Bayesian CS formulation [59]

is shown in figure 3.8. This model, and other CS models which not necessarily have closed form solutions, can be solved via belief-propagation [7], or via Monte-Carlo methods.

However, as with all methodologies, Bayesian algorithms have their drawbacks. Most notable is the use of the most computationally complex recovery algorithms. In particular MCMC methods suffer in high dimensional settings, such as those considered in compressive sensing. There has been an active line of work to address this: most notably Hamiltonian Monte Carlo (see [79]) - an MCMC sampling method designed to follow the typical set of the posterior density.

Belief propagation (BP) [126] is a popular iterative algorithm, offering improved reconstruction quality and undersampling performance. However, it is a computationally complex algorithm. It is also difficult to implement. Approximate message passing (figure 3.6) solves this issue by blending BP and (IT). The algorithm proceeds like iterative thresholding (figure 3.3), but computes an adjusted residual at each stage. The final term in the update:

$$z^{t+1} = y - Ax^t + \frac{||x||_0}{m} z^t \tag{3.34}$$

comes from a first order approximation to the messages passed by BP [73]. This is in contrast to the update from IST (figure 3.3):

$$z^{t+1} = y - Ax^t \tag{3.35}$$

The choice of prior is key in Bayesian inference, as it encodes all knowledge about the problem. Penalising the least-squares estimate with the ℓ_1 norm,

3.6 Compressive Estimation

In this section, we develop some intuition into constructing estimators for the signal s directly on the compressive measurements:

Theorem 3.6.1. *Given a set of measurements of the form:*

$$y = As + n \tag{3.36}$$

where $A \in \mathbb{R}^{m\times n}$, $A_{ij} \sim \mathcal{N}(0, 1/m)$, and $n \in \mathbb{R}^n$ is AWGN i.e. $\sim N\left(0, \sigma^2 I\right)$. We again assume that s comes from a fixed set of models, parametrised by some set Θ.

Then, the maximum likelihood estimator of s, for the case where s can be expanded in an orthonormal basis $s = \sum_{i=1}^{n} \alpha_i \phi_i$:

$$\hat{s} = \sum_{i=1}^{n} m\langle y, A\phi_i\rangle \phi_i \tag{3.37}$$

Proof. The likelihood for this model is, (as y is a normal random variable):

$$f\left(y \mid s\right) = \left(\frac{1}{(2\pi)^{n/2}}\right) \exp\left(-\frac{\left(y - As\right)^T \left(y - As\right)}{2}\right) \tag{3.38}$$

Taking the logarithm and expanding, we find

$$\ln f\left(y \mid s\right) = -y^T y - s^T A^T As + 2\langle y, As\rangle + c \tag{3.39}$$

which is equal to:

$$Inf = -\left|\left|y\right|\right|_2^2 - ||As||_2^2 + 2\langle y, As\rangle \tag{3.40}$$

(where the constant has been dropped). The first term of (3.40) is constant, for the same reasons as in section (3.6). The term

$$||As||_2^2 = \langle As, As\rangle \tag{3.41}$$

can be written as

$$\langle A^T As, s\rangle \tag{3.42}$$

We will assume that $A^T A$ concentrates in the sense of 3.12 and replace 3.42 with its expectation $\mathbb{E}\big(\langle A^T As, s\rangle\big)$

$$\begin{aligned}\mathbb{E}\big(\langle A^T As, s\rangle\big) &= \mathbb{E}\sum_{i=1}^{n}(A^T As)_i^T s_i \\ &= \sum_{i=1}^{n}\mathbb{E}(A^T As)_i s_i \\ &= \sum_{i=1}^{n}(\frac{1}{m}e_i s_i)_i^T s_i \\ &= \frac{1}{m}\langle s, s\rangle\end{aligned}$$

because

$$\mathbb{E}A^T A = \frac{1}{m}I \tag{3.43}$$

as it is a Wishart matrix (see section3.2).
So we can further approximate (3.40):

$$\ln f\left(y \mid s\right) = -\left|\left|y\right|\right|_2^2 - \frac{1}{m}||s||_2^2 + 2\langle y, As\rangle \tag{3.44}$$

The only, non-constant part of (3.44) is the third term and so we define the estimator:

$$\hat{s} = \arg\max_{\Theta}\langle y, As(\Theta)\rangle \tag{3.45}$$

□

Corollary 3.6.2. *Consider the case where $y = As$ (no noise). Then*

$$y^T A\phi_j = \sum_i \alpha_i \phi_i^T A^T A\phi_j$$

So

$$y^T A\phi_j = \sum_i \alpha_i \phi_i^T A^T A\phi_j \sim \frac{\alpha_i}{m}\delta_{ij}$$

giving

$$\widehat{\alpha_i} = m\left(y^T A\phi_j\right) \tag{3.46}$$

Remark 3.6.3. *The matrix $M = A^T A$ is the projection onto the row-space of A. It follows that $||Ms||_2^2$ is simply the norm of the component of s which lies in the row-space of A. This quantity is at most $||s||_2^2$, but can also be 0 if s lies in the null space of A. However, because A is random, we can expect that $||Ms||_2^2$ will concentrate around $\sqrt{m/n}\,||s||_2^2$ (this follows from the concentration property of sub-Gaussian random variables (3.12)).*

Example 3.6.4. *Example: Single Spike We illustrate these ideas with a simple example: estimate which of n frequencies s is composed of.*

A signal $s \in \mathbb{R}^{300}$ composed of a single (random) delta function, with coefficients drawn from a Normal distribution (with mean 100, and variance 1) i.e

$$s = \alpha_i \delta_i \tag{3.47}$$

with

$$a_i \sim \mathcal{N}(100, 1) \tag{3.48}$$

and the index i chosen uniformly at random from $[1, n]$.
The signal was measured via a random Gaussian matrix $A \in \mathbb{R}^{100\times 300}$, with variance $\sigma^2 = 1/100$ and the inner product between $y = As$ and all 300 delta functions projected onto $\mathbb{R}^{100}$ was calculated:

$$\hat{\alpha}_j = m\langle (A\alpha_i\delta_i), A\delta_j\rangle \tag{3.49}$$

We plot the $\hat{\alpha}_j$ below, figure 3.9, (red circles), with the original signal (in blue, continuous line). Note how the maximum of the $\hat{\alpha}_j$, coincides with the true signal.

3.6.1 Estimating Frequency Spectra

3.7 Compressive Sensing Architectures

3.7.1 Modulated Wideband Converter

We try to sense and reconstruct a wideband signal, divided into L channels. We have a (connected) network of J (= 50) nodes placed uniformly at random within the square $[0,1] \times [0,1]$. This is the same

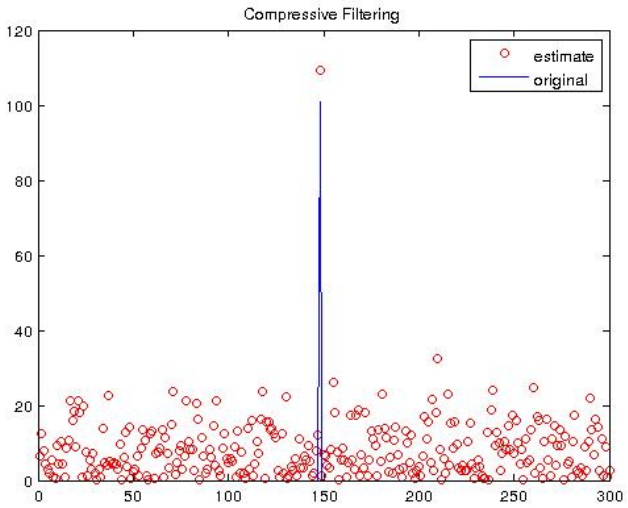

Figure 3.9:

model, as in [129]. The calculations which follow are taken from [129] as well.

The nodes individually take measurements (as in [75]) by mixing the incoming analogue signal $x(t)$ with a mixing function $p_i(t)$ aliasing the spectrum. $x(t)$ is assumed to be bandlimited and composed of up to k uncorrelated transmissions over the L possible narrowband channels - i.e. the signal is k-sparse.

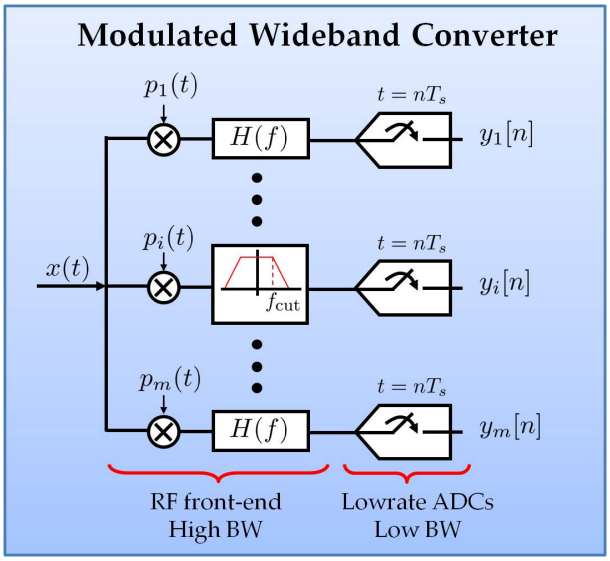

Figure 3.10: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

The mixing functions - which are independent for each node - are required to be periodic, with

period T_p. Since p_i is periodic it has Fourier expansion:

$$p_i(t) = \sum_{l=-\infty}^{\infty} c_{il} \exp\left(j l t \frac{2\pi}{T_p}\right) \tag{3.50}$$

The c_{il} are the Fourier coefficients of the expansion and are defined in the standard manner. The result of the mixing procedure in channel i is therefore the product xp_i, with Fourier transform (we denote the Fourier Transform of x by $X(.)$):

$$\begin{aligned} X_i(f) &= \int_{-\infty}^{\infty} x(t)\, p_i(t)\, dt \\ &= \sum_{l=-\infty}^{\infty} c_{il} X(f - l f_p) \end{aligned} \tag{3.51}$$

(We insert the Fourier series for p_i, then exchange the sum and integral). The output of this mixing process then, is a linear combination of shifted copies of $X(f)$, with at most $\lceil f_N Y Q / f_p \rceil$ terms since $X(f)$ is zero outside its support (we have assumed this Nyquist frequency exists, even though we never sample at that rate).

This process is repeated in parallel at each node so that each band in x appears in baseband.

Once the mixing process has been completed the signal in each channel is low-pass filtered and sampled at a rate $f_s \geq f_p$. In the frequency domain this is a ideal rectangle function, so the output of a single channel is:

$$Y_i\left(e^{j2\pi f T_s}\right) = \sum_{l=-L_0}^{+L_0} c_{il} X(f - l f_p) \tag{3.52}$$

since frequencies outside of $[-f_s/2, f_s/2]$ will filtered out. L_0 is the smallest integer number of non-zero contributions in $X(f)$ over $[-f_s/2, f_s/2]$ - at most $\lceil f_N Y Q / f_p \rceil$ if we choose $f_s = f_p$. These relations can be written in matrix form as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{3.53}$$

where $\mathbf{y}$ contains the output of the measurement process, and $\mathbf{A}$ is a product matrix of the mixing functions, their Fourier coefficients, a partial Fourier Matrix, and a matrix of channel coefficients. $\mathbf{x}$ is the vector of unknown samples of $x(t)$.

i.e. $\mathbf{A}$ can be written:

$$\mathbf{A} = \mathbf{SFDH} \tag{3.54}$$

The system 3.53 can then be solved (in the sense of finding the sparse vector $\mathbf{x}$ by convex optimisation via minimising the objective function:

$$\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \tag{3.55}$$

where λ is a parameter chosen to promote sparsity. Larger λ means sparser $\mathbf{x}$.

3.7.2 Random Demodulator

The random demodulator was introduced by Rice University in [61] and extended in [53], where limitations were placed on the switching rate of the random waveform. The architecture consists of three main steps: demodulation, filtering, and uniform sampling.

We assume that the analogue signal $x(t)$ is bandlimited, periodic, and comprised of a finite number of components from some arbitrary dictionary (for example) $\psi_n(t)$:

$$x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t) \tag{3.56}$$

Initially, the signal is modulated by a pseudo-random sequence $p_c(t)$, which alternates at frequencies at (or above) the Nyquist frequency of $x(t)$. As x is multitoned, the chipping sequence will spread the tones across the entire spectrum. The signal is then filtered - this filter acts as an accumulator and sums the demodulated signal, before being sampled at rate $\mathscr{M}$ with a traditional ADC.

The output of this system, $y[m]$, can be related to the input $x(t)$ via a linear transformation of the coefficient vector α_n.

To find the transformation A, first consider the output $y[m]$, which is the result of convolution and demodulation followed by sampling at rate $\mathscr{M}$:

$$y[m] = \int_{-\infty}^{\infty} x(\tau)\, p_c(\tau)\, h(t-\tau)\, d\tau \,|_{t=m\mathscr{M}} \tag{3.57}$$

and by expanding $x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t)$:

$$y[m] = \sum_{n=1}^{N} \alpha_n \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathscr{M}-\tau)\, d\tau \tag{3.58}$$

we see that the output can be written as:

$$y = Ax \tag{3.59}$$

with

$$A_{m,n} = \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathscr{M}-\tau)\, d\tau \tag{3.60}$$

This random demodulation process ensures that each tone in the signal is identifiable in the filter's passband, and because there are few tones present in the signal they are identifiable from the low rate samples.

An extension of the Random Demodulator is the Random Modulation Pre-Integrator (RMPI). This device was introduced in [127] and is an universal compressive sensing apparatus. It is a variant of the Random Demodulator, but one which uses multiple channels.

The RMPI works by modulating the input signal by a different pseudo-random sequence per channel, where the random sequences oscillate at the Nyquist rate. Then the output of each modulator is integrated over a time period T chosen to be as large as possible (to lower the sampling rate), whilst still allowing fast computation.

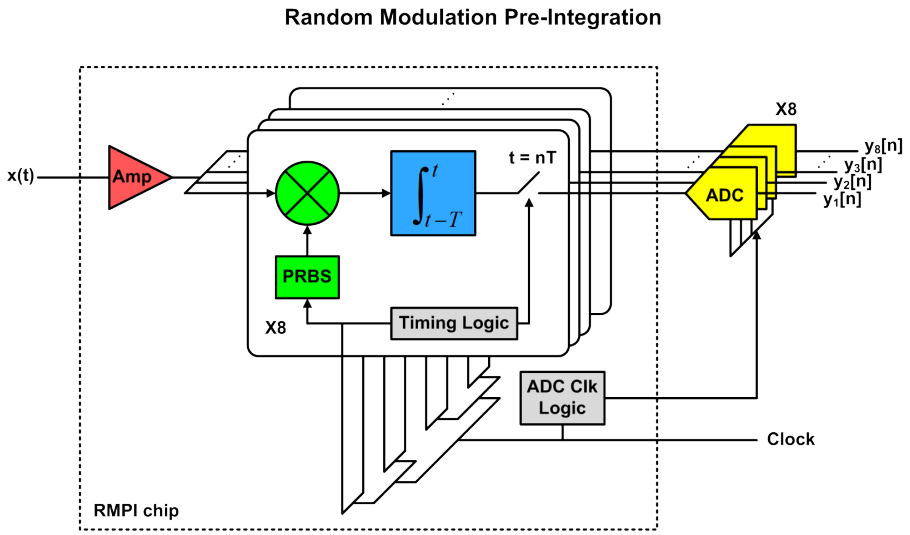

Figure 3.11: The hierarchical model for the Bayesian CS formulation [59]

The main innovation in the RMPI when compared to the Random Demodulator is that it is much easier to mix the signal at the Nyquist rate, than it is to sample at the Nyquist rate. This allows the RMPI to reduce the sampling rate by a factor of n, (where $n >> gg$, a much larger factor than the Random Demodulator.

There are several drawbacks to the Random Demodulator, which are not addressed by either the RMPI or the Modulated Wideband Converter. In particular, they are the need to generate a random chipping sequence, and high complexity from using multiple channels in the RMPI and MWC. The high rate chipping sequence used in the RD and RMPI results in high power consumption, and drains battery life of the device. This issue has been addressed in [72], which proposes using frequency division multiplexing to generate the random sequences. Parallel realisations of the Random Multiplexer further increase the power drain, complexity and cost by each parallel channel employing a high rate mixer, and a relatively low rate ADC.

There is at least one other disadvantage to the RD: circuit realisations of the RD cannot have a chipping sequence which switches at infinite polarity - it is difficult to practically generate a fast switching random sequence. To address this issue Harms et al in [53] modify the RD to use random waveforms with constraints on the switching rate - specifically, fully random waveforms are replaced with Run Length Limited sequences. Harms et al demonstrate that by choosing the signal statistics correctly - specifically the power spectrum of the chipping sequence should match the power spectrum of the tones to be sensed for the performance of the constrained random

demodulator will match that of the random demodulator. Whether this can be achieved in practice is an open question.

3.7.3 Compressive Multiplexer

Slivinsky et al in [107], introduced the compressive multiplexer. This architecture assumes the signal is divided into multiple discontiguous channels, whose occupancy is collectively sparse. The Compressive Multiplexer (Compressive Multiplexer (CMUX)) acquires J independent channels, each of bandwidth $W/2$ Hz, into a single stream running at the Nyquist rate of any single channel.

The process is achieved by initially mixing the channels down to baseband, and then performing multiplying with a chipping sequence. Finally the spread channels are summed and sampled once per chip by a single ADC. This summation occurs across channels and not in time, in contrast to the Random Demodulator (RD), and the Modulated Wideband Converter (MWC).

Mathematically the CMUX can be represented as Φ a $W \times JW$ matrix, formed by concatenating J $W \times W$ matrices horizontally. All elements along the diagonals are considered as Rademacher random variables. The sparsity basis of this model is the Fourier basis. In other words, the samples are acquired as:

$$y = Ax \tag{3.61}$$

where

$$A = \left[\Phi_1 \mathscr{F} \ldots \Phi_J \mathscr{F}\right] \tag{3.62}$$

and $\mathscr{F}$ is the $W \times W$ Fourier matrix.

3.8 Compressive Sampling for Spectrum Sensing

3.8.1 Noise Folding

In all measurement systems, there are two types of noise: measurement noise (noise caused by hardware imperfections) and signal noise (stochastic variations in the signal induced by the transmission environment). That is the received signal can be modelled as:

$$y = A(x + e) + w \tag{3.63}$$

where $y \in \mathbb{R}^m$ is the received signal, $A \in \mathbb{R}^{m \times n}$ is the measurement matrix, $x \in \mathbb{R}^n$ is the signal, $e \in \mathbb{R}^n$ is the signal noise, and $n \in \mathbb{R}^m$ is the measurement noise. This is a different model compared to that presented in the rest of this section: part of the noise is now scaled by the measurement matrix A. It is commonly assumed in the literature that $e = 0$, and that w is a vector drawn from a sub-Gaussian distribution. We assume that that w is non-zero, and is drawn from an n-dimensional

sub-Gaussian distribution. In [36], Davenport et al quantify the impact of this extra scaling. They begin by introducing the following measures of SNR:

Definition 3.8.1. *Input SNR*

$$\text{ISNR} = \frac{||x||_2^2}{\mathbb{E}||e\,|_\Gamma||_2^2} \tag{3.64}$$

where Γ *represents the support of* x*, and the expectation is taken with respect to the Lebesque measure.*

Definition 3.8.2 (Recovered SNR)**.**

$$\text{RSNR} = \frac{||x||_2^2}{\mathbb{E}||\hat{x}-x||_2^2} \tag{3.65}$$

They then show

Theorem 3.8.3 ([36])**.** *Suppose,* $y = A(x+e)$ *where* $e \in \mathbb{R}^n$ *is a zero-mean white random vector, and* x *is* s*-sparse, and* $A \in \mathbb{R}^{m\times n}$ *satisfies the RIP (see definition* (3.2.2)*) with constant* δ *and has orthogonal rows each with norm* $\sqrt{\rho}$*. Then the RSNR of an oracle-assisted recovery algorithm satisfies:*

$$\frac{\rho}{1+\delta} \leq \frac{\text{ISNR}}{\text{RSNR}} \leq \frac{\rho}{1-\delta} \tag{3.66}$$

If the ratio of RNSR and ISNR is measured in dB, we have:

$$\frac{\text{ISNR}}{\text{RSNR}} \sim 10\log_{10}\rho \tag{3.67}$$

In other words, for every doubling of the subsampling factor ρ, the SNR loss increases by 3dB. This degradation is an important tradeoff in the design of CS receivers. It implies that for a signal with fixed bandwidth $W/2$, there is a practical limit to the instantaneous bandwidth $n/2$ for which any desired RSNR can be obtained.

3.8.2 Dynamic Range

Because Compressive Sensing allows us to lower the sampling rate in order to sense a sparse signal, this allows the use of higher resolution ADCs. Thus a CS acquisition system, should exhibit a larger dynamic range than a conventional system. Firstly we define the signal-to-quantisation noise ratio:

Definition 3.8.4.

$$\text{SQNR} = \frac{||x||_2^2}{||x-\text{Q}_b(x)||_2^2} \tag{3.68}$$

where $Q_b(x)$ *is the* b*-bit quantised version of* x*.*

Then [36] shows that:

Theorem 3.8.5. *Suppose $y = Ax$, where x is s-sparse and A satisfies the RIP with constant δ. Let $\hat{x}$ denote the output of applying a recovery algorithm to the quantised measurements $Q_b(y)$, which satisfies:*

$$||\hat{x} - x||_2^2 \leq \kappa \left|\left|Q_b(y) - y\right|\right|_2^2 \tag{3.69}$$

then,

$$\text{RSNR} \geq \frac{\text{SQNR}}{(1+\delta)\kappa} \tag{3.70}$$

3.8.3 CS models

One of the first uses of Compressive Sensing for spectrum sensing was the work of Tian and Giannakis in [49]. In this work, the authors proposed reconstructing the (discrete) gradient of the frequency spectrum. In this work the authors propose the following model:

$$\hat{z} = \arg\min_z ||z||_1 \text{ s.t } (SG)\, z = x \tag{3.71}$$

In this model, $x \in \mathbb{R}^n$ is a discretisation of a continuous, piecewise smooth spectrum, $S \in \mathbb{R}^{m \times n}$ is a sampling operator, for example a partial Fourier matrix or a random set of Nyquist measurements, and $G \in \mathbb{R}^{n \times n}$ is a product matrix:

$$G = \left(F_n^{-1}\Phi\right)^{-1} \Gamma^{-1} \tag{3.72}$$

where F_n is the $n \times n$ Fourier matrix, Φ is the inverse Wavelet matrix (for example, Φ could be the Haar matrix), and Γ is the $n \times n$ discrete derivative matrix.

We then see, that z is the derivative of the Wavelet modulus, and that frequency bands can be estimated from the peaks of z. It is noted in [49] that as the scale factor of the Wavelets increases, the noise variance is reduced.

In [92], Polo et al improve on the methodology of [49]. The main innovation from this work is that to perform the compressive sensing directly, as opposed to Tian et al, where the signal was sampled at a Nyquist rate, before compression via a random sampling operator. Polo et al replace this with an Analogue-to-Information converter.

It has been assumed that the received signal vector in CS systems would be wide-sense stationary (WSS). However, in [109], Sundman et al disagreed with this assumption - they interpret the sensing matrix A in CS measurements:

$$y = Ax \tag{3.73}$$

where the elements of A are chosen randomly from a sub-Gaussian distribution, as a time-varying linear filter. Thus, the CS spectrum sensing system will not be a time-invariant, and the WSS property of the original signal will not necessarily be preserved by the measurement process.

The authors subvert this problem, by instead adopting a definition of the recovered signal based upon the autocorreleation matrix, instead of the autocorrelation sequence of the edge-spectrum.

More recently, a feature detection model has been proposed by Tian, Tafesse, Sadler [115]. The authors note that the cyclic spectrum of communication signals is sparse. The paper directly reconstructs the 2-D cyclic spectrum from the compressive samples, without reconstructing the signal as an intermediate step. This is possible because of the linear relationship between the cyclic covariance function and the cyclic spectrum (the quantities are related by a Fourier transform). Once the covariance vector of the data has been formed, the cyclic spectrum can be recovered using the LASSO. For details see [115].

These ideas have been significantly extended, and now arbitrary second order statistics can be computed from compressive samples, without the original signal being reconstructed as an intermediate step. The applications include, wideband spectrum sensing, incoherent imaging, and power spectrum estimation. The manuscript [94] has more details. In particular the framework is extended to array signal processing, and to source localistaion problems in radar.

In [55] Havary and Valaee propose a bank of filters, far less in number than the number of channels in a frequency spectrum, into which the wideband signal will be fed. The outputs of these filters are used to reconstruct the channel occupancies by comparing the results of a standard ℓ_1 minimisation algorithm, with a pre-defined vector of energy thresholds. Simulations shown in the paper suggest that a 20 channel signal can be accurately sensed with 12-15 filters.

Braun et al in [14] build upon this line of work, and the work of [35] and, by using a compressive matched filter (*smashed filter*) for compressive detection. The authors consider the problem of detecting narrow band pilot sequences in wideband signals, comparing a smashed filter with a matched filter. The authors conclude that in an AWGN scenario, compressive sensing can be used to significantly reduce sampling rates without affecting the reliability of the detector. What is significant about this work, is that neither the signal, nor any second order statistics are reconstructed at any stage. This is in contrast to most applications of compressive sensing where some quantity of interest is reconstructed as part of the signal processing chain. These ideas were significantly extended in [34], which extends the ideas of [35] to estimation, and classification scenarios without an expensive reconstruction algorithm as an intermediate step.

In [63], Lexa et al build upon these ideas, proposing a sub-Nyquist method for Power Spectral Density estimation for WSS signals, based upon multi-coset sampling. The authors show that the system computes an approximation to the true PSD, and that expensive ℓ_1 minimisation techniques need not be resorted to regardless of the sparsity of the underlying signal.

There have also been many distributed models of wideband spectrum sensing introduced in the literature. The first detailed approach to distributed CS was by [43], where the authors considered a network of nodes cooperating amongst each other to sense signals which are jointly sparse among the nodes.

The ideas of [43] are extended in [68], where the authors consider ergodic signals instead of

sparse signals. They establish that these signals can be sensed effectively in the compressive sensing framework.

In [131], the distributed Compressive Sensing framework is applied to wideband spectrum sensing via a sensor network. The authors propose a probabilistic graphical model to capture the occupancy common to all nodes, as well as spectrum occupancy viewed at a single node. That is, the authors consider the situation of a heterogeneous network where nodes are affected by different primary users, the set of which is not necessarily the same for all sensors. Belief Propagation is used to perform the statistical inference, and recover the spectrum occupancy at each node from the cooperative CS samples. Simulations indicate that this method performs wells in comparison with other methods.

Zhang et al in [129] propose a simpler model, where a network of sensors captures a wideband signal with no innovations at any node. Their signal model is based upon the Modulated Wideband Converter [74], but in this scenario, each node independently has a single channel of the converter. Reconstruction is done at a nominated fusion node, which has access to all the measurements taken by the network (via communication). Simulations reported by the authors suggest that the method is promising.

These late two papers, [131], and [129] both transmit the compressive measurements to a Central node, which does the requisite processing. Either via Belief Propagation as in [131], or via the Alternating Direction Method of Multipliers [129]. This is a common pattern in distributed sensing, as it allows the system to gain statistical strength via spatial diversity whilst also allowing fast computation at a single node. However, this method also introduces a single point of failure: should the fusion node fail in some way (lose power, find itself in a deep fade in the radio environment etc), then the system cannot recover the original signal. This motivates the use of distributed solvers to go along with the distributed capture of frequency spectra.

An example of this kind of work is [110], where the distributed CS model is extended to several sparsity models, which capture differing signal intercorrelations amongst nodes. We are primarily interested in the common signal model: that is all nodes within the network observe the same sparse signal, but under differing noise and fading. The paper also develops a distributed greedy pursuit algorithm for inverting the compressive samples, based upon orthogonal matching pursuit. The authors demonstrate via extensive simulations that the distributed algorithm matches the performance of a centralised greedy algorithm.

In [99], distributed estimation of signals is analysed from a statistical point of view. Schizas et al establish that there exist algorithms with guaranteed convergence to the required estimator (for example Maximum Likelihood) in the presence of fading and noise, for sensor networks with ideal links. In particular the authors introduce a decentralised scheme for least-squares and best linear unbiased estimator (BLUE).

The same authors build on this work in [98], and extend their analysis to decentralised MAP estimators. Again they show that when consensus is achieved between sensors, the estimate has

converged to the ideal MAP estimate. In both papers, the authors express the desired estimator (either least-squares or MAP) as a convex optimisation problem over the nodes of the network, and then show that the Alternating Direction Method of Multipliers will suffice as an algorithm to compute the estimate. The framework doesn't require that the estimator be expressed in closed form and the estimation procedure is resilient to communication noise.

Finally, there has been some work on compressive detection applied to spectrum sensing. Notably in [120], and [12] the work of [33] is applied to cognitive radio. In the first paper, the authors

CHAPTER 4

ADMM

4.1 Introduction

This chapter surveys the Alternating Direction Method of Multipliers (ADMM). This is an efficient convex algorithm for inference in high dimensional problems and can be applied to inverting the compressive samples, producing accurate estimates of the original signal.

We begin (in section 4.2) by introducing the algorithm, the type of problem for which it is an appropriate tool, and the algorithm's history. In particular we make connections with MAP estimation in high dimensional statistics. We discuss two worked examples: finding the minimal norm solution of a set of linear equations, and the Least Absolute Shrinkage and Selection Operator. The first example is to give a simple introduction to ADMM via a well known result, whilst the second is particularly relevant to compressive sensing.

In section 4.3, we move on from the examples, to a more advanced discussion where we formalise the proximity operator. The Prox operator is a generalisation of the Euclidean projection, and can be evaluated for a variety of common optimisation objectives. The evaluation is either directly in closed form, or via a simple to evaluate component-wise algorithm. The proximity operator moves the objective in the direction of the minimiser, and thus can be thought of a generalisation of gradient descent. We show some examples of how to compute the proximity operator, for functions which appear in statistical estimation. We also show its connection to ADMM - the algorithm proceeds by iteratively computing the Prox operators (see definition (4.3.1)) of the two functions comprising the objective. This is why it is such an useful algorithm: if the relevant proximity operators are easy to evaluate, then ADMM is an inexpensive and relatively fast algorithm.

Finally, we discuss acceleration techniques for ADMM, in section 4.4. In particular, we discuss schemes by Nesterov for non-smooth functions. We show that generalised gradient descent can be thought of as constructing an approximating polynomial to the objective term by term. By considering

Chebyshev polynomials, we can improve the speed of approximation accuracy and that this is the reason that Nesterov acceleration works.

4.2 Background

We begin this section with some definiitions, pinning down the kind of functions we will be optimising over (closed, proper, convex functions with Lipschitz continuous gradients). We then discuss the Lagrangian formulation of constrained optimisation, discussing some precursers to ADMM. Finally the algorithm is introduced and explained.

4.2.1 Preliminary Definitiona

Definition 4.2.1 (Convexity)**.** *We say that a function f is **convex** is it satisfies the following inequality:*

$$f\left(\lambda x_1 + (1-\lambda)\, x_2\right) \leq \lambda f\left(x_1\right) + (1-\lambda)\, f\left(x_2\right) \tag{4.1}$$

This can be rephrased as:

$$f\left(\mathbb{E}x\right) \leq \mathbb{E}f\left(x\right) \tag{4.2}$$

An intuitive physical metaphor for convexity can be found in MacKay [69]: if a collection of masses x_i are placed on a convex curve f, then the centre of gravity of those masses, which lies at $\left(\mathbb{E}x, \mathbb{E}f\left(x\right)\right)$ lies above the curve.

An equivalent definition of convextiy is that the Hessian (matrix of second derivatives of f) is positive definite (has only positive eigenvalues).

A slightly stronger condition is that of strong convexity:

Definition 4.2.2 (Strong convexity)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be l-strongly convex if for some $l > 0$ it satisfies:*

$$f\left(y\right) \geq f\left(x\right) + \nabla f\left(x\right)^T \left(y - x\right) + \frac{l}{2}\left|\left|x - y\right|\right|^2 \tag{4.3}$$

Definition 4.2.3 (Closed function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be **closed** if $\forall\, \alpha$ the sublevel set*

$$\{x \in \mathrm{dom} f \mid f\left(x\right) \leq \alpha\} \tag{4.4}$$

is closed

Definition 4.2.4 (Proper function)**.** *A function f is said to be **proper**, if f has the property that $f\left(x\right) < \infty$ for at least one x and $f\left(x\right) > -\infty$ for all x.*

Definition 4.2.5 (Closed, Proper convex function)**.** *A function is said to be a **closed, proper convex function** is it is convex, proper and closed.*

Example 4.2.6. *The function:*

$$f(x) = cx^2 \tag{4.5}$$

on $[-1,1]$ *is a closed, proper, convex function.*

In this thesis the type of functions we are minimising, are closed, proper convex with Lipschitz continuous gradients.

Definition 4.2.7. *A function is said to have* ***Lipschitz continuous gradients*** *if its derivative satisfies:*

$$\left|\left|\nabla f(x) - \nabla f(y)\right|\right| \leq L\left|\left|x - y\right|\right| \tag{4.6}$$

We will make use of the following definitions in passing in the following text, so it is useful to have them here:

Definition 4.2.8 (Convex conjugate)**.** *The function* $f^* : \mathbb{R}^n \to \mathbb{R}$ *is the* ***convex conjugate*** *of the function* $f : \mathbb{R}^n \to \mathbb{R}$ *where:*

$$f^*(y) = \sup_{x \in \mathrm{dom} f} \left(y^T x - f(x)\right) \tag{4.7}$$

and finally we define the condition number of f:

Definition 4.2.9 (Condition number)**.** *The* ***condition number*** *of a closed, proper, and convex function* f *is defined to be:*

$$\kappa = \frac{L}{l} \tag{4.8}$$

where L *is the Lipchitz constant, and* l *is the strong convexity constant.*

4.2.2 Lagrangian Formulation

Definition 4.2.10 (Lagrangian)**.** *In constrained optimisation, where a smooth function is minimised subject to constraints:*

$$\min_x f(x) \tag{4.9}$$

$$s.t\, g(x) = b \tag{4.10}$$

we define the Lagrangian to be the function:

$$L(x,\eta) = f(x) + \eta^T\left(b - g(x)\right) \tag{4.11}$$

The variable x *is refereed to as the primal variable.*

Definition 4.2.11 (Lagrange Dual)**.** *The Lagrange dual function is defined to be:*

$$H(\eta) = \inf_x L(x, \eta) \tag{4.12}$$

Definition 4.2.12 (Lagrange Multiplier)**.** *The Lagrange multiplier η is defined as:*

$$\frac{\partial L}{\partial b} = \eta \tag{4.13}$$

That is the Lagrange multiplier is the rate of change of the quantity being optimised as a function of the constraint parameter.

Example 4.2.13 (One dimensional contrained optimisation [13])**.** *For the minimisation problem*

$$\min_x f(x) \tag{4.14}$$
$$s.t\ Ax = b \tag{4.15}$$

The Lagrangian is:

$$L(x, \eta) = f(x) + \eta^T (Ax - b) \tag{4.16}$$

and the dual function is:

$$H(\eta) = -f^*\left(-A^T\eta\right) = b^T y \tag{4.17}$$

where f^* is the convex conjugate of f.

Example 4.2.14 (Dual Ascent)**.** *For the minimisation problem*

$$\min_x f(x) \tag{4.18}$$
$$s.t\ Ax = b \tag{4.19}$$

The Lagrangian is:

$$L(x, \eta) = f(x) + \eta^T (Ax - b) \tag{4.20}$$

and the dual function is:

$$H(\eta) = -f^*\left(-A^T\eta\right) = b^T y \tag{4.21}$$

where f^* is the convex conjugate of f.
The dual problem is:

$$\max_\eta H(\eta) \tag{4.22}$$

We can find an optimum pair of points by gradient descent. First we find a point $x^+ = \arg\min_x L$*, and then we have* $\nabla g = Ax^+ - b$*, which is the residual for the equality constraint. The dual ascent method is:*

$$x^{k+1} := \arg\min_x L \tag{4.23}$$

$$\eta^{k+1} := \eta^k + \alpha\left(Ax^{k+1} - b\right) \tag{4.24}$$

where $\alpha \in$ *is a step size.*

The η *variable can be interpreted as a vector of prices, that an agent must pay in order to satisfy the optimisation problem. Thus the second step in dual ascent (and the final step in ADMM) can be interpreted as a price update.*

4.2.3 ADMM

ADMM is a convex optimisation algorithm which splits large, and typically non-smooth, problems into smaller pieces which are easier to handle. At heart, the algorithm involves finding a zero of a convex objective function, by evaluating proximal operators [88], see definition 4.3.1.

ADMM has a long history, first being investigated by Rockarfellar [93], in the context of minimisation in Hilbert spaces - finite and infinite dimensional vector spaces equipped with an inner product and a metric induced by the inner product, typical examples include $\mathbb{R}^n$, the space of n-dimensional Euclidean vectors, and $\mathbb{L}^2(\mathbb{R})$, the space of square-integrable functions over the reals. Douglas and Rachford [41] also investigated ADMM as a method for solving the heat equation. Subsequently the theory was generalised and extended by Eckstein and Bersekas [44].

ADMM has found applications in Sparse Coding [16], Compressive Radar Imaging [56], Medical Imaging [97], and Optimal Control [83]. This is because many optimisation objective functions can be formulated as a sum of simpler convex functions, and casting the problem in an alternating minimisation framework is straightforward for practitioners.

ADMM also has explicit convergence rates, (see [106], and [81]), which both show (through different methods), that in every step of the algorithm, the iterates moves towards the optimum by a factor of $1 - 1/k$, where k is iteration number.

ADMM as lo has well understood methodologies for tuning and acceleration, [47] shows how to choose optimal step-sizes from and [51] extends the algorithm to an accelerated version and gives quantitative guarantees on the convergence properties.

However, care must be taken when the problem is posed as the sum of more than two convex functions - the direct extension is not necessarily convergent: the paper of Chen[26], shows that ADMM applied to the sum of three functions will not necessarily converge.

ADMM is concerned with minimising a the sum of two functions, subject to a constraint:

$$\begin{aligned} &\arg\min_{x} f(x) + g(z) \\ &\text{s.t } Ux + Vz = c \end{aligned} \tag{4.25}$$

where f and g are assumed to be convex functions with range in $\mathbb{R}$, $U \in \mathbb{R}^{p\times n}$ and $V \in \mathbb{R}^{p\times m}$ are matrices (not assumed to have full rank), and $c \in \mathbb{R}^p$.

ADMM consists of iteratively minimising the augmented Lagrangian

$$L_p(x,z,\eta) = f(x) + g(z) + \eta^T (Ux + Vz - c) + \frac{\rho}{2}\|Ux + Vz - c\|_2^2$$

where, x, z are the primal variables, η is a Lagrange multiplier (or dual variable), and ρ is a parameter we can choose to make $g(z)$ smooth [80].

To perform the minimisation of the Lagrangian we use the following iterations:

$$x^{k+1} := \arg\min_{x} L_\rho\left(x, z^k, \eta^k\right) \tag{4.26}$$

$$z^{k+1} := \arg\min_{z} L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{4.27}$$

$$\eta^{k+1} := \eta^k + \rho\left(Ux^{k+1} + Vz^{k+1} - c\right) \tag{4.28}$$

The first two steps are primal variable updates, and the second step is a dual variable update. We illustrate this second step again with a one dimensional example:

The alternating minimisation works because of the decomposability of the objective function: the x minimisation step is independent of the z minimisation step and vice versa.

Example 4.2.15 (Minimum norm solution of a Linear Equation)**.** *Consider the following problem, of finding the minimum norm solution of a linear system:*

$$\min_{x} ||x||_2 \tag{4.29}$$

$$s.t\ Ax = y \tag{4.30}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$ and we implicitly assume that $y = Ax$ has a solution and that y is in the range of A.

This problem has the well known optimal solution:

$$x^\star = A^T\left(AA^T\right)^{-1} y \tag{4.31}$$

This problem can be formulated as

$$\min_x ||x||_2 + \mathbb{I}(y) \tag{4.32}$$
$$s.t\ Ax - y = 0 \tag{4.33}$$

The Lagrangian for this example, is then:

$$L_p(x, y, \eta) = ||x||_2 + \mathbb{I}(y) + \eta^T (Ax - y) + \frac{\rho}{2} \left|\left|Ax - y\right|\right|_2^2$$

and the ADMM iterations are:

$$x^{k+1} := \left(1 - \frac{\rho}{\left|\left|x^k\right|\right|_2}\right) x^k \tag{4.34}$$
$$z^{k+1} := \Pi\left(x^{k+1}\right) \tag{4.35}$$
$$\eta^{k+1} := \eta^k + \rho\left(Ax^{k+1} + y^{k+1}\right) \tag{4.36}$$

We now illustrate the theory with a more involved example, the LASSO.

Example 4.2.16 (LASSO)**.** *Given a set of measurements of the form*

$$y = Ax + n \tag{4.37}$$

where $x \in \mathbb{R}^n$ is an s-sparse vector we wish to recover, $y \in \mathbb{R}^m$ is a set of noisy measurements, $A \in \mathbb{R}^{m \times n}$ is a design or measurement matrix such that x is not in the null-space of A, and $z \in \mathbb{R}^m$ is AGWN. The signal x can be recovered by minimising the objective function:

$$L = \frac{1}{2} \left|\left|Ax - y\right|\right|_2^2 + \lambda ||x||_1 \tag{4.38}$$

where λ is a parameter which trades off the reconstruction accuracy and sparsity of x: larger λ means sparser x.

We also consider the problem:

$$L = \frac{1}{2} \left|\left|Ax - y\right|\right|_2^2 + \lambda ||x||_0 \tag{4.39}$$

where $||t||_0 = \{|t| \mid t_i \neq 0\}$.

*The ADMM iterations for LASSO, which can be found by alternately differentiating (**??**) with respect to x,z and η, are (in closed form):*

$$x^{k+1} := \left(A^T A + \rho I\right)^{-1} \left(A^T y + \rho\left(z^k - \eta^k/\rho\right)\right) \tag{4.40}$$
$$z^{k+1} := S_{\lambda/\rho}\left(x^{k+1} + \eta^k/\rho\right) \tag{4.41}$$
$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{4.42}$$

where $S_{\lambda/\rho}(\circ)$ *is the soft thresholding operator:* $S_\gamma(x)_i = \text{sign}(x_i)\left(|x_i| - \gamma\right)^+$.
The individual steps in (4.42) *can be found differentiating* (**??**) *with respect to* x *and* z *as follows:*

$$\frac{\partial L}{\partial x} = -A^T\left(y - Ax\right) + \rho(x - z) + \eta$$

as

$$\frac{\partial}{\partial x}||F(x)||_2^2 = 2\left(\frac{\partial}{\partial x}F(x)\right)F(x) \tag{4.43}$$

by the chain rule, and $\partial/\partial x(Ax) = -A^T$ *(see the Matrix Cookbook, [90]) as differentiation exchanges a linear operator with its adjoint.*

Setting (4.43) *to zero and collecting like terms:*

$$\left(A^T A + \rho I\right)x = A^T y + \rho z - \eta \tag{4.44}$$

so we find the optimal x *is:*

$$x = \left(A^T A + \rho I\right)^{-1}\left(A^T y + \rho\left(z - \eta/\rho\right)\right) \tag{4.45}$$

note that this estimator is a weighted average of the ordinary least squares estimate $(A^T y)$ *and a Gaussian prior. This is to be expected, as the minimisation problem w.r.t* x *is an* l_2*-regularised MAP problem.*

for $z > 0$

$$\frac{\partial L}{\partial z} = \lambda + \rho(x - z) - \eta \tag{4.46}$$

from which we obtain:

$$z = x + \frac{1}{\rho}(\eta - \lambda)$$

since $z > 0$ *then* $x + \frac{1}{\rho}(\eta - \lambda I) > 0$ *when* $x + \frac{\eta}{\rho} > \frac{\lambda}{\rho}$.
Similarly for $z < 0$*:*

$$\frac{\partial L}{\partial z} = -\lambda + \rho(x - z) \tag{4.47}$$

setting (4.47) *to zero we obtain:*

$$z = x + \frac{1}{\rho}(\eta + \lambda)$$

since $z < 0$ *then* $x + \frac{1}{\rho}(\eta + \lambda) < 0$ *when* $x + \frac{\eta}{\rho} < -\frac{\lambda}{\rho}$.
at $z = 0$ *we find:*

$$-\frac{\lambda}{\rho} \leq x + \frac{\eta}{\rho} \leq \frac{\lambda}{\rho}$$

i.e.

$$|x + \frac{\eta}{\rho}| \leq \frac{\lambda}{\rho} \tag{4.48}$$

combining (4.47), (4.46), (4.48) *together we find the optimal z is:*

$$z = \text{sign}(x + \frac{\eta}{\rho}) \max\left(|x + \frac{\eta}{\rho}| - \frac{\lambda}{\rho}, 0\right) \tag{4.49}$$

Together (4.45), (4.49) *and the third step of* (4.42) *constitute the steps of the ADMM algorithm.*

This algorithm has a nice statistical interpretation: it iteratively performs ridge regression, followed by shrinkage towards zero. This is the MAP estimate for x under a Laplace prior.

The soft-thresholding operator can be derived by considering the MAP estimate of the following model:

Example 4.2.17.

$$y = Ax + w \tag{4.50}$$

where x is some (sparse) signal, and w is additive white Gaussian noise. We seek

$$\hat{x} = \arg\max_x \mathbb{P}_{x|y}(x|y) \tag{4.51}$$

This can be recast in the following form by using Bayes rule, noting that the denominator is independent of x and taking logarithms:

$$\hat{x} = \arg\max_x \left[\log \mathbb{P}_w(y - x) + \log \mathbb{P}(x)\right] \tag{4.52}$$

The term $\mathbb{P}_n(y - x)$ arises because we are considering $x + w$ with w zero mean Gaussian, with variance σ_n^2. So, the conditional distribution of y (given x) will be a Gaussian centred at x.

We will take $\mathbb{P}(x)$ to be a Laplacian distribution:

$$\mathbb{P}(x) = \frac{1}{\sqrt{2}\sigma} \exp -\frac{\sqrt{2}}{\sigma}|x| \tag{4.53}$$

Note that $f(x) = \log \mathbb{P}_x(x) - \frac{\sqrt{2}}{\sigma}|x|$, and so by differentiating $f'(x) = -\frac{\sqrt{2}}{\sigma}\text{sign}(x)$

Taking the maximum of 4.52 we obtain:

$$\frac{y - \hat{x}}{\sigma_n^2} - \frac{\sqrt{2}}{\sigma} sign(x) = 0 \tag{4.54}$$

Which leads the soft thresholding operation defined earlier, with $\gamma = \frac{\sqrt{2}\sigma_n^2}{\sigma}$ as (via rearrangement):

$$y = \hat{x} + \frac{\sqrt{2}\sigma_n^2}{\sigma}\text{sign}(x)$$

or

$$\hat{x}(y) = \text{sign}(y)\left(y - \frac{\sqrt{2}\sigma_n^2}{\sigma}\right)_+$$

i.e $S_\gamma(y)$.

4.3 The Proximity Operator

In this section we introduce the proximity operator of a closed and proper convex function. We discuss the existence of the operator, and some properties. We then motivate the operator from the perspective of ADMM, before giving some examples.

The Proximity Operator for a closed, convex, and proper function f (the set of all such functions will be denoted Γ) in a Hilbert space $\mathscr{H}$ is defined as [77]:

Definition 4.3.1 (Proximity Operator)**.**

$$\text{Prox}_f(y) := \underset{y\in\mathscr{H}}{\arg\min}\, f(y) + \frac{1}{2}||y-x||^2 \tag{4.55}$$

Intuitively the Proximity Operator approximates a point x by another point y, that is close in the mean-square sense under the penalty f.

The Prox$(\circ)$ operator exists for closed and convex f as $(y) + \frac{1}{2}||y-x||^2$ is closed with compact level sets and is unique as $(y) + \frac{1}{2}||y-x||^2$ is strictly convex.

The corresponding Moreau envelope is defined as

Definition 4.3.2 (Moreau Envelope)**.**

$$\text{M}_f(y) := \min_{y\in\mathscr{H}} f(y) + \frac{1}{2}||y-x||^2 \tag{4.56}$$

The Moreau envelope is a strict generalisation of the squared distance function. M_f is real valued - even when f takes the value ∞, whilst Prox_f is $\mathscr{H}$-valued.

Properties

Theorem 4.3.3 (Moreau '65)**.** *Let $f \in \Gamma$ and f^* be its Fenchel conjugate. Then the following are equivalent:*

- $z = x + y, y \in \partial f(x)$
- $x = \text{Prox}_f(z), y = \text{Prox}_{f^*}(z)$

Theorem 4.3.4 ([77])**.** *Let $f \in \Gamma$. Then for all $z \in \mathscr{H}$*

- $\text{Prox}_f(z) + \text{Prox}_{f^*}(z) = z$
- $\text{M}_f(z) + \text{M}_{f^*}(z) = \frac{1}{2}||z||^2$

Theorem 4.3.5 ([77]). *The Moreau envelope is (Frechet) differentiable, with*

$$\nabla \text{M}_f = Id - \text{Prox}_f = \text{Prox}_{f^*} \tag{4.57}$$

Theorem 4.3.6 ([77]). $\text{Prox}_f : (\mathscr{H}, ||\circ||) \leftarrow (\mathscr{H}, ||\circ||)$ *is 1-Lipchitz continuous.*

Examples

Example 4.3.7 (Indicator). *From the definition*

$$\text{Prox}_I(x) := \arg\min_y I_C(y) + \frac{1}{2}\left|\left|y-x\right|\right|^2 \tag{4.58}$$

$$= \arg\min_{y \in C} \frac{1}{2}\left|\left|y-x\right|\right|^2 \tag{4.59}$$

$$= P_C(x) \tag{4.60}$$

where $I_C(y)$ is the indicator of some set C and P_C is the projection operator onto that set.

Example 4.3.8 (l_2 norm). *For $f(y) = \frac{\mu}{2}\left|\left|y\right|\right|^2$ the* Prox *operator is:*

$$\text{Prox}_f(x) := \arg\min_y \frac{\mu}{2}\left|\left|y\right|\right|^2 + \frac{1}{2}\left|\left|y-x\right|\right|^2 \tag{4.61}$$

$$= \frac{1}{1+\mu}x \tag{4.62}$$

Example 4.3.9 (l_1 norm). $f = \|x\|_1$

$$\text{Prox}_f(x) := \text{sign}(x_i)\left(|x_i| - \gamma\right)^+ = S_\gamma(x)_i \tag{4.63}$$

Example 4.3.10 (Elastic Net). *Consider*

$$f(x) = \lambda\|x\|_1 + \mu||x|| \tag{4.64}$$

$$\text{Prox}_f(x) := \frac{\lambda}{1+\mu}S_\gamma(x)_i \tag{4.65}$$

Example 4.3.11 (Fused Lasso)**.** *Consider*

$$f(x) = \|x\|_1 + \sum_{i=1}^{d-1} (x_i - x_{i-1}) \tag{4.66}$$

i.e the sum of the l_1 and TV norms

$$\text{Prox}_f(x) := \text{Prox}_{l_1} \circ \text{Prox}_{TV} = S_\gamma (\text{Prox}_{TV})_i \tag{4.67}$$

Example 4.3.12 (Consensus)**.** *Suppose we want to solve a problem such as:*

$$\underset{x}{\text{minimise}} \quad \sum_i f_i(x)$$

this could arise in statistical computing where f_i would be the loss function for the i^th block of training data. We can write the problem for distributed optimisation as:

$$\begin{aligned} \underset{x}{\text{minimise}} \quad & \sum_i f_i(x_i) \\ \text{subject to} \quad & x_i - z = 0 \end{aligned}$$

where x_i are local variables (for example local to each node in a spectrum sensing) and $x_i - z = 0$ are the consensus constraints. Consensus and regularisation can be achieved by adding a regularisation term $g(z)$ - for example $g(z) = \lambda||x||_1$ corresponds to the LASSO, and the f_i would be $f_i = ||A_i x_i - b||_2^2$.

As per the previous sections, we form the Augmented Lagrangian:

$$L_\rho(x, y) = \sum_i^n \left(f_i(x_i) + y_i^T (x_i - z) + \frac{\rho}{2} \|x_i - z\|_2^2 \right) \tag{4.68}$$

The ADMM iterations for this Lagrangian are:

$$x_i^{k+1} := \arg\min x_i \left(f_i(x_i) + y_i^{kT} (x_i - z) + \frac{\rho}{2} \|x_i - z\|_2^2 \right) \tag{4.69}$$

$$z^{k+1} := \frac{1}{n} \sum_i^n \left(x_i^{k+1} + (1\rho) y_i^k \right) \tag{4.70}$$

$$y_i^{k+1} := y_i^k + \rho \left(x_i^{k+1} - z^{k+1} \right) \tag{4.71}$$

The z^{k+1} iteration is analytic as we're minimising the squared norm of $x_i - z$ - so we average. With $\|x\|_1$ regularisation we perform soft-thresholding after the z update.

At each iteration the sum of the dual variables y_i is zero, so the algorithm can be simplified to:

$$x_i^{k+1} := \arg\min x_i \left(f_i(x_i) + y_i^{kT} \left(x_i - \bar{x}^k \right) + \frac{\rho}{2} \|x_i - \bar{x}^k\|_2^2 \right) \tag{4.72}$$

$$y_i^{k+1} := y_i^k + \rho \left(x_i^{k+1} - z^{k+1} \right) \tag{4.73}$$

where

$$\bar{x}^k = \frac{1}{n}\sum_i^n x_i^k \tag{4.74}$$

This algorithm can be summarised as follows: in each iteration

- *gather x^k and average to get $\bar{x}^k$*
- *scatter the average to nodes*
- *update y_i^k locally*
- *update x_i locally*

Each agent is minimising its own function, plus a quadratic term (the squared norm) which penalises the agent from moving too far from the previous average.

Note that the 'gather' stage doesn't require a central processor - this can be done in a distributed manner also.

Motivation

We are solving problems of the following form:

$$\min_{x\in\mathscr{H}} f(x) + g(z) \tag{4.75}$$

$$\text{s.t } x - z = 0 \tag{4.76}$$

with $f, g \in \Gamma$. To solve this problem we form the augmented Lagrangian:

$$L_p(x,z,\eta) = f(x) + g(z) + \eta^T(Ux + Vz - c) + \frac{\rho}{2}\|Ux + Vz - c\|_2^2$$

and then performing the following iterative minimisation:

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{4.77}$$

$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{4.78}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{4.79}$$

i.e.

$$x^{k+1} := \underset{x}{\arg\min}\left(f(x) + \eta^{kT}x + \frac{\rho}{2}\left|\left|x - z^k\right|\right|^2\right) \tag{4.80}$$

$$z^{k+1} := \underset{z}{\arg\min}\left(g(z) - \eta^{kT}z + \frac{\rho}{2}\left|\left|x^{k+1} - z\right|\right|^2\right) \tag{4.81}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{4.82}$$

pulling the linear terms into the quadratic ones we get:

$$x^{k+1} := \underset{x}{\arg\min}\left(f(x) + \frac{\rho}{2}\left|\left|x - z^k + (1/\rho)\eta^k\right|\right|^2\right) \tag{4.83}$$

$$z^{k+1} := \underset{z}{\arg\min}\left(g(z) + \frac{\rho}{2}\left|\left|x^{k+1} - z - (1/\rho)\eta^k\right|\right|^2\right) \tag{4.84}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{4.85}$$

i.e.

$$x^{k+1} := \text{Prox}_f\left(z^k - u^k\right) \tag{4.86}$$

$$z^{k+1} := \text{Prox}_g\left(x^{k+1} + u^k\right) \tag{4.87}$$

$$u^{k+1} := u^k + \left(x^{k+1} - z^{k+1}\right) \tag{4.88}$$

with $u^k = (1/\rho)\eta^k$.

The motivation for the Proximal operator should now be clear: to perform the minimisation we simply calculate the proximal operator of each of the functions at each step. For many functions found in Statistics (e.g. the l_p norms, this can be found in closed form, and so ADMM presents a particularly attractive method for finding MAP solutions to regularised statistical problems.

4.4 Acceleration

In the previous sections we have considered un-accelerated ADMM – with this method it is possible to achieve a $1/k$ convergence rate to an optimal solution (should such a solution exist). This means that after k iterations:

$$\frac{\left|\left|x^k - x^\star\right|\right|}{||x^\star||} \leq \mathcal{O}\left(\frac{1}{k}\right) \tag{4.89}$$

In this section, we describe acceleration methods (due to Nesterov, and introduced to signal processing by Beck and Teboule) which allow a modified ADMM to achieve a $1/k^2$ convergence, at a modest increase in algorithmic complexity.

The key insight, due to Nesterov and applied to smooth optimisation problems, was that gradient descent may move in the direction of steepest descent at each step - this direction might not be in

the same direction as the previous step. Thus, by including a 'momentum' term, the iteration path can be forced in the direction of the previous step and the algorithm will converge much faster.

For acceleration, the ADMM algorithm can be modified into the following form:

$$x^{k+1} := \text{Prox}_f\left(z^k - u^k\right) \tag{4.90}$$

$$z^{k+1/2} := \text{Prox}_f\left(x^{k+1} + u^k\right) \tag{4.91}$$

$$u^{k+1/2} := u^k + \left(x^{k+1} + z^{k+1}\right) \tag{4.92}$$

$$a^{k+1} = \frac{1+\sqrt{1+4(a^k)^2}}{2} \tag{4.93}$$

$$z^{k+1} = z^k + \frac{a^k - 1}{a^{k+1}}\left(z^{k+1/2} - z^k\right) \tag{4.94}$$

$$u^{k+1} = u^k + \frac{a^k - 1}{a^{k+1}}\left(u^{k+1/2} - u^k\right) \tag{4.95}$$

$$\tag{4.96}$$

The terms $\left(z^{k+1/2} - z^k\right)$ and $\left(u^{k+1/2} - u^k\right)$, represent the momentum of the iterations. Conceptually, in this modification, we are not just taking a step in the direction of the gradient - we are taking a step in the same direction as the previous step.

In [51] it was shown that for strongly convex objectives (objective functions which are the sum of two functions, which are both strongly convex), this algorithm converges to optimally at a rate $\mathcal{O}\left(\frac{1}{k^2}\right)$.

There are several interpretations as to why this method accelerates gradient descent type algorithms. This method works, for any strongly convex and twice differentiable function, as gradient descent is computing a degree k approximation to the inverse function.

$$\frac{1}{x} = \sum_{k=0}^{\infty} (1-x)^k \tag{4.97}$$

By optimising over the creation of this polynomial using Chebyshev polynomials, we can improve upon the convergence rate of naive gradient descent (or generalised gradient descent).

Another view of Nesterov acceleration was given by (Su, Candes and Boyd in [108]) . In that paper, they show how Nesterov acceleration is a discretisation of a second order ODE. This work explains much of the qualitative behaviour of the accelerations scheme. In particular the damped oscillation phenomena (a process of progressively under and over-shooting the optimum) . This behaviour is perhaps why the acceleration method is relatively less robust in practice when compared to vanilla gradient descent: terminating after a fixed number of iterations could produce an output which wildly overshoots the optimum.

Modelling the scheme as a differential equation also gives insight to which functions would benefit from acceleration – in particular functions which are well conditioned (the ratio of their convexity constant to their Lipchitz constant is small). This is because the behaviour modelled in the

ODE is different in different friction regimes: in particular poorly conditioned functions (low friction) will be subject to more oscillations and thus take longer to converge to equilibrium.

CHAPTER 5

OPTIMISATION ON GRAPHS

5.1 Introduction

This chapter introduces a multi-block generalisation of ADMM, suitable for inference over sensor networks. We propose a decentralised solver for recovering a signal from a set of compressive measurements, independently made at each node. We improve over the model of [129] by not requiring the use of a Fusion Centre to do the reconstruction - this is replaced by softthresholded gradient descent and local communications between neighbouring nodes to recover the signal.

We also improve on the work of [78], by providing explicit, closed form, iterates for a series of functionals commonly used in Compressive Sensing - as opposed to relying on a slow intermediate optimisation algorithm to find intermediate points. To derive our algorithm, we give new proofs of ideas found in [78]

The structure of the chapter is as follows: we give an overview of the (vector) sensing model at each node, and how the global sensing model decomposes into a collection of individual sensing problems at each node.

We then give a derivation of the new algorithm for composite smooth and non-smooth objective functions. In the derivation we form a global objective, and show how it can be decomposed into a collection of separate objectives. We give new proofs of ideas in [78] - specifically we show how the objective decomposes over edges between nodes, and thus the global objective can be written as a sum of local objective functions at each node. Thus each node can execute a series of simple operations, built up from matrix vector multiplies and component wise thresholding, along with a single step of local communication at each iteration.

We demonstrate the general algorithm on two widely used objective functions in compressive sensing - the LASSO (both with ℓ_1 and ℓ_0 regularisation) (see (3.20)) and the Dantzig Selector (see

(3.28)). We demonstrate recovery on synthetic random signals.

We then extend the vector algorithm to matrix sensing problems, and demonstrate recovery in the multiple measurement vector (MMV) case. We demonstrate signal recovery with a distributed Modulated Wideband converter model. We show that the MMV-LASSO recovery achieves adequate performance for spectrum sensing, both in terms of reconstruction accuracy and in terms of signal undersampling.

5.2 Constrained Optimisation on Graphs

We model the network of sensors as an undirected graph $G = (V, E)$, where $V = \{1, \ldots, J\}$ is the set of vertices, and $E = V \times V$ is the set of edges. An edge between nodes i and j implies that the two sensors can communicate. The set of nodes that node i can communicate with is written $\mathcal{N}_i$ and the degree of node i is $D_i = |\mathcal{N}_i|$.

We are considering a distributed variation of the following model:

$$y = Ax + n \tag{5.1}$$

Where, $y, n \in \mathbb{R}^p$, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{p \times n}$. This is a standard scenario. Individually nodes make the following measurements:

$$\mathbf{y}_r = \mathbf{A}_r \mathbf{x} + \mathbf{n}_r \tag{5.2}$$

where $\mathbf{A}_r$ is the r^{th} row of the sensing matrix from (5.2), and (5.2) is formed by concatenating the individual nodes' measurements together.

We assume that a proper colouring of the graph is available: that is, each node is assigned a number from a set $C = \{1 \ldots c\}$, and no node shares a colour with any neighbour. This is so that nodes may communicate in colour order, as opposed to communicating individually thus reducing the total number of communication rounds required.

To find the $\mathbf{x}$ we are seeking (the solution to the linear system (5.2)), to each node we give a copy of $\mathbf{x}, \mathbf{x}_p$ and we constrain the copies to be identical across all edges in the network. Each node, thus has a separate optimisation to solve, subject to the constraint that it is consistent with its neighbours.

The problem then is to solve:

$$\begin{aligned} \arg\min_{\tilde{x}} \sum_{c=1}^{C} \sum_{j \in C_c} f(x_j) + \frac{\lambda}{J} g(x_j) \\ \text{s.t } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \ldots, C\} \end{aligned} \tag{5.3}$$

We can write the global optimisation variable as $\bar{x}$, which collects together C copies of a $n \times 1$ vector $\mathbf{x}$:

Definition 5.2.1. *We define vectors x_c, where $c = 1, \ldots, C$ and write the vector of length nJ:*

$$\bar{x} = \sum_{c=1}^{C} w_c \otimes x_c = \left[x_{c(1)}^T, \ldots, x_{c(J)}^T\right]^T \tag{5.4}$$

where $w_{c(i)} = \mathbb{I}(c(i) = c)$, $\mathbb{I}$ is the indicator function, and we have written $c(i)$ for the colour of the ith node.

These constraints can be written more compactly by introducing the node-arc incidence matrix B: a V by E matrix where each column is associated with an edge $(i, j) \in E$ and has 1 and -1 in the ith and jth entry respectively. Figures (5.1) and (5.2) show examples of a network and it's associated incidence matrix.

The constraint $x_i = x_j$ if $\{i, j\} \in E$ can now be written

$$\sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \tag{5.5}$$

note that $\left(B^T \otimes I_n\right) \in \mathbb{R}^{nE \times nJ}$. Together (5.4) and (5.5), suggests that the problem (5.3) can be re-written as:

$$\begin{aligned} \underset{\bar{x}}{\operatorname{arg\,min}} \sum_{c=1}^{C} \sum_{j \in C_c} f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) \\ \text{s.t.} \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \\ \text{and } \bar{x}_c - \bar{z}_c = 0 \end{aligned} \tag{5.6}$$

where $\beta = \frac{\lambda}{J}$.

The global Augmented Lagrangian [13] for the problem (5.6) can be written down as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} \Big(\sum_{j \in c} f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) + \\ + \theta^T \left(\bar{x}_j - \bar{z}_j\right) + \frac{\rho}{2} \left|\left|\bar{x}_j - \bar{z}_j\right|\right|_2^2 \Big) + \\ + \eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c + \frac{\rho}{2} \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|_2^2 \end{aligned} \tag{5.7}$$

This is, superficially, similar to the Augmented Lagrangian for the Lasso problem [13][Section 6.4]. That is, the terms indexed by j are a straightforward Lasso problem, constrained by edge-wise variables (indexed by c) forcing consistency across the network. However, the problem (as currently written) is not separable across the edges of the network (in the sense that the Lagrangian (5.7)

is composed only of terms with the subscript j) as the final and penultimate term represent the constraint that the nodes agree on their estimates across edges.

To make it possible that 5.7 can be posed as a constrained optimisation problem at each node, we introduce the following variable (so that the the final term of 5.7 is separable across edges of the graph):

Definition 5.2.2.

$$\begin{aligned} u &:= \left(B^T \otimes I_n\right) \bar{x} \\ &= \left(B^T \otimes I_n\right) \sum_{c=1}^{C} w_c \otimes x_c \\ &= \sum_{c=1}^{C} B_c^T \otimes x_c \end{aligned}$$

where we have used the definition (5.4) *in the second line, and the property of Kronecker products* $(A \otimes C)(B \otimes D) = (AB \otimes CD)$ *between the second and third lines, and we write* $B_c = w_c^T B$.

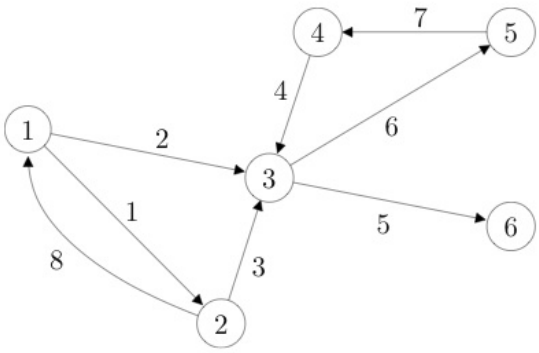

Figure 5.1: An example of a network

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

Figure 5.2: The incidence matrix associated with Figure (5.1)

The terms $\| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \|^2$ and $\eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c$ of (5.7), can be decomposed across edges, using the following lemma:

Lemma 5.2.3 (Edge Decomposition)**.**

$$\left|\left|\sum_{c=1}^{C}\left(B_c^T\otimes I_n\right)\bar{x}_c\right|\right|^2=\sum_{j\in C_1}\left(D_j\left|\left|x_j\right|\right|_2^2-\sum_{k\in N_j}x_j^Tx_k\right) \tag{5.8}$$

and

$$\eta^T\sum_{c=1}^{C}\left(B_c^T\otimes I_n\right)\bar{x}_1=\sum_{l\in C_c}\sum_{m\in N_l}sign\left(m-l\right)\eta_{ml}^Tx_l \tag{5.9}$$

where η is decomposed edge-wise: $\eta=\left(\ldots,\eta_{ij},\ldots\right)$, such that $\eta_{i,j}=\eta_{j,i}$, and is associated with the constraint $x_i=x_j$.

Proof.

$$\begin{aligned}u^Tu&=\sum_{c_1=1}^{C}\sum_{c_2=1}^{C}\left(B_{c_1}\otimes x_{c_1}^T\right)\left(B_{c_2}^T\otimes x_c\right)\\&=\sum_{c_1,c_2}B_{c_1}B_{c_2}^T\otimes x_{c_1}^Tx_{c_2}\end{aligned}$$

BB^T is a $J\times J$ matrix, with the degree of the nodes on the main diagonal and -1 in position $\left(i,j\right)$ if nodes i and j are neighbours (i.e BB^T is the graph Laplacian). Hence, since we can write $B_{c_1}B_{c_2}^T=w_{c_1}^TBB^Tw_{c_2}$, the trace of $B_{c_1}B_{c_1}^T$ is simply the sum of the degrees of nodes with colour 1.

For $c_1\neq c_2$, $B_{c_1}B_{c_2}^T$ corresponds to an off diagonal block of the graph Laplacian, and so counts how many neighbours each node with colour 1 has.

Finally, note that $\eta\in\mathbb{R}^{nE}$ and can be written:

$$\eta=\sum_{c=1}^{C}w_c\otimes\eta_c \tag{5.10}$$

where η_c is the vector of Lagrange multipliers associated across edges from colour c. Now

$$\eta^Tu=\sum_{c_1=1}^{C}\sum_{c_2=1}^{C}w_{c_1}Bw_{c_2}\otimes\eta_{c_1}^Tx_c$$

by the properties of Kronecker products, and the definition of B_c. For $c_1=c_2$, η^Tu is zero, as there are no edges between nodes of the same colour b definition. For $c_1\neq c_2$, η^Tu counts the edges from c_1 to c_2, with the consideration that the edges from c_2 to c_1 are counted with opposite parity. □

Thus, we are able to write (5.7) as:

$$\begin{aligned}L_\rho=\sum_{c=1}^{C}\sum_{j\in C_c}&\left(f\left(x_j\right)+\beta g\left(z_j\right)\right)+\nu^Tx_j\\&\theta\left(x_j-z_j\right)+\frac{\rho}{2}D_i\left|\left|x_j\right|\right|^2+\frac{\rho}{2}\left\|x_j-z_j\right\|^2\end{aligned} \tag{5.11}$$

where we have defined:

$$\nu_i = \left(\sum_{k \epsilon \mathcal{N}_i} sign(k-i)\, \eta_{\{i,k\}} - \rho x_k \right) \tag{5.12}$$

this is a re-scaled version of the Lagrange multiplier, η.

Then by differentiating (5.11) with respect to x_j and z_j we can find closed forms for the updates as:

Theorem 5.2.4.

$$x_j^{k+1} := \text{Prox}_f \left(z_j^k - \theta_j^k \right) \tag{5.13}$$

$$z_j^{k+1} := \text{Prox}_g \left(x_j^{k+1} + \theta_j^k \right) \tag{5.14}$$

$$\theta_j^{k+1} := \theta_j^k + \rho \left(x^{k+1} - z^{k+1} \right) \tag{5.15}$$

$$\eta_j^{k+1} := \eta_j^k + \rho \left(\sum_{m \in N_j} z_m^k - z_j^k \right) \tag{5.16}$$

This algorithm can be thought of as follows: each node performs an iteration of (non multi-block) ADMM - i.e. each node solves an approximate Gaussian least-squares problem and then soft-thresholds - and then exchanges the result of this computation with its one-hop neighbours. This explains the inclusion of an extra Lagrange multiplier: the multiplier θ controls how far each node moves from its previous estimate in each iteration, whilst the multiplier η enforces consistency between nodes. Note that there is no communication of data between the nodes - only the result the computation in each round.

5.2.1 DADMM-Lasso

In this section we derive a version of algorithm 5.16 applied to the specific problem of sparse estimation with the LASSO (repeated here for convenience):

$$L = \frac{1}{2} \left|\left| Ax - y \right|\right|_2^2 + \lambda \left|\left| x \right|\right|_1 \tag{5.17}$$

The Proximity operators of $f = \frac{1}{2} \left|\left| Ax - y \right|\right|_2^2$ and $\lambda \left|\left| x \right|\right|_1$, are

$$\text{Prox}_f (x) = \left(A^T A + \rho I \right)^{-1} \left(A^T y - x \right) \tag{5.18}$$

and

$$\text{Prox}_g (x) = S_\gamma (x)_i \tag{5.19}$$

respectively. We can write the Lagrangian for this problem as:

$$L_\rho = \sum_{c=1}^{C} \sum_{j \in C_c} \frac{1}{2} \left|\left|A_j x_j - y_j\right|\right|_2^2 + \beta \left|\left|x_j\right|\right|_1 + \nu^T x_j + \tag{5.20}$$

$$\theta\left(x_j - z_j\right) + \frac{\rho}{2} D_i \left|\left|x_j\right|\right|^2 + \frac{\rho}{2} \|x_j - z_j\|^2 \tag{5.21}$$

and by differentiating (5.21) We finally arrive at the following algorithm:

Corollary 5.2.5.

$$x_j^{k+1} := \left(A_j^T A_j + (\rho D_J + 1) I\right)^{-1} \left(A_j^T y_j + z^k - \theta^{kT} - \nu^{kT}\right) \tag{5.22}$$

$$z_j^{k+1} := S_{\beta/\rho}\left(x_j^{k+1}\right) \tag{5.23}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{5.24}$$

$$\eta_j^{k+1} := \eta_j^k + \rho\left(\sum_{m \in N_j} z_m^k - z_j^k\right) \tag{5.25}$$

5.2.2 DADMM-MMV

In this section we extend our results to the multiple measurement vector (Multiple Measurement Vector (MMV)) model of [31]. The model can be summarised as follows:

$$Y = AX + N \tag{5.26}$$

Here, $Y, N \in \mathbb{R}^{m \times p}$, $x \in \mathbb{R}^{p \times n}$, $A \in \mathbb{R}^{m \times n}$

The MMV model is an extension of (5.2) where instead of a single sparse vector being estimated, a set of multiple (jointly) sparse vectors are recovered (under the assumption that these vectors share a common set of non-zeros).

The resulting optimisation problem is:

$$\arg\min_X ||X||_{2,1} \text{ s.t } Y = AX \tag{5.27}$$

which can be recast in unconstrained form as:

$$\arg\min_X \frac{1}{2} ||AX - Y||_F^2 + \lambda ||X||_{2,1} \tag{5.28}$$

This can be turned into a distributed optimisation problem, and so solved with (5.16). The augmented Lagrangian for this system is:

$$L_\rho = \sum_{c=1}^{C} \sum_{j \in C_c} \frac{1}{2} \left|\left|A_j X_j - Y_j\right|\right|_F^2 + \beta \left|\left|X_j\right|\right|_{2,1} + \nu^T x_j + \tag{5.29}$$

$$\theta\left(X_j - Z_j\right) + \frac{\rho}{2} D_i \left|\left|X_j\right|\right|^2 + \frac{\rho}{2} \|X_j - Z_j\|^2 \tag{5.30}$$

and by differentiating we arrive at the following algorithm:

$$X_j^{k+1} := \left(A_j^T A_j + (\rho D_J + 1)I\right)^{-1} \left(A_j^T Y_j + \rho Z^k - \Theta - N^{kT}\right) \tag{5.31}$$

$$Z_j^{k+1} := S_{\beta/\rho}\left(X_j^{k+1}\right) \tag{5.32}$$

$$\Theta_j^{k+1} := \Theta_j^k + \rho\left(X^{k+1} - Z^{k+1}\right) \tag{5.33}$$

$$H_j^{k+1} := H_j^k + \rho\left(\sum_{m\epsilon N_j} Z_m^k - Z_j^k\right) \tag{5.34}$$

5.3 Results

We tested our algorithms on random, simulated, data from two classes: to measure reconstruction quality we used random Gaussian signals, and to create ROC curves we used random uniform signals (fence-post signals). The former was generated with MATLAB's $mathrmsprandn$ function, whilst the latter was generated by selecting uniformly distributed vector components and setting the value of those chosen to unity - all other components are left at zero.

We compared the mean squared error (MSE) of our algorithms with a centralised solver (spgl1). We, investigated the performance of the algorithm with various settings of the regression parameter λ, and at various signal-to-noise ratios.

The MSE was calculated as follows:

$$\frac{\left|\left|Z^k - Z^*\right|\right|}{||Z^*||} \tag{5.35}$$

where Z^k is the result of the algorithm at iteration k, and Z^* is the optimal solution.

We first tested the algorithm's ability to reconstruct a sparse signal, to what extent this reconstruction is accurate and useful, and how it would compare with a centralised algorithm with access to all the information. We vary the regression parameter λ whilst keeping the Signal to Noise ratio (SNR) constant. Noise was added to this the system, to more realistically simulate scenarios found 'in the wild'.

Figure 5.3 shows the progress of the algorithm iterations for different values of the regression parameter. It (figure 5.3) is complemented by figure 5.4 which shows the progress of the algorithm iterations and a centralised solver (spgl11) which has all of the information. Collectively these show that for any given value of λ, the algorithm converges to a stable solution. However, some fine tuning of λ may be needed to find an acceptable solution - one in which a node can find a solution at least as good as a central processor with all the information. These figures also display the previously noted [**?**] behaviour that ADMM converges to a good solution in a few iterations, but takes considerably longer to converge to an excellent one.

We then contrasted ℓ_1 and ℓ_0 regularisation in the distributed algorithm. The results are presented in figure 5.5 which shows the algorithm with different regularisation - ℓ_1 vs ℓ_0. Figure 5.6 is another

similar experiment, but with an extended number of iterations. There is some tension between these two figures, as figure 5.5 shows that ℓ_0 regularisation is considerably worse than ℓ_1 penalisation, whilst figure 5.6 shows the exact opposite. Qualitatively speaking, we found that ℓ_1 regularisation provided more reliable results - in the sense of consistently and predictably finding a good solution. This can be explained by noting that ℓ_0 regularised LASSO is a non-convex objective, and whilst it's Prox operator can be computed the algorithm can get stuck in sub-minimal local optima. This is not a problem for ℓ_1 regularised LASSO, as for any given λ there is a single minima, which gradient descent algorithms are guaranteed to converge to (however slowly).

Finally, in the discussion of vector valued problems, we compared our algorithm to that presented in [65], where a linear approximation to the objective is made to facilitate speedy computation. For clarity the steps of the algorithm are:

Theorem 1.

$$x_j^{k+1} := x^k - \frac{1}{d_i}\left(\nabla f(x) + \rho\left(\sum_{j\in\mathcal{N}_i} x_i - x_j\right) + \theta^k\right) \tag{5.36}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(\sum_{j\in\mathcal{N}_i} x_i - x_j\right) \tag{5.37}$$

Figures 5.7 and 5.8 show the results of our experiments: they clearly show our methods are superior - Both in the sense of converging to a better solution, but also doing this in far fewer iterations (1000 vs 10000 or more).

We also simulated the MWC system from section 3.7.1, and we qualitatively assessed the results of the MMV algorithm 5.2.2. The MWC system was simulated, and a the MMV algorithm presented in section 5.2.2 was used to reconstruct the original signal. As can be seen in figure 5.10, the reconstruction is qualitatively similar to the original (pictured in figure 5.9). However, the reconstruction fails qualitatively to be better than simply calculating $A^T y$ - plotted in figure 5.11. This could be for a variety of reasons:

- An incorrect regression parameter $\lambda_m mv$ was used. Wheras for vector inverse problems - problems where a vector $x \in \mathbb{R}^n$ is inferred from measurements $y \in \mathbb{R}^m$ - where the choice of λ is bounded between 0 and $\left|\left|A^T y\right|\right|_\infty$, with a popular choice being $\sqrt{2\sigma \log n}$, finding an appropriate λ when x is reshaped into $X \in \mathbb{R}^{k\times p}$ is unclear.
- It is inappropriate to try to perform the inversion of a vector valued problem as a matrix valued problem.

We think that the problem is a mixture of the two.

5.4 Conclusions

We have demonstrated an alternating direction algorithm for distributed optimisation with closed forms for the computation at each step, and discussed the statistical properties of the estimation.

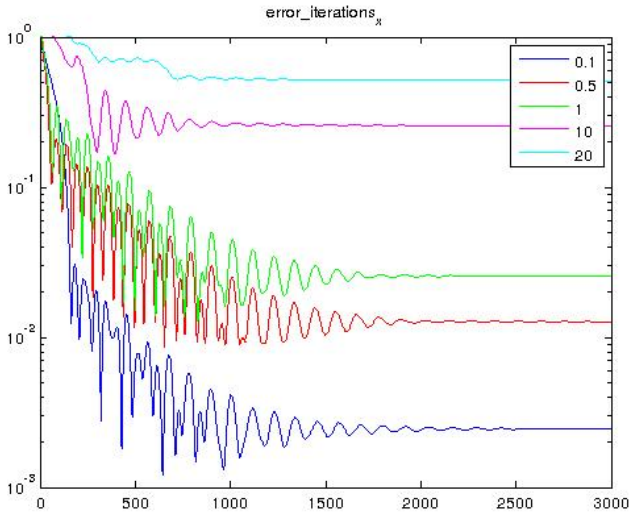

Figure 5.3: The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ

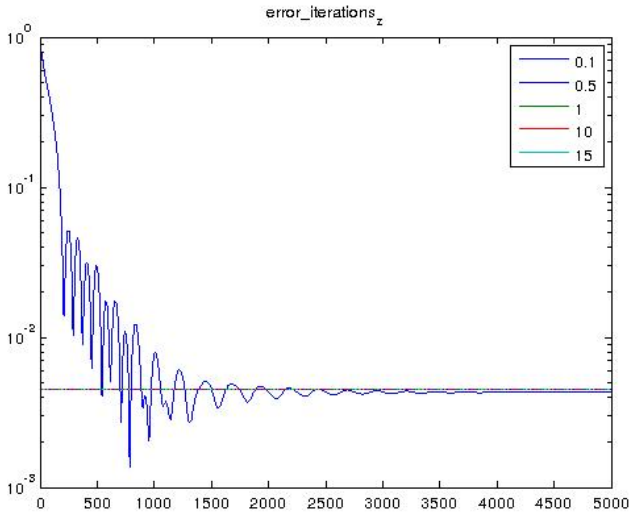

Figure 5.4: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

We have simulated the performance of this distributed algorithm for the distributed estimation of frequency spectra, in the presence of additive (white, Gaussian) and multiplicative (frequency flat) noise. We have shown that the algorithm is robust to a variety of SNRs and converges to the same solution as an equivalent centralised algorithm (in relative mean-squared-error).

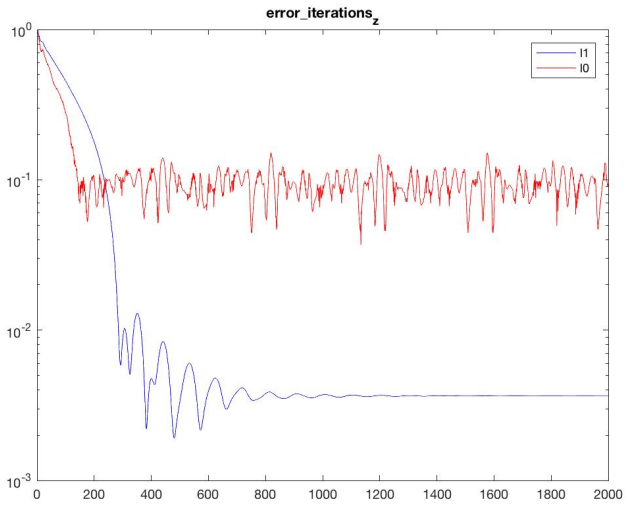

Figure 5.5: The progress of the distributed solver with ℓ_1 (blue) and ell_0 (red) regularisation as a function of the number of iterations.

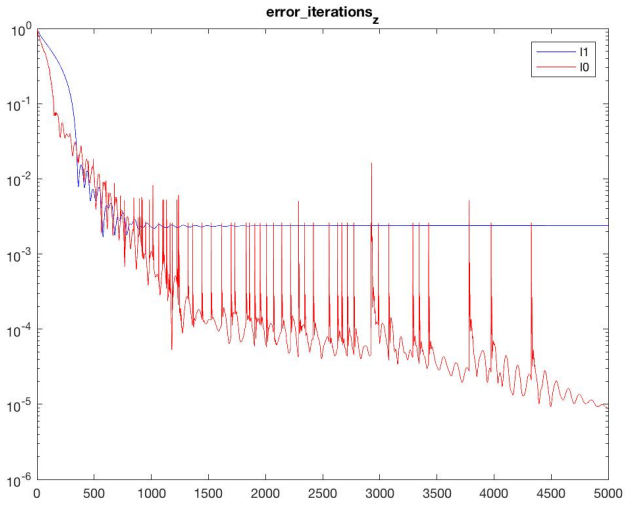

Figure 5.6: The progress of the distributed solver with ℓ_1 (blue) and ell_0 (red) regularisation as a function of the number of iterations. $\lambda_{\ell_1} = P\sqrt{2\log n}$, $\lambda_{\ell_0} = 1000\lambda_{\ell_1}$

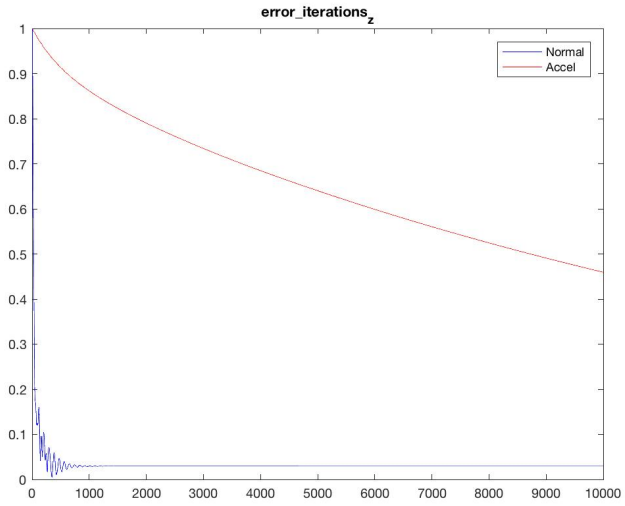

Figure 5.7:

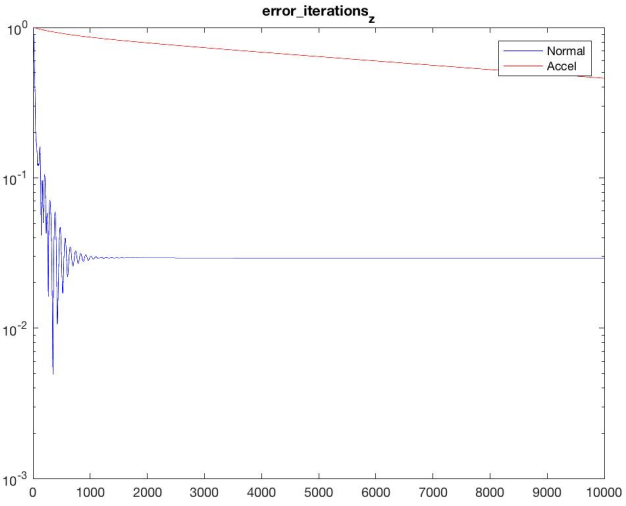

Figure 5.8:

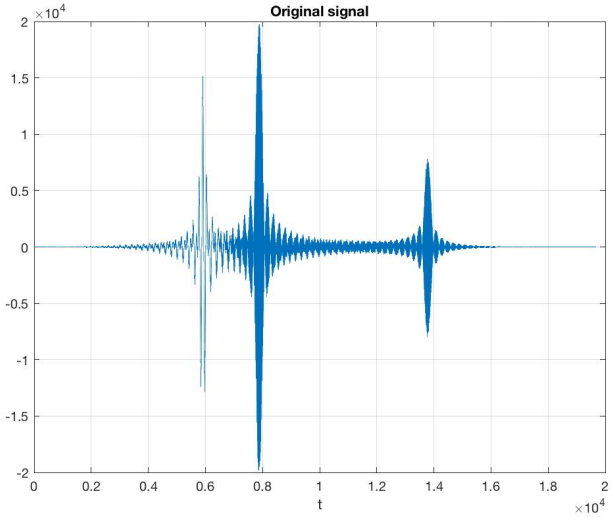

Figure 5.9:

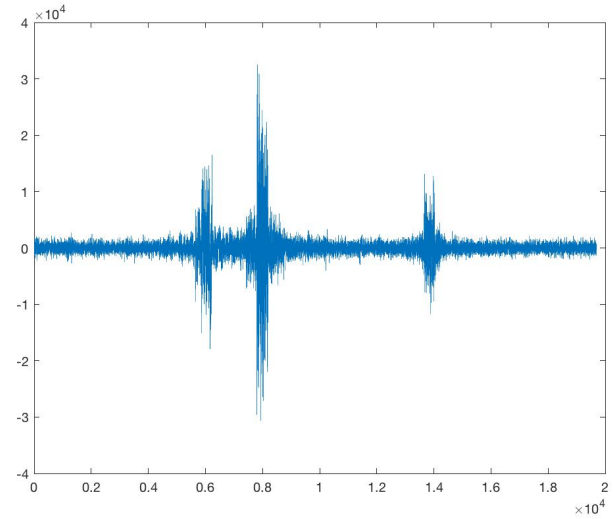

Figure 5.10:

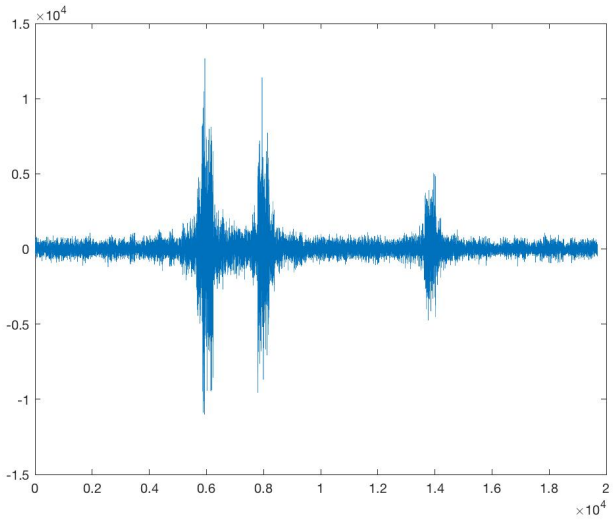

Figure 5.11:

CHAPTER 6

SENSING WITH HEAVISIDE BASIS

6.1 Introduction

In this chapter we develop a new model of frequency spectra, based upon the gradient of the frequency spectrum. We develop methods of band classification directly from compressive measurements, without reconstructing the signal as an intermediate step. This allows are system to operate at sampling rates below those predicted by Compressive Sensing theory - as the previous literature. We also show how by using this model, estimating frequency spectra is amenable to distributed reconstruction. We perform experiments on synthetic and real data (provided by OFCOM), which demonstrate our methods. We compare a distributed reconstruction and classification method with a centralised classification only method, and discuss their relative merits.

Compressive sensing makes the assumption that the signal being sensed is sparse. However we cannot always guarantee that the frequency spectrum will always be sparse: for example, should TVWS become widely utilised, the spectra will not be sparse. However, even for highly occupied spectra, the gradient of the spectrum will be sparse. This is because that TVWS spectra will feature bands with a flat power spectrum when occupied - or occupied bands will be well approximated by rectangular bands. Thus the PSD will feature abrupt changes in its gradient during transitions between occupied and unoccupied bands. This has previously been exploited by [114].

Reconstructing the spectrum from compressive measurements could take place at a fusion centre, but such communications are expensive. It is more efficient therefore to design distributed algorithms where CRs communicate with their neighbours to reach consensus on the reconstruction, given each nodes' private data. However, regularising the reconstruction process would require global co-ordination if Total Variation (TV) (the l_1 norm of the gradient of the signal) regularisation was chosen. This is because the information about the gradient will be spread out amongst nodes in the

network - these nodes may not necessarily be neighbours and so would not be able to share that information.

In this chapter we propose a different model for sensing the gradient of the frequency spectrum to [114] - a model which doesn't require Total Variation regularisation of the objective function.

The structure of the chapter is as follows: in section 6.2 we introduce the signal model and introduce a natural estimator of the signal using compressed measurements only. We demonstrate the results of this method on synthetic signals in **??**, in section 6.4 we show some results of the reconstruction quality of this model using the algorithm developed in chapter 5.4. In section **??**, we apply both of these methods to a larger dataset provided by OFCOM.

6.2 Signal Model

Not all signals are sparse in an orthogonal basis: for example, many images are sparse in an over-complete dictionary (set of bases). In particular, TVWS frequency spectra, assumed to be sparse due to low occupancy - currently only digital TV is broadcast in these bands - may no longer be sparse once opportunistic radios begin operating in these frequencies. This is simply because as more bands are occupied, the sparsity assumption underlying modern recovery methods is violated. Thus any method that proposes to make decisions about spectral occupancy must contend with this.

As an alternative we can aim to reconstruct the gradient of the spectrum, as we assume that transmissions are constant within a band. This is a viable alternative as, even if the frequency spectra is not sparsely occupied, transitions between constant power transmissions will be rare thus ensuring that the gradient of the frequency spectrum will be sufficiently sparse for modern sensing methods to be successful.

Consider the basis for $\mathbb{R}^n$ defined by the function:

Definition 6.2.1 (Heaviside Basis)**.**

$$l_i(x) = \begin{cases} 1 & \text{if } x \leq i \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

That is, l_i is a left-hand step function.

The basis (6.1) can be expressed as a matrix in $\mathbb{R}^{n\times n}$, where basis vectors are written as rows of L_n:

$$L_n = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ \dots & & & & & & \\ 1 & 1 & 1 & 1 & 1 & \dots & 1 \end{pmatrix} \tag{6.2}$$

In general we write L_m for an $m \times m$ matrix of this form.

Lemma 6.2.2. *By direct computation, the inverse of L_n is:*

$$D_n = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & 0 & 0 & 0 & -1 & \dots & 1 \end{pmatrix} \tag{6.3}$$

In neural we write $D_m = L_m^{-1}$ for an $m \times m$ matrix of this form.

Definition 6.2.3. *We model our PSD signal g as a linear combination of the basis functions (6.1):*

$$g(x) = \sum_i q_i l_i(x) = L^T q \tag{6.4}$$

where $q = (q_1, \ldots, q_n)$ are the coefficients in this basis expansion, and l_i are the rows of L. Note that as defined, g is a column vector.

Proposition 6.2.4. *We can recover the coefficients q from g easily since:*

$$D_n^T g = q \tag{6.5}$$

Proof.

$$\begin{aligned} D_n^T g &= D_n^T L_n^T q && (6.6)\\ &= (L_n D_n)^T q && (6.7)\\ &= q && (6.8) \end{aligned}$$

as $L_n D_n = I_n$.

□

Definition 6.2.5. *We can define a matrix representation for the set of basis vectors l_i, by taking all inner products between all pairs of basis vectors:*

$$F_{n,ij} = \langle l_i, l_j \rangle \tag{6.9}$$

This matrix has the representation:

$$F_{n,ij} = \min(i, j) \tag{6.10}$$

An example of such a matrix is:

$$F_n = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} \tag{6.11}$$

Theorem 6.2.6. *This matrix is invertible since:*

$$\det(F_n) = 1 \tag{6.12}$$

Proof. Consider the matrix F_n. Subtract the $(n-1)$th column from the nth. This will not change the value of the determinant of F_n We obtain a matrix with 0 on the final column except the entry $F_n(n,n) = 1$.

$$F_n = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 2 & 2 & 0 \\ 1 & 2 & 3 & 3 & 0 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 2 & 3 & 4 & 1 \end{pmatrix} \tag{6.13}$$

Since the top $(n-1) \times (n-1)$ is F_{n-1} we find that

$$\det(F_n) = 1 \times \det(F_{n-1}) \tag{6.14}$$

By recursion and $\det(F_1) = 1$ we have $\det(F_n) = 1$.

□

This matrix can be factorised as $F_n = L_n L_n^T$.

From this it follows that

$$F_n^{-1} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & 0 & 0 & 0 & -1 & \dots & 1 \end{pmatrix} = D_n^T D_n \tag{6.15}$$

To find the q_i, we correlate (take the inner product of) the signal against the basis (6.1).

Definition 6.2.7 (Inner Product)**.** *We define the inner product between two vectors as follows:*

$$\langle u, v \rangle = u^T v = \sum_i u_i v_i \tag{6.16}$$

where u_i, v_i are the components of the vectors u, v in the i^{th} direction with respect to some orthonormal basis vectors e_i.

In general, we have to adapt this for basis vectors which are not orthogonal, for example:

$$\tilde{u} = \sum_i a_i l_i(x) = L_n^T a \tag{6.17}$$

$$\tilde{v} = \sum_i b_i l_i(x) = L_n^T b \tag{6.18}$$

Then, the inner product can be expressed as:

$$\langle \tilde{u}, \tilde{v} \rangle = a^T L_n L_n^T b = a^T F_n b \tag{6.19}$$

Definition 6.2.8 (Cumulative Sum)**.** *We can express the inner product between the signal g and the set of basis vectors as:*

$$h_j = \langle g, l_j \rangle \tag{6.20}$$

this is a useful quantity, as it is the cumulative sum of the signal g:

$$\begin{aligned} h_j &= \langle g, l_j \rangle && (6.21)\\ &= \sum_x g(x)\, l_j(x) && (6.22)\\ &= \sum_x \sum_i q_i l_i(x)\, l_j(x) && (6.23)\\ &= \sum_i q_i \sum_x l_i(x)\, l_j(x) && (6.24)\\ &= q_i \langle l_i, l_j \rangle && (6.25)\\ & && (6.26) \end{aligned}$$

In matrix language $h = Fq^T$. This is the inner product between the signal g and the basis functions l_i.

Example 6.2.9 (Single Rectangle)**.** *Consider a signal $g(x) \in \mathbb{R}^n$ which is*

$$g(x) = \begin{cases} 1 & \text{if } \frac{n}{4} \le x < \frac{3n}{4} \\ 0 & \text{otherwise} \end{cases} \tag{6.27}$$

in the basis defined by (6.1)*, this function can be expressed as:*

$$g(x) = L_n^T q \tag{6.28}$$

with

$$q_i = \begin{cases} -1 & \text{if } i = \frac{n}{4} \\ 1 & \text{if } i = \frac{3n}{4} \\ 0 & \text{otherwise} \end{cases} \tag{6.29}$$

1: **procedure** ESTIMATE OCCUPANCY(y, A, k)
2: **Inputs**: A set compressive measurements y, a measurement matrix A, and an integer k
3: **Returns**: $\hat{g}$, an estimate of the occupancy of a frequency spectrum.
4: Calculate $g = A^t y$
5: Choose a set of indices using procedure Index Pursuit
6: Between the indices of g take the average.
7: For each piece between the k changepoints, do a t-test to determine whether the signal is noise, or a transmission.

8: **end procedure**

1: **procedure** INDEX PURSUIT
2: **Inputs**: $g = A^t y$, L, F^{-1}, k
3: **Returns**: A set $j_{i=1}^{k}$ of indices
4: Set $z = A^t y$, inds $= \varnothing$
5: For $i = 1 : k$:
6: Calculate $u = Lz$, and $\alpha = F^{-1} u$
7: Find $\max_i |\alpha|$
8: Set $z = z - \alpha_i e_i$
9: Set inds = inds + i
10: **end procedure**

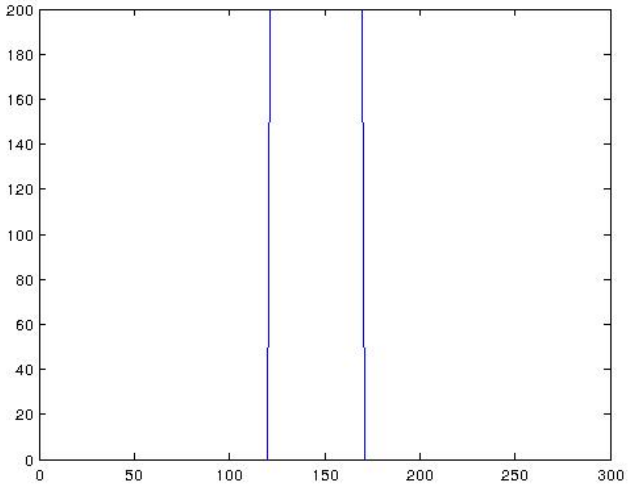

Figure 6.1: A single rectangle signal, simmilar to example 6.2.9. This signal is used as an illustrative demonstration for the method developed in this chapter.

We now describe the procedure **??**, in detail. Given a set of compressive measurements:

$$y = Ag = AL_n q^T \tag{6.30}$$

where $A \in \mathbb{R}^{m \times n}$, $A_{ij} \sim \mathcal{N}(0, 1/m)$, we then compute

$$\langle y, Al_i \rangle = q_j l_j^T A^t A l_j \sim \frac{q_j}{m} \langle l_j, l_i \rangle \tag{6.31}$$

for the set of basis vectors $l_1 \dots l_n$ i.e. the estimator from the previous section, corresponding to this set of basis functions (3.45).

Remark 6.2.10. *This is the key step in this procedure. The relation*

$$\langle y, Al_i \rangle = q_j l_j^T A^t A l_j \sim \frac{q_j}{m} \langle l_j, l_i \rangle \tag{6.32}$$

is valid because of the RIP (3.2.2) *from chapter 3. We are using the fact that the matrix $A^T A$ is δ-close to an isometry (where δ is the RIP constant of A, along with the result of theorem* (3.3.7)*, to estimate the vector h.*

We then form the vector

$$\hat{h} = m \sum_i \langle y, Al_i \rangle l_j \sim h \tag{6.33}$$

Again, this relation ($\sim$) is based on the same reasoning as the previous equation (6.31).
An example can be seen in figure 6.2, for a matrix $A \in \mathbb{R}^{200 \times 300}$.

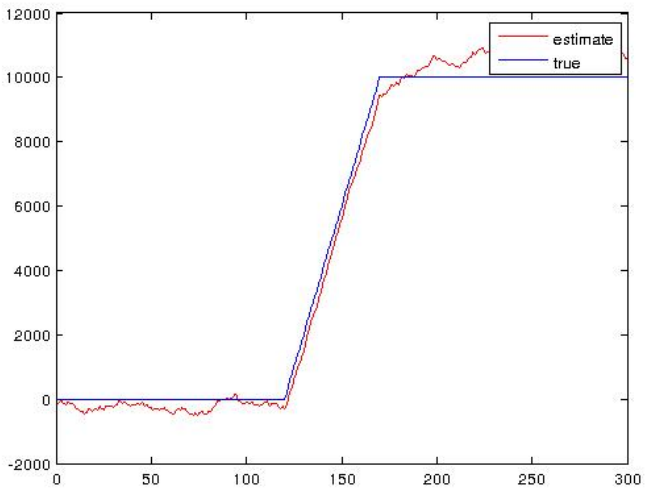

Figure 6.2: An example of the estimate $\hat{h}$ for a single rectangle signal (in red) compared to the true cumulative vector h (in blue). Note how the estimate tracks the true signal, with little deviation.

6.3 Results for Compressive Inference

In this sections we report experiments applying the compressive inference methods described in section 6.2 to a synthetic dataset. We created in signal in $\mathbb{R}^{1000}$) (pictured in figure 6.3), and ran the sensing and classification process at a variety of SNRs and undersampling points. Results are averaged over 100 runs, in each SNR regimen and at each undersampling point.

These results are collected in figures 6.4, 6.5, 6.6, and 6.7. Collectively these figure show that the performance of algorithm **??** degrades with both undersampling and a worsening radio environment. We remark that the algorithm withstands substantial noise before degrading significantly, at all levels of undersampling. For example with an SNR of -10dB, a classifier with 1% of the samples is still 90% accurate. In the more favourable SNR regime of -4.5dB, a classifier with 1% of the samples is still 95% accurate.

Figures (6.8) and (6.9) encapsulate and corroberates this insight. In particular in figure (6.8) at all levels of undersampling the performance of algorithm **??** degrades with worsening radio environments - this is evidenced by lower AUC for each curve as the SNR decreases. However, this is

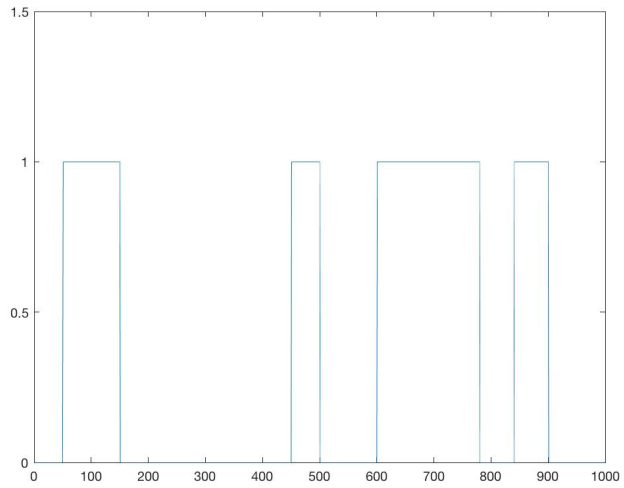

Figure 6.3: Synthetic signal used for experiments in this section

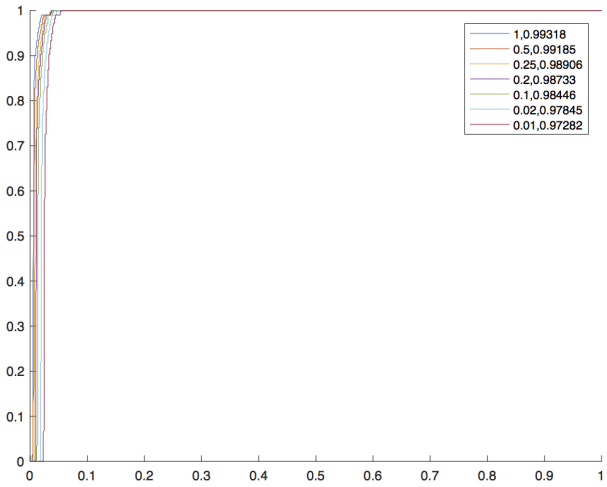

Figure 6.4: ROC curves for the single-shot algorithm (as outlined in table **??**), for a reconstruction of a the signal in 6.3. The first number in the legend is the ratio m/n, whilst the second is the area under the curve.

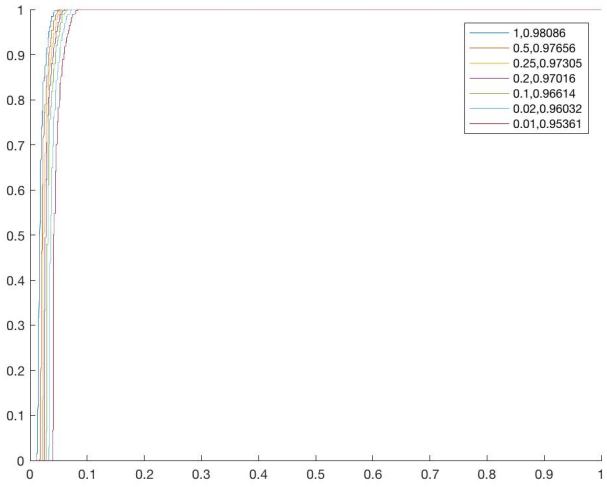

Figure 6.5: ROC curves for the single-shot algorithm (as outlined in **??**), for a signal in $\mathbb{R}^{1000}$, with noise added at an SNR of -4.5dB. The first number in the legend is the ratio m/n, whilst the second is the area under the curve.

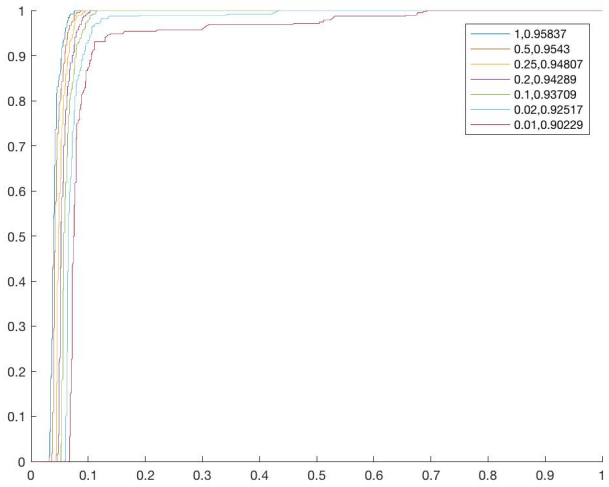

Figure 6.6: ROC curves for the single-shot algorithm (as outlined in **??**), for a signal in $\mathbb{R}^{1000}$, with noise added at an SNR of -10.5dB. The first number in the legend is the ratio m/n, whilst the second is the area under the curve.

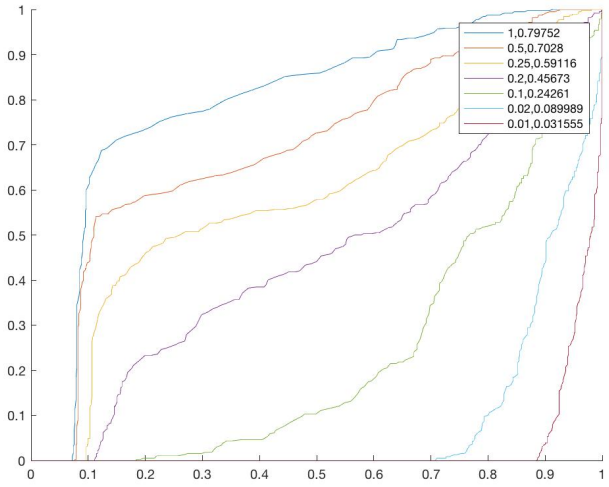

Figure 6.7: ROC curves for the single-shot algorithm (as outlined in **??**), for a signal in $\mathbb{R}^{1000}$, with noise added at an SNR of -18dB. The first number in the legend is the ratio m/n, whilst the second is the area under the curve.

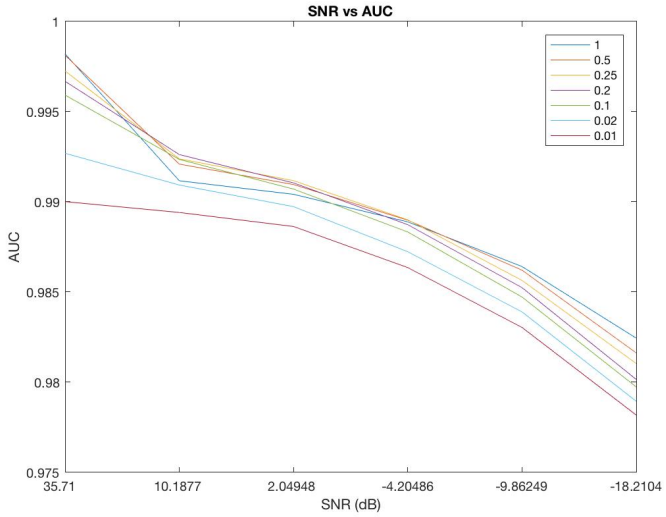

Figure 6.8: SNR vs AUC for different levels of undersampling (as indicated in the legend)

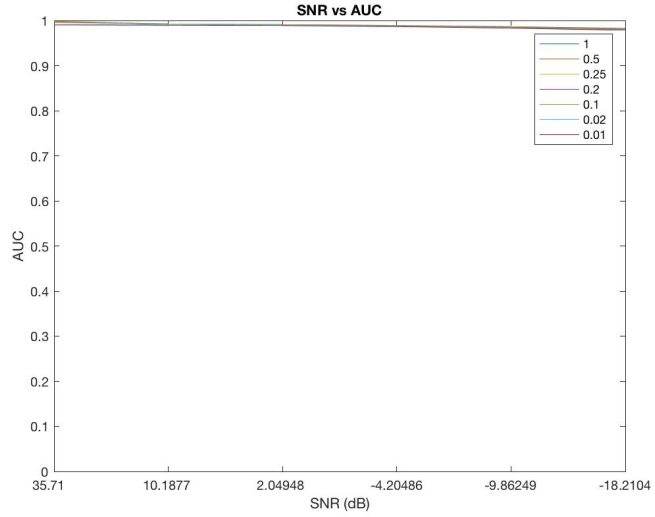

Figure 6.9: SNR vs AUC for different levels of undersampling (as indicated in the legend), this time zoomed out from figure (6.8)

6.3.1 Benchmarking the Algorithm

In this section we benchmark the algorithm (**??**), against a version of the algorithm provided with the accurate change points (as inferred in **??** by **??**). To our knowledge there is no similar prior art, applying similar methods to TVWS data. This is why we have decided to benchmark the algorithm against a hypothetical version which has access to the correct changepoints. Figures 6.10 and 6.10 show ROC curves for various undersampling levels using the synthetic signal 6.3, for algorithm **??** and **??** with oracle changepoints. Each curve was produced after 100 runs. The curves clearly show that the performance of the algorithm matches that of the oracle.

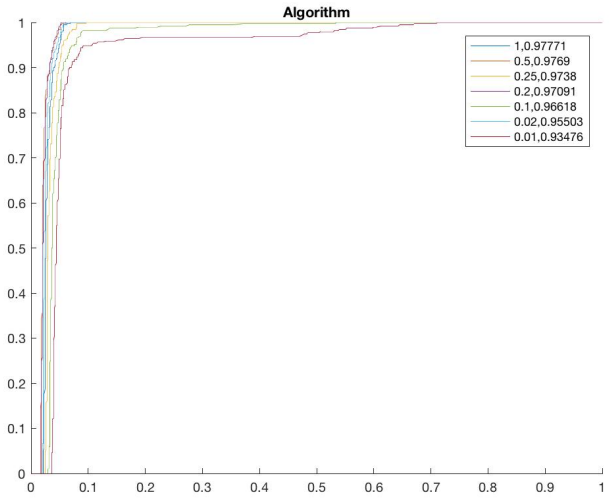

Figure 6.10: ROC curves for the single-shot algorithm (as outlined in table ,l;/l**??**), for a signal in $\mathbb{R}^{1000}$, with noise added at an SNR of -10dB. The first number in the legend is the ratio n/m, whilst the second is the area under the curve.

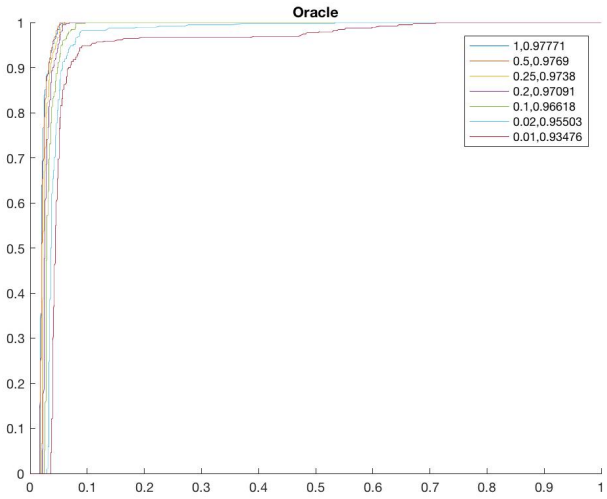

Figure 6.11: ROC curves for the single-shot algorithm with correct changepoints, for a signal in $\mathbb{R}^{1000}$, with noise added at an SNR of -10dB. The first number in the legend is the ratio n/m, whilst the second is the area under the curve.

6.4 Results for Distributed Sensing

The model described in section (6.2), equation (3.53) was simulated. The signal $g \in \mathbb{R}^{300}$ was composed of 3 rectangular pulses, mimicking primary user signals in TVWS, as shown in figure (6.12) (a). The signal was put through a Rayleigh channel, before being sensed by the nodes. The network was generated as a random geometric graph in $[0,1] \times [0,1]$, with 50 nodes. If the network wasn't connected, it was redrawn. 200 mixing patters were drawn i.i.d from a $\mathcal{N}\left(0, \sigma^2 I_{300}\right)$ distribution, with $\sigma^2 = 1/200$, to from the matrix $A \in \mathbb{R}^{200\times 300}$.

Monte Carlo simulations were performed at 18 σ_n^2 values ranging from 1 to 10 and the expected Mean Squared Error (MSE) of solutions of a centralised ADMM solver and a our distributed solver **??** were calculated over 500 repetitions with 1200 iterations (k) per repetition.

The MSE was calculated as follows:

$$\frac{\left|\left|L^t z^k - g^*\right|\right|}{\left|\left|g^*\right|\right|} \tag{6.34}$$

where z^k is the result of the algorithm at iteration k, and g^* is the optimal solution.

The SNR for each repetition was calculated as

$$\frac{\left|\left|g^*\right|\right|}{||w||} \tag{6.35}$$

and averaged over the 500 repetitions. The results are shown in figure (6.13). Following [28], for each repetition we chose

$$\lambda = \sqrt{2\sigma_n^2 \log n} \tag{6.36}$$

The error bars indicate the empirical variance across the 500 repetitions.

These results indicate that for both the centralised and distributed solvers, their performance degrades as the noise power increases in a roughly log-linear fashion. The performance of the distributed algorithm is consistently worse than the centralised version, this contrasts with results from [8]; this is due to the differing sparsity models: [8] use a joint space and frequency model for the sparsity, and as such observe an spatial averaging out of noise when using a distributed solver. The performance of DADMM is within the error bars of the centralised version at low SNR, and gap in performance between the two versions is no more than 10^{-2}. Even at relatively lower SNRs both solvers reach a solution within 10^{-1} of the optimal (as measured by normalised MSE), which will be adequate for the task of spectrum sensing. For example the reconstructions in figures (6.12) (c) and (d) show realisations of the reconstruction from DADMM with $\sigma_n^2 = 5$ and $\sigma_n^2 = 20$ respectively. It is still possible to distinguish the occupied bands from unoccupied frequencies for both reconstructions.

The distributed algorithm has consistently larger variance, than the centralised solver at all SNRs. This is due to individual nodes only having access to a subset of the data to perform calculations on:

the variance will be proportional to the square-root of number of data samples at each node, which are fewer than the total number of samples available to the centralised solver.

In figure (6.14), we plot the progress of DADMM along the solution path for a variety of regularisation parameters λ. The y-axis is the relative (unormalised) MSE between the optimal solution and the current iteration, and the x-axis is the iteration number. We note that for a fixed λ there is a single unique optimal solution, which DADMM converges to (in the sense of stationary error between consecutive iterations). This solution may not be attained in the allotted number of iterations, as the rate of convergence is determined by λ, ρ and the eigenvalues of the Laplacian of G. The paper [105], proves linear convergence for DADMM, with explicit expressions for the rate. In particular the rate convergence of DADMM is affected by the choice of λ: smaller λ corresponds to slower convergence - this is intuitive as solutions with fewer non-zero components should require fewer iterations to fully specify. Notice that for some λs the solution path exhibits phenomenological behaviour similar to damped oscillations: this phenomena has been explored in [81] and [108].

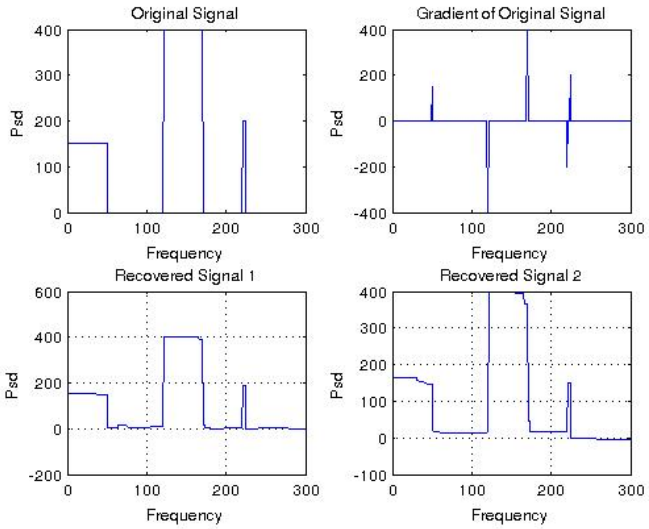

Figure 6.12: Left to right: (a) The original signal. (b) The gradient (6.2.4) of the original signal. (c) Recovery using DADMM, 1000 iterations, $\sigma_n^2 = 5$. (d) Recovery using DADMM, 1000 iterations, $\sigma_n^2 = 20$

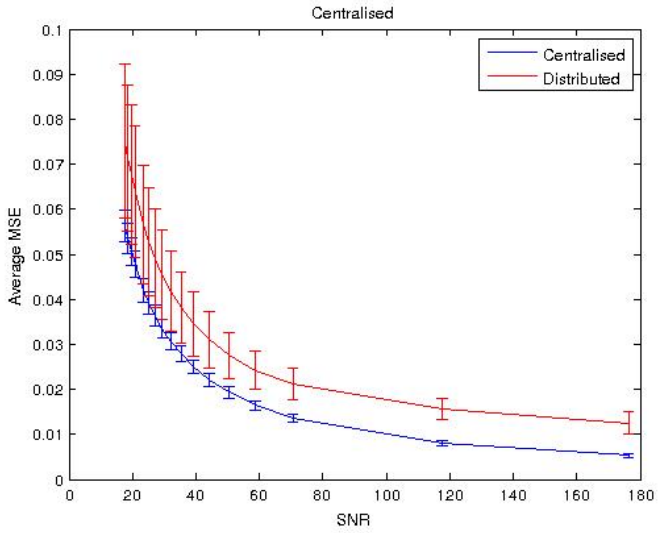

Figure 6.13: MSE vs SNR for the sensing model showing the performance of distributed and centralised solvers. The performance of DADMM is consistently within 10^{-2} of ADMM, and within the error bars of ADMM at low SNRs. The variance of estimates produced by DADMM is larger than ADMM, due to nodes performing computations on a subset of data. Both estimates are consistently within 10^{-1} of the optimal solution, which is sufficient to classify occupied bands.

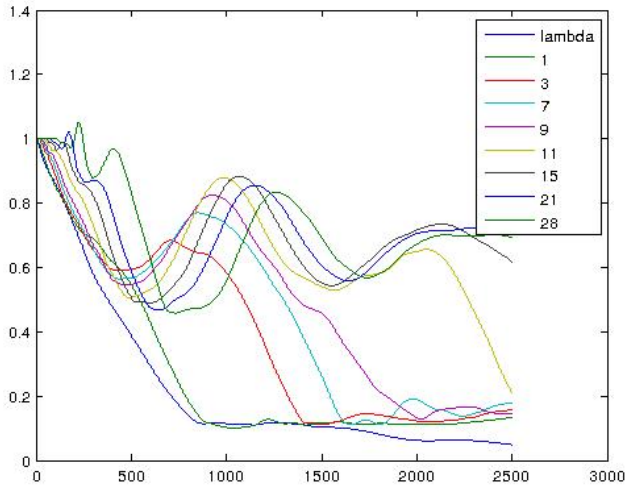

Figure 6.14: The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ. For a fixed λ there is a single unique optimal solution, with higher λ favouring sparser solutions. The convergence of DADMM is slowed by smaller λ. This is intuitive: solutions with fewer non-zero components should be identified in fewer iterations.

6.5 Results on OFCOM data

This section applies the methods of chapters 5.4 and 6, specifically algorithms 5.25, and **??** to a data set of TVWS measurements captured by OFCOM in the band 440-780MHz. The OFCOM dataset was taken in Southwark, UK during May 2014.

This section is organised as follows: we first describe the dataset in detail, and show example PSD captures for it, then we show results of the distributed estimation algorithm (presented in chapter (5.4)), with estimation in the Heavyside basis. This algorithm is performed over a network of nodes, and reconstructs the sensed PSD via an iterative denoising procedure derived from the LASSO functional. It should be noted that this is the largest experiment of its kind: previous work has featured experimental results on far smaller datasets, or featured algorithms who's steps took considerably longer to reach an acceptable solution. We compare results produced by our algorithm to results from cooperative algorithms, and discuss the relative performance in terms of reconstruction quality, classification accuracy and algorithm speed.

We then present results of the techniques outlined earlier in in this chapter (6), specifically algorithms **??** on the OFCOM data set. This algorithm aims at classifying the signal directly from compressive measurements without reconstructing the signal as an intermediate step. The major feature of this methods is that classification can still be done accurately with severe undersampling.

6.5.1 Data Set

This data set was kindly supplied by OFCOM, and is of the 440-790 MHz TVWS band, with a resolution of 25 kHz. This is 1,048,575 data points, or 80 captures of the TVWS band. A single snapshot of the TVWS band is 10,000 data points, The data set is 15.7 MB in size and was captured at Riverside House, Southwark, London, over a period of 1 hour in June 2014.

There was no ground truth signal provided with the data set, so there is no way to quantitatively asses the goodness of our methods on this dataset. However, since the algorithms have been shown to perform well on suitable synthetic data, we be live that the results obtained in this section are valid.

6.5.2 Results: Distributed Estimation with Heaviside Basis

This subsection shows some examples of recovery of the OFCOM data with the DADMM-Lasso algorithm, as described in chapter 5.4. Each example is with a different value of λ the regularisation parameter, which trades off recovery accuracy with recovery sparsity.

These figures show that qualitatively, the distributed algorithm is capable of reconstructing the PSD in the TVWS band. These plots show that the trade off between reconstruction accuracy (in the mean square sense) and sparsity can be quite severe in this basis. We theorise that this is because of co-linearity between adjacent basis functions. This is a known feature of the lasso, and could possibly

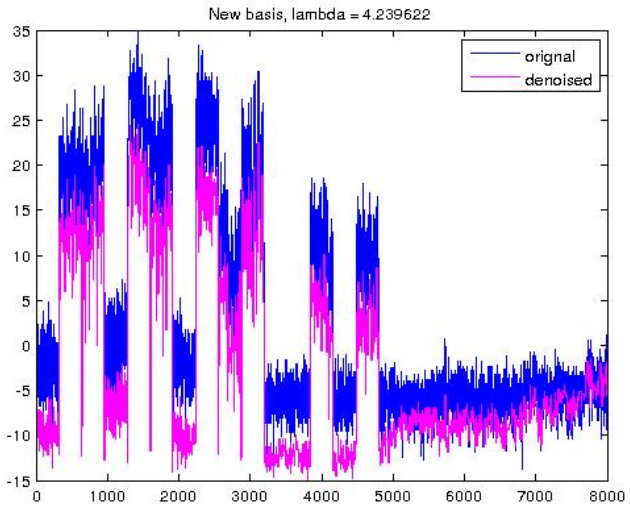

Figure 6.15:

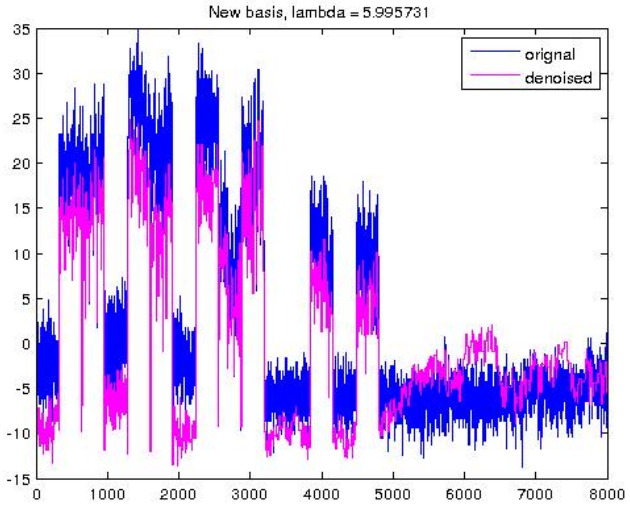

Figure 6.16:

be alleviated by using a different reguluriser e.g. the OWL or OSCAR penalties. It is not clear that these penalties will be amenable to distributed implementation, however.

These plots show that a distributed network is in principle capable of performing large-scale reconstruction of PSD from compressive samples. In particular, no single node has access to the entire set of samples - so the compression rate per node could be quite severe. It should be noted that to our knowledge, this is the only study using data sets this size and with representative real-world data.

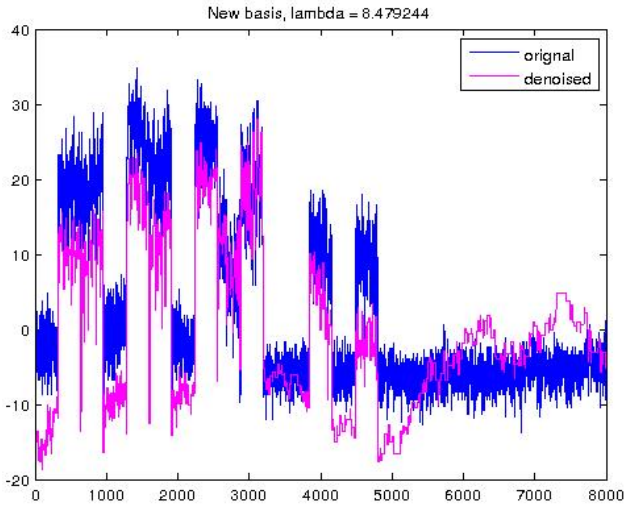

Figure 6.17:

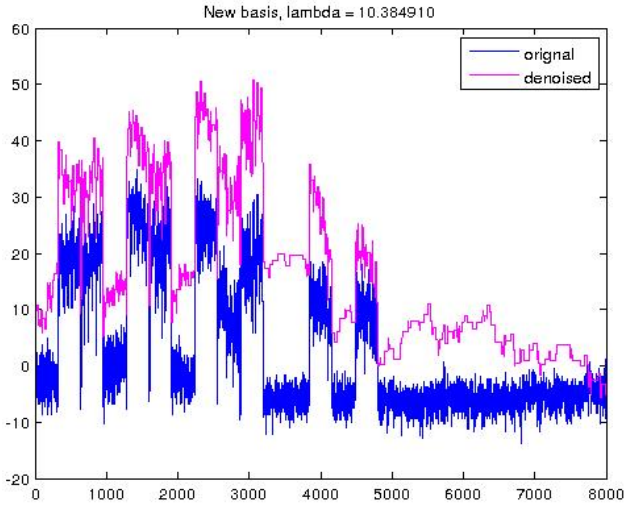

Figure 6.18:

6.5.3 Compressive Estimation

In this section, we apply display some examples of the output of procedure **??**, as applied to the OFCOM data set.

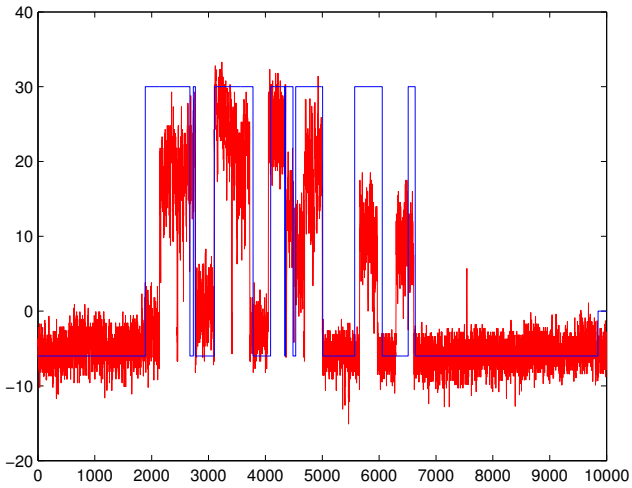

Figure 6.19: Example of classification with OFCOM data, 35 changepoints

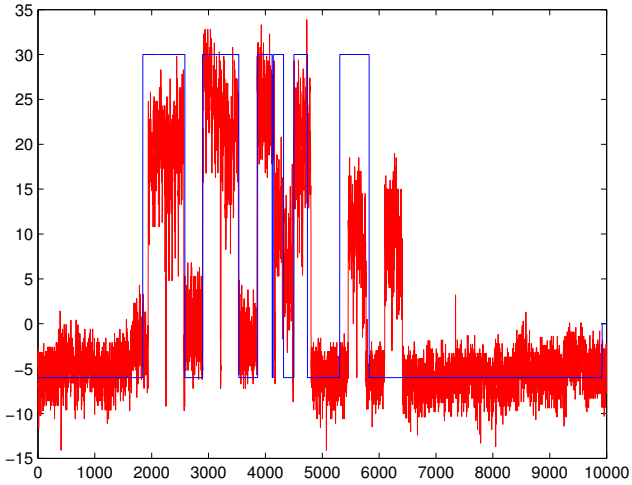

Figure 6.20: Example of classification with OFCOM data, 85 changepoints

These graphs show that the compressive estimation algorithm is capable of accurately classifying TVWS with datasets of this size. The accuracy of classification is highly dependent upon the number of changepoints used.

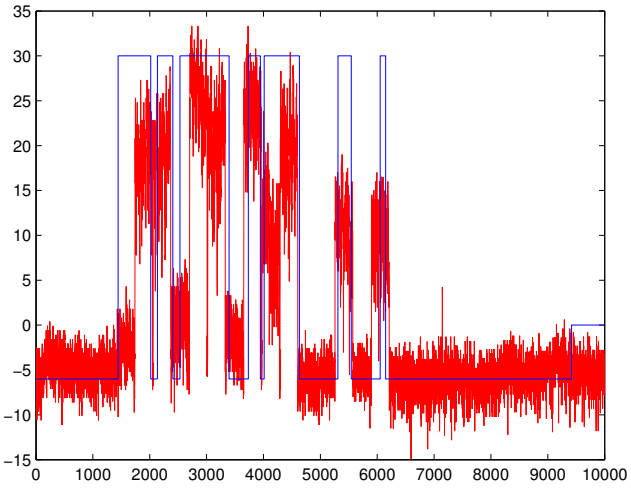

Figure 6.21: Example of classification with OFCOM data, 55 changepoints

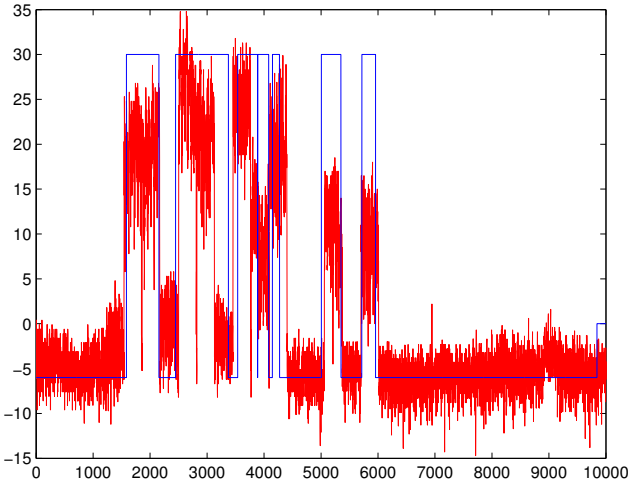

Figure 6.22: Example of classification with OFCOM data, 55 changepoints

CHAPTER 7

SENSING WITH HEAVISIDE BASIS IN TIME AND FREQUENCY

7.1 Introduction

In this chapter we develop a joint model for TVWS spectrum sensing by incorporating frequency spectra gradients and time. The key insight is that static signals (such as TV transmission) will be sparse in time as the rarely change. We can make use of this by creating a model to detect ch ages in the signal across time. In this chapter we develop such a model, as a two-dimensional extension of the ideas of chapter 6.

To create a classification from measurements we develop a compressive and non-compressive method for estimating the joint model. The non-compressive model is a direct extension of ideas from chapter 6, whilst the compressive method reduces a multi-time slot method to a single-time slot method.

We apply these methods to synthetic and real data (again supplied by OFCOM), and discuss the results.

7.2 Time-Frequency Model

In this section we extend the model from section (6.2), to jointly estimate a collection of signals sensed over time.

The rationale for this extension is that many signals are (approximately) constant over time, and thus exit an extra dimension of sparsity. Namely, when expanded in an appropriate basis a multi-time slot signal can be very sparse. This is because we need only sense changes from the underlying initial signal (e.g. a transmission turns on then off), as we assume that the underlying TV transmission is essentially constant.

In this case, we have a series frequency spectra $\in \mathbb{R}^n$, sensed at T time instants. Consider a basis for $\mathbb{R}^{n\times t}$ defined by the function:

Definition 7.2.1.

$$l_{i,t}(x,\tau) = \mathbb{I}(t \leq \tau)\mathbb{I}(x \leq i) \tag{7.1}$$

Remark 7.2.2. *We can express these basis functions in terms of the basis functions* (6.1)*, and similarly defined basis functions over time:*

$$l_t(\tau) = \begin{cases} 1 & \text{if } t \leq \tau \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

That is, l_t is again a left-hand step function.

Definition 7.2.3 (Tensor Product)**.** *For two matrices $A \in \mathbb{R}^{i\times j}$ and $B \in \mathbb{R}^{k\times l}$ the tensor product is defined as:*

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1j}B \\ a_{21}B & 1a_{22}B & \dots & a_{2j}B \\ \dots & & & \\ a_{i1}B & a_{i2}B & \dots & a_{ij}B \end{pmatrix} \tag{7.3}$$

The resulting matrix has dimension $ik \times jl$.
A useful property of tensor products is:

$$(A \otimes B)(C \otimes D) = AC \otimes BD \tag{7.4}$$

Proposition 7.2.4.

$$l_{i,t}(x,\tau) = l_i(x) \otimes l_t(\tau) \tag{7.5}$$

Proof. Consider a fixed frequency, w.l.o.g $i = 1$. Then

$$l_{1,t}(x,\tau) = \mathbb{I}(x \leq 1)\mathbb{I}(t \leq \tau) \tag{7.6}$$

There are T possible vectors, each of length n, for this combination of (i,t). Therefore $l_{1,t} \in \mathbb{R}^{1\times nT}$. As there are n possible frequency vectors, one for each fixed frequency basis function, we have $n \times nT$ vectors. By symmetry of time and frequency, the same argument applies for a fixed time, and we have T, $1 \times nT$ basis vectors in total, one for each fixed time instant. Therefore the joint time-frequency basis is in $\mathbb{R}^{nT\times nT}$. As time and frequency are orthogonal in this representation, the proposition follows. □

Thus this basis can be represented with the following matrix in $\mathbb{R}^{(nT)\times(nT)}$:

Definition 7.2.5.

$$L = L_T \otimes L_n \tag{7.7}$$

where L_k is the matrix (6.2.1) *of size $k \times k$.*

Lemma 7.2.6. *By direct computation, it follows that:*

$$D = L^{-1} = L_T^{-1} \otimes L_n^{-1} = D_T \otimes D_n \tag{7.8}$$

where D_k is the same as (6.2.2)*, and that:*

$$\begin{aligned} F &= LL^T && (7.9)\\ &= (L_n \otimes L_T)(L_n \otimes L_T)^T && (7.10)\\ &= L_T L_T^T \otimes L_n L_n^T && (7.11)\\ &= F_T \otimes F_n && (7.12) \end{aligned}$$

where F_k is defined as in (6.11)*.*

We sense the frequency spectrum at T instants, collecting signals $g_1, \dots g_\tau$.
similarly to 6.4 we model the signal as a linear combination of basis vectors:

$$g = L^T a = \left(L_T^T \otimes L_n^T\right) a \tag{7.13}$$

Definition 7.2.7. *Given a collection of vectors, $\{a\}_{t=1}^T$, we write*

$$a = \operatorname{vec}(a_t) = \sum_{t=1}^{T} e_t \otimes a_t \tag{7.14}$$

where e_t is the standard basis vector:

$$e_t = \begin{cases} 1 & \text{if } t \le T \\ 0 & \text{otherwise} \end{cases} \tag{7.15}$$

We can find an expression for a by taking the inner product of g with $l_{j,s}$ (as (6.20)):

$$\begin{aligned} h &= \langle g, l_{j,s}(x,\tau) \rangle \\ &= \sum_{i,t} g(x,\tau) \langle l_{i,t}, l_{j,s} \rangle \\ &= Fa \end{aligned}$$

The measurements in this model can be written as

$$y_T = A_T g \tag{7.16}$$

with

$$A_T = I_T \otimes A \tag{7.17}$$

and $A \sim N(0, 1/m)$.

Proposition 7.2.8. *We can write the i^{th} column of the matrix L_k as, for $k \leq j$:*

$$L_k^T e_i = \sum_{j=1}^{i} e_j \tag{7.18}$$

where e_j is the j^{th} canonical basis vector in $\mathbb{R}^k$, with has a 1 in position j and is zeros everywhere else.

Proof. The i^{th} column of L_k will have i ones, and $k-i$ zeros. It can be written in terms of the canonical basis e_j, simply as sum up to the last 1 in position i. □

In this case, we can write g as:

$$\begin{aligned} g = L^T a &= \left(L_T^T \otimes L_n^T\right)\left(\sum_{t=1}^{T} e_t \otimes a_t\right) \\ &= \sum_{t=1}^{T} L_T^T e_t \otimes L_n^T a_t \\ &= \sum_{t=1}^{T}\left(\sum_{j=1}^{t} e_j\right) \otimes L_n^T a_t \\ &= \sum_{j=1}^{T} e_j \otimes \sum_{t=j}^{t} L_n^T a_t \\ &= \sum_{j=1}^{T} e_j \otimes b_j \end{aligned}$$

where we define $b_j = \sum_{t=j}^{T} L_n^T a_t$. We can now write

$$\begin{aligned} y_T = A_T g &= (I_T \otimes A)\left(\sum_{j=1}^{T} e_i \otimes b_j\right) \\ &= \sum_{j=1}^{T} e_j \otimes A b_j \end{aligned}$$

Example 7.2.9 (Identical a_t)**.** *Now consider the case where all the a_t are identical. In this case:*

$$\begin{cases} a_T = a & \text{if } t = T \\ a_t = 0 & \text{if } t < T \end{cases} \tag{7.19}$$

so

$$b_j = \sum_{t=j}^{T} L_n^T a_t = L_n^T a = b \tag{7.20}$$

Then

$$y_T = \left(\sum_{j=1}^{T} \right) \otimes Ab = 1_T \otimes Ab \tag{7.21}$$

Let the s^{th} column of L_T^T be denoted by B_s, then

$$\left(L_T^T\right)^{-1} B_s = D_T^T B_s = e_s$$

As before, we form the estimator

$$\hat{h}_T = \sum_{i,t} \langle y_T, A_T l_{i,t} \rangle l_{i,t} \sim \frac{1}{mT} Fa$$

Consider

$$\begin{aligned}
\langle y_T, A_T l_{i,t} \rangle &= \langle A_T^T y_T, l_{i,t} \rangle \\
&= \langle \left(I_T \otimes A^T\right) \left(\sum_{j=1}^{T} e_j \otimes Ab_j \right), (l_t \otimes l_i) \rangle \\
&= \langle \sum_{j=1}^{T} e_j \otimes A^T Ab_j, (l_t \otimes l_i) \rangle \\
&= \sum_{j=1}^{T} e_j^T l_t \otimes b_j^T A^T A l_i \\
&\sim \frac{1}{m} \sum_{j=1}^{T} e_j^T l_t \otimes b_j^T l_i \\
&\stackrel{?}{=} \frac{1}{m} \sum_{j=1}^{T} j \otimes F_n a
\end{aligned}$$

where we used the expectation of a Wishart matrix between the 5^{th} and 6^{th} lines.

7.2.1 Compressive Method

Definition 7.2.10 (Period-change vector)**.** *Let*

$$B = D \otimes I_T \tag{7.22}$$

then we define

$$z = By \tag{7.23}$$

This vector, identifies consecutive time slots in which the signal g is constant (i.e. does not change between sensing instants). As example, consider the following 5-slot signal:

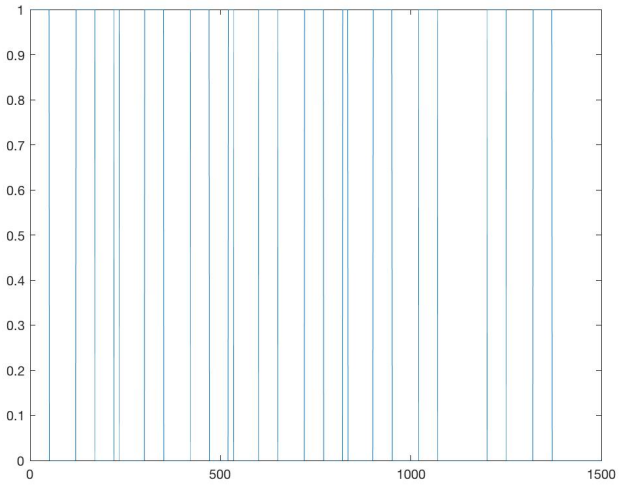

Figure 7.1: Example multiple time-slot signal

Whilst the compressive measurements of this signal are not revealing:

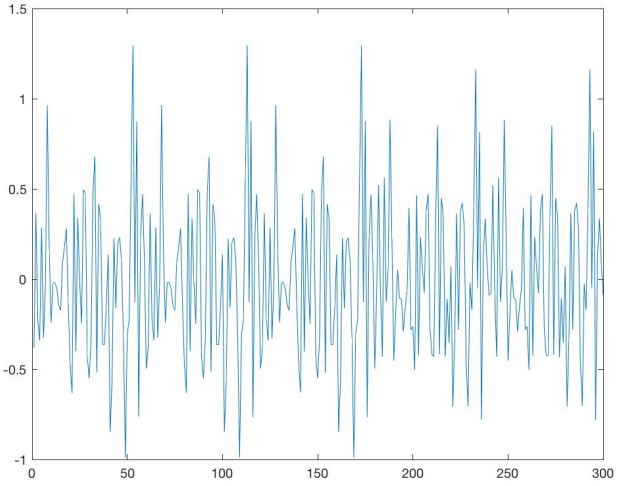

Figure 7.2: Compressive Measurements of the previous signal **??**

The period change vector clearly identifies the periods where the original signal changes:

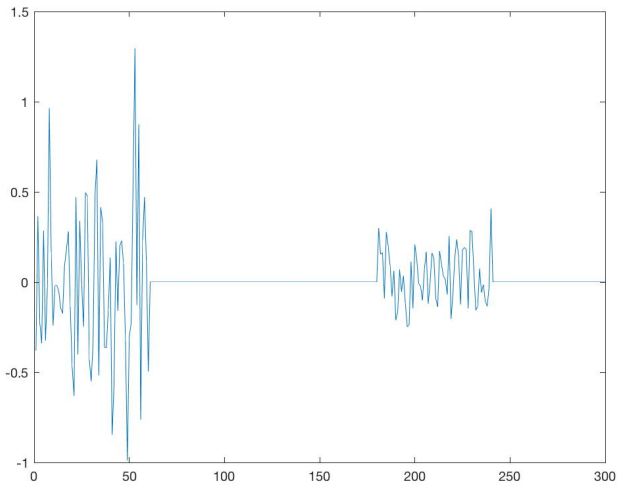

Figure 7.3: Period-change vector for the signal **??**

We can use this vector to estimate a multiple time-slot signal. We use the following algorithm:

1: **procedure** ESTIMATE OCCUPANCY MULTIPLE SLOT(y, A, k)
2: A set compressive measurements y_T, a measurement matrix A_T, and an integer k
3: **Returns**A estimate of the occupancy of a frequency spectrum, along multiple time slots.
4: Compute $z = By$ as per (7.2.10)
5: Count the number of zero consecutive parts of z.
6: For each group, take the mean of y_T for those $t \in T$ belonging to that group, and use the procedure **??** to form an estimate of those groups.
7: **end procedure**

Figure 7.4: Procedure for estimating occupancy of frequency spectra with multiple time slots

7.2.2 Non-compressive Method

In this section we introduce a multi time-slot generalisation to the estimator h (6.20), adapted to the new basis.

Definition 7.2.11 (Time-difference Vector)**.** *We consider the estimator:*

$$w = D^T A^T y \tag{7.24}$$

Remark 7.2.12. *We can see that this is an estimate of α_T as:*

$$\begin{aligned} w &= D^T A^T y && (7.25)\\ &= D^T A^T A g_T && (7.26)\\ &= \left(D_T^T \otimes D_n^T\right)\left(I_T \otimes A_n^T\right)\left(I_T \otimes A_n\right) g && (7.27)\\ & && (7.28) \end{aligned}$$

Example 7.2.13 (Constant Signal)**.** *i.e. the underlying signal remains constant over multiple time-slots.*

We define the sensing matrix as:

$$A = I_T \otimes A_n \tag{7.29}$$

Thus we have

$$y = Ag = 1_T \otimes A_n g_n \tag{7.30}$$

and

$$\begin{aligned} \alpha &= D^T g && (7.31)\\ &= \left(D_T^T \otimes D_n^T\right)\left(I_T \otimes g_n\right) && (7.32)\\ &= \left(D_T^T \otimes D_n^T g_n\right) && (7.33)\\ &= e_T \otimes \alpha_n && (7.34) \end{aligned}$$

So, our estimator (7.2.11) *is, in this instance:*

$$w = D^T A^T y \tag{7.35}$$
$$= D^T A^T A g_T \tag{7.36}$$
$$= \left(D_T^T \otimes D_n^T\right)\left(I_T \otimes A_n\right)\left(1_T \otimes A_n g_n\right) \tag{7.37}$$
$$= e_T \otimes \left(D_n^T A_n^T A_n L_n^T \alpha_n\right) \tag{7.38}$$
$$\sim e_T \otimes \left(D_n^T L_n^T \alpha_n\right) \tag{7.39}$$
$$= e_T \otimes \alpha_n \tag{7.40}$$
$$= \alpha_T \tag{7.41}$$

7.3 Results

We now present some results for the multiple-stage algorithm 7.4.

7.3.1 Synthetic Data

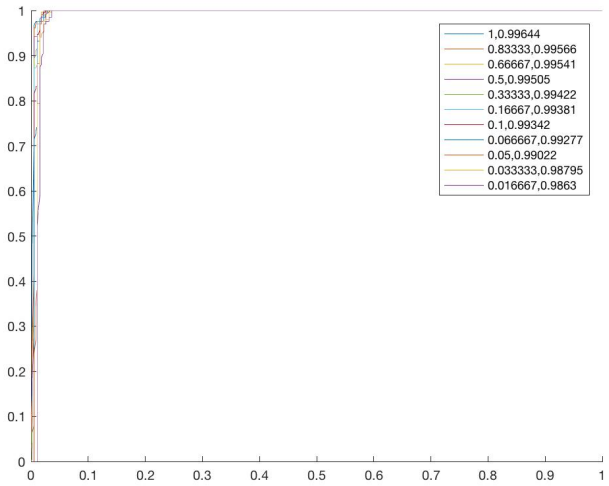

Figure 7.5: ROC curves for the multi-shot algorithm (as outlined in **??**), for a signal in $\mathbb{R}^{300}$, over 5 time slots. The first number in the legend is the ratio n/m, whilst the second is the area under the curve.

In this instance the algorithm is able to provide an estimate, from only two groups i.e. it is able to infer that there is a change in the signal between the 3rd and 4th periods, and use this information to estimate the entire signal.

7.3.2 OFCOM Data

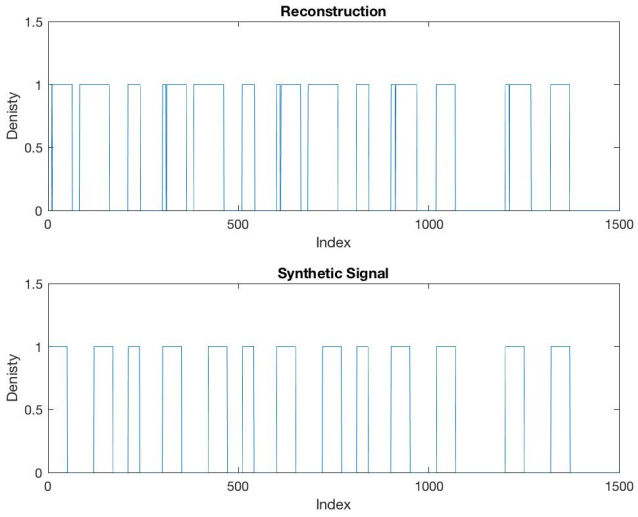

Figure 7.6:

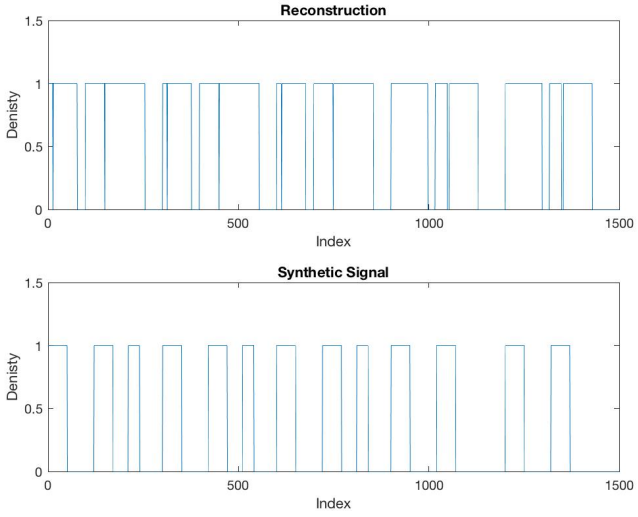

Figure 7.7:

CHAPTER 8

GROUP TESTING

8.1 Introduction and notation

This chapter outlines a contribution to the problem of group testing with non-identical probabilities. The material presented here was previously presented at Allerton 2014.

Broadly, Group Testing is a sparse inference problem where the aim is to identify a small set of positive items contained within a much larger set of negative items, whilst aiming to minimise the number of times the items are checked (testes). As an example, consider trying to identify a sack containing coins weighing 1.1g from 9 other sacks containing coins weighing 1g, using an electric scale. This can be achieved in a single weighing of the coins: enuerate the bags, and take 1 coin from bag 1, 2 from bag 2 etc. Weigh this set of coins. The total should be 55g, should all the coins weigh 1g. The ammont in 0.1g this ammount exceeds tells us which bag has the heavier coins.

Group testing shares many features with compressive sensing. In particular, sparsity and randomness. In both scenarios, we are searching for the non-zero parts of a sparse vector; in both, random linear combinations of items are ued to reduce the number of tests/sample rate.

This chapter introduces the group testing problem, convers relevant literature on variations of this problem, and covers some applications of group testing. It defines the idea of group testing capacity, as well as algorithms which achieve this capacity. In particular we discuss an algorithm of Hwang, and illustrate how it achieves the capacity.

We then extend the definition of probabilistic group testing to the case where the item probabilities are non-identical. We extend the definition of Group Testing capacity similarly. We then give an extension of the Laminar algorithm (presented in [64]) combining ideas from Hwang's algorithm to form a capacity achieving algorithm in the non-identical probabilites case.

We then present an analysis of said algorithm, proving that the algorith does indeed achieve

capacity.

8.1.1 Group Testing

Group Testing originated in the second world war because of the need to test all incoming conscriptees for syphilis. It would have been inefficient and expensive to test each soldier individually, as the rate for syphilis was only 10 per 100,000. Dorfman [40], considered the idea of pooling blood samples and testing the pooled samples for syphilis and only further testing the pools which come up positive.

A typical problem that can be solved by Group Testing is finding a counterfeit coin in a group of otherwise identical coins by weighing groups of coins on a pan balance. For example, given 80 coins known to contain a single counterfeit, which is lighter than the others, what is the minimum number of weighings needed to determine the counterfeit with certainty? You may get lucky and pick the counterfeit in the for the first go: but there's only a $\frac{1}{80}$ chance of that happening. There's also no need to check all $\binom{80}{1}$ combinations of pairs of coins. However, putting more than one coin on a pan reveals the same information - it's better to weigh groups of coins against each other.

Choose the groups so that each weighing can distinguish between the hypothesis that the pans will balance, or than there will be a heavier pan i.e. split the initial group into 3 (27, 27, 26). Continue this process recursively, splitting the remaining group into 3 each time, until you have found the counterfeit. If the two groups of 27 balance initially, take a coin from one of those groups and add it to the group of 26 to make a power of 3. This won't add any new information (you know this coin is not counterfeit) and so won't affect the inference.

The Group Testing problem can be formalised as follows: a set of items is given, along with an upper bound on the number of defectives. The set is described as a vector, where if an item is 0 it is not defective and 1 if it is defective. Before the tests are run, the position of the 1's is unknown.

To find the defective items, a query is run against a subset of $[n]$, where the answer is defined as follows:

$$A(S) = 1\sum_{i} x_i \geq 1 \tag{8.1}$$

Note that the addition is the binary-or in the above summation. The goal of Group Testing is to minimise the number of tests required to reconstruct the defective set.

Algorithms

An initial algorithm to consider is a simple binary search of the set to be tested. That is, given a set of size $N = 2^r$ i.e a power of 2, we can recover a single defective in $\lceil \log_2 N \rceil$ tests.

To do this, create a new set of size $S = 2^{\lceil \log_2 n \rceil}$ which is guaranteed to contain a defective. Label the items with integers, and test the items in the sets $1, 2, \ldots, S/2$ and $S/2+1, \ldots, S/2$ separately. Then repeat the procedure on any groups which have a positive test.

To see why this testing procedure takes at most $\lceil \log_2 n \rceil$ tests, note that the procedure defines a binary tree over subsets of the N items, and so the depth of this tree is $\lceil \log_2 N \rceil$.

For input sets with more than a single defective (say K defectives) the binary search algorithm can be repeated, and each time a defective is found it is removed from the set. The binary search is then repeated, but on a set of size $N-1$. Using this procedure we are guaranteed to find all the defectives in

$$K\lceil \log_2 N \rceil \leq K \log_2 N + K \tag{8.2}$$

tests. However, this is a very inefficient algorithm: early sets are large and so are likely to contain a defective.

The above algorithms return, with certainty, after at most $K\lceil \log_2 \binom{N}{K} \rceil$ tests, the defective set. Much work has gone into combinatorial search algorithms, often more complex than those described above.

This has been motivated by the analogy that the Group Testing problem can be considered a decoding problem where an experimenter receives a binary vector:

$$y = \mathbf{A}x \tag{8.3}$$

$y \in \{0,1\}^K, x \in \{0,1\}^N$, and wishes to decode the vector x to recover the defective set, subject to the constraints of the testing matrix $\mathbf{A}$. The matrix has to satisfy the property that the Boolean sum of any t columns was unique, and did not contain any other column in the matrix. These properties are known as seperability and disjunctness.

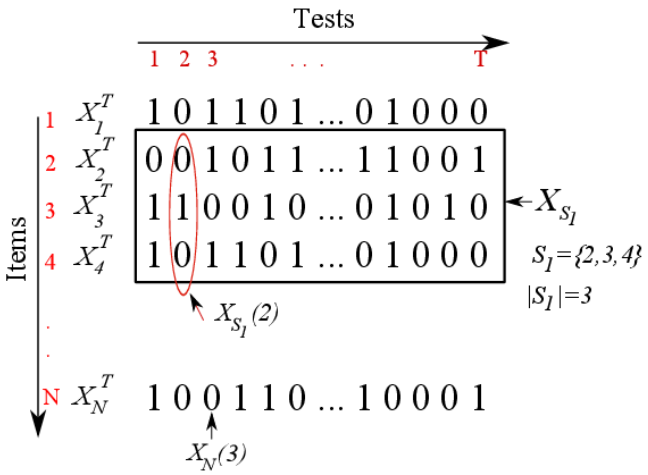

Figure 8.1: The Group Testing model: multiplication with a short, fat matrix [3]

See [42] for more a detailed introduction and analysis of the requisite algorithms.

Hwang's Algorithm

In modern Coding Theory, there has been a move away from exlicit combinatorial algorithms which return the codeword with certainty, towards probabalisitic algorithms which return the correct codeword with an associated probability. The advantage of this has been the development of algorithms which can decode codes close to the Shannon capacity of the channel.

Similarly in Group Testing, the state of the art considers probabilistic algorithms instead of explicit combinatorial designs.

The problems with the binary search algorithm (that initial groups are very large and are highly likely to contain a defective) above can be overcome by instead considering groups whose size is chosen so that the probability that the group will have a positive test is close to half. Equivalently, given a set of size $N = 2^r$ which are known to contain K defectives, in expectation a group of size $\frac{K}{N}$ should contain a defective.

Thus we can use fewer tests than predicted by simple repeated binary search, by testing 'pilot' groups of size roughly $\frac{K}{N}$. Hwang [**?**] gives such an algorithm, and provides an upper bound on the number of tests required to recover the defective set.

The steps for the algorithm are:

1. If $n \leq 2d-2$ then test every item individually. Otherwise set $l = n-d+1$ and define $\alpha := \log \lceil \frac{l}{d} \rceil$.

2. Test a group of size 2^α. If the outcome is negative, the groups is good. Set $n := n-2^\alpha$ and go to 1. If the outcome is positive, then use binary splitting on the group to identify a defective and x good items. Set $n := n-1-x$ and $d := d-1$ and go to 1.

The upper bound on the number of tests is given by the following argument: as there are $\binom{n}{k}$ possible sets of defectives, and in t tests at most 2^t cases can be differentiated, $\lceil \log_2 \binom{n}{k} \rceil$ tests are needed.

Bounds

It has been previously believed that the success probability to recover the defective set given T tests was:

$$P(Success) \leq \frac{T}{\log_2 \binom{N}{K}} \tag{8.4}$$

However, a tighter upper bound has recently been found [**?**] at:

$$P(Success) \leq \frac{2^T}{\binom{N}{K}} \tag{8.5}$$

i.e. the probability of success increases exponentially with the number tests, opposed to linearly.

In the Group Testing literature there exists an 'adaptivity gap' - it seems that adaptive algorithms *do* give a performance improvement over non-adaptive algorithms, in terms of the number of tests required to recover the defective set. This is discussed here, using Hwang's algorithm as a test bed.

Hwang's algorithm is guaranteed to succeed in:

$$T = \log_2 \binom{N}{K} + K \tag{8.6}$$

tests. The Combinatorial Orthogonal Matching Pursuit algorithm, considered in [25], is guaranteed to recover the defective set with probability $N^{-\delta}$ in

$$T = ((1+\delta)\, e)\, K \ln N \tag{8.7}$$

tests. For all N and K) we have:

$$K \log_2 \frac{N}{K} \leq \log_2 \binom{N}{K} \leq K \log_2 \frac{Ne}{K} \tag{8.8}$$

which follows from well-know bounds on binomial coefficients. This allows a contrast between the asymptotic bounds of previous algorithms to be considered in this section. We see that, the regime where $K = N^{1-\beta}$, Hwang's algorithm succeeds with:

$$T = \beta K \log_2 N + K\left(\log_2 e + 1\right) \tag{8.9}$$

tests, whilst the COMP algorithm succeeds with:

$$1.88\,(1+\delta)\, K \log_2 N \tag{8.10}$$

tests. It's worthwhile to contrast these results, to gain some insight into the problem. 8.1.1 suggests that for very sparse problems (β tending towards 1) that Hwang's adaptive algorithm will outperform a simmilar non-adaptive algorithm. Even though the two procedures have the same complexity, they have different constants (1 v,s 1.88 in the sparse case). Thus, there are asymptotic gains (in terms of the number of tests required to recover the defective set) which are offered by adaptive algorithms, and not by non-adaptive algorithms.

These ideas can be summarised in the idea of a *capacity* for Group Testing [**?**]. That is, there is a constant C such that a sequence of Group Testing algorithms with $K = \omega(N)$ will succeed with probability tending to 1. This allows different noise, and dilution models to be considered so that a more complete characterisation of the structural properties of Group Testing is revealed.

Comparison to Compressive Sensing

The goal of Coding Theory is given a vector $x \in \mathbb{F}^m$, where $\mathbb{F}$ is some finite-field, is to construct a 'code-book' C which produces a vector $y \in \mathbb{F}^n$, $n > m$, so that the original vector may be transmitted over a noisy-channel with vanishing error probability. This problem is structurally similar to the

Compressive Sensing and Group Testing problems, but in reverse. In CS and GT we're given 'short' vector, and we wish to infer the 'longer' one satisfying the constraint that we seek the sparsest vector, under some conditions on the matrix Φ. This suggests that there may be some Information-theoretic framework uniting all three disciplines.

In [20] Tao and Candes consider the CS problem as one of error correction of a linear code: however in this case the codewords are drawn from $\mathbb{R}^m$ as opposed to a finite alphabet more common in Coding Theory. This is done by considering Φ as the parity check matrix of a linear code and the signal x as the error pattern. Linear programming can then be viewed as a method for decoding.

Group testing is a combinatorial variant of Compressive Sensing, where the sensing matrix is a binary matrix. The matrix represents combinations (or pools) of items, such that a 1 in the i^{th} row and j^{th} column means that the i^{th} item is tested in the j^{th} pool. The goal of Group testing can then be seen as designing testing pools so to accurately reconstruct the sparse set of interesting items.

In Group Testing, instead of the sensing matrix being to subject to coherence constraints such as those above, the sensing matrices have the property that the support of any column is not contained in the union of the supports of any t other columns. Thus a t-disjunct matrix defines a group testing scheme which an identify any defective set up to size t.

The analogue between Group Testing and Coding is even closer, as GT explicitly considers signals and matrices from Binary alphabets. That is, Group Testing is a closer cousin of Coding Theory than Compressive Sensing, in a sense the inverse problem as in both Coding and Group Testing we are working over a finite field. This is encouraging, as it could allow the reconstruction of the defective set via methods developed in Coding Theory. There has been some work done on this, [100] considers the noisy Group Testing problem and the reconstruction of the defective set via belief propagation whilst [121] gives explicit theorems on conditions for the recovery of the defective set for the case of the binary symmetric channel. [**?**] takes this further and finds the capacity of Group Testing for a number of cases.

8.1.2 The Probabilistic group testing problem

Group testing is a sparse inference problem, first introduced by Dorfman [**?**] in the context of testing for rare diseases. Given a large population of items $\mathscr{P}$, indexed by $\{1,\dots N\}$, where some small fraction of the items are interesting in some way, how can we find the interesting items efficiently?

We perform a sequence of T pooled tests defined by test sets $\mathscr{X}_1,\dots,\mathscr{X}_T$, where each $\mathscr{X}_i \subseteq \mathscr{P}$. We represent the interesting ('defective') items by a random vector $\mathbf{U} = (U_1,\dots,U_N)$, where U_i is the indicator of the event that item i is defective. For each test i, we jointly test all the items in $\mathscr{X}_i$, and the outcome y_i is 'positive' ($y_i = 1$) if and only if any item in $\mathscr{X}_i$ is defective. In other words, $y_i = \mathbb{I}\left(\sum_{j\in\mathscr{X}_i} U_j\right)$, since for simplicity we are considering the noiseless case. Further, in this paper, we restrict our attention to the adaptive case, where we choose test set $\mathscr{X}_i$ based on a knowledge of sets $\mathscr{X}_1,\dots,\mathscr{X}_{i-1}$ and outcomes $y_1,\dots,y_{i-1}$. The group testing problem requires us to infer $\mathbf{U}$ with high probability given a low number of tests T.

Since Dorfman's paper [**?**], there has been considerable work on the question of how to design the sets $\mathscr{X}_i$ in order to minimise the number of tests T required. In this context, we briefly mention so-called combinatorial designs (see [42, 71] for a summary, with [71] giving invaluable references to an extensive body of Russian work in the 1970s and 1980s). Such designs typically aim to ensure that set-theoretic properties known as disjunctness and separability occur. In contrast, for simplicity of analysis, as well as performance of optimal order, it is possible to consider random designs. Here sets $\mathscr{X}_i$ are chosen at random, either using constructions such as independent Bernoulli designs [2, 4, 101] or more sophisticated random designs based on LDPC codes [121].

Much previous work has focussed on the Combinatorial group testing problem, where there are a fixed number of defectives K, and the defectivity vector $\mathbf{U}$ is chosen uniformly among all binary vectors of weight K. In contrast, in this paper we study a Probabilistic group testing problem as formulated for example in the work of Li et al. [64], in that we suppose each item is defective independently with probability p_i, or equivalently take U_i to be independent Bernoulli(p_i).

This Probabilistic framework, including non-uniform priors, is natural for many applications of group testing. For example, see [3], the cognitive radio problem can be formulated in terms of a population of communication bands in frequency spectra with some (unknown) occupied bands you must not utilise. Here, the values of p_i may be chosen based on some database of past spectrum measurements or other prior information. Similarly, as in Dorfman's original work [**?**] or more recent research [104] involving screening for genetic conditions, values of p_i might summarise prior information based on a risk profile or family history.

8.1.3 Group testing capacity

It is possible to characterize performance tradeoffs in group testing from an information-theoretic point of view – see for example [2, 4, 5, 111]. These papers have focussed on group testing as a channel coding problem, with [4, 111] explicitly calculating the mutual information. The paper [5] defined the capacity of a Combinatorial group testing procedure, which characterizes the number of bits of information about the defective set which we can learn per test. We give a more general definition here, which covers both the Combinatorial and Probabilistic cases.

Definition 8.1.1. *Consider a sequence of group testing problems where the ith problem has defectivity vector $\mathbf{U}^{(i)}$, and consider algorithms which are given $T(i)$ tests. We refer to a constant C as the (weak) group testing capacity if for any $\epsilon > 0$:*

1. *any sequence of algorithms with*

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon, \tag{8.11}$$

has success probability $\mathbb{P}(suc)$ bounded away from 1,

2. *and there exists a sequence of algorithms with*

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C - \epsilon \tag{8.12}$$

with success probability $\mathbb{P}(\text{suc}) \to 1$.

Remark 8.1.2. *In the Combinatorial case of K defective items with all defective sets equally likely, $H(\mathbf{U}) = \log_2 \binom{N}{K}$, which is the term found in the denominator in [5, Eq. (1) and (2)]. In the Probabilistic case (as in [64]) we know $H(\mathbf{U}) = -\sum_{i=1}^{N} h(p_i)$ where $h(t) = -t\log_2 t - (1-t)\log_2(1-t)$ is the binary entropy function.*

Remark 8.1.3. *If for* $\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon$, *the success probability* $\mathbb{P}(\text{suc}) \to 0$ *we say that C is the strong group testing capacity, following standard terminology in information theory. Such a result is referred to as a strong converse.*

8.1.4 Main results

The principal contribution of [5, Theorem 1.2] was the following result:

Theorem 8.1.4 ([5])**.** *The strong capacity of the adaptive noiseless Combinatorial group testing problem is $C = 1$, in any regime such that $K/N \to 0$.*

This argument came in two parts. First, in [5, Theorem 3.1] the authors proved a new upper bound on success probability

$$\mathbb{P}(\text{suc}) \leq \frac{2^T}{\binom{N}{K}}, \tag{8.13}$$

which implied a strong converse ($C \leq 1$). This was complemented by showing that, in the Combinatorial case, an algorithm based on Hwang's Generalized Binary Splitting Algorithm (HGBSA) [58], [42] is essentially optimal in the required sense, showing that $C = 1$ is achievable.

It may be useful to characterize the Probabilistic group testing problem in terms of the effective sparsity $\mu^{(N)} := \sum_{i=1}^{N} p_i$. In particular, if the p_i are (close to) identical, we would expect performance similar to that in the Combinatorial case with $K = \mu^{(N)}$ defectives. As in [5], we focus on asymptotically sparse cases, where $\mu^{(N)}/N \to 0$ (in contrast, Wadayama [] considered a model where p_i are identical and fixed). The main result of the present paper is Theorem 8.3.9, stated and proved in Section 8.3.5 below, which implies the following Probabilistic group testing version of Theorem 8.1.4.

Corollary 8.1.5. *In the case where $p_i \equiv p$, the weak capacity of the adaptive noiseless Probabilistic group testing problem is $C = 1$, in any regime such that $\mu^{(N)}/N \to 0$ and $\mu^{(N)} \to \infty$.*

Again we prove our main result Theorem 8.3.9 using complementary bounds on both sides. First in Section 8.2.1 we recall a universal upper bound on success probability, Theorem 8.2.1, taken from [64], which implies a weak converse. In [64], Li et al. introduce the Laminar Algorithm for Probabilistic group testing. In Section 8.2.3 we propose a refined version of this Laminar Algorithm, based on

Hwang's HGBSA [58], which is analysed in Section 8.3.5, and shown to imply performance close to optimal in the sense of capacity.

8.2 Algorithms and existing results

8.2.1 Upper bounds on success probability

Firstly [64, Theorem 1] can be restated to give the following upper bound on success probability:

Theorem 8.2.1. *Any Probabilistic group testing algorithm using T tests with noiseless measurements has success probability satisfying*

$$\mathbb{P}(\text{suc}) \leq \frac{T}{H(\mathbf{U})}.$$

Rephrased in terms of Definition 8.1.1, this tells us that the weak capacity of noiseless Probabilistic group testing is ≤ 1. The logic is as follows; if the capacity were $1+2\epsilon$ for some $\epsilon > 0$, then there would exist a sequence of algorithms with $H(\mathbf{U}^{(i)})/T(i) \geq 1+\epsilon$ with success probability tending to 1. However, by Theorem 8.2.1, any such algorithms have $\mathbb{P}(\text{suc}) \leq 1/(1+\epsilon)$, meaning that we have established that a weak converse holds.

Remark 8.2.2. *It remains an open and interesting problem to prove an equivalent of* (8.13) *as in [5, Theorem 3.1]. That is we hope to find an upper bound on success probability in a form which implies a strong converse, and hence that the strong capacity of Probabilistic group testing is equal to 1.*

8.2.2 Binary search algorithms

The main contribution of this work is to describe and analyse algorithms that will find the defective items. In brief, we can think of Hwang's HGBSA algorithm as dividing the population $\mathscr{P}$ into search sets $\mathscr{S}$. First, all the items in a search set $\mathscr{S}$ are tested together, using a test set $\mathscr{X}_1 = \mathscr{S}$. If the result is negative ($y_1 = 0$), we can be certain that $\mathscr{S}$ contains no defectives. However, if the result is positive ($y_1 = 1$), $\mathscr{S}$ must contain at least one defective.

If $y_i = 1$, we can be guaranteed to find at least one defective, using the following binary search strategy. We split the set $\mathscr{S}$ in two, and test the 'left-hand' set, say $\mathscr{X}_2$. If $y_2 = 1$, then we know that $\mathscr{X}_2$ contains at least one defective. If $y_2 = 0$, then $\mathscr{X}_2$ contains no defective, so we can deduce that $\mathscr{S} \setminus \mathscr{X}_2$ contains at least one defective. By repeated use of this strategy, we are guaranteed to find a succession of nested sets which contain at least one defective, until $\mathscr{X}_i$ is of size 1, and we have isolated a single defective item.

However this strategy may not find every defective item in $\mathscr{S}$. To be specific, it is possible that at some stage both the left-hand and right-hand sets contain a defective. The Laminar Algorithm of [64] essentially deals with this by testing both sets. However, we believe that this is inefficient, since typically both sets will not contain a defective. Nonetheless, the Laminar Algorithm satisfies the following performance guarantees proved in [64, Theorem 2]:

A Set S of $|S| = n$ items, μ of which are actually defective in expectation, a probability vector $\mathbf{p}^{(n)}$ describing each item's independent probability of being defective, and a cutoff θ
ReturnsThe set of defective items Discard items with $p_i \leq \theta$
Sort the remaining items into B bins, collecting items together with $p_i \in \left[1/2C^r, 1/2C^{r-1}\right)$ in bin r.
Sort the items in each bin into sets s.t. the (normalised) probability of each set is less than $1/2$.
Test each set in turn

if The test is positive **then**Arrange the items in the set on a Shannon-Fano/Huffman Tree and search the set for all the defectives it contains
end if

Figure 8.2: Algorithm for the non-iid group testing problem

Theorem 8.2.3. *The expected number of tests required by the Laminar Algorithm [64] is* $\leq 2H(\mathbf{U}) + 2\mu$. *Under a technical condition (referred to as non-skewedness), the success probability can be bounded by* $\mathbb{P}(\text{suc}) \geq 1 - \epsilon$ *using* $T = (1+\delta)(2^{\Gamma + \log_2 3} + 2)H(\mathbf{U})$ *tests, where* Γ *is defined implicitly in terms of* ϵ, *and* $\delta \geq 2e - 1$.

Ignoring the Γ term, and assuming the non-skewedness condition holds, this implies that (using the methods of [64]) $T = 2e(3+2)H(\mathbf{U}) = 10eH(\mathbf{U})$ tests are required to guarantee convergence to 1 of the success probability. In our language, this implies a lower bound of $C \geq 1/(10e) = 0.0368$. Even ignoring the analysis of error probability, the fact that the expected number of tests is $\leq 2H(\mathbf{U}) + 2\mu$ suggests that we cannot hope to achieve $C > 1/2$ using the Laminar Algorithm.

8.2.3 Summary of our contribution

The main contribution of our paper is a refined version of the Laminar Algorithm, summarised above, and an analysis resulting in tighter error bounds as formulated in Proposition 8.3.7 (in terms of expected number of tests) and Theorem 8.3.9 (in terms of error probabilities). The key ideas are:

1. To partition the population $\mathscr{P}$ into search sets $\mathscr{S}$ containing items which have similar probabilities, expressed through the Bounded Ratio Condition 3. This is discussed in Section 8.3.1, and optimised in the proof of Proposition 8.3.7.

2. The way in which we deal with sets $\mathscr{S}$ which contain more than one defective, as discussed in Remark 8.3.2 below. Essentially we do not backtrack after each test by testing both left- and right-hand sets, but only backtrack after each defective is found.

3. To discard items which have probability below a certain threshold, since with high probability none of them will be defective. This is an idea introduced in [64] and discussed in Section 8.3.2, with a new bound given in Lemma 8.3.4.

4. Careful analysis in Section 8.3.4 of the properties of search sets $\mathscr{S}$ gives Proposition 8.3.7, which shows that the expected number of tests required can be expressed as $H(\mathbf{U})$ plus an error

term. In Section 8.3.5, we give an analysis of the error probability using Bernstein's inequality, Theorem 8.3.8, allowing us to prove Theorem 8.3.9.

8.2.4 Wider context: sparse inference problems

Recent work [1, 111] has shown that many arguments and bounds hold in a common framework of sparse inference which includes group testing and compressive sensing.

Digital communications, audio, images, and text are examples of data sources we can compress. We can do this, because these data sources are sparse: they have fewer degrees of freedom than the space they are defined upon. For example, images have a well known expansion in either the Fourier or Wavelet bases. The text of an English document will only be comprised of words from the English dictionary, and not all the possible strings from the space of strings made up from the characters $\{a,\ldots,z\}$.

Often, once a signal has been acquired it will be compressed. However, the compressive sensing paradigm introduced by [22, 38] shows that this isn't necessary. In those papers it was shown that a 'compressed' representation of a signal could be obtained from random linear projections of the signal and some other basis (for example White Gaussian Noise). The question remains, given this representation how do we recover the original signal? For real signals, a simple linear programme suffices. Much of the work in this area has been couched in terms of the sparsity of the signal and the various bases the signal can be represented in (see for example [22, 38]).

8.3 Analysis and new bounds

8.3.1 Searching a set of bounded ratio

Recall that we have a population $\mathscr{P}$ of items to test, each with associated probability of defectiveness p_i. The strategy of the proof is to partition $\mathscr{P}$ into search sets $\mathscr{S}_1,\ldots,\mathscr{S}_G$, each of which contains items which have comparable values of p_i.

Condition 3 (Bounded Ratio Condition)**.** *Given $C \geq 1$, say that a set $\mathscr{S}$ satisfies the Bounded Ratio Condition with constant C if*

$$\max_{i,j\in\mathscr{S}} \frac{p_j}{p_i} \leq C. \tag{8.14}$$

(For example clearly if $p_i \equiv p$, any set $\mathscr{S}$ satisfies the condition for any $C \geq 1$).

Lemma 8.3.1. *Consider a set $\mathscr{S}$ satisfying the Bounded Ratio Condition with constant C and write $P_{\mathscr{S}} = \sum_{j\in\mathscr{S}} p_j$. In a Shannon–Fano tree for the probability distribution $\overline{p}_i := p_i/P_{\mathscr{S}}$, each item has length $\ell_i^{(\mathscr{S})}$ bounded by*

$$\ell_i^{(\mathscr{S})} \leq \ell_{\max}^{(\mathscr{S})} := \frac{h(\mathscr{S})}{P_{\mathscr{S}}} + \log_2 C + \log_2 P_{\mathscr{S}} + 1, \tag{8.15}$$

where we write $h(\mathscr{S}) := -\sum_{j\in\mathscr{S}} p_j \log_2 p_j$.

Proof. Under the Bounded Ratio Condition, for any i and j, we know that by taking logs of (8.14)

$$-\log_2 p_i \le -\log_2 p_j + \log_2 C.$$

Multiplying by p_j and summing over all $j \in \mathscr{S}$, we obtain that

$$-P_{\mathscr{S}} \log_2 p_i \le h(\mathscr{S}) + P_{\mathscr{S}} \log_2 C. \tag{8.16}$$

Now, the Shannon–Fano length of the ith item is

$$\begin{aligned} \ell_i^{(\mathscr{S})} = \lceil -\log_2 \overline{p}_i \rceil &\le -\log_2 p_i + \log_2 P_{\mathscr{S}} + 1 \\ &\le \left(\frac{h(\mathscr{S})}{P_{\mathscr{S}}} + \log_2 C \right) + \log_2 P_{\mathscr{S}} + 1. \end{aligned} \tag{8.17}$$

and the result follows by (8.16). □

Next we describe our search strategy:

Remark 8.3.2. *Our version of the algorithm will find every defective in a set $\mathscr{S}$. We start as before by testing every item in $\mathscr{S}$ together. If this test is negative, we are done. Otherwise, if it is positive, we can perform binary search as above to find one defective item, say d_1. Now, test every item in $\mathscr{S} \setminus \{d_1\}$ together. If this test is negative, we are done, otherwise we repeat the search step on this smaller set, to find another defective item d_2, then we test $\mathscr{S} \setminus \{d_1, d_2\}$ and so on.*

We think of the algorithm as repeatedly searching a binary tree. Clearly, if the tree has depth bounded by ℓ, then the search will take $\le \ell$ tests to find one defective. In total, if the set contains U defectives, we need to repeat U rounds of searching, plus the final test to guarantee that the set contains no more defectives, so will use $\le \ell U + 1$ tests.

Lemma 8.3.3. *Consider a search set $\mathscr{S}$ satisfying the Bounded Ratio Condition and write $P_{\mathscr{S}} = \sum_{j \in \mathscr{S}} p_j$. If (independently) item i is defective with probability p_i, we can recover all defective items in the set using $T_{\mathscr{S}}$ tests, where $\mathbb{E} T_{\mathscr{S}} \le T_{\mathrm{bd}}(\mathscr{S})$ for*

$$T_{\mathrm{bd}}(\mathscr{S}) := h(\mathscr{S}) + P_{\mathscr{S}} \log_2 C + P_{\mathscr{S}} \log_2 P_{\mathscr{S}} + P_{\mathscr{S}} + 1. \tag{8.18}$$

Proof. Using the algorithm of Remark 8.3.2, laid out on the Shannon-Fano tree constructed in Lemma 8.3.1, we are guaranteed to find every defective. The number of tests to find one defective thus corresponds to the depth of the tree, which is bounded by $\ell_{\max}^{(\mathscr{S})}$ given in (8.15).

Recall that we write U_i for the indicator of the event that the ith item is defective, $U_{\mathscr{S}} = \sum_{i \in \mathscr{S}} U_i$ and $l_i^{(\mathscr{S})}$ for the length of the word in the Shannon Fano tree. As discussed in Remark 8.3.2 this search

procedure will take

$$\begin{aligned} T_{\mathscr{S}} &= 1 + \sum_{i \in \mathscr{S}} U_i \ell_i^{(\mathscr{S})} \\ &= \sum_{i \in \mathscr{S}} p_i \ell_i^{(\mathscr{S})} + 1 + \sum_{i \in S} \ell_i^{(\mathscr{S})} (U_i - p_i) \\ &\leq \sum_{i \in \mathscr{S}} p_i \ell_{\max}^{(\mathscr{S})} + 1 + \sum_{i \in \mathscr{S}} V_i^{(\mathscr{S})} \\ &= P_{\mathscr{S}} \ell_{\max}^{(\mathscr{S})} + 1 + \sum_{i \in S} V_i^{(\mathscr{S})} \\ &\leq T_{\mathrm{bd}}(\mathscr{S}) + \sum_{i \in S} V_i^{(\mathscr{S})} \quad \text{tests.} \end{aligned} \tag{8.19}$$

Here we write $V_i^{(\mathscr{S})} = \ell_i^{(\mathscr{S})}(U_i - p_i)$, which has expectation zero, and (8.19) follows using the expression for $\ell_{\max}^{(\mathscr{S})}$ given in Lemma 8.3.1. □

8.3.2 Discarding low probability items

As in [64], we use a probability threshold θ, and write $\mathscr{P}^*$ for the population having removed items with $p_i \leq \theta$. If an item lies in $\mathscr{P} \setminus \mathscr{P}^*$ we do not test it, and simply mark it as non-defective. This truncation operation gives an error if and only if some item in $\mathscr{P} \setminus \mathscr{P}^*$ is defective. By the union bound, this truncation operation contributes a total of $\mathbb{P}(\mathscr{P} \setminus \mathscr{P}^* \text{ contains a defective}) \leq \rho := \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta)$ to the error probability.

Lemma 8.3.4. *Choosing $\theta(P_e)$ such that*

$$-\log_2 \theta(P_e) = \min\left(\log_2\left(\frac{2n}{P_e}\right), \frac{2H(\mathbf{U})}{P_e} \right) \tag{8.20}$$

ensures that

$$\mathbb{P}(\,\mathscr{P} \setminus \mathscr{P}^* \textit{ contains a defective}) \leq P_e/2. \tag{8.21}$$

Proof. The approach of [64] is essentially to bound $\mathbb{I}(p_i \leq \theta) \leq \theta / p_i$ so that $\rho = \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta) \leq \sum_{i=1}^n p_i (\theta / p_i) = n\theta$. Hence, choosing a threshold of $\theta = P_e/(2n)$ guarantees the required bound on ρ.

We combine this with another bound, constructed using a different function: $\mathbb{I}(p_i \leq \theta) \leq (-\log_2 p_i)/(-\log_2 \theta)$, so that

$$\rho = \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta) \leq \sum_{i=1}^n p_i \left(\frac{-\log_2 p_i}{-\log_2 \theta} \right) \leq \frac{H(\mathbf{U})}{-\log_2 \theta},$$

so we deduce the result. □

8.3.3 Searching the entire set

Having discarded items with p_i below this probability threshold θ and given bounding ratio C, we create a series of bins. We collect together items with probabilities $p \in [1/2, 1]$ in bin 0, $p \in [1/(2C), 1/2)$ in bin 1, items with probabilities $p \in [1/(2C^2), 1/(2C))$ in bin 2, ..., and items with probabilities $p \in [1/(2C^B), 1/(2C^{B-1}))$ in bin B.

The probability threshold θ means that there will be a finite number of such bins, with the index B of the last bin defined by the fact that $1/(2C^B) \leq \theta < 1/(2C^{B-1})$, meaning that $(B-1)\log_2 C < -\log_2(2\theta)$, so

$$B \leq \frac{-\log_2(2\theta)}{\log_2 C} + 1. \tag{8.22}$$

We split the items in each bin into search sets $\mathscr{S}_i$, motivated by the following definition:

Definition 8.3.5. *A set of items $\mathscr{S}$ is said to be full if $P_{\mathscr{S}} = \sum_{i\in\mathscr{S}} p_i \geq \frac{1}{2}$.*

Our splitting procedure is as follows: we create a list of possible sets $\mathscr{S}_1, \mathscr{S}_2, \ldots$. For i increasing from 0 to B, we place items from bin i into sets $\mathscr{S}_{b_i+1}, \ldots, \mathscr{S}_{b_{i+1}}$, for some b_i, where $b_0 = 0$. Taking the items from bin i, while $\mathscr{S}_{b_i+1}$ is not full (has total probability $< \frac{1}{2}$) we will place items into it. Once enough items have been added to fill $\mathscr{S}_{b_i+1}$, we will proceed in the same way to fill $\mathscr{S}_{b_i+2}$, and so on until all the items in bin i have been assigned to sets $\mathscr{S}_{b_i+1}, \ldots, \mathscr{S}_{b_{i+1}}$, where $\mathscr{S}_{b_{i+1}}$ may remain not full.

Proposition 8.3.6. *This splitting procedure will divide $\mathscr{P}^*$ into search sets $\mathscr{S}_1, \ldots, \mathscr{S}_G$, where the total number of sets is*

$$G \leq 2\mu + B \leq 2\mu + \left(\frac{-\log_2(2\theta)}{\log_2 C} + 1\right).$$

Each set $\mathscr{S}_j$ satisfies the Bounded Ratio Condition and has total probability $P_j := P_{\mathscr{S}_j} \leq 1$.

Proof. First, note that the items from bin 0 each lie in a set $\mathscr{S}$ on their own. These sets will be full, trivially satisfy the Bounded Ratio Condition 3 with constant C, and have probability satisfying $P_j \leq 1$. For each of bins $1, \ldots, B$:

1. For each bin i, it is possible that the last set $\mathscr{S}_{b_i+1}$ will not be full, but every other set corresponding to that bin will be full. Hence, there are no more than B sets which are not full.

2. For each resulting set $\mathscr{S}_j$, the total probability $P_j \leq 1$ (since just before we add the final item, $\mathscr{S}_j$ is not full, so at that stage has total probability $\leq 1/2$, and each element in bins $1, \ldots, B$ has probability $\leq 1/2$).

3. Since each set $\mathscr{S}_j$ contains items taken from the same bin, it will satisfy the Bounded Ratio Condition with constant C.

Note that the number of full sets is $\leq 2\mu$, since

$$\mu = \sum_{i\in\mathscr{P}} p_i \geq \sum_{i\in\mathscr{P}^*} p_i = \sum_{j=1}^{G} P_j \geq \sum_{j:\,\mathscr{S}_j \text{ full}} P_j \geq \left|\mathscr{S}_j \text{ full}\right| \frac{1}{2}. \tag{8.23}$$

Since, as discussed in point 1) above, the total number of sets is bounded by the number of full sets plus B, the result follows using Equation (8.22). □

8.3.4 Bounding the expected number of tests

We allow the algorithm to work until all defectives in $\mathscr{P}^*$ are found, and write T for the (random) number of tests this takes.

Proposition 8.3.7. *Given a population $\mathscr{P}$ where (independently) item i is defective with probability p_i, we recover all defective items in $\mathscr{P}^*$ in T tests with $\mathbb{E}T \leq T_{\mathrm{bd}}$, where*

$$T_{\mathrm{bd}} := \big(H(\mathbf{U}) + 3\mu + 1\big) + 2\sqrt{\mu\big(-\log_2(2\theta)\big)}. \tag{8.24}$$

Proof. Given a value of C, Proposition 8.3.6 shows that our splitting procedure divides $\mathscr{P}^*$ into G sets $\mathscr{S}_1, \ldots, \mathscr{S}_G$, such that each set $\mathscr{S}_j$ satisfies the Bounded Ratio Condition with constant C and has total probability $P_j \leq 1$. Using the notation of Lemma 8.3.3, $T = \sum_{j=1}^{G} T_{\mathscr{S}_j}$, where $\mathbb{E}T_{\mathscr{S}_j} \leq T_{\mathrm{bd}}(\mathscr{S}_j)$.

Adding this bound over the different sets, since $P_j \leq 1$ means that $P_j \log_2 P_j \leq 0$, we obtain

$$\begin{aligned}
&\sum_{j=1}^{G} T_{\mathrm{bd}}(\mathscr{S}_j) \\
&\quad\leq \sum_{j=1}^{G} \big(h(\mathscr{S}_j) + P_j(\log_2 C + 1) + 1\big) \\
&\quad= \sum_{j\in\mathscr{P}^*} -p_j \log_2 p_j + \mu(\log_2 C + 1) + G \\
&\quad\leq \sum_{j\in\mathscr{P}^*} h(p_j) + 3\mu + 1 + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C\right) \\
&\quad\leq \big(H(\mathbf{U}) + 3\mu + 1\big) + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C\right).
\end{aligned} \tag{8.25}$$

This follows by the bound on G in Proposition 8.3.6, as well as the fact that $0 \leq p_j \leq 1$ means that for any i, $-p_j \log_2 p_j = (1-p_j)\log_2(1-p_j) + h(p_j) \leq h(p_j)$.

Finally, we choose $C > 1$ to optimize the second bracketed term in Equation (8.25). Differentiation shows that the optimal C satisfies $\log_2 C = \sqrt{-\log_2(2\theta)/\mu}$, meaning that the bracketed term

$$\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C = 2\sqrt{\mu\big(-\log_2(2\theta)\big)},$$

and the result follows. □

8.3.5 Controlling the error probabilities

Although Section 8.3.4 proves that $\mathbb{E}T \leq T_{\mathrm{bd}}$, to bound the capacity, we need to prove that with high probability T is not significantly larger than T_{bd}. This can be done using Bernstein's inequality (see for example Theorem 2.8 of [91]):

Theorem 8.3.8 (Bernstein)**.** *For zero-mean random variables V_i which are uniformly bounded by $|V_i| \leq M$, if we write $L := \sum_{j=1}^{n} \mathbb{E}V_j^2$ then*

$$\mathbb{P}\left(\sum_{j=1}^{n} V_j \geq t\right) \leq \exp\left(-\frac{t^2}{4L}\right), \text{ for any } 0 \leq t \leq \tfrac{L}{M}. \tag{8.26}$$

We deduce the following result:

Theorem 8.3.9. *Write* $L = \sum_{j\in\mathscr{P}^*} l_j^2 p_j(1-p_j)$, $M = -\log_2\theta + 1$ *and* $\psi = (L/(4M^2))^{-1/3}$. *Define*

$$T_{\text{nec}} = T_{\text{bd}} + \psi H(\mathbf{U}), \tag{8.27}$$

where T_{bd} *is given in* (8.24).

1. *If we terminate our group testing algorithm after* T_{nec} *tests, the success probability*

$$\mathbb{P}(\text{suc}) \geq 1 - \frac{1}{2}\sqrt{\frac{\mu}{H(\mathbf{U})}} - \exp\left(-\left(\frac{L}{4M^2}\right)^{1/3}\right). \tag{8.28}$$

2. *Hence in any regime where* $\mu\to\infty$ *with* $\mu/H(\mathbf{U})\to 0$ *and* $L/M^2\to\infty$, *our group testing algorithm has (a)* $\liminf H(\mathbf{U})/T_{\text{nec}} \geq 1/(1+\epsilon)$ *for any* ϵ *and (b)* $\mathbb{P}(\text{suc})\to 1$, *so the capacity* $C = 1$.

Proof. We first prove the success probability bound (8.28). Recall that our algorithm searches the reduced population set $\mathscr{P}^*$ for defectives. This gives two error events – either there are defective items in the set $\mathscr{P}\setminus\mathscr{P}^*$, or the algorithm does not find all the defectives in $\mathscr{P}^*$ using T_{nec} tests. We consider them separately, and control the probability of either happening using the union bound.

Writing $H = H(\mathbf{U})$ for brevity and choosing $P_e = \sqrt{\mu/H}$ ensures that (by Lemma 8.3.4) the first event has probability $\leq P_e/2$, contributing $\frac{1}{2}\sqrt{\mu/H(\mathbf{U})}$ to (8.28).

Our analysis of the second error event is based on the random term from Equation (8.19), which we previously averaged over but now wish to bound. There will be an error if $T_{\text{nec}} \leq T$, or (rearranging) if

$$\psi H \leq T - T_{\text{bd}} \leq \sum_{j=1}^{G}\left(T_{\mathscr{S}_j} - T_{\text{bd}}(\mathscr{S}_j)\right) = \sum_{i\in\mathscr{P}^*} V_i.$$

For brevity, for $i\in\mathscr{S}$, we write $V_i = V_i^{(\mathscr{S})} = \ell_i^{(\mathscr{S})}(U_i - p_i)$ and $\ell_i = \ell_i^{(\mathscr{S})}$, where V_i has expectation zero.

We have discarded elements with probability below θ, as given by (8.20), and by design all the sets $\mathscr{S}$ have total probability $P_{\mathscr{S}} \leq 1$. Using (8.17) we know that the V_i are bounded by

$$|V_i| \leq \ell_i \leq -\log_2 p_i + \log_2 P_{\mathscr{S}} + 1 \leq -\log_2\theta + 1. \tag{8.29}$$

Hence, the conditions of Bernstein's inequality, Theorem 8.3.8, are satisfied. Observe that since all $l_j \leq M$,

$$\frac{L}{HM} = \frac{\sum_{j\in\mathscr{P}^*} l_j^2 p_j(1-p_j)}{HM} \leq \frac{\sum_{j\in\mathscr{P}^*} l_j p_j}{H} \leq 1.$$

Hence Theorem 8.3.8 gives that

$$\begin{aligned}\mathbb{P}\left(\sum_{j\in\mathscr{P}^*} V_j \geq \psi H\right) &\leq \mathbb{P}\left(\sum_{j\in\mathscr{P}^*} V_j \geq \psi L/M\right)\\ &\leq \exp\left(-\frac{L\psi^2}{4M^2}\right)\\ &= \exp\left(-\left(\frac{L}{4M^2}\right)^{1/3}\right).\end{aligned}$$

Using the union bound, the probability bound (8.28) follows.

We next consider the capacity bound of 2). Since $-\log_2\theta \leq 2H/P_e$, using (8.24) and (8.27)

$$\begin{aligned}\frac{T_{\text{nec}}}{H} &= \frac{T_{\text{bd}}}{H}+\psi \\ &= 1+3\frac{\mu}{H}+\frac{1}{H}+2\sqrt{\frac{\mu}{HP_e}}+\psi \\ &= 1+3\frac{\mu}{H}+\frac{1}{H}+2\left(\frac{\mu}{H}\right)^{1/4}+\psi, \end{aligned} \tag{8.30}$$

which in our regime of interest is $\leq 1+\epsilon$ in the limit. □

Proof of Corollary 8.1.5. In the case where all p are identical, $\mu = Np$, $H = Np(-\log p)$, so $\mu/H = 1/(-\log p) \to 0$. Similarly, $L = Np(-\log_2 p)^2$ and $M = (-\log_2 p)$ so that $L/M^2 = Np \to \infty$ as required. □

8.4 Results

The performance of the Algorithm 1 (in terms of the sample complexity) was analysed by simulating 500 items, with a mean number of defectives equal to 8. I.e. $N = 500$ and $\mu^{(N)} = 8$.

The probability distribution **p** was generated by a Dirichlet distribution with parameter α. This produces output which can be made more or less uniform, as opposed to simply choosing a set of random numbers and normalise by the sum. Consider the case of two random numbers, (x, y), distributed uniformly on the square $[0,1]^2$. Normalising by the sum $(x+y)$ projects the point (x, y) onto the line $x+y=1$ and so favours points closer to $(0.5, 0.5)$ than the endpoints. The Dirichlet distribution avoids this by generating points directly on the simplex.

We then chose values of the cutoff parameter θ from 0.0001 to 0.01, and for each θ_i simulated the algorithm 1000 times. We plot the empirical distribution of tests, varying theta as well as the uniformity/concentration of the probability distribution (via the parameter α of the Dirichlet distribution). We also plot, the theoretical lower and upper bounds on the number of Tests required for successful recovery alongside the empirical number tests (all as a function of θ).

Note that the Upper bound is not optimal and there still is some room for improvement. Note also that the lower bound degrades with θ_i. The lower bound (T_{LCHJ}) was generated according to theorem (8.2.1).

Figures (8.4) and (8.5) show that the performance is relatively insensitive to the cut-off θ, and more sensitive to the uniformity (or otherwise) of the probability distribution **p**. Heuristically, this is for because distributions which are highly concentrated on a few items algorithms can make substantial savings on the testing budget by testing those highly likely items first (which is captured in the bin structure of the above algorithm).

The insensitivity to the cutoff θ is due to items below θ being overwhelmingly unlikely to be defective - which for small θ means that few items (relative to the size of the problem) get discarded.

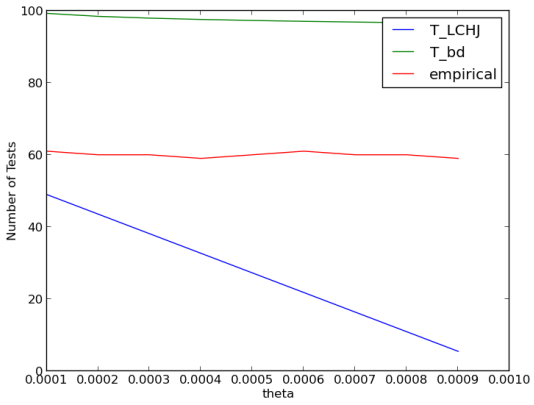

Figure 8.3: Theoretical lower and upper bounds and empirical Test frequencies as functions of θ

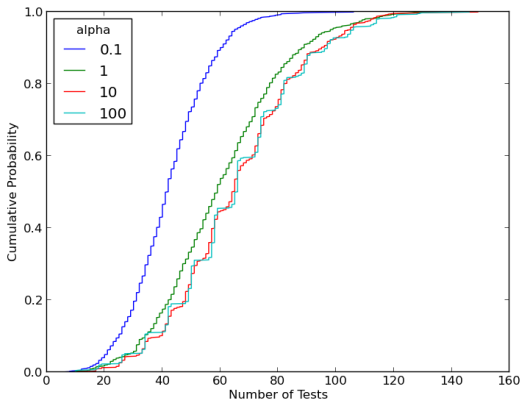

Figure 8.4: Cumulative distribution curves of the modified Hwang algorithm with fixed $\theta = 0.0001$ and α varying

8.5 Discussion

We have introduced and analysed an algorithm for Probabilistic group testing which uses 'just over' $H(\mathbf{U})$ tests to recover all the defectives with high probability. Combined with a weak converse taken from [64], this allows us to deduce that the weak capacity of Probabilistic group testing is $C = 1$. These results are illustrated by simulation.

For simplicity, this work has concentrated on establishing a bound T_{bd} in (8.24) which has leading term $H(\mathbf{U})$, and not on tightening bounds on the coefficient of μ in (8.24). For completeness, we mention that this coefficient can be reduced from 3, under a simple further condition:

Remark 8.5.1. *For some $c \leq 1/2$, we assume that all the $p_i \leq c$, and we alter the definition of 'fullness' to assume that a set is full if it has total probability less than α. In this case, the term $P_{\mathscr{S}} \log_2 P_{\mathscr{S}}$ in* (8.18) *becomes $P_{\mathscr{S}} \log_2(\alpha + c)$, the bound in* (8.23) *becomes μ/α, and since $\big((1-p)\log_2(1-p)\big)/p$ is*

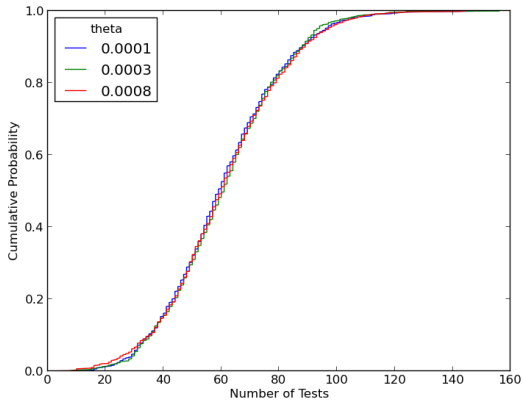

Figure 8.5: Cumulative distribution curves for fixed $\alpha = 1$ and varying θ

decreasing in p, we can add a term $(1-c)\log_2(1-c)$ *to* (8.25). *Overall, the coefficient of* μ *becomes* $f(a,c) := \log_2(\alpha+c)+1+1/\alpha+(1-c)\log_2(1-c)$, *which we can optimize over* α. *For example, if* $c = 1/4$, *taking* $\alpha = 0.88824$, *we obtain* $f(a,c) = 2.00135$.

It remains of interest to tighten the upper bound of Theorem 8.2.1, in order prove a strong converse, and hence confirm that the strong capacity is also equal to 1.

In future work, we hope to explore more realistic models of defectivity, such as those where the defectivity of U_i are not necessarily independent, for example by imposing a Markov neighbourhood structure.

BIBLIOGRAPHY

[1] Cem Aksoylar, George Atia, and Venkatesh Saligrama. Sparse signal processing with linear and non-linear observations: A unified shannon theoretic approach. In *Information Theory Workshop (ITW), 2013 IEEE*, pages 1–5. IEEE, 2013.

[2] Matthew Aldridge, Leonardo Baldassini, and Oliver Johnson. Group testing algorithms: bounds and simulations. *Information Theory, IEEE Transactions on*, 60(6):3671–3687, 2014.

[3] George Atia, Shuchin Aeron, Erhan Ermis, and Venkatesh Saligrama. On throughput maximization and interference avoidance in cognitive radios. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 963–967. IEEE, 2008.

[4] George K Atia and Venkatesh Saligrama. Boolean compressed sensing and noisy group testing. *Information Theory, IEEE Transactions on*, 58(3):1880–1901, 2012.

[5] Leonardo Baldassini, Oliver Johnson, and Matthew Aldridge. The capacity of adaptive group testing. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2676–2680. IEEE, 2013.

[6] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[7] Dror Baron. Bayesian compressive sensing via belief propagation. *Signal Processing, IEEE . . .* , 58(1):269–280, 2010.

[8] G B Bazerque, J.A Giannakis. Distributed spectrum sensing for cognitive radios by exploiting sparsity. *Proc. of 42nd Asilomar Conf. on Signals, Systems, and Computers*, 2008.

[9] Deepa Bhargavi and Chandra R Murthy. Performance comparison of energy, matched-filter and cyclostationarity-based spectrum sensing. In *Signal Processing Advances in Wireless Communications (SPAWC), 2010 IEEE Eleventh International Workshop on*, pages 1–5. IEEE, 2010.

[10] Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

[11] Thomas Blumensath and Mike E Davies. On the difference between orthogonal matching pursuit and orthogonal least squares. 2007.

[12] J Bodart, S Gishkori, J Verlant-Chenet, L Lampe, and F Horlin. Multiband spectrum sensing for cognitive radios based on distributed compressed measurements. In *2015 IEEE International Conference on Communications (ICC)*, pages 7515–7520. IEEE, 2015.

[13] Stephen Boyd. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning*, (1):1–122.

[14] Martin Braun, Jens P Elsner, and Friedrich Jondral. Signal detection for cognitive radios with smashed filtering. 2009.

[15] Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

[16] Hilton Bristow and Simon Lucey. Optimization Methods for Convolutional Sparse Coding. June 2014.

[17] Paul Burbidge, Eddie Levine. Cognitive radio technology a study for ofcom – summary report. Technical report, 2007.

[18] Danijela Čabrić and Robert W Brodersen. Physical layer design issues unique to cognitive radio systems. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 2, pages 759–763. IEEE, 2005.

[19] Danijela Cabric, Shridhar Mubaraq Mishra, and Robert W Brodersen. Implementation issues in spectrum sensing for cognitive radios. In *Signals, systems and computers, 2004. Conference record of the thirty-eighth Asilomar conference on*, volume 1, pages 772–776. IEEE, 2004.

[20] Emmanuel Candès and Terence Tao. Decoding by Linear Programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[21] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007.

[22] Emmanuel J Candes, Justin Romberg, and Terence Tao. Robust Uncertainty Principles : Exact Signal Frequency Information. 52(2):489–509, 2006.

[23] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[24] C Carathéodory. Über den variabilitatsbereich der koeffizienten von potenzreflien, die gegebene werte nicht annehmen. *Math. Ann.*, 64:95=115, 1907.

[25] Chun Lam Chan, Pak Hou Che, Sidharth Jaggi, and Venkatesh Saligrama. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1832–1839. Ieee, September 2011.

[26] Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

[27] Scott Shaobing Chen, David L. Donoho, and Michael a. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, January 1998.

[28] Scott Shaobing Chen, David L. Donoho, and Michael a. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, January 1998.

[29] Daiki Cho and Shusuke Narieda. A weighted diversity combining technique for cyclostationarity detection based spectrum sensing in cognitive radio networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, pages 1–6. IEEE, 2015.

[30] Jon F Claerbout and Francis Muir. Robust modeling with erratic data. *Geophysics*, 38(5):826–844, 1973.

[31] Shane F Cotter, Bhaskar D Rao, Kjersti Engan, and Kenneth Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, 53(7):2477–2488, 2005.

[32] Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *Information Theory, IEEE Transactions on*, 55(5):2230–2249, 2009.

[33] M. a. Davenport, S. R. Schnelle, J. P. Slavinsky, R. G. Baraniuk, M. B. Wakin, and P. T. Boufounos. A wideband compressive radio receiver. *2010 - Milcom 2010 Military Communications Conference*, pages 1193–1198, October 2010.

[34] Mark Davenport, Petros T Boufounos, Michael B Wakin, Richard G Baraniuk, et al. Signal processing with compressive measurements. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):445–460, 2010.

[35] Mark a. Davenport, Marco F. Duarte, Michael B. Wakin, Jason N. Laska, Dharmpal Takhar, Kevin F. Kelly, and Richard G. Baraniuk. <title>The smashed filter for compressive classification and target recognition</title>. pages 64980H–64980H–12, February 2007.

[36] Mark A Davenport, Jason N Laska, John R Treichler, and Richard G Baraniuk. The pros and cons of compressive sensing for wideband signal acquisition: Noise folding versus dynamic range. *IEEE Transactions on Signal Processing*, 60(9):4628–4642, 2012.

[37] Baron Gaspard Riche de Prony. Essai éxperimental et analytique: sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l'alkool, à différentes températures. *Journal de l'école Polytechnique*, 1(22):24–76, 1795.

[38] David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[39] David L Donoho, Juan M Santos, and John M Pauly. Compressed Sensing MRI. (March 2008):72–82.

[40] Robert Dorfman. The Detection of Defective Members of Large Populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.

[41] Jim Douglas and Henry H Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2):421–439, 1956.

[42] Ding-Zhu Du and Frank K Hwang. *Combinatorial group testing and its applications*, volume 12. World Scientific, 1999.

[43] M.F. Duarte, S. Sarvotham, D. Baron, M.B. Wakin, and R.G. Baraniuk. Distributed Compressed Sensing of Jointly Sparse Signals. *Conference Record of the Thirty-Ninth Asilomar Conference onSignals, Systems and Computers, 2005.*, pages 1537–1541.

[44] Jonathan Eckstein and Dimitri P Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[45] Michael Elad. Sparse and Redundant Representations. 2010.

[46] Mário AT Figueiredo and Robert D Nowak. An em algorithm for wavelet-based image restoration. *Image Processing, IEEE Transactions on*, 12(8):906–916, 2003.

[47] Euhanna Ghadimi, André Teixeira, Iman Shames, and Mikael Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. *Automatic Control, IEEE Transactions on*, 60(3):644–658, 2015.

[48] Mohamed Ghozzi, Francois Marx, Mischa Dohler, and Jacques Palicot. Cyclostatilonarilty-Based Test for Detection of Vacant Frequency Bands, 2006.

[49] Georgios B Giannakis and Zhi Tian. Compressed Sensing for Wideband Cognitive Radios. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, pages IV–1357–IV–1360, 2007.

[50] Arthur S Goldberger. Stepwise least squares: residual analysis and specification error. *Journal of the American Statistical Association*, 56(296):998–1000, 1961.

[51] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

[52] Karama Hamdi, Xiang Nian Zeng, Ali Ghrayeb, and Khaled Letaief. Impact of noise power uncertainty on cooperative spectrum sensing in cognitive radio systems. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[53] Andrew Harms, Waheed U Bajwa, and Robert Calderbank. A constrained random demodulator for sub-nyquist sampling. *IEEE Transactions on Signal Processing*, 61(3):707–723, 2013.

[54] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. *The elements of statistical learning: data mining, inference and prediction*, volume 27. Springer, 2005.

[55] Veria Havary-Nassab, Sadiq Hassan, and Shahrokh Valaee. Compressive detection for wide-band spectrum sensing. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3094–3097. IEEE, 2010.

[56] Juan Heredia Juesas, Gregory Allan, Ali Molaei, Luis Tirado, William Blackwell, Martinez Lorenzo, and A Jose. Consensus-based imaging using admm for a compressive reflector antenna. In *Antennas and Propagation & USNC/URSI National Radio Science Meeting, 2015 IEEE International Symposium on*, pages 1304–1305. IEEE, 2015.

[57] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[58] FK Hwang. A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67(339):605–608, 1972.

[59] Shihao Ji, Ya Xue, and Lawrence Carin. Bayesian Compressive Sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356, June 2008.

[60] Kyouwoong Kim, Ihsan A Akbar, Kyung K Bae, Jung-Sun Um, Chad M Spooner, and Jeffrey H Reed. Cyclostationary approaches to signal detection and classification in cognitive radio. In *New frontiers in dynamic spectrum access networks, 2007. DySPAN 2007. 2nd IEEE international symposium on*, pages 212–215. IEEE, 2007.

[61] Sami Kirolos, Jason Laska, Michael Wakin, Marco Duarte, Dror Baron, Tamer Ragheb, Yehia Massoud, and Richard Baraniuk. Analog-to-information conversion via random demodulation. In *2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, pages 71–74. IEEE, 2006.

[62] Oliver Leveque. Random matrices and communication systems an unexpected journey: There and back again.

[63] Michael A Lexa, Mike E Davies, John S Thompson, and Janosch Nikolic. Compressive power spectral density estimation. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 3884–3887. IEEE, 2011.

[64] Tongxin Li, Chun Lam Chan, Wenhao Huang, Tarik Kaced, and Sidharth Jaggi. Group testing with prior statistics. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 2346–2350. IEEE, 2014.

[65] Qing Ling, Wei Shi, Gang Wu, and Alejandro Ribeiro. Dlm: Decentralized linearized alternating direction method of multipliers. *Signal Processing, IEEE Transactions on*, 63(15):4051–4064, 2015.

[66] Jarmo Lundén, Visa Koivunen, Anu Huttunen, and H Vincent Poor. Spectrum sensing in cognitive radios based on multiple cyclic frequencies. In *Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on*, pages 37–43. IEEE, 2007.

[67] Jun Ma, Guodong Zhao, and Ye Li. Soft combination and detection for cooperative spectrum sensing in cognitive radio networks. *IEEE Transactions on Wireless Communications*, 7(11):4502–4507, 2008.

[68] Yanting Ma, Junan Zhu, and Dror Baron. Compressed Sensing via Universal Denoising and Approximate Message Passing. July 2014.

[69] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[70] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[71] Mikhail Malyutov. Search for sparse active inputs: a review. In *Information Theory, Combinatorics, and Search Theory*, pages 609–647. Springer-Verlag, 2013.

[72] Yehia Massoud, Sami Smaili, and Vikas Singal. Efficient realization of random demodulator-based analog to information converters. In *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 133–136. IEEE, 2011.

[73] Christopher A Metzler, Arian Maleki, and Richard G Baraniuk. From denoising to compressed sensing. *arXiv preprint arXiv:1406.4175*, 2014.

[74] Moshe Mishali and YC Eldar. From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals. *Selected Topics in Signal Processing, ...*, 4(2):375–391, 2010.

[75] Moshe Mishali and Yonina C Eldar. From theory to practice: Sub-nyquist sampling of sparse wideband analog signals. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):375–391, 2010.

[76] Hosein Mohimani, Massoud Babaie-Zadeh, Irina Gorodnitsky, and Christian Jutten. Sparse recovery using smoothed l0 (sl0): Convergence analysis. *arXiv preprint arXiv:1001.5073*, 2010.

[77] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[78] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Puschel. D-admm: A communication-efficient distributed algorithm for separable optimization. *Signal Processing, IEEE Transactions on*, 61(10):2718–2723.

[79] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

[80] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[81] Robert Nishihara, Laurent Lessard, Benjamin Recht, Andrew Packard, and Michael I Jordan. A general analysis of the convergence of admm. *arXiv preprint arXiv:1502.02009*, 2015.

[82] Harry Nyquist. Certain topics in telegraph transmission theory. *Proceedings of the IEEE*, 90(2):280–305, 2002.

[83] Brendan O'Donoghue, Giorgos Stathopoulos, and Stephen Boyd. A splitting method for optimal control. *Control Systems Technology, IEEE Transactions on*, 21(6):2432–2442, 2013.

[84] Jan Oksanen, Jarmo Lundén, and Visa Koivunen. Characterization of spatial diversity in cooperative spectrum sensing. In *Communications, Control and Signal Processing (ISCCSP), 2010 4th International Symposium on*, pages 1–5. IEEE, 2010.

[85] Marc P Olivieri, Greg Barnett, Alex Lackpour, Albert Davis, and Phuong Ngo. A scalable dynamic spectrum allocation system with interference mitigation for teams of spectrally agile software defined radios. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 170–179. IEEE, 2005.

[86] Mark S Oude Alink, André BJ Kokkeler, Eric AM Klumperink, Gerard JM Smit, and Bram Nauta. Lowering the snr wall for energy detection using cross-correlation. *Vehicular Technology, IEEE Transactions on*, 60(8):3748–3757, 2011.

[87] Christian Schou Oxvig, Patrick Steffen Pedersen, Thomas Arildsen, and Torben Larsen. Improving smoothed l0 norm in compressive sensing using adaptive parameter selection. *arXiv preprint arXiv:1210.4277*, 2012.

[88] Neal Parikh and Stephen P Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[89] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[90] Kaare Brandt Petersen et al. The matrix cookbook.

[91] V. V. Petrov. *Limit Theorems of Probability Theory*. Oxford University Press, Oxford, U.K., 1995.

[92] Yvan Lamelas Polo, Ying Wang, Ashish Pandharipande, and Geert Leus. Compressive wideband spectrum sensing. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 2337–2340. IEEE, 2009.

[93] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[94] Daniel Romero, Dyonisius Dony Ariananda, Zhi Tian, and Geert Leus. Compressive covariance sensing: Structure-based compressive sensing beyond sparsity. *IEEE Signal Processing Magazine*, 33(1):78–93, 2016.

[95] Anant Sahai, Niels Hoven, and Rahul Tandra. Some fundamental limits on cognitive radio. In *in Forty-second Allerton Conference on Communication, Control, and Computing*, 2004.

[96] F. Santosa and W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[97] Alex Sawatzky, Qi Xu, Carsten O Schirra, and Mark A Anastasio. Proximal admm for multichannel image reconstruction in spectral x-ray ct. *Medical Imaging, IEEE Transactions on*, 33(8):1657–1668, 2014.

[98] Ioannis D Schizas, Georgios B Giannakis, Stergios I Roumeliotis, and Alejandro Ribeiro. Consensus in ad hoc wsns with noisy links—part ii: Distributed estimation and smoothing of random signals. *IEEE Transactions on Signal Processing*, 56(4):1650–1666, 2008.

[99] Ioannis D Schizas, Alejandro Ribeiro, and Georgios B Giannakis. Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56(1):350–364, 2008.

[100] Dino Sejdinovic and Oliver Johnson. Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction. *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 998–1003, September 2010.

[101] Dino Sejdinovic and Oliver Johnson. Note on noisy group testing: asymptotic bounds and belief propagation reconstruction. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 998–1003. IEEE, 2010.

[102] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[103] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[104] Noam Shental, Amnon Amir, and Or Zuk. Identification of rare alleles and their carriers using compressed se (que) nsing. *Nucleic acids research*, page gkq675, 2010.

[105] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the admm in decentralized consensus optimization. *Signal Processing, IEEE Transactions on*, 62(7):1750–1761.

[106] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. pages 1–25, July 2013.

[107] J. P. Slavinsky, Jason N. Laska, Mark a. Davenport, and Richard G. Baraniuk. The compressive multiplexer for multi-channel compressive sensing. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (3):3980–3983, May 2011.

[108] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.

[109] Dennis Sundman, Saikat Chatterjee, and Mikael Skoglund. On the use of compressive sampling for wide-band spectrum sensing. In *The 10th IEEE International Symposium on Signal Processing and Information Technology*, pages 354–359. IEEE, 2010.

[110] Dennis Sundman, Saikat Chatterjee, and Mikael Skoglund. Distributed Greedy Pursuit Algorithms. June 2013.

[111] Vincent YF Tan and George K Atia. Strong impossibility results for sparse signal processing. *Signal Processing Letters, IEEE*, 21(3):260–264, 2014.

[112] Rahul Tandra and Anant Sahai. Snr walls for signal detection. *Selected Topics in Signal Processing, IEEE Journal of*, 2(1):4–17, 2008.

[113] Howard L Taylor, Stephen C Banks, and John F McCoy. Deconvolution with the l1 norm. *Geophysics*, 44(1):39–52, 1979.

[114] Zhi Tian and Georgios B Giannakis. A wavelet approach to wideband spectrum sensing for cognitive radios. In *Cognitive radio oriented wireless networks and communications, 2006. 1st international conference on*, pages 1–5. IEEE, 2006.

[115] Zhi Tian, Yohannes Tafesse, and Brian M Sadler. Cyclic feature detection with sub-nyquist sampling for wideband spectrum sensing. *IEEE Journal of Selected topics in signal processing*, 6(1):58–69, 2012.

[116] R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (...*, 58(1):267–288, 1996.

[117] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[118] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[119] Michael Unser. Sampling-50 years after shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.

[120] Jonathan Verlant-Chenet, Jonathan Bodart, André Bourdoux, Philippe De Doncker, Jean-Michel Dricot, and François Horlin. Multiband maximum likelihood signal detection based on compressive measurements. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 1466–1470. IEEE, 2012.

[121] Tadashi Wadayama. An analysis on non-adaptive group testing based on sparse pooling graphs. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2681–2685. IEEE, 2013.

[122] Fei Wen, Yuan Yang, Peilin Liu, Rendong Ying, and Yipeng Liu. Efficient lq minimization algorithms for compressive sensing based on proximity operator. *arXiv preprint arXiv:1506.05374*, 2015.

[123] Jinming Wen, Xinen Zhu, and Di-Jie Li. Improved bounds on restricted isometry constant for orthogonal matching pursuit. *Electronics Letters*, 49(23):1487–1489, 2013.

[124] Shujing Xie, Lianfeng Shen, and Jishun Liu. Optimal threshold of energy detection for spectrum sensing in cognitive radio. In *Wireless Communications & Signal Processing, 2009. WCSP 2009. International Conference on*, pages 1–5. IEEE, 2009.

[125] Zhuan Ye, John Grosspietsch, and Gokhan Memik. Spectrum sensing using cyclostationary spectrum density for cognitive radios. In *Signal processing systems, 2007 IEEE workshop on*, pages 1–6. IEEE, 2007.

[126] Jonathan S. Yedidia. Message-Passing Algorithms for Inference and Optimization. *Journal of Statistical Physics*, 145(4):860–890, October 2011.

[127] Juhwan Yoo, Stephen Becker, Manuel Monge, Matthew Loh, Emmanuel Candes, and Azita Emami-Neyestanak. Design and implementation of a fully integrated compressed-sensing signal acquisition system. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5325–5328. IEEE, 2012.

[128] Tevfik Yücek and Hüseyin Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys & Tutorials, IEEE*, 11(1):116–130.

[129] Huazi Zhang, Zhaoyang Zhang, and Yuen Chau. Distributed compressed wideband sensing in Cognitive Radio Sensor Networks. In *2011 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2011*, pages 13–17, 2011.

[130] Shibing Zhang and Zhihua Bao. An adaptive spectrum sensing algorithm under noise uncertainty. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[131] Zhenghao Zhang, Zhu Han, Husheng Li, Depeng Yang, and Changxing Pei. Belief propagation based cooperative compressed spectrum sensing in wideband cognitive radio networks. *IEEE Transactions on Wireless Communications*, 10:3020–3031, 2011.

[132] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.